Game & Puzzle Design Compendium

This volume collects all articles from the first six issues of the Game & Puzzle Design journal.

Articles are published verbatim as they appeared in their original form, with some reformatting for consistency in this single volume.

Compiled by Cameron Browne in 2018.

Issue #1
Vol. 1, no. 1, 2015 ................................................................. 9

Issue #2
Vol. 1, no. 2, 2015 ................................................................. 97

Issue #3
Vol. 2, no. 1, 2016 ................................................................. 185

Issue #4
Vol. 2, no. 2, 2016 ................................................................. 273

Issue #5
Vol. 3, no. 1, 2017 ................................................................. 359

Issue #6
Vol. 3, no. 2, 2017 ................................................................. 443

Purchase: http://www.gapdjournal.com/gapd/issues
Submit: http://www.gapdjournal.com/gapd/submit
Game & Puzzle Design Compendium

Compiled by
Cameron Browne

Collecting the first six issues of the Game & Puzzle Design journal
Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com

Editor-in-Chief

CAMERON BROWNE
Department of Knowledge Engineering (DKE)
Maastricht University
Maastricht, 6200 MD
Netherlands

cameron.browne@maastrichtuniversity.nl

Editorial Panel

INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
CARL HOFF, Applied Materials, USA
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
DAVID PARLETT, Parlett Games, UK
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
SPYRIDON SAMOTHRAKIS, Univ. Essex, UK
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINANDS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent publication prepared and published with the kind help of volunteers. Set in Palatino using LaTeX.

Published content © 2018, Game & Puzzle Design.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.gapdjournal.com/submit
### Vol. 1, no. 1 2015

**Editorial**
Welcome to *Game & Puzzle Design* ............................................. C. Browne 13

**Case Study**
Volo: Bird Flight in a Game .................................................. D. Stein 15

**Case Study**
Heptalion ................................................................. N. Romeral Andrés 21

**Nikoli Logic Puzzles**
Masyu ................................................................. J. Goto 28

**Article**
The Nature of Puzzles ..................................................... C. Browne 33

**Short Note**
Uniqueness in Logic Puzzles ............................................. C. Browne 45

**New from Old**
Influence Maps and New Versions of Risk ............................ D. Ashlock and C. Lee 48

**Geometry at Play**
Symmetrical Pentomino Pairs ............................................. K. Jones 54

**Article**
On the Effects of Biasing Win Conditions .............................. I. Althöfer and M. Hartisch 60

**Short Note**
Serendipity in Game Design ............................................. R. Peeters 66

**Game Design Patterns**
Embed the Rules .......................................................... C. Browne 70

**Case Study**
Centipede: From Arcade Game to Board Game ........................ B. Whitehill 81

**Project GIPF**
Project GIPF: Design and Aesthetics .................................... M. Thomashow 85

**Short Note**
Architecture in an Abstract Game ...................................... K. Burm 90

**From the Archive**
Defining the Abstract ..................................................... J. M. Thompson 93
Editorial
Mixing and Fixing Games ................................................................. C. Browne 101

Case Study
The Double Circle Real 5×5×5 ................................................................. C. Hoff 103

Case Study
Tentai Show ......................................................................................... J. Goto 113

Case Study
Try: A Hybrid Puzzle/Game ................................................................. C. Browne 119

Article
Dameo: A New Step in the Evolution of Draughts? ................................. C. Freeling 126

Case Study
Catch-Up: A Game in Which the Lead Alternates ................................. A. Isaksen et al. 136

Short Note
Coalition Control Through Forced Betrayal ............................................ C. Browne 148

Short Note
From Mathematical Insight to Strategy ................................................. N. Romeral Andrés 151

New from Old
Deriving Card Games from Mathematical Games ................................... D. Ashlock and J. Schonfeld 153

Article
Game Mutators for Restricting Play ...................................................... J. Neto and W. Taylor 162

Game Design Patterns
Explore the Design Space ................................................................. C. Browne 169

From the Archive
What Makes a Game Good? ............................................................... W. Kramer 182
Vol. 2, no. 1 2016

Editorial
Game Design in the Age of AI .............................................. C. Browne 189

Case Study
Wrap O-round Weave Five .............................................. C. Hoff 191

Nikoli Logic Puzzles
Shakashaka .............................................. J. Goto and Y. Miyagi 199

Case Study
Eco-Friendly Game Design .............................................. N. Romeral Andrés and C. Browne 203

Article
Characterising Score Distributions in Dice Games ..................... A. Isaksen et al. 210

Article
Narrative Progression Traits for Role-Playing Games .................... S. Samothrakis 224

Game Design Patterns
Bug or Feature? .................................................. C. Browne 232

Article
Elegant Combat in War Games ........................................ F. M. Marqués 240

Maths in Games
Graph Theory in Game and Puzzle Design ................................ D. Ashlock 248

Case Study
Automated Playtesting with RECYCLED CARDSTOCK .......... C. Bell and M. Goadrich 257

From the Archive
Under the Strategy Tree ........................................ R. Abbott 270
Editorial
Nothing New Under the Sun. ...................................................... C. Browne 277

Short Note
Diamond: Improving on a Known Design. ................................. L. Back 279

Case Study
From Computer Operations to Mechanical Puzzles ..................... M. O. van Deventer and I. Kriz 283

Case Study
Physics Laws as Game Rules ................................................. F. M. Marqués 291

Game Design Patterns
Make the Design do the Work ................................................. C. Browne 301

Case Study
How to Make a Better $3 \times 3 \times 3 \times 3$ ........................................ C. Hoff 315

Article
Mitigating Non-Strategic Coalitions ........................................ C. Duncan 329

Short Note
The Development of a Tangram Family .................................... F. Horn 336

Maths in Games
Graph-Based Search for Game Design ................................. D. Ashlock and C. McGuinness 342

Short Note
A Game Design Approach to a Real-World Problem ..................... C. Browne 350

From the Archive
What’s a Ludeme? ................................................................. D. Parlett 355
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editorial</td>
<td>Talking About Other People’s Games</td>
<td>C. Browne</td>
<td>363</td>
</tr>
<tr>
<td>Nikoli Logic Puzzles</td>
<td>Herugolf</td>
<td>Y. Anpuku</td>
<td>365</td>
</tr>
<tr>
<td>Article</td>
<td>Semantic Hashing for Video Games</td>
<td>A. Isaksen, C. Holmgård and J. Togelius</td>
<td>370</td>
</tr>
<tr>
<td>Maths in Games</td>
<td>Limping Boards for Games</td>
<td>C. Browne</td>
<td>377</td>
</tr>
<tr>
<td>Case Study</td>
<td>The Complex 3×3×3</td>
<td>C. Hoff</td>
<td>385</td>
</tr>
<tr>
<td>Short Note</td>
<td>From Mathematical Proof to Puzzle</td>
<td>N. Romeral Andrés</td>
<td>399</td>
</tr>
<tr>
<td>Maths in Games</td>
<td>CliqueR: A Graph Theory Game</td>
<td>D. Ashlock and A. McEachern</td>
<td>402</td>
</tr>
<tr>
<td>Game Design Patterns</td>
<td>Reinvent the Wheel</td>
<td>C. Browne</td>
<td>411</td>
</tr>
<tr>
<td>Article</td>
<td>Games for Three Players</td>
<td>J. Neto and W. Taylor</td>
<td>423</td>
</tr>
<tr>
<td>From the Archive</td>
<td>Some Random Thoughts On Chance and Skill</td>
<td>D. Parlett</td>
<td>434</td>
</tr>
</tbody>
</table>
Vol. 3, no. 2  2017

Editorial
Back to Basics ................................................................. C. Browne  447

Nikoli Logic Puzzles
Stained Glass ................................................................. Y. Anpuku  449

Case Study
Disthex: A New Twist on Hex ........................................ R. Ishii and C. Browne  454

Maths in Games
Groups in Circle Puzzles .................................................. W. Kretschmer  459

Case Study
From Untouchable 11 to Hazmat Cargo ............................... C. Hoff  471

Game Design Patterns
Ludoku: A Game Design Experiment ................................... C. Browne  479

Maths in Games
Edit Games ......................................................................... D. Ashlock and A. McEachern  491

Article
Measuring Drama in Snakes & Ladders ................................. J. Neto and J. N. Silva  500

Article
Dice Design Deserves Discourse .......................................... S. Yermolaieva and J. A. Brown  508

Article
Tension in Puzzles .............................................................. C. Browne  515

From the Archive
Games and Politics ............................................................. R. Garfield  523
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editorial</td>
<td>Welcome to <em>Game &amp; Puzzle Design</em></td>
<td>C. Browne</td>
<td>3</td>
</tr>
<tr>
<td>Case Study</td>
<td>Volo: Bird Flight in a Game</td>
<td>D. Stein</td>
<td>5</td>
</tr>
<tr>
<td>Case Study</td>
<td>Heptalion</td>
<td>N. Romeral Andrés</td>
<td>11</td>
</tr>
<tr>
<td>Nikoli Logic Puzzles</td>
<td>Masyu</td>
<td>J. Goto</td>
<td>18</td>
</tr>
<tr>
<td>Article</td>
<td>The Nature of Puzzles</td>
<td>C. Browne</td>
<td>23</td>
</tr>
<tr>
<td>Short Note</td>
<td>Uniqueness in Logic Puzzles</td>
<td>C. Browne</td>
<td>35</td>
</tr>
<tr>
<td>New from Old</td>
<td>Influence Maps and New Versions of Risk</td>
<td>D. Ashlock and C. Lee</td>
<td>38</td>
</tr>
<tr>
<td>Geometry at Play</td>
<td>Symmetrical Pentomino Pairs</td>
<td>K. Jones</td>
<td>44</td>
</tr>
<tr>
<td>Article</td>
<td>On the Effects of Biasing Win Conditions</td>
<td>I. Althöfer and M. Hartisch</td>
<td>50</td>
</tr>
<tr>
<td>Short Note</td>
<td>Serendipity in Game Design</td>
<td>R. Peeters</td>
<td>56</td>
</tr>
<tr>
<td>Game Design Patterns</td>
<td>Embed the Rules</td>
<td>C. Browne</td>
<td>60</td>
</tr>
<tr>
<td>Case Study</td>
<td>Centipede: From Arcade Game to Board Game</td>
<td>B. Whitehill</td>
<td>71</td>
</tr>
<tr>
<td>Project GIPF</td>
<td>Project GIPF: Design and Aesthetics</td>
<td>M. Thomashow</td>
<td>75</td>
</tr>
<tr>
<td>Short Note</td>
<td>Architecture in an Abstract Game</td>
<td>K. Burm</td>
<td>80</td>
</tr>
<tr>
<td>From the Archive</td>
<td>Defining the Abstract</td>
<td>J. M. Thompson</td>
<td>83</td>
</tr>
</tbody>
</table>

The front cover shows a rendering of a Volo game in action, by its designer Dieter Stein. Volo is described in the article ‘Volo: Bird Flight in a Game’ on pages 5–10.

All content © 2015, *Game & Puzzle Design* and QUT.
Credits

Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high quality work on all aspects of game and puzzle design.

Two issues per year, published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097

http://www.cameronius.com/gapd

Editor-in-Chief
Cameron Browne
Electrical Engineering and Computer Science
Science and Engineering Faculty
Queensland University of Technology
Brisbane, Queensland, 4001
Australia

c.browne@qut.edu.au

Liaison Officer
Néstor Romeral Andrés
Nestorgames
Spain

Copy Editor
Russ Williams
Green Star
Poland

Editorial Panel

INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
NICK BENTLEY, Gr. Yellowstone Coalition, USA
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
DAN TROYKA, Conlin, McK. & Phil., USA
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINANDS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent biannual publication jointly supported by the School of Electrical Engineering and Computer Science (EECS) and the Office of Research, Queensland University of Technology (QUT), Brisbane, Australia. Set in Palatino using \LaTeX.

Published content © 2015, Game & Puzzle Design and QUT.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.cameronius.com/gapd/submit
Welcome to Game & Puzzle Design

Cameron Browne, Queensland University of Technology (QUT)

The term ‘game design’ seems to have shifted in meaning recently. As the digital entertainment industry booms and gains traction in our daily lives, so has game design correspondingly grown as a profession and research topic, with dedicated curricula at many – if not most – universities worldwide. Current research understandably focuses on those games with the greatest impact; video games, and especially big budget titles from major studios. But these are complex beasts with many aspects of design: conceptual art, character development, 3D modelling, narrative flow, motion capture, sound design, game engine architecture, code optimisation, level design, player analytics, and so on.

This journal instead focuses on the core of ‘game design’ in a traditional sense, namely the study of the underlying mechanisms of games and puzzles relevant to play. From the list above, ‘level design’ would be the most relevant aspect here. This is not to say that the journal will only focus on traditional games and puzzles, even if these are the instances that typically reveal their underlying mechanisms most clearly. I personally am particularly interested in the overlap between traditional and digital game design, and where these disciplines converge and diverge, especially when it comes to the design of games and puzzles for mobile platforms.

The smartphone revolution over the last decade has resulted in hundreds of thousands of entries in the iOS and Android app store ‘Games’ categories, at an average rate of over one hundred new games per day, with little or no real quality control. Understanding these games in terms of their underlying mechanisms of play, stripping away theme or ‘skin’ to reveal the core game underneath, could allow us to compare them objectively, and help identify the truly original from the variants and clones.

Game & Puzzle Design is not targetted at one specific readership (academics, practitioners, consumers, etc.) but seeks to find a common ground among all of those interested in the topic. The main criteria for submissions – apart from logical rigour – is their potential to interest and inform readers, and teach them something new about game and puzzle design. This first issue contains a variety of pieces, by authors from a range of backgrounds, which demonstrates the inclusive nature of the journal.

These include case studies, such as Dieter Stein’s account of capturing the natural beauty of bird flocking behaviour in his game VolO, Néstor Romeral Andrés’s Dominoes-inspired game Hep talion and the problem solving process he applied to create it, the Japanese logic puzzle Masyu in the words of its publisher, Kate Jones’s account of games made by pairing symmetrical pentominoes, and Bruce Whitehill’s historical reminiscences of adapting the arcade game ‘Centipede’ to a board game in the 1980s.

There are technical pieces that use computer modelling to refine the rules for a new version of the board game Risk (Ashlock and Lee), and to demonstrate the complexities of biasing the win conditions in games with more than one goal (Althöfer and Hartisch). There are philosophical pieces, such as Raf Peeters’s account of the importance of serendipity in the design process, Mitchell Thomashow’s personal ode to the GIPF series of abstract board games, and GIPF designer Kris Burm’s musings on the correspondences between games and architecture.

This issue also includes some of my own pieces that explore higher level principles of game and puzzle design. These include the use of authorial control in puzzle design to phrase solitaire puzzles as games played between the setter and the solver, the importance of logic puzzles having unique solutions, and the benefits of ‘hiding the rules in the equipment’ when designing games.

The issue concludes with a reprint of J. Mark Thompson’s classic article ‘Defining the Abstract’, a brilliant example of clear thinking that has had a profound impact on many people’s understanding of games and puzzles. We will reprint such classic pieces, or their reworkings in a modern context, under the column From the Archive.

There are several correspondences between the pieces in this issue. Stein’s inspiration of bird flocking behaviour as the basis for a game epitomises Peeters’s assertion that designers should always be looking for new ideas to pursue. Romeral Andrés points out the importance of poka-yoke (mistake-proofing) in game design, while I take this idea and run with it in my piece ‘Embed the Rules’. Peeters and Jones describe different ways to extend standard pentominoes to derive new paradigms for games. Burm’s analogy of games as architecture may shed some light on Thomashow’s description of the GIPF series as a cohesive whole.

The journal will host a number of regular columns. While these may be driven by particular authors who have made a commitment to perpetuating them, it should be made clear that those authors do not ‘own’ those columns as such. Any author can submit pieces under the banner of any regular column, and each issue may contain more (or fewer!) than one piece for each column, from different authors. We welcome suggestions for new columns by contributors.

As an added bonus for readers, a ‘feature puzzle’ will be selected each issue, and sample challenges printed throughout the issue where space permits. The first feature puzzle is Heptalion, the rules of which are described on page 17.

The main purpose of Game & Puzzle Design is to provide a venue for high quality work on the topic, and to encourage the exchange of ideas. We look forward to seeing the direction it will take, as traditional and digital forms of games and puzzles continue to converge, and the art and science of their design becomes better understood. The editorial team, editorial panel and I hope that you enjoy Game & Puzzle Design.

Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design. 

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au
Volo: Bird Flight in a Game

Dieter Stein, Spielstein – Spiele & Rätsel

This article describes the development of the board game Volo, based on the natural beauty of bird flight (il volo in Italian). It shows how a natural phenomenon can serve as an inspiration, and how a chosen theme can promote and influence creativity during the design process.

1 Introduction

In the summer of 2010 I was working on a game design which would further exploit the ordo manoeuvre. This is the connected movement of aligned pieces used in my game Ordo, which in turn was based on the phalanx movement in Robert Abbott’s classic Epaminondas.

For example, Figure 1 shows an ordo manoeuvre in which three pieces in a line move simultaneously as a group. It suddenly came to mind that birds create a similar linear, dynamic formation when flying in flocks, as do sheep in herds, fish in schools, and many other types of animals.

![Figure 1. The ordo manoeuvre.](http://spielstein.com/games/ordo)

1.1 Inspiration

One can appreciate the way flocking birds fly near each other while simultaneously maintaining a safe distance, even if one is not a birdwatcher. Huge flocks of birds are pure beauty and leave us deeply impressed, even touched by this natural spectacle. The large number of birds, each following simple rules, results in a higher order organisation, which appears to act as a new entity, almost an organism of its own. I wanted to design a game that captured this beautiful phenomenon.

A flight formation often seen, especially for larger birds, is the V formation shown in Figure 2. The straight lines followed by the birds, and their simultaneous movement with their flockmates, bore a striking resemblance to the ordo manoeuvre. So what natural rules lead to this behaviour?

![Figure 2. Birds flying in V formation (photo by Henry J. Hipp).](https://www.flickr.com/photos/albertovo5/4113467727/)

In his classic 1987 paper, Craig Reynolds describes three basic rules that individual birds within flocks appear to follow:

1. **Collision Avoidance**: Avoid collisions with nearby flockmates.
2. **Velocity Matching**: Attempt to match velocity with nearby flockmates.
3. **Flock Centring**: Attempt to stay close to nearby flockmates.

From a game design perspective, it was clear that the second rule should be dropped, as incorporating velocity and timing would add undue complexity and ill fit an abstract game. The other two rules, however, were a natural fit for discrete combinatorial play. This paper describes the development of the resulting game, called Volo, based on the natural beauty of bird flight.

---


Volo is played on a hexagonal board with 120 spaces. Two players each start with three ‘birds’ (pieces) in the ‘nests’ at the corners, then take turns either:
1) entering a new bird at an empty point, or
2) moving one or more birds grouped in a line.

The aim is to connect all friendly birds into one contiguous ‘flock’ of any size.

Birds may not be entered directly adjacent to other friendly birds, and may only move (‘fly’) to join other birds if they enlarge another friendly flock. Parts of flocks may be moved to make new groups, but groups, once formed, may never be split. Players are therefore constantly forced to enter new birds in order to progress towards the goal.

If an opponent’s birds are isolated, dividing the board into multiple regions, then the trapped birds are removed at the active player’s choice, such that only one single region with the opponent’s birds remains. Isolated regions cannot be entered by the opponent.

2 Design

This section describes the design process that followed, starting with the basic idea of flock-like piece movement.

2.1 Board

It was immediately obvious that the rectangular grid was not suitable for creating attractive bird-like V patterns. Right angles are too regimented for such a natural movement (Figure 3).

But my second choice of tiling – the hexagonal grid – proved to be perfect. It not only yielded attractive, natural lines for the V formation, but also allowed a sky-like circular game space (Figure 4).

Figure 4. Angles on the hexagonal grid.

2.2 Goal

Now that the grid and basic board shape were decided, I turned to the goal of the game. The movement rules, together with the theme, pointed to an obvious choice: since flocking behaviour was the inspiration for the game, it made sense to bring the birds together into a single flock.

There were surprisingly few games at the time that featured both piece movement and a goal of gathering. One that immediately comes to mind is Claude Soucie’s modern classic Lines of Action from 1969 [3]. But there were not many others, until some recent examples started to emerge such as Ayu [5] Feed the Ducks [4] and Unity [5]. It seems that movement to cluster is a game idea whose time has come.

2.3 Start

The next step in the design process was to choose a suitable setup and appropriate movement rules. One setup option would be to start with all pieces on the board, like the games listed above. Another option would be to bring the pieces into play from off-board, either allowing them to group instantly or stipulating that they are entered at isolated points – or at least not adjacent to any friendly pieces – before the flocking takes place, as shown in Figure 5. This second option was chosen, which recreates Reynolds’ first rule of flocking (Collision Avoidance), making Volo a game with piece movement.

---

At this point, board size came into play. I wanted a large board, to add more strategic action to the game, and to give the relatively expansive flocking formations some space in which to unfold. Moreover, the possibility of long lines of birds moving simultaneously and creating flock changes on a relatively large scale was appealing.

The large board, reminiscent of an open, empty sky, and the expected high number of bird pieces, strengthened the decision that pieces should start off-board and brought into play over the course of the game. However, the rejected setup option left a trace: the game would begin with three ‘early birds’ for each player, spaced as far apart as possible in the corners of the board. To mark these special places, six cells were removed at the corners of the board and the starting points shifted one cell inwards, resulting in distinctive positions which I immediately called ‘nests’. Here again, a personal choice was made: introducing the nests gave the board a pleasing snowflake shape, similar to that of Kris Burm’s YINSH, which added something to the character of the game. Even designers with goals like elegance through minimalism in mind are sometimes tempted to add tiny ornaments, although not without a cause (or excuse).

Similarly, a symmetrical setup was an obvious choice to please the eye. This required another change, so that birds did not simply connect across the board to win in a few moves, which led to the removal of the central cell. It was not a perfectly satisfying decision to ‘drill a hole’ into the sky, but the desire for symmetry won. Figure 6 shows the final board design and starting position.

2.4 End

Standard game design principles were then applied. Termination rules often need to be explicitly stated, to ensure that progress is made and that the game moves towards completion. Simple games with small rule sets often have termination rules already ‘built in’. But with more complex rule sets, it is often necessary to add further rules to make the game robust.

In Volo’s case, this crucial termination rule, added on a subsequent step, proved to be: movement must always result in the creation of a larger flock. Every move must connect to birds which are members of another separate but friendly flock.

2.5 Flocking

It is also forbidden to split flocks once they have been formed. This ensures that flocks are always growing, and that sooner or later players will run out of movement actions and be forced to release another bird into the sky. Figure 7 shows two legal moves using these rules, which create patterns reminiscent of the V formation of flying birds.

The question then arose whether to let players move entire flocks in their V shaped patterns, to reinforce the analogy of flight, rather than single-line ordo moves. I decided against this for three reasons: 1) moving two lines of pieces at once blew out the complexity of some moves too much; 2) it raised the inelegant possibility of flocks flying backwards; and 3) it just proved too clumsy to manipulate two lines of pieces in a move. It is enough that V shapes form naturally during play.

---

6http://www.gipf.com/yinsh
7Game designer Néstor Romeral Andrés later suggested restoring the central cell, and starting with a neutral piece enclosed by the six early birds. This worked, but was not followed up as the game had already been published.
Figure 7. The movement rules create patterns similar to the V formation of flying birds (Figure 2).

Figure 8 shows a typical game position. The marked flock of three birds at the bottom left has five available moves: either 3 or 4 steps upwards, creating flocks of four or six birds, or single file to the upper right to connect with a single friend. Opponent birds block all other move options.

This two-step feature of entering birds and then flocking to them leads to an interesting twist, as the player must make moves that conflict with the goal in order to progress: they must enter single birds that can only form flocks later.

This results in considerable tension, as players are repeatedly faced with the dilemma of how to enter new birds – which increase their movement potential – so as to minimise the damage to their own position. Further, the mover must also consider how much strategic information a new bird’s position discloses to the opponent, as the opponent can step in to block obvious connections. On the other hand, trying to hide your plans can lead to stray birds that are hard to catch.

Figure 9 shows a board position (light player to move) that demonstrates tactical play. Moving bird A to 1 would be a decisive move for the dark player, hence the light player must enter a bird at 2 to block this move. But dark bird B can then move to 3, and the light player cannot stop the winning flight of dark bird A to 4, to create a single dark flock and end the game.

Figure 9. An example of Volo tactics.
3 Refinements
The game seemed almost complete, but a problem emerged: as more birds are entered, it is possible to isolate the opponent’s birds so that they can never reach a winning position, as long as they remain trapped. The obvious fix – allowing birds to fly over occupied cells – would ruin the game, as blocking is a substantial tactical feature in Volo. Further rule refinement was needed.

3.1 Regions
This problem was solved by adding the notion of regions to the game. If the player creates a region containing opponent’s birds who have no path to any of their friends, then those isolated birds are removed. Accordingly, players cannot enter a bird into an empty isolated region.

Figure 5 shows these rules in action. The light player has created three regions (A, B and C) and must choose two from which to remove all opponent’s birds. If region B is left alive, then its single bird would give the dark player the win.

Figure 11 shows such a (rare) situation, in which both players can only enter birds into their own regions, so will both pass for a draw.

4 Epilogue
It turned out that bird flight was not just a metaphor, but fit the core of the game precisely. Adding some backstory about the game’s inspiration to the official rule sheet hopefully projected some of the beauty of the natural world onto the game, to help players ‘feel’ what is going on as they play, and to fully appreciate Volo.

As a designer of combinatorial (often called ‘abstract’) games, I am often asked how I get my
ideas, how things get started. It is difficult to imagine how a game concept is developed, which on the surface may appear simple, but which offers players plenty of tactical and strategic options. I would even say that it is impossible to design a game in its whole strategic depth.

If a game designer is seen as a storyteller who creates some kind of interactive story, it seems easier to comprehend why choices are made. The lines of development regarding progress and excitement in a story are often already set up, and it is in the hands of the designer to find interesting mechanisms (i.e. rules), which match the context and fit together to create a satisfying and joyful experience when the game is played.

Like the storyteller, the game designer’s goal is to give the audience an experience, but it is more difficult to find the key to the ‘story’ in this field. Fortunately, the game designer as a human being is equipped with experiences and knowledge of the real world, and it happens that sometimes the abstract and the natural coincide beautifully.

Some may point out that simplicity is beauty, and boils down to mathematics in the end. But it is not just pure mathematics which leads to a correct solution when inventing games; the designer needs intuition and to not fear making decisions.

5 Conclusion

This case study describes the process of turning an observed natural phenomenon, the flocking behaviour of birds, into a board game. The resulting game, Volo, is published by nestorgames and can be played online at Boardspace.net.

Acknowledgements

Thanks to Cameron Browne for encouragement and revisions, the anonymous reviewers for their helpful suggestions, and Henry J. Hipp for permission to use his ‘Snow Geese’ photo.

References


Dieter Stein is a system administrator and developer, who has been devising abstract games since his youth. His game Mixtour received a Recommendation from Spiel des Jahres in 2013.

Address: Haxthausen 6, Freising, Germany.

Email: dieter@spielstein.de

---

8http://www.nestorgames.com/#volo_detail
9http://boardspace.net/english/about_volo.html
Heptalion

Néstor Romeral Andrés, Nestorgames

Heptalion is a simple combinatorial tile placement game, and associated set of puzzles, designed in 2011. This paper describes the various design problems that arose when trying to create a child friendly game that adults would also enjoy, and the steps I took to resolve them.

1 Introduction

I present a combinatorial abstract strategy game and its derived puzzles. I will focus on how ‘keeping it simple’ revealed unexpected problems and how these were tackled, while finding interesting design techniques along the way.

I describe my design goal first, instead of following an exploratory process, and the mechanisms, components and victory condition naturally followed. It is shown how design constraints can lead to the discovery of interesting mathematical properties and new combinations of components and mechanisms. Finally, the game is transformed into a puzzle, which turns out to be as interesting as the game itself.

2 History

I designed Heptalion in 2011 and published it through my company Nestorgames in 2012 [1]. The published set is shown in Figure 1. A version for Android smartphones was released in 2014.

Heptalion: Players draw an equal number of tiles from the bag at random, and place them face up for all to see. Players then take turns placing one of their tiles on the board, face down, to cover a pair of symbols that match the two symbols on the tile. Last player to move wins.

Heptalion has its roots in 2010, when another of my games, Hippos & Crocodiles [2], was rivalling the sales of my most successful game Adaptoid [3]. Hippos & Crocodiles is very simple. Players take turns placing one of their animal-shaped tiles onto the board, either a hippopotamus or a crocodile (Figure 2), until one can not make a legal move.

Realising that the complexity of a game is not correlated with its market success, I decided to design another game that was even simpler than Hippos & Crocodiles, to see if its success could be reproduced. But this time I wanted to focus on the parents, being the ones who buy the game and have to play with their kids, but have no time to play it themselves. I wanted to create a game that was easy and quick to learn and play, mistake-proof, short and replayable.

With these objectives in mind, the first step was to create a list of design goals, consisting of a set of obvious design problems and potential solutions.

3 Design Problems

The following list describes the design problems I faced when creating Heptalion, and their solutions:

1. Problem: Steep learning curve and resistance to change on behalf of players.
Solution: Use an already existing and well-known game component (e.g., cards, pawns, dice, etc.) so that players feel comfortable with it.

2. Problem: Too many rules force players to check the rulebook frequently, and also may lead to misinterpretation and conflict.

Solution: Move the rules to the components, in a *poka-yoke* approach, so that mistakes and misinterpretations are almost impossible. *Poka-yoke* is a Japanese term that means ‘mistake-proofing’, and refers to any mechanism in a manufacturing process that helps an equipment operator avoid (yokeru) mistakes (poka), by drawing attention to human errors as they occur [4]. More broadly, the term can refer to any behaviour-shaping constraint designed into a process to prevent incorrect operation by the user.

3. Problem: Game length may scare young players and parents.

Solution: Keep the number of turns low (around 10–15), and also the duration of each turn, to avoid ‘analysis paralysis’.

4. Problem: To accommodate several players.

Solution: Make it multiplayer without extending the duration of the game, through a finite number of pieces shared among all players.

5. Problem: The set should be affordable.

Solution: Use a small and cheap-to-manufacture set of components.

6. Problem: Allow a high degree of replayability, so the players get value for money.

Solution: Random setup based on a combinatorial distribution of the components.

7. Problem: Games that are easy to make at home lose sales.

Solution: Use components that are not usually found at home.

Note that item #7 is not a design problem so much as a business problem. However, I have to take such considerations into account when designing, since I am also the publisher. Many amateur game designers seem to ignore such issues when designing their own games, even though they can be very important to publishers.

4 Problem Solving Process

This section describes the problem solving process used to tackle the problems listed above. The key to most of these was to find a magical game component that:

- is widely known (solves problem #1),
- does not have many parts (solves problem #5),
- is sufficiently complex (solves problem #6), and
- is hard to replicate if customised (solves problem #7).

A standard Dominoes set satisfies these criteria nicely. A set of standard dominoes has 28 tiles, showing all possible pairs of numbers from 0 – 6. Although a larger set could have been used for my game, 28 tiles proved to be sufficient. One tile per turn gives each player 14 turns in a two-player game, which solves problem #3.

For a four-player game, each player has only seven turns. This is a bit small, but the game nature encourages players to play several games in a row, which is a desirable feature in games, showing that the game is appealing and replayable.

So problem #4 was partially solved. In order to fully solve it, I released an expansion pack called Octalion that uses a larger board and increases the number of tiles to 36 (so that each player has nine turns in a four-player game).

Octalion also solves problem #7, as a set of 36 dominoes is nonstandard and thus difficult to find. The design hurdles seemed to be dropping quickly so far, except for problem #2; trying to solve it was like trying to open a Matryoshka or Babushka doll, in which solving each layer just revealed further problems to the solved.

I then focused on the victory condition, which for me is the meaning of a game, and the most important part of the rules. The components are the tools that the players use to achieve the goal.

I decided to use the same victory condition as many other successful tile placement games: the last player able to make a move wins. This is a very powerful victory condition, as it does not need to be checked every turn (like a checkmate in Chess) and avoids the need for scoring; the game simply ends when it is not possible to play anymore.

It made sense that players would start with the tiles split amongst them, which is simpler than players needing to draw a tile each turn, or keep track of the number of tiles in hand, etc. Players would take turns placing a tile on the board only where it could fit (which feels natural), but they must be forced to choose from several places to fit them (otherwise it would be boring).
The ideal board would fit all tiles, and allow each tile to be placed in several different places. Each move should eliminate some other potential moves, so that the number of legal moves reduces and the game converges to a result that is not trivially predictable. Moreover, the shape of the board must be roughly rectangular to fit an A4 sheet, as this is the size of the boards that I print.

A 28-tile set of Dominoes contains 56 half-tiles (i.e. squares) in total, so I first tried using a 7×8 grid of 56 squares for the game board. This proportion roughly matched the shape of an A4 sheet, with some room for the game title, but there was a problem.

A 7×8 grid has the desired 56 squares, but it has \((6 \times 8) + (7 \times 7) = 97\) pairs of orthogonally adjacent squares, as shown in Figure 3. These represent the places that double-square Dominoes tiles can be placed. But in order to have the same number of available placements for each tile (for a balanced distribution), this total should be a multiple of 28.

The nearest multiples of 28 are \(3 \times 28 = 84\) and \(4 \times 28 = 112\). As 7×8 allows the maximum number of pairs, it is impossible to reach 112 with 56 squares, hence 84 pairs became the target number. The problem can now be stated as:

*Find an orthogonal grid shape with 56 squares and exactly 84 adjacent pairs.*

Having no idea what shape such a grid might take, I started with a 7×8 rectangle and began moving squares around. Moving a border square to another place of the perimeter (except corners), reduces the number of pairs by 2 (-3 for the removal and +1 for the placement), as shown in Figure 4.

But it is not possible to reach an even number by repeatedly subtracting 2 from an odd number, so I then tried the corners. Moving a corner square to another place on the perimeter (except corners) reduces the number of pairs by 1 (-2 for the removal and +1 for the placement), as shown in Figure 5.

This meant that 13 squares would have to be moved, to reach the target number of 86 pairs. But there was a faster and more flexible way; moving inner squares to the perimeter reduces the number of pairs by 3 on each movement (-4 for the removal and +1 for the placement), as shown in Figure 6.
Figure 6. Moving an inner square to a side subtracts three pairs overall.

Further, removing a square adjacent to a hole and moving it to the perimeter reduces the number of pairs by 2 (-3 for the removal and +1 for the placement), as shown in Figure 7, which is conveniently an even number.

Figure 7. Moving a square adjacent to a hole to a side subtracts two pairs overall.

Playing around with these square movements for a few days, I came across the diamond shape shown in Figure 8. This shape was perfect; it was symmetrical, appealing, and worked well within an A4 page ratio.

Figure 8. The diamond.

However, finding the right shape was only part of the solution, and the actual distribution of symbols within this shape posed a new problem:

Find a distribution of numbers 0–6 within the diamond shape, such that each occurs exactly eight times, and each pair of numbers (28 in total) occurs exactly three times (3 × 28 = 84).

This new problem was a hard one for a non-programmer, and brute force enumeration would be too time consuming, so it had to be attacked from a different perspective. I looked for simpler patterns that might occur in a valid distribution, hoping that this easier task would allow me to build the board.

The breakthrough came when I considered the A-A tiles that contain matching numbers, i.e. 0–0, 1–1, 2–2, etc. We want each A-A tile to have three possible placements, the same as any other tile, and it turns out that this can be achieved efficiently if each number occurs in one of the patterns shown in Figure 9.

Figure 9. Tetrominoes with three adjacent pairs.

These shapes, called tetrominoes, are used in many games and puzzles, such as the well known LITS puzzle from Japanese publisher Nikoli [2]. Note that this set only includes tetrominoes with exactly three adjacent cell pairings, and excludes the fifth 2×2 tetromino with four adjacent cell pairings; the fewer the better in this case.
The trick was to then place seven of these tetromino patterns inside the diamond grid, and number the corresponding grid cells accordingly. After a few hours’ work with the help of a spreadsheet, I found the final distribution shown in Figure 10. The numbers were replaced by coloured symbols (Figure 11), and the game was ready for release. It has since proven popular with players, and does not show any obvious first or second player advantage.

5 Other Solutions

Shortly after releasing Heptalion, Nestorgames customers Mark Tilford and Grant Fikes used computer analysis to find other valid distributions for the diamond shape, and also other shapes with valid distributions. Two of these, shown in Figures 12 and 13, have since been released as expansions for the game.

6 Android Puzzle App

A few months later, Kris J. Wolff (designer of Pilus [7]) proposed developing an Android version of Heptalion. Kris had previously developed an Android app for my game Red [3] but that

---

1 The chosen symbols are colour neutral, so that colourblind players can still enjoy the game: poka-yoke again!
app lacked something important: a set of puzzles. We therefore included some puzzles in the Heptalion app.

The rules for the Heptalion puzzle emerged naturally from the board game. The aim is to place a subset of the Heptalion pieces, according to the game’s rules, to exactly cover a precalculated board shape. This is different to the board game, as all pieces in the solvers’ hand must be placed in order to complete each challenge.

The app creates challenges involving between 3–19 randomly placed pieces such that each challenge has a unique solution. Since not all pieces are included in a challenge, there are usually some unplayable adjacent board spaces, and this adds a new twist in the deduction process for the player. The algorithm for creating challenges is described in Appendix A.

The difficulty of each challenge is estimated as the product of the number of ways each piece can be placed in the initial challenge. For example, the challenge shown in Figure [14] has a difficulty score of 983,040, which makes it of medium difficulty.

6.1 Heptalion Puzzles

Figure [15] shows a Heptalion puzzle challenge using a complete set of 28 tiles. This instance is from [5, p. 85] where it is called ‘Match the Chart’ and uses a standard Domino set, highlighting the intimate link between Heptalion and Dominoes.

Unfortunately, this challenge has at least two solutions (middle and right of Figure [15]), making it solvable by guesswork but not pure deduction, and hence less interesting to myself and players. For this reason, Heptalion puzzles are designed with some pieces removed to have unique solutions and increase their degree of deducibility.

7 Conclusion

In finding a board of the appropriate shape and size for a certain set of tiles, acting under a certain set of constraints, I had to solve a number of design problems in order to achieve the game that I wanted. This paper describes the design steps that led to my game Heptalion and the associated Android puzzle app.

I believe that the techniques discovered along the way – the application of poka-yoke to board game design; the use of maths to tweak the board shape and define the exact subproblems to be solved, and so on – will help with the development of future games.
References


Appendix A: Puzzle Generation

This Appendix describes Kris J. Wolff’s algorithm for creating unique Heptalion puzzles in his Android app for the game.

1. Start with an imaginary 11x11 board.
2. Choose between 3 and 19 pieces to use. The most interesting and difficult puzzles seem to be those with 17, out of the 28 available.
3. Place a piece in a random location.
4. Place each other piece in a random location, such that its two squares each touch at least two squares on pieces already placed. This avoids single square protrusions which would have trivial instantiations.
5. Run through all possible ways to place the chosen pieces on the board. If more than one solution exists, discard and restart.
6. Check that each piece has more than one valid placement (otherwise its placement is trivial). If any pieces have only one valid placement, then discard and restart.
7. Each level is given a ‘difficulty’ rating, calculated as follows: start with difficulty=1, then for each piece multiply the difficulty by its number of valid placements.

Néstor Romeral Andrés is the owner and director of independent board game publisher Nestorgames. He has designed over 70 board games and was awarded Spanish Game Designer of the Year for 2009–10. Address: c/Mar de las Antillas 6, 3-2, Zaragoza, 50012, Spain. Email: info@nestorgames.com

Heptalion Challenge #1

You will find pen & paper Heptalion challenges scattered throughout this issue, where space permits. Use all tiles to cover all symbols. Challenges generated by Kris J. Wolff’s Heptalion app.

```
Challenge

Pieces

Solution
```
Masyu

Jimmy Goto, Nikoli

Masyu is a logic puzzle in which the solver must draw a closed path passing through all printed circles on a square grid, such that the path passes straight through white circles but turns within black circles. This paper describes Masyu and its history.

1 Introduction

Masyu is a pure deduction puzzle that first appeared in the magazine Puzzle Communication Nikoli. It proved interesting and popular, but its design was later improved based on further reader feedback. Like many combinatorial puzzles, its difficulty level can range from very easy to very difficult. Solving Masyu on an arbitrarily large grid has been proven NP-complete.

2 Rules

A Masyu challenge consists of a grid of squares, some of which contain white or black circles. The goal is to draw a single continuous non-self-intersecting path passing through all circles.

The path must go straight through each white circle and must have a right angle turn in the previous and/or next cell. The path must have a right angle turn in each black circle and must go straight through the previous and next cells.

A properly constructed challenge will have a single unique solution. Figure 1 shows an example Masyu challenge and its solution.

The solver can use a variety of reasoning techniques. For example, it is easy to see that a black circle adjacent to the outside border of the grid must have a line extending into the grid interior, as shown in Figure 2(b). Conversely, a white circle at the grid border must have a line going straight through it parallel to the border (c).

The requirement that the path must turn on at least one side of a white circle permits further progress, as shown in Figure 3(c), and the requirement of a single continuous path allows further progress (b) leading to its solution (c).

Figure 1. A challenge (a) and its solution (b).

Figure 2. From challenge (a), paths must extend from black circles (b) and pass through white circles (c).

Figure 3. The path turns at least once per white circle (c).

1http://www.nikoli.co.jp/en/publication/

3 How the Design Evolved

The original idea, from one of our magazine readers, Ryuou Yano, was called ‘Pearl Necklace’ (Shinju no Kubikazari or 真珠の首飾り). The rules were to make a single loop with lines passing through white circles, resulting in (unique) solutions that looked like beautiful pearl necklaces. The rules were simple, using only the white circles, as shown in Figure 4.

Pearl Necklace first appeared in Puzzle Communication Nikoli issue 84, but the rules were so simple that challenges tended to follow the same pattern. When people lost interest in Pearl Necklace, another reader, Asetonitoriru, proposed adding the black circles.

The white and black circles work in opposite ways. White circles disallow turns while black circles enforce turns, as shown in Figure 5. This proved to be an elegant way to allow challenges with the desired variety and difficulty, by introducing a new rule which fit consistently with the rules already established in the original puzzle.

Another key difference with the new movement rule is that the path must extend in a straight line at least one square from the turn. This constraint was found to be necessary to help the solution process, otherwise paths could have too much freedom to be deduced comfortably.

To see what this means in practice, consider again the example challenge shown above. Figure 6(a) shows the deductions that make the first step towards the solution, based on the rules that the loop must:

1. turn within each black circle, and
2. extend in a straight line at least one square from the turn, both coming in and going out.

If this second constraint is relaxed, and the line is allowed to turn immediately, then this only allows the information shown in Figure 6(b) to be deduced from the black circles in the initial challenge. This does not give the solver enough information to progress without undue trial-and-error, which is anathema for pure deduction puzzles. Even worse, it would mean that this challenge would then have more than one possible solution (Figure 7), so would no longer be valid.

Figure 3. Paths must extend or turn (a), and form a single continuous path (b), giving the solution (d).

Figure 4. Pearl Necklace challenge and solution.

Figure 5. White versus black circle moves.

Figure 6. The black line extension rule adds necessary information.
Figure 7. Relaxing the black line extension rule would allow multiple solutions.

Enforcing this second constraint means that deducible and valid challenges can be constructed using fewer circles, allowing more elegant and interesting challenges to be designed.

4 Conclusion

We were happy with improved form of the puzzle and wanted a shorter name for it, and a misreading of the original name in kanji by the company president as ‘Masyu’ (まじゅ meaning ‘evil influence’) made us laugh but stuck.

This improved version first appeared in Puzzle Communication Nikoli issue 90 (March 2000), and there are now two Masyu book collections from Nikoli [2, 3]. Further examples are shown in the Appendices.

Acknowledgements

Thanks to the editors for compiling this piece, and to the reviewers for their helpful suggestions.

References


Appendix A: 10×10 Masyu

This appendix shows a worked 10×10 Masyu example. This challenge, by Nobuyuki Sakamoto, is ‘Puzzle No. 6’ from [3], reprinted with permission.

Figure 8. A 10×10 Masyu challenge.

Figure 9 shows forced path extensions from black circles, cascading inwards from the edges.

Figure 10 shows forced paths through white circles, cascading in from the edges.

Jimmy Goto works for Japanese quarterly magazine Puzzle Communication Nikoli, which specialises in language-independent logic puzzles.

Address: Nikoli Co., Ltd., 4-17-10 Kuramae Taitou-ku, Tokyo 111-0051.

Email: goto@nikoli.co.jp
Figure 11 shows forced path bends adjacent to white circles, due to the rule that at least one path extension must bend on either side of a white circle.

![Figure 11. White circle extensions must bend.](image)

Figure 12 shows forced path continuations that do not create isolated loops. Note the path end marked $a$ in the figure.

![Figure 12. Forced path continuations.](image)

The path end $a$ cannot extend to the right, as it would then be forced to continue through the second (adjacent) white circle, violating the rule that at least one path extension must bend on either side of a white circle. Path end $a$ must therefore extend downwards, leading to the position shown in Figure 13.

![Figure 13. Path end $b$ must continue upwards.](image)

The question now is whether path end $b$ continues upwards or to the right. It cannot turn right, as this would isolate the remaining two open path ends so that they could be closed to form a single continuous path. Hence, path end $b$ must continue straight up, leading to the challenge’s solution (Figure 14).

![Figure 14. Solution.](image)

This example shows typical indicators of a handcrafted – as opposed to computer-generated – design. The initial hints are arranged with a sense of pattern and symmetry. More fundamentally, the solution process itself is quite structured, such as the cascading path extensions through the black and white circles radiating inwards from the edges shown in Figures 9 and 10.

Appendix B: 20×36 Masyu

This appendix shows a more difficult 20×36 Masyu challenge, by Aspilin. This is ‘Puzzle No. 75’ from [3], reprinted with permission.

Again, this example shows strong indicators of a handcrafted design. The visual symmetry of the grid layout of the initial hints is obvious. More importantly, the fact that the circle hints only exist on even rows and columns has the unusual effect of splitting the solution path into two phases (odd and even) which leads to some subtle and interesting solution strategies.

This is an excellent example of the variety of strategies that an experienced designer can incorporate into their Masyu challenges.
Figure 15. A 20×36 Masyu challenge.
The Nature of Puzzles

Cameron Browne, Queensland University of Technology (QUT)

This paper explores the underlying nature of puzzles, and how they relate to games. The discussion focuses on pure deduction puzzles, but with reference to other types of puzzles where appropriate, with examples to support the concepts put forward. These include the notion of puzzles as two-player games between the setter and the solver, the addictiveness of puzzles, and ways in which the setter can exert authorial control, to make challenges more interesting and engaging for the solver.

1 Introduction

Puzzles come in many forms; there are word puzzles, jigsaw puzzles, logic puzzles, dexterity puzzles, physical puzzles, physics based puzzles, to name just a few. While most readers will have an understanding of what the term ‘puzzle’ means to them, the genre as a whole has so far defied exact definition, despite many attempts to do so. But perhaps a precise definition is not all that useful – or even possible – given the variety of puzzles that exist. In this paper, I will look instead at the underlying nature of puzzles rather than attempting to provide yet another definition.

The central thesis of this paper is that most puzzles are games played between the setter and the solver, and that their inherent nature allows sufficient authorial control for the setter to impart their personality upon a well designed challenge, in order to challenge, tease and engage the solver. Several examples are presented in support of this argument, which are mostly taken from actual examples of pure deduction puzzles known as Japanese logic puzzles [6]. These are characterised by having simple rules, a single (deducible) solution, and no language-dependent content. They are not only my favourite type of puzzle, but also illustrate the principles being discussed as clearly and simply as possible.

To avoid confusion in the following discussion, the term puzzle will refer to the actual puzzle game itself, while each instance of a puzzle game presented to the solver will be called a challenge.

2 Puzzles As Games

A puzzle can be defined simply as a task that is fun and has a right answer [2], or more precisely: a question which challenges people to solve, requires their deduction based on its rules to win, and doesn’t depend on chance or other people’s action. [5]

Schuh presents a classification scheme for puzzles, and observes that puzzles can be solved by pure reasoning alone, must have a complete analysis and that you are your own opponent, in the end [4]. While the first two observations are true in most cases and they agree with most other definitions of puzzles, I take issue with the third observation that puzzles are a solitary pursuit undertaken without an opponent.

In his classic paper ‘Defining the Abstract’ (republished in this issue [19]), Thompson makes the astute observation that two-player abstract games may be described as a series of puzzles that the players present to each other. Conversely, I believe that a puzzle may be described as a two-player game played between the setter and the solver. The task of the setter is to produce a challenge that engages and entertains the solver, while the task of the solver is to avoid the traps laid by the setter to complete the challenge.

It is worth emphasising that, unlike a player in a traditional adversarial game, the setter is not trying to ‘win’ against the solver. They are instead trying to provide the most entertaining playing experience, in a role not dissimilar to that of the gamemaster in a role playing game. In the language of game theory, this is not a zero-sum game, as both players can win.

Puzzles are indeed solitary pursuits in a strictly mathematical sense, as the solver is the only agent making actions towards a solution. However, from a strategic or adversarial viewpoint, these actions are directed by the information encoded in the challenge by the setter, which is revealed as the challenge unfolds. The solver may not be an active player during the solution of a challenge, but participates in absentia by influencing the solver’s decisions and actions.

A well designed challenge will include traps and deceptions posed by the setter, which the solver must detect and avoid. In order to see how this works, let’s first look at the concepts of dependency and authorial control in puzzle design.
2.1 Dependency

Dependency in this context refers to the degree to which the steps required to solve a given challenge are dependent on prior steps. A challenge in which progress can immediately be made at many places on the board shows low dependency, while a challenge that only exposes enough information for the solver to make progress at one particular point shows high dependency. The solver exploits that piece of information... which reveals further information... which reveals further information... until the solution is reached.

Figure 1 shows a simple example of this process in action, based on Sudoku challenge #31 from [6] (I assume that readers are familiar with the rules of Sudoku). The 1-hints provide enough information to instantiate another 1 on the top row (a). This additional information allows a 5 to be instantiated in the same row (b), which in turn provides enough information to allow a 9 to be instantiated next to it (c).

The required information is meted out in installments, in a self-perpetuating manner such that each action reveals further information to be acted upon. I have heard this process described as the setter leaving a trail of (informational) bread crumbs to follow. However, I prefer to think of the situation as a tapestry with a loose thread or two, in which the majority of the position is impenetrable except for certain weak points, whose exercise unravels further weak points to follow.

This Sudoku example only provides a superficial instance of this process, as it is an easy challenge with several loose threads to follow. For example, the 8-hints immediately dictate that the central cell must take the number 8. Figure 2 shows a much more striking example, with Slitherlink challenge #80 from [6].

![Figure 1](image1)

![Figure 2](image2)
Slitherlink is a pure deduction puzzle in which a simple closed path must be traced through orthogonal vertices of a square grid, to visit the number of sides indicated on each numbered hint cell [6]. Each adjacent pair of vertices therefore constitutes a move whose value can either be an edge (| or –) or no edge (×).

Figure 2(a) shows the lower left corner of the initial challenge, and (b) shows three edges that must exist where a 3-hint meets a 0-hint. The path thus initiated then bounces off the 0-hints that it encounters (since the path can never visit the side of a zero hint) and moves along the wall to give the position shown in Figure 2(c). A deduction is required at this point; the dotted move can not be an edge as that would cause the path to close prematurely in a cross shape, so it must be no edge × (d). This allows further progress until a deduction is required at position (e), which allows further progress (f) until another deduction is required at position (g), which leads to the completion of the section, as shown in Figure 2(h).

These examples demonstrate how puzzles can hide their own solutions in plain sight, only revealing required information as needed in a self-perpetuating way. Each subproblem requires a solution that provides the next subproblem, and so on. In a well designed puzzle, the solver can almost feel the hand of the setter drip-feeding them information and leading them along by the nose along certain avenues to solution.

Pelánek [8] describes the use of dependency as a metric for automatically measuring the difficulty of Sudoku challenges, based on whether the hints provide enough information to solve the challenge in parallel (i.e. multiple loose threads to follow) or in series (i.e. narrow chain of dependent loose threads). I believe that dependency is a fundamental property of well designed puzzles that runs deeper than just affecting difficulty, as it allows the setter to exert authorial control over the challenges they construct.

2.2 Authorial Control

Authorial control refers to the degree to which the setter can influence the solver’s progress through a given challenge, and manipulate their move choices in absentia. This is the property that makes the setter a second player, in opposition to the solver.

Consider the Killer Sudoku example shown in Figure 3[9]. Killer Sudoku is played according to the rules of Sudoku, except that no hints are provided initially apart from shaded subregions whose component digits must sum to the value shown on each.

Starting with the rightmost column in (a), the 16-region can only contain \{7, 9\} and the 4-region can only contain \{1, 3\}. The latter implies that the 8-region must contain \{2, 6\}, as shown (b). The only combination that satisfies the 13-region is then \{1, 4, 8\}, hence the last two remaining cells of the lower right 3×3 subgrid must have values 3 and 5 (c). The 5 cannot occur in the lower cell of the 9-region, since the only possible completion of this region \{1, 3\} would conflict with the vertical \{1, 3\} just above it, hence this cell must resolve to 3 and its neighbour to 5 (d).

The lower right 3×3 subgrid of this example resolves itself neatly and efficiently, using a minimum amount of information that self-referentially builds upon information released by prior steps. This pattern is unlikely to have occurred by chance, and the solver has the strong sense of an intelligent hand behind its design.
Expert Sudoku players can generally tell whether a given challenge is handcrafted by a human designer or generated by a computer algorithm. Nobuhiko Kanamoto, Chief Editor for Japanese publisher Nikoli, observes that:

Computer-generated Sudoku puzzles are lacking a vital ingredient that makes puzzles enjoyable – the sense of communication between solver and author. [14]

Nikoli have a policy of only publishing handcrafted challenges for their popular line of Japanese logic puzzles, and are sceptical of computer-generated content due to its potential to flood the market with inferior mass product. Challenges may be submitted by amateur fans or experienced designers, but all are hand tested before being approved for publication [11, p. 2].

This communication between setter and solver can only occur if the setter exercises a strong sense of authorial control in their design. For example, consider the computer-generated 6×6 Slitherlink challenge shown in Figure 4(a). Figure 4(b) shows obvious simplifications that an experienced player would immediately spot and complete, while (c) shows the number of obvious simplifications arising from each hint and (d) shows these natural directions of progress for this challenge. This example has multiple starting points and no focused solution path.

Compare this with the handcrafted 6×6 Slitherlink challenge shown in Figure 5(a). This example has only one obvious starting simplification (at the 2 between the two 0s), but it triggers a chain reaction of 84 further simplifications that lead to a complete solution (b) along a few strongly defined directions of progress (c) and (d).

The first challenge may be superficially interesting, as its hints are rotationally symmetrical and it is the more difficult of the two. However, it lacks any underlying strategic structure, and the deductions leading to its solution are homogeneously spread across the board.

The second challenge, on the other hand, has a highly structured solution that unfolds elegantly with each simplification perpetuating the next. It is not symmetrical, nor as difficult to solve, but exhibits a strongly focused sense of authorial control; the solver can feel the hand of the setter and appreciate the craft of the design. This sense of structure tends to be missing from computer-generated designs.

---

2Handcrafted by the author to illustrate this particular point.
2.3 Addictiveness

These mechanisms of dependency and authorial control could go some way to explaining why many players find solving puzzles so addictive. Drip-feeding subproblems to the solver in this manner makes challenges engaging and addictive, as the satisfaction felt at solving each subproblem is a reward that spurs the player on to solve the next, which itself creates more subproblems to be solved.

Stafford explains this effect in terms of a psychological phenomenon known as the Ziegarnik Effect, which refers to the human brain’s tendency to latch onto unsolved problems until they are resolved, with respect to the video puzzle game Tetris [12]. Successful video puzzle games such as Tetris and 2048 typically ‘hook’ the player with such cycles of challenge and reward, as they are presented with continuous sequences of interesting subproblems to solve, each of which feeds the next, until the solution is achieved. This effect may also be described in terms of Gestalt psychology, as the brain’s natural tendency to mentally complete incomplete patterns [13].

In both of these games, Tetris and 2048, pieces are added to the board in a nondeterministic manner, and the players’ immediate subproblem is where to place those pieces to best effect, given the limited movement options available. These games also tap into our natural betting and risk assessment instincts – what happens if I put that piece here? or there? – which builds a sense of anticipation to see whether the next piece will fit the current plan. This gives players a double incentive to continue playing; the satisfaction of completing the immediate subproblem and the revelation of the next piece of hidden information. Note that the same addictive principle is relevant, even though these video puzzle games do not converge to a ‘solution’ as such.

In Japanese logic puzzles, the subproblems to be solved are the necessary deductions, and the rewards are the simplifications that follow each deduction to reveal new information. If you have any doubt that such puzzles are in fact addictive, then next time you solve one, note the urge to complete just one more item... then one more item... then one more item...

Andrews [14] suggests a more direct causal explanation of why people often find the activity of solving puzzles so emotionally rewarding. He explains that MRI brain scans indicate a relationship between a ‘satisfaction centre’ in the brain called the striatum, which is activated by stimuli associated with reward, and areas of the frontal cortex that are involved with logical thought and planning towards goals. He posits that it is this connection between the ‘intellectual’ cortex and the ‘emotional’ striatum that gives us pleasure in response to solving problems, and drives us on to seek further problems to solve.

This addiction for solving puzzles may not even be confined to the human brain. Recent research at the UK’s Whipsnade Zoo [15] found that chimpanzees given particular dexterity challenges appeared ‘keen to complete the puzzle’ for its own sake, regardless of whether those challenges were associated with a food reward or not.

The following section explores the ramifications of these ideas on the form of puzzles.

3 Form

In this context, the form of a puzzle refers to the degrees of freedom that the setter can manipulate, in order to make challenges more interesting, engaging and aesthetically pleasing for the solver. The function of a puzzle refers to the conceptual framework within which such forms exist, i.e. conditions necessary for the puzzle to work.

In Japanese logic puzzles, the subproblems to be solved are the necessary deductions, and the rewards are the simplifications that follow each deduction to reveal new information. If you have any doubt that such puzzles are in fact addictive, then next time you solve one, note the urge to complete just one more item... then one more item... then one more item...

Andrews [14] suggests a more direct causal explanation of why people often find the activity of solving puzzles so emotionally rewarding. He explains that MRI brain scans indicate a relationship between a ‘satisfaction centre’ in the brain called the striatum, which is activated by stimuli associated with reward, and areas of the frontal cortex that are involved with logical thought and planning towards goals. He posits that it is this connection between the ‘intellectual’ cortex and the ‘emotional’ striatum that gives us pleasure in response to solving problems, and drives us on to seek further problems to solve.

This addiction for solving puzzles may not even be confined to the human brain. Recent research at the UK’s Whipsnade Zoo [15] found that chimpanzees given particular dexterity challenges appeared ‘keen to complete the puzzle’ for its own sake, regardless of whether those challenges were associated with a food reward or not.

The following section explores the ramifications of these ideas on the form of puzzles.

In this context, the form of a puzzle refers to the degrees of freedom that the setter can manipulate, in order to make challenges more interesting, engaging and aesthetically pleasing for the solver. The function of a puzzle refers to the conceptual framework within which such forms exist, i.e. conditions necessary for the puzzle to work.

3 Form

In this context, the form of a puzzle refers to the degrees of freedom that the setter can manipulate, in order to make challenges more interesting, engaging and aesthetically pleasing for the solver. The function of a puzzle refers to the conceptual framework within which such forms exist, i.e. conditions necessary for the puzzle to work.
Authorial control allows the setter to impart structure on their designs, in order to impart some of their personality on the challenges they produce. This section examines some relevant aspects of form that puzzle setters can manipulate.

3.1 Symmetry

An obvious way to inject structure into a design is through symmetry. However, it is important to realise the difference between visual (superficial) and strategic (underlying) symmetry.

3.1.1 Visual Symmetry

Visual symmetry is achieved through the symmetrical placement of hints defining each challenge. For example, Figure 6 shows two Slitherlink challenges (#6 and #21 from 6) with rotationally symmetric hint placement. Many publishers, including Nikoli, have a policy of only publishing symmetrical challenges for most of their puzzles. To see the reason for this requirement, consider the pair of challenges shown in Figure 7, from a recent study in automated puzzle design [15] involving a new puzzle game called Hour Maze[^1]. Both challenges were generated by computer, and both describe valid challenges of similar difficulty on the same background maze, but notice how the symmetrical hint set on the right imposes a sense of order that hints at non-random generation.

Visual symmetry offers the superficial appearance of structure; symmetrical challenges look neater and more elegant but are not necessarily more interesting to solve. However, there is no reason to preclude symmetry as a design constraint, if it pleases the setter or solver, and helps elevate puzzle design to an art form.

3.1.2 Strategic Symmetry

Strategic symmetry refers to pattern or repetition inherent in the solution process itself. This is typically more important than visual symmetry, as it reflects the solution process directly, and can lead to bad designs unless used judiciously.

For example, the Slitherlink challenge shown in Figure 8 is highly symmetric both in its visual design (a) and in its solution (b), which is essentially a repetition of the pattern (c) four times. This challenge is highly redundant and boring; the solver typically wants to be presented with novel subproblems to solve within each challenge.

[^1]: The aim in Hour Maze is to fill the grid with coloured number sets 1–12, such that adjacent numbers differ by ±1.
Similarly, consider the Kakuro challenge shown in Figure 9. The aim of Kakuro is to fill the grid with digits \{1, 2, 3, ..., 9\}, such that each consecutive run totals the number shown and does not contain duplicates. This challenge exhibits a high degree of strategic symmetry, with several immediate simplifications being reflected on opposite sides of the grid in the same combinations, creating a high degree of redundancy.

This attempt by the setter to inject some structure into the design may well backfire, unless the solver is happy repeating the same operations in different parts of the board. Indeed, participants in the Hour Maze experiment exhibited a slight negative correlation between wall symmetry and puzzle enjoyment [15], perhaps due to the fact that symmetrical mazes tend to produce strategically symmetric (i.e. redundant) solutions.

Slitherlink challenge #19 from [18] shown in Figure 10, on the other hand, demonstrates a positive example of strategic symmetry. The natural solution path for this challenge, once obvious simplifications have been performed, is as follows:

1. a cascade down the left hand side (a),
2. a cascade up the right hand side (b), and
3. a cascade up the centre connecting them (c).

The long cascades along each side are rotationally similar to each other in a global sense. However, each involves the solution of different local subproblems during their propagation, imparting structure on the solution without redundancy. As a general rule of thumb, **global symmetry should be accompanied by local asymmetry**, and vice versa.

The duelling cascades in this challenge are unlikely to have occurred by chance, and the solver has a real feeling of an intelligent hand at work in the design, who is perhaps having a bit of fun with them. This challenge is a nice example of authorial control in action.
3.2 Pattern

Handcrafted puzzle challenges often include repeated patterns or motifs as an expression of the setter’s personality. In spatial puzzles such as Slitherlink, such motifs might involve: matching cascades as shown in Figure 10 letter, number or animal shapes in the solution path; or any other interesting nonrandom patterns. In fact, Nonograms, another type of Japanese logic puzzle, actually produce works of art (or at least pictures) as grid cells are coloured according to certain rules. Motifs may also occur in number puzzles, such as the ‘snail’ pattern in the Killer Sudoku example shown in Figure 11. The top right 3×3 subgrid contains two regions, totalling 38 and 14 respectively, with one cell on the 38-region exceeding the subgrid boundary (left). This rogue cell must resolve to 7, which is the difference between the sum of the two regions (38+14 = 52) and the disjoint sum of all digits (1+2+3+4+5+6+7+8+9 = 45) that the 3×3 subgrid must total (right). The central cell can then also be resolved to a 7, as digits cannot be repeated in any row or region.

This snail pattern provides an elegant self-resolving starting point for this challenge, which contains a rotated variation each corner, giving the strong impression of carefully structured design. But again, there can be a fine line between amusing the solver through pattern and annoying them through redundancy.

3.3 Ladders

Ladders, i.e. forced sequences of moves typically in a repeated pattern, are another indicator of authorial control. They are different to the patterns described above, as they are implicit in the design and manifest themselves during its solution.

For example, consider the section of a hypothetical Slitherlink challenge shown in Figure 12 (left). The two 0 hints at the top left dictate which two edges of the adjacent 2 hint are ‘on’, which in turn dictate which two edges of the adjacent 2 hint are ‘on’, and so on. This is a contrived example, but such self-perpetuating ladders are often found in actual challenges.

Figure 12 shows a section of a Japanese logic puzzle called Masyu, in which the solver must draw a single non-self-intersecting path through every circle on the board, such that the path turns within each black circle but passes straight through each white circle. See the ‘Masyu’ article in this issue (which includes this example) for more details. In this case, the line of black dots triggers a self-perpetuating ladder from the top left corner inwards (right).

http://www.nonograms.org
Figure 14 shows another example of a ladder, in another Masyu challenge from [2]. In this case, the alternating sequence of black and white circles (top row) forces the self-perpetuating ladder of edges shown (bottom row), as the path must extend straight for two cells from each black circle and also pass straight through each white circle.

Such ladders can also have strategic value, as they can often be clumped into a single unit of information when they are recognised, reducing the solver’s mental workload through modularity [21] or chunking. For example, if a Slitherlink solver notices a diagonal line of 2s as in Figure 12, then the effect of a deduction at one end of the line can often be seen immediately at the other end of the line, without having to think about the intervening items.

### 3.4 Surprise

An element of surprise can keep a challenge interesting and impart the impression of authorial control, typically by establishing a pattern in the solution process and then suddenly disrupting it.

Figure 15 shows a Slitherlink challenge whose obvious progress point is circled on the shaded region in the top right corner (left). This region can only expand downwards, connecting to its neighbouring group (right), but the obvious progress point now jumps to the left side of this extended group (circled). The solver, after following an orderly cascade down the right hand side, must suddenly switch to the other side of the grid.

While the shading in the figure makes this discontinuity easy to spot, larger jumps in more complex situations can be confusing for the solver. Deductions that trigger key information in distant parts of the grid can suggest an intelligent setter actively trying to keep the solver on their toes.

Famous Chess puzzle setter Sam Loyd recognised the importance of surprise, stating that his goal was to compose puzzles whose solutions require a first move that is contrary to what 999 players out of 1,000 would propose [22].

![Figure 15. A discontinuous jump propagates this Slitherlink solution.](image-url)
3.5 Perversity

When it comes to expressing personality, it is hard to beat sheer perversity. Consider the Slitherlink challenge shown in Figure 16 (a). This challenge has an obvious solution (b), but this can be difficult for experienced Slitherlink solvers to spot. The problem is that two adjacent 3-hints form a common pattern that invariably implies three parallel edges in a normal context (c).

![Figure 16. A Slitherlink joke.](image)

Experienced solvers will fixate on this learnt pattern and simply not see the obvious solution; they must unlearn habits ingrained over many hours of reinforcement. Such blatant disregard for tradition allows the innovative setter to subvert the solver’s expectations for a bit of mischief.

Sudoku challenge #99 from [6], shown in Figure 17, is another case in point. Three values can be immediately resolved from the initial hint set with little effort (highlighted), leading the solver to think that this challenge is not so difficult. However, the information soon dries up and its true difficulty becomes apparent; this is actually the most difficult challenge in its collection. Such deception is common in deduction puzzles. If a challenge starts off as being particularly easy, then the solver may be lured into a false sense of security, but can expect tough times ahead.

![Figure 17. An easy Sudoku challenge… or is it?](image)

![Figure 18. Patterns that yield little information.](image)

The Killer Sudoku challenge shown in Figure 18 [23] has two points of interest. Firstly, the top right $3 \times 3$ subgrid contains three regions that fit exactly within the subgrid, whereas typically at least one region would overlap its boundary to provide some information for the solver; this is almost a standard solution pattern, but not quite.

Secondly, the shaded 23, 11, 14 and 7-regions along the bottom row sum to $23 + 11 + 14 + 7 = 55$, hence the two circled cells must sum to 10 (since the nine cells along the bottom row must add to the disjoint sum of all digits $1+2+3+4+5+6+7+8+9 = 45$). However, all of the values available for these two cells, $\{6, 8, 9\}$ and $\{1, 2, 4\}$ respectively, all have pairings that yield 10, hence none can be eliminated and this potential line of enquiry fails. Hence, a normally fruitful solution pattern yields surprisingly little information.

The fact that this challenge immediately throttles an experienced solver’s expectations in two such blatant ways suggests that this challenge was set by a designer wanting to tease the solver. The challenge shown in Figure 19 [24] also demonstrates some unusual design features:

- The hint subregion layout is horizontally, vertically and rotationally symmetrical.
- Single cell subregions exist, which can be immediately instantiated to their shown value.
- Each $3 \times 3$ subgrid is dominated by a plus-shaped subregion separating its corners.
- Every subregion is either: 1) a corner, 2) an edge, 3) a plus-shape, or 4) a $2 \times 2$ junction.

---

6Provided by Jimmy Goto from Nikoli.

7Unless, of course, the challenge is actually rated as ‘easy’.
Killer Sudoku challenges typically involve asymmetrical hint subregion layouts – such as Figure 18 – to allow maximum diversity and strategic depth. To find any of these types of symmetry in a challenge is a surprise, but to find all four speaks of a very eccentric design that pushes this puzzle's design constraints to the limit.

The single cell subregions that immediately instantiate to their shown value, in particular, are unusual and violate the conventions of this puzzle. But this challenge is also engaging and interesting to solve, which indicates a meticulous design carefully constructed by its setter.

4 Conclusion

While this paper only touches on a small number of puzzle types, mostly Japanese logic puzzles, I believe that the principles described have relevance to puzzle design in general. Most important is the way in which the setter can exert authorial control on their designs, in order to impart some structure and personality – manipulating the solver's actions in absentia – to make challenges interesting, engaging and aesthetically pleasing.

Phrasing puzzles as two-player games played between the setter and the solver helps explain these ideas, but this is not to say that puzzles are zero-sum games. The setter does not win by defeating the solver as often as possible, but by providing the most interesting, engaging and aesthetically pleasing challenges as often as possible. Both win if the setter provides interesting challenges and the solver enjoys completing them.

I hope that this discussion is useful for puzzle setters, and interesting to solvers, and might suggest concrete approaches for improving the quality of computer-generated designs.

Acknowledgements

Thanks to Raf Peeters and Stephen Tavener for discussions on puzzle design. Thanks also to Jimmy Goto for insights into Nikoli’s design process, and permission to use Nikoli challenges in this article.

References


---

\[8\] A zero-sum game is one in which a win for one player implies a loss for the other participant(s).

Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design. 
Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia. 
Email: c.browne@qut.edu.au

---

Heptalion Challenge #2

Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.
Uniqueness in Logic Puzzles

Cameron Browne, Queensland University of Technology (QUT)

Pure deduction puzzles typically have a single unique solution. However, some puzzle setters argue that challenges with multiple solutions are also valid, if they can be solved by eliminating choices that lead to ambiguous states. This paper considers the arguments for and against this position, and presents a counterexample that demonstrates the danger of using uniqueness to decide between multiple solutions.

1 Introduction

A characteristic of pure deduction puzzles, such as Japanese logic puzzles, is that each challenge has a single unique solution. This allows such challenges to be solved by deduction rather than guesswork [1].

I was therefore surprised to find a Kakuro challenge with multiple solutions in a publication as respectable as The Guardian [2]. This was the first time that I had ever encountered such a case in print. The aim in Kakuro is to fill each cell with a digit in the range 1–9, such that each horizontal and vertical run adds to the hint total shown, and no digit is repeated within each run [3].

Figure 22 shows the relevant section of the Kakuro challenge in question (all other values have been resolved). Possible values for the final few unresolved cells are shown in small print, and a key cell with possible values 4 or 5 is circled. This challenge has three possible solutions, depending on whether this key cell takes the value 4 or 5, as shown.

After alerting the UK setter of this challenge to what appeared to be a flawed design with no deducible solution, I was also surprised by his response. He maintained that this challenge was indeed valid, and could be solved by deduction based on relative uniqueness.

Figure 1. A Kakuro challenge with three solutions. The circled cell can take the value 4 or 5.
2 The Case For Ambiguity

The setter of the ambiguous Kakuro challenge argued as follows:

Any move \( M \) that leads to multiple solutions can be eliminated. \( \text{(1)} \)

For instance, the value of the circled cell in Figure 2 cannot be 4, as such a move would allow multiple solutions (top row). This cell must therefore take the value 5, producing the single ‘correct’ solution (bottom row).

This argument of deduction by relative uniqueness, for selecting among multiple solutions, seems fair enough at first glance. It adds some much-needed depth to Kakuro, by allowing an additional solution strategy. It also increases the number of possible challenges that can be devised, by allowing cases with multiple solutions that traditional setters would not allow.

However, Japanese publisher Nikoli, the inventor and major supplier of Kakuro, categorically state that uniqueness should not be exploited in this way to solve Kakuro, or any of their other pure deduction puzzles.\(^1\) We now consider the argument for absolute rather than relative uniqueness.

3 The Case For Uniqueness

A serious problem with deduction by relative uniqueness is that it does not work unless the solver also knows that this rule is in force, but uniqueness is generally assumed for such puzzles rather than explicitly stated. For example, the Kakuro rules provided by The Guardian make no mention of uniqueness, making those rules insufficient to solve the ambiguous challenge shown in Figure 2.

Further, there is an obvious corollary to the argument (1) made above:

Any move leading to ambiguous move \( M \) can therefore also be eliminated. \( \text{(2)} \)

Hence, chaining backwards from ambiguous move \( M \), every prior move can also be said to lead to ambiguity and hence be eliminated, until the challenge has no valid moves. Or can it? There is no clear answer to this question, which depends on the setter’s and solver’s interpretations.

3.1 Counterexample

The following counterexample demonstrates the dangers of deduction by relative uniqueness. Slitherlink is a deduction puzzle in which a simple closed path must be traced through orthogonal vertices of a square grid, to visit the number of sides indicated on each numbered cell.\(^4\) For example, Figure 3 shows a simple 2×3 Slitherlink challenge with three valid solutions: \( a, b \) and \( c \).

Given that four edges can be deduced as shown in Figure 3 (top), consider the move indicated by the dotted line. If there is not an edge between these vertices then two possible solutions exist (left), hence this move must be an edge and \( c \) must be the ‘correct’ solution (right).

However, if the same process is applied to the move indicated in Figure 4 (top, dotted), then \( b \) is deduced to be the ‘correct’ solution (right).

\(^1\)Strongly worded personal correspondence.
Figure 4. Deduction by uniqueness yields b.

Deduction by relative uniqueness therefore gives two conflicting ‘correct’ solutions, b and c, depending on processing order. To derive the same solution as the setter, the solver would have to follow the same sequence of decisions in the exact same order, but there is no way to enforce this in practice. Deduction by relative uniqueness is not guaranteed to yield the same solution from among multiple solutions in all cases.

This Slitherlink counterexample could be said to have one valid solution (depending on the order in which the solver made their deductions), two equally valid solutions (through deduction by relative uniqueness) or three equally valid solutions (which it does, after all – see Figure 2). This is clearly an unsatisfactory state of affairs. But if absolute uniqueness is enforced, and such cases of multiple solutions avoided, then all of these problems simply go away, at no real cost. As expert puzzle designer Hiroshi Higashida points out:

Puzzle creators, not only solvers, mustn’t defy rules, either [5] p216.

4 Conclusion

The characteristic of pure deduction puzzles to have a single unique solution is not only elegant, but performs a vital practical function. It guarantees that challenges can be solved by deduction alone, without guesswork or ambiguity, and means that the setter and solver are both playing from the same rule set without the need to make assumptions about implied or hidden rules. Further, uniqueness makes challenges self-checking; if the player has deduced a solution, then it must be the correct one. As tempting as it may be to relax this constraint of absolute uniqueness and instead exploit relative uniqueness as a solution strategy, this is best avoided in pure deduction puzzles.

Acknowledgements

Thanks to Jimmy Goto for clarifying Nikoli’s position on uniqueness.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au
Influence Maps and New Versions of Risk

Daniel Ashlock, University of Guelph

Colin Lee, University of Guelph

An influence map is a way of determining how much one location in a game affects or influences the others. We show how influence maps can be used to produce and grade variations of the popular board game Risk. In our variation, two monoliths on the game map permit troops to pass between them. The influence map is used to determine how much a given placement of the monoliths changes the game.

1 Introduction

Risk™, a publication of Parker Brothers and now a division of Hasbro, is a simple war game played on a stylised map of the world [1]. A basic version of the Risk board is shown in Figure 1 and the rules are summarised below. The adjacencies of countries are the wellspring of strategy for Risk.

1.1 Variations on Risk

A number of official variations (including Risk 2210 and versions based on George Lucas’s Clone Wars and Tolkien’s Lord of the Rings) have been published. Many different rule variations and house rules exist. This column will look at a simple modification of the original game which can be used to disrupt some favourite strategies, such as holing up in Australia, and which changes the game in a quantifiable way.

We propose a science fiction variant, in which two alien monoliths are discovered on different continents, which allow troops to teleport between them. This creates an additional link on the Risk board which acts as a portal.

Figure 1. The Layout of the Risk Board, courtesy of Grøgmint in the Wikimedia commons.

Risk is played with tokens representing armies, five six-sided dice, and a deck of cards emblazoned with one of the countries and a soldier, cannon, or cavalry soldier. Two jokers have no country but all three emblems.

Players put an initial collection of armies on countries selected by dealing out the cards.

The players take turns, gaining new armies in proportion to the number of territories held with bonuses for owning whole continents. In a turn, a player attacks from a country with at least two friendly armies to an adjacent country held by another player, as many times as desired.

Dice rolled by both players determine armies lost on each side. If the defender runs out of armies, the attacker captures the country. In any turn that a country is taken, the attacker draws a card.

At the beginning of the turn, triples with the same emblem, or all different, may be turned in for bonus armies. The bonus grows as triples are cashed in.

The goal of the game is to take control of the entire world.

The location of the monoliths can be agreed upon by the players before country selection starts, or chosen by drawing random country cards (the second card should be redrawn until it is on a different continent to the first). Adding a link is not a radical change, but the set of possible links can be seen as a set of variants, each with strategies substantially different to the original game.

Next, we introduce the idea of influence maps

An influence map is a way of quantifying how

much influence each location on a game board has on the others. Computing influence maps requires knowledge of the adjacency relationship of the regions of the game. For some games, such as Chess, this is quite tricky because the adjacency of squares depends on the identity of the piece occupying the square. In Risk, the adjacency relationship, while central to game strategy, is simple and consists of sharing a border or being explicitly connected by a line printed on the board.

We abstract the Risk board as a combinatorial graph, compute a type of influence map for each country on each other country, and then measure the extent to which monolith links between two countries change the overall influence map.

2 The Math

This subsection gently introduces the math we use to estimate the impact of the placement of different pairs of monoliths on the Risk board.

2.1 A Little Graph Theory

Formally, a graph is a set V of vertices and a set E of edges, in which each edge is an unordered distinct pair of vertices. Informally, a graph can be described as a network, in which vertices are nodes and edges are arcs connecting two nodes.

There are many ways to represent the same graph. For example, a graph can be represented as a drawing, as in Figure 2, as two sets (V and E), or in a variety of matrix formats.

Now, our original question:

For which pair of countries would adding the monoliths change the game the most?

can be reformulated as:

If we can insert a single edge into the underlying graph of the Risk board, which edge would change the connectivity of the resulting graph the most?

2.2 Diffusing Gas Analogy

We now give an analogue explanation of the influence map we will be using. Starting with a graph G, imagine that we pump 1 unit of gas into G at a specific vertex v per time step. The gas diffuses across edges, but each vertex absorbs some constant fraction α of the gas that it holds per time step. The gas currently at each vertex is divided evenly among the vertex and its neighbours, after which the decay mediated by α is applied.
The interplay between the arithmetic increase in gas coming into the system, and the exponential decay caused by \( \alpha \), mean that the level of each gas at each vertex will reach equilibrium. When equilibrium is reached, the amount of gas introduced at some vertex \( v_a \) and any other vertex \( v_b \) will give a measure of the connectivity between \( v_a \) and \( v_b \).

Calculating the amount of gas present at each vertex, for all possible starting vertices, gives a measure of the connectivity of the graph with respect to all vertices. We define the diffusion character matrix of the graph as the \( 42 \times 42 \) matrix in which each column and row corresponds to one of the 42 countries on the Risk board, such that each \( (i; j) \)th entry shows the amount of gas present at vertex \( i \) after the system comes to equilibrium, supposing that the gas was pumped in at vertex \( j \).

Since the amount of gas in the entire graph does not depend on which vertex we pick as our starting vertex to pump gas into, it follows that every column in the diffusion character matrix has a constant total sum. So we can normalise this matrix by dividing by the constant column sum to get a matrix with columns that all sum to 1 and call this a normalised diffusion character matrix.

The columns of the normalised diffusion character matrix can be thought of as discrete probability distributions, where each \( (i; j) \)th entry corresponds to the probability of ending a random walk along graph edges at vertex \( i \) given that you started at vertex \( j \).

The diffusion character matrix is the influence map, telling us how much influence each vertex has on each other. Changes in the connectivity of the map will cause changes in the influence map. One way to measure the difference in the connectivity between two graphs with the same number of vertices, would therefore be to compare their normalised diffusion character matrices.

2.3 Monolith Placements

First we go through and generate all the graphs that can be obtained by each possible placement of those monoliths not in already adjacent territories. Realised in the graph representation, this is all ways of adding one edge to the graph derived from the Risk game board. For each of these new graphs, we compare its normalised diffusion character matrix with the normalised diffusion character matrix of the original graph. We then determine which new graphs have the largest difference from the original graph.

Note that we must be careful about the concrete representations of the graphs. If we choose two different orderings of the countries of the Risk board, we could not simply compare the \((i; j)\)th entries of the two resulting normalised diffusion character matrices, but would need to compare the entries corresponding to the same countries. Similarly, if we add an edge to the graph of the Risk board, we cannot simply compare corresponding entries, since the resulting graph may no longer have vertices ordered in the same way as the original graph.

To get around this ordering problem, we calculate a measure of the skewness of the probability distribution associated with each column of a normalised diffusion character matrix, sort this list, and compare differences in the sorted lists for two graphs. Skewness is a measure of deviation from symmetry, or in this case the degree to which the gas is or is not evenly distributed. We could compare vertices corresponding to countries with the same name, but when edges are added, the strategic meaning changes. Placing skewness measures for the vertices in descending order and then using the distance between these vectors provides a more objective assessment of how much or how little the board has changed.

It is also worth noting that we do not in fact have a strong reason to prefer either of our two skewness measures and we have no reason to suppose that long or short range influence is more important. We therefore present six sets of results, for both skewness measures and three values of \( \alpha \), which controls the range of influence.

3 Biggest and Smallest Changes

We still need values for \( \alpha \) in order to perform our calculations. Since this parameter controls the degree to which we care about longer or shorter distances, we will calculate diffusion character matrices for three values of \( \alpha \): 1/3, 1/2, and 2/3.

We have chosen two measures of the skewness of a probability distribution: its entropy and its column 2-norm. We calculate these measures for each column of the normalised diffusion character matrix to get a list of skewness scores, as follows.

1. Entropy is given by:

\[
\sum_{i=1}^{n} p_i \cdot \log_2(p_i)
\]

where the \( p_i \) are the \( n \) probabilities making up the probability distribution represented by the entries in a column of a normalised diffusion character matrix.

2. The column 2-norm is given by:

\[
\sqrt{\sum_{i=1}^{n} p_i^2}
\]
although we will actually ignore the square root and simply take the sum.

The following charts use the Euclidean distance squared between the sorted lists of skewness measures of the original Risk board game and a single edge difference Risk board graph. Note that a high score for both entropy list difference and column 2-norm difference correspond to a large difference between the skewnesses of normalised diffusion character matrices and hence correspond to large differences in the overall connectivity of the underlying graphs which are surrogates for the degree to which game strategy shifts.

### 3.1 For $\alpha = 1/3$

The largest and smallest entropy list and column 2-norm differences, for $\alpha = 1/3$, are shown.

#### Largest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.063200</td>
<td>South Africa, New Guinea</td>
</tr>
<tr>
<td>0.063526</td>
<td>Peru, Eastern Australia</td>
</tr>
<tr>
<td>0.064973</td>
<td>Argentina, Western Australia</td>
</tr>
<tr>
<td>0.064973</td>
<td>Argentina, New Guinea</td>
</tr>
<tr>
<td>0.069028</td>
<td>Argentina, Eastern Australia</td>
</tr>
</tbody>
</table>

#### Smallest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0017944</td>
<td>Quebec, Western United States</td>
</tr>
<tr>
<td>0.0017297</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.0014588</td>
<td>Western Europe, Scandinavia</td>
</tr>
<tr>
<td>0.0014205</td>
<td>Siberia, Kamchatka</td>
</tr>
<tr>
<td>0.0005681</td>
<td>Yakutsk, Mongolia</td>
</tr>
</tbody>
</table>

#### Largest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00044022</td>
<td>South Africa, Eastern Australia</td>
</tr>
<tr>
<td>0.00044203</td>
<td>Peru, South Africa</td>
</tr>
<tr>
<td>0.00045461</td>
<td>Argentina, South Africa</td>
</tr>
<tr>
<td>0.00045937</td>
<td>Argentina, Eastern Australia</td>
</tr>
<tr>
<td>0.00045971</td>
<td>Peru, Eastern Australia</td>
</tr>
</tbody>
</table>

#### Smallest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000001146</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.000001106</td>
<td>Egypt, Western Europe</td>
</tr>
<tr>
<td>0.000001068</td>
<td>Western Europe, Ukraine</td>
</tr>
<tr>
<td>0.0000010188</td>
<td>Yakutsk, Mongolia</td>
</tr>
<tr>
<td>0.00000088760</td>
<td>North Africa, Middle East</td>
</tr>
</tbody>
</table>

### 3.2 For $\alpha = 1/2$

The largest and smallest entropy list and column 2-norm differences, for $\alpha = 1/2$, are shown.

#### Largest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.21447</td>
<td>Peru, New Guinea</td>
</tr>
<tr>
<td>0.21447</td>
<td>Peru, Western Australia</td>
</tr>
<tr>
<td>0.22802</td>
<td>Brazil, Eastern Australia</td>
</tr>
<tr>
<td>0.24523</td>
<td>Argentina, Eastern Australia</td>
</tr>
<tr>
<td>0.24991</td>
<td>Peru, Eastern Australia</td>
</tr>
</tbody>
</table>

#### Smallest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00029469</td>
<td>Alberta, Greenland</td>
</tr>
<tr>
<td>0.0015425</td>
<td>Irkutsk, Japan</td>
</tr>
<tr>
<td>0.0015059</td>
<td>Quebec, Western United States</td>
</tr>
<tr>
<td>0.00098565</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.00048602</td>
<td>Yakutsk, Mongolia</td>
</tr>
</tbody>
</table>

#### Largest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0016966</td>
<td>Brazil, Eastern Australia</td>
</tr>
<tr>
<td>0.0017238</td>
<td>Argentina, New Guinea</td>
</tr>
<tr>
<td>0.0017238</td>
<td>Argentina, Western Australia</td>
</tr>
<tr>
<td>0.0019256</td>
<td>Peru, Eastern Australia</td>
</tr>
<tr>
<td>0.0019491</td>
<td>Argentina, Eastern Australia</td>
</tr>
</tbody>
</table>

#### Smallest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00023200</td>
<td>Siberia, Kamchatka</td>
</tr>
<tr>
<td>0.00021703</td>
<td>Western Europe, Ukraine</td>
</tr>
<tr>
<td>0.00019740</td>
<td>Yakutsk, Mongolia</td>
</tr>
<tr>
<td>0.00018912</td>
<td>Western Europe, Scandinavia</td>
</tr>
<tr>
<td>0.00014919</td>
<td>North Africa, Middle East</td>
</tr>
</tbody>
</table>

### 3.3 For $\alpha = 2/3$

The largest and smallest entropy list and column 2-norm differences, for $\alpha = 2/3$, are shown.

#### Largest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.67089</td>
<td>North Africa, Eastern Australia</td>
</tr>
<tr>
<td>0.67611</td>
<td>Argentina, Eastern Australia</td>
</tr>
<tr>
<td>0.71088</td>
<td>Venezuela, Eastern Australia</td>
</tr>
<tr>
<td>0.71319</td>
<td>Peru, Eastern Australia</td>
</tr>
<tr>
<td>0.74140</td>
<td>Brazil, Eastern Australia</td>
</tr>
</tbody>
</table>

#### Smallest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000001146</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.000001106</td>
<td>Egypt, Western Europe</td>
</tr>
<tr>
<td>0.000001068</td>
<td>Western Europe, Ukraine</td>
</tr>
<tr>
<td>0.0000010188</td>
<td>Yakutsk, Mongolia</td>
</tr>
<tr>
<td>0.00000088760</td>
<td>North Africa, Middle East</td>
</tr>
</tbody>
</table>
Smallest Entropy List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0013345</td>
<td>Irkutsk, Japan</td>
</tr>
<tr>
<td>0.0011615</td>
<td>Yakutsk, Japan</td>
</tr>
<tr>
<td>0.0010419</td>
<td>Alberta, Greenland</td>
</tr>
<tr>
<td>0.0084558</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.0074116</td>
<td>Quebec, Western United States</td>
</tr>
</tbody>
</table>

Largest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0051404</td>
<td>Peru, New Guinea</td>
</tr>
<tr>
<td>0.0051404</td>
<td>Peru, Western Australia</td>
</tr>
<tr>
<td>0.0053178</td>
<td>Brazil, Eastern Australia</td>
</tr>
<tr>
<td>0.0055653</td>
<td>Argentina, Eastern Australia</td>
</tr>
<tr>
<td>0.0057366</td>
<td>Peru, Eastern Australia</td>
</tr>
</tbody>
</table>

Smallest Column 2-norm List Difference:

<table>
<thead>
<tr>
<th>Score</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000017741</td>
<td>Ural, India</td>
</tr>
<tr>
<td>0.000017652</td>
<td>Yakutsk, Mongolia</td>
</tr>
<tr>
<td>0.000017139</td>
<td>Siberia, Kamchatka</td>
</tr>
<tr>
<td>0.000016462</td>
<td>Western Europe, Scandinavia</td>
</tr>
<tr>
<td>0.000012228</td>
<td>North Africa, Middle East</td>
</tr>
</tbody>
</table>

4 Discussion

It is worth noting that although many of the same edges (pairs of countries) appear in the same top five blocks for the different values of $\alpha$, the order in which they appear varies. The agreement in areas of continents is greater than that of individual pairs of countries. This is somewhat troublesome, but is still not unexpected, as the values of $\alpha$ essentially measure how much we care about long range connectivity (the smaller the value of $\alpha$, the shorter the range we care about the most).

We can nonetheless get a good idea of which edges will lead to the greatest difference in the connectivity of the Risk board. These edges are {Peru, Eastern Australia} and {Argentina, Eastern Australia}. For example, Figure 4 shows the Risk map with a portal link between Peru and Eastern Australia.

4.1 Diffusion Character Influence Maps

Computing diffusion characters is equivalent to inverting a square matrix with as many rows as there are vertices in the graph. This would be tedious to do by hand, but computer software does it quickly. Once it is computed, the diffusion character can be saved and reused for as many purposes as needed.

Figure 4. A portal between Peru and Eastern Australia creates the most different variant.

The influence of a block of vertices can be computed with a diffusion character; simply add up the gas from each vertex in the block on any particular vertex $v$, and you will know the aggregate influence of those vertices on $v$. It is also possible to modify the computation of the diffusion character itself to achieve other results. Suppose that instead of adding one unit of gas to a vertex, you add as many units of gas as there are armies on the vertex’s corresponding country. The diffusion character would then measure the armed might that could be brought to bear, and the ease with which other countries could be threatened could be explored by varying $\alpha$.

5 Conclusion

It seems reasonable and mathematically justifiable to conclude that placing a portal either between Peru and Eastern Australia, or between Argentina and Eastern Australia, will result in the largest change in connectivity and hence largest change in game play. We would be happy to hear from readers who try our new versions of Risk.

Those who have played Risk many times before know that some players like to hide in Australia and control Oceania, others try to use South America as a springboard to controlling Africa, and that some overconfident beginners try to grab North America right away. Vizzini’s advice from the film The Princess Bride, ‘never get involved in a land war in Asia’, also applies to all but the ending stages of Risk. Monolith Risk can change much of this strategic equation, but would probably not soften Vizzini’s advice. This suggests that, in addition to adding links, one might want to postulate inverse monoliths that project force fields which cut links between countries. These could also be planned and categorised with diffusion characters.
Acknowledgements

The Authors thank the Natural Sciences and Engineering Council of Canada (NSERC) and the University of Guelph for their support of this work.

References


---

Heptalion Challenge #3

Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.
Symmetrical Pentomino Pairs

Kate Jones, Kadon Enterprises, Inc.

Pentominoes – shapes made of five congruent squares – provide a natural platform for games and puzzles. In this article, I describe my own experiences designing pentomino packing games, and some interesting puzzles that emerged when studying symmetrical pairs of pentominoes.

1 Introduction

The twelve pentominoes (Figure 1) were originally defined, named and studied by Solomon W. Golomb in his seminal book, Polyominoes [1]. They have been copiously studied, played with, explored and documented as a popular form of recreational mathematics, by thousands of enthusiasts.

One of the many interesting aspects of pentominoes is that certain pairs of them can fit together to form symmetrical shapes. We have done a thorough exploration of all their possible combinations and report the results herewith.

2 Background

In 1976, I read a science fiction novel by Arthur C. Clarke, Imperial Earth [2], which featured pentominoes as an important part of the plot. Clarke had been bitten by the pentomino bug from an article by Martin Gardner in the November 1960 issue of Scientific American [3].

Clarke’s addiction quickly passed on to me, and in 1979 we formed Kadon Enterprises, Inc. [4], to produce and sell a pentominoes set named Quintillions® (Figure 2). This consisted of twelve laser cut wood blocks, each equivalent to five congruent cubes in all their possible different planar adjacencies, hence planar pentacubes.

The laser cutting technique was chosen to assure the best precision of size and accurate fit even in three-dimensional assemblies and with minimum wasted material. For strength and aesthetics, we needed to have the wood grain follow the longest dimension of each piece. Since none of the known rectangles – 4 × 15, 3 × 20, 5 × 12 and 6 × 10 – yielded solutions with all pieces placed lengthwise, we worked out the next best thing, a 4×16 solution, by hand, with four cubes left over.
Exploring various solutions at one point suggested that pairs of pieces often formed a symmetrical shape. The question of how many combinations were possible became the subject of this research project in 1981.

3 Exhaustive Search

A systematic approach required testing each pentomino connected to every other pentomino, thus \( \binom{12}{2} = 66 \) combinations, and for each such pair we tested all ways of joining at least two of their edges. It was interesting to observe that some pairs could combine into more than one symmetrical shape, and that some symmetrical shapes could be formed by more than one pair.

Here, then, are all 52 symmetrical pentomino pairs. The first group (Figure 3) shows the orthogonal twins, in which more than one pair can make the same shape.

The second group (Figure 4) contains the diagonally symmetrical twins. These were more difficult to identify, especially those that have holes. The most elusive was the YP pair, which is a copy of the symmetrical XW pair.

Towards the end of the search, we kept testing pairs made of symmetrical pieces to see whether they had twins among the asymmetrical pieces. This turned up the really surprising pairs. Figure 5 shows the group of orthogonal symmetrical pairs with no twins.

The last group, shown in Figure 6 has the most ornate diagonal symmetries. Six of the pairs \{VW, VX, LZ, NV, FP, NP\} can form more than one shape.

The 52 different symmetrical pentomino pairs are composed of 34 essential pairings. Several pairs can make two shapes, and NV and YP can make three. If we disregard the requirement that at least two unit edges touch, there are ten additional pairs. Working them out is left as an exercise (hint: they use the orthogonally symmetrical I, T, X and U).
Once we identified all the possible symmetrical pairs, we began to wonder if we could make six pairs simultaneously, and how many ways that could be done.

Not having the luxury of a computer, we started to sort them manually, beginning with the I pentomino, which has the fewest mates, based on this ‘congeniality’ index of how many other pentominoes any particular pentomino can have as a partner:

- \( I \rightarrow \{ L, T, U \} \)
- \( U \rightarrow \{ I, L, T, X \} \)
- \( X \rightarrow \{ F, U, V, W \} \)
- \( Z \rightarrow \{ F, L, N, T \} \)
- \( N \rightarrow \{ N, P, T, W, X \} \)
- \( Y \rightarrow \{ L, N, P, T, U \} \)
- \( L \rightarrow \{ F, I, U, W, Y, Z \} \)
- \( N \rightarrow \{ F, P, V, W, Y, Z \} \)
- \( W \rightarrow \{ F, L, N, P, V, X \} \)
- \( F \rightarrow \{ N, P, T, W, X, Y, Z \} \)
- \( P \rightarrow \{ F, L, N, T, V, W, Y \} \)
- \( T \rightarrow \{ F, I, P, U, V, Y, Z \} \)

After two days of struggling with paper and pencil notes, it occurred to us that there had to be an easier way. Our friend Jim Kolb was a BASIC programmer, and we appealed to his expertise to speed up our search.

We provided him with the 52 pairings, and in short order his program turned up some interesting statistics:

- The total number of six simultaneous symmetrical pairs is a prime number: 2,153.
- Of all these, there is exactly one combination in which all six pairs can form more than one shape.
- There are 8 solutions in which all six pairs form only one shape.
- There are 2 solutions in which all six pairs are orthogonal.
- Even though there are so many diagonal pairs to play with, it is not possible to form six of them simultaneously.

Having conquered the question of six simultaneous symmetrical pairs, we took the next step of making larger symmetrical figures by combining all the pairs, such as the ‘Thunderbird’ (Figure 7) and ‘Christmas Tree’ (Figure 8). Solving the ‘Christmas Tree’ is left as an exercise.

---

1Personal correspondence, 9 September 1981. The nine-page list of 2,153 sets of six simultaneous symmetrical pairs took 10 hours to calculate on an ATARI 800.
5 An Unsolved Problem

When exploring any game or puzzle idea, the human brain is never content to leave well enough alone, always pushing for the next level or a new variation. We are born with such inquisitiveness and a propensity for design and analysis. The ‘symmetrical pairs’ problem is no exception.

Having established the inventory of pairs and even played with forming combined constructions, we wondered:

Is it possible to arrange the pentominoes such that each is part of a symmetrical pair with every piece that it touches?

Pushing the boundaries even further, we can ask whether a chain of twelve symmetrical pairs is possible, such that the two end pieces actually meet and form the last symmetrical pair. This question has not yet been answered. Perhaps an enterprising reader will find such a solution or prove that none exists before our own research finds the answer.

Figures 9 and 10 illustrate such chains of symmetrical pairs. Furthermore, both of them, like dominoes, have end pieces that are also compatible, although with intervening spaces. It would as well be interesting to explore the shortest and longest perimeters of such constructions.

6 A Pairs-Rich Rectangle

Once we had identified all symmetrical pairs of pentominoes, another idea raised its head:

What is the maximum number of symmetrical pairs in a 5×12 rectangle?

A day and a half of solving by hand yielded the following composition (Figure 11), which became the Quintillions® box pattern:

Figure 8. ‘Christmas Tree’ solvable with six symmetrical pairs of pentominoes.

Figure 9. A chain of eleven symmetrical pairs with perimeter 72.

Figure 10. A chain of eleven symmetrical pairs with perimeter 68.

Figure 11. A symmetries-packed rectangle.
Not only does this rectangle contain five symmetrical pairs (Figure 12) but three symmetrical triples (Figure 13), and Kadon’s logo \( \text{H} \) – the W pentomino – occupies the centre. The U is part of three overlapping pairs and two entwined triples.

Figure 12. Five symmetrical pairs embedded in the $5 \times 12$ packing (shaded).

Figure 13. Three embedded symmetrical triples embedded in the $5 \times 12$ packing (shaded).

7 Conclusion

Pentominoes are one of the most enduring and fascinating members of the recreational math genre. They present a virtually endless source of activities and explorations. The project reported here explored one particular aspect of pentominoes: how the twelve distinct pentominoes can form symmetrical pairs.

In designing this research, we exhaustively examined the relationships between any two pentominoes (66 distinct pairs, in all possible joinings) to catalog every possibility. No computer program was used in their derivation, but a further investigation into forming six simultaneous symmetrical pairs did use a search program to identify the 2,153 ways that six pairs can be formed.

Because some pentominoes can pair with only a few others, placing them first leaves the more versatile or ‘congenial’ pieces for last. This gives the solver a better chance of success when looking for solutions that require six symmetrical pairs.
Why symmetry? There is something intuitively appealing and aesthetically pleasing in symmetry; a balance, a rhythm. Tessellations and recurring patterns provide a reassuring visual environment, and creating them is intellectually rewarding. The result is elegance, beauty, and open ended engagement with each discovery.

The ‘symmetrical pentomino pairs’ project was an absolute delight, and it leaves one unsolved problem still on the table: the closed chain of twelve simultaneous symmetrical pairs. We look forward to hearing from readers with a good answer.

References


Kate Jones, President of Kadon Enterprises, Inc., is a graphic artist who has been developing recreational mathematics puzzles and games since 1979. Her interests include human consciousness, problem solving, and meme theory.

Address: Kadon Enterprises, Inc., 1227 Lorene Dr., Suite 16, Pasadena, MD, 21122, USA.

Email: kate@gamepuzzles.com

Heptalion Challenge #4

Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.
On the Effects of Biasing Win Conditions

Ingo Althöfer, Friedrich Schiller University

Martin Hartisch, Siegen University

There exist two-player games with more than one way to win. Different weights for different win conditions may make play more interesting or challenging. The following question is addressed for two modern games with elements of chance: How and by how much do changes in the win weights change the character of the game? It turns out that in both games different win weights lead to nonmonotonic frequencies of the win types.

1 Introduction

A two-player board game may have more than one win condition. A basic setting is to give all of them the same win weight. However, there are classic examples with different weights, for instance Backgammon with win weights 1, 2, and 3 for different levels of win (normal win; win before opponent has borne off any pieces; and win before opponent has borne off any pieces and they have a piece on the bar).

In the creation of a new game with more than one win condition, the designer may ask what the ‘best’ win weights are. More abstractly, one may ask how different win weights lead to different characteristics of a game.

To keep things simple, we only investigate finite symmetric two-player games with zero-sum payoffs. So, if a player who wins by condition $A$ gets reward $w(A)$, and the opponent has to pay the same amount $w(A)$. Analogously, a win by condition $B$ leads to a reward of amount $w(B)$.

In this paper, we look at games with two different win criteria: either moving an own piece to a goal cell or capturing all of the opponent’s pieces. Without loss of generality, a player gets 1 point for winning by goal cell and $c$ points for capturing all pieces of the opponent, which we shall call ‘kill win’. In general, $c$ is a real number; it may even be a negative one.

Obviously, optimal play depends on the value of $c$. In a game without chance, only seven cases for the $c$-value have to be distinguished: $c > 1$, $c = 1$, $0 < c < 1$, $c = 0$, $-1 < c < 0$, $c = -1$, and $c < -1$. Backward analysis (see [1]) of such a game needs only five different values for the nodes in the game graph, namely $\{1, c, 0, -c, -1\}$. 0 occurs when a game ends by a draw, for instance by repetition of position.

Things are theoretically more interesting when chance is involved. For instance, at an inner chance node $x$ with successors $y$ and $z$ (each one with probability $\frac{1}{2}$) backward analysis gives expected score:

$$E(x) = \frac{1}{2} \cdot E(y) + \frac{1}{2} \cdot E(z).$$

The repeated occurrence of chance nodes leads to a very large set of possible node values $E(x)$, and different $c$-values may lead to different $E$-values and also to different optimal moves.

Originally, one question drove our investigation: will a higher $c$-value always lead to higher frequencies for a game to end by a kill win? It turned out that the answer is ‘no’. Even more surprisingly, the frequency of kill wins does not always depend monotonically on the $c$-value. We exhibit our findings for two modern games with chance: ‘EinStein Würfelt Nicht!’[1, 2, 3], and ‘Karl’s Race’[4].

The paper is organised as follows: In Section 2 we present two rather different ways to analyse a game: Monte Carlo search and complete backward analysis. In Section 3 the results of Monte Carlo runs for the game EinStein Würfelt Nicht! are given. The data presented in Section 3 are based on a complete backward analysis of Karl’s Race. We conclude in Section 4 with a short discussion and some open problems.

2 Pure Monte Carlo Search

Pure Monte Carlo search with parameter $k$ works as follows: In a position $X$ with player $I$ to move, all direct successor positions $Y_i$ of $X$ are determined. For each $Y_i$, $k$ random games are played to the very end, and the outcomes are recorded. The $Y_i$ with the best score (from the viewpoint of player $I$) is the move chosen to be played. This procedure is called MC($k$). Of course, MC($k$) is

---

1 Althöfer, I., On the origins of EinStein Würfelt Nicht!, http://www.althofer.de/origins-of-ewn.html

only a heuristic and makes suboptimal decisions sometimes.

2.1 EinStein Würfelt Nicht!
The game EinStein Würfelt Nicht! (abbreviated to ‘Ewn’) was designed in Summer 2004, and published in different versions by the companies ‘Edition Perlhuhn’ and ‘3-Hirn-Verlag’. In Spring 2006, Ewn was added to the portfolio of the turn-based internet game server LittleGolem.net.

EinStein Würfelt Nicht! (Ewn) is played on a $5 \times 5$ grid. Each player has six pieces, numbered from 1 to 6.

![Figure 1. Starting configuration for Ewn.](image)

The pieces are arranged randomly in opposite corners of the board, as shown in Figure 1. A move is performed by rolling a six-sided fair die and moving the corresponding piece. If the piece with the rolled number is no longer on the board, the player moves a remaining piece with the next highest or next lowest number. The player starting in the top left corner can only move pieces one square to the right, one square down or diagonally down right. The player starting in the bottom right can only move a piece left, up, or diagonally up left. Any piece in the target cell of the move is removed from the board. The goal is to either reach the opposite corner square or to remove all of the opponent’s pieces.

Ewn was a category in the annual Computer Olympiads in Tilburg (2011), Yokohama (2013) and Leiden (2015). The rules of Ewn can be found below and are explained more thoroughly in [2].

The game tree size (board with $5 \times 5$ cells and $6 + 6$ numbered pieces) was intended to prohibit complete analysis of the game. Indeed, complete analysis seems out of reach so far [3]. However, it was found that pure Monte Carlo search gives rather strong AI players for Ewn. MC(50) plays Ewn as well as most human players, and is a good sparring partner for new players.

2.2 Results
Using MC(50) for both sides of play, we executed seven Ewn matches of 1,000,000 games each, for difference values of $c$. White started in all games. The following table shows: 1) kill win parameter $c$, 2) total number of kill wins, 3) relative number of kill wins per player, and 4) percentage of kill wins.

<table>
<thead>
<tr>
<th>$c$</th>
<th># kills</th>
<th>(White - Black)</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.11</td>
<td>38,542</td>
<td>(15,811 - 22,731)</td>
<td>3.9%</td>
</tr>
<tr>
<td>0.33</td>
<td>43,875</td>
<td>(18,297 - 25,578)</td>
<td>4.4%</td>
</tr>
<tr>
<td>0.50</td>
<td>46,341</td>
<td>(19,463 - 26,878)</td>
<td>4.6%</td>
</tr>
<tr>
<td>1.00</td>
<td>52,287</td>
<td>(22,984 - 29,303)</td>
<td>5.2%</td>
</tr>
<tr>
<td>2.00</td>
<td>47,089</td>
<td>(20,073 - 27,016)</td>
<td>4.7%</td>
</tr>
<tr>
<td>3.00</td>
<td>43,256</td>
<td>(18,345 - 24,911)</td>
<td>4.3%</td>
</tr>
<tr>
<td>9.00</td>
<td>40,792</td>
<td>(17,439 - 23,353)</td>
<td>4.1%</td>
</tr>
</tbody>
</table>

1,000,000 games played for each $c$ means that three digits are statistically significant. The main finding is: with a standard kill weight $c = 1$, more than 5% of games end by kill win. The greater the distance between $c$ and 1, the fewer kill wins. Interestingly, the percentage of kill wins does not increase monotonically with the parameter $c$. Another observation: for all seven $c$ values explored, the second player (Black) had significantly more kill wins than the starting player (White). A natural explanation for this is that Ewn games are short (between 16 and 17 moves on average), hence the first player typically plays more offensively, which increases their risk of having all pieces captured.

The fact that in Ewn kill wins occur with such low probability for all tested values of $c$ may be seen as a game design weakness. But when the game was designed (back in 2004), it was never intended to make the two win types occur with equal frequency; kill wins were only intended as an emergency alternative goal. Otherwise, a player with no legal moves would have to pass until the opponent won by reaching their goal.

3 Backward Analysis

3.1 Karl’s Race
Karl’s Race is a little brother of Ewn, but played on a board with 21 square cells of different sizes (from $1 \times 1$ to $6 \times 6$). Each player has six similar pieces (without numbers) which are placed on the board as shown in Figure 2.
Figure 2. Starting configuration for Karl’s Race.

When the die shows \( d \) pips, a move is made from a square of size \( d \times d \). If the player has no pieces on any square of the rolled size \( d \), the player moves a remaining piece from a square of the next highest or next lowest size. As in Ewn, a win can be achieved either by reaching the goal cell or by capturing all of the opponent’s pieces.

The parameters of Karl’s Race (abbreviated to KR in this paper) are 21 cells instead of 25 cells and pieces without numbers, making its state space much smaller than that of Ewn. This allowed M. Hartisch to perform a complete backward analysis in his masters thesis [4] for versions with up to \( 6 + 6 \) pieces.

3.2 Backward Analysis

According to the official KR rules (from 2006), each player has six pieces. In that case, only about 10% of perfectly played games end by a kill win, when equal weights for the two win criteria are used [4]. For the investigation in this paper, we were looking for situations in which the two types of win occur more similarly often.

Kill wins are more frequent when there are fewer pieces at the start. So it was natural to look at KR with three or four pieces for each side. We abbreviate KR with three and four pieces by KR(3) and KR(4). Both starting configurations are displayed in Figure 3. For each \( c \in \{-1, -0.9, -0.8, \ldots, 1.4, 1.5\} \) (weight for a kill win) we used backward analysis to compute the database of optimal moves for each position.

For negative values of \( c \), the term kill win is a bit misleading as killing gives a negative score; it is more of a kill end with a negative reward. Nevertheless, we will use the term ‘kill win’ even when \( c \) is negative. For each \( c \) and both KR(3) and KR(4), 100,000 games were played by perfect agents and the overall percentage of kill wins was recorded. Figure 4 shows the results.

3.3 Results

The standard kill weight \( c = 1 \) results in about 19% of the games ending by kill win for both games. With \( c \) reduced slightly to 0.9, this percentage drops to 16.7% for KR(3). For \( c \) at \(-0.3\), the frequency of kill wins visibly peaks for both games. There is a nonmonotonic interval (a slight valley) between 0.3 and 0.7. For \( c \)-values below \(-0.3\) another nonmonotonic interval is visible: surprisingly, the rates for both KR(3) and KR(4) drop and then rise again.

For \( c \)-values below \(-1\) no experiments were performed. This is simply due to the nature of the game: it is almost impossible to force the opponent to capture one’s last piece. Hence, a game with \( c < -1 \) almost always ends by reaching the goal cell.

With our goal of a 50/50 balance in mind, we see that for KR(3) a \( c \)-value of 0 provides this ratio almost perfectly. Choosing \( c \) as 0 can be interpreted as counting a kill win as a draw. For KR(4) the percentage of kill wins is always smaller than 50%. For \( c = -0.3 \), the rate is closest to this goal and reaches 0.465.
We should keep in mind that the tiebreak rule used, when encountering two moves with the same score, has a large effect on the results. When both moves have a negative score (i.e. the expected payoff is negative) we select the move with the larger expected game length. This strategically delays our probable loss, to increase the chance of a mistake when playing against an imperfect opponent. On the other hand, we select the move with the shorter expected game length when encountering two positive scores, in order to secure the win as quickly as possible with minimal risk.

The effects of this tiebreak rule are visible particularly for \( c = -1 \). Even when a payout of \(-1\) is certain, i.e. all moves have a negative score of \(-1\), we rather obtain it by letting the opponent reach the goal cell than by making the last capture for a kill win.

4 Discussion

A game is a microcosm, often with simple rules as in the cases of Ewn and KR. Nevertheless, seemingly small changes in the rule set may have severe and unexpected consequences.

4.1 Win Weights in other Games

It would not be popular to change the win weights in the classic game Backgammon. Nevertheless, we ask which weights for the three levels of win, instead of \( \{1, 2, 3\} \), would give the ‘most interesting’ play? And would there be another factor for the doubling cube other than 2 which results in ‘more interesting’ play?

Chess has three possible outcomes: win for White, win for Black, and draw (or stalemate). It has previously been proposed to distinguish between stalemate and a normal draw, for instance by counting a stalemate as a half win for the stalemating side and half loss for the stalemated opponent. This idea was recently revived by correspondence Chess grandmaster A. Nickel, as a way to resolve the problem of too many draws in correspondence Chess tournaments. Nickel called this variant ‘Lasker Chess’.

Havannah by Christian Freeling is a modern classic board game. There are three ways to win a game: by a fork (connecting three of the six sides), by a bridge (connecting two of the corners), or by a ring. Each of these win types have the same value, but could other value tuples be more interesting? As an aside, there are a now strong AI players for Havannah based on Monte Carlo tree search, in at least one of which the different win types were weighted differently to achieve stronger play.

Hearts is a four-player card game. Each one of the thirteen cards in the hearts suit counts \(-1\) point, the queen of spades counts \(-13\) points, and all other cards have value 0. Is \(-13\) the optimal value for the queen of spades? This value is not directly a win or loss weight, but we list it as a parameter that affects the winning chances and optimal play.

---

\(^3\)This variant was proposed by World Champion Emanuel Lasker in 1918.

\(^4\)http://boardgamegeek.com/boardgame/2759/havannah

\(^5\)http://boardgamegeek.com/boardgame/6887/hearts
The dice game Yahtzee\footnote{http://boardgamegeek.com/boardgame/2243/yahtzee} is another case in point. In addition to each player trying to maximise their score over the entire game as per usual, we propose a variant in which any player at least $k$ points ahead of each other player is immediately declared the winner. Such a rule would dramatically change the character of the game, with moderate values for $k$, as players would be encouraged to try for the higher scoring categories earlier in the game than they usually might. The question is, what would be a good choice for parameter $k$?

In an arbitrary three player game, a normal scoring system would typically give 2 points for the winner, 1 point for the runner up, and 0 points for remaining player. A parameterised version might give $c$ points for rank 2, with $c$ between 0 and 2. What are typical consequences for small or large values of $c$? Even the cases with $c > 2$ or $c < 0$ might be interesting.

Also, puzzle games with different win conditions may be designed, for instance including a ‘number of moves’ limit. When a challenge is solved in less than $k$ moves, the full score is given, otherwise the partial solution at moment $k$ is rated by some scoring system. What are good ranges for $k$? How will $k$ affect the playing style of players?

### 4.2 Relevance Beyond Game Design

This impact of minor design changes may also have relevance to economic concerns, such as legislation, tax rules, and the stock market, etc., as seemingly simple changes or new paragraphs may have completely unexpected or unwanted consequences. For example, Althöfer and Baerthel\footnote{Althöfer, I., Combinatorial space trajectories, http://www.althofer.de/space-trajectories.html} analysed a ‘simple’ matrix game model in which a transaction tax was introduced, which in many cases did not calm down the simulated ‘market’ as intended but instead led to higher expected transfers.

Small changes in the conditions of a real world scenario may lead to large and unexpected changes in the behaviour of the agents involved. Surprises are not only possible, but happen quite regularly.

Further, the first author has experience with the design of complicated space trajectories with multiple targets\footnote{Scherer, K., Square the square, http://karlscherer.com/prosqtsq.html}. Such missions may include flybys (typically with high speed) and rendezvous with asteroids or comets. Rendezvous are much more interesting than flybys. Giving score 1 for each rendezvous and score $c$ for each flyby, different values of $c$ often lead to completely different ‘optimal’ solutions.

### 5 Conclusion

We have investigated two-player games with two different win conditions: either by reaching a goal cell or by capturing all of the opponent’s pieces. Normally, both types of win result in the same score, but changing this to give 1 point for one win type and $c$ points for the other, may result in surprising outcomes between strong players. The frequency of the second win type does not depend linearly on $c$, and not even in a monotonic way.

For game designers, this means that the selection of appropriate win weights is a nontrivial business. Even seemingly small changes in a rule set can have a surprising effect on play.

### Acknowledgements

Back in 2005, Joerg Sameith programmed the McRandom tool in C++ which allows to perform experiments with pure Monte Carlo search for a large class of two-player games. McRandom was used for the Ewn experiments reported in Section 3. Thanks also to Karl Scherer for his alternating tilings and mosaics\footnote{Scherer, K., Square the square, http://karlscherer.com/prosqtsq.html}. We further thank the anonymous reviewers for their constructive criticism.

### References


Dr Ingo Althöfer is Professor of Mathematical Optimisation at Jena University, who has a decade long background in computer-assisted game play and design. His most successful commercial board game Torjaeger (Kosmos) sold more than 170,000 copies.  
**Address:** Institut fuer Mathematik, Fakultaet fuer Mathematik und Informatik, Friedrich-Schiller-Universitaet, Ernst-Abbe-Platz 2, 07743 Jena, Germany.  
**Email:** ingo.althoefer@uni-jena.de  

Michael Hartisch is a PhD student at Siegen University, who studied mathematics and has experience in the analysis of several boardgames. His research interests include the use of Monte Carlo methods in various fields of research.  
**Address:** Chair of Technology Management, Institute of Business Admin., Siegen University, Hoelderlinstr. 3, 57076 Siegen, Germany.  
**Email:** michael.hartisch@uni-siegen.de  

---  

**Heptalion Challenge #5**  
Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.
Serendipity in Game Design

Raf Peeters, Smart

The notion of serendipity is well known to game and puzzle designers. This article describes my own personal experiences with serendipity as a professional puzzle game designer, and highlights its importance in the development of two published puzzle games.

1 Introduction

From 1997 to the present I have been working as a product developer and designer at Smart, a Belgian toy company. Most of my work involves the development and preproduction of quality toys, games and puzzles.

I am a designer, not a researcher, but have been involved in every step of the development of many Smart games, from creating the first idea until the product is on the market. The following observations are based on my personal experience in game and puzzle design.

2 Inventing and Serendipity

The idea of ‘inventing’ sounds magical to many people, as if being creative is about waiting for new ideas to pop up. However, I believe that it has less to do with luck or talent, but more with serendipity.

The notion of serendipity is well known in game design circles [1, p.308]. I am referring to the phenomenon of finding something unexpected and valuable, while searching for something else.

‘Finding something unexpected’ still sounds like you just need to be lucky. Although each creative process does include some luck, for me there are two other aspects about serendipity that I find more important: searching and finding.

2.1 Searching

First of all, the inventor needs to be actively searching, rather than just waiting until the right idea comes to mind. As a designer I therefore spend a lot of time reading, thinking, looking around for anything that might be interesting; it may not look or feel like ‘work’ but it is.

A lot of creativity techniques are nothing more than ways to actively search for new ideas. The misconception that inventing is like waiting may have its roots in two famous stories about inventors: Archimedes’ ‘Eureka!’ moment in his bath tub, and Newton’s epiphany while sitting under an apple tree. Such stories give the false impression that inventing is something that happens through passive waiting, rather than through actively searching for new ideas.

2.2 Finding

Secondly, the inventor needs to recognise the potential of the things he or she discovers. It is not enough to search, if you do not know when you have found something good, no matter how unimportant it might look at first.

One reason that potential new ideas are overlooked may be due to the misconception that inventing starts with one big fantastic idea. However, it is often a new combination of many small things, such as a fascinating theme, a technical solution and a nice shape or material. Each of these aspects may not be very special on their own, but the combination brings it to a new level. Designing is a lot like cooking; not only are the ingredients important, but also the manner in which they are prepared.

3 Originality

If you ask people what they find important in a new (puzzle) design, they most often say that it needs to be original. I see the following approaches to novelty:

1. The design tries to improve an existing idea or design (the result will probably be evolutionary).
2. The design starts with an existing idea, but changes a few key elements.
3. The design starts with a new idea (the result might be revolutionary).

Each approach has its pros and cons. Incremental or evolutionary designs work, and can offer real benefits, but the improvements are often small. On the other hand, different is not necessarily better; there is a good reason why a table has four legs and not five. I prefer designs...
that work over those that may be new but which fail to deliver.

The word ‘original’ is often used for designs that are only ‘different’. Original really means that a design is the origin of something new, the first of a new breed, something that will inspire future generations and variations. Making things different is not a guarantee of quality, and in most cases it is just a dead end.

Sadly, ‘different’ is often good enough for marketing purposes, if a product can be labelled ‘new’. However, doing things differently can still be a good approach, if it is taken as a starting point rather than an end point. By this, I mean that the design process should involve different gradients of novelty, an iterative process in which new ideas are tried, leading to further new ideas to make them work, and so on. It is during this process that I often find something truly new and worthwhile; serendipity in action.

4 Example: Penguins on Ice

Penguins on Ice, one of my designs, is one of Smart’s most successful puzzle games [2]. The object is to fit five pentomino-shaped pieces on a 5×5 square grid, so that the penguins on each piece occupy the positions shown on a given challenge card. An example is shown in Figure 1.

There are 60 challenges of increasing difficulty.

![Figure 1. Penguins on Ice challenge and solution.](image)

What makes this game special is that each of the five pieces contain a sliding mechanism, which allows them to be reconfigured into different shapes. Figure 2 shows the five pieces and their valid configurations. Each piece has a three-dimensional penguin on it, so that by design the pieces cannot be used upside down.

The reconfigurable tiles are the aspect of the game that players seem to find most original and interesting. But when I started the design process, the aim was to make a puzzle with fixed pieces that could not be transformed.

A lot of people seem to like puzzles with pentominoes When I was asked to design a puzzle with pentominoes, the first step was to look to existing pentomino based games. What most of these had in common was an abundance of pieces, often the complete set of twelve pentomino shapes. So one way to be different would be to make a game with relatively few pentomino pieces.

I like to create puzzles with as few pieces as possible for several reasons: they look less complicated (and less frightening for the average consumer); setting up challenges is easier; checking solutions is easier; etc. And from an aesthetic viewpoint, fewer pieces usually mean a cleaner and more efficient design.

However, restricting the number of pieces to five did not allow a sufficient number of interesting challenges to be devised. Making the pieces reconfigurable was the innovation that overcame this problem; the game still only has five pieces, but now has 17 possible shapes to choose from.

---

1A pentomino is a connected planar shape made by joining five congruent squares edge to edge.

2See Kate Jones, ‘Symmetrical Pentomino Pairs’, p. 46 of this issue.
Another thing that most existing pentomino puzzles had in common is that they are abstract. So again, to be different, I wanted to add a theme. Ideally, a game’s theme should be consistent with its rules, so that the theme and rules complement and reinforce each other, but there are not many things in real life based on square shapes that can transform. However, a bit of thinking soon found the perfect solution: penguins on ice floes that are constantly sliding and changing shape. Figure 3 shows the final product.

Contrary to what might be expected, the two most visible distinguishing features of this game – the reconfigurable pieces and the penguin/ice theme – came about as solutions to design problems that occurred when we tried something new. This is a clear example of finding serendipitous results while searching for new ideas.

5 Example: Quadrillion

Quadrillion\(^3\) is another puzzle game that I designed for Smart [3]. But instead of reconfigurable pieces, I wanted to try something different: *what would happen if the game board itself was adaptable?*

The result was the four double-sided 4 x 4 sub-grids shown in Figure 4, each with an arrangement of white dots on one side and black dots on the other side. Each subgrid also contains two magnets on each edge, which allow the four to be snapped together in different board configurations. The magnets keep the subgrids together and help ensure that they are correctly aligned with their neighbours.

The object of the game is to make a specific board configuration, then fill this with the ball shaped puzzle pieces (shown in Figure 5), such that all cells are occupied by pieces except for those with a white or black dot.

The next step in the design process was to make a selection of starting positions to include in a challenge booklet. But the algorithm used to check solutions produced something unexpected: *every possible configuration of eleven particular board layouts had at least one solution.*

As long as the player configures the board in one of the key shapes shown in Figure 6, it will always be possible to create a solvable challenge. This allowed us to change the object of the game: instead of setting up a challenge from a booklet and solving it, the player has the more innovative task of devising and solving their own challenges.

For example, if the subgrids are arranged in the 2 x 2 square configuration, then there are 24,576 possible arrangements of the black and white dots, the ‘best’ of which allows 95,385 valid fillings and the ‘worst’ of which allows 216 valid fillings.

\(^3\)Not to be confused with the game Quadrillions, described in the article ‘Symmetrical Pentomino Pairs’ on p. 44.
Figure 6. The eleven solvable tile configurations.

The L-shapes allow 98,304 possible arrangements of the black and white dots. But if the subgrids are arranged in the T-shape, then there is a difficult configuration with only one valid filling.\(^4\)

Again, while trying something different and searching for solutions to the problems that arose, we stumbled upon something else of value. This allowed us to change the object of the game; from setting up a challenge from a booklet and solving it, to the more innovative task of getting players to devise their own challenges.

6 Conclusion

My own experience in game design has led me to appreciate the importance of not only constantly searching for new ideas, but to evaluate them sufficiently that I recognise those of true value. The two examples presented, Penguins on Ice and Quadrillion, both show the discovery of interesting new features that were not originally intended, but which were the serendipitous result of searching for viable solutions in response to trying something different.

Because of time constraints, there is often pressure to limit the design process to something linear: given a problem, analyse it and produce a solution. But this should only be the starting point of the process. It is sometimes better to be adventurous rather than efficient, and to try different things just to see what eventuates, if serendipity is to occur in the design process.

References


Raf Peeters is lead designer at Belgian company Smart, which distributes original toys, games and puzzles to over a hundred countries.

Address: Smart, Neerveld 14, B-2550, Kontich, Belgium.
Email: raf@smart.be

---

\(^4\)Figures calculated by software written by Smart programmer Saskia Oortwijn.
Embed the Rules

Cameron Browne, Queensland University of Technology (QUT)

This article examines the concept of embedding the rules of a game in its equipment, as a general pattern for good game design. Several examples demonstrate the benefits of doing so. An analogy is drawn with the Japanese concept of poka-yoke, or mistake-proofing, in manufacturing design.

1 Introduction

This series on Game Design Patterns aims to explore fundamental principles that encourage good game designs. While this term is reminiscent of the ‘game programming patterns’ described in the book of the same name, its use there refers to software programming practices in video game development, whereas a ‘game design pattern’ here refers to any practice that encourages good designs in games and puzzles. This first installment in the series looks at the concept of embedding the rules in the equipment.

To ‘embed the rules’ means to use relevant features of the game’s physical components (board, pieces, environment, etc.) to enforce implicit rules which then do not need to be explicitly stated to players. This might also be described as ‘hiding the rules in the equipment’, or phrased as an aphorism: hide the forest in the trees.

This approach can have significant benefits when designing games, including: simplifying rule sets; minimising player error; handling degenerate geometric cases; allowing the emergence of implicit strategies; providing tutorial assistance to players; and so on. These benefits are examined in the following sections.

2 Poka-Yoke

Simpler rule sets generally reduce the incidence of player misinterpretation or error – provided that no crucial information is simplified out! – and lead to more conceptually elegant games. Simpler rule sets also have the significant benefit of giving the player less information to remember, allowing them to concentrate instead on strategic planning and actually playing the game, rather than the mundane bookkeeping of calculating which moves are legal or not. The trick is to simplify the rules as much as possible, while ensuring that the game remains complex enough to be interesting.

For example, consider the Heptalion puzzle shown in Figure 1, in which the aim is to place the five tiles, each showing a pair of symbols, to exactly cover the set of matching symbols on the left. These rules are trivially simple and instantly intuitive; no player who has played the game should ever need to reread them.

Despite its simplicity, Heptalion is actually the result of considerable thought and effort by its designer Néstor Romeral Andrés, as described in the article ‘Heptalion’ on pages 11–17 in this issue. This is a case of embedding the rules in the components (i.e. the two patterns on each tile must be played as a pair, symbols must match identical symbols, etc.), making the game so intuitive that misinterpretations and mistakes are almost impossible.

Romeral Andrés makes the astute observation that this process of mistake-proofing is an instance of poka-yoke. This is a Japanese term that refers to any mechanism in a manufacturing process that helps an equipment operator avoid mistakes by drawing attention to human errors as they occur.

In the context of game design, poka-yoke can be seen as the concept of reducing player error, by simply making the equipment not allow such mistakes to occur in the first place, either explicitly or implicitly through its design.

---

1The concept of ‘design patterns’ was introduced in 1977 by Christopher Alexander in the context of architecture.
2Proposed by Richard Reilly, personal correspondence, 12 June 2015.
3Romeral Andrés first encountered poka-yoke when researching the asymmetric design of the VGA plug that virtually eliminated incorrect dockings by users. If only the USB design committee had learnt from this lesson!

2.1 Explicit Embeddings

Rules can be explicitly embedded in the equipment to achieve *poka-yoke* (mistake-proofing). Consider Ploy, shown in Figure 2 which is an early example from 1970 [4]. The markings on each piece act as instructions that indicate direction and distance of travel; each piece can move in the direction of one of its markings, a number of cells up to its total number of markings.

![Figure 2. A game of Ploy about to start.](image)

Such visual cues have been embedded in the pieces of many games since then. The designs on the Mijnlieff pieces, shown in Figure 3, intuitively indicate where the next player must move relative to the piece just placed: orthogonally in line, diagonally in line, adjacent, or nonadjacent.

![Figure 3. The four Mijnlieff piece designs.](image)

The 5×5 *icon grid* embedded in pieces of The Duke [5] shown in Figure 4 explicitly show the player what actions each piece can perform, and where. Navia Dratp [6] uses a similar mechanism, while Confusion: Espionage and Deception in the Cold War [6] subverts this idea by showing movement rules on the pieces, which are initially visible *only to the opponent*, and which must be deduced through play.

![Figure 4. Visual cues on a piece from The Duke.](image)

Abstract rules involving concepts other than movement can also be semantically embedded in the equipment, e.g. by stating them on playing cards. There are many examples of this in popular card games such as Magic: The Gathering [6] and Dominion [6] (Figure 6), in which the cards describe the actions available to players. Even the Chance and Community Chest cards in Monopoly [6] explicitly state instructions for the players, which they do not need to memorise – or even worry about – until each card is drawn.

![Figure 5. Directional tiles of the Trippples board.](image)

Visual cues may also be embedded in the board, to constrain piece movement intuitively. For example, each square in the child’s board game Smess: The Ninny’s Chess [7] is marked with arrows showing valid directions of travel.

Similarly, the 1972 game Trippples [6] involves 64 tiles each showing three directions of travel (Figure 5), which the players alternately place in an 8×8 square grid to form the board for each game. However, players’ movements are not decided by the arrows under their own pieces but under their opponent’s pieces instead, in a twist that is both *poka-yoke* and counterintuitive at the same time.

---

[6] I will use the BoardGameGeek online database as a catch-all reference for games mentioned in passing.
The standard deck of 52 playing cards also demonstrate mistake-proofing in their rotationally symmetrical illustrations. Players’ cards are always the right way up in their hands regardless of what orientation they were drawn in, otherwise many would be upside down when drawn.

A superb example of poka-yoke can be found in the puzzle game Tantrix. The Tantrix set consists of 56 hexagonal tiles showing all ways in which the tile sides can be joined by paths of four colours, such that each tile includes paths of three different colours (except for three straight lines connecting opposite sides).

The Tantrix tiles are numbered 1 to 56, such that each tile has a number on its back in one of the path colours (Figure 7, top). This allows the tiles to be used for a number of puzzle challenges, in addition to the standard game. Starting with the first three tiles, the number 3 is yellow, so the player’s first challenge is to form a closed yellow loop with those three tiles (Figure 7, bottom). Then the next number 4 is red, so the player’s next challenge is to form a closed red loop with those four tiles, and so on, up to tile number 10.

The Tantrix set therefore embeds a set of puzzle challenges within its equipment, by judicious numbering and labelling of the component pieces. Further, each challenge is mistake-proof, without the need for explicit additional rules, as the piece describing each challenge shows the number of tiles required and the colour of the target path.

2.2 Implicit Embeddings

Rules can also be implicitly embedded in the equipment to achieve poka-yoke, typically by the shape of the board and pieces, without the need for further visual decoration.

Figure 8 shows a game of Tixel in progress. Tixel pieces are designed with a circular concavity on one side, so that some can rotate in-place in 45 degree increments while others are blocked from rotating, depending on the placement and orientation of the surrounding pieces. This mechanism simplifies the rule set and enforces this potentially confusing constraint in a clear, intuitive way.
Christian Freeling’s Loonybird Chess, from 1983 [7], shows how existing tropes can be exploited to create new hybrid pieces with implicitly defined movements. The cardinal pieces in the Loonybird Chess set consist of two parts, an upper *hunter* and a lower *carrier*, and take on the role of each part depending on whether they are moving or capturing, respectively.

Figure 9 shows a white ‘knight-rook’, which moves like a knight but captures like a rook, and a black ‘knight-bishop’, which moves like a knight but captures like a bishop. Players familiar with Chess will immediately understand the movements of these unusual pieces, despite not having seen them before, without having to memorise new rules (beyond the upper/lower distinction).

Figure 10. A game of Terrace ready to start (photo by Geni Palladin [6]).

Implicit rules may also be delegated to external sources, such as the word game Scrabble [9] (which has enough rules of its own) tapping the players’ cultural knowledge of language to define which letter combinations are valid and which are not. This outsourcing of rules to an external authority brings a huge richness but also ambiguity to the game, depending on the players’ language, level of vocabulary, dictionary being used, level of competition, etc., necessitating the creation of explicit word lists for official play.

Roleplaying games are another example of the delegation of rules, but this time to the imaginations of the players themselves. Given a basic premise and some fundamental rules to work within, the players control the direction of play through their understanding of the hypothetical game world, and their own interpretations of what actions can and cannot be legally (and sensibly) performed therein [10].

3 Tutorial Help

Another benefit of embedding the rules in the equipment is as a tutorial device for players. Consider the ancient Japanese game of Shogi [10], shown in Figure 11, which has a notoriously high barrier to entry for new players.

Shogi pieces exhibit some degree of mistake-proofing in their design, as they are pentagonal in shape with an apex that points towards the opponent, to clarify whose pieces are whose. But players must recognise the *kanji* characters on each piece and remember their associated movements, and must deal with the fact that captured pieces are reentered onto the board as the opponent’s.

---

9 [http://www.looneylabs.com/looney-pyramids](http://www.looneylabs.com/looney-pyramids)
10 Based on an observation by Richard Reilly, personal correspondence, 19/6/2015.
A number of publishers have sought to reduce this learning curve by showing piece movement visually on the pieces themselves, such as the Kumon ‘Study Shogi’ set shown in Figure 12. Note that these pieces still show their original kanji glyphs, so are primarily tutorial aids.

Figure 12. Kumon ‘Study Shogi’ set (photo by Mike Fogus).

Figure 13 shows another approach to helping new players learn Shogi, through known iconography rather than explicit instruction. Pieces show icons of familiar Chess pieces where possible, with the more exotic pieces (specific to Shogi) represented by icons in a compatible aesthetic style, that should not be hard for a Chess player to learn.

Similarly, players who have trouble remembering the movement rules in Robert Abbott’s popular Chess variant Ultima can play with a tutorial set, in which each piece shows an icon of its closest Chess equivalent, decorated with additional cues as needed. Figure 14 shows such a pictographic piece set by Fergus Duniho.

Figure 13. Hidetchi’s Internationalised Piece Set from Nekomado (photo by Russ Williams).

Figure 14. Pictographic version of Ultima pieces, © 2001–2014 Fergus Duniho.

http://play.chessvariants.org/pbm/play.php
4 Emergent Strategies

It is a natural bonus for the designer – and players! – if emergent strategies occur in a game, as a side effect of the interplay between the equipment, rather than due to explicitly stated rules.

Consider the Trax position shown in Figure 15. Trax is already an elegant and poka-yoke game, as both players share a single tile type, with two distinct sides, on which coloured paths visually define the connectivity for each player. However, the geometry of this tile design created a problem, as holes surrounded by tiles on all four sides could too easily become unplayable, if those four sides did not have two path ends of each colour.

To fix this problem, the game’s designer, David Smith, introduced a forced move rule that ended up adding a new dimension of strategy. Consider the move shown in Figure 15 (middle), which creates a three-sided gap \( f \). There is only one tile orientation that will match the three exposed path ends, so that tile orientation must be played there as part of the move (right). Such forced moves can trigger other forced moves, and so on, to achieve complex and interesting results with a single move.

Smith did not specifically design for this strategy, it emerged as a natural consequence of the game’s geometry. But once players know this rule, it is then the equipment itself that explicitly defines such forced moves.

Trax is also interesting in that it has both an intuitive ‘form a closed loop’ goal, as well as a more arbitrary ‘connect opposite sides of a virtual 8\( \times \)8 grid’ goal. The intuitive goal is almost mistake-proof, while the arbitrary goal can be confusing for players, as it relies on external rules outside the equipment.

\[\text{Figure 16. A partially solved Slitherlink challenge (left) is more comprehensible after region colouring (right).}\]

\[\text{http://www.traxgame.com}\]

\[\text{Personal correspondence, 9 March 2015.}\]
Figure 16 shows another case of a strategy emerging from the inherent geometry of a game. This case involves the deduction puzzle Slitherlink, in which the solver must trace a simple (i.e. closed and non-self-intersecting) path through the orthogonal vertices of a square grid, to visit the number of sides indicated on each numbered cell. This example is challenge #57 from [6].

Figure 16 (left) shows the challenge mostly completed, with a few loose path ends, but it may be confusing for readers not experienced with Slitherlink to deduce where to proceed from here. However, we can turn to mathematics and exploit the Jordan Curve Theorem to give some insight into the problem. This theorem states that any simple curve has an inside and an outside, and any cross-section completely through it will cross its boundary an even number of times [12]. We can, therefore, colour regions of the grid that must lie inside the final path, as shown in Figure 16 (right), which immediately provides the next move to make; the circled path ends must deviate away from each other and not join, otherwise the coloured region would be illegally cut off. Another strategy that follows from the Jordan Curve Theorem is that any horizontal or vertical line through the grid must intersect an even number of edge segments, which can provide crucial additional information when solving complex challenges.

Mathematics, and geometry in particular, is a rich source for embedding implicit rules and constraints into equipment. Consider the connection game, Hex, shown in Figure 16, in which players take turns adding a piece of their colour to an empty cell, and aim to connect their sides of the board with a chain of their pieces [3].

This simple rule set belies the strategic depth of Hex, as the mere inclusion of the concept of ‘connectivity’ brings with it a whole slew of implications. As software engineer Phil Bordelon says: it’s like you get extra rules for free [3, p. 347]. However, it is possible to achieve an even simpler rule set, with comparable depth, through a slight change in board design. In the early 1950s, computer pioneer Claude Shannon proposed the game of Y, shown in Figure 17 in which players aim to connect all three sides of a triangular grid of hexagons with a chain of their pieces [3].

Figure 17. A game of Hex won by Black, who has connected the black sides with a chain of black pieces.

Figure 18. A game of Y won by White.

Y simplifies the Hex equipment by removing the need for the board edges to be coloured and for players to each have a defined direction (how many times have I confused which direction is
mine in a game of Hex, upsetting my plans?). However, Hex is still the more strategically pure game, as players can focus on their single line of connection with considerable certainty, while the divergent connection threats in Y can soon get quite complex. Y could be described as clearer from a design perspective, while Hex is clearer from a strategic perspective.

5 Exploit the Geometry

This section describes ways in which the geometry of the equipment can be exploited, to implicitly enforce rules and help make games more mistake-proof, clear and elegant.

5.1 Deadlocks

It is no coincidence that the two connection games described above, Hex and Y, are played on a hexagonal basis. The hexagonal grid is *trivalent*, i.e. no more than three cells meet around any intersection, which allows the beautiful property that exactly one player must make a winning connection, and no game can ever be deadlocked for a tie. The board geometry itself avoids the need for an explicit tiebreaker rule, hence many connection games have a hexagonal – or at least trivalent – basis. See [3] for more details on this point.

By contrast, consider the hypothetical game shown in Figure 19, which is identical to Hex but played on a square (i.e. nontrivalent) grid. This example shows the white and black paths deadlocked around the central point, and this game cannot now be won by either player. In fact, neither player is ever likely to win this game, which is a huge disincentive to play it.

Connection games designed for the square grid typically involve some mechanism to transcend the geometry and avoid such deadlocks. For example, the game Quax\(^{14}\) avoids this problem by introducing an alternative move type, in which players can bridge across diagonally separated pieces, if such a move would not cross any existing bridge. Figure 20 shows a bridge that wins the game for White.

This bridging mechanism solves the problem neatly, and allows some new and interesting strategies. Admittedly, it does introduce a new movement rule, but it is still *poka-yoke* in that it makes the game playable with as few rules as possible by exploiting the geometry.

The game Gonnect [28] provides another example of simplifying a rule set, while exploiting the geometry to solve a potential deadlock problem at the same time. Gonnect is a blend of Hex and Go, in which two players try to connect their sides of a square grid with a chain of their pieces, but also play with the surround capture, *ko*, and no-suicide rules from Go. However, it eliminates an important rule in Go, as players cannot pass\(^{15}\).

To see the effect that this has on the game, consider the position shown in Figure 19. This position may appear to be heading for a stalemate, but it is not just *cold* – a term from combinatorial game theory meaning that the only moves

---


\(^{15}\)Often called the ‘Gandalf rule’ after his ‘you cannot pass!’ line in Tolkien’s *Fellowship of the Ring*.
available to the mover are disadvantageous – it is freezing, as the next player to move will lose.

Each of the four groups on the board have two eyes (internal holes) which in Go would make them safe from enemy intrusion, as the opponent cannot fill both holes to capture the group in the same move. But since passing is not allowed in Gonnect, and suicide moves are not allowed, the next player to move is forced to play in one of their own life-giving eyes, allowing the opponent to play in the other eye to capture that group next move and achieve a winning advantage. Removing the passing rule not only simplified the rule set but elegantly solved the problem of temporary deadlocks.

5.2 Cycles

The ko rule from Go states that a player cannot repeat the board position from the previous turn, and is necessary to avoid loops of play. Figure 22 shows a classic ko situation, in which White captures a black piece, but Black cannot immediately recapture it back on the next turn. This rule could be described as a ‘bug’ that has become a ‘feature’ over the millennia, and entire volumes and schools of thought are now devoted to ko analysis.

It was therefore a surprise when the computer program LUDI [12] produced the game Nden-grod [16], in which two players compete to make a line of five of their pieces on a hexagonal grid, while using the surround capture and no-suicide rules from Go but not the ko rule! The game played well and did not suffer any problems with cycles, despite the lack of this apparently crucial rule.

Analysis soon revealed what was already known among Go researchers, that moving the surround capture mechanism from a square basis to a hexagonal basis removes the need for the ko rule. This is illustrated in Figure 23 where White has just captured a black piece with move c, but there is no danger of an immediate return capture and hence no need for ko. This is essentially due to the lack of diagonal connections in the hexagonal grid; see [4] for more details. LUDI had inadvertently chosen a geometry that avoided the problem of cycles, simplifying the rule set.

5.3 Stacking

The underlying geometry is obviously fundamental to games in which pieces stack on each other. The wrong choice of geometry here can cause serious problems that require additional rules to fix – if they can be fixed at all – while the right choice can produce elegant solutions with fewer rules. Consider the hypothetical marble stacking game, shown in Figures 24 and 25, in which White has laid a hexagonal platform of white marbles, upon which Black has stacked a single black marble.

In Figure 24, Red stacks two marbles adjacent to the black marble, then adds a further marble on top to achieve a complete packing. All marbles in this example can be described as existing in the same phase, let’s call it ‘phase I’.

Figure 25 shows the other option available to the player, which is to stack a marble across

---

16 Later marketed as Pentalath: http://www.nestorgames.com/#pentalath_detail
from the black marble so as not to touch it. The black marble is still in phase I, but the new marble is in phase II, ruining the packing as no further marbles can now be stacked in this example.

Mixing phases like this not only ruins the stacking, but can have serious implications for connection games played using this geometry. Games may become unwinnable due to the ease with which gaps can be created between groups of pieces, or games may become trivially winnable due to the ease with which potential blocking moves can be spoiled.

The designer can address this problem by introducing a rule that all marbles must be played in the same phase. But this solution is inelegant: it is hard to explain, hard to enforce, and error prone. It relies on players understanding the problem and correctly recognising the phase of each potential move, which is onerous and unrealistic for large board sizes; the players will spend most of their time just working out which moves are legal.

David Bush offers a more poka-yoke solution to this problem in his game Lazo [3]. This is a connection game played with pieces that stack within a hexagonal basis, which avoids the phase problem through the cleverly shaped pieces shown in Figure 26.

Each piece has a triangular peg on the bottom, which forces it to face in a particular direction when slotted into a board hole. Then whenever three adjacent pieces meet around a gap (Figure 26, left), that gap will also be triangular and facing in the same direction. Any piece that stacks is therefore forced to face in the same direction, due to its triangular peg, ensuring that all pieces remain in the same phase (Figure 26, right).

This is another example of exploiting the geometry to hide redundant rules and reduce player error. The inventor of Lazo has since reshaped the pieces to make the interstitial gap larger, so that players can see the colour of the piece underneath more easily, mistake-proofing the game even further through the equipment.

Another solution to the phase problem is to choose a different geometry altogether. For example, Figure 27 shows that marble stacking is not subject to such phase problems using a square basis. This is the reason that my own marble-stacking games, such as Akron [3], Margo [17] and the Shibumi set [18], all use a square basis.

But what about the problem with deadlocks on the square grid, described in Section 5.1? It turns out that stacking games on the square grid transcend this problem, due to an unexpected emergent property.

Figure 28 (left) shows a graph in which the vertices are the centres of the visible marbles in a complete square packing, and the edges correspond to pairs of touching marbles. The dual of this graph (right) is trivalent, which is the necessary condition for deadlock-free connection; a complete packing on this grid is guaranteed to produce a connection between opposite sides.

This fortuitous quirk of geometry makes the square basis suitable for marble packing connection games, such as Akron, without the need for additional deadlock avoidance rules. Further, Akron also has a drop mechanism, in which marbles can be removed under some circumstances to let higher supported marbles drop down. This is another case of exploiting the available geometry (and in this case gravity) to implicitly enforce physical rules that do not need to be stated to the player. The well known Connect Four [6] is another example of a game that exploits gravity to enforce a rule, namely dropping pieces to land on the highest empty slot of a chosen stack.

German designer Michail Antonow used an equivalent tiling to the dual shown in Figure 28 (right) for the board of his excellent connection game Conhex [3], although arrived at his design from a totally different approach.
6 Conclusion

The examples above demonstrate the benefits of embedding the rules in the equipment of a game. Such benefits range from proofing against player mistakes (poka-yoke), to more subtle considerations such as design elegance, improving clarity, and tutorial assistance for players. This is certainly what I aim for in my own designs: to move as much of the game into the equipment as possible.

Assuming that it is generally good to embed the rules in the equipment, the question remains which part(s) of the equipment to move them into. For example, the game For the Crown is a marriage of Chess and Dominion, with labelled pieces and cards that show relevant instructions and legal moves (as opposed to The Duke, in which the movement rules are inscribed on the pieces themselves). This separation of pieces from their relevant information makes it easy and cheap to replace the information cards, to allow new options and expansion packs for the game.

This discussion has focused on abstract board games, and connection games in particular, as these are domains in which embedding the rules has obvious benefits. While I propose this as a general pattern for good game design, there will of course be exceptions: war games in which studying 100-page rule books is a necessary price of admission to the detailed simulation experience; narrative-based games in which added complexity enriches the atmosphere of the game for greater player immersion; and so on.

Acknowledgements

Thanks to: Richard Reilly, Russ Williams, Mike Fogus, Néstor Romeral Andrés and Stephen Tavener for discussions; Christian Freeling, Mike Fogus, Geni Palladin, Michael Kandrac and Jay Tummelson for image permissions; and the reviewers for their suggestions. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

---

18Observation by Nathan Morse, personal correspondence, 18 June 2015.
Centipede: From Arcade Game to Board Game

Bruce Whitehill, The Big Game Hunter

This article describes the process of creating a ‘traditional’ board game from a successful arcade game. This involved rephrasing the key aspects of the arcade version in traditional board game design tropes that Centipede players would still recognise, while shifting the emphasis of game play from dexterity and reflexes to strategy.

1 Introduction

I once had the good fortune to work as a game inventor for Milton Bradley Company in Springfield, Massachusetts. Early on, I was given an assignment: to make a board game out of the popular arcade game Centipede shown in Figure 1, which is said to be the first arcade game designed by a woman. Assignments like this are normal, being based on what drives the larger companies in the toy and game industry in the United States: licensing.

You are told to take a popular movie, television show, book or character, and turn it into a predominantly competitive game that can be played, usually by two to at least four players. You are told the age range and maybe the price point, which gives an idea of the pieces and materials that can be employed. In most cases, you are starting from an ‘artistic’ work and turning it into something playful.

But, in this case, the challenge was quite different; I was beginning with a game – and a very popular one, at that – and changing it into another game. The challenge was that the initial game was one with action, requiring reflexes and speed, producing a sensory rather than a contemplative experience. Another difficulty in adapting any computer or arcade game which has already met with success is that you are expected to replicate this success in an entirely different medium.

2 Approach

In general, adapting an arcade game into a board game requires maintaining the basic components and goal of the original, while substituting a decision making mental process for the rapid action hand and finger movements of the computerised game. This usually requires adding dice or a spinner to the game play, which, inevitably, adds a considerable amount of luck to the operation. Also, because of the solitary nature of arcade games and the components that would be involved in making a board game, I felt it was better to limit the number of players to two.

2.1 Precedents and Subsequents

Centipede, the board game, was released in 1983. In 1981, Frogger became one of the first arcade games to be made into a board game. The popular Pac Man, along with Zaxxon and Donkey Kong, were introduced as board games in 1982.

---

1 Centipede the arcade game is trademark™, Atari®. Centipede the board game is ©1983 Milton Bradley Co.
2 StrategyWiki: http://strategywiki.org/wiki/Centipede
Other popular titles that followed included Q-Bert (1983), Super Mario Brothers (1988) and Tetris (1989). Many other board games based on arcade games have been produced by a myriad of companies since the 1980s.

2.2 Preparation

It was not necessary to study precedents, since each game adaptation is unique. I was not much of an arcade game player, but knew the classics such as Pac Man and Donkey Kong, and had even played the arcade game Centipede. I had just finished working on Fraggle Rock, a children’s board game based on a popular animated television show, where my work day involved watching hours of television. Now my day was to consist of playing Centipede repeatedly in the local video game arcade.

After some futile attempts at getting a decent score, I was still unable to reach a high enough level to see how things change. I observed a young teenager with lightning finger action doing well on other arcade games, and enlisted his help by offering to pay for his games (at Milton Bradley’s expense), provided that he would explain to me what he was trying to do and how he was going about it. Figure 16 shows a screen shot of the Atari® arcade version in action.

I took less than copious notes, jotting down sequences of events, recording what appeared, disappeared and changed on the screen as the game progressed. Back in the research and development facility (the Milton Bradley plant in East Longmeadow), I spent the next few days creating a two-dimensional paper version of the game. I fashioned a path that represented the systematic yet variable route the centipede takes on its journey. In designing a singular creature (that is, the multisegmented centipede), I needed to have each segment independent of the others so that, when shot, a segment could be altered; in this case, turned over. The art department, responsible for the board and pieces, developed the three-dimensional, two-sided plastic centipede segments, which were attached by an interlocking mechanism so they could be easily flipped over.

2.3 Philosophy and Practicality

Centipede was not going to work as a solitaire game (you can not ‘play against the game’ itself) or a multiplayer (more than two players) game. But the board’s flat surface and the centipede’s path allowed for a two-player game in which there were two centipedes, each starting near one player’s home base and heading toward the other, with the aim being to avoid your own centipede and destroy your opponent’s.

In a sense, the task of developing a game based on another one is easy, as all the elements are there already. On the other hand, you have to take ‘used’ material and create something new out of it. You have to consider all the things that are happening in one version and try to incorporate them into the other. There is the moving of the centipede, the shooting, the ‘killing’ of centipede segments, and the various elements that can work for or against you, such as the mushrooms, scorpion, spider and the flea. Then, for everything that happens, you need to determine how often you want it to happen, mathematically working out the preferred relative frequencies of the actions.

The spinner is designed so that centipede movement will occur more often than the other events, i.e. that area occupies more space of the spinner. I had decided on a different percentage for each event, but sometimes, especially in games, art triumphs over function: the art department required the spinner to be symmetrical (and their decisions overruled the designer’s), so the gun space and the shared spider-scorpion space had to be the same size, as shown in Figure Ā. My goal was to make the board game as close to the arcade game as possible, although players would be substituting tactics for action; the text on the box and game board (written by the copywriters) read, ‘Based on the Action-packed Arcade Game by Atari®’, and ‘action-packed’ did not seem applicable for a race game on a board. I had to take a ‘skill and action’ element based on trackball control and adapt it to a strategy component that had nothing to do with speed.
3 From Screen to Tabletop

This section describes the actual steps I took to implement these observations, and create a traditional board game version of Centipede that captured the essence of the original arcade game.

First, there had to be the movable gun action, in this case, restricting the gun to be slid along the base line so that the players could shoot in a straight line toward their opponent. In the arcade game, the scorpion, spider, flea and poison mushrooms all play an important part, but it would be difficult to translate all their actions into a board game. Besides, the game should not become too complicated in an attempt to duplicate all of the actions of the arcade version.

The solution was to eliminate the flea, but incorporate its actions into the scorpion and spider, so that they could be used to place or remove poison mushrooms, respectively (mushrooms are a key element in allowing shortcuts for your centipede). There is the balance between creating a shortcut for yourself but not your opponent. In addition, the spider can be used to temporarily silence your opponent’s gun. Figure 4 shows the final design of the board and components.

In the arcade game, points were awarded for shooting obstacles on the screen. A continual point-accrual system for the board game would be too cumbersome, so the initial decision was to make the board game a race game: the first player to get all six centipede segments into the opponent’s home base wins.

Whereas shooting a segment of the centipede in the arcade game would split it into two centipedes (giving two foes), hitting the opponent’s centipede in the board game would kill the segment and allow the player to turn it over; as the centipede moves one space for each ‘live’ segment in its body, ‘dead’ segments slow it down.

In the arcade game, the player is battling against the machine, which will eventually win: the objective is to destroy the centipede(s) before they reach your home base. In the board game, the player is competing against an opponent and turns alternate. Players have choices throughout the game, and their success depends on which choices are taken and when.

The arcade game did not rely on luck very much, if at all, whereas the board game is driven by the spinner that forces certain actions on each turn: move the centipede, or move and fire the gun, or move either the scorpion or the spider, or about an 18% chance of a ‘Free Choice’ of any of these actions. The game was made in 1983, with no victory point track, no score keeping, and no cumulative actions to keep track of.

As a lithographed sheet of paper glued to an 18.5” x 18.5” piece of cardboard (the same size used by Milton Bradley himself in 1876), Centipede was certainly a simple race game. The final cover art is shown in Figure 5. Centipede is now in the Computer Games Museum in Berlin along with a working copy of the original arcade game, both of which may be played by visitors.

---

3Computerspielemuseum, Karl-Marx-Allee 93a, D-10243 Berlin, Germany: www.computerspielemuseum.de
4 Conclusion

The motivation for creating board game versions of arcade games was to provide a rendering for children old enough to know about these arcade games, but who were too young to enter the arcades or who lived too far away from them. The age range listed on the box (dictated by management) was ‘Ages 7 – 14’. The process of transforming a game from a visual electronic medium to a traditional tabletop one required using as many elements of the arcade game as possible, while combining and simplifying some actions, and incorporating strategic components to replace speed and action.

It would be interesting to know if the people who played Centipede in the arcades also played the board game. My guess is they may have tried it once or twice, but then they probably went back to the high speed, action packed, noisy arcade game. Centipede, the board game, was, I think, more for both the younger siblings and for people like me – the older ones who could never do that well on the dexterity based arcade game, but who enjoyed competing against another analytical player at the dining room table.

Acknowledgements

Thanks to Sybille Whitehill for editorial advice and proofreading.

Bruce Whitehill, known as ‘The Big Game Hunter’ (http://www.thebiggamehunter.com), is a researcher on the history of American games and their links to other countries and cultures. His current research interests include asymmetrical games, Clue and Cluedo and its variants, and changes in game play and theme for the same game in different cultures.

Address: Alte Poststr. 18, D-29693 Eickeloh, Germany.

Email: games@thebiggamehunter.com
Project GIPF: Design and Aesthetics

Mitchell Thomashow

Project GIPF is a system of six conceptually related but distinct abstract board games, which represent a milestone in the evolution of game design. The purpose of this essay, the first in a series on Project GIPF, is to interpret these games from a cohesive design, aesthetic and cognitive perspective.

1 Introduction

Project GIPF is comprised of six abstract board games: GIPF, ZERTZ, DVONN, YINSH, PUNCT and TZAAR. It is clear from designer Kris Burm’s games and writing that he designed Project GIPF as an integrated, aesthetic whole, interested not only in the quality of the games as competitive thinking experiences, but also as works of art, and as a testimony to abstract games as reflective of human creativity through space and time.

Project GIPF has many of the qualities of classic abstract games, and respects that tradition, but it also embodies some innovative and contemporary play dynamics, including balance, order, chaos and improvisation. My intention is to provide a narrative, phenomenological account, assessing my own experience as a player, while situating the games in a broader context. But first I would like to comment on Project GIPF as an exercise in creative imagination.

2 The Games

I conceive Project GIPF as a whole, with each of the games serving as centres. Although all of the games implicitly reference one another, GIPF is the node in the network, through which the others ultimately flow. Consider each game as a playing field in a more complex arena, or perhaps as a habitat in an ecosystem. GIPF contains the whole field, serving as the central organising game, the integrating narrative of the whole series. You may think this is too far flung, but the beauty of GIPF, is that it serves as an abstract game playing narrative. This narrative reflects two levels: the picturesque shifting landscapes of players and boards interacting together in the Project GIPF network; and the cognitive skills required to play the games well.

GIPF is based on a classic concept and can be played on three levels. The players start with 18 pieces each. In turns, they introduce one piece into play until achieving four-in-a-row. A player who forms such a row, takes it from the board and captures any opponent’s piece which extends that row. The purpose is to form successive rows of at least four pieces, until the opponent has no piece left to bring into play or (in the standard and tournament version) no more GIPF-pieces on the board.

You may wonder how an abstract game can be ‘picturesque’. To my game playing aesthetic, there is no more vivid visual entertainment than watching a game of Go as an emergent system. Picture the black and white stones as microorganisms, cellular automata, weather systems, populations – whatever your imagination conjures.

Project GIPF evokes a similar aesthetic of simplicity and Zen-like beauty. The stark and imaginative box covers, shown in this article, promote such imagery. GIPF resembles the origins of continents, ZERTZ a polar ocean, DVONN a volcanic landscape, YINSH a weather front crossing an open ocean, PUNCT the interface between land and sea, and TZAAR a shifting desert setting.

\[1\] TZAAR replaced an earlier game in the series, TAMS, in 2007. TAMS used hourglass pieces for an unusual temporal aspect.

Zértz is played with 6 white, 8 grey and 10 black marbles on a shrinking board. The aim is to capture either 4 white, 5 grey or 6 black marbles, or 3 marbles of each colour. To capture, one must jump a marble over another marble. Both players play with the same marbles, so each threat you create on the board can be used against you. The only way to control play is through making sacrifices, but you must make sure that in the end you capture more than you give away.

Figure 2. Zértz.

Imagine Gipf as continents and oceans forming shape on a young earth. Zértz is about islands, submersion and isolation. Dvonn reflects orogeny and erosion. Yinsh is a tempest of weather fronts sweeping through a landscape. Tzaar is a wind blown plain. Punct represents creatures gaining a foothold in a tidal zone.

Each game in Project Gipf involves its own set of cognitive skills and perceptual abilities that correspond to the playing field, a shifting abstract landscape. Although each game presents unique cognitive demands, there are some overarching principles that permeate the series. The following sections cover three relevant design dynamics that I find particularly interesting: balance, improvisation, and order versus chaos.

2.1 Balance

Balance is the overriding conceptual theme in each of the games. With Gipf and Yinsh, every capture you make requires that you also remove your own pieces. With Dvonn, every capture limits your mobility. Zértz strategy often entails giving up a ball to gain one or more different balls. Tzaar sacrifices mobility for power. You must make sacrifices to gain an advantage.

There is a balance of power: if your position becomes stronger than the opponent’s, you may find your prospects reduced. Your ability improves in each of the games, as you come to recognise how to achieve balance while gaining an advantage.

For example, Yinsh requires players to remove one of their own movement pieces when five-in-a-row is achieved, giving the opponent an advantage on the board. And in Tzaar, you must capture an enemy stack each turn, which becomes harder as the pieces become fewer. There is a balance between progressing toward the goal and weakening oneself on the board, a compelling feature common to all games in the series.

2.2 Improvisation

Improvisation is reflected in the rapidly shifting landscapes and circumstances of each game. It is difficult to see more than a few moves ahead, as each move can unleash a complex chain of events. This is particularly true in Gipf and Yinsh.

As a musician, this aspect intrigues me, as good improvisational music demands the ability to react to a changing musical circumstances. A musician’s skill in improvising is enhanced by understanding musical patterns and structures, which may sound similar but demand different responses in various settings.

Similarly, players come to recognise themes in the form of common patterns of pieces. These can be elusivey familiar, as if you have seen something like it before, but are not sure exactly where. The Project Gipf games are inextricably linked, as common structures become increasingly clear, but manifest themselves differently, depending on which game is being played.

Dvonn is a stacking game. It is played on an elongated hexagonal board, with 23 white, 23 black and three red Dvonn-pieces. The players place all of their pieces on the board, then begin moving. Players total the pieces in their own stacks, and the higher score wins.

Figure 3. Dvonn.
In **YINSH**, players each start with five rings on the board. Every time a ring is moved, it leaves a marker behind, white on one side and black on the other. When markers are jumped over they are flipped. Players try to form lines of five markers of their own colour, each one of which requires them to remove one of their own rings. The first player to remove three of their own rings wins. Each removed ring therefore brings you closer to victory, but also weakens your potential for movement.

**2.3 Order versus Chaos**

Each game is an experiment in order versus chaos, and every match presents a landscape of seemingly random or chaotic changes. A single move can trigger a series of cascading effects that dramatically change the patterns on the board. These may seem random in **TZAAR** and **YINSH**, but they reveal new emergent opportunities, and sometimes a move will create a wholly unexpected pattern, even for experienced players.

Yet as in chaos theory, the appearance of complicated nonlinearity does not necessarily imply a lack of coherent structure. The cognitive challenge in playing the Project GIPF games is to make enough sense out of this chaos to determine what might become a winning move.

There is an intrinsic entropic flow that you, the player, must turn into some coherent tactical or strategic advantage. ‘Find the order in the chaos’, the designer’s aphorism for **YINSH**, serves as guidance for the entire project.

The qualities discussed above (balance, improvisation and order versus chaos) are most directly reflected in **GIPF**, **YINSH** and **TZAAR**, as well as in the overall strategic concept of Project GIPF. By contrast, **ZERTZ** is a highly tactical, plan-ahead game of compulsory capture, in which chaos is found in the unpredictably moving chains of islands and rings. **DVONN** is a familiar game of connection and territory. However, the seeding of the board results in a different game each time, and you must always assess the means and manner of how you choose to co-agulate. **PUNCT** is a more traditional connection game, but its three dimensional quality literally adds a compelling cognitive direction, perhaps a twenty-first century Hex, in which getting from one place to another is not always what it seems.

**3 Relationship**

The most original aspect of Project GIPF is the relationship between the games. The designer offers two related, but discrete methods for connecting the games. With **GIPF** as the home landscape, you can play with so called ‘potentials’ from each of the other games. Players select the type and number of these potentials, each of which reflects its parent game’s main characteristics; a **ZERTZ** piece jumps, a **DVONN** piece can form a tower, a **YINSH** piece adds extra mobility, and a **PUNCT** piece neutralises or converts an opponent.

Further, attempting to use a potential can trigger a challenge, in which the players interrupt the current game to play the corresponding subgame (e.g. **ZERTZ** for a **ZERTZ** potential), to determine whether that potential can be used. In fact, any game may be used as the subgame, e.g. Chess or even **GIPF** itself, by mutual consent.

**PUNCT** is a connection game in which the goal is to link two opposite sides of the hexagonal board. Each turn, the mover either brings a new piece of theirs into play, or moves a piece of theirs already on the board. Connections are typically only achieved through piece movement, and the more pieces brought into play, the more possibilities for connection. **PUNCT** is therefore more about misleading the opponent, by playing seemingly harmless moves with the potential to move to form strong connections.
In TZAAR, each player has 30 pieces, divided into: 6 Tzaars, 9 Tzarras and 15 Totts. The three types of pieces form a trinity and cannot exist without each other. Each turn, a stack can capture and remove an enemy stack of equal or lesser height by moving onto it, or can move onto a friendly stack to strengthen it by making it higher. A player loses if they run out of any of their three piece types, or if they cannot make a capture. Strengthening stacks reduces the number of stacks and hence overall potential for movement, while capturing too many enemy stacks risks leaving your own stacks lacking in height and power; it is again a game of balance.

From a design perspective I deeply admire how these games connect, and how they form an interactive system, with the specifics of the interactions left up to the players. Each game is a world in its own right. Nevertheless, the meta-narrative implications are obvious, as ambitious players switch landscapes to determine the use of potentials. This is evidently where the deepest and most original strategic implications reside.

As you gain strength in any particular game, so you become more adept at using its potentials, and players get to select the types of potentials that they play with. The ultimate GIPF player will develop competence, if not excellence, in all of these arenas, in order to succeed at this most complex version of Tournament GIPF.

This is an extraordinary time commitment, especially given the proliferation of modern board games, and the eclectic tastes of most contemporary players. If the games in Project GIPF are as deep as they seem, then mastery in any one game requires a huge investment of time and experience. Yet the series as a whole respects the eclectic taste of an abstract gamer: suggesting that playing all of the games at once, or perhaps some in combination, yields insights into any specific game. Indeed, each game, while unique, also reiterates its companions. TZAAR and DVONN utilise towers, mobility, and capture in similar yet contrasting ways. GIPF and YINSH explore mobility and transformation through pattern removal. ZïRTZ and PUNCT at first glance seem less connected to the whole, yet both have themes of expansion, connection, sacrifice and isolation that serve as reminders of the other games.

One of the least appreciated, or at least most implicit, virtues of abstract games is how they teach various cognitive skills: spatial and temporal relationships, sequential arrangements, territory and topology, anticipation and scenario development, structure and pattern, and the various themes referred to above: balance, improvisation and order versus chaos. Abstract board games are often recommended for elementary school students, and I believe that the Project GIPF games in particular offer a wonderful opportunity for early learning. Through such immersion, the student/player can learn about aesthetics (these games are ever-changing works of art), design (complexity emerges from simple concepts), and respectful competition (how to appreciate a game played well together). Project GIPF is a brilliant game design concept because these qualities are intentionally developed, not in a theoretical manner, but rather as intrinsic to a philosophy of aesthetics, learning and play.

4 Conclusion

In the last few decades, there has been an extraordinary proliferation of games. As Huizinga reflects in the classic Homo Ludens [2], game playing is intrinsic to human culture. But which of the games currently being played will still be played several centuries in the future? More likely it will be an abstract game, as such games transcend boundaries of culture and language; you can probably teach your opponent how to play without even speaking their language. Abstract games epitomise a purity of form and pattern, which is the point of Project GIPF.

Wrapped in aesthetics, cognitive depth, and the pure fun of moving pieces on a board, the entire GIPF series is unique as a whole system. Perhaps in its interconnected complexity, based on relatively simple games, with its clear allusions to the classic games that preceded it (Go, Reversi, Draughts, Hex, etc.), it will become a twenty-first century classic in its own right. This series of games feels like it could have been played in the remote past, and could well have relevance in the distant future. But I feel that it could only have been invented at this moment in human history.

Alas, I must plead inexperience playing with potentials!
when visceral relationships still matter, and moving stones on a playing board lends meaning and purpose to the human condition.

Acknowledgements

This piece is based on an earlier review of Project GIPF in *The Games Journal* [3], significantly reworked and updated, Thanks to Kris Burm for providing images and rules summaries, and to Smart Games for image permissions.

References


Mitchell Thomashow is the author of *Ecological Identity, Bringing the Biosphere Home* and *The Nine Elements of a Sustainable Campus*. He writes for COUNTER magazine and is currently exploring music, games, ecology and improvisation.  
**Address:** 62 Cedar St, Apt 511, Seattle, Washington, 98121, USA.  
**Email:** mitchellthomashow@gmail.com

Heptalion Challenge #6

Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.
Architecture in an Abstract Game

Kris Burm

A friend pointed out to me that abstract games feel like architecture. I agree that the art form with which they seem to have the most in common is indeed architecture. For that reason, I consider a designer of abstract games to be in fact a ‘game architect’.

1 Exterior and Interior

Although architects have more facilities to realise visionary dreams, the similarities between a building and an abstract game are obvious due to their applied nature. Just like a building, a game also has a purpose through which it is connected to its use. A building that cannot be entered or counteracts its function is like an unplayable game. For both, the boundaries that must be respected to keep the real world in balance with a free creation are often very concrete and therefore one is obliged to respect them.

Practicability and makeability are dependent on environment, materials and processes. Light, space, material, surrounding and functionality are at an architect’s disposal, in the same way that pieces, grids, spaces, ways to move and capture, and goals are available to a game designer.

The functionality of a building is the purpose of a game. The shape and external characteristics of a building as well as of a game must support that functionality and purpose. Entering a building is synonymous with starting a game. Wandering around in it is exploring the game. Spending more time in it and experiencing the light, space and layout, working or living in it, is discovering the possibilities a game has to offer.

One could ask if it is necessary to enter a building in order to enjoy it. No, a building can also be admired from outside, and nor is it a prerequisite to have seen the exterior to understand the interior. But both perspectives together make the experience complete, discovering how the outside continues in seclusion on the inside and, in the other direction, how the inside presents an assumption about the surrounding exterior.

2 Elements of Surprise

The Brazilian architect Oscar Niemeyer (1907–2012) claimed that it is not enough to just design the right building with the right functions; there also need to be elements of surprise in it. ‘Surprise is key in all art’ is one of the famous quotes attributed to him.⁴

These elements of surprise reside in the interaction of the applied features in a previously untried manner, and in how the obvious and the ambiguous cancel each other out. You could say that the two are being played out against each other. If the combination – or confrontation – of the expected and the unexpected is executed with a clear vision, then that which could be presumed as indistinct is brought to the point of being distinct. This is paraphrasing what I have often put forward about the design of abstract games.

This relationship between the exterior and the interior is also relevant to games. The shape of the pieces, the amount per player, the number of spaces (squares, intersections, . . . ) and the layout of the board on the one hand; and, on the other hand, how these external characteristics are incorporated by the rules and appear to only have a useful destination within this specific context, is what can make a game more than just a game. Only then can the outside (the material) and the inside (the exploration of possibilities) resonate with each other, and by playing the game it becomes clear why the ratios between the features are exactly so and not otherwise. That goes equally for a building that is more than just a ‘right building with the right functions’.

3 Stepping Into a Game

Both architecture and abstract games are applied artforms. But what if you look at a building, or walk around in it, without the postulate that it is a building? Or if you are exposed to an abstract game without realising that it is a game? If the functionality is removed and you only take into account what a particular building or game is separate from its use, does that building or game then become redundant? Are they then constructs that have lost their legitimacy?

At the Insel Hombroich Museum in Neuss, Germany, stands the Turm (1987–1989), a piece by German architect and sculptor Erwin Heerich (1922–2004). From the outside, the Turm (tower) looks like an enormous brick cube with edges of roughly ten metres (Figure 1).⁵

⁴http://www.reuters.com/article/2012/12/06/us-brazil-niemeyer-idUSBRE8BS0120121206

Figure 1. The Turm (tower) by Erwin Heerich.

The top half has two hollow cube shaped volumes, diametrically opposed to each other, and each side has a narrow door. The construction looks like a robust block, and even though you know and feel upon entering it that the interior structure correlates with what you saw on the outside, it is hard to reconcile the two as being one and the same building.

The ‘surprise’ is that something can be so monumental and at the same time so light, spacious and elegant. But an even greater surprise for me was the sensation of having entered a game. The similarities were blatant; it almost felt virtual, a place where you are not actively present, but where that which surrounds you takes the initiative to determine that you are there.

Nothing but flat, horizontal and vertical surfaces, all white. In the centre, two straight beams shoot upwards to support two massive cubes: the struts of the counterparts of the cube shaped hollows, of the nonexisting volumes on the exterior. The part that could be considered the ceiling – the top side of the cube minus the smaller cubic hollows forming an angular ‘8’ or ∞ shape – is a design that consists of square misty glass panes arranged in a grid, suggesting a playing board, unreachably high (Figure 2).

Although light floods in, the misty panes do not allow you to observe the outside space . . . but it is there! There is the horizon of the universe that surrounds you, there is the limit to what you can see, feel and comprehend. Being in that space was – at least, to me, at that moment – very similar to being absorbed in a game situation and realising that there is a vast unknown behind the pieces on the board. I felt the urge to make a move.

Figure 2. The inner ceiling of the Turm.

4 Conclusion

The reason that I remember the Turm so well is because I did not for a single moment think of it as a building, as something ‘functional’ or ‘useful’. None of the definitions that might be given to the concept of ‘building’ were relevant.

If you play without any notion of the concept of ‘game’, then you are no longer moving pieces across the board with the aim to win. Playing to win is relevant, of course, or you would not be playing a game, but it should not be the main concern. At stake is a sense, an imagining, or even an awareness – not to be confused with the understanding! – of what an abstract game is doing to you. It is as much an emotional as a cognitive process. Experiencing a game is as important as trying to get the most out of your next move.

Despite a game’s tangibility to the game itself, and the fact that playing is a shared experience, this is something that one can neither describe nor explain. One can of course try to put it into words, as I have tried with the example of the Turm. But it is too easy to remain stuck in vague rhetoric, circling the core of the matter, which is often the case when being knocked off your feet by a work of art.
Acknowledgements

This piece is based on an extract from ‘The Right Move’, a manifesto by the author on abstract games as art (currently available only in Dutch).

Kris Burm is a designer of abstract two-player games including Project GIPF, a series of six interconnected board games.
Address: 2970 Schilde, Belgium.
Email: donco@gipf.com

Heptalion Challenge #7

Use all tiles to cover all symbols. See ‘Heptalion’ (p. 17) for details.

Heptalion Challenge #8
Defining the Abstract

J. Mark Thompson

This article, first published online in 2000, explores the nature of abstract strategy games, and identifies four key qualities that a game should possess if it is to have lasting merit: depth, clarity, drama and decisiveness. An addendum summarises the story behind the article, and my thoughts on the piece 15 years later.

1 Introduction

Games belong to a unique category of artefacts. Like literary works, their internal logic is firmly grounded in humanity in what kind of things human beings are able to understand and to do, and what kind of things give human beings intellectual pleasure. Like epic poetry, many of them are very old, and have attained their present perfection through the contributions of many minds, by virtue of which they express the spirit of a culture. And yet like musical compositions, they exist on a level of abstraction that floats above any physical realisation. On one hand a good game may be as esoteric as a theorem of pure mathematics, and on the other it is as human as anthropology. Nothing we do is as human as thinking, and no thought is as human as the thinking we do for mere amusement.

There are many ways to categorise games. For example, there are outdoor games involving athletics; but worthy as these are, I have nothing to say about them here. There are games in which an element of chance is decisive, such as roulette or Snakes and Ladders: these games are appropriate only for children (if they are complicated) or for gamblers (if they are simple). Roulette, for example, is so simple it would have no interest unless money were involved. On the other hand, Snakes and Ladders is so complex it would have no interest unless money were involved. On the other hand, Snakes and Ladders is so complex it cannot interest an adult, even a gambler, who would correctly perceive it as a tediously complicated way of flipping a coin. Still, it has value in showing children what a game is and how to play: there are rules that must be followed, there is a winner and a loser, if you lose it does not matter much because you may win another time. In fact, a strong element of chance is desirable in a game for children because without it they would have no chance of winning when playing against their elders, and so would not learn the attraction for games that every civilised person should feel.

But as long as it is not usually a decisive factor, an element of chance does not exclude a game from serious and intelligent interest. Backgammon is an excellent example; even though a match between good players may be decided by the favour of the dice, a good player will almost always defeat a poor one. The same goes for Pachisi, and many card games, such as Bridge or Poker, also call for thinking of the highest level.

Many board games are built around a theme. For example, the fine recent game Settlers of Catan has the theme of settling and developing an uninhabited island. Other games have been devised, with varying degrees of success, around themes of the stock market, murder mysteries, the battle of Gettysburg, the plays of Shakespeare, etc. Many commercial publishers put out games that seem void of originality or interest, relying on a popular theme to attract customers. Games for children are again a separate case, since a game need not be original to be fresh to them. But speaking of adult games, although a theme might enhance the play of a good game, a good game never gets its value from its theme.

2 Abstract Strategy Games

The games that interest me most are abstract strategy games. The word ‘abstract’ is used because such games usually are presented with no theme, or in which the theme is not important to the experience of playing. Abstract games are thus the ‘purest’ of games. Chess, for example, although it has been said to have a theme of war between medieval armies, is clearly an abstract strategy game. Apart from the names of the pieces there is nothing about the game itself suggesting war; it is more suggestive of geometric patterns.

Abstract strategy games furthermore minimise the element of chance. It is essential to their definition that such games have perfect information: each player, when deciding his move, must have complete information about the current position of the board (I include in ‘position’ qualities that may be physically undetectable, such as whether a player may castle), or equivalently, about the original position of the board and all
moves made so far. Examples of perfect information games would include Chess and Backgammon; games like Stratego, Kriegspiel or the recent Stealth Chess are not perfect information games. Also, there must be no chance elements introduced by such mechanisms as dice, cards, or dominos drawn at random: Backgammon is not an abstract strategy game. And players must move alternately (rather than simultaneously, as in, say, RoboRally). There are ordinarily only two players, since competition between more than two generally leads to temporary alliances to defeat whichever player acquires an early advantage, and strategy succumbs to politics.

Contrary to some common assertions, there is an element of luck even in abstract strategy games. For instance, a player might make a move without seeing its value, and later find that any other move would have lost the game. But regardless of such considerations, playing an abstract strategy game is an exercise in logical thought. There is an intimate relationship between such games and puzzles: every board position presents the player with the puzzle, What is the best move?, which in theory could be solved by logic alone. A good abstract game can therefore be thought of as a 'family' of potentially interesting logic puzzles, and the play consists of each player posing such a puzzle to the other. Good players are the ones who find the most difficult puzzles to present to their opponents.

3 Design Qualities

The design of a good abstract game must therefore allow an inexhaustible supply of interesting puzzles to be discovered in the possible positions of its board. Equally important, these puzzles must be discoverable and to some degree soluble by the players. Anthropologists from another planet who wanted to study the way human beings think would do well to study our abstract strategy games.

I will consider four qualities a game must possess to have lasting merit: depth, clarity, drama, and decisiveness.

3.1 Depth

*Depth* means that human beings are capable of playing at many different levels of expertise. For most board positions, until the last stages of the endgame, the puzzle of finding the best move should not be completely soluble. In a deep game, a player must exercise nice judgment in deciding what is the best move in most situations. Depth gives a game lasting interest because the player continues to learn how to improve his play for a long time. If a game has a large following, its depth can actually be measured by recording the results of games and determining how many distinct 'levels' there are: if the players in class 1 all lose regularly to the players in class 2, who lose to players in class 3, etc., up to class n, then the value of n measures the depth of the game. I am told that Go appears to be the deepest of the world's classical games, though some modern games (such as Star and Poly-Y) are contenders that cannot be measured because they still do not have enough players.

3.2 Clarity

But in addition to depth a good game must have *clarity*. Clarity means that an ordinary human being, without devoting his career to it, can form a judgment about what is the best move in a given situation. For example, if a player has a move that will win the game immediately, it should not ordinarily be difficult to find it. Although Chess problems have been devised where a winning move is hard to find, this is usually done by finding a position that misdirects the player's instincts.

In a game that lacks clarity, the player simply has no instincts. Even in the mid game there should be some rules-of-thumb which will usually lead a player to a better position. Robert Abbott, the inventor of the Chess variant Ultima, has lost interest in his creation because he feels it is 'opaque'. Though Ultima has many defenders, anyone who tries to invent a new and original game will find clarity an important issue. The difficulty, with a newly invented game, is to discern whether a game is 'invincibly opaque', or whether with sufficient experience its rules of strategy would begin to clarify.

3.3 Drama

A good game should also have *drama*: it should be possible for a player to recover from a weaker position and still win the game. Victory should not be achievable in a single successful blow; the suspense should continue through an extended campaign. Otherwise, an early disadvantage makes the remainder of the game uninteresting: the doomed player rightly guesses that the puzzle he is trying to solve has no solution and that thinking about it is futile. A game's drama might be measured roughly by matching a strong player

---

against a weak player, and having them switch sides after the strong player achieves an advantage. In a dramatic game, the strong player will still have a chance of winning. But the difficulty of defining ‘advantage’ clearly will make drama harder to measure than depth.

Chess is a dramatic game, but its drama apparently becomes more and more subtle as the players become more expert. Good Chess players rarely play a game to checkmate: they resign when it becomes clear they cannot win, in other words, when the game has ceased to be dramatic. Masters of the game resign when it becomes clear they must lose a piece without gaining in exchange either an enemy piece, or a positional advantage; grand masters may even resign at the loss of a pawn. The drama of Chess, for them, must consist of the alternation of very delicate shades of positional advantage.

3.4 Decisiveness

But in addition to drama, a game must also have decisiveness: it should be possible ultimately for one player to achieve an advantage from which the other player cannot recover. Abalone has been criticised as lacking decisiveness: there appears to be a strategy which a weaker player can adopt (clumping his pieces together and never extending them, even to attack), which makes it impossible for the stronger player to win. In such a game it is the stronger player who faces a puzzle (How can I push my advantage to a victory?) with no solution.

In ‘Peak Performance’, an episode of Star Trek: The Next Generation, an obnoxious alien who is a master of a game called Strategema defeats the android Data. In a rematch, Data plays an obstinately defensive strategy, declining even the most promising attacks, until the master of the game resigns in a fury, unable to win. Pity the poor alien, for Data did more than defeat him; he demonstrated that the game to which he had devoted himself was indecisive, and hence futile.

Edward de Bono’s L-Game \([6]\) is indecisive, and a game between two perfect players would continue forever. De Bono is pleased with this feature, and remarks on it in his instructions for the game, proving that it takes all kinds.

Even Chess at the highest levels is becoming drawish; in matches between world championship contenders, dozens of games are played and most end in draws. Imagine how unsatisfying it would be if contestants for the world championship played fifty games and the victor won 3-2 with 45 draws; one could not help but wonder whether, if the match had been ten games longer, the other player might have been champion.

4 Conclusion

I list these four qualities because they seem to me to be in tension with one another by pairs: depth vs. clarity, drama vs. decisiveness. For example, if a usable algorithm is known which will always reveal the best move in any situation of a game, then the game’s clarity is perfect, but it has lost all its depth. The same is true if any ‘winning strategy’ is known, meaning an algorithm which allows either the first or second player always to win. Such a game (like Bridg-It \([1]\) or Nim \([3]\)) has been ‘solved’. Similarly, if a game provides the underdog with too many opportunities for recovery, it achieves drama but becomes indecisive, or if a player in a stronger position can win too easily the game becomes undramatic. Only rare games achieve the perfect balance, and this makes such games interesting to contemplate as well as to play.

It is common for serious players at Chess and Go to report that they can gain insights into the personality of their opponent from his style of play alone, even in correspondence games between players who have never met. The human spirit is perceived in the mere algebraic notation for the moves. The abstract game, this extraordinary medium of expression, should hold an honoured place among the liberal arts.

Acknowledgements

I would like to thank the editors of The Games Journal for initially publishing this article in 2000 (online) and the editors of Game & Puzzle Design for the chance to republish it here (in print).

References


Appendix A: Historical Context

The above piece was written in 1999, as I started looking for a less stressful way of making a living than teaching high school mathematics. I had decided to write a series of web pages (what would now be called a ‘blog’) on some of the various and obscure games in my collection, indexed by a main page, and felt that I should write an introductory text to explain my interest in games and general thoughts about them. So I typed one up, and edited it, and edited it again – obsessing over what I wanted to say – until finally uploading the result. The internet was a lot smaller in those days, and it turned out that a game hobbyist needed no particular authority to attract interest from other game hobbyists. I soon began to get thoughtful email responses from readers.

One such response was from the late Burt Hochberg, an accomplished Chess author and previous editor of GAMES magazine. He wanted to publish my still untitled article in a new online board games periodical he was organising called The Games Journal, so I decided to expand it and brush it up even further, adding the depth/clarity, drama/decisiveness section. In July 2000, I was proud to see my little essay begin a new and more prominent life in Burt’s online magazine.

Burt titled the article ‘Defining the Abstract’ as it began with a general explanation of the term abstract strategy game. I seemed to remember this term from a book on recreational mathematics that I had read ten or fifteen years previously, and did not invent it, as is often assumed. I did not intend to coin a neologism, but only to explain a ‘term of art’ that already existed in the game playing community. However, my paper seemed to accelerate the spread of this term throughout this community. The word abstract overemphasises the lack of theme associated with these games, and although I did remark on their usual lack of theme in my piece, this feature is admittedly inessential, concerning only presentation.

Because of the title and newness of the term abstract strategy game, some readers thought that I was setting myself up as the arbiter of this category, or that I was asserting that abstract games are superior as a class to other games. Neither was my intention, and certainly there are nonabstract games, such as Backgammon and Poker, that are better than trivial abstract games such as Tic-Tac-Toe or ‘thought experiment’ games such as Misere Hex (a misere game inverts the winning condition).

Games with overt chance elements or hidden information exercise the intellect in ways that are important and different to abstract games. Backgammon requires us to make decisions without knowing what will happen next; Poker requires us to make decisions without knowing what the other players know. But a good abstract game has its own beauty in that it conceals nothing. All of the information is available, but there can be so much that the player must find theories to make sense of it all. These just happened to be the games that I wanted to write about.

‘Defining the Abstract’ represents my first foray into online publishing, so I have been impressed and surprised by the attention – even notoriety – that the piece has attracted over the years. I value the contacts that it has led to, and the resulting discussions and projects which I have been delighted to take part in.

J. Mark Thompson creates and programs databases for a living.
Address: Bolingbrook, Illinois, USA.
Email: markthom1862@gmail.com
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editorial</td>
<td>Mixing and Fixing Games</td>
<td>C. Browne</td>
<td>3</td>
</tr>
<tr>
<td>Case Study</td>
<td>The Double Circle Real $5 \times 5 \times 5$</td>
<td>C. Hoff</td>
<td>5</td>
</tr>
<tr>
<td>Case Study</td>
<td>Tentai Show</td>
<td>J. Goto</td>
<td>15</td>
</tr>
<tr>
<td>Case Study</td>
<td>Try: A Hybrid Puzzle/Game</td>
<td>C. Browne</td>
<td>21</td>
</tr>
<tr>
<td>Article</td>
<td>Dameo: A New Step in the Evolution of Draughts?</td>
<td>C. Freeling</td>
<td>28</td>
</tr>
<tr>
<td>Case Study</td>
<td>Catch-Up: A Game in Which the Lead Alternates</td>
<td>A. Isaksen et al.</td>
<td>38</td>
</tr>
<tr>
<td>Short Note</td>
<td>Coalition Control Through Forced Betrayal</td>
<td>C. Browne</td>
<td>50</td>
</tr>
<tr>
<td>Short Note</td>
<td>From Mathematical Insight to Strategy</td>
<td>N. Romeral Andrés</td>
<td>53</td>
</tr>
<tr>
<td>New from Old</td>
<td>Deriving Card Games from Mathematical Games</td>
<td>D. Ashlock and J. Schonfeld</td>
<td>55</td>
</tr>
<tr>
<td>Article</td>
<td>Game Mutators for Restricting Play</td>
<td>J. Neto and W. Taylor</td>
<td>64</td>
</tr>
<tr>
<td>Game Design Patterns</td>
<td>Explore the Design Space</td>
<td>C. Browne</td>
<td>71</td>
</tr>
<tr>
<td>From the Archive</td>
<td>What Makes a Game Good?</td>
<td>W. Kramer</td>
<td>84</td>
</tr>
</tbody>
</table>

The front cover shows a cutaway rendering of Carl Hoff’s Double Circle Real $5 \times 5 \times 5$ twisty puzzle. The inner workings of the Double Circle Real $5 \times 5 \times 5$ are described on pages 5–14.

All content © 2015, *Game & Puzzle Design* and QUT.
Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com

Editor-in-Chief

CAMERON BROWNE
Department of Knowledge Engineering (DKE)
Maastricht University
Maastricht, 6200 MD
Netherlands

cameron.browne@maastrichtuniversity.nl

Liaison Officer

NÉSTOR ROMERAL ANDRÉS
Nestorgames
Spain

Copy Editor

RUSS WILLIAMS
Green Star
Poland

Editorial Panel

INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
CARL HOFF, Applied Materials, USA
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
DAVID PARLETT, Parlett Games, UK
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
SPYRIDON SAMOTHRAKIS, Univ. Essex, UK
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINDANS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent publication prepared and published with the kind help of volunteers. Set in Palatino using LaTeX.

Published content © 2018, Game & Puzzle Design.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.gapdjournal.com/submit
**Editorial**

**Mixing and Fixing Games**

Cameron Browne, Queensland University of Technology (QUT)

This issue is all about two things: mixing and fixing. Some articles describe games created by mixing ideas from existing designs, some describe games that came about – or were improved – by fixing particular design problems, and some combine both aspects.

This division between mixing and fixing may seem rather arbitrary, but it has a theoretical basis. It comes from a model of human creativity proposed by mathematician Robert J. Weber in 1992 [1], who observed that the primary mechanisms for invention include:

1. **Joining** existing features in new ways.
2. **Adding** new features to existing ones.
3. **Refining** features through fine-tuning.
4. **Transforming** the feature space through abstraction.

In the context of game and puzzle design, ‘mixing’ refers to mechanism 1, i.e. the joining of existing features or ideas in new ways, while ‘fixing’ refers to mechanism 3, i.e. the refinement of features (and the solution of problems) through fine-tuning.

This division of papers was quite unintentional. However, writing on the topic of creativity for this issue raised my awareness of such concerns, and the separation soon became abundantly clear. These two mechanisms appear especially relevant to the pursuit of game design.

**Fixing Games**

The lead article in this issue, ‘The Double Circle Real 5×5×5’ by Carl Hoff, was inspired by a limitation of existing 5×5×5 Rubik’s-type cube designs: that the internal cubies are hidden from the solver and do not come into play. The mechanism devised by Carl – which produced this issue’s eye-catching front cover image – is mind-bending enough to visualise even with the help of CAD software! So to produce a physical, working 5×5×5 twisty puzzle that incorporates all internal cubies as part of the visible exterior mechanism is a truly impressive feat.

Jimmy Goto then describes how the inventor of the logic puzzle ‘Tentai Show’ fixed a perceived problem in the aesthetics of its original design, by incorporating colour into each challenge. This did not have any effect on actually solving the challenges, but added a new dimension to the puzzle (Weber’s second type of creativity), to entertain solvers and give setters more scope for creative expression in their designs.

Christian Freeling’s article ‘Dameo: A New Step in the Evolution of Draughts?’ gives a brief history of the key steps in the development of the Draughts family, and highlights the problem of drawishness in competition-level play. He describes how his game Dameo emerged through a desire to address this problem, to produce a faster-playing and more decisive alternative.

The problem of imbalance in play is one that game designers must constantly grapple with. Isaksen et al. describe a mathematical game with an inherent balancing mechanism, that directly addresses this problem, in ‘Catch-Up: A Game in Which the Lead Alternates’. Analysis reveals their simple balancing mechanism to have some subtle and interesting strategic implications.

Similarly, my short note ‘Coalition Control Through Forced Betrayal’ examines the problem of imbalance in multiplayer games caused by non-strategic (i.e. social) coalitions that can arise between players. Rather than employing a forced revenge rule to address this problem, I demonstrate that reversing this approach to enforce betrayals between players can be more effective, using two simple hypothetical games.

Néstor Romeral Andrés turned to mathematics to solve a problem with his game Omega, as described in his short note ‘From Mathematical Insight to Strategy’. The problem was not with the game itself, but with its perceived complexity in the eyes of players, and a simple insight provided an intuitive strategy that made the game more tractable and enjoyable to play.

João Pedro Neto and William Taylor describe ‘Game Mutators for Restricting Play’. These are simple metarules, such as movement limitations based on group connectivity or delaying piece capture, that can be applied to flawed rule sets, in order to correct undesired behaviours in the resulting games.

**Mixing Games**

Weber’s model of creativity is explored more fully in my piece ‘Explore the Design Space’, which demonstrates how new games can be created by mixing ideas from existing games, using a family Browne, C., ‘Mixing and Fixing Games’, Game & Puzzle Design, vol. 1, no. 2, 2015, pp. 3–4. © 2015
of path-based tile games as an example. I suggest ways to focus the search, in order to hopefully find good combinations more quickly, and demonstrate how modifying subsets of existing games through simple transformations (Weber’s fourth type of creativity) can find new and fruitful regions in the design space.

The second piece in our regular New from Old column, ‘Deriving Card Games from Mathematical Games’, is all about mixing existing ideas in new ways. Daniel Ashlock and Justin Schonfeld demonstrate how mathematical principles from graph theory can be applied to known – and unknown – mathematical games, in order to produce novel deck-based card games of distinct character.

My own article ‘Try: A Hybrid Puzzle/Game’ describes a blatant example of creating a new game from existing ideas. One of these ideas comes from a solitaire puzzle and the other from a strategy board game, but both merge seamlessly to transform a flawed triangular Sudoku into a more interesting design.

Try is this issue’s ‘feature puzzle’, and you will find sample challenges printed throughout the issue, in approximate order of difficulty. Solutions can be found on the journal’s website: http://www.gapdjournal.com/issues/

This issue concludes with a reprint of Wolfgang Kramer’s classic 2000 piece ‘What Makes a Game Good?’ This is a marvellously concise summary of the types of characteristics that designers should be aware of when developing and fine-tuning their games. While there will always be exceptions to such guidelines, and personal preference will come into play, these are in my opinion still among the best three pages that any game designer can read.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

Try Challenges #1 and #2

Fill the grid with numbers 1 to 5, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.
The Double Circle Real $5 \times 5 \times 5$

Carl Hoff, Applied Materials

Twisty puzzles, such as the Rubik’s Cube, are typically thought of as geometric solids cut with a set of planes that allow a subset of the puzzle to be rotated. Stickers are applied to their surface such that a solved state is arrived at when each face is restored to a solid colour. This paper considers the 125 individual cubies that make up a $5 \times 5 \times 5$ twisty puzzle, and presents a method for including all 125 cubies in the solution, not just the 98 that appear on the surface.

1 Introduction

In 1974, Ernő Rubik invented the $3 \times 3 \times 3$ twisting puzzle, hereafter referred to as the ‘$3 \times 3 \times 3$’. Produced and marketed by Ideal Toys in 1980 as the Rubik’s Cube, it launched twisty puzzles into the mainstream. In 1981, Péter Sebestény applied for a patent for the $4 \times 4 \times 4$, shown in Figure 1.

![Figure 1. Péter Sebestény with his $4 \times 4 \times 4$ cube. Image from speedcubing.com, with permission.](image)

Later that same year, Udo Krell applied for a patent for the $5 \times 5 \times 5$, and this trend continues today. Moyu released the $13 \times 13 \times 13$ in 2014 and Oskar van Deventer’s $17 \times 17 \times 17$ has been available from the online 3D printing service Shapeways since 2011.

As we progress to ever higher $N \times N \times N$ cubes, more and more of the implied puzzle is being left hidden and unsolvable. With the $3 \times 3 \times 3$, only 26 of the 27 component cubies are stickered, i.e. visible to the player. Of these 26, only 20 have a unique position and orientation in the solved state, as the orientation of the face centres is irrelevant to the solution. This problem gets worse for the $4 \times 4 \times 4$, which has 8 of its 64 cubies unstickered, and even worse for the $5 \times 5 \times 5$, which has 27 of its 125 cubies unstickered.

1.1 Rubik-Type Cube Design

To understand why these internal cubies were left out, we need to examine the design rules in use at the time. Péter Sebestény listed the following design rules for a Rubik-Type Cube (RTC):

1. The RTC is a regular three-dimensional cube, with edges of equal length.
2. The RTC appears to be made of smaller regular unit cubes, the same number of unit cubes along the edges.
3. The edges of the unit cubes on the surface of the RTC appear to be of the same length, irrespectively of their unseen shape in the interior.
4. The unit cubes that are of the same distance from a freely chosen face of the RTC form a layer. Each layer that is parallel to this chosen face can rotate independently around the axis that is perpendicular to this face and goes through the centre of the RTC.
5. No part of the internal structure of the construction can be seen from outside, no matter how the layers are moved.
6. The RTC consists of only mechanical components; there are no magnetic or electric parts. It may contain springs.
7. The RTC is time-independent and retains its form. It needs no battery that can lose its charge, and can be used at any time. Ageing, wearing off, melting from heat and breakage due to forceful use do not contradict this requirement.

---

1http://www.shapeways.com
2Expanded version: https://drive.google.com/file/d/0B8WmsMACU4YubGxuOTJ2RkQ4bkE

These design rules not only left unreachable cubies hidden inside the puzzle, but they also capped the development of higher $N \times N \times N$ cubes as well. After the $5 \times 5 \times 5$, it was not until 2004 that Panagiotis Verdes presented the $6 \times 6 \times 6$ and $7 \times 7 \times 7$ at the International Puzzle Party in Helsinki. He also presented a patent that covered all levels up to and including the $11 \times 11 \times 11$.

To achieve this, Verdes had to break a design rule that had been considered fundamental up to that point. He could not keep all the unit cubes, also known as cubies, all the same apparent size. For $7 \times 7 \times 7$ and above, the outer layers need to be wider than the others, to allow the corner cubies room to have an attachment point to a foot that holds them in the mechanism. Even for the $6 \times 6 \times 6$ it is difficult to make the visible cubies the same size, as these attachments must be very thin, although there do exist subsequent $6 \times 6 \times 6$ designs with cubies of the same apparent size.

1.2 Using the Hidden Cubies

In order to bring out the interior cubies, more design rules had to be broken. The first puzzle to sticker all 27 cubies in a $3 \times 3 \times 3$ was Aleh’s $3 \times 3 \times 3$ Circle Cube [3] from 2008 (Figure 2).

In this design, we now start seeing some of the internal structure as the planar cuts now no longer extend all the way through the puzzle but come out of the face they are parallel to as circles. You can view each of these circles as a window that allows a side view of the slice layer below. The centre square is a face of the previously hidden central cubie, also called the $1 \times 1 \times 1$ or core.

Immediately adjacent to that square yet still inside the circle, we have side views of the neighbouring face centres. Note that since we can now see the sides of the face centres, these cubies now have a fixed orientation in the solved state.

The remaining wedge-shaped pieces inside the circles are side views of the $3 \times 3 \times 3$ edges directly behind them. Since the edges already have unique positions and orientations, these pieces do not add any new information. Since each solvable piece in this puzzle has a unique position and orientation, it is called a Super cube. The fact that all the interior pieces are exposed makes it a Multi-cube, so this is a Super Multi-$3 \times 3 \times 3$.

Daqing Bao had earlier unveiled his $(4 \times 4 \times 4)+(2 \times 2 \times 2)$ Cube in 2007 [4]. Bao’s design also made use of circles, which could be seen as windows into the interior pieces. These revealed the $2 \times 2 \times 2$ that had until then been hidden inside the $4 \times 4 \times 4$.

Figure 2. Aleh’s $3 \times 3 \times 3$ Circle Cube [3].

Figure 3. Mapping the pieces of the Crazy $4 \times 4 \times 4$ to a Multi-$4 \times 4 \times 4$, as the circle radius increases.
1.3 Super Solved Cubes

In April of 2004, Per Kristen Fredlund released a program called CubixPlayer2 [6] that allowed exploded views of $N \times N \times N$ cubies to be solved as Super Multi-Cubes. CubixPlayer2 allowed you to select an $N \times N \times N$ from a $2 \times 2 \times 2$ up to a $7 \times 7 \times 7$. One could explode the cubies to see their side colours, this gave each face cubie a specific position and orientation in the solved state. One could also enable a level of transparency to allow the interior cubies to be seen and solved. Figure 4 shows such a view of the $5 \times 5 \times 5$.

The program referred to the state where all exterior surfaces were restored to a solid colour as solved. If all exterior cubies were in their unique position and orientation, as revealed in the exploded view, this state was called super solved.

If all of the interior cubies were super solved as well, through the use of the transparent feature, then the state was called super super solved.

It was not until September 2009 that the second use of ‘super’ was dropped in favour of the term ‘multi-’ to refer to the interior cubies [7].

In November 2006, Fisher revisited the topic of hidden cubies on the TwistyPuzzles.com forum [4]. He discussed the notion of a $3 \times 3 \times 3$ inside a $5 \times 5 \times 5$, and stated: ‘It is fairly obvious that the only way to play with such a puzzle is on a computer’.

What better way to motivate a designer than to give him or her the opportunity to do the impossible? This is where the quest for a physical Super Multi-$5 \times 5 \times 5$ began. I called this design the ‘Real $5 \times 5 \times 5$’ as it would be the first $5 \times 5 \times 5$ to live up to its name and use all 125 cubies.

2 Designing the Real $5 \times 5 \times 5$

The initial focus was simply to find a way to incorporate the $3 \times 3 \times 3$ components into a $5 \times 5 \times 5$ design, and the first inspiration came from a Pascal Obispo music video [9]. The set of this video is intended to be the inside of a giant $3 \times 3 \times 3$, but it is apparent that there are more than three turnable layers on each axis. Therefore, it is more accurately viewed as the interior of a $5 \times 5 \times 5$, with a void where the $3 \times 3 \times 3$ should be.

My first attempt at a Real $5 \times 5 \times 5$ design [10] recreated the $5 \times 5 \times 5$ seen in the Pascal Obispo music video. The $3 \times 3 \times 3$ would be 27 perfect cubes contained inside the interior void of the $5 \times 5 \times 5$. To see the interior puzzle would require windows or transparent parts, but the design was more a proof of concept than something thought of as practical. So this design, seen in Figure 5, was never actually manufactured.
Matthew Sheerin then worked out how circular windows, similar to those seen in Aleh’s 3×3×3 Circle Cube and the Crazy 4×4×4 Cubes, could be used to bring the interior cubies to the surface of a 5×5×5. This required two circles on each face, but did not tackle the interior mechanism. Sheerin’s choice for placing the circular windows is shown in Figure 6. This figure is a stereogram; cross your eyes to see the 3D effect.

2.1 3D Printing

I worked out the details for the mechanism from 5 to 23 October 2010. The design was presented on 25 November 2010, after several animations and images had been rendered using the public domain ray tracing software POV-Ray.

At the time, the only modelling software that I was familiar with was POV-Ray. This software used composite solid geometry (CSG) and did not support files in STL format, which has since become a standard format for 3D model printing. This was not a concern back in 2010, as online modelling services such as Shapeways had not yet established themselves, and the 3D printing of puzzles had not become commonplace.

However, once 3D printed designs became more common, I soon realised that new modelling software should be explored. SolidWorks was the software of choice, as it could export files in the widely used STL format, and allowed for edges to be easily filleted on complex shapes. So in November of 2013, it was decided to re-design the puzzle from scratch using SolidWorks with the intent of getting the model 3D printed. Instead of waiting until the design had been finished, I decided to share the entire process with the puzzle community in a thread on the Twisty-Puzzles.com website.

By this time, I had designed several other puzzles and become familiar with 3D printing so many changes and tweaks to the design were made while porting the model from POV-Ray to SolidWorks. Figure 7 is an image of the completed POV-Ray model as it stood back in 2010.

---

3 http://www.povray.org
4 http://www.solidworks.com
Figure 8. Models of the Real 5×5×5 core: POV-Ray version (left) and SolidWorks version (right).

Note the void left by the removed piece near the 5×5×5 corners. These were part of the 5×5×5 wings in the layer behind the corner. They contained no new information, so were left out to simplify the mechanism.

3 The Parts of a Real 5×5×5

The core or 1×1×1 of the Double Circle Real 5×5×5 first designed in POV-Ray involved just seven parts, as shown in Figure 8. The arms off the core would need to be removable to add a part of the 3×3×3 which would later rotate on them, but otherwise the POV-Ray model was never intended to be printed so had no place for screws.

3.1 Core and Internal 3×3×3

These seven parts ballooned out to 19 when ported to SolidWorks, as shown in Figure 8 (right). The central core was turned into a ball and each arm was cut into three parts. The inner 3×3×3 arm has a square foot which locks into a square hole in the ball core, to keep it from rotating. The 5×5×5 arm has tabs which lock into the 3×3×3 arm.

The entire assembly is held together with twelve screws and the last part is the cap used to cover the screw heads that is glued to the 5×5×5 arm. With nineteen printed parts, twelve screws, and six stickers, this is the most complex 1×1×1 to my knowledge.

The 3×3×3 face centre, shown in Figure 9, did not change much from the POV-Ray model to the SolidWorks model, apart from being hollowed, the addition of fillets, and the addition of a groove for a foot from the 5×5×5 face centre.

A wing is an edge cubie which is not the centre of the edge.

Figure 9. A 3×3×3 face centre (highlighted).

Figure 10. A 3×3×3 edge (highlighted).
The $3\times3\times3$ edges, shown in Figure [10] had what appear to be PacMan-shaped cutouts added to their surface (far right). These were needed to distinguish the four edges in each central slice layer, as without these recesses, only the opposite faces would be stickered and the four edges in the same central slice would appear identical.

Each of the $3\times3\times3$ corners, one of which is shown in Figure [4] is a single part. These were the hardest parts to design, however, as they each required thin necks to the three small disconnected areas exposed on the puzzle’s surface.

Another design decision affecting the shape of the $3\times3\times3$ corners was the decision to remove the void left on the surface of the POV-Ray model by the removal of a redundant portion of the $5\times5\times5$ wing. This moved the outer circle in such that it was touching the outer tip of the exposed surface of the $3\times3\times3$ corners.

### 3.2 External $5\times5\times5$

The $5\times5\times5$ face centres, like the $3\times3\times3$ face centres, are not exposed on the face they represent. Instead, the four sides of the face centre cube that are adjacent to the face being represented are exposed, as shown in Figure [12].

The $5\times5\times5$ T-centres, shown in Figure [13] and X-centres, shown in Figure [14] are the cubies that are normally present in the centre of the faces they represent. For the Double Circle Real $5\times5\times5$, they are now composed of disconnected parts on a subset of the adjacent faces.

---

6 A T-centre is a face cube in a horizontal or vertical line with the face’s centre.

7 An X-centre is a face cube on one of the two diagonals of the face.
3.3 $5\times5\times5$ Corners

The $5\times5\times5$ corner, shown in Figure 17, is the tenth and final piece type. It too is a single part, and could not be designed with a mechanism similar to the $3\times3\times3$ corner, as the $3\times3\times3$ must move independently under the $5\times5\times5$ corner.

Each $5\times5\times5$ corner, therefore, has a small foot that moves in a track, which is cut into the $3\times3\times3$ corners and the hidden part of the $3\times3\times3$ edges. Figure 18 shows the $3\times3\times3$ found inside the Real $5\times5\times5$. Note the similarity of this track mechanism to that used in the commercial $2\times2\times2$ Dreamball puzzle, shown in Figure 19.
4 The Puzzle

The Double Circle Real $5 \times 5 \times 5$, a physical version of which is shown in Figure 20, joins the ranks of the many twisty puzzles designed and built by hobbyists [14]. Figure 21 shows the puzzle in a scrambled state.

There are many resources for the study of twisty puzzle design [15] and the various mechanisms available [16]. For example, many designers, including myself, regularly frequent the forums at TwistyPuzzles.com [17], which provides a wealth of knowledge and useful terminology base for those interested in this hobby.

4.1 Complexity

Twisty puzzles in general open up a world of very complex and mathematically intriguing designs. Many researchers have investigated algorithms for solving these puzzles [18], and it has been proven that ‘God’s Number’ for the Rubik’s Cube, i.e. the maximum number of moves required in any optimal solution, is 20 [19].

The close connection between these puzzles and group theory leads to a number of related mathematical topics, which are beyond the scope of this paper. Still, some basic comparisons show the relationship between the $3 \times 3 \times 3$, $5 \times 5 \times 5$, and Real $5 \times 5 \times 5$. The numbers of state permutations for these puzzles are given in Table 1.

From here we can define the Multi-$5 \times 5 \times 5$ as a standard $3 \times 3 \times 3$ inside a standard $5 \times 5 \times 5$. Note that the Multi-$5 \times 5 \times 5$ does not give each piece a unique position and orientation in the solved state.

The Real $5 \times 5 \times 5$ is a Super Multi-$5 \times 5 \times 5$, as it has a Super $3 \times 3 \times 3$ inside a Super $5 \times 5 \times 5$. Counting the permutations for these puzzles gives the results for the Multi-$5 \times 5 \times 5$ and Real $5 \times 5 \times 5$, as shown in the bottom half of Table 1. The factor of 1/2 is the result of parity restrictions between the inner $3 \times 3 \times 3$ and the external $5 \times 5 \times 5$. Using commutators, it is easy to find pure sequences for solving the $5 \times 5 \times 5$ corners and middle edges.

---

*8 A commutator in this context is a sequence of operations for transforming one given state to another.*
Using the concept of what David Singmaster calls evisceration, i.e. replacing every face layer move by a move of its adjacent inner slice and vice versa [5], provides move sequences for solving the inner $3 \times 3 \times 3$. This shows that apart from parity, there are no restrictions on the inner $3 \times 3 \times 3$.

### Table 1. Number of possible state permutations for each puzzle type.

<table>
<thead>
<tr>
<th>Puzzle</th>
<th>Calculation</th>
<th>Permutations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3 \times 3 \times 3$</td>
<td>$(8! \times 12!/2) \times (3^6/3) \times (2^{12}/2)$</td>
<td>$\approx 4.3 \times 10^{19}$</td>
</tr>
<tr>
<td>Super $3 \times 3 \times 3$</td>
<td>$(8! \times 12!/2) \times (3^6/3) \times (2^{12}/2) \times (4^6/2)$</td>
<td>$\approx 8.8 \times 10^{22}$</td>
</tr>
<tr>
<td>$5 \times 5 \times 5$</td>
<td>$(8! \times 12!/2) \times (3^6/3) \times (2^{12}/2) \times (24!/4!^6) \times (24!)$</td>
<td>$\approx 2.8 \times 10^{10}$</td>
</tr>
<tr>
<td>Super $5 \times 5 \times 5$</td>
<td>$(8! \times 12!/2) \times (3^6/3) \times (2^{12}/2) \times (24!/2) \times (24!/2) \times (4^6/2)$</td>
<td>$\approx 5.2 \times 10^{93}$</td>
</tr>
<tr>
<td>Multi-$5 \times 5 \times 5$</td>
<td>$((3\times3\times3 ,</td>
<td>, 5\times5\times5))/2$</td>
</tr>
<tr>
<td>Real $5 \times 5 \times 5$</td>
<td>$((8! \times 24!) \times (12!/2) \times (3^8/3) \times (2^{12}/2) \times (24!/4!^6)^2 \times (8!/2)$</td>
<td>$\approx 2.3 \times 10^{116}$</td>
</tr>
</tbody>
</table>

#### 5 Conclusion

Counter to the comments made by Tony Fisher back in 2006, you can play with higher order Super Multi-$N \times N \times N$ twisty puzzles that exist in physical form, without the need for software simulations such as CubixPlayer2. In fact, Oskar van Deventer has now designed a Super Multi-$6 \times 6 \times 6$ using a totally different approach[10] to that described here.

I intend to continue developing my design, in order to produce a Double Circle Real $6 \times 6 \times 6$. The inner cubies which were ignored for so long are now finally being considered on equal footing with the cubies that are fortunate enough to be on the surface.

The explosive growth in puzzle innovation, that Ernő Rubik set in motion with the $3 \times 3 \times 3$, continues to this day and there is no end in sight. The advent of affordable 3D printing has encouraged many new designers to enter this rewarding area of exploration.

### Acknowledgements

Thanks to Tony Fisher for the motivation to accomplish the impossible, Matthew Sheerin for pointing out the double circle cut pattern and encouraging me to pursue its physical creation, and Ernő Rubik for setting this fascinating field of study in motion.

Thanks also to the forum members at TwistyPuzzles.com for their encouragement and support. I would not be the designer that I am today without the open sharing and assistance that they have provided. Thanks to Brandon Enright and Jaap Scherphuis for the verification of the permutation calculations presented in this paper. The editors would like to thank Tom Yuval for his helpful comments on the manuscript.

### References


---

9http://www.jaapsch.net/puzzles/cubic5.htm#p16
10http://shpws.me/CGQm
Try Challenges #3 and #4

Fill the grid with numbers 1 to 5, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.
Tentai Show

Jimmy Goto, Nikoli

Tentai Show is a logic puzzle in which the solver must deduce the shapes of rotationally symmetric regions within a grid, then colour certain regions to reveal a hidden picture; this is puzzle as art.

1 Introduction

Tentai Show is a pure deduction puzzle, invented for Nikoli by Japanese designer Gesaku, which first appeared in the September 2001 issue of Puzzle Tsushin Nikoli 96. It is unusual in that solving each challenge reveals a picture hidden in the hints, combining logic with aesthetics to blur the line between puzzles and art.

2 Rules

A Tentai Show challenge consists of a grid of squares, with circles and dots marked in some cells and on some grid lines. Figure 1 (left) shows a typical challenge.

The aim is to divide the grid into regions by drawing bold lines along certain grid lines, such that every region:

1. contains exactly one circle or dot, and
2. has 180° rotational symmetry.

The regions containing black dots are then coloured in to reveal a picture hidden within the challenge. For example, the challenge shown in Figure 1 has been solved to reveal an image of the numeral ‘4’ (right). Each challenge should have a single unique solution [1].

3 Example

Figures 2 and 3 show the typical working involved in solving a larger example, to demonstrate relevant tactics. Starting with the initial challenge in Figure 2(a), it is easy to deduce some bold lines and regions based on the two rules (b), and then build on this information to deduce further bold lines and regions (c).

Figure 1. A challenge (a) and its solution (b).

Figure 2. A challenge (a), some obvious lines to start with (b) and some deductions built on them (c).

Figure 3. Linked cells (d) allow further deductions to complete the solution (e), which is an umbrella (f).

The arrows in Figure 3(d) indicate cells that the corresponding stars must own, as no other stars can possibly reach them. This defines the regions for these two stars, and allows the solver to complete the solution (e). Colouring in the black star regions reveals the hidden picture to be an umbrella (f).

4 Design

An early version of the puzzle only contained white stars, and so did not contain hidden pictures, as shown in Figure 4(a) and (b).

Players still liked this version, which is essentially equivalent to the final puzzle, as the differentiated star colours have almost no bearing on a challenge’s solution. However, the designer Gesaku felt that a vital element was missing, and had the brilliant idea to make some stars black, so that a picture could be encoded in each challenge. The hidden picture gives the player something extra to work towards, a bonus reward which can be aesthetically pleasing and sometimes amusing. It also gives the designer another avenue to express their personality and creativity in their designs. Instead of painting by numbers, this is painting by shapes, with a binary palette.

The colouring of cells is reminiscent of the tetromino-packing puzzle LITS, also published by Nikoli [2], but LITS does not use rotational symmetry as a constraint or reveal hidden pictures. The related class of Nonogram puzzles do reveal hidden pictures [3], but these do not use rotational symmetry either, and the picture is revealed pixel-by-pixel rather than region-by-region. Tentai Show has a character all its own.

I say above that the differentiated star colours have almost no bearing on the solution of a given Tentai Show challenge. However, they do dictate the placement of the initial hints, as the derived regions must follow the shape of the hidden picture; the hints define the picture, but the picture also constrains the hints.

Players could possibly extrapolate a partially completed picture to help solve the remaining parts of the grid. However, this should not occur very often, as the hidden pictures are not usually symmetrical themselves, and it can be difficult to judge exactly which remaining pixels must be on or off to complete them. In fact, even knowing what the final picture will be may not help much in solving the actual region shapes, unless the solver can differentiate each individual pixel.

4.1 More Than Two Colours

The puzzle’s designer, Gesaku, points out that Tentai Show is not limited to just two colours. For example, Figure 5 shows a challenge with five colours and Figure 6 shows its solution.

Additional colours allow more attractive pictures, but can make challenges easier to solve by providing additional visual clues, as same-coloured regions tend to form smaller, contiguous clusters. Nikoli prefers to print two-colour versions of the puzzle, due to both the additional cost of colour printing, and because the quality of our published challenges is most important to us, so we prefer to print their most challenging (two-colour) forms.

---

4.2 Challenge Design

Gesaku typically designs Tentai Show challenges using the following process:

1. Define the pixelated target image.
2. Place obvious smaller regions in constrained parts of the image.
3. Place larger, more complex shapes in open areas.
4. Fill in remaining gaps with smaller, simpler shapes.

Each region placement must of course be tested for correctness as it is made, so that the resulting challenge will produce a single unique solution. Figure 5 demonstrates how this process might be applied, using the challenge shown in Figures 2 and 3 as an example. Note that each region’s star is added as its region is placed.

Starting with the stylised umbrella image (a), tightly constrained parts of the picture around the handle are defined with small, simple regions (b). Larger, more complex regions are then defined in open areas, in both the foreground and background (c), and the remaining gaps filled with smaller, simpler shapes to give the final solution (d). The regions are then removed, leaving their stars, to give the actual challenge (Figure 2, left).

Note that this example is a gross simplification applied to a trivial case. Designing full challenges, such as the one shown in Appendix A, is something of an art form.

5 Derivation of the Name

The name Tentai Show has a double meaning that refers to both its inherent rotational symmetry and its form as a collection of stars.

‘Tentai Show’ is actually tentaisyo when anglicised from its Japanese form (天体ショー), where taïsyo refers to ‘symmetry’. In Japanese, sen-taisyo means line symmetry, men-taisyo means plane symmetry and ten-taisyo means point symmetry. Ten-taisyo is most relevant here, as Tentai Show involves 180° rotational symmetry around a point (i.e. the star) within each region.

---

\[\text{Figure 7. Possible steps in creating a challenge for the umbrella example.}\]
Moving the break between the words from *ten taisyo* to *tentai syo* creates another Japanese word *tentai*, which means ‘heavenly object’ such as a star. Anglicising the second word from *syo* to ‘show’ gives Tentai Show, which makes the puzzle sound like an astronomical show. The puzzle is also sometimes called ‘Galaxies’ or ‘Spiral Galaxies’ in English, which reinforces its rotational astronomical theme.

6 Conclusion

Tentai Show is first and foremost a logic puzzle, but each challenge encodes a hidden image, which blurs the line between puzzles and art. The development of Tentai Show demonstrates how designers can add a little something extra to an already good design, in order to entertain players and produce an even better result.

There are now two printed Tentai Show collections from Nikoli \[4, 5\]. An apt example from these collections is given in the Appendix.

Acknowledgements

Thanks to the Editor-in-Chief for assembling this piece from the notes provided and redoing the figures. Thanks also to the reviewers for their helpful suggestions.

References


*Jimmy Goto* works for Japanese quarterly magazine *Puzzle Communication Nikoli*, which specialises in language-independent logic puzzles.

**Address:** Nikoli Co., Ltd., 4-17-10 Kuramae Taitou-ku, Tokyo 111-0051.

**Email:** goto@nikoli.co.jp

Appendix A: 20 × 36 Example

This appendix includes one of my favourite Tentai Show challenges. Figure 8 shows the 20 × 36 example ‘Puzzle No. 93’ from [5] by Tsugesan.

I believe this to be a masterpiece of Tentai Show design. Not only does it encode a surprising and interesting picture that is most relevant to the theme of the puzzle (revealed in Figure 9 on the subsequent page), but its solution involves some complex and unusual regions that require a certain degree of skill to deduce.

I encourage you to try this challenge yourself before turning to see the solution. But even if you do glance at the solution first, then deducing the region shapes can still be an interesting process, which is one of the attractive features of the Tentai Show puzzle.
Figure 8. A $20 \times 36$ Tentai Show challenge by Tsugesan.
Figure 9. Solution showing astronaut image (inset).
Try: A Hybrid Puzzle/Game

Cameron Browne, Queensland University of Technology (QUT)

Try is a new logic puzzle that juxtaposes a strategy game rule onto a solitaire puzzle. This article describes Try, some basic strategies, and the design process behind it, most importantly the creative leap that produced a novel puzzle from familiar elements.

1 Introduction

Try is a pure deduction puzzle in the same family as Sudoku \[1\], but with an additional constraint borrowed from a strategy game. The rules are as follows:

Try is played on a triangular tessellation of \(N\) hexagons per side, with some hint cells initially assigned numbers from 1..\(N\). The aim is to assign every cell a number 1..\(N\) such that:

1. No number occurs more than once along any line (Sudoku rule).
2. No connected group of odd numbers touches all three sides (Y rule).

1.1 The Game of Y

The Y rule is borrowed from the game of Y, invented by Claude Shannon in the 1950s\[1\], which is one of the earliest and most fundamental connection games \[6\]. The aim in Y is to complete a chain of your pieces connecting all three board sides, called a Y, as shown in Figure 1. Corners count for both incident sides.

One of the attractive features of Y is that a winning chain of one colour that touches all three sides precludes any possible winning chain for the other; exactly one player must win each game. This means that the Y rule in Try can be rephrased as: all three sides must be connected with a connected group of even numbers.

1.2 Triangular Sudoku + Y = Try

Figure 2 shows a typical Try challenge for size \(N=5\) (left) and its solution (right). The hint set for each challenge must be carefully chosen to give a single unique solution \[4\].

By comparison, the solution shown in Figure 3 is illegal because the odd-valued cells (circled) form a group that connects all three sides, even though the Sudoku constraint has been satisfied and no number occurs more than once along any line.

The practice of circling cells that are guaranteed to be odd helps clarify the Y aspect of a solution in progress, and will be adopted throughout this paper. A simple mnemonic is: ‘O’ is for ‘Odd’.

Figure 1. A game of Y won by White.

Figure 2. A Try challenge and its solution.

Figure 3. An illegal solution for this challenge.

---

Footnote:

1The Y board was later redesigned in the 1970s with a non-regular tiling to bring the corner cells more into play \[2\], but Shannon’s original design played on a regular grid of hexagons is more relevant here.

2 Strategies

The following section describes some basic strategies for solving Try challenges, based on the two core Sudoku and Y rules.

2.1 Sudoku-Based Strategies

Sudoku-based strategies tend to be simple exercises in bookkeeping.

2.1.1 Triangulation

For example, Figure 4(left) shows a triangulation method for eliminating candidate values, in an \( N=7 \) game. This method is as follows: if two cells in line are both reduced to two candidate values, then those two values can be eliminated from any other cell in line with each of these cells. For example, the challenge shown in Figure 4(left) has two source cells that must contain 4 or 6, as all other values are eliminated by the hint cells in line with them. Candidate values 4 and 6 can therefore be eliminated from the three dotted cells, as these are in line with each of the source cells.

Figure 4(middle) shows the candidate values along this triangle, and Figure 4(right) shows the result after candidate values 4 and 6 have been eliminated from the three triangulated cells; one of the cells can already be resolved to a 3.

A similar elimination can be applied along a straight line if three cells along it contain the same three candidate values, four cells contain the same four candidate values, etc. Of course, a single cell containing a single (known) value eliminates that value from all other cells in line with it. This is classic Sudoku deduction on a hexagonal grid, as is resolving values at cells for which all candidates but one have been eliminated.

2.1.2 Reverse Triangulation

The triangulation process can be reversed to allow further eliminations. For example, Figure 5(left) shows three cells in line that contain the values \( \{1, 2, 3\} \), and a \( \{1, 2\} \) cell that is known to be either 1 or 2.

If the two top \( \{1, 2, 3\} \) cells take the values 1 and 2, then the adjacent \( \{1, 2\} \) cell would have no possible values, hence at least one of the two top \( \{1, 2, 3\} \) cells must contain a 3 (Figure 5(right, underlined). This allows the 3 to be eliminated from the bottom \( \{1, 2, 3\} \) cell.

2.1.3 Edge Completion

Another Sudoku-based strategy exploits the fact that every board edge must contain every number from 1 to \( N \), which facilitates elimination along the three outer board edges. Lines through interior cells do not contain all numbers from 1 to \( N \), hence offer less deductive information in this respect. That is, lines of length \( N-1 \) contain one fewer numbers, lines of length \( N-2 \) contain two fewer numbers, and so on.

2.2 Y-Based Strategies

While Sudoku-based strategies are pattern-oriented and tend to be somewhat formulaic, Y-based strategies involve the more fluid concept of connection and take many forms. Y-based strategies add depth to the game and can require significant analysis, especially for larger board sizes.
2.2.1 Odd Connectivity

For example, Figure 6 (left) shows the challenge from Figure 3, with odd cells circled (right). The odd group consisting of 1 and 3 is connected to the left side, and is already virtually connected to the bottom side, as the two possible even values (2 and 4) can not block all three connecting cells marked + (right). The cell marked ? must therefore be even, in order to stop this odd group connecting all three sides. It must take the value 4, since it is already in line with a 2.

Figure 6. The cell indicated ? must be even.

Figure 7 (left) shows this cell resolved to a 4, which in turn identifies another odd cell above it. This new odd cell touches the left side is virtually connected to the right side, as the only even candidate that can now be placed along that side (i.e. 2) can not block both connecting cells marked +. The cell marked ? must therefore be even, to block an odd Y, and must take the value 2 since it is already in line with a 4, and the top corner must now be odd. Figure 7 (right) shows this progress.

Figure 7. The cell indicated ? must also be even.

The two empty odd cells on the left side must contain the values 3 and 5 (Figure 8, left), hence the bottom left corner can be resolved to a 4 due to the Sudoku rule, as this is now the only unclaimed number along the left side (Figure 8, right). The cell marked ? must also be even, to block an odd Y, and must take the value 2. Similarly, the cell to its immediate left is the only shaded cell that connects all three sides. It must take the value 4, since it is already in line with a 2.

Figure 8. The cell indicated ? must also be even.

The remaining bottom row values can then be resolved using classic Sudoku reasoning, as shown in Figure 9 (left). This provides enough information to similarly resolve the remaining values to give the solution (right).

Figure 9. The remaining values fall out.

2.2.2 Even Connectivity

Recall from Section 1.1 that in order to block a connected group of odd numbers from touching all three sides, then a connected group of even numbers must touch all three sides, due to the complementary nature of Y. This can be exploited to give useful information for solving challenges.

For example, consider the partially solved challenge shown in Figure 10 (left). A solid wall of odd-valued cells cuts off the 2 in the top corner, so the only possible way for a group of even numbers to connect all three sides is through the shaded region shown on the right.

Figure 10. Even’s only path is shaded.

The bottom right corner is the only shaded cell that touches the right side, so this cell must be even, and since there is a 2 in line with it, then it must take the value 4. Similarly, the cell to its immediate left is the only shaded cell that connects all three sides. It must take the value 4, since it is already in line with a 2.

Figure 10. Even’s only path is shaded.
to this corner, so it must also be even and must take the value 2. The rest of the solution falls out easily from here.

2.2.3 Odd Cell Identification

Consider the position shown in Figure 11 (left). Since the pair of numbers include all even numbers for size $N=5$, then all empty cells in line with these two cells can be marked as ‘odd’ (right).

![Figure 11](image1.png)

**Figure 11.** Cells in line must be odd.

This strategy involves Sudoku-style elimination, as already described in previous examples, such as Figure 7. However, it also has a strong benefit for Y-based methods, as these rely on identifying odd-only cells. It is worth highlighting as a strategy in its own right, as it is useful for quickly identifying odd cells without having to worry about their specific candidate values.

Solutions involving Y-based strategies are generally more interesting than those requiring Sudoku-based strategies alone. The Sudoku constraint provides a framework within which the puzzle functions, but the Y constraint is its real heart and soul.

3 Design

Try evolved gradually, over several weeks, through the design process outlined below.

3.1 Pascal’s Triangle

The initial impetus behind Try was to create a new and interesting deduction puzzle based on Pascal’s Triangle [5], in which adjacent pairs of numbers in a triangular grid are added to give the number below, as shown in Figure 12.

![Figure 12](image2.png)

**Figure 12.** Pascal’s Triangle.

However, this soon proved to be a naive ambition. Pascal’s Triangle has been studied for centuries and many puzzles derived from it, such as the Pascal’s Triangle puzzle [5] shown in Figure 13, in which the aim is to complete the triangle, respecting Pascal’s addition rule, such that $y = x + z$. Such puzzles are regularly set as programming exercises for computing students.

It is also difficult to make such puzzles interesting, and I had a growing suspicion that perhaps not all puzzle solvers would find adding lots of numbers ‘fun’. The real rewards in such puzzles lie in the deductive process, not in simple number crunching [13].

3.2 Sudoku Constraint

The Sudoku rule was therefore added to impose a deductive element on the puzzle. However, the standard Sudoku structure does not map nicely to a hexagonal basis.

Ideally, a Sudoku grid should subdivide into equal-sized subgrids able to contain all $N$ numbers, e.g. $3 \times 3$ subgrids in the square $N=9$ case. However, triangular tessellations of hexagons do not subdivide as neatly, as shown in Figure 14.

![Figure 14](image3.png)

**Figure 14.** Triangle subdivision is unsatisfactory.

---

3Programming exercise from: [http://rosettacode.org/wiki/Pascal%27s_triangle/Puzzle](http://rosettacode.org/wiki/Pascal%27s_triangle/Puzzle)
Subdividing this \( N=5 \) triangle produces four equilateral \( N=3 \) subtriangles, which would each require six numbers to fill, one more than is available in the 1 to 5 range. Further, each subtriangle would share an edge with at least one other, leaking information between pairs of subtriangles and changing the character of the puzzle.

It was not feasible to subdivide the Try grid into subtriangles, which had a fundamental impact on its design. While each cell in square-based Sudoku is the intersection of two lines (horizontal and vertical) and a subregion, each cell in Try is the intersection of three lines (along the three hexagonal axes). There is therefore less potential for strategic variety in Sudoku-based deductions in Try; having no scope for local deductions based on subregions, all deductions must be global.

Additional rules or constraints were needed to compensate for this lack of strategy. The ideal additions would fit naturally within the existing rules and geometry of the design, with as little impact as possible, and maximise strategic depth while minimising rule complexity \[9\].

### 3.3 A Creative Leap

It was an obvious step to try the connective goal from Y, one of my favourite board games, which is played on a triangular tessellation of hexagons. But how to adapt this connective rule from a turn-based adversarial game played with black and white pieces, to a solitaire puzzle played with numbers? The solution was simple: divide the numbers into two groups, odd and even, and assign one group to the setter and the other to the solver.

In mathematician Robert J. Weber’s model of human creativity \[8\], discussed in more detail elsewhere in this issue \[4\], new ideas are typically created by either joining, adding, refining or transforming existing ideas. The merging of Sudoku and Y was a joining of two existing ideas, with a hint of transformation to change piece-based cell ownership into odd/even parity.

These two existing ideas came from significantly different regions of the design space, in what Noy et al. \[10\] might call a ‘creative leap’. The design space will generally contain clusters of ideas, and merging ideas within a cluster will often work but not usually produce anything very novel. It is when ideas are merged from different clusters that true novelty occurs, and the more distant the clusters, the more potential to surprise.

### 3.4 Odd-Sized Boards

It quickly became obvious that Try challenges which required Y-based strategies to solve were more interesting than those that could be solved by Sudoku-based strategies alone. And there was a simple way to bias things in favour of the Y aspect: only use odd-sized boards.

Since the convention in number-based puzzles is to start the numbering at 1, odd sizes will contain one more odd than even numbers, as highlighted in Figure 15. This is enough to tip the strategic balance in the Y direction.

\[
N=5: \begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & \end{array}
\]

\[
N=7: \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & \end{array}
\]

\[
N=9: \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & \end{array}
\]

**Figure 15.** Odd sizes contain more odd numbers.

This greater representation of odd numbers facilitates Y-based strategies, as odd-only cells are then more likely to occur. This gives the setter more scope for authorial control in their challenges \[11\], and the solver more scope for applying Y-based strategies in their solutions.

This benefit is most obvious on smaller boards, such as the \( N=5 \) example shown in Section 2.2. It is noticeable on medium-sized boards, but yields diminishing returns as \( N \) increases.

### 3.5 Single Digits

Bearing in mind the benefits of odd board sizes, Try is best played at sizes \( N=5, 7 \) and 9. \( N=5 \) is the smallest board that allows a valid solution (you might like to prove this for yourself) and \( N=9 \) is the largest that can be played with single-digit numbers.

Hexadecimal digits could be used for larger board sizes, but for most players it would be less intuitive whether hexadecimal digits A, B, C, D, E and F are even or odd. Try at sizes above \( N=9 \) would be only for those of a mathematical nature.

This limit on board size is not just for reasons of aesthetics or notational convenience. The Y aspect is lost on larger boards, where seven or eight even candidates can occur per cell, making it less likely to allow the odd-only cells required for Y-based strategies.

### 3.6 Name

Once the design was finalised, it only remained to name this new puzzle game, a process that can require as much creativity as developing the design itself. The game was originally called ‘Tri-doku’, to capture the notion of Sudoku played on a triangular grid, then changed to ‘Trydoku’ to also flag the Y concept in the name.\[4\]

---

\[4\] Thanks to Greg Schmidt for this suggestion.
The name was then shortened to ‘Try’, which captures the essence of the game in the fewest possible letters. This is a contraction of ‘triangular Y’ that also hints at its deductive nature, as players must try hypotheses regarding which candidate numbers can legally occur in which cells, in order to make deductions.

The name ‘Try’ has a nice fundamental ring to it, but the downside of using such a common noun is potential anonymity, as it fails what I call the Google Test. A Google search for the terms try game returns over a billion hits, making it unlikely that a user searching for the game online will find relevant information amongst all the noise, without further clarification. This is an issue even for well known games with common words as titles, such as Go, but it can be a real concern for new games yet to establish a presence.

3.7 Generalisation

The solver’s implied task, of connecting the three board sides with a set of even numbers, can be described as the task of connecting the three board sides with a multiset of numbers that satisfy \((n \mod 2) = 0\), i.e. the even numbers \(\{2, 4, 6, 8, \ldots\}\). The Y rule can then be generalised to \((n \mod b) = 0\), where base \(b\) is some number \(\geq 2\).

For example, if \(b = 3\), then the Y rule would dictate that the three board sides must be connected by a multiset of numbers from \(\{3, 6, 9, \ldots\}\), which precludes connection by any multiset from \(\{1, 2, 4, 5, 7, 8, \ldots\}\).

The advantage of using a base \(b\) greater than 2 is that the Y rule would then provide an even greater constraint, as there would be fewer ways to make a legal Y than an illegal Y. This would allow a greater number of Y-based eliminations and Y-based strategies to be applied, potentially making challenges more interesting to solve, and could be a way to bring the Y aspect more into the game for larger board sizes.

However, a base other than 2 could make the Y rule confusing for players and ruin the intuitive elegance of the puzzle. Most players should be comfortable with the distinction between even and odd numbers, but asking them to distinguish between say \((n \mod 3) = 0\) and \((n \mod 3) \neq 0\) suddenly makes things look complicated.

3.8 A Hybrid Puzzle/Game?

It may be contentious to describe Try as a ‘hybrid puzzle/game’, as its Y rule is really a deduction constraint. However, if each challenge is viewed as a game played between the setter (the author who sets the challenge) and the solver (the player who attempts to solve the challenge) [11], then the setter can exert authorial control in their designs and, in effect, play a limited game of Y (as Odd) against their opponent, the solver (as Even).

As the solver follows the deductive steps to solve a given challenge, revealing more information with each step, the Y threats implicit in the challenge reveal themselves and provide additional information towards the solution. The solver must make choices that result in an even Y and constantly ask themselves: how can I stop the setter achieving an odd Y? A well designed challenge should impart the strong feeling of an intelligent hand working against the solver [11].

There is a clear distinction between the lower-level bookkeeping of the Sudoku-based strategies and the higher-level connective concerns of the Y-based strategies. The hand of the setter is behind both, as an absent player seeking to outwit and entertain the solver, but the Y-based deductions offer greater scope for depth and variety.

4 Computer Version

![Screenshot of the Try application](image)

5 A multiset, or bag, is a set of that may contain multiple instances of elements.

6 This example was suggested by mathematician Daniel Ashlock.
Figure 16 shows a screenshot of a Try editor implemented in Java. The editor loads $N = 5, 7$ and 9 challenges from a pregenerated database, and allows the user to interactively solve them.

The application helps the user by shading odd cells, rather than circling them, for visual clarity. It also automatically performs eliminations in response to user moves, reducing the trivial bookkeeping burden and making potential deductions more evident.

4.1 Challenge Generator

Try challenges are pregenerated by a separate application, using the following process:

1. Packing: First, a random packing of numbers that satisfies the puzzle’s rules is generated. This can be computed efficiently and avoids the need for storing the exponentially large quantities of valid packings, even after rotations, reflections and permutations are removed.

2. Hint Placement: A set of cells is chosen as the starting hint set, which is either reflectively or rotationally symmetrical, for aesthetic reasons. This step is repeated until a hint set is found with a unique deducible solution, according to Deductive Search [13].

3. Hint Reduction: The hint set is iteratively reduced, respecting its generative symmetry, as long as the challenge still has a unique and deducible solution.

4. Quality Check: The resulting challenge is tested for difficulty and estimated interestingness. It is discarded if it is too hard (deductions requiring many layers of embedding [13]), too easy (cells with trivial resolutions), or not interesting enough (can be solved without Y-based strategies).

This yields symmetrical challenges that are guaranteed to have unique deducible solutions, and are likely to be interesting. Future work will involve player surveys to validate and fine-tune the quality metrics, to maximise the quality of automatically generated challenges. This work will be summarised in a future paper, which will also include more detail on the automatic challenge generation, validation and evaluation processes.

5 Conclusion

Try is a new deduction puzzle that captures the essence of both the solitaire deduction puzzle Sudoku and the adversarial board game Y, in a single rule set. The story of its design highlights the benefits of drawing inspiration from widely differing sources, to achieve surprising and worthwhile results. Example Try challenges are printed throughout this issue, where space permits, to give a taste of its character.

Acknowledgements

Thanks to Greg Schmidt and Ken Shoda for playtesting Try, and the reviewers for their helpful suggestions. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

Available at: http://www.gapdjournal.com/extras/Try-1-0.jar
Dameo: A New Step in the Evolution of Draughts?

Christian Freeling, Mindsports

Draughts or Checkers is a family of games with a long evolution, but one which seems to have ground to a halt during the last half century or so. Anglo-American Checkers has been solved by computer \(^1\) to give a draw, International Draughts faces serious issues with drawishness in top-level play, while other forms such as Turkish, Frisian and Russian Draughts operate in the periphery. This paper presents my game Dameo, a relative newcomer to the Draughts family, that I believe takes an evolutionary step in the right direction.

1 Introduction

The Draughts family of games has evolved through the centuries, and existed in many forms \(^2\). The many versions share common familiar mechanisms, such as moving or jumping to empty squares in line, mandatory capture by jumping over enemy pieces, and men reaching the far row to be promoted to more powerful kings. Even though they differ in details (such as board size, starting position, whether kings move long range, direction of movement, etc.), they mostly share a long tradition of various fascinating strategy and tactics.

But in the most prevalent standardised form of International Draughts, as well as other popular variants, there is a problem of draws between very strong players, who have in effect become too skilled at the game in its current form. This article presents the evolution of the game and the increasing body of knowledge about it, along with several attempts to modify it to eliminate the drawishness which appears in top-level play, while still preserving the fundamental character of the game and its traditional tactics, which generations of players have enjoyed.

2 The Development of Draughts

The earliest ancestor of Draughts is generally thought to be Alquerque \(^2\). The famous Libro de los juegos (Book of Games) \(^3\), commissioned by Alfonso X, King of Castile, León and Galicia (1221–1284) and completed in 1283, contains a small and ambiguous section about the game.

Alquerque has characteristic Draughts mechanisms (move, or jump to capture). However, there are not yet the concepts of piece promotion to king or maximal capture \(^2\). Note too that some board locations have only orthogonal links to neighbours, while others also have diagonal links, so that the movement ability of a piece (in terms of orthogonal and diagonal movement) depends on its current location, unlike later games in the Draughts family.

Alquerque is played on a 5×5 square grid, with lines connecting adjacent points.

![Figure 1. Alquerque.](image)

Players take turns either:

1. moving one of their pieces to an adjacent empty point, or
2. jumping one of their pieces over an adjacent enemy piece to an adjacent empty point on the opposite side.

Chains of jumps are possible. A jump must be made, if possible.

More information exists about the evolution of Draughts in 16\(^{th}\) century Europe, where the availability of checkered Chess boards suggested their use for games other than Chess, triggering a period of rule experiments. An important property of the checkered board is that it suggests the use of only one subgrid, with pieces limited to diagonal moves, as in modern International

---

\(^1\)The author’s page gives an overview: http://mindsports.nl/index.php/on-the-evolution-of-draughts-variants

\(^2\)Maximal capture means that the mover must make a sequence of moves that captures the most pieces.

Draughts and American Checkers. Further East, the non-checkered Shatranj board prevailed, causing Draughts to evolve in a different direction.

2.1 Movement Direction

In the West, Draughts evolved on the diagonal subgrid. The game now known as Anglo-American Checkers must have been among the first to appear, as it used this board in the simplest and most natural manner: men move and capture diagonally forward only, while kings move and capture forward and backwards, also diagonally, and also with short jumps. Capture is compulsory in Checkers but maximal capture is not required, although some variants did try it. A parallel direction in the evolution of Draughts was orthogonal movement, as per Dama or Turkish Draughts.

**Turkish Draughts** is played on an 8\times8 square grid.

![Figure 2. Turkish Draughts.](image)

Men move to adjacent squares and capture adjacent enemy pieces by jumping over them, orthogonally forward or sideways only. A series of successive jumps is possible, possibly changing direction after each jump, and each captured piece is removed immediately after jumping it. Players must capture as many pieces as possible on a turn (maximal capture).

Men that reach the far row are promoted to kings, which move and capture in all four directions, and can fly long distances in a line, instead of being restricted to adjacent moves and jumps.

2.2 The Sticker

Even with restrictions such as men only moving and capturing forward, a rich variety of tactics existed and were documented in early versions of the game.

For example, a **sticker** (‘plakker’ in Dutch) is a man exposed to capture which cannot be immediately taken, as a maximal capture elsewhere taking higher priority. Stickers, therefore, exploit the maximal capture rule by offering short-term sacrifices for longer-term material or positional gains. Figure 3 shows an example, from a manuscript by Alonso Guerra from 1595 [4].

![Figure 3. Problem by Guerra: White to play.](image)

In this particular variant, men move and capture diagonally forward only, the king moves long, and maximal capture is required. The solution is shown in Figure 4. White has just moved c5–b6 to force Black to capture a7\times c5 (a). White then moves e3–d4 (b) which forces Black to make a maximal capture with the long-range king c1\times f4\times c7 (c).

White now has only one remaining piece, the sticker on d4, but can use it to capture d4\times b6\times d8K and promote to a king (d), which will then win easily against the two remaining black men (left as an exercise). This is not a particularly deep sequence, but it is nicely hidden and indicates the direction of things to come in the development of Draughts.

The sticker pervades the body of Draughts combinations whether in the International or Turkish game, or in modern descendents such as Croda and Dameo. It is inherently absent from variants that do not require maximal capture, such as Checkers or Russian Draughts.

---

3 A long-range king is one that can make long jumps over enemy pieces, ignoring intervening empty squares.

4 A **combination** in this context is a sequence of forced moves that leads to some advantage for the initiator.
2.3 Polish Draughts

Polish Draughts, a 10×10 game with largely the same rules as the current international game, emerged in the 17th or 18th century. Evolution occurs under pressure, and once Polish Draughts had taken root it immediately blossomed — and the pressure would disappear for the next two and a half centuries.

**International Draughts (or Polish Draughts)** is played on a 10×10 square grid.

![International Draughts](image)

Figure 5. International Draughts.

Men move diagonally forward and capture adjacent pieces in any diagonal direction. A man reaching the far row is promoted to a king, which can fly long in any diagonal direction while moving or jumping. Maximal capture is mandatory.

By the end of the 19th century, the rules were set, and theory would start to accumulate, shaping the game’s strategy and effectively preventing any further rule changes. These rules featured diagonal forward movement and a long-range king, with mandatory maximal capture in all four diagonal directions. This permits the important generic tactic mentioned above, the sticker. But there were also a number of additional rule details that would have a major impact on the game, in particular on its tactics. These are:

- To be promoted, a man must end its move on the back rank. A man that visits the back rank in a multiple capture move but does end on the back rank is not promoted.
- A multiple capture must be completed before the captured pieces are taken off the board.
- In a multiple capture, a piece may visit a square more than once, but it may not jump a given piece more than once.

International Draughts is the current standard form of the game, with international rankings and tournaments. Many variants exist and are played, for example Anglo-American Checkers, which is played on an 8×8 board without long-range kings.

2.4 The Coup Turc Manoeuvre

The multiple capture rules of Polish Draughts allows for a manoeuvre called the coup turc or ‘Turkish capture’, a pattern that ironically cannot occur in Turkish Draughts itself. For example, Figure 6 shows a coup turc by White, after Black has just moved b6–c5.

To trigger the coup turc, White moves a5–b6 (a), forcing Black to perform the maximal capture c7×a5×c3×a1K and promote itself to a king. White must then make the maximal capture e5×c7×e9×g7 (b), forcing Black’s new king to capture a1×h8.

---

5 The origin of Polish Draughts is often given as 1723 Paris, although this is disputed.
6 World Draughts Federation: http://fmjd.org
7 Coup turc example from: https://nl.wikipedia.org/wiki/Coup_Turc
White then moves g5–f6, which forces Black to make the maximal capture h8×d4×i3×g5 (c). Importantly, captured pieces are removed only after the move has been completed, so the king can not capture the man on f4, since the captured-but-not-yet-removed man on e3 blocks the jump. The king can not keep moving since the captured-but-not-yet-removed man on f6 also blocks further movement, so it must end its move on g5, as shown Figure 6 (d).

The white man on f4 is the only white piece remaining on the board. However, it is enough to eliminate Black with the indicated move f4×h6×j8×h10×f8×d6×b4 (d) to complete the coup ture.

Note that under the rules of Turkish Draughts, the king would have been able to capture all white pieces on the board on the previous turn; such subtleties are critical to Draughts combinations. The coup ture only occurs rarely in actual play, but it demonstrates the subtlety and beauty that is possible in such combinations.

3 20th Century and Drawishness

In the first decades of the 20th century, International Draughts was increasingly popular in Europe. In 1947 the Fédération Mondial du Jeu de Dames was founded by French, Dutch, Belgian and Swiss players. Draughts was a serious sport now, with ample media coverage for its heroes. Most newspapers featured at least a weekly column with Draughts problems. World Champion Piet Roozenburg was named Dutch Sportsman of the Year in 1948. The sport became more international when Russia entered the scene, considering the game a good vehicle to illustrate the superiority of socialism.

Building on the phenomenal legacy of Roozenburg and his predecessors, strategies became ever more refined, and a shadow that could easily have been ignored just a few decades earlier began to grow. Not in the lower and middle echelons, where combinations would play a major and often decisive role, but in top-level match play, where grandmasters could see possible adverse combinations as if they were roadblocks marked with fluorescent light. Grandmasters seldom make mistakes, and with top players ever more efficiently manoeuvring around opponents’ threatening combinations, games began to increasingly end in draws.

The first match to end in 20 draws was Andreiko-Koeperman in 1971, but Dybman and Gantwarg repeated the performance in 1987. Several more matches ended with 18 or 19 draws in the intervening years.

To solve the problem, the matches from 1995 onwards were organised in sets of regular games, blitz games and a tiebreak. The 2009 match between Schwartzman and Georgiev started off with 12 regular games, ending in 12 regular draws, then proceeded with the so-called ‘micro matches’ in which Schwartzman finally reached the required three wins.

Some players may dismiss this problem with drawishness by arguing that it appears only in high-level match play. However, such top-level games are supposed to be a sport’s best promotion, and if the best that a player can realistically hope for is a draw, then this may not inspire top players to strive for victory. This is perhaps one reason that Draughts is currently losing its status as an international spectator sport – especially when compared to Chess – even though its ‘grassroots’ popularity among casual players remains undiminished.

It seems obvious to me that the game at the top level has become frozen by its own accumulated theory, and that all new theory will only chart new roads towards draws. This was the impetus for my investigation into historical Draughts variants, for ways to reduce this tendency towards draws. This progression is described in the following sections.
3.1 Inspiration from Fanorona

Looking back on Draughts history, the 17th century game Fanorona is the only traditional descendent of Alquerque that features ‘contact capture’: moving a piece into contact with, or out of contact with, an enemy piece causes capture. Like Checkers, the game has been solved by a computer program \(^8\). Contact capture is no less logical than capture by jumping, so back in the 1980s I tried to put it into the time-tested framework of International Draughts. That worked, but the game lacked Draughts’ opportunities for penetrating combinations.

So I decided to integrate linear capture (the ability of a line of men to capture an opposing line) and, in consequence, linear movement (the ability of a line of men to move together). These additions worked so well that they outweighed the somewhat awkward necessity of additional rules, and it resulted in Bushka, a largely accepted, albeit somewhat exotic, member of the Draughts family.

Fanorona is played on a \(5 \times 10\) square grid (two adjacent Alquerque boards).

![Figure 7. Fanorona.](https://www.boardgamegeek.com/boardgame/4386/fanorona)

Each turn consists of one or more steps along a line to an adjacent point; each step must cause a capture, if possible, either by approach or withdrawal. Stepping to, or from, a point adjacent to an enemy piece (or line of enemy pieces) in the same line captures them. If the step simultaneously approaches one enemy piece and withdraws from another enemy piece, then only one of the captures occurs (mover’s choice). If a player cannot capture, then the player simply makes a one-step move. The goal is to capture all enemy pieces, or leave the opponent unable to move.

Afterwards, I considered linear movement without linear capture in a draughts game. It would make speeding up or slowing down no longer dependent on capture alone, as it is in the regular game. It would add flexibility to overall progress and lend more diversity to style. But with movement and capture both going in the same direction, I could see the ugly outlines of gridlock in the opening stages even without trying, so the idea was shelved in the early 1980s. And there it had been lying for almost two decades, when I encountered Croda.

3.2 Croda: A Major Breakthrough

Around 1995, Ljuban Dedić, a former Yugoslav champion of International Draughts and later president of the Croatian Draughts association, invented a brilliant game named Croda. \(^9\)

Croda is played on an \(8 \times 8\) square grid.

![Figure 8. Croda.](https://www.boardgamegeek.com/boardgame/19843/croda)

The rules are the same as International Draughts except that:

1. All capture is orthogonal only.
2. Men may move orthogonally or diagonally forward.
3. Kings may move only along open orthogonal lines.

Dedić had noticed that Turkish Draughts had two significant advantages over Draughts:

- Long multiple captures have more scope, as the orthogonal direction of capture aligns with the board edges.
- In International Draughts, a player needs four kings to capture a lone king (except in

---

\(^8\)https://www.boardgamegeek.com/boardgame/4386/fanorona

\(^9\)https://www.boardgamegeek.com/boardgame/18922/bushka

\(^10\)https://www.boardgamegeek.com/boardgame/19843/croda

\(^11\)In 1999 Leo Springer, an endgame expert and son of a former world champion, drew my attention to Croda.
a handful of exceptional 3-vs-1 positions). In Turkish Draughts two kings are sufficient, and thus a one-man majority may be enough to win.

So he set out to rebuild International Draughts with orthogonal movement, and this proved to be remarkably simple:

- Use the rules of International Draughts.
- Let men move orthogonally or diagonally forward, and let all capture be orthogonal.
- Start on the first three rows.

Simply replacing the sideways movement of Turkish Draughts by a diagonally forward movement brings the advantages of aligned (orthogonal) capture, forced progress, and a substantially smaller margin of draws, while keeping Draughts’ magnificent array of combinations. However, I noticed an important difference: the absence of one-on-one opposition.

This is one of the reasons that not all standard coups and combinations can be ‘translated’ to the new game, although the large majority of them can. What is lost is more than compensated by a series of new combinations specific to Croda. In terms of strategy, the emphasis may shift somewhat from tempo (the number of available moves behind a closed front) to pace (the average overall progress), but this is arguably an additional advantage rather than a disadvantage.

The only thing that bothered me was the initial position. I considered the initial position that is now Dameo’s, but though I saw that it not only looked better but arguably was better, I did not consider it sufficiently better to suggest the change.

Croda was a new car on a time-tested chassis, and all the Draughts world had to do was take the keys and drive away. Why did they ignore it? My bet is that Draughts players understand their specific game much better than its generic fundamentals.

4 Dameo: A Seamless Fit

After a couple of weeks, I realised that Croda would not suffer opening gridlock from introducing linear movement, as its move and capture options did not follow the same lines. And indeed it did not; in fact, these mechanisms fit together seamlessly. The game Dameo appeared to me literally in seconds in 2000, including its starting position, shown in Figure 9.

Dameo is played on an 8 × 8 square grid.

<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
</tr>
</tbody>
</table>

The rules are the same as Croda except:

1. An unbroken line of like-coloured men may move one step orthogonally or diagonally forward, if the square in front is vacant.
2. Kings may move along open orthogonal and diagonal lines.

Dameo had two immediate – and one subsequent – advantages over its predecessors:

- It adds flexibility in terms of overall progress (allowing for a more or a less aggressive style, as you like) while taking nothing away: every move possible in a Croda position is also possible under Dameo’s rules.
- It decreases the heaviness on the sides, which are popular alleys for progress and attack in ‘square’ Draughts games. Dameo requires economical manoeuvring between the sides and the centre.
- Two generic basic tactics emerged based on linear movement, the Double Square formation and the Ladder. The former plays a key role throughout the opening and middle game.

4.1 The Coup Turc in Dameo

Many traditional Draughts combinations also work for Dameo. For example, Figure 10 shows the Dameo equivalent of the coup turc manoeuvre shown earlier.

12 Situations in which single remaining men face each other, and if one moves, the other will definitely be able to jump it or move to block it. For further details, see: www.mindsports.nl/index.php/draughts-dissected
White has just moved d4–c5 to trigger the sequence leading to the *coup turc*. Black is forced to capture d5×b5 (a), then White a1–a2 (b) forces Black to maximally capture and promote with a3×a1×c1K (c).

White must then capture with c3×a3×a5×c5 (d), forcing Black’s new king to capture c1×c6 (e). Dameo permits moving a line of pieces, and White now does so, moving two pieces e2–c4 (f). This move leaves Black doomed, and is the start of the *coup turc* proper.

Black must make the maximal capture c6×c2×h2×h4×d4 shown in Figure 10 (g). Since captured men are removed only after the move is complete, the man on d3 is still covered by the man on d2, and since the man on c4 can not be jumped twice, the king must stop at d4. White can then capture d3×d5×f7×h7×h5 to clear the board, except for the black man at g6 (h). This is a winning position for White.

4.2 New Tactics in Dameo

Dameo’s linear movement also gives rise to the following basic tactics.

4.2.1 The Ladder

Figure 11 shows a simple example of a ladder, or repeated sequence of similar steps, in Dameo. A ladder is initiated by a diagonal linear move, which forces the opponent to capture in a zig-zag pattern along the line just moved from head to tail. White moves the line of three stones e2–b5 (a), forcing Black to make the maximal capture ladder move a5×c5×c3×e3 (b). White then clears the board with capture g2×e2×e4×e6×c6 (c).
Figure 12. A more complex ladder scenario in Dameo.

Figure 12 shows a more complex ladder scenario, in which White will be in trouble if the advanced black piece on a3 breaks through. White could exchange a man with c2–d3, forcing Black a3×c3 then White d3×b3, but that would leave White with an even more depleted left side. But White can use a ladder to force a winning play.

White moves a line of three stones g2–g5 (a), which forces Black to make the capture a3×c3×c1K and promote (b). White then moves g5–f6 (c), which forces Black to capture with f7×f5 (d), then White move g4–g5 (e) forces Black’s maximal capture g6×g4×g2 (f).

White uses the ladder tactic with f2–d4 (g), which forces the ladder capture d5×d3×f3 (h). White makes the maximal capture h2×f2×f4×f6×d6×d8K and promotes (i), Black is forced to capture two men in response e5×e3×e1K and also promote (j), then White cleans up with move d8×g8×g1×d1×a1×a8 (k) to clear the board (l). This simple ladder (g) allowed White to escape trouble and win.

4.2.2 The Double Square

I call the formation shown in Figure 13 (a) the double square, which can be resolved as follows. White’s move c3–e5 (a) is a sticker that forces Black to maximally capture c5×c3×e3 (b), which in turn forces White to capture e5×e3×c7 (c).

Each player has lost two men through this exchange, and moved one closer to promotion (d). The usefulness of resolving a given double square therefore depends on its position and context.

Note, however, that this simple example is also an endgame problem in its own right. For instance, if White had instead played the symmetrical line move d3–b5 diagonally left instead of right, then the advanced black man would have ended on b3 rather than e3, and from there have access to the corner, resulting in a draw. White made the right choice by playing c3–e5, and can force victory from here.

The double square formation shapes games throughout their opening and the middle phases. Its importance cannot be overemphasised.

13Provided by Leo Springer.
14Details can be found at: http://mindsports.nl/index.php/arena/dameo/67-basic-tactics?showall=&start=7
5 Conclusion

All sports evolve because contestants get better, including sports of the mind. Checkers and Draughts are in no danger as enjoyable games for casual players, but are in decline as top-level competitive sports, perhaps due to the tendency for top-level players to draw most games.

One way to revive interest in these games as a sport may be to find variants that are faster to play and more decisive at the master level, where it matters most. This was the impetus behind the search for Dameo, which, like its immediate parent Croda, is built on the time-tested framework of International Draughts.

Dameo has already shown that the scope of its combinations at least equals – if not surpasses – that of its parent games; it is faster to play, giving players more leeway in speed and style, but more importantly it has a considerably smaller margin of draws. It serves the credo that the powers of imagination and calculation that grandmasters display should be revealed by their game, not hidden by it. I believe that Dameo allows top players to reveal those powers again.

Acknowledgements

Thanks to Leo Springer for drawing my attention to Croda in the first place, and for his contribution of Dameo problems and endgames, to Ed van Zon for his unwavering support for Mindsports, and to the late Ljuban Dedić for Croda, the brilliant restoration of Draughts that provided the springboard for Dameo. The editors thank Henri Anemaat for his helpful suggestions.

References


Christian Freeling is a Dutch designer of strategy games including the Draughts-related Dameo, Hexdame, Emergo and Bushka. He is a co-founder of the online board games website Mindsports (http://www.mindsports.nl).

Address: Keteldiepstraat 44, 7523PM Enschede, The Netherlands.
Email: christian@mindsports.nl

Appendix

5.1 Singlegrid and Doublegrid Games

In International Draughts, Checkers and almost all other variants, pieces all move and capture along the same lines; let us call these singlegrid games. Returning for a moment to Draughts’ earliest ancestor Alquerque, notice that there are 12 points on the board that allow orthogonal moves only, and 13 points that allow both orthogonal and diagonal moves.

Superimposing a checkered board, as shown in Figure 14, reveals this to be equivalent to moving orthogonally from light squares, and orthogonally or diagonally from dark ones. Alquerque could be called a doublegrid game, with a piece’s movement options based on its location.
In Armenian Draughts, diagonal movement options were introduced into an otherwise orthogonal game. In Frisian Draughts, an orthogonal capture option was introduced into an otherwise diagonal game. In these games the options were added *ad hoc* to their respective parent games, presumably to increase decisiveness or enhance tactics.

One might conclude that the seeds for doublegrid were there all along, from the very beginning to the present time, and that occasionally a seed took root. The mainstream of national and local variants, however, remained singlegrid in that capture and movement both used the same single direction: either diagonal (as in most traditional Western variants) or orthogonal (as in Turkish Draughts), but not both directions in the same game. In that sense, Croda and Dameo may be considered a new breed and possibly new steps in the evolution of Draughts.

---

**Try Challenges #5 and #6**

Fill the grid with numbers 1 to 7, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.
Catch-Up: A Game in Which the Lead Alternates

Aaron Isaksen  Tandon School of Engineering, New York University
Mehmet Ismail  Department of Economics, Maastricht University
Steven J. Brams  Department of Politics, New York University
Andy Nealen  Tandon School of Engineering, New York University

Catch-Up is a two-player game in which the players’ scores remain close throughout the game, making the eventual winner – if there is one – hard to predict. Because neither player can build up an insurmountable lead, its play creates tension and drama, even between players of different skill. We show how the game is played, demonstrate that its simple rules lead to complex game dynamics, analyse some of its most important properties, and discuss possible extensions.

1 Introduction

It is a challenge to design interesting two-player games with simple rules that keep the score close, even between players of different skill. When the game score is close, players experience tension and drama by not knowing too far in advance who will win. This drama has been discussed qualitatively [2, 10] and quantitatively [9].

To enhance tension, games often have catch-up mechanisms, sometimes called rubber banding [10]. Players who are behind can receive a boost to help them recover, and players who are ahead are prevented from maintaining or accelerating their lead.

Economists describe the desire to minimise inequality as inequity aversion, wherein people prefer rewards to be allocated evenly [12]. Designing games with inequity aversion can create a more balanced competitive experience, allowing experts and novices to enjoy playing together as the score will remain close throughout the game. A game is also often more enjoyable if one is not losing by a large amount. However, too much catching up can lead to games in which the winner is not determined until the very end, making early moves meaningless.

1.1 Catch-Up

We present Catch-Up, a minimal game [5] with simple rules that can be learnt quickly, invented by the authors with these ideas in mind [1]. The rules are as follows.

Catch-Up starts with a set of numbers $S$.

1. Two players, $P_1$ and $P_2$, begin with scores, $s_1$ and $s_2$, of zero. $P_1$ starts by removing a number from $S$, which is added to his or her score.
2. The players then take turns removing one or more numbers from $S$, one by one, until the acting player’s score equals or exceeds the opponent’s current score.
3. When this is no longer possible, the acting player receives any remaining numbers. The player with the higher score wins; the game is drawn if scores are tied.

Figure 1. An example game of Catch-Up won by player $P_2$.

---

The demo game and code used in this paper are available at: [http://game.engineering.nyu.edu/catch-up](http://game.engineering.nyu.edu/catch-up)

Players are therefore uncertain who will win a game of Catch-Up until the end. The game is surprisingly complex, given the simplicity of its rules, with no trivial heuristics that enable players to win every time.

We illustrate with several examples how Catch-Up is played, discuss optimal strategies and heuristics, analyse some important properties, and discuss possible extensions of the game. Figure 1 shows a short game won by player P₂, by way of example.

1.2 Combinatorial Aspects

We study this combinatorial game using an approach similar to that used in Scientific American articles by Martin Gardner [6] and Winning Ways For Your Mathematical Plays [7], but we also provide context for game designers. In particular, we show how the set – the numbers the players start with – affects the game’s complexity and play dynamics.

Catch-Up is a combinatorial perfect information game, so even though the players have close scores throughout the game, there exist optimal strategies to win or tie. Thus, the scoring mechanism does not necessarily reflect who is more likely to win the game: a player may be in a game-theoretic winning position even though his or her score is lower than the other player’s. Catch-Up shows that creating scoring systems in which the current score is a reliable and meaningful indicator, in games with significant catch-up mechanisms, is indeed a challenge.

Catch-up mechanisms exist in many games, from board games using variable scoring (e.g. Hare & Tortoise [8]) or time tracks (e.g. Tokaido [9]) to video games with variable powerups (e.g. Super Mario Kart [10]). The game Catchup, by Nick Bentley [11], uses a catch-up rule that permits the player who is behind to add an extra piece each turn. Zhang-Qi [12] is similar (though we were not aware of it when designing ours) but uses a specific 32-element set, places markers on a uniquely shaped board, and describes the catch-up rule as one of two optional rules.

2 Examples of Play

The rules of Catch-Up are presented in the shaded box on the preceding page. We explore several example games to show that the rules, although minimal, define a game with interesting non-trivial properties.

We use the notation Catch-Up(S) to describe the game played with set S. For example, Catch-Up(1,...,N) is played with S = {1,...,N}, the consecutive positive integers from 1 through N. For clarity, P₁ is referred to as she and P₂ as he.

2.1 Catch-Up(1,...,4)

Figure 2 shows the full game tree for Catch-Up(1,...,4). Assuming optimal play by P₁ (triangle) and P₂ (square), winning, drawing, and losing positions, and moves for the acting player are indicated. Numbers show the numbered pieces selected on that move. Thicker lines indicate optimal plays. Above each node, {B}:D gives the remaining numbers B and the score differential D.

One possible game might play out as follows, which is shown in steps in Figure 1. The set starts with $S = \{1, 2, 3, 4\}$. P₁ initially removes (3), and is ahead 3 – 0. Play then switches to P₂, who can choose from $\{1, 2, 4\}$ and removes (2). Since the score is 3 – 2 and P₂ is still behind, P₂ needs to remove another number. P₂, choosing from $\{1, 4\}$, removes (4). Thus, on P₂’s turn, the entire move was to remove (2, 4), and the score is now 3 – 6. Since P₂ is ahead, play switches back to P₁.

![Figure 2. The full game tree for Catch-Up(1,2,3,4).](http://boardgamegeek.com/boardgame/72711/zhang-qi)

---

3http://boardgamegeek.com/boardgame/72711/zhang-qi
The set contains only \{1\}, which \(P_1\) removes. The game ends with a final score of \(4 - 6\), so \(P_2\) is the winner by 2. This game could also have ended in a draw as follows: \(P_1\) selects (2), \(P_2\) selects (1, 4), and \(P_3\) selects (3), tying the game at 5 – 5 and illustrating how \(P_2\) can force a draw.

Because of Rule 2, players always start their turns either tied or behind the other player. This means the player’s task is at least to catch up to the other player, but neither player can ‘snowball’ or jump far ahead. Conversely, this same rule means that players will always end their turns either tied or ahead of the other player.

In order to keep players from memorising strong opening moves, we propose that players play with a randomised set – with repeated or missing numbers – such that there are too many possible game trees for players to memorise.

### 2.2 Physical Implementation

If Catch-Up is played as an abstract mathematical game, it requires detailed bookkeeping, which some players may find difficult. We propose a version played with physical pieces on a table, as shown in Figure 1, illustrating the moves described in Section 2.1. The pieces are designed to fit next to each other, such that the lengths can be quickly determined to see whose turn it is.

We believe the physical version is more pleasurable to play because the physical pieces simplify the arithmetic calculations, making the game more accessible. If the shortest piece is 1 centimetre long, a tie game of Catch-Up(\{1, ..., 12\}) would end up being \((1 + ... + 12)/2 = 39\) cm long, with the largest win margin at most 12 cm long.

### 2.3 Puzzle-Like Quality

Catch-Up has a puzzle-like quality, making it challenging to find solutions that lead to a win or draw. For example, Figure 3 shows two subtrees of Catch-Up(\{1, ..., 7\}).

![Figure 3. Two subtrees of Catch-Up(1, ..., 7).](image-url)
In Figure 3 (top tree), $P_2$ (represented by squares) is in a winning position, but he must proceed carefully. This position was reached by $P_1$ (represented by triangles) initially removing $\langle 3 \rangle$, $P_2$ removing $\langle 5 \rangle$, and $P_1$ removing $\langle 6 \rangle$, leaving a set of $\{1, 2, 4, 7\}$ and a score difference of 4. One move leads to a win, one move leads to a draw, and all other moves lead to losses (random play here would lead to a 7/8 chance of choosing a sub-optimal move). $P_1$’s optimal move is to choose the largest sum possible, removing either $\langle 1, 2, 7 \rangle$ or $\langle 2, 1, 7 \rangle$. Each uses the same numbers and reaches the same score; strategically equivalent moves are indicated with ‘or’ in Figure 3.

This strategy of maximising one’s lead, however, does not always work. The subtree of Figure 3 (bottom tree) is reached by $P_1$ initially removing $\langle 2 \rangle$, $P_2$ removing $\langle 5 \rangle$, and $P_1$ removing $\langle 7 \rangle$, giving a score difference of 4. If $P_2$ then maximises his score by choosing $\langle 3, 6 \rangle$, worth 9 points, this leads to a forced draw. But if $P_2$ chooses the lower valued $\langle 1, 6 \rangle$, worth only 7 points, he forces a win. Making this even more tricky, choosing $\langle 3, 4 \rangle$, also worth 7 points, leads to a forced loss for $P_2$.

In Section 3 we discuss various simple strategies and heuristics that beginning players might use to help navigate the game tree. This shows the relative effectiveness of each heuristic.

### 2.4 Maximising Is Not Optimal

In the previous section, we showed that a strategy of selecting the largest sum of numbers possible is not always an optimal strategy, though it is an obvious heuristic that a player might try. As another example, in a game of $N = 5$, with a set $\{1, 2, 3, 4, 5\}$, if $P_1$ always selects the numbers that gives her the largest lead, she will lose: $P_1$ initially removes $\langle 5 \rangle$, $P_2$ can then remove $\langle 1, 3, 4 \rangle$, forcing $P_1$ to choose $\langle 2 \rangle$ and lose the game 7 – 8.

This happens specifically because of the inequity aversion of Rule 2. If, for example, players were required to select a fixed number of numbers on each turn; then a maximising-score strategy would be dominant, making the game trivial. By contrast, the rules of Catch-Up lead to a game tree that makes optimal choices non-trivial: there are no immediately obvious choices to win every game.

### 2.5 Endgame

On every turn, a player of Catch-Up comes from behind or from a tied score. However, there are many cases in which a player, who will lose if the opponent plays optimally, can still come back to win very late in the game if the opponent makes a mistake on his or her last move. This implies that both players must focus on winning up until their very last moves.

In Figure 4, we show an example of a subtree of Catch-Up($\{1, ..., 7\}$) wherein optimal play produces a loss for $P_1$, but there is still a chance for a win with the last moves in the game if $P_2$ plays non-optimally. To reach this position, assume $P_1$ chooses $\langle 1 \rangle$, $P_2$ chooses $\langle 2 \rangle$, $P_1$ chooses $\langle 3 \rangle$, and $P_2$ chooses $\langle 6 \rangle$, so the score difference is 2 and $\{3, 4, 7\}$ remain in the set. Now, if $P_1$ chooses $\langle 7 \rangle$, she will lose when $P_2$ is forced to choose $\langle 3, 4 \rangle$. However, if $P_1$ chooses $\langle 3 \rangle$ or $\langle 4 \rangle$ – putting her 1 or 2 ahead – then $P_2$ must choose $\langle 7 \rangle$ to win.

### 2.6 Drawn Games

Drawn games are sometimes possible in Catch-Up if the sum of the numbers in $S$ is even. Whether optimal play leads to a draw, or a win for $P_1$, depends on $S$. Games that end in a draw may be dissatisfying for some players because there is no winner (although draws do not seem to bother many Chess players, for example).

Whether Catch-Up permits draws is solely determined by the set $S$. In the case of Catch-Up($\{1, ..., N\}$), it depends on the value $N$. For all $n \geq 0$, games of the form $N = 4n + 1$ and $N = 4n + 2$ always have a winner by at least one point, because the sum of all the points 1, 2, ..., $N$ is odd; there is no way to split them evenly. Conversely, games of the form $N = 4n + 3$ or $N = 4n + 4$ can have games that end in a draw, because the sum of all the numbers is even. We provide a proof of this in Appendix 5.1 and we calculate in Section 4.9 how often draws will occur as a function of $N$.

For games of the form Catch-Up($\{1, ..., N\}$) with $N = 4n + 3$ or $N = 4n + 4$, which can have games that end in a draw, we have calculated up to $N = 20$ that optimal play by both players leads to a draw (see Section 3.1). However, optimal play in any even-sum game of Catch-Up($S$) for any arbitrary $S$ does not necessarily produce a draw. Consider an even-sum game with repeated numbers $S = \{2, 2, 2, 3, 3\}$, shown in Figure 5 which sum to 12. Here $P_1$ can force a win by initially choosing $\langle 2 \rangle$. Drawn games are still possible for this set, but they are not the result of optimal play.
Furthermore, it is easy to see that some even-sum games do not even permit a draw. Consider Catch-Up(\{2, 4, 6, 8, 10\}), which is even-sum, but obviously no subsets of these numbers can produce a 15 – 15 tie.

2.7 Importance of the First Move

One criticism of catch-up type mechanisms is that the early moves in the game have no importance. We show here that the first move \(P_1\) makes in Catch-Up(\{1, ..., 7\}) has an impact on the percentage of ways that \(P_1\) can win, lose, or draw. In Table 1, each row shows the change from 50%–50% in the percentage of ways that the game can end in a win, lose, or draw, given that \(P_1\) makes the indicated first move.

![Figure 5. An even-sum game in which \(P_1\) can force a win, but which can also end in a draw.](image)

<table>
<thead>
<tr>
<th>Move</th>
<th>(\Delta) Win%</th>
<th>(\Delta) Lose%</th>
<th>(\Delta) Draw%</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>0.60%</td>
<td>0.60%</td>
<td>-1.19%</td>
</tr>
<tr>
<td>(2)</td>
<td>-2.46%</td>
<td>3.65%</td>
<td>-1.19%</td>
</tr>
<tr>
<td>(3)</td>
<td>6.43%</td>
<td>-6.90%</td>
<td>0.48%</td>
</tr>
<tr>
<td>(4)</td>
<td>3.10%</td>
<td>-1.90%</td>
<td>-1.19%</td>
</tr>
<tr>
<td>(5)</td>
<td>0.32%</td>
<td>-0.79%</td>
<td>0.48%</td>
</tr>
<tr>
<td>(6)</td>
<td>-4.13%</td>
<td>3.65%</td>
<td>0.48%</td>
</tr>
<tr>
<td>(7)</td>
<td>-3.85%</td>
<td>1.71%</td>
<td>2.14%</td>
</tr>
</tbody>
</table>

Table 1. Percentage change with \(P_1\) moving first.

By choosing (3), \(P_1\) increases the ways of winning by 6.43% and reduces the ways of losing by 6.90%. Conversely, choosing (6) decreases the ways of winning by 4.13% and increases the ways of losing by 3.65%. Clearly, the first move has an impact on the ability of non-optimal players to achieve a win, loss, or draw; but this has no bearing on optimal play.

3 Strategies

Catch-Up, for any finite set \(S\), is a finite two-person zero-sum game of perfect information, so there exists a pair of optimal strategies such that (i) \(P_1\) can guarantee a win, (ii) \(P_2\) can guarantee a win, or (iii) the game is a draw. In order for a perfect-information game to be non-trivial, the optimal strategy should not be obvious to play.

In addition, different strategies should present a heuristic tree [10], such that there are some simple heuristics that new players can learn, and better performing but more complicated heuristics for more sophisticated players.

3.1 Optimal Play

We cannot yet prove whether Catch-Up(\{1, ..., \(N\)\}) is a win, loss, or draw for \(P_1\) for any \(N\); however, for a given set, we can efficiently run a minimax algorithm with alpha-beta pruning and transposition tables [12] to solve the game value, assuming optimal play by both players. We have calculated the game values for Catch-Up(\{1, ..., \(N\)\}) up to \(N=20\). Results for optimal play are shown in Table 2 in the optimal play row, with -1 being a loss for \(P_1\), 1 being a win for \(P_1\), and 0 being a tie game.

As described in Section 2.6, Catch-Up(\{1, ..., \(N\)\}) games of the form \(N = 4n + 3\) or \(N = 4n + 4\) permit draws. We have calculated that these games, up to at least \(N = 20\), are draws with optimal play. We believe that this pattern holds for all \(n\), though we have not been able to prove this and can only offer it as a conjecture. Using Monte-Carlo tree search [13], we have explored values of \(N = 23, 24, 27,\) and 28 and did not find any contradictions.

<table>
<thead>
<tr>
<th>(N)</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal play</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. Optimal play values for Catch-Up(\{1, ..., \(N\)\}) relative to \(P_1\): 1 is a win, -1 is a loss, and 0 is a draw.
3.2 Human-Playable Heuristics

Although machines can efficiently search a game tree for optimal moves, humans do not think in the same way and, generally, do not find it enjoyable (or possible) to exhaustively explore every move when playing a game.

In order for a strategy to work for human players, we need effective heuristics that are accessible and can be easily used. And for a game to have lasting depth, simple heuristics must be generally less effective than more complex ones, so there is a benefit for continued study and improvement.

We analysed several simple human-usable heuristics for playing Catch-Up. For these heuristics, if multiple moves could be chosen, one of them is picked at random. We do not claim that these are the only heuristics for players, or that players should follow any of them. Instead, they provide a starting point for strategies that new players might try, which help us understand if the game can be enjoyed by beginners.

1. Random: Players choose any move at random.
2. MaxScore: Players maximise their scores on every turn, extending their leads by as much as possible.
3. MinScore: Players minimise their scores on every turn, keeping their scores as close as possible.
4. UseMostNums: Players use as many numbers as possible, reducing the numbers available for the opponent.

For Catch-Up({1, ..., 10}), which is an odd-sum game, optimal play by both players leads to a loss for player 1, but it is difficult for humans to play optimally. Instead, we can test the various simple heuristics and compare how they perform against each other.

For example, Table 3 shows the probabilities of player 1 winning when playing each of her heuristics against each of player 2’s 100,000 times. The value in each cell indicates the percentage of games in which player 1 wins; a value of 1 means that player 1 always wins, whereas a value of 0 means that player 1 always loses. Values > .5 in Table 3 are good for player 1, whereas values < .5 are good for player 2. Players are assumed to use the same heuristic throughout the entire game, without switching or adapting within a game to what the other player is doing.

The P1 Random vs P2 Random cell shows that random play gives close to a 50% chance of winning, which indicates completely unskilled play will not favour one player over the other. Looking at the first column, we see the effect of player 1 using each heuristic against player 2 Random, and that P1 MaxScore is the best of the four heuristics, improving player 1’s win rate to approximately 63%, whereas P1 MinScore is a bad heuristic, reducing the win rate to around 37.6%. Similarly, if we look at the first row, which shows the effect of player 2 using each heuristic against player 1 Random, we see that P2 MaxScore is the best heuristic for player 2.

However, if both players adopt the MaxScore heuristic, this is bad for player 1, reducing player 1’s win rate to around 14.3%. Player 1, playing against a P2 MaxScore heuristic, would do better to use the P1 MinScore heuristic, which was previously a bad choice. But this can lead to player 2 in turn switching to the P2 MinScore heuristic, in which player 2 now wins every game. Likewise, player 1 now does better by switching back to the P1 Random heuristic.

Given these simple heuristics, we already see an interesting pattern, in which there is not one dominating heuristic. This is an indication that Catch-Up does not have a trivial or obvious solution for human players. We believe this rock-paper-scissors balance, in which different heuristics perform better in some cases but not others, but no one heuristic dominates, is an important characteristic of deep and interesting games.

These heuristics do not necessarily generalise to other sets S. Just because a heuristic does well in Catch-Up({1,...,10}) does not mean it does well in Catch-Up({1,...,9}), another odd-sum game. For example, P1 MinScore vs P2 MaxScore wins 79.7% for player 1 in the former game, but flips to only a 34.1% win rate for player 1 in the latter game. Clearly, these heuristics offer only a glimpse into optimal play of Catch-Up.

3.3 Climbing the Heuristics Tree

We can simulate a player adopting a new, better heuristic by combining the previous four heuristics. Instead of deciding between multiple moves randomly, we can apply a second-level heuristic.

<table>
<thead>
<tr>
<th></th>
<th>P2 Random</th>
<th>P2 MaxScore</th>
<th>P2 MinScore</th>
<th>P2 UseMostNums</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 Random</td>
<td>50.01%</td>
<td>41.66%</td>
<td>60.87%</td>
<td>46.53%</td>
</tr>
<tr>
<td>P1 MaxScore</td>
<td>63.04%</td>
<td>14.28%</td>
<td>29.78%</td>
<td>48.06%</td>
</tr>
<tr>
<td>P1 MinScore</td>
<td>37.56%</td>
<td>79.74%</td>
<td>0.00*</td>
<td>37.79%</td>
</tr>
<tr>
<td>P1 UseMostNums</td>
<td>52.70%</td>
<td>49.73%</td>
<td>60.54%</td>
<td>50.87%</td>
</tr>
</tbody>
</table>

Table 3. Win percentages for P1 when playing different heuristics against P2 in Catch-Up({1,...,10}).
to choose between multiple moves. For example, $P_1$ using $\text{UseMostNums + MinScore}$ would first pick moves to use the most numbers, and if there is more than one remaining move to choose from, she chooses the move that sums to the smallest number. As before, any final remaining options are eliminated by selecting one at random.

If $P_2$ adopts this $\text{UseMostNums + MinScore}$ combination heuristic, but $P_1$ stays with the original heuristics, $P_2$ now wins every game against two $P_1$ heuristics, and wins a slight majority of games otherwise, as shown in Table 4.

Table 4. $P_2$ win rate using combination heuristic.

<table>
<thead>
<tr>
<th>$P_1$</th>
<th>$P_2$ UseMostNums + MinScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>45.88%</td>
</tr>
<tr>
<td>MaxScore</td>
<td>0.00%</td>
</tr>
<tr>
<td>MinScore</td>
<td>0.00%</td>
</tr>
<tr>
<td>UseMostNums</td>
<td>45.55%</td>
</tr>
</tbody>
</table>

Note that the purpose of this section is not to present the reader with the best heuristics, but to show that Catch-Up provides a compelling platform for developing effective heuristics for human play.

4 Properties of Catch-Up

In analysing the properties that follow, we do not always have analytical proofs for all $S$, or for all values $N$ for $S = \{1, \ldots, N\}$, so we offer conjectures and computational analysis of games for relatively small $N$. Some of these metrics have been used to determine whether a game is well designed, which have been shown to be effective at generating new game designs [9].

4.1 Total Points Scored

Catch-Up ends only when all the numbers have been incorporated in either $P_1$’s score $s_1$ or $P_2$’s score $s_2$. For Catch-Up($\{1, \ldots, N\}$), the sum of the players’ scores will be equal to the triangular number $T(N)$ [14 Seq. A000217] :

$$s_1 + s_2 = T(N) = \sum_{i=1}^{N} i = \frac{N(N + 1)}{2} \quad (1)$$

4.2 The Lead is Always $\leq \max(S)$

A player can never be winning by more than $M = \max(S)$, the largest number in $S$, which includes the final move. Thus, a designer can choose elements of $S$ to force the game always to be within a range of $M$ points.

This is relatively easy to prove. Let $P_1$ be the acting player, and $P_2$ be the opponent. A turn must end when a $P_1$ ties or exceeds $P_2$’s score, so right before choosing the last number that ends a turn, $P_1$ must either be starting tied or be behind, so $s_i - s_j \leq 0$. The largest number that can possibly be chosen as the last selection on the turn is $M$. Thus, at the end of the current turn, the score difference can be no more than $M = \max(S)$.

4.3 Maximum Points Per Turn

We can analyse the maximum number of points that can be earned on a turn of Catch-Up($\{1, \ldots, N\}$). On $P_1$’s first turn, she would choose $N$, the largest number available. On $P_2$’s turn, he can first select numbers that sum to $N - 1$ (if $P_2$ were to exceed $N - 1$, then the turn would immediately end) plus the largest remaining number, $N - 1$. This can be done by selecting $(1, N - 2, N - 1)$, which gives $P_2$ a maximum sum of $2N - 2$ on a single turn. Note that $P_2$ would also have achieved this if $P_1$ first chose $(N - 1)$, and $P_2$ responded by choosing $(N - 2, N)$, also giving a total of $2N - 2$.

4.4 Game-Tree Size

The game-tree size gives the total number of unique play-throughs, iterating through all valid moves of the game. This is equivalent to counting the number of terminal nodes in the game tree. For simplicity, we consider each permutation of a player’s removal choices in a single turn to be a distinct branch, although the order of removals within a turn does not matter during play.

Large game trees are more difficult for players to utilise in play, as they do not permit memorisation of the best moves; however, they also make it computationally harder for analysis by adversarial search. By increasing the size of the set $S$, the game tree rapidly increases in size.

The game-tree size is exactly $N!$, which is the number of ways the numbers in the set can be picked, and then assigning turns after determining the order, the numbers are picked to make it a valid game of Catch-Up. Table enumerates all possible games of Catch-Up($\{1, \ldots, N\}$) for up to $N = 18$ and counts the number of terminal nodes, verifying the game-tree size is indeed $N!$.

4.5 State-Space Size

State-space size is the number of possible states of the game, reflecting the fact that many states can be reached from multiple moves [15]. This process converts the game tree into a directed acyclic graph, because a game state represented in the graph can have multiple parents.
<table>
<thead>
<tr>
<th>N</th>
<th>game-tree size (N!)</th>
<th>state-space size</th>
<th>$P_1$ optimal play value</th>
<th>max branch factor</th>
<th>min depth</th>
<th>win %</th>
<th>tie %</th>
<th>loss %</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>11</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>16.67%</td>
<td>66.67%</td>
<td>16.67%</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>33</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>33.33%</td>
<td>33.33%</td>
<td>33.33%</td>
</tr>
<tr>
<td>5</td>
<td>120</td>
<td>90</td>
<td>1</td>
<td>16</td>
<td>3</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>6</td>
<td>720</td>
<td>236</td>
<td>1</td>
<td>36</td>
<td>3</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>7</td>
<td>5040</td>
<td>591</td>
<td>0</td>
<td>78</td>
<td>3</td>
<td>38.57%</td>
<td>22.86%</td>
<td>38.57%</td>
</tr>
<tr>
<td>8</td>
<td>40320</td>
<td>1453</td>
<td>0</td>
<td>150</td>
<td>4</td>
<td>39.39%</td>
<td>22.14%</td>
<td>39.39%</td>
</tr>
<tr>
<td>9</td>
<td>362880</td>
<td>3484</td>
<td>-1</td>
<td>272</td>
<td>4</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>10</td>
<td>3628800</td>
<td>8174</td>
<td>-1</td>
<td>474</td>
<td>4</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>11</td>
<td>39916800</td>
<td>18799</td>
<td>0</td>
<td>886</td>
<td>5</td>
<td>41.90%</td>
<td>16.19%</td>
<td>41.90%</td>
</tr>
<tr>
<td>12</td>
<td>479001600</td>
<td>42587</td>
<td>0</td>
<td>1470</td>
<td>5</td>
<td>42.47%</td>
<td>15.06%</td>
<td>42.47%</td>
</tr>
<tr>
<td>13</td>
<td>6227020800</td>
<td>95126</td>
<td>0</td>
<td>2448</td>
<td>5</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>14</td>
<td>87178291200</td>
<td>210064</td>
<td>0</td>
<td>3894</td>
<td>6</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>15</td>
<td>1.30767E+12</td>
<td>459225</td>
<td>0</td>
<td>6158</td>
<td>6</td>
<td>43.86%</td>
<td>12.28%</td>
<td>43.86%</td>
</tr>
<tr>
<td>16</td>
<td>2.09228E+13</td>
<td>995349</td>
<td>0</td>
<td>10284</td>
<td>6</td>
<td>44.23%</td>
<td>11.53%</td>
<td>44.23%</td>
</tr>
<tr>
<td>17</td>
<td>3.55687E+14</td>
<td>2141652</td>
<td>1</td>
<td>16048</td>
<td>6</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>18</td>
<td>6.40237E+15</td>
<td>4579850</td>
<td>-1</td>
<td>24762</td>
<td>7</td>
<td>50.00%</td>
<td>0.00%</td>
<td>50.00%</td>
</tr>
</tbody>
</table>

Table 5. Measures for Catch-Up(\{1, ..., N\}) for values of \(N = 3\) to \(N = 18\).

In Catch-Up, the necessary states to track are: current player, current score and numbers remaining in the set\(^4\). We do not have an analytical bound for the state-space size, but empirical data generated for small \(N\), shown in Table , demonstrates that it grows much more slowly than the game-tree size. For large \(N\), the state-space size is much smaller because there are many ways to reach the same game state using different moves.

For example, for any Catch-Up(\{1, ..., N\}) for \(N \geq 3\), the following game traces all reach an identical game state with tied score 3 − 3: ⟨3, 1, 2⟩; ⟨⟨3, 2⟩, 1⟩; ⟨⟨2, 3⟩, ⟨1⟩; ⟨⟨1⟩, ⟨⟨3⟩, ⟨2⟩⟩. Thus, huge benefits occur from caching results in a transposition table [12] when exploring the game graph for optimal moves.

### 4.6 Game-Tree Depth

The depth of a game tree for Catch-Up(S) can be no deeper than |S| turns. Thereby, the designer or players can control the length of the game by choosing the size of S.

This maximal depth occurs when each player selects the smallest number in the set, with each ending a turn with only one number removed. This gives a total of \(N\) turns. Games can certainly end sooner, because on some turns a player may select more than one number, decreasing the number of turns for that path in the tree.

The minimum length of the game is also determined by the size of the initial set. We do not have an analytical lower bound for Catch-Up(\{1, ..., N\}), but we present calculated minimum-depth values in Table.

### 4.7 Maximum Selections per Turn

For Catch-Up(\{1, ..., N\}), we can calculate \(K\), the maximum number of numbers that can be selected on a turn. This can help a designer understand how long a turn will take for players to evaluate. We show in Appendix 5.2 that \(K\) is \(O(\sqrt{N})\) and has an exact analytical value of:

\[
K = \lfloor \sqrt{2N - 7/4} + 1/2 \rfloor
\]

### 4.8 Branching Factor

The maximum branching factor, which we call \(B_{\text{max}}\), tells us how many possible moves there are on a turn in the worst case. The higher the branching factor, the more complicated a game can be for a player to explore. The maximum branching factor for Catch-Up(\{1, ..., N\}) is:

\[
B_{\text{max}} = O \left( N^{\sqrt{2N+1}} \right)
\]

A derivation of this upper bound is provided in Appendix 5.3.

In Table , we show the empirical maximum branching factor, which is the maximum of the number of first moves by \(P_1\) and the number of replies (to first moves) by \(P_2\). This table clearly shows that the maximum branching factor is exceedingly high for a game, making it difficult to explore the entire early game tree for large \(N\).

As Catch-Up proceeds, there are fewer numbers in S to choose from, so the branching factor \(B_t\) for each turn \(t\) will decrease until the final move, which forces the last player to select all remaining numbers. The average branching factor \(B_{\text{avg}}\) will be less at each layer of the game tree.

---

\(^4\)If this information is known, it does not matter how the removed numbers were chosen to get to this state.
We do not have an analytical bound for $B_{\text{avg}}$, but can calculate it empirically for small $N$, allowing us to generate the entire tree.

In Table 6 we give the average and maximum branching factors per level of the game tree for, as an example, Catch-Up($\{1, ..., 12\}$). Level $l$ of the tree represents the possible game states and moves available on turn $l$. To calculate the average branching factor, we expand the entire game tree and then calculate how many moves there are available on each level of the tree divided by the number of unique states on that level. The average and maximum branching factors peak on Turn 2 ($P_1$’s first turn) and then rapidly decrease as the game progresses to the end.

### 4.9 Win/Loss/Draw Ratios

It is useful to understand if a game is balanced by looking at a game’s win/loss/draw ratios. For small $N$, we can analyse the entire game tree to calculate the percentage of wins, losses, and draws. The Win %, Tie %, and Loss % columns in Table 7 show the results of exploring all possible games of Catch-Up($\{1, ..., N\}$) from $N = 3$ to $N = 18$. As explained in Section 2.6, tie games are impossible in games where $T(N)$ is odd, and these tie percentages are indicated as 0.00%. As $N$ increases, the chance of a random game ending in a draw decreases, which suggests that the games are not too ‘drawish’. We also note the games are balanced between $P_1$ and $P_2$, suggesting that there is no inherent advantage in going first or second if not playing optimally.

### 4.10 Solutions that Lead to a Draw

For Catch-Up($\{1, ..., N\}$), it is also possible to enumerate the moves which will lead to a draw by finding the assignment of positive and negative numbers using the following equation:

$$1 \pm 2 \pm ... \pm N = 0$$

(2)

Assigning positive numbers to $P_1$ and negative numbers to $P_2$ gives us all possible solutions that lead to a draw.

The number of unique assignments of plus and minus for large $N$ can be calculated using a generating function $[14$, Seqs. A063865, A058377], which was first discovered by Euler and has been shown to have an asymptotic upper bound of $\sqrt{6/\pi} * N^{-3/2} * 2^N$.

One way to solve this is to find strict partitions of $T(N)/2$. Strict partitions are sets of non-repeating integers that add up to a given sum; for example, a strict partition of 10 is $\{2, 3, 5\}$ or $\{1, 3, 6\}$ but not $\{1, 1, 3, 5\}$ or $\{2, 2, 2, 2, 2\}$. We can use a strict partition to give the unique integers that can sum to half the total score for the game, which is the condition for a draw. One can generate partitions $[16]$ and then remove the ones with repeated integers, or generate them directly using generating functions $[14$, Seq. A000009].

Every solution to Equation 2 can be reached by having each player choose the smallest numbers in their assigned partition until their turns end, although the resulting draw by this method is likely not to be the result of optimal play.

### 5 Conclusion

One of the most interesting properties of Catch-Up is the complexity of the game tree, given its minimal game rules. This makes it challenging for players to calculate optimal moves by backward induction and adversarial search, necessitating the use of heuristics to play the game. Catch-Up rules not only can encourage drama and tension in games, but they also have interesting mathematically emergent properties. We hope this analysis provides game designers with alternative ways of thinking about using catch-up mechanisms in their own games.

In two-player or multiplayer games, the players’ current scores often provide a clue as to how well they are doing in comparison to the other players. One interesting aspect of Catch-Up is that until the last few moves, the scores do not provide this information because the lead switches on every turn (except for ties). Thus, players need to generate other methods of evaluating the state of the game so that they can tell if they are ahead or behind, but these are not obvious in a game like Catch-Up. Players accustomed to treating current scores as an indication of who is winning may find this to be an interesting feature, or an unpleasant surprise.

We believe that our work on Catch-Up offers lessons that might help guide designers when constructing their own games:
• The structure and form of the pieces can greatly change how a game is perceived. The accessibility of the physical version facilitates play because it does not require the players to keep track of their scores.

• Catch-up mechanisms are intended to keep players feeling that the game is close. In combinatorial games, however, it can be disguising the actual state of the game and the likelihood of each player to win, lose, or draw.

• The starting conditions for a game – the set $S$ in Catch-Up – can have a huge impact on the solution space and play dynamics.

• Simple heuristics are easy to implement in software and can help determine if new players can successfully compete.

• Proving if a game has good characteristics is often significantly more difficult than simulating them; yet much can still be learned from simulating game play.

We conclude by posing several open questions for future study:

• What sets of numbers $S$ are most enjoyable for players?

• How do repeated or non-consecutive numbers in $S$ change the game and the properties we have analysed?

• Is it possible to prove our conjecture that optimal play always leads to a draw in even-sum games of Catch-Up($\{1, \ldots, N\}$)?

• What is the analytical bound for state-space size? Is there a better upper bound for the branching factor?

• Is multiplayer Catch-Up, where the player with the lowest score goes next, a playable game with interesting properties?

• Which heuristics are most effective across different sets $S$ in Catch-Up?

• What are the most interesting ways to break ties? For example, one could break a tie by comparing excess sums, calculated by summing the leads that players achieved on their turns. By not allowing ties, one can do a Nim-like analysis of Catch-Up by treating it as an impartial game, using the Sprague-Grundy theorem [17].

• What happens if one of the players starts with a non-zero score? One could start $P_2$ with a positive score, so $P_1$ moves first but starts from behind. This changes the analysis of odd-sum games of Catch-Up($\{1, \ldots, N\}$) such that they may end in draws.

Acknowledgements

Thanks to Dan Gopstein for comments, suggesting the physical version for Catch-Up and proposing a multiplayer variant; to Kaho Abe for help in 3-D printing a version of the game; to Frank Lantz for various suggestions. We would like especially to thank the anonymous reviewers and Editor-in-Chief for their detailed, thoughtful, and helpful corrections, comments, and recommendations, as well as for pointing out several existing games (including Zhang-Qi) that use catch-up mechanisms which we were not aware of. Also thanks to Dan Isaksen for discussions on integer partitioning theory and for ideas on strategies to find an analytical proof for the game value of Catch-Up($\{1, \ldots, N\}$). Part of this research has been done during Mehmet Ismail’s visit to NYU; he would like to thank the Department of Politics for its hospitality.

References


Appendix

5.1 The Existence of Draws

For Catch-Up(\{1, ..., N\}), we prove which games will permit draws and which enforce a winner, based on the value of N.

To begin, the final score of both players in Catch-Up will always add to the sum of all the numbers in S, because the game will only end once all numbers are assigned to either P_1 or P_2. From Equation 1, the total score T(N) for a game with a set S = \{1, ..., N\} is N(N + 1)/2.

The key factor here is to determine if the sum T(N) is even or odd. If T(N) is odd, such that T(N) \mod 2 = 1, then there is no way to partition S into two subsets S_1 and S_2 such that the final scores are equal. If T(N) is even, such that T(N) \mod 2 = 0, then there is a way to assign the numbers such that the players have equal scores at the end.

If we write N = 4n + k, where n \geq 0 and k \in \{1, 2, 3, 4\}, we can determine, for all N, which games will have even and odd sums:

\[
T(4n + k) \mod 2 = (4n + k)(4n + k + 1)/2 \mod 2 = 8n^2 + 4nk + 2n + k^2/2 + k/2 \mod 2 = \frac{k^2 + k}{2} \mod 2
\]

Thus, parity is independent of n, and we can show if it is odd or even for each k \in \{1, 2, 3, 4\}:

- k = 1 : \frac{k^2 + 1}{2} \mod 2 = 1 \text{ (odd)}
- k = 2 : \frac{2^2 + 2}{2} \mod 2 = 1 \text{ (odd)}
- k = 3 : \frac{3^2 + 3}{2} \mod 2 = 0 \text{ (even)}
- k = 4 : \frac{4^2 + 4}{2} \mod 2 = 0 \text{ (even)}

Aaron Isaksen is a PhD Candidate in Computer Science at the NYU Tandon School of Engineering, researching computer-aided game design, and a professional independent video game developer. Address: NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. Email: aisaksen@nyu.edu

Mehmet Ismail is a PhD Candidate in Economics at Maastricht University, researching game theory, decision theory and behavioural economics. Address: Department of Economics, Maastricht University, PO Box 616, 6200 MD Maastricht, The Netherlands. Email: mehmet.s.ismail@gmail.com

Steven J. Brams is Professor of Politics at New York University, whose research interests are game theory, social choice theory and fair division, with application to international relations, voting and elections, literature, history, philosophy and religion. Address: Department of Politics, 19 West 4th St, New York University, NY 10012, USA. Email: steven.brams@nyu.edu

Andy Nealen is Assistant Professor of Computer Science at the NYU Tandon School of Engineering, co-director of the Game Innovation Lab, and co-creator of the video game Osmos. His research interests are in game design and engineering, computer graphics and perceptual science. Address: NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. Email: nealen@nyu.edu


Aaron Isaksen is a PhD Candidate in Computer Science at the NYU Tandon School of Engineering, researching computer-aided game design, and a professional independent video game developer. Address: NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. Email: aisaksen@nyu.edu

Mehmet Ismail is a PhD Candidate in Economics at Maastricht University, researching game theory, decision theory and behavioural economics. Address: Department of Economics, Maastricht University, PO Box 616, 6200 MD Maastricht, The Netherlands. Email: mehmet.s.ismail@gmail.com

Steven J. Brams is Professor of Politics at New York University, whose research interests are game theory, social choice theory and fair division, with application to international relations, voting and elections, literature, history, philosophy and religion. Address: Department of Politics, 19 West 4th St, New York University, NY 10012, USA. Email: steven.brams@nyu.edu

Andy Nealen is Assistant Professor of Computer Science at the NYU Tandon School of Engineering, co-director of the Game Innovation Lab, and co-creator of the video game Osmos. His research interests are in game design and engineering, computer graphics and perceptual science. Address: NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. Email: nealen@nyu.edu

Appendix

5.1 The Existence of Draws

For Catch-Up(\{1, ..., N\}), we prove which games will permit draws and which enforce a winner, based on the value of N.

To begin, the final score of both players in Catch-Up will always add to the sum of all the numbers in S, because the game will only end once all numbers are assigned to either P_1 or P_2. From Equation 1, the total score T(N) for a game with a set S = \{1, ..., N\} is N(N + 1)/2.

The key factor here is to determine if the sum T(N) is even or odd. If T(N) is odd, such that T(N) \mod 2 = 1, then there is no way to partition S into two subsets S_1 and S_2 such that the final scores are equal. If T(N) is even, such that T(N) \mod 2 = 0, then there is a way to assign the numbers such that the players have equal scores at the end.

If we write N = 4n + k, where n \geq 0 and k \in \{1, 2, 3, 4\}, we can determine, for all N, which games will have even and odd sums:

\[
T(4n + k) \mod 2 = (4n + k)(4n + k + 1)/2 \mod 2 = 8n^2 + 4nk + 2n + k^2/2 + k/2 \mod 2 = \frac{k^2 + k}{2} \mod 2
\]

Thus, parity is independent of n, and we can show if it is odd or even for each k \in \{1, 2, 3, 4\}:

- k = 1 : \frac{k^2 + 1}{2} \mod 2 = 1 \text{ (odd)}
- k = 2 : \frac{2^2 + 2}{2} \mod 2 = 1 \text{ (odd)}
- k = 3 : \frac{3^2 + 3}{2} \mod 2 = 0 \text{ (even)}
- k = 4 : \frac{4^2 + 4}{2} \mod 2 = 0 \text{ (even)}
Therefore, games of the form $N = 4n + 1$ and $4n + 2$ will always have a winner by at least one point, and games of the form $N = 4n + 3$ or $4n + 4$ can, but are not required to, end in a draw.

5.2 Maximum Selections per Turn

We prove the claims of Section 4.7 to calculate $K$, the maximum number of numbers that can be selected on a turn. On the first turn, no matter what set $S$ is, $P_1$ can only select one number. $P_2$ can then choose from $N - 1$ numbers.

From Section 4.3, we know that the greatest sum of numbers that can be earned on a turn is $2N - 2$; before the last number is selected on the turn, the sum of points earned can be no more than $N - 1$. We want to find the maximum number of selections that sum to $N - 1$, and then add 1 for the final selection that ends the turn.

The maximum number of selections occurs if the player selects $\{1, 2, \ldots, k\}$ such that the sum $1 + 2 + \ldots + k$ is as large as possible while still $\leq N - 1$:

$$1 + 2 + \ldots + k = \frac{k(k+1)}{2} \leq N - 1$$

This is quadratic in $k$, so we can use the quadratic formula with $a = 1, b = 1, c = -2N + 2$ to find the positive $k$ that maximises the sum. Adding 1 for the final number that takes the sum to $\geq N$ to end the turn, we have the maximum number of selections on a turn $K = k + 1$ as:

$$K = \lfloor \sqrt{2N - 7/4} + 1/2 \rfloor$$

which is $O(\sqrt{N})$ since $K < \sqrt{2N} + 1/2$.

5.3 Branching Factor (Derivation)

We find an upper bound for the maximum branching factor $B_{\text{max}}$ for Catch-Up(\{1, ..., $N$\}) as follows:

On the first turn, $P_1$ can only select one number, so the branching factor for turn 1 is $N$. In general, when playing with any set $S$, the first-turn branching factor is $|S|$.

For the remaining turns, we can calculate an upper bound $B_{\text{max}}$ for the maximum branching factor for Catch-Up(\{1, ..., $N$\}) based on the results of Section 4.7. A player can select at least one number and at most $K$ numbers on a turn, and those numbers can be permuted except for the final one selected, so we have an upper bound for the branching factor as:

$$B_{\text{max}} < K \sum_{i=1}^K \binom{N - 1}{i}(i-1)! < \sum_{i=1}^K \binom{N}{i}!$$

Because $\binom{N}{i}! = \frac{N!}{(N-i)!} = \frac{N!}{(N-i)!} \leq N!$ and generally $K < N/2$ since $K = O(\sqrt{N})$, we have $\binom{N}{i} \leq \binom{N}{K}$ for $i \leq K$, and therefore:

$$B_{\text{max}} < \sum_{i=1}^K N^i \leq \sum_{i=1}^K K^i < K N^K$$

Thus, we have the final upper bound for the maximum branching factor:

$$B_{\text{max}} < (\sqrt{2N} + 1/2) N^{(\sqrt{2N} + 1/2)}$$

$$B_{\text{max}} = O\left(N^{\sqrt{2N}+1}\right)$$

Try Challenges #7 and #8

Fill the grid with numbers 1 to 7, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.
Coalition Control Through Forced Betrayal

Cameron Browne, Queensland University of Technology (QUT)

Multiplayer games often suffer from the problem of non-strategic coalitions between players. This short note explores the use of an explicit revenge rule to counteract this problem, and finds it wanting, whereas an explicit betrayal rule appears to offer more potential for reducing coalition effects. These results are demonstrated on two hypothetical games.

1 Introduction

Games with more than two players can have inherent problems due to non-strategic coalitions that might exist between players outside the context of the game, even pure strategy games with no hidden information and no chance elements. Three-player games tend to be especially prone to such problems, as there is not an even number of players to form stable teams.

Two common problems are the kingmaker effect [1], in which a player with no hope of winning is able to decide which opponent will be the eventual winner, and the petty diplomacy problem [11], in which two players tend to form an alliance at the expense of the third player or victim. Strategic coalitions can be beneficial for some games, as a balancing mechanism in which trailing players cooperate against the leader to prolong the contest, but non-strategic coalitions are generally detrimental to three-player games.

1.1 McCarthy’s Revenge Rule

McCarthy’s revenge rule is a metarule intended to reduce the kingmaker effect in what Straffin calls three-person winner-take-all games [3]. It can be expressed as:

If you are prevented from winning by a double-crosser, try to take the double-crosser with you. [4, p. 159]

or:

If I find myself in a situation where I can no longer win but must choose which other player will win, I will look back to see who has put me in this undesirable situation and choose to make that player lose. [3, p. 390]

This rule was proposed by computer scientist John McCarthy while playing So Long Sucker, ‘a game of negotiation, alliances and backstabbing’ [9], with its co-inventors including John Nash. This rule makes intuitive sense, and has an impressive heritage – McCarthy was one of the founders of artificial intelligence and Nash famously one of the founders of game theory and modern economics – so why does this rule not feature in the rule sets of more multiplayer games?

I believe that this is due to two main reasons. First, it was posed as a principle for rational players with no external agenda beyond the game, and this ideal situation rarely occurs in practice. Straffin’s subsequent analysis even states that it does not take coalitions between players into account [3]. Second, it could simply be too difficult to enforce, as it may not always be clear which opponent is most to blame for an unfavourable position.

This paper proposes a simple development of this idea, to encode the revenge principle explicitly in the rules, so that players are forced to follow this principle regardless of their personal agendas, as a way of reducing the effect of non-strategic coalitions in multiplayer games.

2 Forced Revenge

There is an inherent logical problem with applying McCarthy’s revenge rule directly to a rule set: if two players have formed a non-strategic coalition, then neither will want to hurt the other the most. This could actually make the coalition problem worse, as such a revenge rule would force the allies to act even more explicitly against the victim, who is fighting them both for dear life.

For example, consider the following hypothetical game called Revenge, invented for this exercise. The degree to which players hurt each other is indicated by the number of pieces captured. In order to test this game’s robustness to coalition effects, let us assume that White and Black are allies who have prearranged a non-strategic coalition against the victim Red. [1]

---

1 Coalitions between players that satisfy some personal agenda rather than being of strategic benefit to either.

2 For black-and-white readers: White = light, Red = medium and Black = dark.

Revenge is played on a hexagonal grid, with pieces in three colours, as shown in Figure 1(a).

Three players (White, Red and Black), take turns moving one of their pieces to an adjacent cell, which can be empty or occupied by an enemy piece, which is captured and removed.

If the mover can capture a piece belonging to the opponent who has hurt them more than the other opponent (i.e. captured more of their pieces) then such a capture must be made. The last surviving player wins.

Figure 1 shows three moves in a game of Revenge. White’s first move 1 captures a red piece, which is then listed in White’s capture pile below the board (b). Red is then forced to capture a white piece due to the revenge rule, as White has captured more red pieces than Black; Red makes capturing move 2 (c). Black is free to make any move, and eliminates Red with move 3 (d).

The victim was eliminated by the allies almost immediately. The forced revenge rule offers little protection against coalitions.

3 Forced Betrayal

Now consider the following game, called Betrayal, in which the revenge rule is reversed, so that players are forced to hurt the opponent who has hurt them the least.

Betrayal is played on a hexagonal grid, with pieces in three colours, as shown in Figure 2(a).

Three players (White, Red and Black), take turns moving one of their pieces to an adjacent cell, which can be empty or occupied by an enemy piece, which is captured and removed.

If the mover can capture a piece belonging to the opponent who has hurt them less than the other opponent (i.e. captured fewer of their pieces) then that capture must be made. The last surviving player wins.

Figure 2 shows five moves in a game of Betrayal. White again starts by capturing a red piece with move 1 (b), but this time Red is forced to reply by capturing a black piece with move 2 (c), as Black has captured fewer red pieces than White. Black is then forced to attack their ally and capture a white piece with move 3 (d), as White has captured fewer black pieces than Red.

White then steps adjacent to the red piece with move 4 (e). This move is surprising, but is the best that White can do; any other move would give victory to Red (left as an exercise). Red eliminates White with move 5 (f), then the game is drawn, as both remaining players can elude capture indefinitely. The forced betrayal rule counteracted the White/Black coalition, by forcing one ally to turn against the other.

Figure 2. In a game of Betrayal, the victim (Red) can elude defeat if one ally is forced to attack the other.

3 Including any opponent eliminated from the game.
4 Discussion

While these two example games may not be masterpieces of design, and are demonstrated on trivially small boards, they still serve to highlight the key point: that forced revenge can exacerbate coalition effects, while forced betrayal can effectively counteract coalition effects, by forcing allies to act against each other. This includes both non-strategic coalitions and the kingmaker problem, as the betrayal rule appears to remove from players some of the freedom to collude.

The notion of ‘hurting’ an opponent can be a slippery one to define. For example, in the Betrayal game shown in Figure 2, White superficially hurts Red by capturing a red piece on the first move, but this ultimately works out to Red’s advantage; White could even be said to ‘help’ Red win with this move. However, rule sets need to be precisely defined, and the best we can do here is define ‘hurt’ to mean immediate material loss. It is in fact this discrepancy, between the immediate result imposed by the rules and the ultimate result of the game, that allows the betrayal mechanism to work.

Some games have explicit rules to deal with alliances. For example, the three-player version of Shogi called Sannin Shogi [6] involves explicit alliances between players, which may be stated voluntarily before the game or formed automatically during play, when two players attack the third. Alliances trigger a number of additional rules to come into play, in order to balance the game more fairly. The betrayal rule, on the other hand, is an attempt to implicitly dampen alliance effects seamlessly within the rules of normal play.

Forced betrayal does have some downsides. The need to explicitly define the degree to which each player has hurt each other player requires some bookkeeping and might be confusing for players; just look at the apparent complexity of the otherwise trivial games shown above. Also, being forced to betray otherwise benign opponents may be distasteful to some players. This mechanism, if used, would need to be carefully implemented and presented to players.

5 Conclusion

Forced betrayal is an under-utilised mechanism that offers a potential way to control coalition effects in multiplayer strategy games. The fact that it was so easy to invent an example, which shows its benefits so clearly, suggests that it might have practical application for more fully formed games in future.

Acknowledgements

Thanks to Don Kirkby for suggesting the Sannin Shogi example. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

References


https://pediapress.com/books/show/chess-variants-by-wikipedians/

Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au
From Mathematical Insight to Strategy

Néstor Romeral Andrés, Nestorgames

This short note describes how the design process of a group-making game exploited a mathematical insight, based on the constant e, to allow a more intuitive strategy for players.

1 Introduction

Omega is a strategy board game invented in 2010 in which players strive to maximise their points using multiplicative scoring. The rules are as follows.

- Omega is played by \( P = 2 \) to 4 players, on a hexagonal tiling of hexagons, which starts the game empty.
- 1. Players take turns placing one stone of each colour at any empty cell, i.e. \( P \) stones must be played each turn.
- 2. The game ends when a full round of moves by all players can not be completed (some cells may remain empty). All players, therefore, have the same number of moves.
- 3. Each player’s score is the product of the size of each group of their colour. The player with the highest score wins.

\[ \text{White: } 1 \times 2 \times 2 \times 3 \times 4 = 48 \]
\[ \text{Black: } 1 \times 4 \times 7 = 28 \]
\[ \text{Red: } 1 \times 2 \times 4 \times 5 = 40 \]
\[ \text{Blue: } 1 \times 2 \times 3 \times 6 = 36 \]

1.1 Design Goals

The primary design goal for Omega was to create a game with a balanced winning condition. Instead of players trying to create the most groups or the least groups, or the biggest group or the smallest group, I wanted players to strive for something ‘in between’ in order to win. This would theoretically allow more subtle strategies. Furthermore, I wanted to make players also play the opponents’ stones each turn, so that they did not focus solely on their own formations.

Group creation and multiplicative scoring are not novel mechanisms, but Omega expands on these ideas as follows:

- Fixed number of stones of each colour.
- The number of groups to be multiplied is variable and undefined at the start.

2 Maximising the Score

Given \( n \) stones split into two groups, the strategy to maximise their multiplicative score is simply to make both groups equal in size. This is easily demonstrated as follows.

Let \( a \) and \( b \) be the number of stones of each group, hence \( a = n - b \). Maximising the score \( a \cdot b \) is, therefore, equivalent to maximising \( (n - b) \cdot b = n \cdot b - b^2 \). Taking the derivative and equating with 0, we have \( n - 2 \cdot b = 0 \), which gives \( b = \frac{n}{2} \) and \( a = \frac{n}{2} \).

Recall, however, that the design goal was to force players to create several balanced groups. The question then becomes: is there an optimum group size for more than two groups?
2.1 Optimum Group Size

In August 2010, programmer Greg Schmidt implemented Omega for his AXIOM general game system[^1] and observed that the optimal group size appeared to be 3, regardless of how many groups are created.

To check this mathematically, consider the following: \( n \) stones placed in groups of size \( x \) will give \( \frac{n}{x} \) groups with a total score of \( x^\frac{n}{x} \), and the optimum group size will be the point at which the derivative of this formulation equals zero. This can be derived as follows.

Maximise \( x^\frac{n}{x} \), where \( n \) is the number of items and \( x \) is the optimum subgroup size.

\[
d\frac{x^n}{dx} = 0 \quad (1)
\]

\[
-\frac{nx^n \log x - nx^n}{x^2} = 0 \quad (2)
\]

\[
-\frac{nx^n \log x + nx^n}{x^2} = 0 \quad (3)
\]

\[
\log x = 1 \quad (4)
\]

\[
x = e \quad (5)
\]

The optimum group size turns out to be the base of the natural logarithm, \( e \), which is \( \approx 2.718 \) and is the limit of:

\[
e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n \quad (6)
\]

This value must be discretised, as only whole stones can be played, and rounding \( e \) to the nearest integer does indeed give an optimum group size of 3. If \( n \) is not exactly divisible by 3, then try to make each group as close to 3 as possible.

It turns out to be a well known fact in the mathematical community that 3 is the optimum item size to achieve a maximum product. For example, Problem 1.1.4 in Larson’s Problem-Solving Through Problems gives a similar result through a different route[^2] p. 7). But I still find it intriguing that a mathematical constant so closely related to infinity lies at the core of a game based on finite integer multiplication, and it pleases me that the central letter in the name ‘Omega’ is ‘e’.

3 Impact on Strategy

This insight was a godsend for Omega. The need to multiply often large numbers of group sizes each turn, in order to plan moves, had initially made the game intractably opaque for some players. However, the insight that the optimum group size is 3 provided a simple and intuitive strategy for players to follow: make your group sizes as close to 3 as you can, and make your opponents’ group sizes as different from 3 as you can.

The fact that players place a stone of each colour per turn works well with this strategy. Further, it brings out a strong connective aspect in Omega, as the best way to defuse enemy groups of 3 is to join them together into larger, less efficient groups, and the best way to protect groups of 3 is to disconnect them from nearby friendly groups with walls of enemy pieces.

Figure 2 shows a completed game of three-player size-4 Omega, in which all three players have formed groups of 3 to achieve the maximum score of \( 3 \times 3 \times 3 \times 3 = 81 \).

![Figure 2. A completed three-player game.](http://www.boardgamegeek.com/thread/563815)


Acknowledgements

Thanks to the Editor-in-Chief for help writing this note and providing the figures.

References


Néstor Romeral Andrés is the owner and director of independent board game publisher Nestorgames. He has designed over 70 board games and was awarded Spanish Game Designer of the Year for 2009–10. Address: c/Mar de las Antillas 6, 3-2, Zaragoza, 50012, Spain. Email: info@nestorgames.com

[^1]: http://www.boardgamegeek.com/thread/563815
Deriving Card Games from Mathematical Games

Daniel Ashlock, University of Guelph
Justin Schonfeld, University of Guelph

In a mathematical game, such as the Prisoner’s Dilemma or Rock-Paper-Scissors, players are permitted to choose from the available moves each turn. When that free choice of moves is replaced by cards with moves printed on them from a fixed deck, the mathematical game becomes a card game. This transformation often yields games of very different character, giving a simple technique for generating card-game mechanics from existing – and novel – mathematical games. We explore ways to translate games’ payoff matrices into scoring systems, and an evolutionary algorithm to assess their resulting balance.

1 Introduction

MATHEMATICAL games are a staple of economics, planning, and academic game theory. Possibly the best known of these games is the Prisoner’s Dilemma (PD) [1], invented by Melvin Dresher and Merrill Flood of the RAND Corporation, and placed in a prison setting by mathematician Albert Tucker in 1950. The game is used to model cooperation and conflict and, in its single-shot version, shows why rational agents might avoid cooperating.

The mechanics of the PD are simple. Two agents choose, simultaneously, to cooperate or defect. If both players cooperate, they receive the cooperation payoff C. If both players defect, they receive the defection payoff D. If the agents’ moves are not the same, then the cooperator receives the sucker payoff S and the defector receives the temptation payoff T. In order to be considered PD, these payoffs must obey the following inequalities:

\[ S < D < C < T \] (1)

and:

\[ S + T < 2C \] (2)

If the game is iterated, i.e. repeated, and if the number of rounds is not known to the participants, then this game can be used to model how cooperation arises between rational agents. Its payoff matrix can be represented as a directed graph, whose vertices represent available moves and whose edges are weighted with the score that the move at the tail of the arrow yields, when played against the move at the head of the arrow.

Figure 1 shows this digraph for PD. States are labelled with moves and edges are annotated with scores for the tail of the arrow playing against the head. For two-move games, there is relatively little advantage to this technique, and this figure is included for illustrative purposes. The digraph becomes more useful when a game has more moves.

In this article, we examine the impact of placing a restriction on which moves can be made in the form of hands of cards with moves printed on them. This mechanic permits mathematical games to be reincarnated as card games. In addition, transferring the moves to cards makes the implementation of variations and generalisations transparent, as special cards are added to the deck.

This study also outlines a method for exploring the dynamics of a game, and measuring its degree of balance, using a simple evolutionary algorithm. This article specifies tools that have proven useful in academic research in hopes that some or all will also prove useful for game design.

2 Deck-Based PDs

When a mathematical game is transformed into a card game by placing its moves on a limited set of cards, the new game will be called the deck-based version of the mathematical game. Thus a deck-based Prisoner’s Dilemma (DBPD) would use a deck with Cs and Ds printed on the cards, and [2] explores a first attempt in this direction. If each player is given only two cards (C and D) and the game only lasts a single round then we have simply recreated the original mathematical game.
However, if we give each player more cards and make the game last multiple rounds, then we see novel dynamics develop. This transformation resulted in three new games, one trivial, and none of which were anything like the original game.

The game was split as follows. If \( T + S > C + D \), then the game is a discoordination game in which the best possible move is to play so that you do not match the opponent’s move. If \( T + S < C + D \), then the game becomes a coordination game in which it is desirable to match the opponent’s move. The trivial version of PD occurs when \( T + S = C + D \). In this case, a player’s score does not depend on the order in which he plays his cards.

Given this transformation of the original game into very different deck-based games, there is clearly potential in examining the deck-based versions of mathematical games. We also need a tool for spotting when different payoff matrices make substantially different games.

2.1 Deck-Based Game Mechanics

A deck-based game is **strict** if a player must use all the cards in his hand. The original version of a mathematical game is called the **open** version of the game. It is possible to transition smoothly from a strict to an open version of a given game by letting the hands grow larger than the number of rounds played. If a player of deck-based PD has, for example, more \( C \) and more \( D \) cards each in his hand than there are rounds in the game, then the game is effectively an open one.

A deck-based game based on a simultaneous mathematical game, in which the players must choose their move without knowledge of the opponent’s move, may be played by any number of players as follows. All players reveal their cards simultaneously, then each player is awarded the sum of the scores received against the moves on the cards revealed by all the other players.

A **hand** is the set of cards or actions currently available to the player. The **deck** is the total set of cards available in the game. A deck can be **shared**, as in traditional card games such as Poker and Blackjack, in which all players are getting cards for their hands from the same deck.

Alternatively, each player can have an individual deck of cards which only they have access to, as in Magic: The Gathering. The **hand management** mechanic refers to situations in which players are rewarded for playing cards in a specific order or group. The payoff for a card depends on the game state, not only the actual cards.

A game can be **cooperative** if all the players are working together or **competitive** if the players are playing against one another. If the players are choosing cards from a limited (and often shared) subset, this is called **card drafting**. If the players are constructing their individual decks as play proceeds, usually through a drafting mechanic, this is referred to as **deck building**.

3 Deck-Based Game Types

This section motivates the need for tools for card game design by discussing some modern games and their deck-based mechanics. Recently, there has been a surge in the diversity of card games. The BoardGameGeek online database lists over 17,000 different games with cards as the primary or major mechanic in the ‘Card Game’ category.

The mathematical analysis of card games, especially betting games, has a long and storied history, with Beat the Dealer by Edward O. Thorp being one prominent example of mathematical analysis applied to a deck-based game. We present here an incomplete list of deck-based game types, with representative examples.

3.1 Collectible Card Games

A **collectible card game** (CCG) is a game in which the cards are produced in varying frequencies and distributed in randomised packs to the players. The players then construct decks from a subset of the total set of cards. Most CCGs are turn-based with players playing from individual decks. There are often strict constraints about the number and type of cards that can be used from the total card set in a deck. One of the earliest and most famous examples is Magic: The Gathering, with an estimated 20 million or more players.

3.2 Living Card Games

**Living card games** are a more recent innovation in which the cards are distributed in a piece-meal fashion but with each pack containing an identical set of cards. The idea is to allow players the experience of customising a deck from a large set of cards but without the financial cost that comes with purchasing randomised packs of cards. Examples include Warhammer Invasion: The Card Game, A Game of Thrones: The Card Game and Star Wars: The Card Game, all from Fantasy Flight Games.

---

1. http://magic.wizards.com
2. https://boardgamegeek.com
3.3 Resource Management

With the explosion of ‘Eurogames’ or ‘German games’, there has been a dramatic rise in the number of board games that include cards, hand management, and deck building as substantial elements. Many of these games share common roots with mathematical games and challenge the players to make decisions with imperfect information. Two prominent examples are the deck-based games Lost Cities $^4$ and Battle Line $^5$, both developed by mathematician Reiner Knizia. In both games, the players share a deck of cards and are forced to play cards from their hands with access to only partial information: cards played, cards in their hand, and knowledge of the total set of cards in the shared deck.

Many turn-based card games, such as Drag-onheart $^6$, give each player an identical deck of cards to play with, each card representing an action, but not all actions equally represented.

3.4 Simultaneous Play

Although turn-based card games make up the majority, there are many simultaneous-play deck-based card games. The game Yomi $^7$, for example, simulates a 2-dimensional arcade fighting game, but with mechanics very similar to Rock-Paper-Scissors. Each player has a custom deck of cards from which they draw their hands. Players each pick a card from their hands and then simultaneously reveal them. Attack actions beat throws, blocks beat attacks, and throws beat blocks. The game contains several special cards and mechanics that modify this core principle, but the Rock-Paper-Scissors roots remain at the core of the gameplay.

Other simultaneous-play games, such as 7 Wonders $^8$ in which players compete to build the best civilisation – work from a shared deck. Each player takes a hand of cards from the deck, picks one to play, and then passes the rest of their hand to the player to their left. In the next turn, they must pick from the reduced set of cards that was passed to them.

3.5 Cooperative/Competitive Play

A few games, such as The Resistance $^9$ and Saboteur $^{10}$ combine cooperative and competitive game mechanics. These games assign roles to the players, either as a cooperator or a defector, usually with the cooperators outnumbering the defectors. The defectors must then choose the right time to cooperate and the right time to defect so as to prevent the cooperating players from achieving their objective. Such games can be considered special cases of the Iterated Prisoner’s Dilemma (IPD), in which the cooperators have a deck with only cooperative actions while the defectors have both cooperative and defecting actions to work with.

3.6 Dynamic Rules

An important and relatively early turn-based game is Fluxx $^{11}$. The unique feature of Fluxx is that many of the cards modify the rules of the game, changing both mechanics, such as permitted hand sizes, and victory conditions. A game that permits on-the-fly modification of its rules is more complex and potentially more challenging.

4 Digraphs as Design Tools

The digraph in Figure 1 is annotated with the score values. In many games, the score is the number of victories, and so an unweighted digraph with winners pointing to losers may be used to diagram the game. Examples of these digraphs for the games Rock-Paper-Scissors (RPS) and Rock-Paper-Scissors-Lizard-Spock (RPSLK) appear in Figure 2.

---

$^4$http://riograndegames.com/Game/126-Lost-Cities
$^7$http://www.sirlingames.com/pages/yomi
$^8$http://www.rprod.com/?page=description-22
$^9$http://www.indieboardsandcards.com/resistance.php
$^{10}$https://boardgamegeek.com/boardgame/9220/saboteur
$^{11}$http://www.looneylabs.com/games/fluxx
For these games, three scores need to be specified – the scores for Win, Lose, and Tie – with the obvious constraint \( L \leq T \leq W \) and the added condition that if all three are equal, then the game is rather pointless. The absence of an arrow indicates no victory or loss. For example, there is no arrow from rock to itself.

Examine the digraphs in Figure 2. The digraphs show that the games are intransitive – there is no dominant strategy because each move is in a directed cycle. The digraph also gives us a sense of a way in which the intransitivity of RP-SLK is even more pronounced than that of RPS. Each move is in two different directed cycles that include all the possible moves and which are oriented in opposing directions. The next natural question is how digraphs can be used in the design of deck-based games.

Rock-Paper-Scissors-Dynamite (RPSD) adds the move dynamite to standard RPS. The move dynamite beats every other move. The digraph model of this game, shown in Figure 3, motivates the use of a simple statistic to inform the design of a card game.

The **indegree** of a node in a digraph is the number of incoming arrows – in the digraph notation, the number of moves that beat the move labelling the node. The **outdegree** of a node is the number of arrows leaving it – the number of moves that the move labelling the node can beat. If we diagram the moves in the form Move(indegree:outdegree), then RPSD has the diagram \( R(2:1), P(2:1), S(2:1), D(0:3) \). The **local balance** of a move is the ratio of its indegree to outdegree. The lower the local balance of a move, the more powerful it is - correspondingly the higher the local balance, the weaker the move.

In RPSD, dynamite has a local balance of zero, it is unbeatable. A node with an outdegree of zero cannot beat anything and so would have a local balance of \( \infty \) (the informal result of dividing by zero). This would be a worthless move. These concerns lead to the following rules of thumb for approximately balancing a deck-based game:

- Moves with a local balance of zero are unbeatable. This suggests they should be very rare, like jokers in a standard deck of playing cards. This is true in simultaneous or turn-based games.
- Moves with a balance of (or near) one are ‘regular’ cards. Cards of this sort should make up the bulk of the deck.
- Moves with a large-to-infinite balance are worthless or nearly worthless, so should either be avoided or be cards for dealing with special rare situations.
- Strategy is strongly influenced by the structure of the cycles. RPS has a cycle in one direction, RPSLK has cycles in two directions. In a turn-based game, the relationship of the lengths of the cycles to the number of players should be considered. If cycle lengths are larger than the number of players, it becomes possible for the last player in a round to select either end of a cycle, while shorter cycles allow every component move to be played.
- When a digraph has edges labelled with scores, or differences of scores, compute the indegree and outdegree of a node by summing the incoming and outgoing scores, respectively.
- Many mathematical games are defined by inequalities. The digraph induces (or encodes) some of these in the following fashion. The payoff for a move at the base of an arrow must be at least as good as the one at its head.

It is important that moves do not have scores; each move has a potentially different score for each other move made against it, which avoids problems with intransitivity. In RPS, for example, all three moves can reasonably be scored as one for the action at the tail and zero for the action at the head.

- In a turn-based game, it is good if moves with in- or out-degrees of 1 are rare. These represent moves in which subsequent players have only one choice of move unless they want to make a play that does not interact with the initial player’s move. Higher degrees represent more potential choices for subsequent players.

## 5 Evolutionary Methods

A common bottleneck in game development is the difficulty of performing sufficient play-testing. Human play-testers are individually superior to software agents, but software agents can be deployed in vast hordes at relatively low cost. This
study uses an evolutionary algorithm to train game-playing agents to a better-than-random level of competence and then applies these agents to test for game balance and interest.

5.1 Evolving Intelligent Agents

The flow-chart shown in Figure 4 describes the algorithm we use for evolving intelligent computer players. This figure shows that an elite set of agents for a given run are the 24 agents that had the highest score, so their superiority is purely relative to the population they evolved in.

**Figure 4.** Flowchart for an evolutionary run.

The algorithm is run 30 times, with different random number seeds, to locate a sample of strategies for the game. The agents consist of orders in which to play a deck of cards. When new agents are created by modifying copies of parents, a pair of cards are exchanged, modifying the order of play. This is an extremely simple algorithm, based on one used in several previous studies of game playing agents [2, 4, 5].

The term *elite agents* refers to the current best in a co-evolving population of agents, not agents that are elite in a global or general sense. The currently best agents are, in essence, examples enriched with some competence at playing the game, and so provide a competence-biased sample of play strategies. Note that this algorithm assumes a multi-player game; changing the number of players from 36 to two would permit evaluation of the two-player version.

5.2 Evolutionary Assessment

If we are designing a card game from a mathematical game specified by a digraph, then the simple, obvious payoff matrix is that victory gets one point and anything else gets zero. This makes an agent’s score equal to the number of victories it achieves. The payoff matrix, however, is an excellent target for the designer to manipulate the character of the game and even induce new strategies and mechanics. Doing this requires the ability to detect different strategy regions like the coordinated and dis coordinated games that arose in deck-based IPD.

In the initial work on DBPD, a simple type of evolutionary algorithm was used to detect the play character of different payoff matrices. The game was multi-player, with each agent playing one card in each round, and each player was given a hand with an equal number of cards for each move, e.g. 10 cards, 5 cooperates, 5 defects. The algorithm evolves the order in which each player plays their cards, and potential players might look like the following:

**Player 1:** CCCDDCDDD
**Player 2:** CDDDCDCDCC
**Player 3:** DDDDDCCCCC

The payoffs in this example are implicitly negated, i.e. players want to minimise their pay-outs, which represent years in prison: two cooperators each earn 1 year in prison, while two defectors each earn 2 years in prison, but a cooper ator and defector earn 3 and 1 years respectively. After 10 rounds of play, each player has received a total of 30 years in prison, as shown in Table 1.

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
<th>Player 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C (1+3)</td>
<td>C (1+3)</td>
<td>D (0+0)</td>
</tr>
<tr>
<td>C (3+3)</td>
<td>D (0+2)</td>
<td>D (0+2)</td>
</tr>
<tr>
<td>C (3+3)</td>
<td>D (0+2)</td>
<td>D (0+2)</td>
</tr>
<tr>
<td>D (2+2)</td>
<td>D (2+2)</td>
<td>D (2+2)</td>
</tr>
<tr>
<td>D (0+2)</td>
<td>C (3+3)</td>
<td>D (2+0)</td>
</tr>
<tr>
<td>C (3+1)</td>
<td>D (0+0)</td>
<td>C (1+3)</td>
</tr>
<tr>
<td>C (1+1)</td>
<td>C (1+1)</td>
<td>C (1+1)</td>
</tr>
<tr>
<td>D (2+0)</td>
<td>D (2+0)</td>
<td>C (3+3)</td>
</tr>
<tr>
<td>D (0+0)</td>
<td>C (3+1)</td>
<td>C (3+1)</td>
</tr>
<tr>
<td>D (0+0)</td>
<td>C (3+1)</td>
<td>C (3+1)</td>
</tr>
</tbody>
</table>

**Table 1.** Prison time received after 10 rounds.

The age of an agent is the number of generations since the agent was created. The maximum age in a given generation is a measure of the rate at which new types of agents take over the game. Figure 5 shows a plot of the maximum agent age during the first 480 generations of evolution for a
population of hands, for playing deck-based versions of RPS and PD. Each set of agents is evolving the play order of hands of 60 cards divided evenly among the types of moves available.

The agents playing PD survive to substantially greater ages than for those playing RPS. This means the rate at which new agent types are arising and taking over the population is much higher for RPS than PD.

The maximum-age-over-evolution behaviour of a game can be used to estimate its balance. Suppose we have an unbalanced game with a single dominant strategy, then when an agent discovers this strategy its age will increase indefinitely.

The plots shown in Figure 5 indicate that no agent has discovered such a dominant strategy in RPS or PD during the first 480 generations. In fact, examining plots of the age over the course of evolution for thirty replicates, for both games, shows that the maximum age statistic does not enter an unrestrained climb in any of them.

This is evidence that either these games do not have a dominant strategy, in the form of a particular order for playing the hand, or that this strategy is hard to locate via evolution. Note that a younger and equally-fit strategy may overwrite an older one, so it is important to break fitness ties in order of age — favouring older players — when using evolution to detect a dominant strategy. With this cautionary note, we claim that evolution is a good first-pass tool for detecting a lack of game balance, resulting from dominant strategies. The effectiveness of evolution as a tool, though, depends on the ability of the agents being evolved to explore the strategy space of the game being balanced. An additional example of evolution used to help balance a game appears in 6, where evolution was used to balance an RPS-like computer game.

The presence of a dominant strategy is not the sole concern. RPS has a very high rate of overturn, in other words, of the emergence of new strategies; PD has a lower rate. The strategy in RPS involves knowing and outguessing the other players — a human rather than rule-based dynamic. In PD, this sort of dynamic is important, but there are deeper strategic issues. This means that the distribution of maximum ages may be a feature that is related to how interesting the strategy of a game is. This will require substantial calibration, and is an area for future work.

5.3 RPS Payoff Matrices

One of the outputs of the evolutionary algorithm is the final population of hands for the elite agents in the last generation. This has the form of a file with rows containing each player’s card order. In the original research on DBPD, these populations made the bifurcation into coordination and discoordination games abundantly clear. In one group, moves were split fairly evenly between C and D in columns of the file, while in the other, columns were entirely one move or the other. Thus, this final generation of play orders can be used to spot different strategic regions in payoff space.

This form of analysis was applied to two different payoff matrices for RPS. The first awarded one point for victory and zero for any other outcome. The second awarded two points for victory, one point for a tie, and zero points for a loss. With a 24-member final population, the cards played in any one round consist of 24 choices of the three possible cards, giving $3^{24}$ or roughly $2.82 \times 10^{11}$ possibilities. It is more useful to examine, in each play of the game, the fraction of cards that are rock, paper, or scissors in that play. Since the fraction of scissors is one minus the fractions of rock and paper, there are essentially two degrees of freedom, so we can perform a two-dimensional frequency plot of the types of plays – based on fractions of card types – for each of two possible payoff matrices. This leads to the visualisation shown in Figure 6.

For each possible fraction of rock and paper moves – leaving the scissors moves as the remainder – Figure 6 shows how often each fraction of possible moves occurred. The leftmost column is for moves with no rock plays at all, and the rightmost column is for all-rock moves. The top row is for moves with no paper plays, and the bottom row is for all-paper moves. The corners of the rectangle, therefore, represent coordinated moves, with only one type of move made. The bars to the left of dividing lines show the frequency of the moves under the first payoff matrix, the bars to the right of the column show the frequency of the moves under the second payoff matrix.
Figure 6. The distribution of joint move types for RPS, with two different sets of payoffs.

The results displayed in Figure 6 make it clear that, while it is a more complex change than was observed for PD, the two different RPS payoff matrices yield very different distributions of cards in individual plays. This shows the two payoff matrices tested are probably in distinct regions of the deck-based RPS strategy space. As noted above, the moves in which all players lay down the same type of card are at the vertices of the plot triangle. This means that, for the (1, 0, 0) payoff, synchronised moves are more likely to occur. This concentration on synchronised moves means that there are many combinations of moves that simply did not occur for the (1, 0, 0) payoff scheme, but which did occur for the (2, 1, 0) payoff scheme. Note the preference for (1, 0, 0) payoffs at the three corner positions in Figure 6; this is due to the fact that making synchronised moves is advantageous.

This strategy for comparing payoffs can be automated to map the full space of possible payoffs by sampling different payoff values and then clustering them. The clusters, sets of points with relatively low mutual distance, would represent the strategic regions to the possible deck-based game. The ability to locate such regions is valuable to implement the new mechanic, payoff modifying cards, described in the next section.

5.4 Fluxx-like Mechanic Payoffs

Both the coordination and discoordination versions of the PD might be somewhat entertaining, but it would be nice to spice it up a bit more. The division between the three possible types of play comes from comparing the two numbers $C + D$ and $S + T$. Suppose that, in addition to being printed with a $C$ or $D$, some cards in the game also changed one or more of the payoff values. Then a player could, by playing such a card, change the version of the game being played.

Since the players do not have access to the mathematical analysis dividing DBPD into different types of games, they would need to discover
that the strategy of the game had shifted. A card that made strategy irrelevant (recall this happens when \( S + T = C + D \)) has the potential to cause consternation, as players discover that there is no strategy until the next payoff change.

This proposed mechanic creates a second game, controlling the values of the payoffs, and is similar to Fluxx, which uses rule-modifying cards. An important difference is that the rule-modifying cards in Fluxx are more explicit about the changes in the rules. It would not be immediately obvious that changing a payoff value changes the optimal strategy (or can cause an absence of strategy). The difference between the two sets of payoff values for RPS shows that cards that change payoff values has the potential for strategy shifts in RPS. It also shows why mapping the different strategic areas in a deck-based game is a potentially valuable game design tool.

6 Further Game Mechanics

Consider the simultaneous game Rock–Paper–Scissors–Lava–Water–Rust (RPSLWT). Rock, paper, and scissors interact in the usual way for RPS, while lava ‘melts’ all rock. Not only does the player that plays lava get a victory for each rock played by another player, they also cause the rock cards to be discarded without interacting with other non-lava cards.

If many scissors were played, this is bad for the players who played rock. If many papers were played, this may actually be good for players who played rock. Similarly, water ‘spoils’ paper and rust ‘ruins’ scissors. These new cards negate another type of card. We use a different type of arrow to denote a negating move. The digraph for RPSLWT is shown in Figure 7.

Since a negating move may be good or bad for the players whose cards are negated, and is only good to the extent that other players choose to play negatable cards, it creates a richer, more complex strategy space.

Modifying RPS, the cards in question are dynamite-like, but less powerful than dynamite, and so a deck might have more of each of these types of cards than would be a wise number for dynamite cards. There are a number of other possible mechanics that might merit different types of arrows. One move could also be the source of multiple different types of arrows.

Figure 8 shows three example hands of RPSLWT. The first hand shows: rock, rock, paper, scissors and lava. The lava scores two because there were two rocks; paper would have scored 2 except that the lava negated the two rock plays and so it scores 0; scissors cuts paper for a score of 1; rock scores 0 because it was negated.

![Figure 8. Three hands of five-player RPSLWT.](image)

The second hand in Figure 8 has two rocks which each score two for the two scissors, two scissors that score zero because there is no paper, and a (completely wasted) water move that scores zero because there is no paper.

The third hand has paper negated by water, and water that scores one for the paper. The two scissors score zero because the paper was negated, and the rock scores two from the two scissors.

7 Conclusion

This paper presented methods for deriving card-game mechanics from mathematical games, and elucidated ways in which shifting to a deck-based game changes the original mathematical game. Diagramming with directed graphs to simplify the identification of design issues was demonstrated, along with the use of a simple evolutionary algorithm to detect resulting balance issues. Potential game mechanics including rule-modifiers and negating moves were given.
This article is part of a series called *New from Old*, in which we investigate ways of deriving new games from existing mechanisms and rule sets. If you develop an interesting game using these ideas, please let us know.

**Acknowledgements**

The authors thank the Natural Sciences and Engineering Council of Canada (NSERC) and the University of Guelph for supporting this work.

**References**


**Try Challenges #9 and #10**

Fill the grid with numbers 1 to 9, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.
Article

Game Mutators for Restricting Play

João Pedro Neto, University of Lisbon

William Taylor, University of Canterbury

When creating a new rule set, game designers usually have one or two main concepts that they wish to translate into actual rules. Sometimes the interactions between rules provide too much scope for offence or defence that remove any sensible or interesting tactics, thus producing a boring or trivial game, even sometimes a broken game that needs redesigning. This article proposes different ways to restrict move flexibility in order to reach an appropriate balance that a good abstract game requires.

1 Introduction

The concept of restricting move options within a game is not new. Many classical games include ludemes that restrict piece movement with the goal of improving overall play. These can be strategic in nature, such as the ko rule in Go, or more tactical in nature, such as the bearing off restrictions in Backgammon.

Such restrictive rules can often be applied more generally to a variety of games, which leads to the concept of game mutators. This article describes, through example, how such mutators can be applied to solve design problems in games, by restricting piece movement to reduce or eliminate undesired behaviour.

1.1 Restricting Cycles in Go

The classic board game Go involves the simple ko rule which prohibits repeating the previous board position. This is usually sufficient to prevent cycles of moves that would otherwise ruin the game, and adds significant strategic depth to the game [1].

However, some unusual situations can still lead to cycles longer than two moves, so there exist stricter superko rules in some Go rule sets. Situational superko prohibits repeating any previous board state with the same player to move, while the even stricter positional superko prohibits repeating any previous board state, regardless of whose turn it is to play.

For example, Figure 1 shows a Go position in which this distinction between situational and positional superko matters, with White to move. If White plays at a, then Black can capture at b. With positional superko, White cannot recapture, so on Black’s next move, Black could capture the remaining two White stones to secure the corner. With situational superko, however, White can immediately recapture b, restoring the board state but with Black to move.

Figure 1. The specific superko rule matters.

1.2 Game Mutators

A game mutator is a rule, or set of connected simpler rules, that transforms a game into a new game. The distinction between a mutator and a ludeme is blurry, since neither concept is formalised. Herein, we understand game mutators to be ludemes that function as generic game transformations.

The mutator concept is well known in computer science. The idea of a simple procedure that can be applied to a framework to change its behaviour is widely used [2], and has particular importance in the construction of modular and expandable videogame engines. For example, the Unreal system uses a scripting language to specify game parameters and allows the creation of mutators that easily extend games in many ways (often called ‘modding’), thus adding new behaviour to game objects [3].
1.2.1 Progressive Mutators

In the board game context, a classic mutator is progressive, which transforms a game like Chess from alternating turns with one move each into a game with progressively longer sequences of moves on each turn. The Progressive mutator fits well with Chess, and Progressive Chess is an enjoyable game with interesting tactics. However, Progressive Go, without any further adaptation, does not make much sense. For example, living groups are almost impossible to achieve if a player can fill all eyes of an enemy group in one turn. Even with a rule to forbid liberty-less groups within a single multi-move turn, the ability to suddenly make very long walls changes the whole character of the game into something brutally simpler.

1.2.2 Multi-Move Mutators

A mild version of a multi-move mutator is 12*, i.e., the first player plays one move, and subsequently the players play two moves each per turn. Very often it is sensible to restrict the two moves within a turn to be played with different pieces. The 12* mutator is often played as a form of equalisation, to reduce a first-player advantage. Connect4 is a well-known Gomoku variant that applies this mutator. Another way to eliminate first-player advantage is with the classic mutator known as the Pie Rule or Swap Rule. Another classic mutator is Miser, reversing the victory condition.

1.2.3 Restrictive Mutators

The authors, during their game playing, have discovered new restrictions that improved specific games, and even salvaged otherwise failed games. Experience has shown that some of these restrictions are flexible mutators, applicable to a large family of games, and these are called restrictive mutators. While modifying a game, the mutators are also intended to preserve some of the ‘spirit’ (gameplay, strategy, tactics) of the original game. This article presents a collection of restrictive mutators and some of the games that evolved in this ecosystem. These games are original and underwent extensive testing by the authors.

2 Group Restrictions

Multi-move games are interesting for at least two reasons. First, they are faster and hence more practical to play by correspondence (mail, email, etc). Second, and more importantly from a design perspective, the mutated rule set may have more strategic or tactical depth, since it generally increases the number of possible moves per turn.

However, this increase might break the original game. As mentioned, Progressive Go does not work well. Another example is Progressive Hex, which turns a remarkable game into a mindless race to connect with increasing numbers of placements per turn. Taking this into consideration, if the designer wants to salvage a progressive variant, there should be rules to restrict the excessive flexibility provided by the steadily increasing number of moves per turn.

One method is to slow down the move increase. The standard Progressive mutator increases the number of moves each turn (1, 2, 3, . . .), but more moderate progressive mutators can propose sequences like 2, 2, 2, . . ., which is a classical mutator found in Marseillais Chess created around 1920, or variants addressing first-player advantage, such as 1, 2, 2, . . ., or 2, 4, 4, . . ., or 1, 3, 4, 4, . . ..

Even these mild progressive mutators can introduce problems when applied to certain games. In fact, this section’s rule sets are games with mild progressive mutators that needed an extra transformation to become playable. This extra transformation is group restriction.

Group restriction is a simple concept. Assuming a well-defined definition of ‘group’ in the context of a given game (e.g., a set of orthogonally connected stones of the same colour, as in Go), it is forbidden to place two or more stones such that they would be in the same group. The group restriction must be satisfied at the end of the player’s turn, after all pieces are placed. (Naturally, one adapts the notion to suit a specific game. In a game with movement instead of placement, one could forbid moving stones to – or from – the same group.)

2.1 Coy Lips

The first example presented is a game called Coy Lips that use the concept of tetrahexes, i.e., a group of four connected stones in a hex tiling. There are seven possible tetrahexes named \(C, O, Y, L, I, P, S\), hence the game’s name. The rules are as follows:

- Players place stones in a 2, 4, 4, . . . move sequence.
- The stones are played on empty cells with group restriction, i.e. all new stones must belong to different groups at the end of the player’s move.

\(^4\)https://en.wikipedia.org/wiki/Connect6
\(^5\)http://www.chessvariants.org/multimove.dir/marseill.html
A stone placement can be passed, and two consecutive four-pass turns end the game.

A group with more than four stones is allowed, but no subgroup of it counts as a tetrahex for victory.

Players win immediately by forming a tetrahex of each of the seven types in their own colour. If the game ends before this happens due to consecutive passes, then the player with most completed tetrahexes wins; if the scores are tied, then the player who built the first final tetrahex wins.

Figure 2 shows a Coy Lips end game. White is missing only their Y tetrahex, which can be formed at either cell marked a. Black cannot play on both of those cells to block them, due to the group restriction rule, and so resigns.

If Coy Lips did not have group restriction, in each move a player could simply place four contiguous stones to create a single new tetrahex, turning the game into a much less interesting race to finish. With the group restriction, tetrahexes are built more gradually, and flexible planning is more important. Players have time to hinder the opponent’s forming patterns, while building their own. Players cannot form a tetrahex in one move, so they must expect interference by the opponent.

Another source of tactical interest comes from the fact that not all tetrahexes are equally easy to build. Trying to build a single O or Y can be more easily blocked by the opponent, so players must create multiple compatible frameworks in order to achieve at least one of each pattern. More flexible tetrahexes, like L and P, are often completed near the end, using the extra frameworks built to achieve the harder tetrahexes.

2.2 Criss-Cross

The next game is called Criss-Cross and is based on the game Cross by Cameron Browne. In Cross, a player must connect three non-adjacent board sides, but loses by connecting two opposite sides (unless also connecting three non-adjacent sides to win).

In Criss-Cross, a player connecting two adjacent sides (without winning) does not lose, but instead must remove a stone from the connecting group to break the connection. Another change is the use of the 12* move protocol. With only those changes, the game would need a large board to remove some of the influence of playing a simple yet powerful pair of connected stones.

Figure 4 shows how group restriction improves the game and better preserves the spirit of the original’s gameplay. Without group restriction, the secure rhombus bridge connections of Hex and many other games are no longer secure, because both miai cells can be filled in one turn by the opponent, as White move 1 breaks Black’s

6https://boardgamegeek.com/boardgame/59668/cross
7Miai is a Go term describing a pair of options, such that if player A takes one then player B can take the other.
connection in the upper left diagram. A secure virtual connection would need four stones, as in the upper right diagram. Conversely, two stones separated by a single empty cell, which is unsafe in Hex, is safe in unrestricted Criss-cross, assuming no edges or opponent pieces nearby.

![Figure 4. Criss-Cross without group restriction would have different blunter tactics.](image)

The group restriction mutator reverses this reversal, bringing us back to Hex intuition. Furthermore, unrestricted multi-move too easily creates unsubtle fast ladder sequences like the bottom left diagram. In contrast, the bottom right diagram uses group restriction: Black must make connection threats in order to advance eastward. Here (without considering the whole board context), White was able to stop Black’s advance. With group restriction, players should typically have support stones across the board and not simply play all their stones together. The difference in tactical subtlety is clear.

Another advantage of group restriction is that a smaller board is possible, making the game faster to play. Figure 5 shows a match on a hex-hex board of size 4. Black wins immediately if allowed to play at a, and wins on their following turn if they connect at b, so White plays on both a and b. However, this creates a white connection between opposite edges, so White must remove a stone to break this connection. The game continues several more turns, but Black has the upper hand, and White eventually resigns.

In our opinion, the game’s depth and its intricate positions, despite the board size, are only possible due to the interaction between the game’s original goal and the combination of multi-moves and group restriction. Unrestricted 12* moves allow too much freedom and ruin basic tactics such as the virtual rhombus connection, widely used in connection games like Hex, as seen in Figure 4.

![Figure 5. Criss-Cross example endgame.](image)

### 3 Non-Linear Restrictions

A common problem in applying mutators is making a game ‘too simple-minded’, as the most obvious way of using the new freedom is to make what may be termed a series of ‘bulldozing moves’: using the number of available moves to build unstoppable winning structures. Another way of dealing with this, applicable where group restriction may not apply adequately, is the non-linear restriction. This means that any two stones played within the same turn may not be in the same (orthogonal or diagonal) line, even if they are in different groups. The effect of this restriction in preventing simple-minded bulldozing can be seen in the next game.

This game is called Super-Pente. Capturing is possible by the usual Pente [6] (or Gomoku-Ninuki) custodial capture rule: two adjacent opponent stones trapped in a 4-in-a-row between a newly placed stone and an existing stone of your colour are removed. Players place stones using the 12* progressive mutator, with the non-linear restriction mutator also applied. The goal is to make 4-in-a-row instead of the usual 5-in-a-row, which highlights the power of this balancing mechanism. Note that capturing five pairs also wins, as in the original Pente.

Unrestricted multi-moves in Super-Pente are too powerful. Figure 6 shows the simplest pattern, an open pair of black stones. On the left diagram, with unrestricted multi-moves, White easily plays at both ends, capturing the black pair. With the non-linear restriction, White must play

---

8A hex-hex is a hexagon tessellated by hexagons, with \(n\) cells per side, giving a total of \(1 + 6n(n - 1)/2\) cells.
more subtly. The right diagram shows a possible continuation. Note that one of Black’s four stones threatens to win with 4-in-a-row, while the other threatens to capture a white pair. White cannot block both threats, due to the non-linear restriction. The non-linear restriction enables appropriate tactical complexity in this multi-move Pente variant.

Figure 6. Super-Pente gains tactical interest with the non-linear restriction.

Figure 7 shows a longer example game. The upper left white 10 is forced, to block Black’s threat to win with 4-in-a-row. The other white 10 nicely creates an open line of three white stones (10, 8, 4). But the non-linear restriction prevents Black from simply blocking both ends of White’s open three. (White 10 also threatens a horizontal 4-in-a-row via 10, 2, 4!) So Black resigns.

Figure 7. Super-Pente example.

4 Delayed Captures

This next mutator was invented specifically to make a Go variant, but could conceivably be applied to other games in which captures, or near-captures are used, such as Reversi, Pente, etc. In Go, the new rule is simple: liberty-less groups stay on the board until specifically removed, which uses up a whole move. Corey Clark’s Rampart\textsuperscript{9} is another game that uses this type of delay capture mechanism. This is probably somewhat uninteresting as a modifier for plain Go, but seems worthwhile for other variants, such as the following.

4.1 Delayed Removal Go

Delayed Removal Go is played with a 12* move sequence. On each move, the player may either play a stone to an empty cell, remove a liberty-less group, or pass. Superko applies to end-of-turn situations.

If superko (forbidding board state repetition) applies to stone removals as well as placements, then the new game arguably deviates too far from the ‘spirit of Go’, because two-eyed groups (which are safe in Go) become surprisingly vulnerable to attacks.

Figure 8 shows an example. Black has just placed the two stones at 1, attacking White’s corner; note that the black stones are themselves without liberties! White could remove them – and would, in fact, like to – because if they remain, then Black will remove the three liberty-less white stones on Black’s next turn, leaving White’s entire corner dead with only one eye. However, White cannot remove both black stones, as repeating an earlier board state is illegal, due to the ko rule.

White should, therefore, remove one stone and make a large ko threat elsewhere, large enough that Black is forced to respond with both moves, so that White can then remove the remaining black stone and perhaps connect the two white groups. This reduces the danger, but even so, Black will be able to return here and drop two stones in for another assault, again and again.

\textsuperscript{9}https://www.boardgamegeek.com/boardgame/133923/rampart
This problem led to altering the superko restriction: removals are always legal, and superko applies only to turns in which a stone is placed. Thus, the ‘spirit of Go’ (two eyes being alive) is retained. In designing games or applying mutators, it is essential to test for such unexpected side-effects and make any necessary repairs.

Even with superko only applying to turns with placement, interesting new types of ko fight appear, more in the spirit of Go. Figure 9 shows a position in which Black must capture White’s stone 1 to save the white group. Further, White can keep placing a new white stone there each turn, renewing the threat every turn. This adds a new level of complexity, which, in certain situations, might be nested with traditional ko fights.

5 Cell Values

Another way to dampen the effect of multi-move modifiers is to assign a cost to each cell, with higher costs on the more useful cells (for example, centrally located cells). Then the players must apportion their ‘ration’ of move points as they see fit – either many cheap cells, or one or two expensive ones. This adds another layer of skill in deciding where to spend multiple moves.

5.1 Hex-Y

The following example is called Hex-Y, a connection game played on the board of Figure 10 with a 134 point budget to place stones on empty cells, and group restriction. Placing stones around the centre costs more than placing them at the edges.

The goal is to connect all three side colours (black, grey, white) with a single connected group. The hexagonal board shows the points required to place a stone on each hex (cells without number cost one point). Figure 11 shows the initial moves of a typical Hex-Y game. The first moves create influence and connecting threats over the edges, via low-cost cells.

The group restriction adds tactical richness that improves multi-move sequences. Figure 12 shows an example of a common tactic: making multiple connection threats that cannot all be countered in one turn, due to the group restriction. Black’s stones at 1 threaten to break White’s connection in two different places. White cannot reply to both, because that would mean placing two white stones adjacent to the same white group. Thus, Black will be able to prevent White’s victory next turn.
Figure 12. White cannot stop both black threats.

6 Conclusion

When creating new games, a game designer sometimes combines ludemes that produce an original rule set with too much liberty, e.g., the move possibilities are so large that it is possible to find tactics that easily guarantee a win or tie. If no simple solution is found, the new game may be judged broken and discarded.

This article describes mutators that are useful to restrict gameplay so that a balance can be achieved where the new idea can still be expressed without the existence of simplistic tactical solutions. These mutators were especially designed to cope with multi-move turns. Given the resulting increase of move options, the new variants usually have greater depth than the original rule sets.

Several original games were presented using one or more restrictive mutators that made it possible to combine multi-move turns with other ludemes and still retain interesting game dynamics in the spirit of the original game, while introducing new possibilities as well.

Acknowledgements

João Pedro Neto’s work is supported by BioISI grant UID/MULTI/04046/2013, from FCT/MCTES/PIDDAC, Portugal.

References


João Pedro Neto is a professor of Computer Science at the University of Lisbon. A continual area of interest is the study, design, collection, publication, and play of abstract games, for the past 20 years together with Bill Taylor, Jorge Nuno Silva, and others. He helps organise a yearly national event in Portugal to disseminate new abstract games among young students.

Address: Biosystems & Integrative Sciences Institute, University of Lisbon, Campo Grande, 1749-016, Portugal. Email: jpn@di.fc.ul.pt

William Taylor is a former lecturer in mathematics at the University of Canterbury, Christchurch New Zealand, with research interests in probability theory and discrete mathematics. He has given seminars on abstract games at Board Games Studies colloquia and other venues.

Address: Department of Mathematics, University of Canterbury, Christchurch, 8041, New Zealand. Email: W.Taylor@math.canterbury.ac.nz
Explore the Design Space

Cameron Browne, Queensland University of Technology (QUT)

This article outlines some simple strategies for optimising the search for new games in the conceptual game design space. While most games are created through the combination of existing ideas, naive combinatorial search is not enough. A brief exploration of a family of path-based tile games, and games derived from them, is presented by way of example.

1 Introduction

How are games created? Why are some designers more skilled at producing good designs than others?

In a Platonist view of the world, the games are all out there, like undiscovered mathematical constructs and theorems, just waiting to be found. And while finding new designs is only part of the process – identifying the true value of new designs can require just as much skill – it makes sense that a systematic exploration of the potential design space might benefit the search.

This paper outlines some strategies for exploring the conceptual game design space efficiently, using as a case study a simple path-based tile game from 1960, and some of the many subsequent designs that build upon it.

2 Models of Creativity

The concepts outlined in this paper refer to two key models of human creativity.

2.1 Boden’s Model

A widely accepted model of human creativity proposed in 1990 by cognitive scientist Margaret A. Boden, posits three distinct forms of creativity:

1. Combining familiar ideas in new ways.
2. Exploring the conceptual space for previously undiscovered ideas.
3. Transforming the conceptual space.

The aim in each case is ‘to come up with ideas or artefacts that are new, surprising and valuable’ [10 p1]. Combinatorial creativity, the first form, is especially relevant to the field of Computational Creativity, where the question is how to direct computers – which are excellent at trying combinations of things – towards exploring more fruitful parts of the conceptual space.


2.2 Weber’s Model

Around the same time, mathematician Robert J. Weber proposed another model of human creativity from a similarly technical perspective, based on the historical development of hand tools [2]. Weber observes that the mechanisms for invention include:

1. Joining existing features in new ways.
2. Adding new features to existing ones.
3. Refining features through fine-tuning.
4. Transforming the feature space through abstraction.

Weber also highlights the question of how to handle the combinatorial explosion of ‘promiscuous ideas’ when joining all possible ideas in the conceptual space. He suggests the use of search heuristics, and different levels of joining, to focus the exploration.

2.3 Creative Leaps

Noy et al. define a creative leap as ‘the momentary intersection of two different matrices of association in the design space’ and describe it as the crux of the creative act [3]. A creative leap draws together new combinations of ideas from different regions of the design space, but a key point is that some degree of overlap between these concepts must generally exist. This is essentially the mechanism behind the theory of conceptual blending, in which related concepts are blended within associated frames [4]. This suggests strategies for exploring the design space effectively, as follows.

2.4 Implications for Game Design

Boden’s model is a higher-level description of what mechanisms are at play, while Weber’s model is a lower-level description that gives instructions for how to create new artefacts. Yet both models highlight the importance of combinatorial search, and distinguish it from transformational forms of creativity.
This distinction is most relevant to game design, where truly transformational ideas or developments are few and far between. The emergence of connection games in the last century is an example of a simple idea that transformed the abstract board game landscape, but even that was based on an existing mathematical principle and borrowed from another context [3]. The component parts of almost any game today can be traced to some earlier game or mathematical principle, with very few exceptions.

This makes the exploratory form of creativity especially important for game design. Given this vast pool of ideas from millennia of human invention, how to best combine them to create new and interesting games? This is an especially difficult question for game design, as the true value of a game only emerges when it is actually played, which can be a time-consuming process.

A complete combinatorial search of the conceptual space is not feasible, so it is the skill of the game designer to intuitively identify those areas of the design space most likely to yield rewards, and which ideas are most likely to work harmoniously with others. The following discussion demonstrates this exploratory approach to game design, and ways to limit combinatorial explosion, using path-based tile games as an example.

3 Example: Black Path

The Black Path game, invented by Larry Back in 1960, is the quintessential path-based tile game [6, pp. 746–48]. The game is played with the two square tiles shown in Figure 1 on an \( m \times n \) square or rectangular board.

The game begins with the first player placing a tile on a board edge – not a corner! – which defines the starting point of a path from that edge, indicated by the arrow in Figure 2 (left). Players then take turns placing a tile, in any rotation, to continue the path. A player loses if their move runs the open path end into the edge of the board. The remaining parts of Figure 2 show various steps throughout a completed game.

The question is now: what other interesting games can be derived from this basic idea?

3.1 Degrees of Freedom

Given this starting point in the design space, it is useful to consider the relevant degrees of freedom, as it is the recombination of these that will produce further designs. The main degrees of freedom here involve the game’s rules and geometry, as is usually the case:

1. Tile placement rules.
2. Winning conditions.
3. Tile shape.
4. Path shape (end points, overlap, etc.).
5. Path foreground or background.
6. Number of paths and/or colours.
7. Number of dimensions.

This already gives a number of combinations to try, but it is worth further clarifying the underlying geometry since it is so important to this type of game. Figure 6 shows the three regular tilings: triangular, square and hexagonal.

We shall constrain our search to games based on these tilings, as they are easy to work with (allowing uniformly shaped pieces), aesthetically appealing, and familiar to players. By focussing on these three key geometries which are most likely to give good results, we have already limited a potential combinatorial explosion over all possible tilings. The danger is that some good designs on exotic tilings might be excluded; this point is considered later, in Section 5.6.1.

This already gives a number of combinations to try, but it is worth further clarifying the underlying geometry since it is so important to this type of game. Figure 6 shows the three regular tilings: triangular, square and hexagonal.

We shall constrain our search to games based on these tilings, as they are easy to work with (allowing uniformly shaped pieces), aesthetically appealing, and familiar to players. By focussing on these three key geometries which are most likely to give good results, we have already limited a potential combinatorial explosion over all possible tilings. The danger is that some good designs on exotic tilings might be excluded; this point is considered later, in Section 5.6.1.

Figure 2. Snapshots of a completed Black Path game on a 4×4 board; the last player to move loses.

\[\text{Figure 1. The Black Path tiles.}\]

[2] A degree of freedom in this context is any feature that can be modified to produce a different game.
4 Path-Based Variations

This section describes some path-based games conceptually similar to the Black Path game. This does not necessarily mean that they were derived from it — at least one predates it — but that they exist in the same region of the design space, separated by some combinatorial difference(s).

4.1 Square Path Tiles

There are many path-based games that use square tiles. This is the easiest of the key geometries to work in, and the most familiar to players. Square tiles are also ideally suited to this type of game, as their four sides nicely match two paths with two ends each. The following list presents just a few representative games of this type.

4.1.1 Turnabout

Turnabout, designed by an unknown author in 1982 [3, pp. 215–18], looks very similar to the Black Path game, but with modified tile placement and winning conditions: players can place a tile on any empty square, and one player (Offensive) wins by connecting opposite board sides with a path while their opponent (Defensive) wins by preventing this. Figure 4 shows a game won by the Offensive player.

Turnabout may appear similar to the Black Path game, but the connective goal and split roles of the two players constitute creative leaps [3] that offer something new.

4.1.2 Trax

Trax, designed by David Smith in 1980 [3 pp. 183–88], uses path shapes similar to the Black Path tiles but distinguishes them by colour (Figure 5) and does not use a board.

Two players, White and Black, take turns placing a tile of their choice to extend an existing path end, making additional forced moves wherever two open path ends of the same colour dictate that one particular tile must be played at any particular cell. Players win by making either: 1) a closed loop of their colour (Figure 6), or 2) a path of their colour that spans eight rows or eight columns.

Trax is a great example of elegant game design. It simplifies the equipment by removing the board while at the same time adding a significant strategic element by making each tile contain both friendly and enemy colours, making each move fraught with both potential danger and benefit.

4.1.3 Thoughtwave

Thoughtwave, designed by Eric Solomon in 1973 [3 pp. 207–212], involves square tiles with both simple and branching paths, giving tiles with 1 to 4 sides with path ends, as shown in Figure 7. Players have identical sets of 24 tiles and share the paths shown on each, and win by completing a path between the board sides of their colour.
This branching path motif is found in many later connection-based games, such as Ta Yü [3, p. 291], Carcassonne, etc.

### 4.1.4 Lightning

The game Lightning involves square tiles each with a simple black path, but in this case the path ends are not constrained to edge midpoints but also reach some corners, as shown in Figure 8.

Lightning is an interesting case as it was patented in 1892, predating the other games listed here by up to a century, and making it the first true connection game by decades [3, pp. 173–176]. It was not inspired by the Black Path game, but exists in the same region of the design space.

### 4.1.5 The Squiggle Game

The Squiggle Game from 1979 [3, p. 202] is similar in principle to the Black Path game, except that each tile has two path ends from each side giving four paths per tile, as shown in Figure 9.

Players start by placing a piece of their colour on a path. Each turn, the mover places a tile adjacent to their piece, and then moves the piece as far as possible along the path thus extended, removing it if it runs into the board edge or collides with another piece. The owner of the last surviving piece wins.

This notion of active path ends harks back to the Black Path game, as players strive to keep their piece’s path alive as long as possible before it is run into the board edge (or another piece). However, the Squiggle Game involves additional strategies due to the more complex path shapes and the presence of moving and interacting pieces. It was rebranded and released as Tsuro in 2004, shown in Figure 10.

### 4.1.6 Knots

Knots from 1991 [3, pp. 221–24] uses similar tiles to The Squiggle Game, except that some paths branch, as shown in Figure 11.

The aim in Knots is for players to connect their opposite sides of the board with a continuous path, as per Thoughtwave and many other connection games. The owner of the last surviving piece wins.

Figure 7. Thoughtwave tiles and a game in play.

Figure 8. The Lightning tiles.

Figure 9. Some Squiggle Game tiles.

Figure 10. A game of Tsuro. Photo by Ernie Lai.

Figure 11. Some Knots tiles.

Figure 12. Detail of a game of Metro. Photo by Łukasz Rygało.
4.1.7 Metro

Metro[^3], designed in 1997, uses tiles with split paths similar to The Squiggle Game and Knots, as shown in Figure 12. Players strive to create paths between stations printed on the board.

4.1.8 Waroway’s Game

Waroway’s Game from 2001 uses rectangular rather than square tiles (Figure 13), but is included here as each tile consists of a pair of squares that share paths [^3, p. 313].

![Figure 13. Waroway’s Game.](image)

4.2 Hexagonal Path Tiles

The examples so far have explored different path shapes within square tiles, but we now consider different tile shapes. The hexagon is an obvious choice: it has an even number of sides, forms a regular tiling that fills the plane, occurs often in nature and is aesthetically appealing. Hexagonal tilings also have some attractive geometric properties that can be especially relevant for games. For example, they pack without diagonal neighbours and form *trivalent* deadlock free points where three incident cells meet around an intersection [^3, pp. 363–65].

We therefore examine hexagonal tiles next, as it is good practice to explore those parts of the design space most likely to produce fruitful results. There are in fact so many path-based games with a hexagonal basis, that only one representative example is presented here.

4.2.1 Tantrix

Tantrix is a boardless tile placement game from 1988, in which two to four players randomly draw tiles and place them to continue existing paths, in an effort to form closed loops of their colour [^3, pp. 288–89]. The set consists of 56 tiles, which represent all combinations of ways in which three paths may be coloured within a hexagon using four colours, such that each tile has no more than one path of any colour. Figure 14 shows the four basic path shapes (the case of three straight lines all crossing the centre is not used).

![Figure 14. Four of the Tantrix tiles.](image)

Tantrix is a refinement of previous path-based games on hexagonal tiles, such as Kaliko[^3, pp. 286–87], but has enjoyed much greater success than its predecessors. This is helped by the fact that the Tantrix tiles are cleverly numbered to encode sub-puzzles that can be played separately to the main game [^7], allowing mini-sets to be sold as cheap and attractive gifts suitable for gamers and non-gamers alike[^4].

4.3 Triangular Path Tiles

Continuing our exploration of the design space, we now consider path-based games that use triangular tiles. Triangles have an odd number of sides, which is inconvenient as simple paths typically have an even number of end points (i.e. zero or two), and only connecting two sides with a path per tile is not very interesting.

The fact that each corner has two orthogonal and three diagonal neighbours (see Figure 6) also complicates tile placement and spatial relationships. There are very few path-based games with a triangular basis; one is shown below.

4.3.1 Vasco

Vasco, from 2008, is played with triangular tiles containing paths in two colours[^5]. Figure 15 shows how the odd number of tile sides is handled, by allowing path ends to merge at a common side. The aim is for players to complete a closed loop of their colour, as shown in Figure 16, which will necessarily use path sections of merged colour.

![Figure 15. A Vasco tile (front and back).](image)

[^3]: https://www.boardgamegeek.com/boardgame/559/metro
[^4]: Personal correspondence with Tantrix inventor Mike McManaway.
[^5]: http://www.cameronius.com/games/vasco/
Vasco demonstrates how observing the available degrees of freedom, and the relevant constraints, can yield simple solutions to produce playable games. The fact that there are so few path-based games on triangular tiles suggests that this subregion of the design space is not very fruitful, and should be a low priority for designers (unless they are looking primarily for novelty).

5 Area-Based Variations

The examples listed in the previous section only hint at the number of path-based games, of which there would be hundreds if not thousands. There is even a separate category for them at the US Patent Office: 273/275 (Amusement Devices: Games/Path Forming). This region of the conceptual space has been well explored, so how can a designer further innovate on this theme?

5.1 Foreground to Background

The defining feature of these tile-based games is the path patterns on the tiles. The examples in the previous section show how a range of games can be created by changing the path shapes and associated rules. We now consider a more fundamental leap, based on path shape, that opens up whole new regions of the design space.

Rather than focussing on the foreground of the tiles (i.e. the tile paths), we now focus on the background (i.e. the areas defined by the tile paths). For example, Figure 17 shows two square tiles, with regions coloured using the second Black Path tile paths as boundaries. These are described as white-dominant (left) and blue-dominant (right).

5.2 A (Minor) Transformation

This switch from foreground to background is a simple mathematical transformation. It does not constitute a creative transformation of the design space in terms of Boden’s model (outlined in Section 2), as that would involve defining entirely new types of game, which is not the case here.

However, this switch does constitute a transformation in terms of Weber’s model, as we are abstracting the key features of the problem and moving the search to a different – but overlapping – part of the design space. Boden’s transformation might be called a major transformation of the design space, while Weber’s transformation might be called a minor transformation within the design space, to differentiate these two concepts.

Abstracting the problem to a higher level can be a powerful approach, essentially taking a metalevel upwards in the design space. It allows all combinations previously tried to be retried within this new context, allowing known ideas to be harnessed in novel and hopefully useful ways.

Note, however, that such minor transformations may only suit a subset of the existing ideas, depending on their relevant features, and that the transformation could act as a filter for those ideas that do not fit the new context. For example, it makes sense to use the separate paths of the second Black Path tile to denote separate area, as shown in Figure 17, but it does not make sense to use the intersecting paths of the first Black Path tile for this purpose, as the resulting areas would not be as cleanly separated.

I have personally found this area-based region of the conceptual space fruitful as a designer; all of the following examples are original games. This demonstrates how a simple shift of emphasis, strategically up-stepping to take a creative leap to an overlapping part of the design space, can yield rewards.

5.3 Square Area Tiles

We will now revisit some concepts found in the games listed above, but applied to the tile background (area) rather than foreground (path). Square tiles proved most fruitful in the past, so will be considered first.

\(^6\)Thanks to Russ Williams for this observation.
5.3.1 Che

In Che, from 2008, two players take turns placing two of the tiles shown in Figure 17 to match sides with existing tiles. Players aim to complete a closed area of their colour, as shown in Figure 18.

![Figure 18. A game of Che won by Blue.](image)

Che is a simple but elegant game with some basic strategy. It has something of the character of Trax, but not the depth as it lacks forced moves.

5.3.2 Xutoli

Xutoli, also from 2008, uses the same tiles, except in this case players take turns placing a single tile, and win by completing a pair of identical and non-overlapping $2 \times 2$ patterns. For example, the mover has just won the game shown in Figure 19 by completing two ‘X’ patterns (highlighted).

The game is named after letter shapes resembled by the six unique $2 \times 2$ patterns: ‘X’, ‘U’, ‘T’, ‘O’, ‘L’ and ‘I’. Two non-overlapping ‘U’ patterns also exist in the position shown, but they are in different rotations so do not count as a win.

![Figure 19. A winning pair of patterns in Xutoli.](image)

As a secondary win condition, the player with the largest region of their colour wins if no winning pair of patterns is formed before the tiles run out. This rule was added to resolve the problem of deadlocks, but it also brings tile colour back into the game as a factor, encouraging players to play tiles with their colour dominant, adding a much-needed secondary layer of strategy.

5.4 Hexagonal Area Tiles

While some area-based games on square tiles were devised, hexagonal tiles proved far more fruitful in this case.

5.4.1 Mambo

Mambo, designed in 2007, was actually my first game of this type, based on an intuition that three colours on a hexagonal tile would allow greater scope for design. The Mambo tile is shown in Figure 1 in its three rotations per side.

![Figure 20. A mambo tile (front and back).](image)

Two players, Red and Blue, take turns placing a tile, either side up, to match at least one existing tile. A player wins if any enemy group can not grow any further.

For example, the owner of the central group loses in both positions shown in Figure 21 as neither group can be grown any further. Note the existence of a null point, marked $\times$, where no tile can be placed.

Such null points give Mambo a different character to Trax, and add a new tactical element; such points are generally filled by forced moves in Trax before they become unplayable. Mambo involves a weak form of forced move, in which tiles are automatically played when all three colours surrounding empty cells are known, as shown in Figure 22. Compare this to the strong forced moves of Trax, which can be made as soon as two incoming paths of the same colour are known, which greatly reduces the incidence of null points.

---

8[http://www.cameronius.com/games/xutoli](http://www.cameronius.com/games/xutoli)
Mambo involves some interesting tactical plays, and can be enjoyable to play. However, the rules are not as elegant as might be hoped, having to cover several special conditions, and the game is prone to unsatisfying denouements if either player establishes an unbeatable ladder, which they can force until the tiles run out, or some arbitrary limit is reached.

Palago, designed in 2008, is a simplification of Mambo that uses only two colours: white and dark. The tile design is otherwise similar, as shown in Figure 23.

Palago has good scope for tactical and strategic play, and games are typically played on a knife-edge of tension, with each move setting up immediate win threats that must be addressed by the opponent; momentum is everything.

The tiles also have an artistic application and allow players to make interesting abstract shapes, whether as part of a game or as a creative venture akin to doodling, such as the deer shape shown in Figure 25. Palago is published by the owners of Tantrix, and like Tantrix it is presented as both a game and a Tangram-like puzzle-within-a-game, as players can play a solitaire mode in which they must replicate predefined shape outlines with Palago tiles.

Gates is a game system, designed in 2008, that substitutes Palago tiles for the pieces of existing connection games, to create more complex variants. This is another example of abstracting upwards in the design space, but this time to create a
C. Browne Explore the Design Space

Figure 26 shows a game of Y Gates, which White has won by completing a white region connecting all three sides of the board.\[12\]

The Gates tiles have therefore been superimposed on the existing connection game Y [3, pp. 77–83], to filter connectivity according to the path shapes. They can also be applied to a variety of other connection games that use a hexagonal grid – and many do! – including Hex, Cross, etc.

5.5 Triangular Area Tiles

Triangular tiles are also problematic for area-based tile games, for the reasons given in Section 4.3. Only one example is known.

5.5.1 Trichet

Trichet, also from 2008, involves the two tiles shown in Figure 27.

Two players take turns placing a tile of their choice, to match at least one existing tile corner. The aim is similar to that of Palago: to create a closed region of your colour more complex than a circle, such as the dark region in Figure 28.

Figure 27. The two Trichet tiles.

Figure 28. The closed dark region wins.

Trichet also features a strong forced move mechanism, in which points at which only one tile can be placed are immediately filled with that tile, which can trigger further forced moves. For example, move \(a\) in Figure 29 triggers forced moves \(a'\) and \(a''\), which in turn triggers forced move \(a'''\).

Figure 29. Strong forced moves in Trichet.

5.6 Other Tile Shapes

So far we have only considered square, hexagonal and triangular tile shapes, to reduce potential combinatorial explosion. However, such dogged constraints can miss some interesting designs, so it is good to keep an eye out for interesting exceptions, such as the following case.

5.6.1 Trilbert

Figure 30 shows a field tessellated by the second Black Path tile in random orientations, creating a set of continuous contours. Such tessellations, called Truchet-like tilings, are well known to computer graphics practitioners as a quick and easy way to produce attractive random textures [8].

\[12\]http://www.cameronius.com/games/gates/
Truchet-like tiles were first proposed in 1987 [2], well after the Black Path game and Trax were released. It is not clear whether there is any connection between them, but this could be a case of the mathematical sciences drawing inspiration from game ideas.

Trilbert, designed in 2007, is played with pieces in the Truchet-like shape highlighted in Figure 30, with each half coloured separately in each player’s colour [3].

Players take turns placing a Trilbert tile to interlock with at least one existing tile. A player wins by fully enclosing an empty region with their colour, marked $x$ in Figure 31. Trilbert is not a deep game, but blurs the line between games and puzzles by adding a competitive element to a jigsaw-like playing experience.

Designing these games was a process of refining such degrees of freedom once the basic mechanisms had been decided, to find the rule sets that produced the most balanced and interesting games. These degrees of freedom also suggest launching points for searching for new but related games.

### 6 3-Dimensional Variations

The games described so far have all been played using flat 2-dimensional tiles. We now consider another leap in the design space, to 3-dimensional versions of existing designs. This is another upwards abstraction that allows the reuse of existing ideas – which were themselves probably based on existing ideas – within a new context, giving a recursive nesting of concepts.

#### 6.1 Boche

In Boche, designed in 2008, two players share a pool of cubes which show a Che tile on each face. Figure 32 shows the construction of such cubes from two identically shaped but differently coloured pieces.

Boche is played on the unusual board shown in Figure 33 which forces the cubes to sit at an angle (on a vertex rather than a face) and build a triangular pyramid as they stack. A player wins by connecting the three board sides with a region of their colour. The figure shows a win for White.
Boche is a truly 3-dimensional game and is probably the most unusual game presented here. But note that it is made up of previous ideas; the cube design is from Che, the winning condition is from Y, the angled cube stacking is found in other games, such as Inside from 2007. It is the novel way in which these ideas are combined that gives this game its unique character.

### 6.2 Blobs

Blobs, from 2009, uses cubes of a different design in three colours. Each cube has a red-dominant face, a blue-dominant face, and four white-dominant faces with red and blue corners, as shown in Figure 34.

![Figure 34. Three views of a Blobs cube.](image)

The game is played without a board. Players take turns adding a cube, in any orientation, to match the visible upper face of at least one existing cube, and aim to either: 1) fully enclose a white region with their colour; or 2) have a region of their colour fully enclosed by white.

Strong forced moves are enforced, i.e. if a move creates any point at which only one orientation can be played, then a cube is placed there in that orientation. For example, Figure 35 shows move $m$ triggering forced moves $m'$ and $m''$, to win the game for Blue (light).

![Figure 35. Forced moves win for Blue (light).](image)

The dominant region on each face acts as a connector that extends regions of the relevant colour, while the two quarter-arcs on each face act as closers that help close regions. The fact that the four white-dominant faces include closers of each player’s colour means that players can be forced to make moves that help the opponent if they are not careful, leading to some interesting tactical decisions.

Blobs is not a true 3-dimensional game as it is played on a flat surface; the pieces are 3-dimensional, reflecting their inspiration, but only their uppermost faces count; the game could just as well be played with three types of flat square tiles. The Boche cube design is reminiscent of Palago while the Blobs cube design is reminiscent of Mambo, highlighting the inspirations for these two games.

### 6.3 Osbo

Osbo is a game from 2008, played with the Celtic knot-inspired dice shown in Figure 36.

![Figure 36. The six faces of an Osbo die.](image)

The knot is continuous over all six faces, and is actually composed of two interwoven knots. Figure 37 shows an Osbo die unfolded and coloured to reveal this.

---

15http://www.boardgamegeek.com/boardgame/30179/inside
16http://www.cameronius.com/games/blobs/
17http://www.cameronius.com/games/osbo/
18The Osbo dice design has recently been ‘independently rediscovered’ and released as Knot Dice.
In Osbo, players take turns rolling a certain number of dice, then use the faces showing uppermost to extend their patterns. Players score 1 point for each unit length of their longest closed knot, hence the left player would win the game shown in Figure 38.

Osbo is a path-based tile game in disguise. As with Blobs, the fact that the pieces are cubes gives it a 3-dimensional aspect, but only the uppermost faces are used.

6.3.1 Celtic!

To reinforce the fact that Osbo is a 2-dimensional game played with 3-dimensional pieces, consider the derived game Celtic! which was invented in 2009. Celtic! is played with flat tiles identical to the Osbo die faces, but with coloured backgrounds, as shown in Figure 39. Two players – Orange (light) and Blue (dark) – own the tiles with their background colours and share the tiles with white backgrounds.

The game ends when no more pieces can be played to extend any further knot ends within a 5×5 window, and is won by the player with the highest knot score, which is given by the number of tiles of their colour visited by a single closed knot. For example, Blue wins the game shown in Figure 39 as the longest knot visits 9 blue (dark) tiles but only 7 orange (light) ones.

7 Discussion

Putting this all in perspective, the examples shown above describe game designs from three different but overlapping regions of the conceptual design space: path-based tile games, area-based tile games, and related 3-dimensional dice games. The relationship between the games and these regions of the design space is summarised in Figure 40.

The arrows indicate a logical progression from the ideas found in one game to the next. This does not necessarily mean that the subsequent games were derived from their predecessors – I can not speak for other designers – although I can confirm that the arrows leading to the games of my own design (Vasco, Celtic! and the top two regions) do indicate a chain of derivation.

The design space is visualised as a surface folded back on itself, to save space and highlight region overlap. Black arrows indicate local steps to variant designs within each region, while dotted arrows indicate larger creative leaps between regions; these are the transformational jumps within the design space. Only a subset of games map between each pair of regions.

On a final note, it is worth pointing out that the games listed in this piece are necessarily examples of successful game designs. It would be impractical to also list the multitude of unsuccessful designs that were tried during their development, and in fact impossible to do so for other designers. This introduces some bias into our survey of the game design landscape, as only a few local maxima are considered, but I believe that these still provide useful landmarks for illustrating the concepts being explored.

\[^{19}\text{http://www.nestorgames.com/#celtic_detail}\]
8 Conclusion

The examples above show how a range of interesting designs can be derived from a single starting point, through a systematic exploration of the conceptual design space. The key points are to: 1) identify relevant degrees of freedom, 2) prioritise those regions of the design space most likely to give good results (while keeping an eye open for interesting outliers along the way), and 3) abstracting concepts in new ways so that existing ideas can be tried in new contexts.

The designs listed in Section 5 were all created in a relatively short period around 2008, during an intense exploration of the path- and area-based regions of the conceptual design space. Some of these games will hopefully provide inspiration for further games in the future, as new regions and even design spaces continue to be explored.

Acknowledgements

Thanks to Russ Williams, Richard Reilly and the reviewers for their helpful suggestions. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au
What Makes a Game Good?

Wolfgang Kramer, Kramer-Spiele

This article, first published online in The Games Journal in 2000, lists a number of objective criteria that go beyond personal preferences in deciding whether a given game is ‘good’ or not. These criteria form a set of guidelines indicating the characteristics that a good game should have.

1 Introduction

Games are a matter of taste! The perceived value of a game depends greatly on the individual preferences of those who play it. Some players prefer games of luck; others prefer games of tactics; still others enjoy communicating with fellow players. Then there are those who like games based on reaction, manual skills, or memory, etc. But whether a game is considered good or of little appeal does not depend entirely on personal preferences. There are also objective criteria that must be considered.

1.1 Originality

Any new game must be original. It has to possess elements that have never – or at least not in this particular combination – been part of a game before.

1.2 Freshness and Replayability

The more a game makes its players want to play again, the better the game. An important aspect of this is the course the game takes should be as different as possible each time it is played. A game lacking this quality will soon become boring. A good game will be as exciting each time it is played as it was the first time.

1.3 Surprise

A game should be rich in surprises. Repetition in sequence, progress, and events should be strictly avoided.

1.4 Equal opportunity

At the start of the game, every player should have an equal chance of winning. In particular, the first player should have neither an advantage nor a disadvantage over the rest of the field.

1.5 Winning Chances

A similar rule applies to the end of a game. Every player must have at least a theoretical possibility of winning until the very end. This possibility might be infinitesimal, but it must be present.

1.6 No ‘Kingmaker Effect’

A game loses its appeal if, at any stage, a player who no longer has any hope of winning can somehow determine the winner. This problem arises primarily in strategy games.

1.7 No Early Elimination

All players should be involved in the game until it is almost over. No one should be eliminated until the very end.

1.8 Reasonable Waiting Times

Nothing kills players’ interest as easily as long periods of inactivity while they wait their turns. Chess provides a useful counter-example: a player can use the waiting time to plan his or her next move.

1.9 Creative Control

Any game that is not based on chance must give players the opportunity to affect its progress and direction. Nothing is more boring for a player than the feeling that he or she is being ‘played by the game’ instead of the other way round. A good game should be challenging.

1.10 Uniformity

The title, theme, format, and graphics of a game must give a unified impression.

1.11 Quality of Components

Durability, functionality, and the visual appeal of the materials contribute greatly to the perceived value of a game.
1.12 Target Groups

Games differ in the demands they put on their players. Some games require special skills. It is important for game rules to be consistent. A strategy game, for instance, cannot be influenced in any way by luck. Imagine a player conceiving a plan, deciding on a particular sequence of play, and then having to roll dice in order to execute them. Clearly, the two concepts are at odds.

Although it would seem logical to expect rules to be consistent, there are a great many games whose target groups are not clearly defined. It is often hard to tell whether a game is meant for players interested in strategy, luck, or some combination of the two, or maybe for people who like communication games.

Games of chance must have simple rules and offer few alternative possible moves. This should result in short turns and a generally fast-paced game. Games of strategy, at the other end of the spectrum, should offer abundant alternatives each move. This will let players realise their potential. It must be possible for a player to achieve mastery.

1.13 Tension

Every game has its own unique tension curve. But long periods of relatively low tension must be avoided in any game. Figure 1 is an illustration of a common tension curve.

The graph shown in Figure 2 illustrates two games with multiple tension peaks. Game A with more frequent peaks and less-pronounced valleys is the better, more interesting game.

1.14 Learning and Mastering a Game

Surely it is an advantage for a game to start quickly and be easy to learn, and the clearer and simpler the rules, the better. A game also benefits from incorporating elements that players are familiar with from everyday life. These elements do not have to actually replicate real life; a general similarity or familiar logic will suffice.

Not all games suffer from having complex rules. In general, the more opportunities players have to influence the course of a game, the more readily the players will tolerate a complex set of rules.

1.15 Complexity and Influence

Short, simple games must have short, simple rules. Complex games, on the other hand, may have more complex rules. These concepts are illustrated by the graph shown in Figure 3.

The extent to which a player can influence the game increases along the x-axis, while the complexity of its rules increases along the y-axis. Once we place some games into this coordinate system, we immediately notice a void lying above the diagonal in Region 1. The games are all located below the diagonal in Region 2. The unavoidable conclusion is that complex sets of rules are acceptable only in conjunction with the players’ relatively high level of influence on the course of the game.
2 Are Good Games Necessarily Successful?

Unfortunately, no. There are many good, even great, games that have had little or no success. In Germany, for example, several attempts have been made to market Twixt, Acquire, and Focus, all of which failed, sad to say. There is more to a successful game than just being good. The game must be introduced to the market in the proper way. Marketing and advertising are of the essence, although even those strategies can do little to boost a game that does not reflect current taste. The special ingredients for success that a game needs to start an avalanche and keep delighting people for years to come would seem to consist of timeliness (zeitgeist), intuition, and luck. Never in a million years would I have anticipated the enormous success of such games as Trivial Pursuit, Magic: The Gathering, or Pokémon.

3 Conclusion

All this is not an attempt to instruct you on how to invent a good game. Rather, it is a set of guidelines on the kind of characteristics a good game should have. These two sentences best express the qualities of a good game:

A good game will stay with us all our lives.
A good game makes us long to play it again.

– Wolfgang Kramer

Acknowledgements

This article was originally published online in *The Games Journal* in 2000, where it was translated from German by Anne Kramer.

*Wolfgang Kramer* is a German board game designer who has published over 100 titles, including some of the world’s most popular games, and has won the *Spiele des Jahres* multiple times.

**Address:** Appartement 293, Oskar-Schlemmer-Str. 5, D 70191 Stuttgart, Germany.

**Email:** Kramer-Spiele@t-online.de

---

1 All games mentioned can be found on the BoardGameGeek online database: http://www.boardgamegeek.com
# Table of Contents

**Vol. 2, no. 1 2016**

**Editorial**  
Game Design in the Age of AI. .................................................. C. Browne 3

**Case Study**  
Wrap O-round Weave Five. .................................................. C. Hoff 5

**Nikoli Logic Puzzles**  
Shakashaka. ................................................................. J. Goto and Y. Miyagi 13

**Case Study**  
Eco-Friendly Game Design. ............................................. N. Romeral Andrés and C. Browne 17

**Article**  
Characterising Score Distributions in Dice Games. ......................... A. Isaksen *et al.* 24

**Article**  
Narrative Progression Traits for Role-Playing Games. ..................... S. Samothrakis 38

**Game Design Patterns**  
Bug or Feature? ............................................................. C. Browne 46

**Article**  
Elegant Combat in War Games. ............................................. F. M. Marqués 54

**Maths in Games**  
Graph Theory in Game and Puzzle Design. ................................ D. Ashlock 62

**Case Study**  
Automated Playtesting with RECYCLED CARDSTOCK. ............... C. Bell and M. Goadrich 71

**From the Archive**  
Under the Strategy Tree. ..................................................... R. Abbott 84

---

The front cover shows a rendering of Carl Hoff’s Wrap O-round Weave Five (WOW5) puzzle ring.  
The design and development of the WOW5 are described on pages 5–12.

---

All content © 2016, *Game & Puzzle Design* and QUT.
Credits

Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com

Editor-in-Chief

CAMERON BROWNE
Department of Knowledge Engineering (DKE)
Maastricht University
Maastricht, 6200 MD
Netherlands

cameron.browne@maastrichtuniversity.nl

Liaison Officer

NÉSTOR ROMERAL ANDRÉS
Nestorgames
Spain

Copy Editor

RUSS WILLIAMS
Green Star
Poland

Editorial Panel

INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
CARL HOFF, Applied Materials, USA
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
DAVID PARLETT, Parlett Games, UK
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
SPYRIDON SAMOTHRAKIS, Univ. Essex, UK
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINANDS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent publication prepared and published with the kind help of volunteers. Set in Palatino using \LaTeX.

Published content © 2018, Game & Puzzle Design.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.gapdjournal.com/submit
Game Design in the Age of AI

Cameron Browne, Queensland University of Technology (QUT)

The big news in the game AI community recently has been the victory of Google DeepMind’s ALPHAGO program [1] over top professional Go player Lee Sedol. 19×19 Go is a notoriously difficult game and was seen by many researchers as the holy grail of game AI. Such a victory was not expected for years to come, but it is now clear that computers outperform humans in almost all games of strategy to which they have been applied.

So what effect will such superhuman AI performance have on the games that we design? My answer is: little, if any. Note that I am talking about AI for playing games, not AI for designing games; that is a whole other can of worms, to be addressed in future issues.

Superhuman AI

Since its inception, the field of AI has been inextricably linked with the study of games. Games of pure strategy provide a convenient litmus test for the sophistication of AI approaches, and there has been a constant drive by researchers to outperform previous results on increasingly complex games. One by one we have seen them topple, starting with simple games such as Tic-Tac-Toe, Connect Four, Go-Moku, Nine Men’s Morris, to more notable successes of recent years including Chess, Draughts and now 19 Go.

There is an important distinction between actually solving a game and playing it at a superhuman level. Chess and Go have not been solved, but computer players are now so strong that the distinction becomes less important in practical terms; a human playing against a computer will lose. Affordable Chess programs that run on standard desktop machines now exceed the strength of DEEP BLUE when it famously defeated Kasparov in 1997 and even smartphone apps now play Chess at Grandmaster level.

The game Arimaa was designed specifically to defeat computer players. However, superhuman AI performance was recently achieved even for this game, using standard alpha-beta tree search methods over half a century old [2].

This trend is also true for puzzle games, even nondeterministic ones. Consider the game 2048, in which players slide numbered tiles around a grid such that matching tiles which overlap merge to double their value, while a random tile is inserted with each move. A competent human player will typically achieve the titular 2048 tile and a score in the vicinity of 50,000 points, but an AI recently put such efforts to shame, achieving a 32768 tile and a score of 839,732 [1].

The game of Pylos is an interesting case. While developing an autonomous robot player for Pylos, Aichholzer et al. [3] solved the game by generating a database of all possible positions, from each of which a winning line can be forced if it exists. This will of course defeat any human player, but only if this database of positions (which occupies 30GB of storage) is accessible. Its AI uses a dumb brute force approach with no strategic understanding of the game, which I believe is the crux of the issue.

A criticism of DEEP BLUE was that its strength stemmed from then state-of-the-art hardware, and clever algorithms by its programmers, rather than any inherent strategic understanding of Chess. Similarly, the landmark performance of ALPHAGO was achieved through a clever combination of two ‘black box’ methods – artificial neural networks and Monte Carlo tree search – to produce strong moves that neither the program, nor its programmers, nor its opponents, really understood. It is this missing element of strategic analysis that keeps games interesting for humans.

Jonathan Schaeffer points out that just because a game can be solved does not mean that it is a bad game [1]: ‘What makes a game fun? Intellectual challenge, social aspects, simplicity of the rules, aesthetics... many things.’ Herik et al. [5] observe that a game’s susceptibility to computer solution depends more on its decision complexity than its computational complexity. Decision complexity is related to the strategic depth of a game, which is a vital part of what keeps games interesting for us even after repeated plays.

There is no denying that computer players now outperform us at most games of strategy. But this does not necessarily affect our enjoyment of these games against other human players, only against overly strong AI opponents. A game is only ruined if a winning strategy can be formulated in such a way that the human brain can understand and exploit it, without the need for powerful computers or extensive databases. I reiterate my claim that superhuman AI performance should have little impact on the games that we

---


---

design, if any, provided that those games are designed to be played by humans against humans.

This Issue

This issue covers a range of game types: strategy, war, dice, card and role-playing games; and logic and physical puzzles. We again lead with a piece by Carl Hoff, who again provides the eye-catching front cover image. Carl’s article ‘Wrap O-round Weave Five’ describes his motivation for designing a physical puzzle ring with an interlocking weave around its entire circumference, and how he went about achieving this goal with his ‘WOW5’ ring. This work demonstrates the use of computers to assist the manual design process, rather than their more usual application as combinatorial number-crunchers.

Jimmy Goto and Yuka Miyagi then describe the logic puzzle ‘Shakashaka’, and how its designer’s search for beauty and complexity inadvertently captured a style of mathematical artwork from the 1700s. Shakashaka is this issue’s ‘feature puzzle’, and you will find sample challenges printed throughout, in order of difficulty. Thanks to Nikoli for providing these challenges.

In ‘Eco-Friendly Game Design’, Néstor Romeral Andrés and I show how off-cuts from the manufacture of game components can be up-cycled into new games rather than thrown out as waste. This is demonstrated through several examples from Néstor’s catalogue of games.

For their piece ‘Characterising Score Distributions in Dice Games’, Aaron Isaksen and colleagues use computer analysis to model the effect of different combinations of dice on simulated combat results in dice-based games. They demonstrate that subtle differences in the dice used can have a significant effect on game balance, and discuss how to find the right combination of dice to achieve desired in-game behaviours.

Spyridon Samothrakis then offers a system for classifying role-playing games, based on their game-theoretic outcomes, in ‘Narrative Progression Traits for Role-Playing Games’. He suggests the use of narrative progression traits to help design balanced, interesting new role-play games, based on this classification scheme.

My own article ‘Bug or Feature?’ returns to the computational theme, with a somewhat programmatic approach to game design as an optimisation problem. I demonstrate through example how apparent design bugs can sometimes be recognised as positive features when fully understood, and how actual bugs can often be turned into useful features with simple rule tweaks.

F. Miguel Marqués offers a personal view of how combat mechanisms in war games have evolved over the years to become more elegant and user-friendly, in ‘Elegant Combat in War Games’. Miguel describes ways of simulating the ‘fog of war’ with abstract rules, and extols the virtues of the ‘design for effect’ approach.

Daniel Ashlock’s first instalment in the Maths in Games column, ‘Graph Theory in Game and Puzzle Design’, demonstrates how most games and puzzles played on boards or maps can be decomposed into mathematical graphs. He explores the relationships between games, puzzles and graphs through several examples, and how this can be used to design new games.

Connor Bell and Mark Goadrich’s article ‘Automated Playtesting with RECYCLED CARD-STOCK’ introduces the RECYCLE language for describing card games. They contribute to the research topic of AI for game design by providing a computer system for modelling card games, automatically evaluating them for quality, and assisting the user in designing new games.

This issue concludes with a reprint of Robert Abbott’s classic piece ‘Under the Strategy Tree’ and its later addendum. The addendum is of special note in the annals of game design, as this is where Abbott pretty much nails the concepts of clarity and depth, and the relationship between the two, in three simple sentences.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

2This discussion neatly side-steps the tangential question of what it will actually mean to be ‘human’, when the brain/machine interface develops to such an extent that we become superhuman ourselves.
Wrap O-round Weave Five

Carl Hoff, Applied Materials

Puzzle rings are simple mechanical puzzles, with very few parts, that have been well explored by designers over hundreds of years. This paper describes a recent and rare innovation in this field; a puzzle ring in which the weave wraps around its entire circumference. I describe the design and physical manufacture of such a ring, which produced a surprisingly difficult puzzle to solve.

1 Introduction

There are many types of puzzle rings and their exact history and origins are debatable. This paper will focus on just the subset of puzzle rings that are composed of three or more bands that are interlinked via a weave pattern. This type of ring has been made for hundreds of years with the number of bands varying from three to twelve or more. The advantage of having the bands interlinked is that the bands always remain together, so a single band cannot get lost or misplaced.

Threesome, from the online store Puzzle Ring Maker, is the first puzzle ring design that I was involved with. The weave pattern was proposed in 2007 by Bram Cohen, the inventor of the BitTorrent protocol, but appears to date back much earlier. Oskar van Deventer prototyped this particular ring using the 3D printing service Shapeways. My contribution was to make some simple design tweaks and to adjust the filleting. The final design, shown in Figure 1, was made by Jeff Bell (alias Vardan) who operates Puzzle Ring Maker.

The Threesome design typifies the type of puzzle ring that this paper will focus on. The three bands are all interlinked when scrambled, as seen in Figure 2. This interlinking is accomplished through the use of a weave pattern that occupies the top (i.e. forward-facing part) of the ring, to give it a decoration or feature that orients the ring when it is worn; the weave itself serves the same function as the central setting on a standard ring. The remainder of the bands are typically simple parallel segments that are rather boring compared to the weave pattern on top.

2 Background

While the exact origin of puzzle rings has been lost over time, they were popular as early as the 15th century [2, p. 47], and were often used as betrothal and wedding rings. They are mentioned in Robert Herrick’s poem *The Jinnmall Ring or True-Love Knot* (1648) and John Dryden’s play *Don Sebastian* (1690) and two of William Shakespeare’s plays: *Twelfth Night*, Act 2 Scene 2 (1601) and *Othello*, Act 4 Scene 3 (1603).

---

1 http://puzzleringmaker.com/index.php?ref=item&id=82
2 http://shpws.me/Cte9
3 Thou sent’st to me a true love-knot, but I / Returned a ring of jinnmals to imply / Thy love had one knot, mine a triple tie.
4 A curious artist wrought ’em / With joints so close as not to be perceiv’d; / Yet are they both each other’s counterpart.
5 It is too hard a knot for me to untie!
6 Marry, I would not do such a thing for a joint-ring.

William H. A. Davidson patented a four-band puzzle ring in 1887 [3], which became the most common puzzle ring design of the 20th century. Jose Grant, one of the most prolific puzzle ring designers of the last century [4, 5], discovered puzzle rings in the Middle East and spent many of his later years (1968–2010) developing the concept and inventing many of the designs still found in today’s Jose Grant catalogue [7].

There are currently a handful of puzzle ring designers and avid collectors of the hundreds – if not thousands – of puzzle ring designs available today. Figures 3 and 4 show two different puzzle ring collections.

### 2.1 Motivation

Despite their variety, a common feature of all puzzle ring designs to date is that the interlocking puzzle part forms the frontal decoration while the majority of each band, that sits under or behind the finger, is a plain strip. This raises the question: can a puzzle ring be designed with a weave pattern that wraps around its entire circumference?

A puzzle ring with a complete wrap-around weave would look interesting from any angle, and not require a specific orientation on the finger to show off its ‘puzzle’ nature; this would have obvious benefits for people – like me – who like to rotate the rings on their fingers. This paper describes the steps I took to design, prototype and manufacture such rings for the first time.

### 3 Construction Techniques

Until very recently, just about all puzzle rings were made through a casting process. In this process, moulds are made from an original master, and those moulds are injected with molten wax. The solidified wax replicas of the master are then cast using the lost-wax method. The bands are cast as separate pieces and the unwoven ‘straight’ (i.e. circular arc) parallel section of each band provides a place where it can be cut both for assembly of the puzzle ring and for the removal of material to allow the ring to be resized.

Vardan details the casting process [8] and the assembly and resizing process [9] in his online postings. So these unwoven straight parallel sections contain the portion of the band that is cut and soldered back together.

---

However, this is not the only function that these unwoven straight parallel sections serve. Looking at the Threesome ring, for example, the unwoven parallel sections are where the bands can be pulled apart to allow the ring to be scrambled. It is easy to imagine that if the weave were to wrap all the way around the ring, then the three bands would simply be locked up so that the ring could not be scrambled, which would eliminate its puzzle aspect. For these reasons, all prior puzzle rings of this type isolate the weave to just a fraction of the ring, leaving the rest of the ring as these unwoven straight parallel bands.

4 The Innovation

A designer always desires to do something new. Just because all puzzle rings to date contain this unwoven straight parallel section, must they always? This section of the ring seems less interesting than the woven part. Another problem is that the weave, which is wider and thicker than the rest of the ring, can become uncomfortable or look odd if the ring rotates on your finger. And as a puzzle, the unwoven straight sections can provide too many clues and make its solution too easy, as they must all end up on the same side of the ring in the solved state. These reasons all prompted a desire to design a ring with a weave wrapping all the way around. This way all views of the ring would be visually interesting, any orientation on the finger would be equally attractive and comfortable, and one could expect a more challenging puzzle.

Once research for this project began, it was discovered that the idea of wrapping the weave all the way around the ring was not entirely new. Bram Cohen proposed two weave designs in 2007 which he intended to be wrapped around the entire ring [1]. Still, neither of his designs had been implemented (until now).

The needed spark of innovation was not the only hindrance keeping such puzzle rings from becoming a reality. Puzzle ring producers were not pursuing such designs, as it would require a new set of moulds for each ring size they wanted to offer for sale. Without the straight parallel sec-
tions, there was no portion of the ring where one could remove material to vary the sizing, as all portions of the band were critical to how each band interacted with all the others.

It was not until 3D printing became available that the ability to wrap the weave all the way around the ring became practical. Shapeways’ interlocking metal option, which itself is still a pilot program, is what has enabled the pursuit of this idea. Casting could still be used to produce a one-off ring, but 3D printing is needed to enable any volume.

The way Shapeways makes the ring also has its own set of advantages and disadvantages. The designer creates the model using some computer-aided design software, such as SolidWorks, in which the designs here were created. Rescaling the model digitally in the computer is trivial, so one can easily create a ring of any desired size. Shapeways then takes the model and connects the interlocked bands with sprues, i.e. connecting rods which will form the narrow passages that allow the material to spread throughout the mould. Shapeways next prints the model in wax using a high-resolution printer, which is placed in a container where liquid plaster is poured in around it. After the plaster is set, the wax is melted out, allowing the plaster to serve as a mould into which molten metal is poured.

Once the metal has hardened, the plaster is broken away. This reveals the bands still connected by the sprues. The sprues are then cut away and the puzzle bands can be cleaned and polished. This construction technique requires the puzzle ring to be modelled in the scrambled state, since the bands cannot touch one another or they would fuse into a single solid object. The designer must take great care that the bands are in a valid orientation relative to one another, as it is certainly possible to print the bands in a state that would render the puzzle unsolvable.

These disadvantages are compensated by the bands being created in an already linked state and in the correct size. There is no need for any solder joints, which should result in stronger rings, as solder joints can be weak points.

There are currently four major drawbacks of the 3D printing method, as follows:

1. The use of sprues which connect all the bands means that only a single metal can be used. If one instead wants a ring composed of multiple metals (for example, Vardan’s tri-gold design with yellow, white and rose gold bands, shown in Figure 5), then the true casting process used by professional ring makers is required.

2. The only materials Shapeways currently offers in interlocking metal are brass and silver, so even a gold ring is not yet available with this method.

3. Shapeways only allows up to six interlocking parts, so it is currently not possible to make metal rings with seven or more bands via this method.

4. Shapeways’ design guidelines for interlocking metals require that the clearance between each band be at least 1mm. This is not always possible; for example, I will present a design in Section 7 in which the bands must remain nearly in contact with each other, even in the scrambled state.

Figure 5. Tri-gold puzzle ring by Vardan.

5 The Weave Pattern

The next hurdle in making a puzzle ring, in which the weave pattern wraps all the way around, is finding a weave pattern that will allow this. Oskar van Deventer and Bram Cohen have both worked with puzzle rings for several years and have published many of their findings.

Weave Five, shown in Figure 6 is a ring which van Deventer designed while experimenting with Shapeways’ interlocking metals and presented in August 2015. The weave pattern was designed by Cohen, who specifies his designs as ASCII art, as shown in Figure 7.

10https://www.shapeways.com/materials/interlocking-metal
12http://www.solidworks.com/
13https://www.etsy.com/listing/150546579
This weave pattern is interesting in its own right. It has the special property that the five rings are linked in a loop, as a chain joined end-to-end, and that adjacent bands in the loop are non-adjacent in the ring. This weave pattern stood out as having a potential of working with the wrap-around concept. The fact that five bands formed a linked loop and that the bands themselves were loops made it appealing to stretch the weave itself to also form a complete loop.

The two diagrams shown in Figure 8 are based on Cohen’s ASCII art. The upper diagram shows the typical representation with the straight segments extending to either side of the weave. The lower diagram shows just the weave with the straight segments removed.

Short of 3D printing the ring to see if this weave would allow the bands to be disassembled, an examination of the weave pattern revealed the following points. These created confidence that this ring would work even with the weave wrapped around its entire circumference:

1. The red band (V) appears to be in front of the other band except for a 60° segment of the ring’s circumference. It was believed that this band should be able to pivot up (i.e. out of the page) and out of the way of the others, with the axis of rotation being on a line between the tips of the two arrows seen in Figure 9 (upper).

2. The blue band (I) which is an identical copy of the red band (V) should pivot up and out of the way for a similar reason.

3. Without the red (V) and blue (I) bands in the way (Figure 9, lower), the orange band (II) is in front of the remaining bands, except for a single crossing with the green band (IV). Similarly, the green band (IV) is behind the others, except at that same crossing.
6 Printing and Testing

I modelled this chosen weave pattern in SolidWorks, as shown in Figure 10, rendered in polished copper, satin copper, polished gold, satin gold and polished silver. But the only way to be certain that the design would work was to make an actual copy.

The initial prints of WOW5 worked exactly as designed. Shapeways’ design specifications for their Strong & Flexible Plastic[15] allowed for the ring to be printed in the solved state with only a 0.2mm tolerance between the bands. A print in coloured Strong & Flexible Plastic is seen in Figure 11. These prints have severely humbled their designer. While they come apart as designed, they have proven extremely difficult to solve.

7 The First Metal WOW5

With the design proven, a sample in interlocking polished brass was ordered from Shapeways. Due to the different design specifications for this material, the ring needed to be printed in the scrambled state. Therefore, it was already scrambled when it was received on 13 January 2016, as shown in Figure 12.

It seemed destined to stay scrambled until two puzzle ring enthusiasts from Virginia, whose extensive collection of puzzle rings is shown in Figure 3 but who wish to remain anonymous, solved the WOW5 on 22 February 2016 and presented a solution video[16] to receive a prize of a brass copy of the ring. However, even with a solution video to follow, WOW5 proved extremely difficult to solve.

This prize represented the first metal copy of the WOW5, and the satisfaction that such an object was not only physically possible to construct but had a workable solution. The prize was so well received by its recipients that they solved the scrambled ring before leaving the parking lot after picking it up, as shown in Figure 13 and Figure 14.

---


[16]Please email me directly for a link to the solution video.
provides additional views of a solved brass copy of the WOW5.

![Figure 10. Views of the solved brass WOW5.](image)

Figure 13. First metal copy of the WOW5.

Figure 14. Views of the solved brass WOW5.

8 Other Designs

The concept of wrapping the weave all the way around a puzzle ring is indeed viable. It not only can be used to create fiendishly difficult puzzle rings but gorgeous works of art as well. But this area of exploration opens up many more questions. Was it extremely lucky that the weave five pattern allowed this wrap around concept to work? Must it only produce incredibly difficult puzzles? The answer to both questions is, fortunately, no.

Remembering that Cohen had proposed some three-band wrap-around designs, those have now been modelled as well. The first design, Three Phase, makes a very nice and beginner’s level puzzle ring. Figure 15 shows a polished brass copy of Three Phase on the author’s finger.

Figure 15. Render of the Three Phase design.

The second design, Borromean Rings, is seen in Figure 16. This design has been tested in laser-sintered nylon and works as a beginner’s level puzzle ring. However, due to the nature of this weave pattern, the bands remain in near contact with each other even when scrambled. So far it has not been possible to create the 1mm clearance between the bands that is required before this ring can be printed using interlocking metals.

More complex weave patterns have also been explored. WASP, Wrap Around Sixth Perception, is a wrap-around ring using the same weave pattern as Oskar van Deventer and Bram Cohen’s design Sixth Sense. Shapeways has now printed a sample in laser-sintered nylon, which can be scrambled. A rendering of the WASP model is shown in Figure 17. The model for WASP in the scrambled state must be created before it can be printed in metal and that work is ongoing in parallel with the writing of this paper. Considering how difficult solving WOW5 proved to be, it is expected that WASP is even harder to solve.

Figure 16. Render of the Borromean Rings design.

9 Conclusion

In closing, this exploration in the area of wrap-around puzzle rings has been very rewarding. The final design of the WOW5 provided both a functional and elegant solution to my original question of whether a puzzle ring could practically be designed with a complete wrap-around weave.

Several new puzzles were created throughout the process, and have been well received within the puzzle community. WOW5 itself received a warm welcome when presented at G4G12, the 12th Gathering for Gardner [8].

Acknowledgements

Special thanks to Bram Cohen and Oskar van Deventer for their body of work on puzzle rings, and their support of this exploration of the wrap-around concept. Thanks to Varden for his offer to proof-read this article and for being willing to cast and finish some of these designs. With his help, tri-gold samples of some of these designs, including WOW5, should exist in the near future.

Thanks to the two unnamed puzzle enthusiasts from Virginia known for their collection of puzzle rings and their 50+ years of puzzle ring solving experience. Without their solution video of WOW5, the metal copies of WOW5 would likely all still be scrambled. Their willingness to proof-read this article is also appreciated. Varden has also produced a solution video, which is easier to follow and available upon request.

Thanks to those who gave permission for the photos used in this article, especially Oskar van Deventer for Figures 1, 2 and 6. All images were used with permission from their owners.

Finally, thanks to Toshia Wrenn, who helped me explore another of Shapeways’ pilot materials: 3D printed aluminium. Toshia conducted some tests on an aluminium copy of WOW5 to see if the bands could be anodised different colours. Unfortunately, the 3D printed aluminium was too grainy to form nice nano layers, so random oxidation just led to a darkening of the bands.

References


Carl Hoff is an epitaxy process service engineer for Applied Materials currently supporting Global Foundries in Malta, New York, United States. Research interests include twisty puzzle design and theory, packing puzzles, and puzzle rings. Address: 1103 Lofts Way, Malta, NY, 12020-5210, USA. Email: carl.n.hoff@gmail.com
Shakashaka

Jimmy Goto, Nikoli

Yuka Miyagi, Nikoli

Shakashaka is a logic puzzle in which the solver must colour black triangles within a square grid, in order to create white squares and rectangles according to certain constraints. This paper gives a brief account of the history and design of this delightfully geometrical puzzle.

1 Introduction

Shakashaka is a pure deduction puzzle, invented for Nikoli by Japanese designer Guten, which first appeared in the June 2008 issue of Puzzle Tsushin Nikoli 123.

Shakashaka initially caused some disagreement among our editing section as to its longevity when first published, but it has since been enthusiastically embraced by our readers, especially young female readers, who refer to the puzzle as ‘Kawaii’ after a favourite mosaic-style game, which translates as ‘cute’ in English. It is now so popular that it has become a regular feature of Puzzle Communication Nikoli, from volume 127 onwards, a year after it first appeared.

2 Rules

Shakashaka is played on a grid of squares, with some cells initially coloured black, some of which are marked with hint values. Figure 1 shows a typical challenge.

There are four ways to fill a coloured a triangle between opposite corners of a cell, as shown in Figure 2.

Figure 2. The four triangle rotations.

The aim in Shakashaka is to draw such triangles in some of the empty cells, so that:

1. hint cells are immediately adjacent to the number of triangles indicated, and
2. all remaining white regions form squares or rectangles.

Figure 2 shows the solution for this challenge.

Note that the final squares and rectangles may be angled diagonally, and that some grid cells may not need to have a triangle drawn in them to solve the challenge. Hint cells without a number can have any number of adjacent triangles.

3 Worked Example

A basic tactic for Shakashaka is: do not create acute angles. For example, Figure 4 shows the 2 hint cell in the top left corner of the above challenge (upper), and the four possible triangle rotations

Figure 3. Unique solution for this challenge.

for the cell below (lower). We know that the cell below this hint must take a triangle, as the hint value 2 and it only has two neighbours. Note, however, that only the leftmost triangle can possibly be drawn in this cell, as all other rotations create white regions with acute $45^\circ$ angles that could not possibly form part of a square or rectangle.

Figure 4. The cell below the 2 hint must take the triangle in the leftmost rotation.

Figure 5 (a) shows this tactic applied to the challenge, and similar deductions applied to the top row. The cell to the right of the 1 hint must be empty as it now has a triangular neighbour, and both cells adjacent to the 0 hint must also be empty. Such empty cells are marked with a dot (●) for convenience. From here, it is possible to make further similar deductions (b) and it becomes obvious that the white region in the upper left must be an angled rectangle (c).

Figure 6 shows similar deductive steps (d), (e) and (f) that yield the solution to this challenge.

4 Design

The puzzle’s designer, Guten, explains that the name ‘Shakashaka’ comes from the Japanese term shakaku (斜角), which means ‘oblique angle’. This was the original name of the puzzle, but was changed to the catchier derivation ‘Shakashaka’ when submitted to Nikoli for publication.

Guten was initially worried that the puzzle may not have enough depth or originality to satisfy hard-core puzzlers, but has been surprised by its subsequent popularity and the positive feedback received from players. An interesting phenomenon he has noticed is that some younger players can solve Shakashaka challenges intuitively and surprisingly quickly, whereas older (more experienced) players tend to approach it too logically and take much longer.
Shakashaka was originally designed to only involve diagonal white regions in the final solution, but Guten changed this to also include orthogonal white regions as soon as he saw that the two worked in conjunction, for greater variety. His focus at all times of the design process was to provide the most entertaining challenges for players.

We personally find Shakashaka charming in the way that it reveals its hidden geometric shapes; we feel like archaeologists excavating ancient ruins to reveal the footprints of lost buildings, stadia, squares, and so on. We appreciate this blending of the aesthetic with the logical in a puzzle design.

5 Truchet Tiles

The shape formed by a triangle half-colouring a square cell between opposite corners, shown in Figure 2, is known as a Truchet tile. This was named after French clergyman Sébastien Truchet (1657–1729), who proposed the use of such tiles for decorative purposes in a short 1704 paper [1], in which he provided a combinatorial analysis of the many aesthetically interesting patterns that can be created with the four possible orientations of such a tile. Figure 7 shows a detail of one of Truchet’s tilings from [1].

Figure 7. Truchet tiles and a Truchet tiling.

Figure 8 (top) shows a modern development of the Truchet tile by metallurgist and historian Cyril Stanley Smith [2] known as the Truchet-like tile, which consists of two circular 90° arcs with radius equal to half the tile width, centred at opposite corners of the tile. Random tilings of these Truchet-like tiles form visually interesting sets of contours, as shown in Figure 8 (bottom), and are commonly used in computer graphics to create easily generated background textures.

The Truchet-like tile design can be found in board games including Meander, Black Path [13], vol. 2], Trax, hexagonal variants such as Mambo and Palago [4], and many others [1]. There even exist several US patents for games with Truchet-like tiles or pieces [5].

It is therefore interesting that Guten returns to the original formulation of Truchet’s tile for Shakashaka rather than the more commonly used Truchet-like variant. Guten did not know of Truchet tiles when inventing Shakashaka, but says that he would have been tempted to find a different basis if he had, probably resulting in a less minimalist and elegant design.

6 Conclusion

Shakashaka is a logic puzzle with a strong geometrical nature. The fact that players build shapes as they solve each challenge seems to have broader appeal beyond the usual hard-core puzzler demographic.

There are now two published Shakashaka collections from Nikoli [6, 7]. Appendix A shows

\[http://boardgamegeek.com/geeklist/54579/games-inspired-truchet-tiles\]
an example $20 \times 10$ challenge, by the game’s designer, from the first collection. Further examples can be found throughout this issue as our ‘feature puzzle’, in approximate order of difficulty.

Acknowledgements

Thanks to the Editor-in-Chief for assembling this piece from the notes provided and redoing the figures. Thanks also to Yoshinao Anpuku for comments and the reviewers for their suggestions.

References


Jimmy Goto works for Japanese quarterly magazine Puzzle Communication Nikoli, which specialises in language-independent logic puzzles. 
Address: Nikoli Co., Ltd., 4-17-10 Kuramae Taitou-ku, Tokyo 111-0051.
Email: goto@nikoli.co.jp

Yuka Miyagi also works for Japanese publisher Nikoli. Address: Nikoli Co., Ltd., 4-17-10 Kuramae Taitou-ku, Tokyo 111-0051.
Email: miyagi@nikoli.co.jp

Appendix A: $20 \times 10$ Shakashaka Example

This appendix shows a $20 \times 10$ Shakashaka challenge by Guten, from Shakashaka 1 [6].

Figure 9. An example $20 \times 10$ Shakashaka challenge by Guten from [6] © Nikoli.
Eco-Friendly Game Design

Néstor Romeral Andrés, Nestorgames

Cameron Browne, Queensland University of Technology (QUT)

Most manufacturing processes create waste by-products, and games are no exception. This article demonstrates how such by-products can be successfully ‘upcycled’ and reused as game components in their own right, through several practical examples. We distinguish between strong and weak upcycling, and show that both approaches have ecological benefits in reducing waste material and economic benefits in reducing manufacturing costs.

1 Introduction

UPCYCLING, also known as creative reuse, is the practice of recycling waste by-products into new high-quality goods, such that ‘old products are given more value, not less’ [1]. This practice of minimising and effectively utilising wastage is not only an environmental matter, but can have economic benefits [2]. Large companies that manufacture in volume can save significant amounts of money, and it can mean the difference between success or failure for small companies working on tight margins; ‘eco’ here stands for both ‘ecological’ and ‘economical’.

This is very true of game manufacturing, in which the production of each component will typically produce superfluous by-products. However, there is constant pressure on game publishers to release new designs with novel components, which encourages the opposite of reusing old products. So how can game designers implement upcycling to benefit from it?

We identify three levels of upcycling:

1. **Strong**: By-products are used exclusively to make new products.
2. **Weak**: By-products are used in new products along with other parts.
3. **False**: By-products are used in new products along with other parts that create further by-products.

Strong upcycling, which effectively produces zero wastage, is obviously the ideal form to be pursued. Weak upcycling is still desirable but dilutes the benefits, while false upcycling offers few real benefits.

This paper presents case studies of game designs that demonstrate the principles of strong and weak upcycling. All examples are published by Nestorgames, an independent board game company owned and run by the first author.

2 Strong Upcycling

The first example shows how the manufacture of one particular game created by-products that led inadvertently – but almost inevitably – to the design of another game.

2.1 Stax

Stax, shown in Figure 1 is a tile placement game invented by the second author in 2010 [1].

Each tile is a hexagon with a circular hole cut out and embellished with 0 to 6 notches or pointers along some of the six hexagonal axes. Each player has 13 such tiles of their colour, representing all unique combinations of 0 to 6 notches.

The rules of Stax are not important here; suffice it to say that for the game to work: 1) the notches on each tile should be visible from both sides, and 2) players should be able to see ‘through’ tiles stacked on others to check the notches of the tiles underneath.

Laser cutting the notched holes out of the tiles was the obvious solution. This produces attractive pieces that fit the game’s specification, at a
fraction of the cost that laser-engraving both sides of each tile would have incurred.

2.2 Manufacture

![Figure 2. Template for laser cutting Stax pieces.](image)

Stax sets are manufactured by laser cutting pairs of acrylic sheets in the appropriate colour, glued together to achieve the appropriate thickness. Figure 2 shows the path template provided to the laser cutter.

The hexagonal tile shape allows an efficient packing that minimises wastage, and the unused outer areas are reused for cutting components for other games. However, an unavoidable by-product produced by the cutting process is the collection of notched holes, as shown in Figure 3.

2.3 By-Product: Xats Pieces

As Stax began selling, the number of notched hole off-cuts began to pile up; one for every Stax tile produced. However, the off-cut pieces are themselves attractive, with the inverse hole notches forming organic-looking spiky arms. They immediately say ‘game component’ to any game designer who sees or plays with them.

So how could this waste by-product be reused in a game rather than simply thrown out? We started by giving the set of pieces a name – ‘Xats’, the inverse of ‘Stax’ – and each individual piece an identifying label, as shown in Figure 4.

![Figure 3. Off-cuts produced by the manufacture of Stax.](image)
An obvious use of the pieces is as a packing puzzle. For example, the pieces can be packed to completely fill the shape shown in Figure 5 if the \textit{a} piece is removed and the \textit{b} piece is duplicated to give the required number of spiked arms.

However, this exact puzzle has prior art. It was released pre-1960 as Kwazy Quilt\textsuperscript{2} and later as Beat the Computer No. 0\textsuperscript{3}. This ruled out Xats as a packing puzzle, which would have been less desirable anyway, as its production would have created even further wastage in cutting the puzzle’s frame.

Another possible use of the Xats pieces was to simply reuse them to play Stax; they are after all the right shape, number and colour. However, this soon proved impractical as the smaller pieces did not stack well, and were difficult to place so that they lined up in a regular grid, without the hexagonal outer frames of the Stax pieces.

Instead, the first author sought to devise a game using the Xats pieces that satisfied the following constraints:

1. \textit{All} of the off-cut pieces should be used.
2. No other material should be used, not even a board (pure recycling)\textsuperscript{4}.
3. The game should be reminiscent of its parent game Stax.
4. It should be a brain burner that shows the potential of using spare parts.
5. It should be innovative; we want to recycle parts, not ideas.
6. It should allow the in-game recycling of pieces, e.g. through capture and refund, in keeping with its heritage\textsuperscript{5}.

The following section describes the first author’s design process that led to the game of Xats.

### 2.4 Pieces In Search of a Game

The first thing to note is that the Xats tiles, with their spiky arms, do not tessellate neatly. Pieces would be difficult to line up and rotate accurately in relation to each other on an unconstrained playing surface, making it difficult for players to see movement or attack lines. The game should therefore involve independent piece placement; further, it should involve independent piece \textit{stacking} to reflect its parent game Stax.

The spiked arms on the pieces are their most prominent features, so should be incorporated as an important feature of the game. This is done in two ways:

1. Number of spikes (\textit{cardinality}).
2. Distribution of spikes.

An obvious mechanism suggested itself at this point: to stack pieces on existing pieces, such that all arms match the distribution of arms of the piece(s) underneath, i.e. no arm can overhand empty space.

<table>
<thead>
<tr>
<th>\text{a}</th>
<th>\text{b}</th>
<th>\text{c}</th>
<th>\text{d}</th>
<th>\text{e}</th>
<th>\text{f}</th>
<th>\text{g}</th>
<th>\text{h}</th>
<th>\text{i}</th>
<th>\text{j}</th>
<th>\text{k}</th>
<th>\text{l}</th>
<th>\text{m}</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Table of ‘can be placed on’ relations.

\footnote{2http://www.jaapsch.net/puzzles/kwazy.htm}
\footnote{3http://www.huybers.net/poly/btc.html}
\footnote{4Apart from a container to store the set. Nestorgames uses organic cotton bags from a ‘fair trade’ source.}
\footnote{5Although such recycling rules must, of course, avoid infinite cycles in play.}
Table 1 indicates which pieces can be placed on which others, according to this mechanism. The grey dots along the diagonal represent pieces placed on their own shape (which were excluded from the rules to move games towards a conclusion). This information is presented in a different form in the coverage graph shown in Figure 6 in which the directed lines indicate which pieces can be placed on which others. It is up to the reader which format is more visually meaningful; a study in itself!

![Figure 6. Coverage relationships between pieces.](image)

While these analyses tell us something about the pieces, the dependency graph shown in Figure 7 reveals the surprisingly ordered logical structure inherent in the arm distributions. It was obvious that some pieces were more important and some less important than others, in terms of the number of pieces that they can stack on and that can stack on them, which boded well for their strategic potential as a game set.

![Figure 7. Dependency graph of piece coverage.](image)

2.5 Rules of Xats

A simple set of rules was initially devised:

Each player starts with a full set of Xats pieces in their colour in hand.

Players take turns placing a tile in hand onto the playing area, to either:

1. Stack on an existing tile with a greater cardinality, such that all spikes are matched.

2. Start a new stack if no stacking move can be made.

The game’s winning condition, based on cardinality, was always as follows:

A player wins if, at the end of their turn, the number of their tiles in hand equals the total cardinality of their topmost pieces in play.

For example, Figure 8 shows a game in action between White and Blue. Blue, whose turn it is to play, can create a new stack as they are not able to play on any existing stacks. The indicated move will win, as Blue will then have five pieces in hand and the total cardinality of topmost blue pieces in play will equal $1 + 4 = 5$. Blue also has one other winning move.

![Figure 8. A winning move for Blue.](image)
This initial rule set played well and satisfied most of the desired design constraints. It used all of the Xats pieces and no other material, was reminiscent of the stacking game Stax, was something of a brain burner, and was novel as far as we could determine. However, it did not allow the in-game recycling of pieces. To this end, the following recycling rule was added:

Each turn, the mover may either play a tile (as described above) or destroy a stack by:

1. Selecting a piece in hand whose cardinality equals the height of a stack.
2. Discarding that piece from the game forever.
3. Returning the pieces in the stack to the hands of their owners.

This combination of rules worked well, and is now the official Xats rule set.

2.6 Discussion

The recycling of destroyed stacks back to their owners resonates nicely with the eco-friendly ethos of the game, but also adds to the game by extending its duration and depth, and allowing powerful defensive moves that effectively reset the spike count.

The fact that a piece is ‘burnt’ with each destruction avoids infinite cycles in play. This also adds a nice element of balance to the game, as higher towers require pieces of greater cardinality to be burnt, yet those are the most valuable pieces as they allow free moves onto the table (rather than forced stacking moves) more often. An annotated sample game is provided on the journal’s web site, to give a feel for Xats and demonstrate some of the relevant tactics and strategies.

Xats succeeds in its design goals and demonstrates in-game recycling of its fully recycled pieces. The piece shapes themselves encode certain properties (colour, cardinality, distribution) that allow simpler rules, in a classic case of poka-yoke or embedding the rules in the equipment.

In terms of efficiency, reusing the Stax off-cuts as Xats pieces saves the approximately 20% of material that would otherwise be wasted. However, the true saving is in the reduced cutting time, which is where the real expense lies, as we now get two game sets cut for the price of one. A three-player version of Xats is provided, so that pieces in all three of the Stax colours – white, blue and red – can be reused.

Recall that upcycling occurs when ‘old products are given more value, not less’ [1], which is certainly the case here. The Stax off-cuts are not recycled into plastic waste to be melted or ground down into base material for new products, but are instead upcycled, whole and complete, into an interesting new game. It could even be that Xats is a better game than Stax, although this is hard to gauge using sales figures, as the number of Xats sets available depends – by its very nature – on the number of Stax sets produced.

3 Weak Upcycling

Our second example demonstrates what we call weak upcycling, i.e. the creative reuse of waste by-products in conjunction with other non-recycled parts to create a new product.

3.1 Nestorings

Figure 9 shows the Shibumi board, another game designed by the second author and published by the first author in 2011. These boards are again manufactured by laser-cutting acrylic sheets, which creates two by-products per 22mm hole: a 17mm internal disk which is used as a game piece in other games (another example of recycling off-cuts), and a 22mm surrounding ring with no immediate use. We internally called these surplus rings nestorings for convenience.

Figure 9. Shibumi board, disk and surplus ring.

16 such surplus rings are created for each 4×4 Shibumi board manufactured, and, even worse, 36 are created for each 6×6 Margo board of similar design. The rings soon started to pile up and represent a significant amount of wastage.

---

6http://www.nestorgames.com/#xats_detail
8http://www.nestorgames.com/#shibumisamurai_detail
3.2 Initial Attempts

Surplus rings were first used in the game Crosshairs\(^9\) to support elevated pieces representing fighter planes in action, as shown in Figure 10.

![Figure 10. Rings support planes in Crosshairs.](image)

They were also used for the game Line or Colour\(^10\) shown in Figure 11 which features a Yinsh-like mechanism in which the rings influence smaller playing pieces that they surround.

![Figure 11. Line or Colour uses nestorings.](image)

However, neither of these uses solved the surplus ring problem. Crosshairs only used a small number of rings compared to the amount that was piling up, and Line or Colour required an equal number of rings to be cut in another colour, reducing its benefit as a wastage solution.

Another approach tried was to use the rings to create a configurable playing board for the game Quantum Leap\(^11\) shown in Figure 12. This did use a lot of surplus rings per set, but being a new and relatively expensive game, it saw little uptake. Nestorgames therefore looked to other game designers for a more permanent solution to the problem, by running a game design contest.

![Figure 12. Quantum Leap board made of rings.](image)

3.3 Rings Game Design Contest

The Nestorgames Rings Game Design Contest was run on the BoardGameGeek web site in 2014 and open to all game designers\(^12\). The contest received 16 entries, three of which were selected for publication: Po-Go, Particle Accelerator and Go with the Floe, shown in Figures 13 – 15.

![Figure 13. Po-Go.](image)

Po-Go and Particle Accelerator were the most successful in terms of recycling, as they reused a reasonable number of rings and did not introduce new parts beyond custom boards (which do not produce wastage) and stock game components. Go with the Floe used a greater number of surplus rings per set, but required the additional manufacture of custom animal-shaped pieces.

---

9http://www.nestorgames.com/#crosshairs_detail
10http://www.nestorgames.com/#lineorcolour_detail
11http://www.nestorgames.com/#quantumleap_detail
12https://boardgamegeek.com/thread/1221878/nestorgames-rings-game-design-contest
This combined approach of several weak upcycling solutions proved effective in this case. Not only is the Nestorgames surplus ring problem now under control, but we also have several new games that might not have existed if this problem had never occurred.

4 Conclusion

The paper describes Nestorgames’ eco-friendly approach to game design, in both an ecological and an economic sense. We demonstrate that strong upcycling of surplus by-products is both achievable and beneficial in the context of board games, and that weak upcycling, though less efficient, can still be worthwhile if multiple partial solutions are found.

We suggest the following basic steps for game designers wishing to repurpose by-products of their own manufacturing processes:

1. Identify by-products of interest.
2. Focus on distinguishing features.
3. Map these to game concepts.

Finding workable solutions to such waste management problems – i.e. turning garbage into games – is not only an interesting puzzle in itself, but can be most satisfying in the results that it produces. Designing under such rigid constraints can have the counterintuitive effect of actually stimulating greater creativity in the designer and lead to pleasant surprises [2]. While we may not be saving the planet by reusing some bits of plastic, we find upcycling a good philosophy that should be followed more often.

Acknowledgements

Thanks to the reviewers for their suggestions. Cameron Browne is supported by a QUT Vice-Chancellor's Research Fellowship.

References


Néstor Romeral Andrés is the owner and director of independent board game publisher Nestorgames. He has designed over 100 board games and was awarded Spanish Game Designer of the Year for 2009–10. Address: c/Mar de las Antillas 6, 3-2, Zaragoza, 50012, Spain. Email: info@nestorgames.com

Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design. Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia. Email: c.browne@qut.edu.au
We analyse a variety of ways that comparing dice values can be used to simulate battles in games, measuring the ‘win bias’, ‘tie percentage’, and ‘closeness’ of each variant, to provide game designers with quantitative measurements of how small rule changes can significantly affect game balance. Closeness, a metric we introduce, is related to the inverse of the second moment, and measures how close the final scores are expected to be. We vary the number of dice, number of sides, rolling dice sorted or unsorted, biasing win rates by using mixed dice and different number of dice, allowing ties, re-rolling ties, and breaking ties in favour of one player.

1 Introduction

Dice are a popular source of randomness in games. We examine the use of dice to simulate combat and other contests. While some games have deterministic rules for exactly how a battle will resolve, many games add some randomness, so that it is uncertain exactly who will win a battle. In games such as Risk, two players roll dice at the same time, and then compare their values, with the higher value eliminating the opponent’s unit. Other games use a hit-based system, such as Axis and Allies, where a die roll of a strength value or less is a successful hit, with stronger units simulated by larger strength values and larger armies rolling more dice. In both games, stronger forces are more likely to win the battle, but lucky or unlucky rolls can result in one player performing far better, making a wide difference in scores.

Given a very large number of games played between players, unlucky and lucky rolls will balance out such that the player who has better strategy will probably end up winning; however, people might not play the same game enough times for the probabilities to even out. Instead, they play a much smaller, finite number of rolls spread across one session, or perhaps a couple of play sessions. The gambler’s fallacy is the common belief that dice act with local representativeness: even a small number of dice rolls should be very close to the expected probabilities. Therefore, it can often be frustrating when rolling poorly against an opponent: players often blame the dice, or themselves, for bad rolls, even though logic and reason dictate that everyone has the same skill at randomly rolling dice. Game designers may want to avoid or reduce this kind of negative player experience in their games.

Although there are thousands of games based on dice (the BoardGameGeek online database lists over 7,000 entries for dice games, and hundreds of games are described in detail in [6, 7]), we specifically examine games where players roll and compare the individual dice values, as in Figure 1. Each player’s dice are sorted in decreasing order and then paired up. Whichever player rolled a higher value in a pair wins a point. The points are summed, and whoever has more points wins the battle.

![Figure 1. An example of a dice battle.](image)

We use the term battle to denote any event resolved randomly within a larger game. The word is normally used to refer to combat, but our analysis can be used any time players compare dice outcomes in a contest. In this paper, we will use the terms battle and game interchangeably.

---

1 https://boardgamegeek.com/boardgametopcategory/1017/dice

We examine different variants and show how different factors affect the distribution of scores and other metrics which are helpful for evaluating a game. By adjusting the dice mechanics, a designer can influence the:

1. expected *closeness* of battle outcomes,
2. *win bias* in favour of one of the players, and
3. *tie percentage*, i.e. the fraction of battles that end in a tie.

The variants we examine include: different numbers of dice; different numbers of sides per die; different ways to sort the dice; and various ways to break ties.

Dice have come with various numbers of sides for millennia [8]: some of the oldest dice, dating back to at least 3,500BC, were bones with four flat sides and two rounded ones. Eventually, 6-sided dice were created by polishing down the rounded sides. The dot patterns we see on today’s 6-sided dice also come from antiquity. Ancient dice also come in the form of sticks with four long sides for Pachisi or two long sides for Senet [9]. The common dice in use for modern games are 4-, 6-, 8-, 10-, 12- and 20-sided, but other variants exist. In this paper, we use the notation \( ndk \) to mean a roll of \( n \) dice with \( k \) sides each, e.g. \( 5d6 \) means a roll of five 6-sided dice.

By understanding how rules and randomness affect closeness, a designer can then choose the appropriate combination to try to achieve their desired game experience. A designer may prefer for their game to be highly unpredictable with large swings, intentionally increasing the risk for players to commit their limited resources. In addition, randomness can make a game appear to be more balanced because the weaker player can occasionally win against the stronger player [10]. Large swings may be more emotional and chaotic, and the ‘struggle to master uncertainty’ can be considered ‘central to the appeal of games’ [11]. Or, a designer may prefer for each battle to be close, to limit the feelings of one side dominating the other in what might be experienced as unfair or unbalanced, in a trait known as *inequity aversion* [12][13]. Similarly, a designer may prefer to allow ties (simulating evenly matched battles), or wish to eliminate the opportunity for ties (forcing one side to win). Finally, a designer might want to vary the rules between each battle within a game, to represent changing strengths and weaknesses of the players or to provide aid to the losing player. A designer can adjust randomness to encourage situations appropriate for their game.

For most sections in this paper, we calculate the exact probabilities for each outcome by iterating over all possible rolls, tabulating the final score difference. Because each outcome is independent, we can parallelise the experiments across multiple processors to speed up the calculations (details about the calculations are given in the Appendix). There are other methods one could use to computationally evaluate the odds, such as a dice probability language such as AnyDice\(^2\) or Troll [14], or by using Monte Carlo simulation (we use simulation when examining re-rolls in Section 8). Writing the analytical probabilities becomes difficult for more complex games, and we feel that presenting such equations is less useful for most game designers.

### 2 Metrics for Dice Games

Quantitative metrics have been used to computationally analyse outcome uncertainty in games, typically for the purpose of generating novel games [15][16]. Here we focus only on metrics that examine the final scores of the dice battle; we do not evaluate anything about how scores evolve during the battle itself (which we believe would be essential for more complicated games). But for simple dice battles, which are a component of a longer game, we can just focus on the end results.

*Win bias* and *tie probability*, are similar to those used in previous work, but one of our metrics, *closeness*, is something we have not seen used before in game analysis. We now define these metrics precisely.

#### 2.1 Score Difference

Let \( s_A \) be the final score of a battle for Player A, and \( s_B \) be the final score for Player B. The *battle score* or *score difference*, \( d \), is given by \( d = s_A - s_B \). If we iterate over all the possible ways that the dice can be rolled, counting the number of times that each score difference \( d \) occurs, we can derive a *score difference probability distribution*, \( D(d) \). This describes the probability of achieving a score difference of \( d \) in the battle. We calculate \( D(d) \) by first counting every resulting score difference in a histogram-like data structure, and then dividing each bin by the total sum of all bins.

#### 2.2 Win Percentage

The *win percentage* is the percent probability of Player A winning a battle. This can be calculated by summing the probabilities where the score difference is positive and is therefore a win for

\(^2\)http://anydice.com
Player A. This is calculated as $100 \sum_{d>0} D(d)$ and will be between 0 and 100. Loss percentage is the percent probability of Player A losing a battle, and is calculated as $100 \sum_{d<0} D(d)$, also between 0 and 100.

2.3 Win Bias

The difference of the win and loss percentages gives the win bias:

$$\text{win bias} = 100 \left( \sum_{d>0} D(d) - \sum_{d<0} D(d) \right)$$

This will range from -100 to 100. Games with a win bias of 0 are balanced, with no preference of Player A over Player B. If the win bias is $> 0$, then Player A is favoured; if $< 0$, then Player B is favoured. This metric is similar to the balance metric in [15], but here we include the effect of ties and are concerned with the direction of the bias. A non-zero win bias is often desired, for example, when simulating that one player is in a stronger situation than the other.

2.4 Tie Percentage

Tie percentage indicates the percent probability of the battle ending in a tie, which we define as:

$$\text{tie percentage} = 100 \left( D(0) \right)$$

Some designers may want a possibility of ties, while others may not. This metric is analogous to drawishness in [15].

2.5 Closeness

Finally, we present closeness, our new metric which measures how much the final score values centre around a tied game. Games that often end within 1 point should have higher closeness than games that often end with a score difference of 5 or -5. The related statistical term precision is defined as the inverse of the variance about the mean. For closeness, we define this as the square root of the inverse of variance in the score difference distribution about the tie value $d = 0$:

$$\text{closeness} = \frac{1}{\sqrt{\sum d^2 D(d)}}$$

To explain this, we look at the denominator, which is similar to the standard deviation, i.e. the square root of the variance. However, we do not want this to be centred about the mean, as in the typical formulation. A game that always ends tied 0-0 would have a variance of 0, but so would a game that always ends in a 5-0 win, because the outcome (mean) is always the same. Yet 5-0 is certainly not a close score. Thus, we centre the second moment around 0 since close games are those with final score differences close to 0. Finally, we take the inverse because we want the metric to increase as the scores become closer and to decrease as the scores become further apart.

This formulation captures the familiar notion of a ‘close game’ and hence has some intuitive meaning. Closeness approaching 0 means that the final score differences are very spread out. Closeness approaching $\infty$ means the scores are effectively always tied. A closeness of $C$ means that a majority of the score differences will fall between $-1/C$ and $1/C$. If a game can only have a score difference of -1 or 1, its closeness will be exactly 1, no matter if it is biased or unbiased. If tie scores are also allowed, i.e. score differences of -1, 0, or 1, then we would expect the game’s closeness to be higher – and indeed this is the case, as closeness will always be $> 1$.

3 Rolling Sorted or Unsorted

Many games ask the players to roll a handful of dice. A method to assign the dice into pairs is required. In Risk, the dice are sorted from largest value rolled to smallest, which is the approach taken here. We also consider games in which the dice are rolled one at a time (or one die is rolled several times) and left unsorted. We now show how these two methods significantly change the distribution of score differences.

3.1 Sorting Dice, with Ties

We first look at the case where each player rolls all $n$ of their $k$-sided dice and then sorts them in decreasing order. The two sets of dice are then paired and compared. If a player rolls more than one copy of the same number, the relative order of those two dice does not matter.

Figure 2 shows the distribution of score differences when each player rolls $n = 5$ dice and sorts them. We vary $k$, the number of sides. Ties are allowed, with neither player earning a point. We see the games all have a win bias of 0, as expected from the symmetry formed by players having the same rules. Tie percentage decreases as we increase the number of sides, as expected, since the more possible numbers to roll, the less likely the players will roll the same values. Increasing sides also decreases closeness, making higher score differences more likely to occur. For the case of 5d8 and 5d10, every possible score difference is approximately equally likely: large score differences are about as likely as close scores.
5d2 stands out as having a bell-shaped curve with significantly higher closeness: close games are more likely, but ties are also more likely as well. Nonetheless, 2-sided dice (e.g. coins) are not typically used in games, partly because standard coins are difficult to toss and keep from rolling off the table (Coin Age\footnote{https://boardgamegeek.com/boardgame/146130/coin-age} and Shift\footnote{https://laboratory.vg/shift/} are notable counter-examples, and some countries use square coins). However, stick dice – elongated dice that only land on the two long sides – do not roll away, and might be interesting items for game designers to investigate.

Figure 3 shows how changing the number of 6-sided dice affects the distribution of score differences. They remain symmetric with a win bias of 0, and after 2d6, adding more dice decreases the tie percentage. Closeness decreases as more dice are added, which makes sense as larger score differences are possible with more dice. 1d6 has a closeness greater than 1, because it allows 0-0 ties as well as games that end 1-0 or 0-1; without any ties, it would be exactly a closeness of 1.

3.2 Dice Unsorted, with Ties

We now examine the case where the dice are rolled and left unsorted. The dice could be rolled one at a time, possibly bringing out more drama as the battle is played out in individual dice pairings. Both players still roll \( n \) dice, but pair them in the order in which they were rolled rather than sorted by value, as shown in Figure 4b. The player with the higher value earns a point, and if tied then neither player earns a point.

Although we will think of the dice being rolled one at a time (and actually generate them in our simulations this way), it is also possible for players to roll a handful of dice to quickly create a sequence, as shown in Figure 4b. A player first rolls a handful of dice on the table. The dice are then put in order from left to right as they settled on the table. If two dice have the same horizontal position on the table (as the \( \) and \( \) do in the example), the die further away from the player will come before the die that is near.

3 One reviewer mentioned that their preferred method for rolling unsorted \( n \)d6 is to throw them against the inside of a sloped box lid: the dice line up in a random order as they slide to rest, and can be jiggled as needed.
In Figure 5, we examine how changing the number of sides of dice changes the distribution of ties and close games. We compare 2-sided dice (coins), 4-sided, 6-sided, 8-sided, and 10-sided dice. In all cases, the game is balanced, because the win bias is 0. With more sides, ties are less likely and closeness is generally lower. More sides therefore increase the odds of a lopsided victory with greater score differences between the players.

In Figure 6, we examine how changing the number of dice rolled affects the score difference. All of these games are balanced, since the win bias remains 0 in all cases.

Figure 5. Rolling 5dk unsorted, with ties.

Figure 6. Rolling n 6-sided unsorted, with ties.

Tied games are much more common when rolling an even number of dice. Comparing with Figure 3, we see that rolling unsorted increases the tie percentage. With more sides, closeness is lower, as we saw with rolling sorted.

### 3.3 Sorted versus Unsorted

In Figure 7, we review the effect of changing the way that dice are rolled, while keeping the same number of dice and number of sides (5d6 in this case). Rolling sorted has a flat distribution that leads to a higher likelihood of larger score differences, while rolling unsorted has a more normal-like distribution where closer games are more likely and closeness is higher. Higher closeness also increases tie percentage.

Game designers can choose the method appropriate for their games. In addition to choosing between rolling sorted or unsorted, the number of dice and number of dice sides can be changed. Using fewer sides on the dice increases closeness and tie percentage. Using fewer dice increases closeness and also generally increases the tie percentage. We address ties in the next sections.

### 4 Resolving Tied Battles

In the previous section, dice pairs tied in value did not give a point to either player. This leads to some situations in which the players get a 0 score difference and the overall game is tied (with as much as 24.6% for the 5d2 case). For games in which n is even, a score difference of 0 can occur (becoming less likely as k increases).

A game designer might want to make such tied games impossible. One simple way would be to have Player A automatically win whenever
the battle ends with a score difference of 0 – however this would have a massive bias in favour of Player A. In the above example, this would add an additional 24.6% bias which is likely unacceptable when trying to make the games close. To eliminate the bias over repeated battles, Player A and B could take turns receiving the win (perhaps by using a two-sided disk to indicate who will next receive the tiebreak).

Another simple way that would not have bias would be for the players to flip a coin (or some other random 50% chance event) to decide who is the winner of the battle. Using dice, the players could roll 1d \( k \) and let the player with the higher value win the battle. If they tie again, they repeat the 1d \( k \) roll until there is not a tie – we analyse this type of re-rolling in Section 8.

In the next few sections, we will examine some rule changes that make score differences of 0 impossible when \( n \) is odd. When \( n \) is even, score differences of 0 can still occur, and one of the above final tiebreaker methods can be used.

## 5 Favouring a Player

We now investigate breaking tied dice pairs by always having one player win a point when two dice are equal. We examine the case in which Player A will always win the point (as in Risk, in which defenders always win ties against attackers), but in general the same results apply if A and B are swapped. Favouring one player causes a bias, helping that player win more battles, so we examine ways to address this bias below.

### 5.1 Rolling Sorted, A Wins Ties

Figure 8 shows the score distributions when tied dice give a point to Player A. These distributions are clearly not symmetric, and are heavily skewed towards Player A, as reflected by the positive win bias. As one would expect, giving the ties to Player A causes that player to have an advantage over B. Increasing the number of sides on the die decreases the win bias; this is expected, as with more sides on the dice, the less likely it is for the players to both roll the same number. When \( n \) is odd, we also see that even score differences are no longer possible, and, most importantly, a tied score difference of 0 is no longer possible, so tie percentage is always 0. For the first time, we see an example of closeness increasing as the number of sides increases, because the distributions are less skewed towards large 5-0 lopsided wins.

### 5.2 Rolling Unsorted, A Wins Ties

By switching to rolling dice unsorted, the closeness is increased for all numbers of dice, and the distribution is more centred, but there is still a significant bias towards Player A, as we can see from Figure 9. This is an improvement, but one might desire another way to reduce the bias.

In conclusion, resolving tied pairs in favour of one player eliminates ties, but creates a large win bias. However, this can be reduced with more sides on the dice. This bias occurs for both rolling sorted and unsorted, although rolling unsorted
results in higher closeness and lower win bias. We now consider other ways to reduce this bias.

6 Reducing Bias with Fewer Dice

The bias introduced by having one player win ties can be undesirable for some designers and players, so we now look at a method of reducing this bias by having Player A roll fewer dice than Player B, to make up for the advantage they earn by winning ties. This method is used in Risk, in which the tie-winning bias towards Player A (defender) is reduced by allowing Player B (attacker) to roll an extra die when both sides are fighting with large armies. When rolling sorted, the dice are sorted in decreasing order, and the lowest-valued unmatched dice are ignored. When rolled unsorted, if one player rolls fewer dice, then there is no way to decide which should be ignored. We therefore only examine the case of rolling sorted.

Figure 10 shows the effect of requiring Player A to roll fewer dice. Rolling two or three fewer dice significantly favours Player B, and rolling the same number of dice favours Player A. However, Player A rolling 4d6 against Player B rolling 5d6 has a relatively balanced distribution, no longer significantly favouring one player over the other.

<table>
<thead>
<tr>
<th>Game</th>
<th>win bias</th>
<th>tie %</th>
<th>closeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d6 vs 5d6</td>
<td>-35.61</td>
<td>32.37</td>
<td>0.608</td>
</tr>
<tr>
<td>3d6 vs 5d6</td>
<td>-23.63</td>
<td>0.00</td>
<td>0.451</td>
</tr>
<tr>
<td>4d6 vs 5d6</td>
<td>-3.23</td>
<td>20.40</td>
<td>0.357</td>
</tr>
<tr>
<td>5d6 vs 5d6</td>
<td>38.21</td>
<td>0.00</td>
<td>0.282</td>
</tr>
</tbody>
</table>

Figure 10. A rolls fewer dice to control bias.

Having Player A roll fewer dice reduces the bias introduced by having them win all ties. Rolling one fewer dice is the best choice that leads to the smallest win bias, and having both players roll more dice also reduces the win bias, but decreases the closeness. Instead of having the players rolling different numbers of dice, we next examine having the players roll dice with different numbers of sides.

7 Reducing Bias with Mixed Dice

Another way to reduce Player A’s tie-winning bias is to give Player B some dice with more sides. For example, we could have Player A roll five 6-sided dice and have Player B roll three 6-sided dice and two 8-sided dice, to give them a small advantage to help eliminate the advantage A receives for winning ties. Because bias does not occur when we allow ties, we will only examine using mixed dice for games in which Player A wins ties.

7.1 Mixed Dice Sorted, A Wins Ties

Figure 12 shows the distribution of score differences for mixes of 6d6 and 8d8 for Player B, while Player A always rolls 5d6. Adding more 8d8 adjusts the bias in favour of Player B, but adding too many creates a strong bias for Player B.
A. Isaksen et al.  
Characterising Score Distributions in Dice Games  217

Figure 12. Mixing d6 and d8 to control bias.

The fairest of these is when Player B rolls 2d6 and 3d8 against Player A’s 5d6 (drawn as a solid line in the figure), with a win bias of -2.24%. We analysed all possible mixes of five dice made of 6-sided, 8-sided, and 10-sided dice, and found that only three cases have a win minus loss bias under 10%, as shown in Figure 13.

The bias is still most balanced when Player B rolls 2d6 and 3d8 against Player A’s 5d6. However, by rolling 3d6/1d8/1d10, we can get a slight bias towards Player A, if that is desired.

Figure 13. Least biased mixes of d6, d8, and d10.

7.2 Mixed Dice Unsorted, A Wins Ties

We can do the same type of experiment for all variations of Player B rolling unsorted a mix of five d6s and d8s against Player A’s 5d6, getting the results as shown in Figure 14. By using 2d6 and 3d8, we can reduce the bias down to a small 1.61% in favour of Player B.

Mixed Dice Rolled Sorted, A Wins Ties

<table>
<thead>
<tr>
<th>Game</th>
<th>win bias</th>
<th>tie %</th>
<th>closeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d6/0d8</td>
<td>38.21</td>
<td>0.00</td>
<td>0.282</td>
</tr>
<tr>
<td>4d6/1d8</td>
<td>24.36</td>
<td>0.00</td>
<td>0.294</td>
</tr>
<tr>
<td>3d6/2d8</td>
<td>10.80</td>
<td>0.00</td>
<td>0.302</td>
</tr>
<tr>
<td>2d6/3d8</td>
<td>-2.24</td>
<td>0.00</td>
<td>0.305</td>
</tr>
<tr>
<td>1d6/4d8</td>
<td>-14.56</td>
<td>0.00</td>
<td>0.305</td>
</tr>
<tr>
<td>0d6/5d8</td>
<td>-25.98</td>
<td>0.00</td>
<td>0.301</td>
</tr>
</tbody>
</table>

Figure 12. Mixing d6 and d8 to control bias.

Mixed Dice Rolled Unsorted, A Wins Ties

<table>
<thead>
<tr>
<th>Game</th>
<th>win bias</th>
<th>tie %</th>
<th>closeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d6/0d8</td>
<td>30.68</td>
<td>0.00</td>
<td>0.424</td>
</tr>
<tr>
<td>4d6/1d8</td>
<td>20.34</td>
<td>0.00</td>
<td>0.440</td>
</tr>
<tr>
<td>3d6/2d8</td>
<td>9.48</td>
<td>0.00</td>
<td>0.450</td>
</tr>
<tr>
<td>2d6/3d8</td>
<td>-2.24</td>
<td>0.00</td>
<td>0.305</td>
</tr>
<tr>
<td>1d6/4d8</td>
<td>-14.56</td>
<td>0.00</td>
<td>0.305</td>
</tr>
<tr>
<td>0d6/5d8</td>
<td>-25.98</td>
<td>0.00</td>
<td>0.301</td>
</tr>
</tbody>
</table>

Figure 14. Mixing d6 and d8 to control bias.

Mixed Dice Rolled Sorted, A Wins Ties

<table>
<thead>
<tr>
<th>Game</th>
<th>win bias</th>
<th>tie %</th>
<th>closeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>3d6/1d8/1d10</td>
<td>2.67</td>
<td>0.00</td>
<td>0.307</td>
</tr>
<tr>
<td>2d6/3d8</td>
<td>-2.24</td>
<td>0.00</td>
<td>0.305</td>
</tr>
<tr>
<td>3d6/2d10</td>
<td>-5.36</td>
<td>0.00</td>
<td>0.310</td>
</tr>
</tbody>
</table>

After trying all variations of five d6s, d8s, and d10s, we found only five cases in which the magnitude of the win bias is less than 10, which are shown in Figure 15. Player B rolling 2d6/3d8 gives the lowest absolute win bias (-1.61), while Player B rolling 3d6/1d8/1d10 gives the lowest bias that favours Player A (2.9).

In conclusion, when Player A wins tied pairs, we can reduce their win bias by Player B roll different-sided dice. Looking at all mixes of five dice composed of d6, d8, and d10, we found that rolling 5d6 against 2d6/3d8 produces the smallest win bias, when rolling both sorted and unsorted. In fact, no mix of dice completely eliminated the win bias. Next, we examine a final way to break ties that does eliminate win bias.
We now examine re-rolling tied dice as a final way to deal with tied pairs of dice. For example, it is common for players to roll 1d6 at the start of a game to decide who moves first, and to re-roll the die in case of a tie. While this can be generalised to ndk, it is cumbersome; this section explains why we believe that this is inadvisable in practice. Re-rolling can go on for many iterations, so we use Monte Carlo simulation to evaluate the probabilities empirically instead of exactly, since these games can theoretically continue indefinitely, with increasingly unlikely probability. We used \( N = 6^{10} = 60,466,176 \) simulations per game, as this is the same number used in our previous cases (see Appendix for this calculation). We use the \( \approx \) symbol to denote empirical rather than exact values in the figures.

### 8.1 Rolling Sorted, Re-Rolling Tied Dice

We first examine the case in which we roll a handful of dice and then sort them from highest to lowest. Untied dice pairs are scored, then tied dice pairs are re-rolled in sub-games until there are no more ties. Each player’s score from their initial roll, and all subsequent re-rolls, are summed together to give their overall score. The resulting score difference distributions for 5dk are shown in Figure 16. The battles are all unbiased, symmetrical, and without ties.

For 5d4, 5d6, 5d8 and 5d10, the distributions are effectively flat with low closeness and have approximately the same shape as when rolling sorted with ties (as in Figure 2) but now do not permit tie games. Compared to 5d4 and higher, 5d2 has a higher closeness. However, this closeness comes at the significant cost of requiring many re-rolls, as demonstrated in Figure 17. This shows that having more sides decreases the probability of a re-roll, and with 5d2 or 5d4 it is likely that players will have to make two or more re-rolls in practice, which is undesirable. Higher-sided dice are less likely to tie, so the probability of re-rolling decreases quickly when using six or more sides.

---

6That is, dice with a greater number of sides.
We also examine the effect of changing the number of dice while holding the number of sides fixed at 6 in Figure 18. The distributions are all flat, but closeness can be increased by using fewer dice, as we have seen in previous sections. The probability of re-rolls is also affected by the number of dice, as shown in Figure 19. For 1d6 and 2d6, the most common outcome is no re-rolls. Increasing the number of d6 makes re-rolling more likely, but in any case the probability of having to make many iterative re-rolls decreases rapidly (in contrast with d2).

Because neither player is favoured, the metrics can be analytically calculated from the binomial distribution \( \binom{n}{w} p^w (1 - p)^{n-w} \), with \( w \) being the number of wins for Player A in the battle, \( n \) dice rolled, and probability \( p = .5 \) (no matter the value of \( k \)) of Player A winning each point. Given a score difference \( d \), we can calculate \( w = (n + d) / 2 \).

Figure 19 shows the identical score difference distribution for all dice, no matter how many sides they have. The battle is unbiased, with no ties, and has a closeness of .447. However, they do not cause the same number of re-rolls, as shown in Figure 21, generated with Monte Carlo simulation. Dice with more sides reduce the required number of re-rolls.

Rolling fewer dice increases the closeness (Figure 22). Rolling fewer dice also reduces the probability of re-rolls (Figure 23).
8.3 Sorted, A Wins Ties, Re-Rolls Highest

We now consider a hybrid case, in which Player A wins all ties but must re-roll when they the die’s highest value (e.g. a 6 on a 6-sided die). This effectively means that Player A rolls \((k - 1)\)-sided dice while Player B rolls \(k\)-sided dice. This gives an advantage to Player B to compensate for Player A’s tie-winning advantage.

Interestingly, this has the same effect as in the previous re-roll sections, for both rolling sorted or unsorted. Therefore, the plots are the same as in Figures 16, 18, 20, and 22. The benefit is that there is no re-rolling of ties. Only Player A has to re-roll when they roll \(k\) (unless they have access to \((k - 1)\)-sided dice).

We can show analytically why this is unbiased for the simple case of one \(k\)-sided die. Player A re-rolls a \(k\) result, which is the same as rolling a \((k - 1)\)-sided die. If Player A rolls a value of \(i\) with probability \(1/(k - 1)\), then they win when Player B rolls a value \(\leq i\) with probability \(i/k\), since A wins ties. Calculating the expected number of wins for Player A, over all values of \(i\) from 1 to \(k - 1\) we have:

\[
\sum_{i=1}^{k-1} \frac{1}{k-1} \frac{i}{k(k-1)} = \frac{1}{2}
\]

which is independent of \(k\) and is always 1/2.

We have shown that re-rolling tied pairs gives unbiased results, but at the cost of requiring the players to re-roll, which can take longer. The expected number of re-rolls decreases with higher-sided dice. But we have shown other effective methods of eliminating ties which do not require re-rolls, so we suggest using these alternatives.

9 Risk & Risk 2210AD

We can use these results to examine how the original Risk compares with the popular variant Risk 2210AD [18]. In both games, the players roll sorted dice and the defender wins tied dice. As shown in Section 5.1, this gives a strong advantage to the defender when rolling the same number of dice, which can lead to a static game in which neither player wants to attack. To counteract this, both games allow the attacker to roll an extra die (3d6 versus 2d6). As shown in Section 6, this flips the advantage towards the attacker. An attacking advantage encourages players to play more aggressively.

In Risk 2210AD, special units called commanders and space stations replace one or more \(d6\) with \(d8\) in battles. As shown in Section 7, mixed dice bias the win rate towards the player rolling higher-valued dice. An attacker can use this to decrease the closeness and increase the predictability of battles, and a defender can use this to compensate for rolling fewer dice.

The rules in dice games require careful balancing, as the exact number of dice and number of sides can often have a large impact on the statistical outcome of the battles, as we have shown. Risk and Risk 2210AD are no exception, and their dice mechanisms appear to have been carefully tuned to provide reasonable win bias and closeness values.

10 Conclusion

We have demonstrated the use of \(win\) \(bias\), \(tie\) \(percentage\), and \(closeness\) to analyse a collection of dice battle variants for use as a component in a larger game. We introduce closeness, which is related to the precision statistic about 0, and matches the intuitive concept of a game being close. We have not seen this statistic used before...
to analyse games. The results of the previous sections let us make some general statements about this category of dice battles in which the number values are compared.

In Section 3, we showed that when tied pairs are allowed, rolling dice unsorted will result in higher closeness, i.e. a lower chance of games with large point differences; however, this comes at the cost of increasing the tie percentage. Using dice with fewer sides increases closeness, but also increases the tie percentage. Using fewer dice also increases closeness and generally increases the tie percentage.

Battles that end tied with a score difference of 0 can be broken with a coin flip or other 50/50 random event, as discussed in Section 4. However, we also explored rule changes that would cause odd-numbers of dice to never end in a tie. Breaking tied dice pairs in favour of one player, as shown in Section 5, eliminates ties but creates a large win bias, although this bias can be reduced by using higher-sided dice. This bias occurs for both rolling sorted and unsorted, although rolling unsorted gives higher closeness and slightly lower win bias.

In Section 6, we further reduced the win bias by having the favoured player roll fewer dice. Rolling one fewer die is the best choice that leads to the win bias closest to 0, and having both players roll more dice also makes the win bias closer to 0 but also decreases the closeness.

In Section 7, we reduced the win bias by having the unfavoured player roll different sided dice. Looking at all mixes of five dice composed of d6, d8 and d10, rolling 5d6 against 2d6/3d8 produced the smallest win bias, for both rolling sorted and unsorted. However, there was no way to completely eliminate the win bias.

In Section 8, we examined breaking tied pairs by re-rolling them. This gives unbiased results, but at the cost of a potentially lengthy re-rolling process. Using higher-sided dice or fewer dice reduces the expected number of re-rolls that will occur, but we recommend other tie-breaking methods that are less cumbersome for the players.

One surprising outcome of this study is that nd2 sorted with ties may be an under-used dice mechanic for games. This has high closeness and can be implemented by coins or stick dice, whose flat sides obviate some practical problems of round coins such as rolling off the table.

This article focused on comparing dice values, but we are also doing a similar study of hit-based dice games, which includes analysing the effect of critical hits, following the same framework presented here.

For finer-grained control over the game experience, the designer can instead use a bag of dice tokens (e.g. small cardboard chits with dice faces printed on them) or a deck of dice cards to enforce that certain distributions are obeyed with local representation. This is choosing without replacement instead of choosing with replacement, which typically occurs in dice games. We are currently experimenting with examining similar games that use bags of dice tokens, using an exhaustive analysis similar to that done here.

In summary, there is no single best dice battle mechanism, and the designer must make a series of tradeoffs. We hope that this paper can provide some quantitative guidance to designers in search of dice games that exhibit particular properties or present a certain feel to the players. For designers wishing to use rules that we did not discuss in this paper, we hope it would not be difficult to use the techniques describe above to evaluate how the players might experience the distribution of score differences by measuring win biases, ties and closeness.

Acknowledgements

We wish to thank Steven J. Brams and Mehmet Ismail for comments on an early draft of the paper, and to the Editor-in-Chief Cameron Browne and anonymous reviewers for their insightful reviews and constructive suggestions.

References

Appendix

In this appendix, we give analytical results for the probabilities and number of possible outcomes for many of the games studied in this paper. A more complete coverage of these probabilities and combinatorics can be found in [17].

A fair k-sided die has equal probability of rolling each of its k sides, so the probability of rolling any particular number is 1/k. Therefore, the total probability of rolling a value v or less is \( \sum_{i=1}^{v} 1/k = v/k \).

If we roll n dice unsorted, there are \( k^n \) different ways to roll the dice. Each way of rolling the dice, since the order matters, has an equal \( 1/k \) chance. For example, if we roll five 6-sided dice unsorted, there are \( 6^5 = 7,776 \) possible outcomes each with 1/7,776 probability. If Player A is rolling a dice, and player B is rolling b dice, then there are \( k^{a+b} \) possible outcomes. So, if each side rolls five 6-sided dice unsorted, there are \( 6^{10} = 60,466,176 \) possible games that can occur, each equally likely. Rolling 5d10 against 5d10 has 10,000,000,000 different possible outcomes.

If we roll the n dice sorted, then we can describe the probabilities using the multinomial distribution, a generalisation of the binomial distribution when there are k possible outcomes for each trial. If one knows the outcome of a sorted roll had \( x_i \) copies of \( i \) (i.e. \( x_1 \) 1s, \( x_2 \) 2s, etc.), such that \( x_1 + x_2 + \ldots + x_k = n \), the number of ways that particular outcome could have been rolled is:

\[
\frac{n!}{x_1!x_2!\ldots x_k!}
\]

(5)

The probability of rolling that outcome is:

\[
\frac{n!}{x_1!x_2!\ldots x_k!} k^{-x_1}x_2\ldots x_k
\]

(6)

For rolling n k-sided dice sorted, the number of different possible results a player can roll is:

\[
\binom{n+k-1}{k-1}
\]

(7)

For example, for 5d6, there are \( \binom{5+6-1}{6-1} = \binom{10}{5} = 252 \) unique ways to roll the dice, although...
these are not of equal probability. For two players, there are \(252^2 = 63,504\) ways to evaluate the game. This means that the rolling sorted calculations can be made much faster by only calculating each unique outcome once, but then multiplying the results by Equation 5, the number of ways each result can occur.

---

**Shakashaka Challenges #1, #2 and #3**

Half-colour empty cells with triangles to create white squares and rectangles, as per the rules on p. 13.

**Challenge #1**
Challenge by ‘the axe and sword’ © Nikoli.

![Challenge #1](image1)

**Challenge #2**
Challenge by ‘cubic function’ © Nikoli.

![Challenge #2](image2)

**Challenge #3**
Challenge by Komeida © Nikoli.

![Challenge #3](image3)
Narrative Progression Traits for Role-Playing Games

Spyridon Samothrakis, University of Essex

We categorise popular (tabletop) role-playing game (RPG) settings according to the content of their game-theoretic outcome (positive-sum, negative-sum, constant-sum) and the arbitrariness/causality of that outcome (stochastic vs deterministic). Six categories are captured by assuming that all players collectively represent one agent, the prescribed game universe defines the game, and some of the other world entities represent other players. We show how this categorisation of games can inform the design of rules for new games based on the game setting by introducing ‘narrative progression traits’, which offer a method of tracking the progress of player characters.

1 Introduction

Role-playing games (RPGs) are an eclectic mix of narrative and elements from board games and miniature war games [1]. An imaginary universe is fleshed out in a series of source books that are coupled with rules that allow players to create stories within that universe. One of the players receives a special role, usually called the game master (GM), whose purpose is to oversee the correct execution of the mechanics of the game, bring the setting to life and push the story forward. The rest of the players act as player characters (PCs), who are individual agents with their own goals within the game universe. The archetypal game of this type is Dungeons & Dragons [2].

The GM is responsible for: 1) handling the mechanics of the universe, and 2) acting on behalf of non-player characters (NPCs). NPCs are agents who populate the universe and act in conjunction with other characters, and whose roles can be as limited or as extravagant as the game dictates. The setting of the game is defined by its narrative combined with its mechanics.

RPGs are usually played on a table but without using a board as such. The typical equipment for each PC includes a character sheet, pencil and set of dice, as shown in Figure 1. The GM equipment typically includes: 1) a rulebook, which is often accessible to the players; 2) a GM screen, with concise reference information for the GM’s eyes only; and 3) some dice, as shown in Figure 2.

Most studies in role playing games have focussed on the nature of PCs, possibly due to the strong influence of the online games community [3]. Other studies exist which look more broadly at the entire RPG experience [4]. In this article, we will focus on the relationships between the game universe, PCs and NPCs.

The rest of the paper is organised as follows: Section 2 describes the relationship between RPG games and multi-agent environments; Section 3 proposes six categories for classifying RPGs with examples; Section 4 proposes a mechanism for including narrative progression traits in games based on the categories they belong to and explains why such mechanisms are important; and Section 5 concludes the article with a short discussion.
2 RPGs as Multi-Agent Environments

The link between RPGs and multi-agent environments has been long established \[5, 6\]; one can see these games as massive asymmetric extensive-form games, where players (and, consequently, player characters) are expected to maximise some notion of (long term) reward (e.g. maximise treasure, gain experience points, avenge a death etc.). The rewards provided by a game as characters progress are not necessarily of the same type; PCs might have to maximise multiple orthogonal types of rewards.

For the purpose of this article, however, we consider all rewards as existing along a single dimension, although this simplification may not be strictly true. A common abstraction in real multi-objective problems is to create linear combinations of rewards, in a process known as scalarisation \[7\]. For example, in settings where the rewards are experience points and wealth, a linear combination of the two can reasonably define a single-dimensional progress metric. With qualitative rewards, such as revenge and treasure, a non-linear mapping might be more suitable.

Each game can be broken down into two sets of agents: the PCs and the NPCs. If, at the end of a series of adventures or a long session, the PCs and the NPCs collectively have more total reward, then we label the game positive-sum. If the total amount of reward stays the same, we are going to label the game constant-sum. If the total amount of reward is collectively less for everyone involved, then the game is labelled negative-sum.

Note that this is different to the element of progression \[8\], which does not necessarily coincide with rewards. An example is the progression of someone captured as a slave; their strength might improve from working in the mines, but the possibility of permanent servitude would be mind crushing. Even if the game does not have mechanics to address such a concept, the narrative created in the game may push toward an outcome that is worse for the PCs overall.

What constitutes reward depends on the narrative of the game setting, and might not be explicitly tied to the game mechanics/system. Notice that negative-sum vs positive-sum concerns the overall trend and not just the present situation; a bad situation that improves positive-sum, while a good situation that gets worse is negative-sum. A key point is that the characterisation involves all players, PCs and NPCs, collectively.

We distinguish monsters from NPCs. That is, not all entities in a game are viewed as players with agency; monsters can be seen as forces of nature. Thus, exterminating a tribe of kobolds is not to be considered genocide, but rather reshaping the environment, like digging a dam or building a house. Games in which PCs are monsters, such as most World of Darkness (WoD) games, might make normal humans seem lacking in agency. If a game does not make a clear distinction between monsters and NPCs, one will be made in this text. This distinction between automatons and conscious agents is often found in stories.
Another element of extensive-form games is stochasticity, and most RPGs involve an element of luck. Here we are going to be concerned with the way the reward structure of the game treats those. For example, in most RPGs, battles have an element of luck which can be offset by skill. On the other hand, a game that involves random deaths or random encounters with extremely powerful monsters has a high element of stochasticity in the way it deals rewards.

It should also be noted that most role playing games can be tailored to suit the style and preferences of its players; even the most grim setting can be turned into a cheerful, comedic experience. This is a perfectly valid use of the material, but playing against the intended spirit of a game makes it harder to formally categorise the type of playing experience it provides.

Finally, although a game setting might not be positive-sum for the characters, RPG sessions should of course be positive-sum for the players. That is, the players should enjoy the experience if the game is to continue, especially if they are participating in a campaign that will continue through many sessions. This is analogous to players enjoying a social game of Poker game or Chess, even though the game itself is strictly competitive and some players will lose.

3 Six Categories

Given the short analysis above, we identify six categories, formed by three types of reward outcomes in stochastic and deterministic settings. Representative games from each category are summarised in Table 1, and their relationships shown graphically shown in Figure 3. The breakdown of NPCs, PCs and universe can be seen in Table 2. It is worth noting that our analysis is not definitive. Different categories can arise depending on campaign focus and interpretations.

3.1 Deterministic Constant-Sum

This category involves games with certain Chess-like zero-sum qualities; someone’s loss is necessarily someone else’s gain. There are rules in the universe which, if followed to their logical conclusion, can lead to a rewarding life. Study hard and you will succeed is an optimistic view of this world; study hard and there is a small chance that you will succeed is a more cynical but accurate recognition of the forces outside one’s control.

The rules are such that one has to compete with NPCs and outperform them, and improvements come at the direct expense of others. A game in this category is King Arthur Pendragon [9], which leaves little to chance, and is strongly antagonistic towards NPCs given its inherent element of power politics.

3.2 Deterministic Positive-Sum

This category involves games that do not have an inherent element of conflict; if conflict arises, it is the result of lack of knowledge on the parts of the PCs or NPCs. The universe remains judgemental, but strong competition with other NPCs is not necessary to succeed. Games in this category are the most cheerful of all, and the inhabitants of such universes – at least those to which the game implicitly attributes agency – could end up living well for eternity. Adventures are often based on some invasive evil motivated by a sort of ignorant malice, such as Sauron from The Lord of the Rings. Everyone prospers until some evil entity ruins everything out of spite, purely to dominate for the sake of domination rather than necessity.

Games in this category include the classic Tolkien-esque worlds of Forgotten Realms [10], games such as Ars Magica [11] that bring a strong element of hope in the discovery of knowledge, lighthearted games such as Ghostbusters [12] and superhero games such as Mutants & Masterminds [13]. NPCs in these games are mostly misguided caricatures, whose evil is a conscious choice rather than a vehicle for material gain or advancement.

3.3 Deterministic Negative-Sum

This game type pits the PCs and NPCs in competition for a share of ever-decreasing rewards. Punishments are being delivered, but they are not arbitrary. The degree of one’s suffering typically depends on how well one manages to outcompete other NPCs and other PCs. Games of this type have rules allowing players to control more than one PC due to high mortality rates (e.g. the Advanced Dungeons &Dragons game world Dark Sun [14] has character trees).

Usually, some kind of ecological or other disaster has befallen the game setting, which keeps on dying as the game unfolds. In other settings, a malevolent presence governs the game world. An example of this is the computer in Paranoia [15], which is an insane AI imposing nonsensical rules, which characters violate at their peril.

3.4 Stochastic Constant-Sum

Universes of this type are like a Poker game; competition exists, but rewards are dealt in a quasi-random fashion. PCs and NPCs are in competition with each other, but the rewards are not delivered predictably. One can do one’s best and still fail to gain any meaningful reward.
**Table 1.** Categorisations of some popular RPGs. Parentheses denote shared game systems.

<table>
<thead>
<tr>
<th>Game</th>
<th>Randomness</th>
<th>Reward Regime</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shadowrun</td>
<td>Stochastic</td>
<td>Constant Sum</td>
<td>Cyberpunk with a magic element.</td>
</tr>
<tr>
<td>Dark Sun (D&amp;D)</td>
<td>Deterministic</td>
<td>Negative Sum</td>
<td>Post-apocalyptic fantasy universe.</td>
</tr>
<tr>
<td>Forgotten Realms (D&amp;D)</td>
<td>Deterministic</td>
<td>Positive Sum</td>
<td>Set in a Tolkien-esque fantasy setting.</td>
</tr>
<tr>
<td>Eclipse Phase</td>
<td>Stochastic</td>
<td>Positive Sum</td>
<td>Positive trans-human with horror.</td>
</tr>
<tr>
<td>Call of Cthulhu</td>
<td>Stochastic</td>
<td>Negative Sum</td>
<td>The universe is cruel and indifferent.</td>
</tr>
<tr>
<td>Mage: The Ascension (WoD)</td>
<td>Stochastic</td>
<td>Constant Sum</td>
<td>Game of existential discovery.</td>
</tr>
<tr>
<td>Vampire: The Masquerade (WoD)</td>
<td>Stochastic</td>
<td>Constant Sum</td>
<td>Strict competition among the undead.</td>
</tr>
<tr>
<td>Pendragon</td>
<td>Deterministic</td>
<td>Positive Sum</td>
<td>Based on the Arthurian legend.</td>
</tr>
<tr>
<td>Dark Heresy</td>
<td>Stochastic</td>
<td>Negative Sum</td>
<td>Science-fantasy in a grim universe.</td>
</tr>
<tr>
<td>Warhammer Fantasy</td>
<td>Stochastic</td>
<td>Constant/Negative</td>
<td>Fantasy set in medieval Germany.</td>
</tr>
<tr>
<td>Mutants &amp; Masterminds</td>
<td>Deterministic</td>
<td>Positive Sum</td>
<td>The classic superhero game.</td>
</tr>
<tr>
<td>Traveller</td>
<td>Stochastic</td>
<td>Constant Sum</td>
<td>Space opera/galactic exploration.</td>
</tr>
<tr>
<td>All flesh must be eaten</td>
<td>Stochastic</td>
<td>Negative Sum</td>
<td>Unexpected zombie apocalypse.</td>
</tr>
<tr>
<td>Mindjammer</td>
<td>Stochastic</td>
<td>Positive Sum</td>
<td>Trans-human adventures in space.</td>
</tr>
<tr>
<td>Paranoia</td>
<td>Deterministic</td>
<td>Negative Sum</td>
<td>Humorous futuristic game.</td>
</tr>
<tr>
<td>Ghostbusters</td>
<td>Deterministic</td>
<td>Positive Sum</td>
<td>Light-hearted game ghost catching.</td>
</tr>
<tr>
<td>Ars Magica</td>
<td>Deterministic</td>
<td>Positive Sum</td>
<td>The renaissance of magic in Europe.</td>
</tr>
<tr>
<td>Fiasco</td>
<td>Stochastic</td>
<td>Negative Sum</td>
<td>Game in which failure is expected.</td>
</tr>
</tbody>
</table>

3.5 **Stochastic Positive-Sum**

Universes of this type will invariably lead to PCs and NPCs gaining rewards without having to strictly compete, although those rewards will occur randomly. Cooperating to find a treasure, for example, might lead to widely different rewards for different teams. Drinking from the fountain of life might extend one’s life by a day or a century. This category includes trans-human games such as Mindjammer [20] and Eclipse Phase [21].

Eclipse Phase is a controversial inclusion in this category. In a setting of existential horror, characters advance almost infinitely, unhindered by mechanics or back-story, as negative events are things of the past. Adversaries (AIs named TITANS) are currently dormant and can rarely impact the game.

3.6 **Stochastic Negative-Sum**

This category involves games and worlds that are deeply negative. The universe will punish everyone just for existing, and will do so capriciously. The most prominent game in this category is Call of Cthulhu [22], in which a malevolent universe delivers arbitrary punishments. All Flesh Must Be Eaten also fits here, as the characters are faced with a zombie apocalypse.

Warhammer-themed games, such as Dark Heresy [23] and Warhammer Fantasy Roleplay [24], arguably fall in this category, due to the randomly corrupting influence of evil gods. Fantasy games can, however, also be considered to be constant-sum mode, as their worlds often seem more balanced than futuristic worlds. Fiasco, a semi-comical game of active failure [25], is another example.

4 **Narrative Progression Traits**

Figure 3 plots the games mentioned above based on their randomness (y-axis) versus reward outcome (x-axis). In most such games, however, the overall feeling of well-being or reward of the PCs and NPCs, as portrayed in the setting, is not actually represented in the game system. For example, as far as the game system is concerned, a gladiator in the arenas of Dark Sun is considered ‘successful’ merely due to skill bonuses from constant sport-fighting between battles. We can probably agree, however, that such a life would not be a good life by any reasonable standard. The game system does not have rules reflecting such life quality aspects of the narrative, and a character’s life is reduced to mere level advancement.

In addition, video games and massively multiplayer online RPGs (MMORPGs) promote a cer-

---

1 The reference to ‘tanks’ in the 5th edition of Shadowrun means ‘PCs that can take damage’. 
tain world view that makes the narrative less relevant than combat. Stories are seen merely as a way to increase character attributes, gain treasure, and so on; the characters’ impact on the universe is disregarded, or left completely up to the GM.

Rules that directly deal coherently with the narrative could add tremendously to the game experience, although I understand if some readers disagree. One counter-argument is that additional rules to cover such narrative aspects would significantly increase the length of current rule books. Core books are getting larger all the time (see Figure 4), and not all players will enjoy having even more material to study.

There is no fixed school of thought regarding such consistency, although Tolkien himself had the following to say:

_Fantasy is a natural human activity. It certainly does not destroy or even insult Reason; and it does not either blunt the appetite for, nor obscure the perception of, scientific verity. On the contrary. The keener and the clearer is the reason, the better fantasy will it make._ [26]

A second counter-argument is that making systems more rule-heavy is detrimental to the games. Indeed, there do seem to be some very successful RPGs (e.g. Fiasco [25], Fate [27] and Apocalypse World [28]), that aim to minimise the rules in favour of a strong narrative.

These games’ rules achieve their aim efficiently, by sacrificing the simulationist aspect of RPGs, and foregoing the war gaming legacy of RPGs almost completely; for example, there is no initiative phase in Apocalypse World. Whether such rules are preferable to more classic systems (e.g. Figure 5) is a matter of personal taste. Similarly, our attempt here to infuse more traditional games with explicit narrative rules may suit some players more than others.

### Table 2. Agents in some popular games: PCs, NPCs and Monsters/Universe.

<table>
<thead>
<tr>
<th>Game</th>
<th>PCs</th>
<th>NPCs</th>
<th>Monsters/Universe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shadowrun</td>
<td>Shadowrunners</td>
<td>Other Shadowrunners</td>
<td>Megacorps</td>
</tr>
<tr>
<td>Dark Sun (D&amp;D)</td>
<td>Adventurers</td>
<td>Dragon King minions</td>
<td>Dragon Kings, Dragon</td>
</tr>
<tr>
<td>Forgotten Realms (D&amp;D)</td>
<td>Adventurers</td>
<td>Antagonists</td>
<td>Classic RPG monsters</td>
</tr>
<tr>
<td>Eclipse Phase</td>
<td>Firewall Agents</td>
<td>Other trans-humans</td>
<td>Titans, existential threats</td>
</tr>
<tr>
<td>Call of Cthulhu</td>
<td>Investigators</td>
<td>Cultists</td>
<td>Great Old Ones</td>
</tr>
<tr>
<td>Mage: The Ascension (WoD)</td>
<td>Traditions</td>
<td>Technocracy</td>
<td>Reality</td>
</tr>
<tr>
<td>Vampire: The Masquerade (WoD)</td>
<td>Vampires</td>
<td>Other Vampires</td>
<td>Ancient Vampires</td>
</tr>
<tr>
<td>Pendragon</td>
<td>Lords</td>
<td>Other evil Lords</td>
<td>Monsters</td>
</tr>
<tr>
<td>Dark Heresy</td>
<td>Inquisitorial Retinue</td>
<td>Heretics</td>
<td>Chaos/Xenos/Empire</td>
</tr>
<tr>
<td>Warhammer Fantasy</td>
<td>Adventurers</td>
<td>Chaos</td>
<td>Chaos vs Order</td>
</tr>
<tr>
<td>Mutants &amp; Masterminds</td>
<td>Good superheros</td>
<td>Bad villains</td>
<td>The laws of physics</td>
</tr>
<tr>
<td>Traveller</td>
<td>Space-faring adventurers</td>
<td>Other denizens</td>
<td>The universe</td>
</tr>
<tr>
<td>All Flesh Must Be Eaten</td>
<td>Common folk</td>
<td>Other common folk</td>
<td>Zombies</td>
</tr>
<tr>
<td>Mindjammer</td>
<td>Trans-humans</td>
<td>Trans-humans</td>
<td>The galaxy</td>
</tr>
<tr>
<td>Paranoia</td>
<td>Agents</td>
<td>Other Agents</td>
<td>The Computer</td>
</tr>
<tr>
<td>Ghostbusters</td>
<td>Ghostbusters</td>
<td>Everyday folk</td>
<td>Ghosts</td>
</tr>
<tr>
<td>Ars Magica</td>
<td>Mages</td>
<td>Other Mages</td>
<td>The church, monsters</td>
</tr>
<tr>
<td>Fiasco</td>
<td>Ambitious everyday folk</td>
<td>Relatives</td>
<td>An unforgiving world</td>
</tr>
</tbody>
</table>

---

_Figure 3. Example games plotted according to randomness and reward outcome (subjectively)._
Overall, we hope that the better alignment between the narrative, the total rewards, and the game, can allow access to a new part of the design space [7], in which one can maintain the war game legacy while augmenting the narrative. Such alignment partially exists in games such as Call of Cthulhu with its sanity mechanism [22], or Vampire: The Masquerade with its humanity mechanism [17].

I propose a complete system of advancement based on the categories proposed above, alongside the development of character traits that fit the game setting, coupled with the effects that such advancement has on the rest of the game setting. Taking into account the effects of one’s actions in the whole setting should allow for new ways of portraying a character’s personality. We now explore three types of such narrative traits: constant-sum, positive-sum and negative-sum.

### 4.1 Constant-Sum Traits

The setting explores competition, and this should be emphasised in both the development of PCs and the characteristics of NPCs. Narrative progression traits include aspects such as fairness and cruelty. Loss and gain are complementary, so experience points can be won only at the expense of others, e.g. treasure is hoarded and can be stolen.

Game mechanics can also cover social interactions, e.g. when a player attempts to persuade an NPC to behave a certain way or perform a certain action. In a stochastic world, the rolls can form part of a strictly competitive game (in the game-theoretic sense). In a deterministic world, the character with the higher skill level could simply win the social scenario.

Note that it is also harder to assign moral qualities in such a universe; cooperation is practically meaningless, apart from short-term coalitions of looters, as is altruism. It is interesting to note that classic RPG settings emphasise competition, with treasure hoarding and experience point acquisition, yet characters often behave – or are expected to behave – rather paradoxically in ways we would consider morally good. This good behaviour is typically explained away by comparison to their sadistically evil adversaries, but the setting tells us nothing as to why these adversaries are so evil.

### 4.2 Positive-Sum Traits

The game can emphasise elements of narrative progression traits independent of other agents. Personal knowledge within the setting might include: characters getting better at some ad-hoc in-game play; exploration of remote locations; conflict against a corrupt few – or their conversion; and so on. Most game settings seem to assume that everyone can progress, while existing in a world in turmoil, with little examination of why these adversarial forces exist.

Special in-game attributes of wickedness are extremely meaningful here; harming others for personal gain or amusement creates a strikingly evil character in an otherwise nice universe. In the stochastic version, one might create attributes such as stoicly, i.e. the ability to withstand random, short periods of bad luck. This might indeed apply to all stochastic categories, but stoicism could be a more relevant trait in a positive-sum world in which a PC might think: Everyone else is doing fine, but I am suffering for no reason.

The system could encourage social interaction by ensuring some kind of gain for all players form the interaction, regardless of its outcome; even a failed social interaction could still yield some benefit for the character.
4.3 Negative-Sum Traits

The inevitability of further loss can be captured here with in-game attributes such as alienation, disconnection, or even possibly enlightenment, which can be linked to both NPCs and PCs to guide their behaviour. One can also add elements such as sanity and paranoia. These should be more pronounced in the stochastic version, as the world’s lack of certainty should have more impact on each character’s personality. Progression may occur at either end of the scale of these attributes.

Levels of exhaustion can be used to describe the constant struggle to exist in a world with constantly diminishing resources. Social interactions – and possibly battles – can be represented in a negative light. The grinding nature of everything leaves everyone worse off; interactions with others are to be avoided, as they will be a source of pain for everyone.

The specific character traits would create a stronger link between the game setting and the game system, creating different types of character progression among different types of games. Ideally, these traits would also influence the narrative, in return.

5 Conclusion

We have categorised role playing games according to some of their game-theoretic and causal properties. There is understandably a sense of ambiguity to the categories, which is inevitable as the games are too complicated to be broken down into simple mechanics. A more objective study of games and narrative might involve the word vectorisation [30] of associated keywords, to perform an optimal classification.

We have introduced elements of progression that tie game narratives closer to the setting, as it is all too easy for a game group to ignore the setting and play even the darkest of games as a ‘happy violence’ trope. By thinking in terms of future rewards, designers can help limit this, and encourage games to be played in the intended spirit.

But is this coupling between narrative and setting worth pursuing? I would argue that it is. Obviously, players can ignore mechanisms that link the setting to the system, but their existence provides a strong sense of theme and guidance, allowing everyone to be involved in a more satisfying and narratively coherent game.

References

Spyridon Samothrakis is a Research Fellow at the Institute for Data Science and Analytics, University of Essex, Colchester, UK, whose research interests include artificial intelligence, reinforcement learning and neural networks.

Address: Institute for Data Science and Analytics, Wivenhoe Park, Colchester CO4 3SQ, UK.

Email: ssamot@essex.ac.uk

Shakashaka Challenge #4

Half-colour empty cells with triangles, as per the rules on p. 13. Challenge by Guten © Nikoli.
Bug or Feature?

Cameron Browne, Queensland University of Technology (QUT)

This article explores the relevance of the ‘bug or feature’ concept found in computer programming to the process of game design. Several examples are presented of successful games with either apparent bugs that proved beneficial on closer analysis, or actual bugs whose solution provided worthwhile emergent benefits. Game design is posed as a bug fixing process.

1 Introduction

There is an old joke among computer programmers that when a customer complains about a bug in a piece of software, it is sometimes easiest to just describe the resulting behaviour as a ‘feature’ of the program. For example, an error that inadvertently deletes a non-critical database every millionth entry might be described as a ‘memory saving feature’.

This analogy can be extended to the practice of game design, which often starts with an initial idea for a desired mechanism or behaviour and some preferred set of equipment, followed by an iterative process of identifying bugs in the design and fixing them, hopefully improving the game with each iteration. In this context, a bug refers to some undesirable behaviour and a feature refers to some desirable behaviour, resulting from the interaction between the rules and the equipment.

This article explores two aspects of the bug/feature dichotomy relative to game design; bugs that are actually features and bugs that can be turned into features. Sometimes it is not clear whether a particular design aspect is a bug or a feature, until the situation is studied in greater depth.

We are not interested in mere problem solving here; the design of any game could be described as one long process of bug fixing. Instead, we are interested in bugs that produce some emergent and unexpected benefits, either through side effects or through their solution, that add some significant feature to the game. The best examples play off the detrimental behaviour of the bug to produce some beneficial result.

2 Bug or Feature

This first set of examples includes games that contain apparent design bugs that have turned out to be positive features (or can be viewed as such), without modification.

2.1 Mambo

Mambo is a tile placement game for two players, Red and Blue, who take turns placing one of the Mambo tiles shown in Figure 1 (left) in any orientation to match at least one adjoining tile. The aim is to kill an enemy group by stopping it from further growth. For example, Red has killed the central Blue group to win in Figure 1 (right).

![Mambo Tiles and Red Win](Figure1.png)

The rules for Mambo were initially simpler and required only that players close an enemy group. However, this initial rule set had a problem in that players could create unplayable null points that no tile placement could match, such as the point marked × in Figure 2 and thus protect their groups from closure to avoid defeat ad infinitum.

![Null Point](Figure2.png)

The term bug refers here to an error in a program or mechanical system that produces unexpected behaviour.

http://www.cameronius.com/games/mambo/

This posed a serious problem, as such null points cannot be eliminated from the game without the addition of cumbersome rules for handling special cases. However, a change of perspective saved the day; simply changing the winning condition from ‘close an enemy group’ to ‘stop an enemy group from further growth’ solved this problem elegantly. Embracing null points as a part of the game added strategic depth and made wins much more likely to occur, resulting in a nicely balanced yet aggressive game in which both players are typically only one move away from defeat. Thus, an apparent bug proved to be a key feature of the game, with a simple change of perspective (and slight rule tweak).

2.2 Blue

Blue is a tile placement game for three players, denoted White, Blue and Grey, who take turns placing one of the tiles shown on the left of Figure 3 on a square grid each turn, such that edge colours match adjacent neighbours. Players score 10 points for each completed line of their colour and 1 point for each completed circle of their colour. Figure 3 (right) shows a game in progress, in which White has 11 points, Blue has 23 points and Grey has 31 points.

The tiles were initially intended to achieve fully packed tessellations, such as that shown in Figure 3. However, a bug emerged on the very first playtest, when spaces with two adjacent sides of the same colour proved unplayable, similar to null points in Mambo. For example, no tile can legally be played at the top right space in Figure 4, marked ×. This problem of unplayable spaces would have plagued most designers of tile placement games.

However, it soon became apparent that such unplayable spaces add an important tactical element to the game, as they allow players to judiciously block their opponents from completing point-scoring lines or dots. This feature proved so important to the game that additional tokens are provided in the published set, for players to explicitly mark such unplayable spaces as they occur. This is another case of an apparent bug that proved to be a key feature of the game.

2.3 Margo

Margo is a 3-dimensional version of Go, in which marbles played on a square grid stack upwards. The capture rules are similar to Go; groups with no freedom (adjacent empty board holes) are captured and removed. For example, Figure 5 (left) shows a black group in atari with one freedom remaining at the cell marked +.

Move 1 (right) removes this freedom and captures the group, but the question now arises as to what should happen to the pinned black piece marked z. Should this piece also be removed? If so, which of the pinning white pieces should drop down to fill the gap that its removal would leave? And how then should captured pieces that are pinned from all angles and hidden from the player’s view be removed?

The solution was easy: just leave such pinned pieces where they are, to remain active in the game as zombies. This soon proved to be one of the most interesting aspects of the game, with many tactical and strategic implications. For example, zombies allow groups to live safely with a

---

3http://www.nestorgames.com/#blue_detail
4Personal correspondence from designer Néstor Romeral Andrés.
5Atari is a term from Go that refers to the immediate threat of capture.
single eye, unlike Go, such as the white group in the lower left corner of Figure 6 (left).

![Figure 6. The white group is safe.](image)

Black cannot play in the corner cell as the white zombies would survive the move and mean that the black piece would have no freedom, which is not allowed. Instead, White can build from this surprisingly strong base and extend their group to attack further into the board (right). Zombies can be dangerous, and players must weigh the pros and cons of any move that would create enemy zombies very carefully. What at first appeared to be a crippling bug in the game’s geometry proved to be a key feature on deeper analysis.

### 2.4 The L Game

Figure 7 shows the starting position of the L game, in which two players (White and Black) take turns moving their L-piece to occupy four board cells (at least one of them different), then may optionally move one of the two neutral (grey) pieces to an empty cell. A player loses if they cannot move their L-piece on their turn.

The L game was designed almost 50 years ago by psychologist Edward de Bono ‘to produce the simplest possible game that could still be played with a high degree of skill’ [6, p. 120]. However, it has a serious flaw that would cripple it in the eyes of today’s designers: a simple strategy exists that allows players to avoid defeat indefinitely.

![Figure 7. The L game starting position.](image)

This strategy, discovered a decade after the game’s invention [2], would seem to eliminate the desired ‘high degree of skill’, as any player who knew the strategy was as unbeatable as any other. However, the L game still stands as an icon of elegant design [4], has a certain meditative appeal to it, working out the winning strategy can itself be an interesting challenge, and in terms of value for money not many games offer infinite non-trivial play for such simple equipment. So what is clearly a bug in the game could charitably be viewed as a feature in a certain light.

### 2.5 Reversi

When is a bug not really a bug? Consider the case of Reversi, which most readers should be familiar with, in which two players take turns placing a piece of their colour such that the move caps a line of enemy pieces with a mover’s piece at both ends, and the enemy line is then flipped to the mover’s colour.

The four corner cells of the board constitute degenerate cases as these are the only cells on which pieces can never be flipped once they are placed, as each corner represents a terminal cell of both lines leading into it. For example, the white piece in the top left corner of Figure 8 can never be flipped.

![Figure 8. A Reversi position.](image)

While these degenerate cases may be called a bug in some informal sense, this is really stretching the analogy, as they do not violate the game’s inherent ‘flip if capped’ rule; it is simply impossible to cap them. Instead, this makes the corners key strategic points in the game. This is clearly a feature of the game and not a bug.

### 2.6 Petty Diplomacy

In multi-player games, petty diplomacy is the tendency for temporary coalitions to form between players against their common opponent(s). This is described as a serious problem by Schmitberger, to the extent of making many existing three-player games unplayable [11, pp. 44–45]. However, is this really such a problem?
I make the distinction between strategic and non-strategic coalitions [7]. Strategic coalitions are those that occur within the framework of the game, for example, when two losing players temporarily cooperate to haul back the leading player, which is also known as the tall poppy syndrome. This can actually be cast in a positive light, as it provides a natural balancing mechanism that does not need not be stated in the rules and prolongs the contest. It is not necessarily a bad thing in all cases, provided that a player is eventually able to win by establishing a strong enough position to overcome such alliances.

Non-strategic alliances are those that form outside the game, such as secret pre-arranged agreements between friends against players they do not like. These are obviously anathema to games of strategy and should be eliminated as much as possible.

Many multi-player games include specific rules to minimise the effects of petty diplomacy. However, one notable exception is So Long Sucker! [9], which actually encourages strategic coalitions as an integral part of the game and reveals in the emergent chaos.

I believe that the dangers of petty diplomacy have been overstated, and that it may actually be a bug for some games but a feature for others. Unfortunately, this relies on players competing in the right spirit and not exploiting non-strategic coalitions.

### 3 Bug into Feature

This second set of examples includes games with design bugs which have been turned into design features, through judicious rule changes. Note that these are not simple bug fixes, but cases in which some inherent flaw has been turned into a positive feature with a simple twist.

#### 3.1 Go

The surround capture rule in the game of Go raises the danger of infinite cycles in play. For example, the position shown in Figure 9 (left), in which White captures a black stone (middle) but in doing so puts the capturing piece in immediate danger of recapture, which would repeat the board position from the previous turn.

To avoid such cycles, go has a ko rule that forbids immediate recapture, so that Black is not allowed to make the move marked × in Figure 9 (right). There are also two different forms of superko rule, called positional superko and situational superko, that forbid the repetition of any previous board position.

It would be sacrilege to any serious Go player to suggest that any aspect of the game is not perfect. However, I believe that infinite cycles in play are clearly a bug of the surround capture rule on the square grid,

It would be sacrilege to any serious Go player to suggest that any aspect of the game is not perfect. However, I believe that infinite cycles in play are clearly a bug of the surround capture rule on the square grid [6] and that the ko rule is a bug fix; a very good bug fix, admittedly, as ko battles have proven to be a key element of the game, to which significant study has been devoted.

Another solution to this bug is found in Capture Go, also known as Atari Go, in which the first player to make a capture wins the game. This neatly sidesteps the problem by cutting the game short before a cycle can possibly occur, but results in a less deep game. The ko rule is probably the most striking case in any board game of a crippling bug turned into a sublime feature, through a simple rule tweak.

#### 3.2 Gonnect

Gonnect is played using the rules of Go, but with two important differences [28]:

1. Players win by connecting their sides of the board with an orthogonally connected chain of their pieces.
2. Players cannot pass.

The first rule makes Gonnect a connection game [9] as much as a territorial game, giving it a unique and interesting character, while the second rule is necessary to fix an inherent bug due to deadlocks.

For example, Figure 19 shows a game in progress that has apparently reached an impasse,
as both players have safe groups and cannot intrude into their opponent’s groups due to Go’s ‘no suicide’ rule. However, the fact that players cannot pass means that the mover is forced to fill in one their own eyes, which makes that group vulnerable to capture and puts the mover in a losing position.

![Figure 10. A temporary deadlock in Gonnect.](image)

This ‘no pass’ rule elegantly solves the problem of deadlocks by simplifying the Go rules, if one considers the absence of passing to be the default case unless passing is specifically allowed.

### 3.3 Trax

Trax is a tile placement game in which players place tiles in an effort to make a closed path of their colour within a given area limit [3, p. 183]. Its designer, David Smith, encountered the same problem with unplayable spaces as found in Mambo and Blue, but over 40 years earlier.

Consider Figure 11 (left), in which Black has just placed tile $a$. White move $b$ (right) would create an unplayable space marked $\times$, as no tile has three black sides.

![Figure 11. Move $b$ would cause an unplayable space.](image)

To fix this problem, Smith added a forced move rule stating that if any tile placement creates any positions at which exactly one tile can legally be placed, then those tiles must be placed there as part of the move, possible triggering further forced moves. This not only addressed the unplayable space problem by making them much rarer in practice, but allowed beautiful sequences of forced moves that add a strategic dimension.

For example, Figure 12 shows how move $a$ triggers a sequence of forced moves $b, c, d_1$ and $d_2$ which complete a closed black path to win the game for Black. The forced move rule not only fixed a bug, but created a key feature of the game. There can be different ways to handle a given bug. While unplayable spaces are embraced in Mambo and Blue as part of the game, here they are greatly reduced by forced moves.

### 3.4 Chess

Another well-known example is the promotion of pawns in Chess. Consider the position shown in Figure 13, in which White has just moved their pawn to the far row and is about to promote it to a greater piece.

Since pawns can only move forwards, they would otherwise lodge on the far row with no possible moves and play no further part in the game, except perhaps as stationary blockers. Forward pawn movement is the bug and promotion is the fix.

Promotion adds a new dimension to the game, by providing a game-changing discontinuity – the weakest piece suddenly becomes the strongest piece – which makes each pawn a potential time bomb worthy of more respect than mere cannon fodder.

![Figure 12. Move $a$ triggers forced moves $b, c, d_1$ and $d_2$ which win for Black.](image)

---

7 Position from ‘How to Play Chess’: https://www.youtube.com/watch?v=FJ2CdBsTis4
Further, the choice of which piece to promote to is not always a given, and can provide an interesting problem in itself. For example, White would lose if they promote their advanced pawn in Figure 13 to the obvious Queen, and instead must underpromote it to a lesser piece – a knight in this case – to guarantee victory.

Another solution to the problem of forward-only pawn movement is found in the game of Breakthrough, in which players start with a row of pawns on their home row and win by moving one of their pawns to the far row\(^8\). Again, the bug is fixed by modifying the win condition to cut the game short as soon as the problem occurs, leading to a much simpler game.

3.5 Andantino

Andantino is another minimalist game designed by David Smith [3, p. 289], in which two players take turns placing a hexagonal tile of their colour such that:

1. Each tile placement is adjacent to at least two existing tiles.

2. A player wins by making a line of 5-in-a-row of their colour, or by completing a connected chain of their colour that encloses at least one enemy tile.

Figure 14 shows the starting position with Black to play (legal moves are indicated by dots). Without the ‘two existing tiles’ rule, it would be too easy to block line threats in Andantino. For example, move \( a \) in Figure 15 would easily negate White’s most obvious line threat, and the 5-in-a-row goal would play a reduced part in the game.

Most \( n \)-in-a-row games are played on boards, where distant threats are typically the key to winning. For example, Figure 16 shows a strong (winning) move for White at the point of intersection of two lines on the square grid in Gomoku. The ‘two existing tiles’ rule is an elegant fix to this problem, which exploits the underlying geometry of the game to make it harder to block lines.

There is also a practical need for this rule, as it ensures that the main body of pieces remains clustered together in a tight hexagonal formation, rather than widely spread out where placement errors can accumulate to make the (implied) hexagonal grid hard for players to see. Further, it adds an extra constraint that players can exploit for tactical gain, allowing interesting passages of forced play, and requiring players to plan ahead in order to achieve desired tile placements.

\(^8\)https://en.wikipedia.org/wiki/Breakthrough_(board_game)
For example, Figure 17 (left) shows a position with White to play. White move a forces blocking reply b, allowing White to play c, which sets up a winning fork (indicated +).

Figure 17. A winning play for White.

The ‘two existing tiles’ rule fixes the problem of weakened line threats to bring them more into the game, while adding a tactical aspect that requires players to plan ahead more carefully.

3.6 Swap Rule

Figure 18 shows a 9×9 game of Hex, which is played with extremely simple rules: players take turns placing a piece of their colour on an empty cell, and win by connecting the board sides of their colour with a chain of pieces of their colour [2].

Figure 18. A winning opening in 9×9 Hex.

However, this rule set has a crippling flaw in that the first player has a huge (winning) advantage if allowed an unconstrained first move. For example, Black should win after opening in the centre cell, as shown in the figure, unless they make a serious mistake.

To fix this bug, Hex is played with an additional rule called the swap rule or pie rule: in reply to the opening move, the second player may elect to swap colours instead of moving. This stops the first player making an overly strong opening move and results in more balanced games. For example, the dark cells shown in Figure 18 are proven to be winning moves for the opening player on the 9×9 board, and should be swapped by White [12].

The swap rule is a somewhat inelegant solution, that ruins the simplicity of this otherwise minimalist rule set, but is a necessary evil if the game is to work between players of similar skill. However, this rather clumsy hack is turned into something of an art form in the game of Unlur.

Unlur is played on a hexagonal grid of hexagons, by two players, White and Black. White aims to connect any two opposite board sides with a chain of white pieces, while Black aims to connect any three non-adjacent board sides with a chain of black pieces. An important twist is that the players are not initially assigned a colour; the game begins with a contract phase in which both players place black pieces, until one of them passes to declare themselves Black and their opponent White. Thereafter, players take turns placing a piece of their colour.

For example, Figure 19 shows a typical opening sequence after three moves. This position is probably strong enough that the next player should pass to claim the Black role.

Figure 19. An opening sequence in Unlur.

The swap rule has thus been embraced in Unlur and transformed into a contract phase that neatly balances out this game’s very unequal goals. In fact, this contract phase is an innovative feature that defines this game to a large extent; the decision of exactly when to pass is often the most important decision in a game of Unlur. This clumsy bug fix from Hex, rather than being downplayed, has been expanded and seamlessly integrated into the rules in this case.

---

4 Fuzzing

The sections above describe cases of bugs that turned out to be features, and bugs that were turned into features. However, some bugs remain bugs and have no easy fix; these are the game designs that you do not see. But perhaps even these might have a use.

Returning to the analogy of bugs in computer programming, fuzzing is the practice of stress testing software by deliberately introducing bugs into its input, in order to test its resilience to error [13]. Could a similar approach also be taken to game design?

This might entail systematically testing a given rule set with situations known to cause bugs with previous similar rule sets. For example, when designing a tile placement game, the obvious bug to test for is the occurrence of unplayable points, as seen in the Mambo, Blue and Trax examples. The designer should test their proposed rule set with every possible way that such a situation can occur, as a form of boundary value testing. Does the problem occur? If so, does the rule set handle it sufficiently? If not, can it be turned into a feature of the game? Does the problem produce any unexpected emergent behaviour that might inspire further improvements to the game or even entirely new games?

5 Conclusion

The examples presented above show how apparent bugs in a design are not always bad, and, even if they are, can often still be exploited to good effect. The most interesting features of games can emerge from fixing bugs.

It is important to identify bugs in the design process, but equally important to determine whether they actually are bugs. Before discarding an iteration of the rule set that appears flawed, ask yourself: is this behaviour a bug or a feature? In either case, how can it be exploited to best effect?

Some of the above examples (Go, Gonnect and Trax) were also included in my article ‘Embed the Rules’ published in the first issue of Game & Puzzle Design [7]. This is not coincidence – or laziness! – as it is an efficient design practice to fix bugs implicitly through the judicious use of rules or geometry, and to incorporate the resulting side effects into the game, where possible.

Acknowledgements

Thanks to Greg Schmidt for discussions, Russ Williams for alerting me to the practice of fuzzing, Stephen Tavener for suggesting pawn promotion in Chess and Néstor Romeral Andrés for the Blue example. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

References

Elegant Combat in War Games

F. Miguel Marqués, CNRS

This article is a personal view of how the combat mechanism used in traditional war games has evolved through the introduction of more elegant algorithms. It reviews some game examples to illustrate the innovations that have made the strongest impressions in the war gaming hobby.

1 Introduction

The war game was born in early 19th century Prussia when Georg von Reisswitz proposed Kriegsspiel (game of war) as an officer training exercise [1, 2]. In the 1850s, Helmuth von Moltke became Chief of Staff of the Prussian Army and generalised Kriegsspiel as a tool to prepare for the wars to come. A century later, war games entered the hobby world [3] as conflict simulations with a tendency for increasing detail; war games are hence also known as consims.

This tendency has led consim players to disparage more recent, lighter war games as ‘games with a war theme’ for their lack of detail. However, I feel that some have succeeded in abstracting the details to recreate the given conflicts in a more streamlined and elegant way. My definition of war game is thus broad: a game that strives to recreate a particular war, battle, or form of warfare. It ranges from the more traditional, highly detailed games that require many days to play, to the more modern, simpler and lighter ones.

1.1 The Fog of War

Apart from their theme, the main factor that sets war games apart from abstract open information games is the uncertainty inherent in war, which has been known as the fog of war since the time of Kriegsspiel. In Moltke’s words: No battle plan survives contact with the enemy [4, p. 35]. He did not believe war to be random, but realised that once war broke out, many parameters that escaped our control would mess with the original plans. Since the outcome could not be predicted, he viewed military strategy as the ability to prepare for a range of possible outcomes.

Moltke’s philosophy can be extended to games, since even in those with open information and deterministic confrontation mechanisms, a good player has to consider the decision tree with all possible reactions by the opponent, as discussed in [7]. War games add an additional layer of uncertainty on top of that tree, since combat results are generally uncertain.

War games typically recreate the fog of war around combat using three different methods:

1. Randomising the combat result.
2. Limiting the command of combat units.
3. Hiding information from the opponent.

Traditionally, however, they have mostly focussed on the first approach. The veil is removed from the game state, with all information available to all players, who are given unlimited command over their units (typically cardboard counters with strength and movement values on a hexagonal grid), and the fog of war is condensed to the roll of a die and a cross reference into a combat results table (CRT). For example, Figure 1 shows the CRT from the game Napoleon at Waterloo [1](1971), in which the attacker or defender retreat or are eliminated, depending on their strength ratio and die roll.

![Figure 1. A typical combat results table (CRT), from Napoleon at Waterloo.](http://bgg.cc/game/3573)

However, the condensation of the fog of war into a CRT is conceptually wrong. In game-theoretic terms, imperfect or incomplete information is replaced by randomness. Moreover, such a simple mechanism cannot easily reflect the peculiarities of each different conflict. Designers therefore attempted to simulate these by adding many fiddly rules, requiring, for instance, that modifiers be computed for the effects of terrain, weather or politics, using additional tables and charts. As a result, rulebooks for these games came to resemble law books, with almost as many exceptions as rules, and combat resolution felt more like work.

Even though these layers of complexity can help make such games proceed with greater historical accuracy, this approach makes war games less elegant in my opinion. I consider a game to be elegant when very few, simple rules lead to the game play that the designer intended in a streamlined way.

The algorithms do not have to be fast – nothing is faster than rolling a die – but the concepts must be simple, relevant to the situation they are modelling, and reflect the myriad of potential details in a way that is imperceptible to the players. This is possible even in war games, in what is called design for effect [7], leading to a similar game play using fewer and simpler rules. The scientist in me follows the principle of Ockham’s Razor and prefers the simpler option.

The CRT approach, besides its conceptual issues and its frequent layer of detailed exceptions, also has a pragmatic inelegance: one must read a table away from the board, where the action takes place. The ‘solution’ of printing the CRT on the board merely transformed this inelegance into an aesthetic one. Nowadays, the implementation of combat uncertainty has evolved towards more elegant schemes, by changing the randomisation methods, restricting command omnipotence, and hiding information. In the following sections I will discuss, through selected examples, different approaches that I feel have made combat resolution in war games more elegant.

2 Block Games: Napoleon: The Waterloo Campaign

Early on, an alternative to traditional war games appeared, which replaced cardboard counters with wooden blocks. Columbia Games is the best-known publisher of such block games, and Napoleon: The Waterloo Campaign (1974) is one of their best, appearing in a new deluxe fourth edition 40 years later (see Figures 2 and 3).

The blocks stand facing their owner (as in Stratego), and are rotated to modify a unit’s current strength, reflecting fog of war in two ways:

- Only the owner knows a unit’s identity.
- Only the owner knows a unit’s strength.

Another improvement arises when units meet and fight: there is no CRT. The blocks are laid down and moved to a separate battle board (Figure 3), in which they are set up in three columns (left, centre and right), and manoeuvre between them and fire at opposing units by rolling a number of dice corresponding to their current strength. Each successful roll reduces the enemy target’s strength, and battles end when an army retreats or one of its three columns collapses.

This system reinforces the feeling of command. Players must exploit their limited command through bluff and forced marches to outmanoeuvre the enemy at the strategic level (Figure 2), so that when combat occurs their units can quickly join to reinforce, plan retreat routes if a battle goes badly, and so on. Players have some control during combat resolution itself by manoeuvring at the tactical level (Figure 3). One might try to rout the enemy, or retreat before one’s own line collapses.

It does indeed increase the consim aspect in an elegant way. All information is on the boards with no need for complex arithmetic. Players experience the fog of war in terms of which enemy units they will face, and how strong they are. Rolling many dice also reduces random volatility, although this approach is more time-consuming, especially if there are many battles or a given battle lasts a long time.

\[2\] For a general discussion on elegance in games see [6].

\[3\] http://bgg.cc/game/1662
3 Battle Cards: 
**Hannibal: Rome vs Carthage**

In the mid-1990s, publisher Avalon Hill added cards to a war game world dominated by tables, charts and dice, with *We the People* (1994). These are known as *card-driven* war games, as the cards become the engine at both the strategic and tactical levels, by absorbing the fog of war and streamlining the rest of the design. An example that has stood the test of time is *Hannibal: Rome vs Carthage* (1996), which is discussed below.

![Figure 4. Hannibal: Rome vs Carthage, Smart Ltd Russian edition (photo by Evgeny T.).](image)

At the strategic level, a deck of event cards allows players to activate their generals and move their troops on the board, extend their political control over provinces, or cause the event described in the card to occur (Figure 4, left card). Detailed effects of historical events are conveniently self-contained on the cards, instead of being spread throughout the rulebook.

At the tactical level, another deck of cards resolves battles when units meet on the board. These are simple cards representing tactical manoeuvres (Figure 4, rightmost card), and in each round of battle the defender must match the attacker’s card to avoid defeat, and can then try to become the attacker themselves. The number of battle cards dealt for each player depends on several factors: the commanding general’s rating, number of units, and political leadership in the area. The player with more cards generally has a better chance to win, but luck and the fog of war – players do not know which battle cards their opponent holds – can lead to surprising results.

This approach of command limitation, card-driven strategic movement and political control, while different to typical block games, also provides the feeling of command in a very elegant way. Players manoeuvre to fight battles in the most favourable conditions – better general, more troops, in a friendly area, etc. – and during battle try to keep or gain the initiative, while considering possible retreat paths. Some old-school war gamers, however, have built variant CRTs to replace the combat deck with a single die roll.

Despite the originality of the card-driven approach and the absence of CRT, the game still has other tables printed on the board to handle attrition, retreats, sieges and naval combats. The rulebook is in the traditional style: relatively long with many subsections and notes. While I do enjoy the card-driven war game family, these games do not sit at the top of my elegance scale.

4 Blind Chess: 
**Bonaparte at Marengo**

In the mid-2000s a breath of fresh air came from Simmons Games with *Bonaparte at Marengo* (2005). All the components can be seen in Figure 5: some long wooden blocks and three tokens; no cards, no tables, no charts... not even dice! It is interesting to note that the designer was initially just trying to go back to the roots of *Kriegsspiel*, with its rectangular blocks on a gridless map, and in doing so ended up with a game far removed from conventional modern war game design.

![Figure 5. The Austrians push the French line East in Bonaparte at Marengo (photo by R. Simmons).](image)

The units are red and blue wooden sticks, evoking 19th century war maps, and the board consists of various polygonal locales with units deployed on their edges to confront adjacent enemies. To avoid charts, numbers were simplified to the minimum: units normally move only one

---

4 [http://bgg.cc/game/620](http://bgg.cc/game/620)
5 [http://bgg.cc/game/234](http://bgg.cc/game/234)
6 Note that publisher Phalanx has recently announced the 20th anniversary edition of Hannibal: Rome vs Carthage, which will replace these additional tables with new, more elegant mechanisms.
7 [http://www.simmonsgames.com/products/Marengo](http://www.simmonsgames.com/products/Marengo)
step, and their strength is 1, 2 or 3. There are no terrain charts; slow terrain is reflected by smaller polygons, and combat restrictions are printed on their edges. Combat resolution is simple – the higher strength sum wins – and combat is deterministic, as in Chess, so no CRTs are needed. Where, then, is the fog of war?

As in other block games, each unit’s information is visible only to its owner. The units are very simple, with only a type (infantry, cavalry or artillery) and strength (1, 2 or 3). Units are typically in the middle of a locale, but move to an edge to block enemy access. Players can attack an adjacent locale by:

1. **frontal assault** against a blocked edge, which causes losses to both sides, or
2. **manoeuvre attack** against an undefended edge, which can cause all enemy units to retreat and losses only to the defender.

A player has a command limit of three movements or assaults per turn. There is no randomness other than initial setup of the French units. In practice, this game turned out to be a spectacular case of design for effect, surpassing even the designer’s expectations. The red and blue tokens in Figure 5 track the morale of each army, decreasing with every loss, and a player loses if their army’s morale reaches zero (territorial control serves as a tie-breaker). Therefore, breaking the defender’s line mostly by assault will not be very efficient and can cost a player the game, so the emphasis naturally falls on manoeuvre attacks that do not cost morale, leading to battle lines such as the one shown in Figure 5.

This looks and feels like a Napoleonic battle. The defender forms a line along several edges and keeps some reserves to avoid being outmanoeuvred. The attacker targets flanks, forcing the enemy to commit all units to some edges of a polygon, so that the attacking cavalry can cross another edge with little opposition. Further, players will occasionally need to assault weak points of the line to unblock stalemates, with sometimes overwhelming domino effects.

This game is a masterpiece and would be **the** definition of elegance itself, except for some fiddly issues regarding road movement. Although the rules are simple, they are very different to traditional systems, so that first-time players are sometimes confused, and previous experience with war games might even be more of a hindrance than a help. Its lack of randomness makes it similar to an abstract strategy game, and as such it can be unforgiving, and a novice will have a hard time against an experienced player as an early mistake can be fatal. The designer further developed these ideas in two larger games, inspired by the battles of Austerlitz and Gettysburg, but Bonaparte at Marengo deserves all the credit as the pioneer.

## 5 Restricted Command: Command & Colors: Ancients

The previous examples implemented part of the fog of war by limiting command. Another elegant approach appeared at the turn of the century, in the US Civil War game Battle Cry (2000), the first of the Commands & Colors family by Richard Borg. Different versions of this system have been published since in various settings, including World War II, Ancient, Napoleonic and Samurai, but the most representative example is probably Commands & Colors: Ancients (2006).

![Figure 6. Hannibal traps the Romans in Commands & Colors: Ancients (photo by Rob Judy).](http://bgg.cc/game/14105)

Figure 6 shows the main game components: standing wooden blocks, cards and dice. The standing blocks have nothing to do with fog of war as each has stickers on both sides, giving the impression of armies facing each other without the need for miniatures. They give the game an ancient look, and one can almost imagine Greek or Roman generals pushing these wooden blocks in their command tent on the eve of battle. Other versions of Commands & Colors, with fewer unit types, do use miniatures.

Commands & Colors has no CRT, and battles are resolved by a single roll of special dice. Units are light (green), medium (blue) or heavy (red), and the dice exhibit a face of each colour (the ‘Colors’ aspect), a generic face (swords), a leader.
face (helmet), and a flag. Each unit attacks with a given number of dice, and scores a hit for every target unit colour, every sword if the unit uses them, and every helmet if supported by a leader, plus a retreat for every flag. The result is thus immediately clear, with no need to check tables or calculate odds, which was one of the designer’s aims. It is possible to calculate the probabilities which are implicitly encoded in the dice and unit types [9], but most players follow their intuition.

What really sets Commands & Colors apart is the set of cards (the ‘Commands’ aspect) that players use to activate units on the board. Not only is unit command limited, but it is restricted. Some cards only activate units of a given type, or in a given section, or connected to a leader, or in a given formation, etc. Players may therefore sometimes have no cards with which to activate a given unit, just as their historical counterparts could not always force units to obey their will. Some players, used to having full access to all of their units in a game, may introduce ‘house rules’ allowing any card to activate any unit, but they would then miss out on the intended experience.

What are traditional, complex consims trying to simulate? Ancient commanders’ most difficult challenges were situational awareness and command transmission, and this still applies to tactical warfare today. Where is the enemy? Where are my units and what are they doing now? How do I abort or rectify this manoeuvre? Traditional war games pretend to simulate war, but by allowing perfect intelligence and communication they fail to capture the essence. They become puzzles of perfect information that players try to solve, by analysing the most efficient manoeuvres and strength ratios that lead to victory. If ancient commanders were given the chance to play one of these games, I doubt whether they would recognise the experience as similar to their own.

In the chaos of battle, commanders lost perspective and relied on mounted messengers amidst the noise and dust clouds. Alexander the Great, for example, anticipated important manoeuvres by increasing the number of messengers in certain parts of his line. But messengers could be killed, orders misunderstood, men could panic, delays desynchronise manoeuvres, etc. The Commands & Colors system captures this crucial aspect of war in a unique and elegant way; historically better commanders are dealt more cards, but there is no certainty that a unit will do what the player wants at any given time.

This mechanism makes these games not only simpler, but better consims, incorporating elements such as bluffing and boldness. In more traditional war games, players will not try to vainly cross a killing zone if enemy units will automatically shoot them down. However, in Commands & Colors players may dare to do so, since those defending units may not be able to receive orders, giving time to attack them. Players can guess about their opponent’s situation and plans by observing how they play their cards… or are they bluffing? A canny player might have no cards to support a flank offensive, but still re-deploy units there as if they did. An ancient commander would probably enjoy such a game.

6 Support Chains: Tetrarchia

In the next two sections, I will turn to the lighter end of the scale and discuss games that use generic tokens in a more abstract way, but which still depict wars and use a combat mechanism.

The first example is my latest design, Tetrarchia [13](2015), a cooperative war game in which the four Emperors, or Tetrarchs, that ruled Rome from 293AD to 305AD must preserve the Empire against internal and external threats. The 3rd century saw a progressive deterioration of the Roman world. Revolts and armies (led by usurpers or enemies) triggered each other in a dreadful spiral towards disintegration. Diocletian, by sharing power with the three other Tetrarchs, reversed the tide through extensive campaigns requiring pacification of the provinces in the area and close cooperation in order to defeat the armies.

Revolts are handled by placing control tokens and removing revolt tokens, but when Emperors and armies meet a battle takes place. Combat is resolved by rolling a pair of dice, Roman and Barbarian (see Figure 7), and the historical aspects of pacification and cooperation are implemented through territorial support, inspired by Hannibal: Rome vs Carthage, in which armies on friendly

---

12 In different implementations of the system, this might be: a heroic hoplite facing elephants; a GI attacking Tiger tanks; a French Cuirassier charging a British battery; and so on.

13 http://nestorgames.com/index.html#tetrarchia_detail
provinces get more combat cards, and through combined attack. Barbarians add to their roll the number of revolt tokens linked to them in a chain, Romans do the same with their control tokens, and the result is doubled if there are other armies or Emperors linked to the two combatants.

The Emperors only have a few tokens, and need at least six on their Empire’s frontiers to win the game. Meanwhile, armies advancing towards Rome lay a trail of revolt tokens along their path. Players should thus attack those armies early before they gather support, or cut them from their base to have a chance to beat them. The system forces players to plan traps along the Barbarian paths in order to link several Emperors to the army when combat triggers.

The arithmetic of battle is simple – add tokens to a roll and possibly double it – which resolves combat quickly and promotes strategies associated with the geometry of the board. The result was a very pleasant example of design for effect.

7 Hidden Reserves

The previous example resolved combat using dice, complemented by tokens on the board. An alternative is to use the tokens on the board, complemented by cards.

7.1 Hellas

In Hellas14 (2002), by late designer Franz-Benno Delonge, two Greek city-states fight for control over ten cities, on a growing modular board.

There are two types of pieces: ships explore to expand the board, and hoplites attack other cities. Combat is deterministic, as in Bonaparte at Marengo, and is resolved by comparing each side’s number of hoplites. Fog of war is provided by three decks of God cards, which can be played to affect combat, among other purposes (Figure 8).

Hellas is a very light abstract card-driven war game: even if the two sides are ostensibly Athens and Sparta, the game does not simulate any particular conflict, but rather the warlike character of the era. The hoplites on the board give an indication of each army’s strongholds and offensive plans, but players’ secret cards act as their hidden reserve. When players reinforce, they can do so with hoplites, ships or cards. Cards can have decisive effects, e.g. permitting a second attack in a turn, a farther attack than normally possible, cancelling an opponent’s card, and so on.

If one approaches Hellas as a traditional war game and focusses on the armies, then the deterministic combat appears too simple and the card element too dependent on luck. However, if one focusses on the cards instead, then the hand management over the changing board will create enjoyable tension from the start, requiring balance between exploration, conquest and reinforcement. The result is a unique conjunction of card game, tile-laying game, and light war game. However, its design is inelegant in some aspects, as players require a minimum knowledge of the cards to compete effectively, and the luck of the draw may favour one player over the other as both draw from common decks.

7.2 Popular Front

A similar principle is exploited in Popular Front15 (2010), in which up to six players join the two sides that fought the Spanish Civil War.

Again, generic tokens on the board represent troops (Figure 9), basic combat is deterministic with the higher number of troops winning, and cards represent a hidden reserve. Without adding complexity, a few twists make this game much less abstract and closer to a streamlined simulation of the conflict, despite the lack of special rules and faction powers. For example, the turn

---

14http://bgg.cc/game/4529
15http://www.numbskullgames.com/#popular-front/cpvc
order is randomised each round, adding another layer of fog of war.

Each player has their own identical deck, diminishing the luck of the draw, and the card types are few, with mostly intuitive effects. Players play cards face down successively until one passes, which creates tense combat resolution. Cards can also be played to get more troops, move them, and lead the political race. Players must optimise their network and out-think their opponent in order to keep their reserve mostly for combat.

As in Hellas, the number of troops per city is limited, and players must leave at least one behind to defend the city, so offensives naturally lose momentum. The game clock advances on the victory track every time a player conquers a city, so players must plan well to retain possession of their conquests.

### 7.3 Friedrich

A different example of the card reserve idea is found in Friedrich\[^{16}\](2004), from Histogame. The designer defines it as a novel concept that, without being a strict simulation, recreates the nature of the Seven Years War\[^{10}\], that in 1756 opposed Prussia to the main continental powers of the era. Players move generals on the board carrying armies, as in Hannibal: Rome vs Carthage, but here the fog of war involves hidden information, as players privately record their troop numbers and allocations but only reveal the total number to their opponents.

When combat occurs, players announce their armies and complement them with tactical battle cards. The unorthodox battle cards come in four traditional suits (clubs, hearts, spades and diamonds, shown in Figure 10). A player can only play cards of the suit matching the map sector in which their general is currently fighting, and whichever player is currently trailing in total strength chooses to play or to stop the battle, in order to limit damage and save cards for future battles. Traditional war gamers criticise the card aspect as too abstract, but it leads to interesting hand management strategies and over-the-board manoeuvring that recreates the war nicely.

The same cards allow the purchase of reinforcements and, in the game’s successor Maria\[^{17}\], the designer added a political function similar to that in Popular Front. These very different implementations display the richness of the use of cards as hidden reserves in combat, especially when they have additional uses that introduce strategic choices.

## 8 Elegant Combat

This article started with the paradigm of traditional 20th century war games: cardboard counters on hexagonal grids, perfect information and unlimited control, charts, modifiers, and finally a die roll on a CRT. Within this framework, history and fog of war could be recreated only through complicated mechanisms, which diverged from the elegance of the original Kriegsspiel. We have reviewed different attempts by designers to escape this paradigm by introducing new concepts in a quest for the origins, aided by new materials: wooden blocks, cards, special dice, etc. These all share the goal of design for effect, i.e. abstracting away details to allow simpler mechanisms to recreate a given war or battle, and implement the fog of war by limiting intelligence or control.

Elegance is subjective. However, there is no denying that wooden pieces on beautiful maps evocative of the era, the lack of tables and charts, and simple combat resolution, combine to make war games more elegant, both aesthetically and mechanically. For example, wooden blocks convey limited intelligence in a straightforward way while improving a game’s visual appeal. Cards with multiple functions add layers of decision and variability, and player choices influence whether specific events will happen at all; history can play out differently with each game.

However, games that require familiarity with a large deck of unique event cards are not strictly elegant. Friedrich, with a standard deck of cards, is more elegant, and the even more minimalist and cardless Bonaparte at Marengo is hard to beat. Many war game designers aspire to create such an elegant minimalist game, with few units, no cards and no dice. The challenge lies in adding a theme to recreate a historical event, as Simmons achieved successfully, rather than merely creating another abstract strategy game.

---

\[^{16}\]http://www.histogame.de/e_friedrich.html
\[^{17}\]http://www.histogame.de/e_maria.html
When balancing elegance with playability, however, the absence of luck makes this truly minimalist approach very unforgiving, and Commands & Colors: Ancients becomes my first choice. It abstracts many things, but leads to session reports that sound like real battle reports, and illustrates perfectly the notion of design for effect. It successfully recreates the fog of war in command transmission by restricting orders, which may not suit players used to having total control over their armies. But this adds tension, bluff and boldness; all improvements in my view.

The last sections covered lighter games with territorial support, cooperation, and hidden reserves. Simple tokens represent armies, but they are only the tip of the iceberg, as a lot of an army’s strength lies in hidden cards. When two of these icebergs clash, the resolution is not random, but depends on players’ management of the sunken part, and outmanoeuvring the opponent replaces luck. Tips of icebergs moving on the board add tension, bluff and strategy, and as a bonus the games look more aesthetically elegant.

9 Conclusion

Elegance in simulated combat is not a tradeoff for realism. Pushing pieces on a board has little to do with war, and no games simulate the real experience, otherwise few would want to play them. However, the designer should aim to provide some feeling of what it was like to be in that situation. The examples presented in this paper transform perfect information puzzles, with the sporadic tension of dice rolls, into games in which hidden information and limited command increase tension and encourage bluffing. If real commanders were asked, I believe that they would choose the latter games as better recreations of their wars.

Tradition is a heavy burden, though. However innovative or elegant new mechanics might be, veteran players often raise their shields against novelty. The war gaming hobby needs new and younger members to survive, and the designs that are now emerging show how alive it is. By making combat resolution more elegant, especially the implementation of the fog of war, modern designers simultaneously reach back to the roots of Kriegsspiel and ahead to new players, showing that ‘war game’ is not just a synonym for complex calculations and tedious checks of tables and charts. As war game designer Richard Borg points out: You should fight the player in front of you, not the rules.18

Acknowledgements

Thanks to Brady Severns, Kevin Duke and the reviewers for their suggestions; Joel Eddy, Evgeny T., Rachel Simmons, Rob Judy, EndersGame, Chris Norwood, and Stefan M. for image permissions; BoardGameGeek for hosting the images and information about the example games; and the journal editors Cameron Browne and Russ Williams for their help developing this piece.

References


Miguel Marqués was born in Spain and works as a nuclear physicist at the French CNRS. He is interested in board games, sports and history, and designs games based on these topics. Address: LPC (ENSEAEN), 14050 Caen cedex, France. Email: marques@lpccaen.in2p3.fr

---

18 Personal correspondence.
1 Introduction

Graph theory has historically been an orphan step-child of mathematics, a field that has received very little respect. A good deal of the progress in graph theory has been made by recreational, as opposed to professional, mathematicians, adding to the feeling it was not serious.

However, with the advent of the computer age, graph theory has proven most useful, with graphs serving as a language for designing and discussing computer networks, transportation networks, and even biological networks such as food webs and interaction patterns of biomolecules. Graph theory has now become a respected and practical branch of mathematics.

What is the relevance of graph theory to games and puzzles? Almost any game or puzzle played on a board implicitly involves at least one abstract combinatorial graph, which is a term that refers to the relationships between locations, objects, concepts and actions in the game. This article is intended to serve as an introduction to such graphs, which are often woven into the fabric of the game.

We will explore graph theory by scrutinising puzzles and games from 1736 to the present, and we will see how recognising and using this theory as a tool can aid in a more thorough and careful design process, as well as more balanced and intentional designs.

2 Reference Games

In this section, we introduce two historical examples of graph theory in a puzzle and in a game.

2.1 The Bridges of Königsberg

In the city of Königsberg, corresponding to modern day Kaliningrad, the citizens had an after-dinner activity. A river with a substantial island runs through the city and seven bridges joined the north shore, the island, the south shore, and the upstream angle between two branches of the river. The goal of the activity was to cross each bridge exactly once and return to start. This problem is shown graphically in Figure 1.

The great mathematician Euler proved that this problem has no solution [3]. His solution contained the seeds to a rapid technique for establishing the solvability of a whole class of puzzles and an effective technique for constructing solutions. This technique and the class of problems are described further in Section 4.

2.2 Icosian

Mathematician William Rowan Hamilton devised the game Icosian [2] shown in Figure 2.

For more mathematically detailed coverage of this topic, see the classic Winning Ways for your Mathematical Plays [1] and Richard Guy’s ‘Graphs and Games’ [2 chap. 9].

1https://www.ria.ie/library

The Icosian board is a flattened representation of a dodecahedron\(^3\) and the aim is to place numbered pegs in the board holes to form a closed path in numerical order, along the lines. Icosian was not a commercial success; it was both somewhat intimidating and also, once learnt, too easy. We will get back to Icosian shortly, after introducing some basic graph theory terminology, from \[4\].

### 3 Some Graph Theory Terms

A **combinatorial graph**, or simply **graph**, is defined by a set of **vertices** and a set of **edges**.

The two vertices at either end of an edge are called its **ends**.

The ends of an edge are said to be **adjacent** or **neighbours**.

The **degree** of a vertex is the number of edges that end at that vertex.

If \(v\) is a vertex then the set of all its neighbours is written \(\Gamma(v)\) and called the **neighbourhood** of \(v\).

A **walk** is a list of vertices in which adjacent elements of the list are neighbours.

A **path** is a walk that does not repeat any vertex.

A graph is **connected** if there is a path between any two vertices.

A **circuit** is like a walk but repeats one vertex; the first and last vertices must be the same.

A **cycle** is a path that repeats one vertex; the first and last. The pentagon around the outside of the graph in Figure 4 is a cycle of length 5.

A graph that can be drawn in the plane without any of its edges crossing one another is called a **planar** graph. Edges are permitted to meet at vertices, of course.

An **Euler cycle** or **Eulerian circuit** is a walk that includes every edge exactly once, and begins and ends at the same vertex. Note that this is not actually a cycle, despite the name.

A **Hamilton cycle** is a cycle that includes every vertex in the graph.

---

\(^3\)A dodecahedron is a solid with twelve pentagonal faces, that many readers will recognise as a 12-sided die.
the goal of Icosian is to find a Hamilton cycle in the graph shown in Figure 4.4

3.1 Drawing Graphs

From an abstract mathematical point of view, it is irrelevant how a graph is drawn. But from ergonomic and aesthetic viewpoints, the physical layout of graphs matter a great deal. The following algorithm by Tutte 5, used to produce the drawing of the Icosian board given in Figure 4, is therefore useful for visually presenting graphs.

Given a connected planar graph \(G(V, E)\), find and display, as a regular polygon, a cycle of the graph that will not disconnect the graph if its vertices are removed. Freeze the vertices of this cycle, and for all other vertices, solve the system of linear equations given by the principle ‘my position is the average of my neighbour’s positions’. This will produce vertex positions permitting all edges to be drawn as line segments without any crossings.

For designers who do not want to solve large systems of equations, there is a slower but easier approximation: repeatedly select a vertex not in the frozen cycle and move it to the average of its neighbour’s current positions. This iterative process converges to the result of Tutte’s algorithm.

3.2 Icosian or Dodecian?

The Icosian board can be viewed as twelve pentagons, representing a flattened dodecahedron, so one may wonder how the name ‘Icosian’ – which suggests an icosahedron 4 – came about. This probably arose from an interesting fact about Platonic solids 5, i.e. solids in which all faces are instances of the same regular polygon, in that a dodecahedron is the dual of an icosahedron.

A dodecahedron has twelve pentagonal faces and twenty vertices, with three faces meeting at each vertex. An icosahedron has twenty triangular faces and twelve vertices, with five faces meeting at each vertex. Using planar duality 4, we may exchange the role of faces and vertices to transform a dodecahedron and icosahedron into one another.

Two vertices in the original shape are at the ends of an edge, while two faces in the dual solid are adjacent if the share an edge, each having that edge as one of their sides. Shared edges form the basis of the adjacency relationship between faces. A cube and octahedron have a similar relationship, and a tetrahedron is dual with itself. Similar dualities can be seen in planar forms, as the regular triangular and hexagonal tilings of the plane are transformations of each other, and the regular square tiling transforms into itself.

We can speculate that Rowan Hamilton considered the graph in his game to be equally valid when viewed as a dodecahedron or icosahedron, thanks to their duality, and that he chose the more elegant name ‘Icosian’ over ‘Dodecian’ or the rather awkward ‘Dodecahedrian’.

4 Non-Solution of Königsberg

The seven bridges problem is unsolvable. When a citizen enters a land mass, they cross a bridge. When they leave it, they also cross a bridge. This means each traversal of a land mass uses up two bridges. In order to return to the starting point, a citizen must leave it by one bridge and return by another. This means that the number of bridges at each land mass must be even, because they are used up in pairs. The north shore, south shore, and upstream angle each have three bridges ending at them; the island has five. The situation – all land masses having an odd number of bridges ending there – could really not be worse. This result can be stated formally as follows:

**Theorem:** A connected graph has an Eulerian circuit if and only if all of its vertices are of even degree.

The argument in the first paragraph of this section shows that even degrees are needed, but we still need to show that if there are even degrees then an Eulerian circuit exists. To demonstrate this, let us generalise the seven bridges problem to a well-known drawing problem.

![Figure 5](image)

**Figure 5.** This graph can be drawn without lifting the pencil, e.g. ABCDGFBEHGFDEAHDA.

A well-known type of puzzle is to determine whether a shape, such as the one shown in Figure 5, can be drawn without lifting the pencil.

---

4 An icosahedron is a Platonic solid with twenty triangular faces, often used for a 20-sided die.
5 See: [http://mathworld.wolfram.com/PlatonicSolid.html](http://mathworld.wolfram.com/PlatonicSolid.html)
If we consider the places where three or more lines meet as vertices, and the lines between these meeting places as edges, then we see that this puzzle consists of finding an Eulerian circuit.

4.1 Euler’s Construction

Consider a connected graph in which all vertices have even degree. The idea is to repeatedly identify circuits and splice them together until all edges have been used.

Starting at any vertex, follow a circuit without using any edge twice, until no more steps are possible. Since all vertices have even degree, the ability to reach a given vertex means there is another edge available for escape until returning to the starting vertex. If we return to the start vertex and have an Eulerian circuit, then we are done, otherwise we remove the used edges from consideration; since a circuit was removed, all remaining edges will have even degree. Repeat the circuit-discovery procedure and splice in the new circuit to the old one at the new starting point. After every iteration there is a circuit, so eventually all edges will be used.

4.1.1 Example Application

To solve the puzzle in Figure 5, start with the square ABCDA. Now, starting again at A, trace out the circuit AHDGCFBEA and splice this into the existing square as: ABCD-AHDGCFBEA. Splicing replaces the terminal A with AHDGCFBEA. What is left? The inner square EFGHE. Splice this into the growing circuit at E: ABCD-AHDGCFB-EFGHE-A, remove the splicing connectors, and we now have an Eulerian circuit with vertex list ABCDAHDGCFBEFGHEA. Note that this solution is different to the one given in the caption of Figure 5; many different constructions can be generated.

If it is not required to start and end at the same vertex, then more shapes can be drawn without lifting the pencil, and these will be shapes with only two vertices of odd degree. Consider the shapes shown in Figure 6; the graph on the left has two vertices of odd degree (highlighted). If we leave such a vertex and never return, then it can have an odd degree. Similarly, if the final vertex has odd degree, and is not the starting vertex, then it can be of odd degree as well. The graph on the left can be drawn without lifting the pencil, but the graph on the right cannot.

This way of using up all the edges is called an Euler walk. We thus add to the Eulerian circuit theorem the following Euler walk theorem:

**Theorem:** A connected graph with exactly two vertices of odd degree has an Euler walk beginning and ending at the vertices of odd degree.

This completes the description of shapes that can be drawn without lifting the pencil. Both an Eulerian circuit and an Euler walk cover all lines in a shape. Note that the constraint in both theorems – that the graph is connected – is needed to avoid lifting the pencil.

5 Icosian and its Generalisation

The problem of finding an Euler walk or circuit – or proving that one does not exist – is completely solved and fast in practice. Hamilton cycles are another kettle of fish.

Determining whether a graph has a Hamilton cycle turns out to be NP-complete. In practice, however, most graphs have a Hamilton cycle, and sometimes they are easy to find. There is a joke among graph theorists that a simple approximation of this algorithm is to simply say ‘yes, this graph has a Hamilton cycle’ without even looking at it, which will work most of the time. In the case of Icosian, an experienced puzzle solver should be able to spot a Hamilton cycle in seconds, such as the solution shown in Figure 7.

In computer science, NP-complete problems are computationally difficult, and, in the worst cases, the fastest known algorithms still require time exponential in the size of the input.
However, the Icosian board is a wonderful design and can be used for other more interesting graph-related games and puzzles. One example is the game Graph Domination, described next, which was developed on the Icosian board but can be played on a variety of boards.

6 The Game of Graph Domination

Graph Domination is a collection of games, invented by the author, which demonstrates the design of a game based on a graph.

**Rules for Graph Domination**

1. One player is designated *Red* and the other is designated *Blue* by mutual agreement. Blue moves first, then players take turns moving.

2. Each player has an *entry point* for their colour, as shown Figure 8.

3. The board consists of vertices (circles) connected by edges. Two vertices with an edge between them are *adjacent*.

4. On their turn, a player may either place one of their tokens onto their entry point, if it is empty, or move one of their tokens from a vertex to an empty adjacent vertex, except as specified in rule 5.

5. No player may place or move a token on a vertex that is adjacent to a vertex occupied by an opponent’s tokens.

6. A player must place or move a token if possible, otherwise they must pass that turn.

7. Play continues until neither player can move.

8. A player’s score is 3 points for each vertex occupied by one of their tokens, plus, for each empty vertex, 1 point for each of their tokens adjacent to that vertex.

9. The player with the highest score wins. If scores are equal then the game is a tie.

10. Players may wish to play two games, starting once in each, and add their two scores.

The first version, for two players, requires tokens in two colours, and is played on the board shown in Figure 8, with two (symmetric) entry points indicated. Blue enters at the upper (blue) vertex, and Red enters at the lower (red) vertex. The entry points can be considered as opposite vertices of a dodecahedron, or (equivalently) opposite faces of an icosahedron.

Figure 8. The Icosian board modified for Graph Domination.

Figure 9 shows a possible final position of a game of Graph Domination. Blue’s score (uppermost seven filled circles) is $3 \times 7 + 7 = 28$ for seven occupied spaces and seven adjacencies (one of the empty circles has two blue tokens adjacent to it). Red’s score (lowermost seven filled circles) is similarly $3 \times 7 + 7 = 28$, so the game is a tie.

Figure 9. This game is tied.

6.1 Is the Board Fair?

If the board is rotated $180^\circ$, and the inner and outer pentagons exchanged to match up their vertices, then this transformation preserves adjacencies and swaps the red and blue entry points. This is an example of a *symmetry* or *graph isomorphism* that demonstrates that there is no positional advantage to either player.

Note that this does not address any potential first move advantage; it is not yet known whether
either player has an advantage in Graph Domination played on the Icosian board. Rule 10 (two games per match) was added for this purpose.

Any permutation of a graph’s vertices which preserves the edge relations is called a symmetry of the graph or graph isomorphism. If the positions and moves of a game correspond to the vertices and edges of a graph, then a symmetry that exchanges the starting positions of two players clearly shows that neither starting position has a positional advantage over the other. This is one way that graph theory can be used to help analyse a game design.

6.2 More Graph Domination Boards

Figure 10 shows a Graph Domination board with three rings of seven positions, and Figure 11 shows a board with five rings of three positions. These boards are parts of an infinite family with two or more rings and three or more positions per ring, which allow Graph Domination to become as complex as desired.

These boards share with the Icosian board the property that there is a symmetry that exchanges the two entry points. This can be seen by visualising the board as the interior view down a length of pipe; turn the pipe to swap its ‘near’ and ‘far’ ends, then rotate it to realign the two entry points.

6.3 Notes on Strategy

Rule 5 of Graph Domination is critical, implementing the idea of zones of control, similar to those in war games. The presence of such zones of control makes the strategy more complex, and makes Graph Domination a better tool for teaching game design. It has been used as an outreach activity with high-school students, who made the following observations on strategy:

- Do not let the opponent move adjacent to – or worse, on to – your starting point, or they can block you from adding further tokens.

- There is a trade-off between getting many tokens on the board and deploying them quickly, in order to block the opponent’s advance. Once a frontier is formed, its owner can fill in behind it without difficulty.

- Different boards can require different strategies. Boards with fewer layers or fewer rings lead to shorter games.

- It is not meaningful to compare the scores of games played on different boards.

Typically, when a class was shown a few alternate boards, some students would immediately go off and invent more complex boards. The boards used in this article are all planar graphs and avoid giving any player a positional advantage. There are many other types of boards that could be used, such as those formed by non-planar graphs, but we will leave those for future articles. Students often asked whether the game can be played with more than two players; it can be, as shown in Figure 12.

The three-player board shown in Figure 12 differs from the two player game in having two entry points per player, but also has a provable lack of positional advantage. The paired entry points for each player solve a serious problem with the potential to block a player’s entry point very quickly when there are two other players. A majority of students did not like this game as much as the two-player version, as it was easier to play to spoil the game than to win.
Figure 12. Setup for the three-player game.

### 7 Graph-Board Correspondence

The famous *four colour problem* [6] states that, if we colour a map so that countries sharing a border have distinct colours, then no more than four colours are needed. This proof involves transforming a map into a graph.

The *graph of a map* consists of vertices corresponding to the countries and edges corresponding to shared borders, as shown in Figure 13. A larger example, showing the graph of a Risk board, appears in [7].

Regular tilings are popular game grids. Many games are based on other regular polygonal arrays. For example, Figure 15 shows game grids based on regular octagons and squares (left) and a single irregular pentagon (right).

Regular polygonal tilings, called *grids*, are used to help regularise unit movement and distance measurements in many games, especially war games. Less regular tilings do not quantise movement and distance so neatly, but can enable other interesting game mechanisms to emerge.

### 8 Other Graph-Based Games

This section describes two additional mathematical games with strong underlying graph bases.

#### 8.1 Hex

The game of Hex [1, vol. 2] is played on a regular hexagonal tiling such as that shown in Figure 16. Exactly one player must win each game.

**Hex:** Two players take turns placing a piece of their colour at an empty cell, and win by connecting the sides of their colour with a connected chain of pieces of their colour.

Hex is interesting, in part, because it has been shown to have a very high computational complexity. It is a good example of a game with a strong underlying graph-based nature [9] with very simple rules but complex strategy.
8.2 Sprouts

The game of Sprouts, invented by mathematician John Conway [1, vol. 2], was popularised in the July 1967 edition of Mathematical Games, a regular column in Scientific American. Sprouts starts with a sheet of paper with a given number of dots drawn on it, and is played as follows.

**Sprouts:** Two players take turns drawing a curve from one dot to another (or from a dot to itself), such that:

1. a) No dot has more than three incident curves.
2. b) No curves ever cross.
3. After drawing each curve, the mover must add a new dot along its path.
4. A player with no moves loses.

Figure 17 shows a game of Sprouts starting with two dots. Sprouts has been subjected to significant analysis, and for six or more dots involves reasonably complex strategy [9]. It is an example of a game that constructs a (planar) graph.

9 Transforming Games to Graphs

Defining the distance between vertices as the shortest path between those vertices – the distance is infinite if no path exists – is a mathematical metric that obeys the usual rules of distance in the plane or space. If a game or puzzle is described as a graph with edges representing possible moves, then it is easy to use various path finding algorithms such as Dijkstra’s algorithm, the A* algorithm, and other forms of dynamic programming [10] to assist game design. Neighbourhoods in the graph represent zones of control in the game, and it becomes possible to efficiently compute and enumerate the shortest and alternate paths to a goal.

Another issue is the construction of automated players, or the solutions of games, to resolve issues such as first player advantage. This requires more sophisticated algorithms to search the *game trees* that describe the space of possible moves. We will examine these in a future article.

Another interesting application is to represent puzzles as graphs in which vertices represent partial solutions (or the current state), and edges represent moves, such as filling in a digit in Sudoku or sliding a piece in Sokoban or Rush Hour. Solving a challenge step-by-step then corresponds to traversing the graph from vertex to vertex. The graph captures, in a concise form, the solution space of the puzzle. This opens the door to systematically and automatically answering questions such as: *does this puzzle have a unique solution?* or: *what is the smallest number of moves to solve this challenge?*

10 Conclusion

An understanding of basic graph theory permits game designers, or anyone interested in the algorithmic solution of puzzles, to abstract the essential features of boards, solution spaces, or even rule sets. The abstraction gives a different, more concise, view of some aspect of a game. It may not be a better for addressing a particular design issue, but is an additional tool for the designer.

This article introduces both graph-theoretic terminology and examples of games that incorporate graphs in a fundamental way. It touches on concepts of symmetry, algorithmic graph analysis, graph drawing, and planar duality. Future work will describe graph-based algorithms for solving and designing puzzles, with examples.
The game of Graph Domination has proven to be a useful teaching tool over the years, both in graph theory classes and in educational outreach activities. Like games and puzzles, graph theory is a natural path to reducing students’ anxiety about mathematics and computer science.

Acknowledgements

The author thanks the Canadian Natural Sciences and Engineering Research Council and the University of Guelph for their support of this work.

References


Daniel Ashlock is a Professor of Mathematics at the University of Guelph in Guelph, Ontario, Canada. His research interests include applications of computational intelligence to both mathematical games and games played by people. Address: Department of Mathematics and Statistics, University of Guelph, Ontario, N1G 2W1, Canada. Email: dashlock@uoguelph.ca

Shakashaka Challenge #5

Half-colour empty cells with triangles, as per the rules on p. 13. Challenge by ‘-4’ © Nikoli.
Automated Playtesting with Recycled CardSTOCK

Connor Bell, Hendrix College

Mark Goadrich, Hendrix College

Game designers benefit greatly from playtests of their prototypes in development. However, finding experienced, independent playtesters for repeated plays is often difficult. This paper describes RECYCLE, a card game description language, and CARDSTOCK, an implementation of RECYCLE which automatically playtests card games with both random and intelligent players. Our system can assist game designers by providing insight into the average player decision branching factor, turn order advantage, game length, and the potential for strategy. We demonstrate its use by playtesting variants of the games Agram, Pairs and War.

1 Introduction

When game designers seek to create fun and engaging games, they strive to avoid common design pitfalls. First, players can struggle from 'analysis paralysis' in a game with too many viable choices [1][2]. Alternatively, a game with too few options leaves little room for player agency and control over their fate [3][4]. A game must find a balance between the opportunity for strategy and the size of the branching factor of options available to the players.

Second, it is easy to inadvertently create a game that is unbalanced, giving certain players an inherent advantage in the game simply based on their turn order. A game should strive to provide a fair experience for all players [5]. Finally, an end condition of a game could be created that is either unachievable or only winnable through extended play. Modern board and card games tend to limit playing time to a reasonable length of less than an hour [1].

However, the rules of a game combined with the choices and motivations of the players can be categorised as a complex adaptive system [6]. This means that emergent properties such as fairness and game length cannot be readily understood from examining the rules in isolation, but are only discoverable in the moment of play.

In pursuit of these virtues of meaningful choice, fairness, and appropriate length, game prototypes typically undergo multiple iterations and modifications while being developed. Depending on feedback from early playtests, a game designer could seek to improve the play experience by changing some elements of the game, for example, by tweaking the point values, altering the mechanics, restructuring the winning conditions, or modifying the components [7]. Each of these changes constitutes a new game variant that must be subsequently playtested. This creates a cycle that can quickly exhaust the time and patience of the available playtesting volunteers.

We introduce RECYCLE, a card game description language, and CARDSTOCK, an implementation of RECYCLE with which designers can automatically playtest card games. After encoding the rules of a game in RECYCLE, designers can perform multiple Monte Carlo simulations of a game to explore the emergent properties and learn the shape of its play space. By separating the game description from the game engine, designers can focus exclusively on modifying the rules and mechanics, and then quickly test for unexpected side-effects using both random and semi-intelligent automated players.

We focus our work on card games because they provide a small yet interesting space of games for automated playtesting tools. Many card games incorporate randomness, via the ability to shuffle and deal from a fixed set of cards without replacement, and hidden information, either through playing cards face down or through player ownership of sets of cards [8].

We first examine related work in the development of card game description languages, followed by a detailed explanation of the elements of RECYCLE and a sample encoded game. We then briefly discuss the CARDSTOCK implementation and the options for automated players. We demonstrate the utility of RECYCLE and CARDSTOCK by playtesting variants of three card games to discover their strengths, weaknesses and potential for strategy. Finally, we discuss RECYCLE’s limitations and conclude with thoughts on future work.

2 Related Work

they affect the position; and the terminating and winning criteria’. When creating a description language for the elements and mechanisms of card games, one must therefore include the deck, card locations, suit-following rules, and point values of cards.

Font et al. [10, 11] described a context-free grammar $G_{\text{cardgame}}$ that serves as a card game description language [12]. They demonstrated its use with implementations of Texas Hold’em Poker, and reduced versions of Uno and Blackjack. While much of their work was dedicated to exploring evolutionary mutations of these games, their core language showed the feasibility of such a system, and contained support for card locations, tokens for tracking points, clearly defined player turns, and conditions which must be met to advance game flow.

However, Font et al. made a number of simplifying assumptions in $G_{\text{cardgame}}$:

- All games must use the standard French 52-card deck.
- Each player has only one hand of cards.
- The game table can hold multiple locations for cards, but they all face up, except for the source deck, which faces down.
- It is awkward to move cards from one location to another, relying on intermediary moves to accomplish passing cards to another player.
- Card precedence is fixed at the beginning of the game and cannot be altered based on game state.
- Player turn order is also fixed at the beginning of the game, and while players can pass or drop out, they cannot alter their turn order.
- Each game is a sequence of stages, allowing no way to repeat earlier stages.
- There is no clear separation between player actions and control flow, such that players, and not an external arbiter, determine when a given stage of a game is complete.

These assumptions hamper the broad applicability of $G_{\text{cardgame}}$, eliminating many mechanisms and genres from their game space, especially trick-taking games. The following section discusses our grammar RECYCLE and how we overcome the above limitations.

3 RECYCLE

RECYCLE is a game description language that allows for an algorithmic representation of common core mechanisms and elements of card games. RECYCLE stands for REcursive CYclic Card game LanguagE, referring to the primary feature of the language: the recursion of game stages containing cycles of player turns. The language resembles the LISP programming language [13], often having a large number of nested instructions that control the flow of the games.

The process of writing game rules in a natural language can be fraught with ambiguities, often necessitating clarifications after publication. Encoding a game in RECYCLE can be useful for illuminating the underlying formal structure of a game design, providing insight into avenues for targeted or large-scale refinement, and resolving potential ambiguities.

RECYCLE contains several features common to most programming languages, such as Boolean operators (and, or, !=, ==, etc.), integer manipulation (basic arithmetic), and comments for programmers (using ; ). Blocks of RECYCLE code indicate data elements, control flow, or conditional execution. RECYCLE also includes reserved words which correspond closely to the semantics of card games.

Other features that are more unique to RECYCLE, such as moving cards, keeping score on a scratch pad, cycling through the players, and assigning value to cards, are explained in the following sections. We use where possible Agram as a running example.

Agram is a simple Nigerian trick-taking card game for 2 to 6 players. Players are dealt six cards from a reduced French deck, and play six tricks. To win a trick, players must follow the suit of the lead player with a higher card; there is no trump suit. The object of the game is to win the last trick.

3.1 Cards

At the heart of every card game, there are cards and players. Cards in RECYCLE are represented as key/value trees, where certain attributes can be represented as children of others. For example, in the standard French deck of 52 cards, the suit is indicative of the colour, such that Hearts are always red, Spades are always black, etc. The card ‘7 of diamonds’ might look like the following tree:

https://www.pagat.com/last/agram.html
This makes the card more decomposable than just an association of keys and values. The deck block creates a set of cards, one for every possible combination of attribute values. A call to deck which constructs the standard deck of 52 cards looks like the following:

```
(deck
  (RANK (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K))
  (COLOR (RED (SUITE (HEARTS, DIAMONDS))))
  (BLACK (SUITE (CLUBS, SPADES))))
```

In this example, RECYCLE creates combinations of all ranks with all suits, and the colours come along with their respective suits. To create multiples of a type of card, an integer parameter is added before the attributes to indicate how many copies are desired.

The only reserved word in the snippet above is deck; all of the keys and values in the tree are user defined, requiring no other declaration or context for the terms. In RECYCLE, we use the convention that reserved words are lowercase, and user-defined elements are capitalised.

### 3.2 Card Locations

Cards are often given meaning in a game based on their organisation into physical locations. Locations for cards are associated with either the game or with a player. They can be invisible or visible locations, denoted as iloc and vloc in RECYCLE. When the players in the system request the game state, all cards in locations which are vlocs, and those ilocs associated with that player, will be completely known. All cards in other ilocs, belonging to either the game or other players, will be hidden and therefore unknown.

Cards can be instantiated in a location using the deck command as follows:

```
(create deck (game iloc STOCK)
  (deck ...))
```

Cards can be retrieved from a card location using either an integer index, or the keywords top and bottom. These cards can then be moved between locations, using the move command:

```
(move (top (game vloc DISCARD))
  (top (game iloc STOCK)))
```

This command moves the top card from DISCARD to become the top card in STOCK, both table locations as indicated by the keyword game. To move multiple cards at once, the two locations can be followed by an integer n, or the keyword all to continue moving cards until the first location is empty.

The command size takes a location and returns an integer with the value of the number of cards in that location. For example to find the size of the DISCARD location, one would say:

```
(size (game vloc DISCARD))
```

#### 3.2.1 Memory vs Physical Locations

The card locations and actions discussed above conform to the idea that a physical card can only be in one physical location at any time. A card can also exist in any number of virtual memory locations, or mems, at a time, however, these 'locations' only signify game state, not the physical presence of the cards. For example, in many trick-taking games, the mem location LEAD is used to remember the card which led the trick. Once the first player chooses a card to play, the card is 'remembered' as the LEAD card.

```
(remember
  (top ((current player) vloc TRICK))
  (top (game mem LEAD)))
```

All memory locations are visible to all players, but are not used to record private player knowledge of game state history.

#### 3.2.2 ‘Where’ Clauses

Depending on the game state during execution of a RECYCLE program, a card location may contain an assortment of cards. Filtering the cards at a location is a necessary feature for enforcing the rules of many card games. Again, in many trick-taking games, it is important to ensure that the subsequent players follow the lead player’s chosen suit if possible. To this end, consider the following code:

```
(!=
  (size
    ((current player) iloc HAND
      where (all
        (== (cardatt SUITE each)
          (cardatt SUITE
            (top (game mem LEAD))))
      )) 0))
```

This checks if the current player’s hand contains at least one card for which the suit of the card matches the suit of the top card of the memory location LEAD.
3.3 Card Ownership

Another relationship implemented in RECYCLE is card ownership. Given a card, the function `owner` will return a reference to the player who owns the card, allowing indirect access to integer storage and other card locations for that player.

3.4 Integer Storage

Typically some type of storage is necessary to keep track of the underlying game information over time. This is often represented by tokens or chips in the game, or recordings on a scratch pad. RECYCLE offers integer storage collections for players, teams and game, as follows:

```r
((current player) sto TRICKSWON)
```

The functions `set`, `inc`, and `dec` respectively modify the value of an integer storage by setting to, incrementing by, or decrementing by a given value. The given value can be a literal integer, e.g. 5, the result of reading the value of another storage, or the result of evaluating an expression.

Figure 1 depicts the internal state of a game of Agram where the fourth trick is in progress. Players have a hand and a trick card location. The game remembers the value of the lead card. Each of these elements can be captured by the above blocks in RECYCLE.

3.5 Control Blocks

Three operations are used to control game flow: `choice` to present game action options to the player; `comp` to perform scripted computer moves; and `stage` to nest and aggregate other `choice`, `comp`, and `stage` blocks.
3.5.1 Choice

Choice blocks present the current player with a collection of actions. The keyword \texttt{any} can be used in place of a specific index when accessing a card from a location, allowing the representation of rules such as: \texttt{play any card from your hand to the table which has the suit Hearts}. The engine will enumerate all cards in the hand location and present all possible actions to the player, asking them to choose one:

\begin{verbatim}
  (move (any ((current player) iloc HAND
             where (all (== (cardatt SUIT each)
                           HEARTS))))))
\end{verbatim}

Actions often have preconditions, so to account for these, choice blocks have the following structure:

\begin{verbatim}
  (choice
    ((condition)
     ...actions...
    )
    ((condition)
     ...actions...
    )
  )
\end{verbatim}

This allows for intermixing of choices of which the player gets to pick one. If there are no conditions for an action, such that it should always be available, this can be represented with the empty condition \texttt{()}. Note that each condition is checked every time, regardless of the outcomes of the others. It is often helpful to make the conditions mutually exclusive, resulting in only one specific set of actions enumerating each round.

3.5.2 Computer Moves

Computer move blocks, or \texttt{comp}, indicate actions which should be performed once by the computer each time they are encountered. They typically correspond to the game actions performed by a designated dealer. In contrast to \texttt{choice} blocks, actions in \texttt{comp} blocks should not contain the keyword \texttt{any}. Common tasks in these blocks include simple actions such as shuffling the deck or dealing cards to each player, or complex actions such as evaluating the cards in a location to determine a player’s score or assigning a point to the player who is the owner of the card with the maximum score across selected players’ card locations.

3.6 Player and Team Cycles

One of the most common elements of card games is a cyclical ordering of turns. In fact, stages incorporate this behaviour by default, cycling through the players or teams each iteration. The control flow becomes more complicated when the ordering is interrupted, so \texttt{RECYCLE} offers several functions to temporarily interrupt the ordering. These calls start with \texttt{cycle} and are followed by two parameters: when the change should take place and who should go instead.

The first parameter can have one of two values: \texttt{current} or \texttt{next}. Using \texttt{current} results in an immediate change, while \texttt{next} applies the change when the stage loops. The second parameter can be an integer, \texttt{current}, \texttt{next}, or \texttt{prev}.

If no changes are applied during the iteration following a change, the stage will proceed normally. Consider the following examples from the naturally cycle the value of \texttt{(current player)} or \texttt{(current team)} depending on whether it is a \texttt{(stage player ...)} or \texttt{(stage team ...)}. Each stage has an end-condition, which is a Boolean expressions that can query some testable statement about the game state. As long as the end-condition is not met, the stage will run and then cycle to the next player or team before reevaluating the end-condition. Consider the following stage:

\begin{verbatim}
  (stage player
    (end (== ((all player)
             iloc HAND) 0))
    (comp ()
      (move
        (top ((current player)
             iloc HAND))
        (top (game vloc DISCARD))))
  )
\end{verbatim}

This stage will continue to execute until all players have moved the cards in their hands to a discard location. As long as the end condition is false, the code contained will run. When executing the stage, the value of \texttt{(current player)} changes each time, wrapping around from the final player back to the first. As this value changes, however, it does not affect the value of \texttt{(current player)} in surrounding contexts, and will return to its prior value when the stage exits.

Stages can also contain other stages. In Agram, an individual deal is over when all players’ hands are empty. Within a deal, each trick is complete when every player has exactly one card on the table in front of them. This relationship is represented as a nested stage, with the deal as the outer-most stage, and the trick on the inside.
game Uno. To reverse the order, in the event of a reverse card being played, the following cycle action is performed:

(cycle next prev)

If a skip card is played, the following cycle action will skip the next player by setting the current player to be the next player:

(cycle current next)

3.7 Scoring

RECYCLE offers two functions to handle win conditions: initialize points and scoring. Together, these calls represent card precedence and the declaration of winners.

3.7.1 Precedence

In card games, precedence is the idea of a card having more value than another. For example, in Agram, the lead suit will have more value than any other suit. Also, aces have more value than tens, which in turn outrank nines, etc. This can be translated into a mathematical function which, given the rank and suit of a card, awards that card a number of points. For a given trick, the following RECYCLE code defines such a function, depending on the suit of the LEAD card:

(initialize points PRECEDENCE
  (all (SUIT (cardatt SUIT (top (game mem LEAD)))) 100)
  (all (RANK (A)) 14)
  (all (RANK (10)) 10)
  (all (RANK (9)) 9)
  (all (RANK (8)) 8)
  (all (RANK (7)) 7)
  (all (RANK (6)) 6)
  (all (RANK (5)) 5)
  (all (RANK (4)) 4)
  (all (RANK (3)) 3)
)

In this example, the function can be referred to by its name, PRECEDENCE. It will return an integer for each card evaluated against the rules. For example, if Hearts was the LEAD suit, the 3 of Hearts would be awarded 100 points for having the suit Hearts and 3 points for its rank, for a total score of 103. These functions need not map to all possible values or be one-to-one. If the function is not one-to-one, it is possible that the max or min functions will not be able to find a unique card, resulting in undefined behaviour.

3.7.2 Winners

The last element of a game description in RECYCLE is always scoring. For convenience, scoring is implicitly a stage player, so references to (current player) within the call are valid and encouraged, as it would be atypical for a game to assign a score to a player without knowing which player was in question. The scoring block has two parameters: the first is either min or max, indicating whether the winner is the player with the lowest score or the highest, and the second is any block of RECYCLE which returns an integer and, as mentioned earlier, will likely include a reference to (current player).

3.8 Agram Encoding

To illustrate how the previous elements can be combined to create a complete game, Listing 1 shows a full description of Agram in RECYCLE.

In lines 2 through 10, the number of players are defined, the teams are defined as individuals, indicating no alliances, and the deck is instantiated to the STOCK location. Because the rules for Agram dictate that there is no Ace of Spades, two separate create deck calls were necessary.

In lines 12 through 14, the STOCK location, now containing all of the requisite cards, is shuffled, and 6 cards are dealt to each player.

Lines 20 through 25 handle the case in which at least the first card of the trick has been played and the player is unable to follow suit, and are therefore allowed to play any card from their HAND to their TRICK location.

Lines 26 through 33 handle the case in which at least the first card of the trick has been played and the player can (and therefore must) play a card which follows suit.

Lines 34 through 38 give the first player the freedom to play any card, and subsequently remember that card to the LEAD location, ensuring that the following players will be forced into one of the two cases above.

The comp in lines 39 through 53 determines the winner of each trick using the scoring function defined in lines 41 through 46, clears the LEAD memory location, sets the next leader to be the player who holds the winning card, and clears the cards from the last trick. In the event all of the players’ HAND locations are empty, indicating the game is over, it awards 1 point to the player who won the most recent trick.

Line 54 declares that the winner of the game is the player with the highest value in their SCORE storage bin. From the prior rules, we know that only one player will receive a point each game, making this a decidable and unique maximum value and owner.
01 (game
02 (setup
03 (create players 4)
04 (create teams (0) (1) (2) (3))
05 (create deck (game iloc STOCK) (deck (RANK (3, 4, 5, 6, 7, 8, 9, 10))
06 (COLOR (RED (SUITE (HEARTS, DIAMONDS))))
07 (BLACK (SUITE (CLUBS, SPADES))))
08 (create deck (game iloc STOCK) (deck (rank (A))
09 (COLOR (RED (SUITE (HEARTS, DIAMONDS))))
10 (BLACK (SUITE (CLUBS))))
11 (comp ()
12 (shuffle (game iloc STOCK))
13 (move (top (game iloc STOCK))
14 (top ((all player) iloc HAND)) 6))
15 (stage player
16 (end (== (size ((all player) iloc HAND)) 0))
17 (stage player
18 (end (> (size ((all player) vloc TRICK)) 0))
19 (choice
20 ((and (== (size (game mem LEAD)) 1)
21 (== (size ((current player) iloc HAND where
22 (all (== (card att SUITE each)
23 (card att SUITE (top (game mem LEAD)))))) 0))
24 (move (any ((current player) iloc HAND)
25 (top ((current player) vloc TRICK))))
26 ((and (== (size (game mem LEAD)) 1)
27 (!= (size ((current player) iloc HAND where
28 (all (== (card att SUITE each)
29 (card att SUITE (top (game mem LEAD)))))) 0))
30 (move (any ((current player) iloc HAND)
31 (top ((current player) vloc TRICK))))
32 ((== (size (game mem LEAD)) 0)
33 (move (any ((current player) iloc HAND)
34 (top ((current player) vloc TRICK))))
35 (remember (top ((current player) vloc TRICK))
36 (top (game mem LEAD)))))
37 (comp
38 ()
39 (initialize points PRECEDENCE (3)
40 (all (SUITE (card att SUITE (top (game mem LEAD)))) 100)
41 (all (RANK (A)) 14) (all (RANK (10)) 10)
42 (all (RANK (9)) 9) (all (RANK (8)) 8) (all (RANK (7)) 7)
43 (all (RANK (6)) 6) (all (RANK (5)) 5) (all (RANK (4)) 4)
44 (all (RANK (3)) 3))
45 (forget (top (game mem LEAD)))
46 (cycle next (owner (max (union ((all player) vloc TRICK))
47 using PRECEDENCE))
48 (move (top ((all player) vloc TRICK))
49 (top (game vloc DISCARD))))
50 (== (size ((all player) iloc HAND)) 0)
51 (inc ((next player) sto SCORE) 1))
52 (scoring max (((current player) player) sto SCORE))
53 )
54
Listing 1. Rules for Agram in RECYCLE.
4 CARDSTOCK

CARDSTOCK is a run-time and analytics engine for RECYCLE games. It is written in C# with Mono and Xamarin\(^2\) for cross-platform support, while relying on ANTLR\(^3\), a parsing library, to generate the parse tree. The source code for CARDSTOCK is available from the online code repository site GitHub\(^4\).

In order to make our language as locally interpretable as possible, we perform run-time existence checks on all card locations and integer storage names. The first time a location is referenced, it is instantiated as an empty location, and the first time a storage is referenced, it is instantiated with the value 0.

CARDSTOCK runs simulations for a given game by iterating over the nodes in the parse tree. This iteration allows the game to be interrupted every time a choice block is encountered. This is a powerful stopping point for two reasons: the state, as well as the iterator, can be cloned with either perfect or imperfect knowledge to allowing hypothetical play-throughs given the current state, and the state can be tracked for the purpose of detecting game state repetition.

To perform experiments in CARDSTOCK, a user specifies the game file to be loaded, the number of simulations to be run, and the types of players involved in each game. When run in debug mode, a majority of the actions performed in the game, as well as some useful state information, are reported to the user. When run in release mode, only the resulting scores from each game are displayed, drastically speeding up the execution time.

When the simulations have been completed, the win percentages for each player are displayed, along with the number of cycles that were detected and the average number of decisions made per game, which can be used to estimate game time. Additionally, the number of available choices at every decision, or the branching factor, is also logged throughout.

5 Players

CARDSTOCK’s default players choose an action to take from a given list of possible decisions using a discrete uniform probability distribution. Simulating a game with these random players can give designers a general insight as to whether the game is behaving as expected, somewhat analogous to fuzz testing in software development\(^1\).

This testing can also detect if the end conditions are always met or if instead the game can become stuck in an infinite loop. But if the game requires competent strategy to proceed and terminate, then random players may not provide the best information about these types of games.

To automatically incorporate intelligent players, we implemented PIPMC players, a combination of Perfect Information Monte Carlo and Pure Monte Carlo players\(^15, 16\). These players can clone the game state each time they are asked to make a decision, with the caveat that they randomly assign possible cards to locations that are invisible to the player. This ensures that the player’s search for good moves relies on the information visible to them as well as a large number of random assignments of the other cards, without explicitly showing them information that human players would not have.

PIPMC players are not guaranteed to make optimal decisions through this averaging over clairvoyance approach. Within cloned game states, the card assignment will appear fixed, thus the players could fail to choose moves that would add or hide information about the game state. However, our goal is to only indicate the existence of potential for strategy\(^17\).

For each potential action in their decision list, our PIPMC players clone twenty hypothetical games, simulate a completed game for each clone consisting of only random players, and record the outcome. PIPMC will then choose the action that earned the best average score, accounting for the minimisation or maximisation nature of the game in play.

6 Results

To date, we have encoded the following games in RECYCLE: Agram, GOPS, Hearts, Lost Cities, Pairs, Spades, Uno, War and Whist\(^7\). Here, we highlight three of these games – Agram, Pairs and War – using CARDSTOCK to explore the game design properties of turn order advantage, game length, average branching factor, and the potential for strategy.

The following experiments emulate the decision process that could have been used to discern the given game design from among competing variants. For each game and variant, we conducted 10 epochs of 100 simulations, for a total of 1,000 simulations per variant. The following results show the pooled means and standard deviations of these experiments.

\(^1\)https://www.xamarin.com/
\(^2\)http://www.antlr.org/
\(^3\)https://github.com/mgoadric/cardstock
\(^4\)https://www.pagat.com/whist/whist.html
6.1 Agram

As described earlier, Agram is a small trick-taking game. With only six cards in a player’s hand and six tricks to be won, we asked: Is there a balance between player choice and the potential for strategy?

Figure 2 shows the average player decision branching factor with default random players. The lead player in each trick is shown separately from the three aggregated results of the three later players. We can see the effect of being forced to follow suit when possible. The lead player can always play whatever card they desire, but following players are then limited to approximately 2.5 card choices on average for the first three tricks and tapering off thereafter. There is a definite advantage to being the lead player in terms of player choice.

![Figure 2. Branching factor for lead versus following players in Agram.](image)

To investigate the potential for strategy in Agram, we ran simulations for two through five players, using one PIPMC with the remaining players random. We report in Figure 3 the win percentage for the PIPMC player in comparison to the expected probability of winning for a random player, given the assumption that the game is balanced.

![Figure 3. Gain of PIPMC player over expected value of random player in Agram.](image)

PIPMC players are able to control their fate, outperforming the expected value by approximately 20 percentage points across all player sizes. However, there is still enough randomness in the game to confound their ability to win.

Given the above, we next investigated: What is the smallest number of tricks that still allows for a fair game? We can easily explore variants of Agram by changing one number in the RECYCLE description. Our first set of variants altered the number of cards dealt to each player from one to six, while fixing the number of players at two. Figure 4 shows the results for each hand size using random players.

![Figure 4. Fairness of turn order in two-player Agram when limiting number of cards dealt.](image)

We can see that with four, five or six cards, the game appears fair, however, a clear bias for the first player emerges as the number of cards is reduced. We believe that because the lead suit becomes the highest precedence, it is very unlikely in such limited games that the following player is able to follow suit and thus is doomed to failure.

These results track with the known variants of Agram. In particular, the version in which players are dealt only five cards instead of six is known as Sink-Sink. There are no established variants of smaller size, perhaps due to a human player’s refusal to repeatedly play a game that is unfair.

![Figure 5. Fairness of turn order in Agram when varying which trick will decide the winner.](image)

Our second set of variants retains the deal of six cards, but changes the number of tricks played

---

6Rules for Sink-Sink can be found on the Pagat page for Agram: https://www.pagat.com/last/agram.html
before determining the winner. As above, we fix
the number of players at two and run simulations
with random players. In Figure 5, we see that
most variants are relatively balanced games, ex-
cept for when the game is decided after playing
only one trick.

6.2 Pairs

Pairs is a press-your-luck card game for two to
six players, described by designers James Ernest
and Paul Peterson as 'a new classic pub game'.

**Pairs** uses a custom deck of 55 cards, contain-
ing one card of value 1, two cards of value 2,
etc., up to ten cards of value 10. Each round
players are dealt one card to a face-up hand,
and then players cycle in turn order to either
end the round by scoring the minimum value
card in play or draw another card into their
hand. If the drawn card is the same value as a
card currently in their hand, the player scores
that many points and the round is over. The
first player to score a set number of points over
multiple rounds is the loser, so players strive to
minimise their points.

First, we examined the question: how can turn
order be manipulated to create a fair game? We fo-
cussed on games with four random players. In
the four-person version, the game is over when
one player earns at least 16 points. Initially, we
found that all players have an expected value of
slightly over 9 points. This points to a fair game,
however, we found evidence of turn order bias
within an individual round.

![Figure 6. Unfairness of turn order in Pairs.](image)

In these experiments, players were not given
the option of stopping early, to isolate the effect
of drawing another card. Figure 6 shows the
expected value for each player in one individu-
al round with two different rules to determine
the first player. The rules of Pairs designate
the player with the minimum valued card as the first
player. This led to an unbalanced game, in which
the second player has the worst expected value,
as seen by the blue (left) bars.

However, we can see the benefit of this ap-
proach in comparison to a variant in which the
first player is chosen randomly. For this variant,
there is a strong disadvantage in being the first
player, with an expected value of 2.75 points, as
opposed to being the fourth player, with an ex-
pected value of 1.37. While both are unbalanced,
the published rules adhere more closely to a uni-
form chance of winning for all players.

The designers of Pairs propose a continuous
variant, where the round is not over when a
player bows out or draws a pair. Instead, only the
current player's hand is discarded, and on their
next turn, they must take a card from the stock.
This variant can be captured in RECYCLE with a
single stage and much simpler structure. In sim-
ulated games with four players, we found this
one-round variant to be well balanced. We also
noted another property of the continuous variant;
the expected value for each player rose from 9
points to 11.5 points. This variant could therefore
have more tension, as all players are closer to the
threshold for losing the game.

![Figure 7. Game length as a function of number of
players in Pairs.](image)

Second, we investigated: what methods can be
used to keep game length constant while varying the
number of players?. The advertised game length for
Pairs is 15 minutes. The rules scale the number of
points necessary to lose a game by the number of
players, using the formula: \((60/\text{numPlayers}) + 1\).
To explore how this simple rule affects the length
of the game, we simulated games with only ran-
dom players for two through five players. We
forego estimating the clock time for each deci-
sion, and instead report in Figure 7 the average
number of calls to a choice block for a player
decision.

We found that there is a very consistent cor-
respondence between the number of players and
the length of the game, demonstrating that the
scaling is having the desired effect. A different

---

7https://boardgamegeek.com/boardgame/152237/pairs
rule could be implemented to provide perfect consistency; however, the loss of simplicity would not be worth the complication.

Finally, we asked: with only two choices per turn, does Pairs retain the potential for strategy? We ran simulations for 2 through 5 players, using one PIPMC and leaving the remaining players random. Since the goal in Pairs is to not lose as opposed to win, we show in Figure 8 the non-loss percentage for the PIPMC player in comparison to the expected probability of not losing for a random player, given the assumption that the game is fair. We can see the PIPMC is drastically better than the uniform random players, quickly approaching a non-loss probability of 1.

![Figure 8. Gain of PIPMC player over expected value of random player in Pairs.](image)

6.3 War

Finally, we examined War, a two-player simultaneous trick-taking game.\(^8\)

In War, half of a shuffled standard deck is dealt to each player. Each round, players play the top card from their hand into the trick. The winner of the trick adds the won cards to the bottom of their hand. The goal of the game is to collect all the cards in the deck.

There is no potential for strategy in War, due to the player having only one option on each turn. However, we include this game in our analysis because RECYCLE can also detect when a game has the potential to enter an infinite cycle. In particular, we confirm through Monte Carlo experiments the results of Lakshtanov and Roshchina that certain implementations of the rules can lead to endless loops.\(^9\)

In our first implementation, we placed the cards at the bottom of both players decks in a deterministic order: the first player’s card followed by the second player’s card. This enabled cycles to occur in the game state. We found that 16.8% of simulated games were cyclical. After revising our RECYCLE code for War to shuffle the cards before moving them to the bottom of the deck, no cycles occurred.

6.4 Performance

As noted above, for all of our experiments, we ran each configuration 1,000 times. While execution times may vary depending on the complexity and branching factor of the game and the hardware used, the results shown in Table 1 are indicative of the times taken to conduct 1,000 simulated games of Pairs and Agram. PIPMC players can be orders of magnitude slower than random players.

<table>
<thead>
<tr>
<th>Game</th>
<th>Random</th>
<th>PIPMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairs</td>
<td>12s</td>
<td>14m</td>
</tr>
<tr>
<td>Agram</td>
<td>25s</td>
<td>75m</td>
</tr>
</tbody>
</table>

Table 1. Timings for 1,000 simulated games.

The run-time for random players reflects the average game length, but the PIPMC execution time is influenced by both game length and branching factor. The random runs for Agram were \(\approx 2\) times longer than for Pairs, but, because Agram has a higher branching factor than Pairs, the PIPMC runs for Agram were \(\approx 5\) times longer than for Pairs.

7 Conclusion

We developed a new language, RECYCLE, for describing card game states and mechanisms, and built a system, CARDSTOCK, to run automated tests on encoded games. Upon encoding games in RECYCLE, one can readily find elements to modify and potentially improve the game. Using this system, we evaluated various properties of card game designs to understand the effects of subtle rule changes within Agram, Pairs and War.

7.1 Limitations

While a wide variety of card games can be encoded in RECYCLE, there currently remain limitations to be addressed.

- Games cannot reference actual physical relationships between card locations. For example, in the Golf variant Crazy Nines, players arrange cards in a \(3 \times 3\) grid and earn points based on patterns in rows, columns, diagonals, or blocks.

8https://www.pagat.com/war/war.html
9https://www.pagat.com/draw/golf.html
- All game actions must be sequential. While RECYCLE can emulate simultaneous moves such as blind-bidding through some sleight-of-hand with `comp` blocks, it cannot capture the concept of an out-of-turn-order interrupt found in many ‘take-that’ card games, such as Munchkin.\(^{10}\)

- The visibility rules apply only to card locations, not cards themselves. Individual cards in a location cannot be exposed to other players. Also, a player can always see their own locations without regard to visibility, which prevents us from encoding games such as Hanabi,\(^{11}\) in which players cannot see their own hand.

### 7.2 Future Work

The approach taken above can be applied to many other card games. We first plan to compare Hearts, Spades and Whist, along with other members of the trick-taking family to understand their similarities and differences. Also, once a large enough library of games are encoded in RECYCLE, it will be possible to attempt to apply a distance metric, to cluster games based on their mechanisms and perhaps find new families and relationships.

Currently, our intelligent players are a baseline for detecting if player choices in a game matter. We plan to incorporate more powerful strategies such as Information Set Monte Carlo Tree Search \(^{19}\), with multiple players of various strengths playing against each other, in an attempt to determine if a game is suitable for novice, intermediate, and expert players. We also hope to significantly decrease the run-time of our code so that we can efficiently examine larger games.

In a similar vein, showing that intelligent players can defeat random players in a game is not necessarily sufficient to support a claim of high game quality. It is also desirable to have multiple paths to victory, so that players cannot exploit one particular overly strong strategy. This could be accomplished by artificially limiting the choices of players to a subset of options and examining whether distinct strategies emerge.

We have begun implementing a web interface to CARDSTOCK, that will eventually allow designers to graphically encode games in RECYCLE, submit their designs for execution by our experimental setup to generate multiple runs, and then explore display graphs and statistics based on these runs.

Finally, we plan to develop a system to automatically generate games using the RECYCLE language. We believe that the properties described above, and others, can be quantified more precisely, such as divergence from fair turn order and percentage gain of PIPMC players, in order to provide clear metrics of game quality \(^{20}\). Using these statistics, we hope that an evolutionary search through the space of games possible in RECYCLE could result in the creation of novel mechanisms and interesting game designs \(^{21}\).

### Acknowledgements

Thanks to Brent Yorgey, Ian Shrum, and Mario Muscedere for excellent lunch-time card game discussions, and to Gabe Ferrer, Chris Camfield, John McAvey, Collin Shaddox, Laura Goadrich, and the anonymous reviewers for their feedback on this article.

### References

http://www.thegamesjournal.com/articles/Essence.shtml


Connor Bell is an undergraduate computer science major and Software Engineer: Tools and Infrastructure at Google. Address: 1600 Washington Ave, Hendrix College, Conway, AR, USA. Email: bellca@hendrix.edu

Mark Goadrich is an Associate Professor of computer science at Hendrix College and designer of the card games Nanuk and Gene Pool. Address: 1600 Washington Ave, Hendrix College, Conway, AR, USA. Email: goadrich@hendrix.edu

Shakashaka Challenge #6

Half-colour empty cells with triangles, as per the rules on p. 13. Challenge by Guten © Nikoli.
Robert Abbott, Logic Mazes

This article presents Robert Abbott’s musings on the concept of clarity, as they occurred to him while inventing his game Epaminondas. It was first published in Games & Puzzles magazine in 1975 [1], then reposted by Abbott in 2006 on his own web site with an important addendum [2].

1 Introduction

Letting a game inventor write about his own game is not quite the same as letting an author review his own book. Self-review may not be respectable in an established art form like the novel, but it does have a long tradition in art forms struggling for acceptance. The most recent examples are the underground film-makers of the sixties who reviewed their own works, and the ‘happenings’ and other art events that were usually accompanied by polemics from the creators explaining what they were doing. Even though games are as old as any art form (in fact, it makes more sense to say that art is a game form than that games are an art form), games are not generally accepted as worthy of critical study (except maybe as simulations or mathematical models). Thus it is fortunate that game inventors have the pages of Games & Puzzles to explain why their games are good and why they are important.

Epaminondas is my latest game [3], and it has just been published by Philmar Ltd. [4] A brief description of the rules is given below. I will not actually review the game (I will even forgo explaining why it is called Epaminondas), except I will say it has great clarity. I wanted to devote the major portion of this article to a discussion of clarity, a concept I came to understand while working on Epaminondas.

Clarity is essentially the ease with which a player can see what is going on in a game. It is a useful idea for a game inventor to keep in mind during the development of a game, and it is useful in the criticism of games. Most important, it explains what makes a game deep.

A lot has been written about the ‘depth’ of games like Chess and Go without anyone really explaining what depth is. Most people assume that depth can be explained purely in terms of logic or game theory. This is not true. If you look at games only in terms of the size of their strategy trees, it turns out that any perfect-information, non-chance game is complex enough to be beyond complete human understanding – thus in this sense all these games have equal depth.

Epaminondas is played on a $12 \times 14$ square grid, initially set up as shown. [These rules are condensed from Abbott’s original description.]

![Figure 1. Epaminondas starting position.](image)

Two players take turns, moving either one of their pieces a space in any direction or a line of their adjacent pieces (called a phalanx). A phalanx is moved as a unit, any number of spaces up to a maximum equal to the number of pieces moved, along the direction of its line.

Captures occur when a phalanx encounters an enemy phalanx or piece. The phalanx stops at the first enemy piece encountered, and captures that piece and all adjacent enemy pieces behind it, as long as the phalanx thus captured is smaller than the capturing phalanx.

The object of the game is to advance pieces across the board. When a player reaches the farthest row, the opponent must immediately capture that piece (or another on that row) or move one of their own pieces onto their own farthest row. If the opponent cannot do this, the player has won.

I am excluding here any games that have a known perfect strategy or games that are over in a few moves. What I mean is, if you go far enough down the strategy tree (say about ten moves, which is normally farther than a human can see)

---

1 And more recently re-published by Nestorgames: http://www.logicmazes.com/games/epam.html

then all games have enough branches so a human cannot understand them all. Certain games, like Chess, have more choices (branches) from each board position than do other games, like Wari. But you just have to go farther down the tree in Wari and you will find enough branches to boggle the mind.

2 Impossible to Think Ahead

I recently played Stay Alive, a game made out of a lattice of slats, each slat having a pattern of holes. The game involved moving the slats to allow marbles to fall through.

This was a simple, fun game where no one tried to think ahead at all. In fact, it was impossible to think ahead since it would have been very difficult to figure out what the next alignment of holes would be. But the pattern of holes in the slats was known; so it would theoretically be possible to work out all the board positions that would follow from any position. Also, since chance was not involved, Stay Alive must be considered a perfect-information, non-chance game. If you work your way far enough down its strategy tree you could come up with as many choice points as are considered in a game of Chess. Yet no one could consider Stay Alive to be as deep as Chess.

3 The Trouble with Ultima

As an example of a game utterly lacking in clarity there is Ultima, unfortunately one of my earlier inventions (although it is a board game, it is described in my book Abbott’s New Card Games [17]). In Ultima, each piece uses a different form of capture – leaping, intercepting, withdrawing, etc. This is an interesting idea, but the resulting game is so complex that it is difficult to see more than one or two moves ahead, and too many pieces are captured simply by surprise attack.

For those readers who know this game (and there are people who still insist on playing it) I can point out that the Withdrawer and Immobilizer are fairly clear pieces, the first because you can easily see what it is attacking and the second because it freezes a portion of the board and creates a focus for the action. It is the other pieces, especially the Pawns and the Coordinator, that cause the confusion. One of these years I hope to revise this game by replacing the confusing pieces. It is interesting to note that my original idea for Epaminondas came during a moment of frustration while working on Ultima.

4 L-Game ‘Elegantly Minimal’

One final point: clarity has nothing to do with simplicity, or even with elegance. I will give two examples.

Edward de Bono’s L-Game [6] is elegantly minimal – it uses only four pieces and is played on a board of only $4 \times 4$ squares. It is not, however, clear. I find it very hard to picture what the board will look like when I turn my ‘L’ over, I find it harder still to visualise my opponent’s responses, and it is impossible for me to look ahead to my next move. I fear I will have to read de Bono’s The 5-Day Course in Thinking [6] before I will be able to see far ahead in the L-Game.

The second example is Wari or any of the Mancala games. These games do not appear to be simple. Reading their rules will give you the impression that they would be very difficult to play. How can you possibly think more than one move ahead when you have to count the beans in each of your cups then count around the board to see where the last bean from each cup would be dropped? Yet when you play one of these games, you find it has a surprising clarity – you easily remember where the last bean from each cup will go and can see how these points would change as more beans are added to the various cups.

---

2https://www.boardgamegeek.com/boardgame/2770/stay-alive
3We retain the author’s original (US) spelling for this piece’s name.
5 Conclusion

Here is a summary of the points I wanted to make. A game can be simple yet lack clarity, and conversely a game can be complicated but still clear. Playing a game soon reveals its degree of clarity. The greater the clarity of a game, the farther you can see into it, and therefore the greater its depth for you.

Acknowledgements

This article was first published in 1975 in *Games & Puzzles* magazine [1], then re-posted in 2006 on the author’s Logic Games web site with addendum [2]. Thanks to the author’s family for permission to reprint it here. The piece is printed verbatim except for updated references and condensed rules for Epaminondas.

References


Addendum

This article turned out to be quite influential; though, unfortunately, most people misunderstood it. Here is what they thought I was saying: The depth of a game depends on the size of the strategy tree. That was exactly the idea I was arguing against. But even those who misunderstood the article thought it was good. However, one game inventor, Christian Freeling, wrote that the article was tautological. Yeah, well of course it is if you do not understand it.

Maybe here is a better way to express my idea: *The apparent depth of a game does not depend on how far you can travel down the strategy tree of the game. It instead depends on how far you can see down the strategy tree. And how far you can see depends on the clarity of the game.*

I discovered that the idea of clarity can also be applied to the logic mazes I have been working with. A maze should of course be confusing, but the rules for travelling though the maze should not be confusing. If the rules are confusing, then it is hard to see the consequences of a move, you cannot see more than one move ahead, and you cannot really plan ahead.

Robert Abbott is an American computer programmer and the designer of many well-known games and puzzles, including Ultima, Confusion, Eleusis, Epaminondas, and Theseus and the Minotaur. Address: New York, USA. Email: abbott@logicmazes.com

Shakashaka Challenge #7

Half-colour empty cells with triangles, as per the rules on p. 13. Challenge by Sample Rocket © Nikoli.
Vol. 2, no. 2 2016

Editorial
Nothing New Under the Sun. ......................................................... C. Browne 3

Short Note
Diamond: Improving on a Known Design ........................................ L. Back 5

Case Study
From Computer Operations to Mechanical Puzzles ............... M. O. van Deventer and I. Kriz 9

Case Study
Physics Laws as Game Rules ..................................................... F. M. Marqués 17

Game Design Patterns
Make the Design do the Work ..................................................... C. Browne 27

Case Study
How to Make a Better $3 \times 3 \times 3 \times 3$ ......................................................... C. Hoff 41

Article
Mitigating Non-Strategic Coalitions .............................................. C. Duncan 55

Short Note
The Development of a Tangram Family ........................................ F. Horn 62

Maths in Games
Graph-Based Search for Game Design ..................................... D. Ashlock and C. McGuinness 68

Short Note
A Game Design Approach to a Real-World Problem ...................... C. Browne 76

From the Archive
What’s a Ludeme? ................................................................. D. Parlett 81

The front cover shows a rendering of Carl Hoff’s Sphere Hypercube puzzle design.

The design and development of the Sphere Hypercube are described on pages 41–54.

All content © 2016, Game & Puzzle Design and QUT.
Credits

Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com

Editor-in-Chief

CAMERON BROWNE
Department of Knowledge Engineering (DKE)
Maastricht University
Maastricht, 6200 MD
Netherlands

cameron.browne@maastrichtuniversity.nl

Liaison Officer

NÉSTOR ROMERAL ANDRÉS
Nestorgames
Spain

Copy Editor

RUSS WILLIAMS
Green Star
Poland

Editorial Panel

INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
CARL HOFF, Applied Materials, USA
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
DAVID PARLETT, Parlett Games, UK
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
SPYRIDON SAMOTHRAKIS, Univ. Essex, UK
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINANDS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent publication prepared and published with the kind help of volunteers. Set in Palatino using \LaTeX.

Published content © 2018, Game & Puzzle Design.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.gapdjournal.com/submit
Nothing New Under the Sun

Cameron Browne, Queensland University of Technology (QUT)

TRULY original design ideas that transform the gaming landscape are rare. Almost all new games are created by combining ideas from previous games in novel and interesting ways, in a process of combinatorial creativity [10].

The good news is that this relatively small pool of known ideas can be combined in a seemingly endless manner, to create the many fascinating games that continue to emerge. The bad news is that most new games necessarily contain elements that players will recognise from other games, making some designs seem less original than they really are. This Editorial is an appeal for common sense from both designers and players.

Inspiration or Appropriation?

Given a new game, how much should its design differ from its closest contemporary to be considered ‘original’? There is no clear answer to this question, and I believe it largely misses the point. As long as designers credit their inspirations, and their designs provide some improvement over the prior art, then that would seem sufficient.

Consider the case of Squava, a game that maps the rules of 4-in-a-row game Yavalath verbatim from a hexagonal grid to a smaller 5 × 5 square grid. This apparently trivial change actually has a tangible effect, as the reduced area and increased 8-connectivity give the game a different character. Some players even prefer it as a ‘Yavalath-lite’: games are shorter, can be played on standard square boards, and square grids are easier to draw than hexagonal ones. The game’s designer clearly states its inspiration – even the name ‘Squava’ is an homage to its parent – so I have no issue with its status as its own game.

The video game 2048 is a more high-profile example. 2048 was an instant hit upon its release in 2014 and became something of a ‘new Solitaire’, but drew a vehement backlash by those who decried it as a clone of earlier game Threes. Again, the differences in game-play seem trivial – tiles move as far as possible per push, 1+1 merge while 1+2 do not, etc. – but these combine to give a different character. Some players even prefer it as a ‘Yavalath-lite’: games are shorter, can be played on standard square boards, and square grids are easier to draw than hexagonal ones. The game’s designer clearly states its inspiration – even the name ‘Squava’ is an homage to its parent – so I have no issue with its status as its own game.

Creative Freedom

American novelist Jonathan Lethem observes that the practice of reusing ideas from previous works has always been an accepted driving force in literature and other arts [3]. He extols the virtues of the public commons and the dangers of commodifying creative concepts, which can only stifle the wider enjoyment of the very things that are supposedly being protected. He states:

Don’t pirate my editions; do plunder my visions. [3, p. 68].

Copyright should exist to protect the works through which artists make their living, but the ideas they express should be gifts to the world. What if the inventor of Hex had patented and jealously protected the concept of connection in games? This invention would have closed down a whole field of games rather than opening it up.

As an amateur game designer, I can afford to have a laissez-faire attitude to this issue. When someone ‘borrows’ or ‘independently reinvents’ one of my ideas and releases it without reference, I usually just request that appropriate reference be made. When a Chinese company releases a pirated version of one of my games, I am secretly pleased that there is such a demand for it.

However, unjustified claims of plagiarism are extremely hurtful to game designers. I know designers accused of plagiarism despite clearly crediting their inspirations. I know one accused of plagiarism in advance, when one of his games was released in a different form by a more famous designer years later. But I also know designers who deliberately remain ignorant of the prior art so as to not limit their creative options.

With the wealth of information now available online, and the ease with which it can be accessed, there should be no excuse for designers not to be aware of similarities between their inventions and what has gone before. The onus is on designers to check the prior art and cite existing precedents; even if their rediscoveries were arrived at independently, this shows transparency and an awareness of the state-of-the-art, and places the

---

1https://www.boardgamegeek.com/boardgame/112745/squava
2http://gabrielecirulli.github.io/2048/

new work in context. The onus is then on players to appreciate what is being created from a limited pool of ideas, to judge new designs on their merits, and to focus on where they differ from previous designs rather than where they overlap.

This Issue

Most of the papers in this issue highlight the notion of drawing inspiration from prior art... as do perhaps most of the papers in all previous issues?

The opening piece by Larry Back, ‘Diamond: Improving on a Known Design’, describes the process of taking a flawed masterpiece and attempting to remove those flaws. Oskar van Deventer and Igor Kriz then draw on the mathematics of permutation groups as the inspiration for a new type of mechanical puzzle, in ‘From Computer Operations to Mechanical Puzzles’. Miguel Marqués describes the difficulties of plausibly modelling basic laws of physics as game mechanisms in ‘Physics Laws as Game Rules’, and how to capture the very first minutes of our universe in a board game. My own piece ‘Make the Design do the Work’ illustrates, through example, how games can be designed to minimise the mental effort required by players simply to follow the rules, freeing up their brains for the more interesting task of strategic planning.

Carl Hoff builds on existing work with ‘How to Make a Better 3 x 3 x 3’, in which he literally takes twisty puzzles to a new dimension to reveal another family of variants to be explored. Carl’s graphics grace our cover for the third issue in a row. This does not mean that Carl is our new in-house artist, or that the journal has taken a slant towards mechanical puzzles; it is simply a testament to the quality of Carl’s illustrations.

Craig Duncan then describes ways to address known problems with three-player games in ‘Mitigating Non-Strategic Coalitions’, Fred Horn describes ‘The Development of a Tangram Family’, Daniel Ashlock and Cameron McGuinness outline ‘Graph-Based Search for Game Design’ with example, and I present ‘A Game Design Approach to a Real-World Problem’.

We conclude with a reprint of David Parlett’s essay ‘What’s a Ludeme?’, which elucidates the history of this important term. Ludemes are the conceptual units from which games are composed, and constitute the building blocks from which new games are created.

This issue’s feature puzzle (shown below) is based on Fred Horn’s Gloop tiles. The challenges throughout the issue were generated by computer to guarantee correctness and uniqueness, and hand-tested for deducibility and quality.

As befits this Editorial’s theme, I devised this puzzle format of the Gloop tiles specifically for this issue, only to be told by Horn that he had proposed the same format to a games company a month earlier. I take this convergence as a positive sign that the design has some merit.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

Gloop Challenge #1

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32. The bottom row shows one sequence of deductions to solve this challenge.
Diamond: Improving on a Known Design

Larry Back, Freelance Game Designer

This case study describes some known problems with the board game Kensington, and how these inspired me to design a similar game that addresses these problems to provide an arguably superior result. Key steps in the design process are described.

1 Introduction

GAME design is an iterative process of refinement, as the designer hones the rules to produce the best possible playing experience. Games released prematurely – before an optimal rule set is crafted – will typically not fare very well with players, but can provide ideas for other designers to develop. This short paper describes one such case, in which an arguably flawed game inspired the design of an arguably better one.

2 Kensington

The board game Kensington, released in 1979, is played on the board shown in Figure 1. Two players, Red (dark) and Blue (light), each have 15 pieces of their colour and take turns moving as follows.

1. Placement phase: Players add a piece of their colour at an empty point, until all pieces have been placed.
2. Movement phase: Players move a piece of their colour to an adjacent empty point.

Occupying the three points of a triangle (a mill) entitles the player to move any opponent’s piece to any empty point. Occupying the four points of a square (a double mill) entitles the player to move any two opponent’s pieces to any two empty points.

The game is won by the player who occupies all six points of a white hexagon or a hexagon of their colour.

The following optional rule is also stated

It is not permitted for a player to rebuild a triangle or a square until another two moves have passed.

Kensington was well marketed, and received considerable public interest on its release, largely due to its attractive and – for the time – innovative board. The game even inspired a strategy book. However, interest among players soon waned and the game is now out of print and more of a curiosity.

2.1 The Problem with Kensington

Kensington’s decline can be traced to some serious problems inherent in its design, as discussed by players on the BoardGameGeek fora related to the game. Specifically, the game is too decisive which ironically also makes it too drawish.

\[\text{Note that a ‘move’ is understood to be a single move by a player, not a round of moves by all players.}\]

\[\text{https://boardgamegeek.com/boardgame/2197/kensington}\]
2.1.1 Too Decisive

Forming a mill (or double mill) in Kensington gives the mover a strong advantage, and the first player to do so will usually win the game. Each time a mill (or double mill) is formed, the mover is able to defuse the opponent’s most threatening pattern, preventing the opponent from forming their own mills to catch up.

The game’s designers were apparently aware of this problem, as the optional ‘no immediate rebuild’ rule stops a player moving a piece between two adjacent mills each turn to rebuild them and totally dominate the opponent. The fact that this rule is optional may have been to keep the core rule set as elegant as possible, but may also indicate that they may have underestimated the seriousness of the problem.

But the problem still exists even if this optional rule is enforced, since players may legally make a mill, break it next turn, then remake it on the following turn. This reduces the severity of the advantage but does not entirely remove it. An even stronger form of this rule may be warranted, in which players must wait two rounds rather than two moves before rebuilding a mill.

2.1.2 Too Drawish

An unfortunate side effect of the game’s overly decisive nature is that players are obliged to play defensively and block opponent’s mills as a high priority, especially during the placement phase, and mills can be hard to achieve against a skilled opponent set on defence. This can lead to games in which neither player comes close to winning before they are abandoned as draws.

The layout of the board, with the players’ home hexagons on opposite sides, encourages players to concentrate their pieces in their home areas. This may have been intended to reduce the occurrence of draws, by discouraging players from intruding into the opponent’s home area, but if so then it did not prove very successful.

3 Improving the Design

While I found Kensington flawed, I appreciated the aesthetics behind it, and liked the idea of inventing a game by first designing a unique board then devising rules to fit. After experimenting with a number of different designs, I eventually came up with the Diamond board shown in Figure 2 in 1985, based on the semi-regular tesselation 3.3.4.3.4 [2]. I later used a modified form of this tiling for my connection game Onyx [5, 6, 7].

It was not my intent to create a board that was similar to the Kensington board, but with its interlocking squares and triangles, the new board nonetheless reminded me of it. Therefore, it seemed only natural to create some Kensington-like rules for this new game, which I called Diamond.

3.1 Movement Rules

Having decided on the Diamond board, it took about a minute to devise some intuitive movement rules. These were initially similar to the Kensington rules, except that players won by occupying the four points of a square.

I liked the idea of a capturing rule based on triangles as in Kensington, but for my game I wanted specific pieces to be captured. This led to the simple rule that after a piece was moved to a triangle already occupied by two pieces, one belonging to each player, the outnumbered opponent’s piece would be captured and removed.

The resulting game was reasonably interesting to play, but tended to end too quickly, and suffered from isolated pieces being too ineffective, as two pieces had to coordinate to make a capture. The game seemed partially formed at this point, and it was not until ten months later that I hit upon the capture rule that elevated this partial game into something much more interesting.

3.1.1 Neutral Captures

The key idea that transformed the game was to replace captured white or black pieces by neutral red pieces rather than removing them. Figure 3
shows two examples of this; the outnumbered en-
emy piece in the completed triangles is converted
to red but not removed.

![Figure 3. Capture moves in Diamond. Outnum-
bered enemy singletons are converted to red.](image)

The result of this rule change was beyond
what I had hoped for. It not only lengthened
games on average, but made isolated pieces much
more useful, as a single piece could now block
the opponent from completing a square without
fear of it being removed. Further, the neutral
pieces do not belong to (and hence cannot be
moved by) either player; they temporarily block
the points they occupy from play, resulting in an
ever-changing battle field.

### 3.1.2 Non-Capturing Moves

Further improvements were made to the rules
over the subsequent years. For example, moves
that corner two enemy pieces on a move in differ-
ent triangles, as shown in Figure 4 (c) and (d), do not result in the conversion of either of the two
outnumbered pieces.

![Figure 4. Non-capture moves in Diamond. Out-
numbered enemy pairs survive.](image)

This rule was primarily intended to stop Black
starting the movement phase with a double cap-
ture to gain an overwhelming advantage. How-
ever, it soon proved to have additional benefits,
by reducing the opportunities for single captures.
This introduced some strategic subtleties into
the game by creating situations in which non-
capturing moves were preferable even when cap-
turing moves were available. It also introduced
some tactical subtleties, as players could now de-
 fend pieces under threat of capture by playing
an additional piece into a capturable position, to
create a double-capture situation that made both
pieces safe.

I also decided that self-capture would be un-
natural in Diamond. Therefore, moving a piece to
complete a triangle never results in self-capture,
even if that piece is outnumbered by enemy
dieces within that triangle, as shown in Figure 5.

![Figure 5. No self-capture in Diamond.](image)

Finally, I added a rule that neutral pieces with
no adjacent white or black pieces could be re-
moved instead of moving a piece, to reduce the
chance of draws and increase the scope for strate-
gic and tactical play. The ‘no adjacent white or
black’ restriction imposes a delay of at least two
moves between a piece being converted and its re-
moval from the board, in an elegant and implicit
way.

### 3.2 The Game of Diamond

The complete rules for Diamond, summarised be-
low, were published in the February 2013 issue
of GAMES magazine. Note that Diamond is
played with the swap rule, by which White may
elect to change colours instead of making their
first move, effectively stealing Black’s opening
move. This discourages Black from making an
overly strong opening play, to address the prob-
lem of first move advantage.
**Diamond** is played on the board shown in Figure 2. Two players, Black and White, each have 12 pieces of their colour. Players take turns moving as follows.

1. **Placement phase:** Players add a piece of their colour at an empty point, until all pieces have been placed. No captures occur in this phase.

2. **Movement phase:** Players move a piece of their colour to an adjacent empty point. Captures are performed as described above.

Instead of moving a piece, the mover may elect to remove a neutral red piece from the board, provided that it has no adjacent white or black pieces.

The game is won as soon as a player owns all four points of a square. The game is drawn if:

1. Both players agree.
2. The same position occurs for a third time with the same player to move.
3. 50 consecutive moves occur without a neutral piece being created or removed.
4. The mover has no legal moves.

### 4 Conclusion

I believe that Diamond might be the game that Kensington was meant to be. There are many similarities between the two games: Diamond has the same simple pieces that move in the same simple way; its objective is similar (but easier to achieve); and captures are also triangle-based. However, there are also significant differences between the two, notably that draws are less likely to occur in Diamond, and players cannot build and rebuild mills repeatedly by moving a piece back and forth, which is a feature of Kensington that I never liked. Diamond may not appeal to everyone, but I hope that abstract gamers who appreciate classic games with elegant rules will also appreciate Diamond.

### Acknowledgements

I would like to thank the Editor-in-Chief for providing the figures, fleshing out the concepts outlined in my original notes, and revising the material for publication.

### References


Larry Back has been an avid game player and inventor since 1982. He has represented Canada in four World Othello Championships, and has had several board games published including Onyx, 77 and Three Crowns.

**Address:** Toronto, Ontario, Canada.

**Email:** larry_back@yahoo.com

---

**Gloop Challenge #2**

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.
From Computer Operations to Mechanical Puzzles

M. Oskar van Deventer, Freelance Mechanical Puzzle Designer

Igor Kriz, Department of Mathematics, University of Michigan

This article is about transforming a virtual puzzle implemented by computer operations into a physical puzzle. We describe why a puzzle motivated by a mathematical idea of the sporadic simple Mathieu group $M_{12}$ is an interesting concept. We then describe how it was turned into two different physical puzzles called Topsy Turvy and Number Planet. Finally, we will show one version of a practical algorithmic solution to the puzzle, and describe challenges that remain.

1 Introduction

Ever since E. Rubik invented his famous cube in 1974, a boom of twisty puzzles began. The puzzles which emerged are too numerous to list here (for some information, see [1, 2]). However, after an algorithm by David Singmaster for Rubik’s cube was published in 1981 [3], it soon became clear that only small modifications of the strategy for finding the algorithm for Rubik’s cube also produce algorithms for the other twisty puzzles.

The meta-strategy is as follows: by repeating some short sequences of moves, often commutators, i.e. moves of the form $aba^{-1}b^{-1}$ where $a$, $b$ are simpler moves, one eventually finds moves which leave all but a small number of pieces in place. Additionally, positions of pieces affected by the move can be customised by finding setup moves, i.e. sequences of moves which get other pieces to the places where the original move took place, performing the move, and then backing out the setup sequence to move the chosen pieces instead. Invariably, with such generalised moves, which move only chosen small sets of pieces, the puzzle can be then solved in an incremental way.

Has the puzzling potential of twisty puzzles therefore been exhausted, i.e. is there no twisty puzzle which would require a completely different strategy? Algebra suggests that this should not be the case, and that substantially new kinds of puzzles should exist. The mathematics of a puzzle is determined by its permutation group [4], which is basically the set of all possible moves one can make on the puzzle. In all of the twisty puzzles on the market, the permutation group consists of almost all permutations, with restrictions on only a few types, which do not alter the strategy for solving the puzzle substantially. However, in mathematics, we know that some very strange permutation groups exist. The strangest ones are called sporadic simple groups [5]. The word ‘simple’ here refers to the fact that they are building blocks for finite groups, just as prime numbers are basic building blocks of natural numbers. However, it may be a misnomer to call those groups simple, since they are typically extremely complicated.

Could one make a twisty puzzle based on a sporadic simple group, which would then require a completely novel strategy, different in essential ways from Singmaster’s algorithm? The problem is that sporadic simple groups usually contain very strange permutations which often arise from higher mathematics, but do not have any easy interpretation in the real world. The idea of making a hand held puzzle from one of these groups certainly seemed out of reach.

2 Computer Implementation

In 2008, the second author, along with University of Michigan student Paul Siegel, investigated the question of whether a mechanical model of a puzzle was really necessary to make it interesting. Could a puzzle be a computer program, where the player would, by moving or clicking a mouse, instead of twisting a physical puzzle, execute a move? Would anyone be interested in solving such a puzzle?

They soon realised that interesting computer puzzles could be made with several sporadic simple groups [6]. The simplest, and perhaps most popular, such puzzle used the Mathieu 12 ($M_{12}$) sporadic simple group. This puzzle has two basic moves, Invert and Merge, as indicated in Figure 1. The Invert move reverses the order of the numbers. The Merge move is akin to a card shuffle. The object of the $M_{12}$ puzzle is similar to the object of the Rubik’s Cube: scramble and solve by using only the two permutations.

---


To get a glimpse of how strange the $M_{12}$ group is, it is a fact that any five chosen numbers $1, \ldots, 12$ can be moved to any five positions in a chosen order by repeating the Invert and Merge moves, but the positions of all the other numbers are then uniquely determined. Thus, this puzzle has $12 \times 11 \times 10 \times 9 \times 8 = 95,040$ positions. Every move (except the trivial, or dummy move, which leaves all the places where they were before), moves at least eight of the twelve pieces to different positions. Therefore, no incremental strategy based on moving only a small number of pieces can work for $M_{12}$.

### 2.1 Recursive Strategy

There is, however, an appropriate strategy for this problem. If 1 was not already in its correct position, by repeating of the Merge move, it can be eventually moved to the position where the number 12 should be, and then by the Invert move, it could be moved to its correct position. With 1 in its correct position, let us think about moves which do not disturb the position of 1. To find such moves, perform some random sequence of moves which does move 1 out of place on purpose, and then use the above strategy to get it back to its correct position. This way, we get moves which do not change the position of 1. By finding enough such moves, we can find a set of moves which eventually can be combined to put 2 into its correct position without disturbing the position of 1. Now repeat the strategy to find moves which leave 1 and 2 in place, and could be used to move 3 to its correct position. Repeat the same process for 4 and 5.

Siegel soon noticed that the recursive strategy described above soon leads to sequences involving thousands of moves, so he invented a way to write macro instructions, encoding sequences of moves as a single move, and also a way of using such macros to make new macros. These improvements made the computer version of $M_{12}$ challenging yet fun. These results for $M_{12}$ were published in a 2008 *Scientific American* article [6].

### 2.2 Mechanical Challenge

We realised that it would be even more interesting to find a mechanical version of the $M_{12}$ puzzle, which would not permit the kind of cheating with recursive macros, which lead to extremely long sequences of moves, even sometimes involving thousands of moves. This raised some general questions:

- Can a puzzle represented by computer sequences be implemented in a mechanical design?
- Can a hand-held puzzle be engineered from a prescribed set of moves represented by abstract permutations in such a way that only the prescribed moves and not additional cheating moves are possible, which would spoil the puzzle by allowing unintended positions?
- If it is possible, what are the specific challenges and methods for finding a solution of such a puzzle?

These questions are difficult to answer in general. We will show here some special properties of the $M_{12}$ group which allow the problem to be approached in this specific case. As far as we know, the approaches we developed are new and have not previously used in puzzle design.

### 3 Designs

Two different designs, which can be treated as separate case studies, were developed.

#### 3.1 Topsy Turvy

The first design was developed by the first author alone, and is based closely on the computer model. The Invert operation is easily implemented, as one can just turn around the set of tokens. The Merge operation is more challenging.

The initial idea was to implement the Merge operation using some ad hoc construction, but a suitable mechanism could not be found. However, the inverse operation, Split, might be easier to implement, as the second author confirmed.
The first author had recently developed several puzzle mechanisms to manipulate dropping tokens, including the Jukebox and Pachinko puzzles shown in Figures 2 and 3. Jukebox uses physical switches to alternate moving tokens left and right. Pachinko uses a pattern of grooves that holds one or two tokens until one more token is inserted that pushes the other tokens down. This design provided a suitable basis for a 12-splitter mechanism needed for the Split operation.

Figure 4 shows a working prototype of resulting puzzle, Topsy Turvy, in which a large crank is used to move the twelve tokens into the 12-splitter. The Invert operation is implicitly implemented, as the crank can turn left or right.

When turned, the twelve tokens are dropped in one by one. The first eleven of the tokens will land stably on top of each other. However, when the 12th token drops, it rolls down over the 11th and while dropping it pushes the 11th out of position. Then the 11th pushes the 10th out of position, which pushes the 9th, which pushes the 8th, etc. In a cascade, the whole stack of tokens falls apart, with the even tokens moving to the right and the odd ones to the left.

Several prototypes had to be built to get the mechanism right, in medium density fibreboard (MDF) and acrylic, using the laser cutter of colleague Peter Knoppers, and tokens of cast tin. In order to have the mechanism turn smoothly, four large gears act as ball bearings (Figure 5), carrying the weight of the crank mechanism. A ratchet mechanism (Figure 6) is used to force the completion of moves once they are started.

3https://www.youtube.com/watch?v=H8ZcYvU0sLY
The first prototype had the serious problem that a user can continue turning while the tokens are still dropping; the tokens will collide when they are caught by the crank mechanism at the bottom, enabling illegal moves and blockage. Fellow puzzle designer George Miller found an elegant solution; a toggle switch which limits the rotation of the crank between $-240^\circ$ and $+240^\circ$ (Figure 7). With the switch in place, a user has to turn the crank all the way back, which takes sufficient time for the tokens to settle at the bottom. Magnets were used to make the switch bi-stable.

Another problem was that tokens could skip the entry if the crank is turned too fast. Knoppers’ solution was to place a pin at the top entry of the grooves, which forces the tokens down. With all major problems solved, the third prototype was found to work satisfactorily.

### 3.2 Number Planet

During the development of Topsy Turvy, the second author suggested a completely different implementation, based on a special planar permutation that also implements the $M_{12}$ group up to isomorphism [6]. The two permutations are Rotate and Swap, as shown in Figures 8 and 9, and the idea was to have a rotating mechanism that performs these two permutations.
Together, we found a mechanism that could achieve this, shown in Figure 10. Eleven tokens are placed in a circle, enabling the Rotate operation. Five disks and rings of varying size are used to perform the pair-wise swaps needed for the Swap operation. Notice that the 0 and 1 are not explicitly swapped. The tokens can be in two rows. After making the five turns for a swap, the tokens are in the other row, implicitly swapping 0 and 1.

However, we were not satisfied with this design’s round tokens, which became unstable when the mechanism was operated. A better mechanism would be possible if the 0 and 1 were not surrounded by the 11–2 swap. We independently looked for a better planar permutation, using a Python program and a Maple program, respectively, and within minutes of each other both found the desired planar permutation shown in Figure 11.

It is curious that no other planar permutation exists which would generate $M_{12}$ together with the 11-cycle, which was proven by the Maple program enumerating all possible permutations. The first author then made a 3D design of the successful permutation using trapezoid-like tokens, shown in Figure 12 which are more stable when they push each other during a Rotate operation.

The puzzle, called Number Planet by the second author, was prototyped using 3D printing technologies. The first prototype, made by UM3D using fused deposition modelling (FDM), was completely stuck due to a modelling error. A second prototype, shown in Figure 13, worked reasonably well, but was still cumbersome as its many subassemblies made it hard to put together.

A third prototype, made by TNO Netherlands using nylon-based selective laser sintering (SLS) painted with textile acid dye, shown in Figure 14, was finally a success. Figure 15 shows details of the pieces.
Even better versions of the design have been made since then\textsuperscript{4}, including a smoothly running Number Planet prototype using triangular-shaped tokens\textsuperscript{5}. Figure 16 shows a recent redesign of the trapezoidal token-based Number Planet in the shape of a perfect ball, made with the help of University of Michigan student Varun Gala. A smooth prototype of the spherical design, printed on a Pro Jet printer, is shown in Figure 17.

3.3 Solution Algorithms

With working prototypes available, the reader may wonder how these puzzles should be solved. First of all, it should be noted that both implementations feature two different permutations, as shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Topsy Turvy</th>
<th>Number Planet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Left</strong></td>
<td>1-2-3-4-5-6-7-8-9-10-11-12 → 11-9-7-5-3-1-2-4-6-8-10-12</td>
<td>0-1-2-3-4-5-6-7-8-9-10-11-12 → 0-2-3-4-5-6-7-8-9-10-11-12</td>
</tr>
<tr>
<td><strong>Right</strong></td>
<td>1-2-3-4-5-6-7-8-9-10-11-12 → 2-4-6-8-10-12-11-9-7-5-3-1</td>
<td>0-1-2-3-4-5-6-7-8-9-10-11 → 0-1-9-4-3-6-5-8-7-2-11-10</td>
</tr>
</tbody>
</table>

Table 1. Permutations by puzzle.

Secondly, both mechanisms implement the $M_{12}$ group, which has $12 \times 11 \times 10 \times 9 \times 8 = 95,040$ permutations. The $M_{12}$ group has the property that if five tokens are at their correct place, then the other seven tokens are correct too\textsuperscript{4}. Using this information, one can make an oracle or ‘God’s Table’ which enumerates the solution sequence from each possible state. This is exactly what George Miller did, in a Python program small enough to be used effectively on a smart phone, so that one can always have a solution at hand.

\textsuperscript{4}https://www.shapeways.com/product/2VAKFZVH8/number-planet
\textsuperscript{5}See https://www.youtube.com/watch?v=UvjXQQZQA-s for a video of its operation.
Every position can be solved in nine moves, counting a Swap or any power of Rotate as one move. Although the oracle provides the fastest solution, it is impossible for a human being to memorise. The recursive method is not practical for mechanical versions of the puzzle as the move sequences get too long.

It is, however, possible to develop an algorithm which combines both approaches. Using this algorithm, the puzzle can be solved in about 10 minutes. We shall use the symbol $a$ for the Rotate move and $b$ for the Swap move. The strategy is first to get 1 in its correct position. This can be accomplished as indicated above. Once 1 is in its correct position, we are interested in moves which leave 1 intact.

The following table shows what move to execute when 2 is in the position shown:

<table>
<thead>
<tr>
<th>2 in Posn</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$ba^5b$</td>
</tr>
<tr>
<td>4</td>
<td>$ba^6b$</td>
</tr>
<tr>
<td>5</td>
<td>$ba^3b$</td>
</tr>
<tr>
<td>6</td>
<td>$ba^4b$</td>
</tr>
<tr>
<td>7</td>
<td>$bab$</td>
</tr>
<tr>
<td>8</td>
<td>$ba^5baba^7$</td>
</tr>
<tr>
<td>9</td>
<td>$ba^7b$</td>
</tr>
<tr>
<td>10</td>
<td>$ba^8b$</td>
</tr>
<tr>
<td>11</td>
<td>$ba^2ba^8bba^2b^2$</td>
</tr>
<tr>
<td>0</td>
<td>$ba^8b$</td>
</tr>
</tbody>
</table>

Table 2. Moving 2 home.

After 1 and 2 are in their correct places, we then look for moves which do not disturb these numbers. The following table shows what move to execute when 3 is in the position shown:

<table>
<thead>
<tr>
<th>3 in Posn</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$ba^4ba^7ba^6b$</td>
</tr>
<tr>
<td>5</td>
<td>$ba^2ba^7ba^6b^3$</td>
</tr>
<tr>
<td>6</td>
<td>$ba^8ba^7ba^6b^3$</td>
</tr>
<tr>
<td>7</td>
<td>$ba^9ba^3ba^2b^7$</td>
</tr>
<tr>
<td>8</td>
<td>$ba^10ba^9ba^7ba^4$</td>
</tr>
<tr>
<td>9</td>
<td>$ba^4ba^3ba^6b^2$</td>
</tr>
<tr>
<td>10</td>
<td>$ba^8ba^7ba^9ba^6$</td>
</tr>
<tr>
<td>11</td>
<td>$ba^8ba^3ba^6bba^2b^4$</td>
</tr>
<tr>
<td>0</td>
<td>$ba^8ba^3ba^6bba^10$</td>
</tr>
</tbody>
</table>

Table 3. Moving 3 home.

Now that 1, 2, 3 and 4 are in their correct respective places, we need to find moves which do not disturb these numbers. The following table shows what move to execute when 5 is in the position shown:

<table>
<thead>
<tr>
<th>4 in Posn</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$ba^2ba^7ba^6bba^9ba^3$</td>
</tr>
<tr>
<td>6</td>
<td>$ba^10ba^7ba^6b^7ba^10$</td>
</tr>
<tr>
<td>7</td>
<td>$ba^2ba^8ba^4ba^2b^7ba^10$</td>
</tr>
<tr>
<td>8</td>
<td>$ba^8ba^3ba^6bba^2b^4$</td>
</tr>
<tr>
<td>9</td>
<td>$ba^8ba^3ba^6bba^10$</td>
</tr>
</tbody>
</table>

Table 4. Moving 4 home.

Now that 1, 2, 3 and 4 are in their correct positions, we must find moves which do not disturb these numbers. The following table shows what move to execute when 5 is in the position shown:

<table>
<thead>
<tr>
<th>5 in Posn</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>$ba^6ba^9ba^8ba^2b^3$</td>
</tr>
<tr>
<td>7</td>
<td>$bababa^5ba^9ba^7$</td>
</tr>
<tr>
<td>8</td>
<td>$ba^10ba^8ba^4ba^7ba^10$</td>
</tr>
<tr>
<td>9</td>
<td>$ba^6ba^2ba^6ba^6ba^6b$</td>
</tr>
<tr>
<td>10</td>
<td>$ba^2ba^8ba^7ba^6ba^10$</td>
</tr>
<tr>
<td>0</td>
<td>$ba^6ba^2ba^6ba^6bba^5b$</td>
</tr>
</tbody>
</table>

Table 5. Moving 5 home.

Can a human being find a solution to Number Planet on their own, without the aid of a computer, as Singmaster found a solution to Rubik’s Cube? A solution using heuristics only is not known at this time; the challenge for serious puzzlers is still on!

4 Conclusion

We have demonstrated that converting a computer based permutation puzzle to the physical world is possible, although it is dependent on the exact moves involved in the model, and their special properties have to be used in an innovative way. In the examples presented here, we showed how gravity and rattles can be used to physically implement a shuffle design, and how planarity can be used for implementation in the form of a rotating mechanism.

Regarding solving the puzzle, we demonstrated that recursive solutions are impractical for physical versions. We showed how a computer algorithm can be used to generate an oracle, and how partial information from such a table can be used to formulate a practical algorithm.

We also constructed a permutation puzzle which is completely new and qualitatively different to Rubik’s cube, and remains an interesting challenge for puzzlers, as the problem of finding a solution using logical analysis without computer assistance remains open.
With high definition 3D printing, we were able to achieve a smoothly turning mechanism for our puzzle. The puzzle is pleasant to look at and hold, and is fun to play. Our method for constructing the puzzle from the computer model employs general principles which were not used before and which can be applied to the design of other puzzles. However, further innovation would be required to make another meaningful puzzle as novel as the one presented here, and a new invention would be needed to apply our method to another sporadic simple group. From this point of view, our construction is unique.

5 Acknowledgements

We thank the reviewers for their useful feedback and the editors for formatting this piece.

References


M. Oskar van Deventer is one of the world’s most prolific designers of mechanical puzzles, many of which are commercially available. 

Address: Leidschendam, The Netherlands. 

Email: m.o.vandeventer@planet.nl

Igor Kriz is Professor of Mathematics at the University of Michigan. His primary mathematical interest is algebraic topology.

Address: Mathematics Department, University of Michigan, Ann Arbor, MI 48109-1109, USA.

Email: kriz.igor@gmail.com

Gloop Challenges #3 and #4

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.

Gloop Challenge #3

Gloop Challenge #4
Physics Laws as Game Rules

F. Miguel Marqués, CNRS

Physics laws are the rules of the ‘game’ we all live in. However, designing a board game out of them is not the straightforward process one would expect. This article discusses some of the issues that may be encountered, and illustrates several concepts through a design exercise, a board game that recreates the very first minutes of our universe.

1 Introduction

Let me start by saying that I am a physicist that designs board games. This paper dealing with the link between physics and games, most of what follows corresponds to my personal view from my experience in both fields.

The aim of physics is finding the laws that explain how the universe we live in works. These laws should not only describe the phenomena that we witness today in our surroundings; they should let us reconstruct how the universe evolved from its origin to its current form and, ultimately, let us model the direction it will take in the future. We look for simple patterns within complex environments, trying to derive from them laws that are few and simple.

In thematic board games, designers try to abstract the events they want to recreate into few and simple rules. This exercise is required to a lesser extent in video games, in which complex algorithms may be hidden behind the pressing of a button, but is essential in board games, where the players must execute manually on the board the movement and behaviour of all the game pieces. This abstraction often requires more or less arbitrary choices and simplifications, that often lead to more or less complex rule sets.

So on one hand we have designers trying to translate the events of their games into few and simple rules, and on the other hand we have real events that obey few and simple rules already. Logically, one should expect a lot of board games about physics, since the abstraction work has already been done by nature in the form of laws which could be taken almost directly as rules for board games. So why are (good) board games about physics so rare? How should one proceed in order to make a game from physics laws?

In this paper I will try to answer these questions. I will distinguish between teaching tools and serious games, then deal with the concept of design for effect and discuss some examples of games that have translated a law into rules, with uneven success. In order to illustrate the wider scope of the problem, I will then spend more time considering a case study, in which all the steps from the physics to the final game will be accessible. It will be shown that the abstractions performed by nature in the form of laws are not always well suited for a board game.

1.1 Games or Teaching Tools?

The majority of board games about physics I am aware of fall into three main categories:

1. Pedagogical tools used for the popularisation of science.
2. Detailed simulations of complex processes.
3. Board games with a physics theme.

At science exhibitions I have observed an increasing trend to propose tools that give a flavour of physics in a playful framework. Their main purpose is to attract young people to the field, but can be also used to teach physics at an introductory level. For example, the ‘Billotron’ [1] uses marbles rolling on a flat surface to illustrate Rutherford’s famous 1911 experiment designed to discover the atomic nucleus. Further, the ‘Supernova Fountain’ [2] displays the complex instabilities that appear in the first stages of a supernova explosion with flowing shallow water. However, most of the teaching ‘games’ I have seen are not very good as games. They do teach, or give an idea of, some physics processes, so they fulfill their mission, but they were not designed to provide a competitive, challenging or fun gaming experience. In fact, most of their creators are scientists unaware of the board gaming hobby.

Setting these objects aside as teaching tools, we are left with the other two categories, in which the game itself is the goal. The border between detailed simulations and board games with a theme is not well defined in general. In almost any gaming theme one can find complex, highly detailed games and lighter, more abstracted ones, the latter being the more popular offer nowadays. If the laws of physics are simple, though, why should the games be complex?
2 Simple Laws, Complex World

Looking around, it is easy to realise how complex our world is – and how beautiful! We belong to the first generations to have revealed the cleverness of our universe’s structure: very simple rules and interactions lead to a huge number of layers, structures, and even living organisms. And some of those have been able to find the rules! Therefore, the laws that govern a given interaction between two bodies may be simple, but when several kinds of interaction combine, or more than two bodies fall within the interaction range, the interplay between these simple individual ‘recipes’ becomes wonderfully complex.

Take Newton’s law of gravitation, for example. A body of mass $M$ attracts other bodies at a distance $d$ inducing an acceleration proportional to $M/d^2$. This is very simple. Double the mass, double the acceleration; double the distance, quarter the acceleration. However, add other massive bodies, let them all move, and soon things become convoluted. Of course, since the forces are simple and have analytical form, even the most complex trajectories can be calculated using a computer, and thus be implemented in video games. But board games cannot benefit from this assistance.

![Figure 1. A game of Triplanetary in progress.](image)

Let us consider the ancestor of many space board games, Triplanetary1 (1973), that used vector movement modified by gravity. It assumed the most simple scenario: the massive bodies (sun, planets and moons) are fixed on a hexagonal grid and their gravitational fields extend only one hex away, so that they do not overlap. Therefore, the actual law was made even simpler, by discretising the mass dependence to heavy or light (moons, with optional gravitational pull) and the distance dependence to either 1 or ‘infinity’.

Even under these drastic assumptions, however, tracking the movement of the space ship units required the use of markers on a laminated map (Figure 1). The game met some success in the 70s and 80s, and Steve Jackson Games acquired the rights to publish a 3rd edition in 1991, but it never happened. It was a reasonable simulation of movement in empty space, that displayed clearly how space probes make use of other planets’ gravitational pull to accelerate without using fuel. However, drawing and calculating the trajectory of units on a board was probably acceptable forty years ago, but not in the present board gaming market.

3 Complex World, Simple Games

If the world is complex, and we do not want our game to be, then why should we copy almost every ingredient into the game? I recently discussed the concept of design for effect in war games [3], which can be extended to any board game based on a real situation.

On one hand, we can try to incorporate any aspect of the action into the game, organise the related rules into layers, add exceptions to those rules in order to force the sequence of events... Designers have long thought that this was the best way to recreate a given scenario, since the sequences of events and the endgame results were close to the real ones. But the cost is that following fiddly layers of rules often makes players feel like they are working rather than playing.

In the design for effect philosophy, designers try to pin down the few main concepts that are relevant to the events they are modelling in their game, and reflect the myriad of potential details in a way that is imperceptible to the players. The two key words are ‘game’ (which should be interesting to play) and ‘imperceptible’ (the players should feel the experience without noticing the details). This approach is definitely more challenging for the designer, but in my opinion leads to better games.

Some argue that by abstracting away the details the simulation aspect decreases. However, a loss in simulation detail is compensated by a gain in simulation experience felt by the players. But let us leave people’s feelings aside and return to physics.

3.1 Gamifying Simple Laws

Let us consider two examples of recent games that exploit physics principles without going into explicit details. The first one is Gauss2 (2009),

---

1http://www.sjgames.com/triplan/
2http://nestorgames.com/#gauss_detail
a game about attraction and repulsion between magnets. The basic law behind it can be seen also as the electric force that attracts/repels charges of opposite/same sign, which has a form very similar to the gravitational one we have seen: \( \frac{q_1 q_2}{d^2} \), the product of the two charges divided by the square of their relative distance.

In Gauss, the two colours represent charges of opposite sign and all distances are considered small enough so that the result of the force is always the same: push/attract charges of same/opposite sign until they hit an obstacle (other charges or the board boundary). Moreover, the force acts only along the six directions of the hexagonal grid (see Figure 2).

\[\text{Figure 2. Red pushes red and attracts blue.}\]

The aim of the game is forming groups of your pieces, trying to avoid the outer ring, by dropping pieces on the board. A true challenge because your own pieces repel themselves! Even if the game is about plastic pieces on a board, and the underlying law has been simplified to a minimum, players comment about how playing Gauss evokes their memories of science classroom with red and blue metal magnets clashing together and spreading away. A successful design for effect, and a challenging game.

Another example is Momentum (2010), which simulates the transmission of momentum through a row of metallic balls hanging from pendula known as Newton’s cradle. Drop the ball at one end and its momentum is transmitted through the row to its opposite end, where the final ball completes the oscillation. Then the latter swings back, striking the row again, repeating the pattern while the intermediate balls remain immobile spectators to the play of the two outer balls. The underlying law of physics here is the conservation of energy and momentum.

In Momentum, players drop pieces trying to hit rows already on the board along the eight directions of the grid, like a multiple Newton’s cradle. In every row, the furthest piece is pushed one space (see Figure 3). The aim is to place all your pieces on the board, while kicking the opponent’s pieces outside. Again, even if players are not manipulating a complex structure of steel balls hanging from pendula, precisely equilibrated and aligned, they feel like they are.

\[\text{Figure 3. The blue piece just played pushes the furthest piece along each line radiating from it.}\]

Therefore, these games succeed in simulating the experience of people in science classrooms, even if the real processes have not been simulated in high detail. But most importantly, they provide players with a very clear aim, that leads them to an interesting gaming challenge. It is worth noting that similar games using much more sophisticated components exist. In Polarity (1986), players try to place real magnets on a cloth board. In Abalone (1987), players push rows of marbles along a hexagonal grid in relief on which the marbles slide. Gauss and Momentum, however, communicate the same experience with more familiar pieces on standard boards.

4 Case Study: The Big Bang

Let us now discuss the gamification of a physics case that is fascinating and complex, but that we can already describe in broad outline: the formation of our universe! The main stages of the process are sketched in Figure 4. At some point, a huge explosion we have named Big Bang liberated all the energy in our universe, which from there on expanded and saw its temperature decrease until the present 3K (−270°C), as explained in [5, 6].

3http://nestorgames.com/index.html#momentum_detail
4http://bgg.cc/game/380
5http://bgg.cc/game/526
The energy from this explosion materialised into equal amounts of matter and antimatter, which then annihilated each other into energy again, following a deadly cycle which could only be broken by a tiny excess of matter. The origin of this slight excess, responsible for the matter that we see today and of which we are made, is not fully understood yet. After the first second, the surviving matter had taken the form of protons and neutrons (Stage 1). Those two particles began to combine and in the very first minutes formed the lightest nuclei, mostly hydrogen and helium (Stage 2), but could go no further.

The universe underwent a long and quiet period until electrons were slow enough for them to be captured by those light nuclei, forming the first atoms (Stage 3). Then gravity took the lead and the neutral atoms began to condense into clouds, which further condensed into stars (Stage 4). Inside stars, hydrogen fused again into helium, and thanks to the strong gravitational fields three helium nuclei could fuse into carbon, going beyond the limit reached at the end of Stage 2. From carbon, fusion kept going until the formation of iron, which can no longer sustain fusion reactions, and then stars collapsed under their own gravity and exploded (Stage 5).

The extreme violence of these explosions, known as supernovae, created the environment needed to build up heavier nuclei beyond iron and up to uranium, and dispersed them into space. And then back to Stage 4; atoms condensed into clouds and new stars, but now those clouds contained most of the elements, and around these second generation stars there could be rocky planets on which life could develop (Stage 6).

This complex process, spanning 14 billion years, can nevertheless be sketched in six main stages. What kind of game can be designed out of this? Some thoughts follow:

- The space and time scales are too vast, the stages too diverse, the interactions governing them too different. We should thus focus on parts of the overall process.
- Stages 1 and 2 look easier to game, with only a few well identified pieces. But a goal for the players must be found.
- Stage 4 is interesting, starting with the formation of carbon. However, the ingredients of the nuclear reactions involved are many, and the role of gravity hard to implement.
- Stage 5, the explosion of stars, is even harder, and stage 6, the formation of planets, does not seem very dynamic.

We start by considering the formation of carbon, the basis of life as we know it.

### 4.1 A Race to Carbon

Carbon has a relatively light nucleus, with six protons and six neutrons in its most abundant form, carbon-12 ($^{12}$C). If we want to design a game about the formation of carbon, then protons and neutrons should be the natural game pieces. In Figure 4, those pieces appeared at Stage 1, formed hydrogen and helium at Stage 2, and then waited until Stage 4 to continue forming heavier systems. We can better understand why with the diagram in Figure 5, in which we see all the nuclei that exist up to carbon-12.

For each combination, the number of protons (red, top line) gives the element name (from 1 to 6: hydrogen, helium, lithium, beryllium, boron and carbon), and the number of neutrons (blue, bottom line) defines its mass number (protons plus neutrons). For example, lithium-8 has three protons and five neutrons.

---

6. An atom’s nucleus is formed by combinations of protons (charge +1) and neutrons, each element having a characteristic number of protons. The atom consists of a nucleus surrounded by a cloud of electrons (charge −1). The number of electrons in the cloud equals the number of protons in the nucleus, so that the atom is neutral.
Figure 5. Nuclei up to carbon-12, classified by number of protons (red) and neutrons (blue).

Only some combinations are allowed, and just a few are stable (white cells). The rest are unstable because the equilibrium of ‘pieces’ is too unbalanced, and after a given time they will undergo radioactive decay in search of balance, by transforming a proton into a neutron (pink cells on the upper left) or a neutron into a proton (cyan cells on the lower right)\(^7\). The colour shades correspond to the varying decay times – the darker the shorter – that range from millions of years to tenths of a second.

At the end of Stage 2, free neutrons had disappeared and most of the pieces were in the form of the most stable \(^1\)H and \(^4\)He. The reason why the process stalled is displayed by the forbidden symbols in Figure 5: none of their binary combinations (\(^2\)He, \(^5\)Li, \(^8\)Be) are allowed. Due to this quirk of physics, hydrogen and helium had to wait a billion years until gravity could play a significant role inside stars, enabling the ignition of more complex reactions and, in particular, the one that fuses three \(^4\)He directly into one \(^12\)C.

For the game, we could then use red and blue pieces on a hexagonal grid, similar to that in Figure 2 but bigger, and let the players fuse them into stacks following the patterns of Figure 5. These are some potential game issues:

- Several combinations are unstable, so in addition to the fuse action there should be a decay action, in which a proton/neutron in the pink/cyan stacks was replaced by a neutron/proton. It could be a random mechanism (as the real decay) using dice, or a voluntary choice of the players.

- Players should use the Figure 5 chart as an aid in their race to carbon. They could fuse stacks to increase their size, then choose to follow the stable (white) diagonal or either of the two coloured regions and then return to the diagonal via decays.

- Players would not be ‘red’ and ‘blue’; they would need (and share) all the pieces.

- The quirk of physics leading to the triple fusion of \(^4\)He into \(^12\)C should be a key ingredient. Players could fuse two stacks along empty straight lines, or three adjacent stacks. And, of course, the result should be a valid known nucleus.

The components and mechanisms seem clear, the aim of the players not so much. If the winner is the first player forming carbon-12, and since both players share the same pieces, then the game could stall with players not wanting to do the next-to-last move. No player would want to put two \(^4\)He nuclei together, enabling the opponent to add a third one.

We could instead award points for the formation of each element, as an incentive for both players to contribute to the race, but this would require a detailed balance analysis to determine the optimal number of points per element, or make the game a cooperative one in which the players solve a puzzle and try to maximise the formation of carbon stacks.

In any event, Figure 5 is too convoluted to make an effective player aid. Even nuclear physicists would need to constantly refer back to the aid to check what can or cannot be done, and we are looking for a game, not homework. Further, the unstable combinations have decay times ranging from tenths of a second to millions of years (\(^{10}\)Be), so we should establish a hierarchy. Moreover, from a practical point of view, moving stacks of up to 12 pieces and replacing pieces inside them would be cumbersome.

Even if the initial idea was good, the game boundaries clear, and the number of pieces small, this ‘Race to Carbon’ would be far from the simplicity and elegance of Gauss or Momentum, and the tweaks required to address its various issues would take it even further away. If we want to design the Big Bang for effect, we must escape this frame.

4.2 A Race to Helium

Returning to the timeline in Figure 4, using protons and neutrons as the game pieces was the...
The best part of the previous idea, and the complexity of the nuclear chart up to carbon-12 was the worst. So let us keep the good idea, but limit it to Stages 1 and 2, to give a race to build helium-4, one of the most stable bricks in the universe and the precursor of carbon.

The orange (lower left) region of Figure 5 shows how simple the nuclear chart becomes; it contains only two game pieces and four composite stacks, with only one of them (\(^3\)H) unstable. The player aid becomes trivial even for non-scientists: no more than two protons or neutrons per stack, and not only two of them alone. Players should fuse pieces up to \(^4\)He, and the decay option would be open only for the neutron and \(^3\)H, with a straightforward hierarchy (\(^3\)H decays more slowly).

This is conceptually closer to Gauss or Momentum, with stacks of at most four pieces. However, forming \(^4\)He would be relatively easy, so the aim of the game could not be being the first one to do so. Players could instead aim to make the most \(^4\)He, sharing the red and blue pieces and keeping track of how many they created. But this could again lead to deadlocks, as players would be disinclined to create \(^2\)H atoms near existing ones, otherwise the opponent could fuse them into a \(^4\)He nuclei.

4.3 Player vs Antiplayer

We have found an appropriate framework for the game, but lack a mechanism that generates competition and a clear aim for the players, mostly due to the fact that they share the red and blue pieces. So what about incorporating other elements from the physical scenario as game pieces?

Hidden between the Big Bang and Stage 1 (Figure 4), there was a huge production of matter and antimatter in almost equal quantities, followed by a huge annihilation of both into light. The matter we see around us today comes from a tiny, still mysterious excess that survived. If we started the game before Stage 1, then we could incorporate such elements as antipieces.

Antiparticles have the same properties as their corresponding particles but the opposite charge. For example, antielectrons are positive and antiprotons negative, but those two particles can combine to form an antihydrogen atom, with properties similar to a standard hydrogen atom. Or an antineutron plus an antiproton can form an antihydrogen-2 nucleus.

Figure 6 shows the mirrored antimatter images of the nuclear chart up to helium-4. Again, the nuclei shown in the upper right region are classified by the protons (red, top line) and neutrons (blue, bottom line), while their antimatter counterparts shown in the bottom left region are classified by their antiprotons (black, top line) and antineutrons (grey, bottom line).

This solves the problem of sharing the pieces, as one player would use red and blue pieces, while their opponent – the ‘antiplayer’ – would use the black and grey ones. Each player now has a clear aim, to build the most \(^4\)He, \(^8\)since both \(^4\)He are different, and there is no longer a need to keep track of exact particle counts throughout the game.

Moreover, a law of physics provides a new ingredient that leads to lots of player interaction: annihilation. The original idea was based on a ‘quiet’ construction of nuclei, but with these new pieces players can now not only build their own nuclei in parallel, but also annihilate the opponent’s! We can even make it more interesting by forcing the players to choose between these two options, leading to an interesting dilemma similar to that found in the game of Tzaar:

\[
\text{Shall I make myself stronger or my opponent weaker?}\]

5 Big*Bang

This section describes the process of finalising the rules for the resulting game of Big*Bang.

---

\(^8\)Antiparticles are usually noted with a bar, e.g. \(^4\)He, but I use the same symbol for both here, for simplicity.

\(^9\)Quote from the official Tzaar web site: http://www.gipf.com/тааr
5.1 Refining the Goal

We have now found a suitable frame for the game (synthesis of helium), its pieces (protons, neutrons and their antiparticles), mechanisms of play (fusion, annihilation and decay), and some ideas for its goal (such as building the most helium). We have also introduced player interaction through matter/antimatter annihilation. However, unlike most games that involve capture, such annihilations result in the removal of both players' pieces, which has a somewhat self-defeating feel to it and doesn’t let players strengthen their own pieces or position. But since annihilation transforms each matter/antimatter pair into light, we can instead use this mechanism as a new aim: to produce the most light.

This allows two paths to victory: a pacifist path that involves building the most helium, and a bellicose path that involves annihilating matter/antimatter pairs into light. If each player concentrated in one path, the game would not be very fun or strategic, and would lead to draws. This could be addressed by introducing a scale with which to compare the relative magnitude of each victory condition, but this would complicate matters and make the game confusing for players. Instead, I chose a third option.

There is another victory condition that is easy for players to understand and that respects the laws of physics. The formation of helium was followed by the formation of stars, and those were powered by the fusion of hydrogen. At the endgame, after particles have disappeared through annihilation and others have fused into stacks, the board will look like clusters of nuclei. If we identify the clusters of each player as their stars, an interesting victory condition would be to form the star with the most hydrogen fuel. Players should therefore fuse particles into hydrogen and then helium; produce light by annihilating pairs; but at the same time keep some hydrogen 'alive' in some of the clusters that appear towards the end. This makes the number of victory conditions odd, so ties should be unlikely.

What about the decay of unstable combinations? With respect to the carbon-12 game idea, we are left with only two of them, each player’s neutron and $^3$H. To add more interaction, we could let players force the decay of the opponent’s unstable stacks, in order to disturb their plans. Since decay here means replacing a neutron with a proton, it would be available only when protons would start leaving the board through annihilation. Therefore, the players themselves would regulate the decay ‘clock’ (the timing and impact of decays) depending on how much annihilation they chose, making no two games play the same.

The turn sequence would be:

1. Either fuse a pair of your stacks or annihilate a pair stack/antistack.
2. Force a decay, if possible.

5.2 Board Design

As with Gauss or Momentum, the game is abstract in the sense that it focusses on simple concepts that evoke laws governing real processes, so is best played on a regular grid. However, it still has a strong theme, which might be lost on players if the board is a sterile grid.

Figure 7 shows the published Big*Bang board; the evocative background image depicts the cosmic microwave background of Stage 3 in Figure 4. This image is based on data from the satellites COBE (1992), WMAP (2003) and Planck (2013) \[9\]. It represents the oldest light in our universe, with darker (slightly cooler) areas corresponding to the concentration of matter due to fluctuations that gravity amplified to form the first galaxies.

![Figure 7. The published Big*Bang board.](image)

For the game board, I chose an ellipsoidal hexagonal grid to match the image. This left space in the corners of the board for simple player aids (Figure 6) and the three victory conditions.

5.3 Final Game

The final game has recently been released as Big*Bang; the asterisk in the title evokes the first explosion and the subsequent annihilation. The rules are given on the game’s official website and are summarised in the following blue box.

---

\[10\] The full Big*Bang rule book is available at: http://www.nestorgames.com/bigbang_detail.html
Big*Bang is played on an hexagonal grid (shown in Figure 7) with 84 circles, on which 21 each of protons, neutrons, antiprotons and antineutrons are randomly placed. The player owns the 42 particles and the antiplayer the 42 antiparticles.

Players alternate taking turns, where each turn involves either fusing or annihilating stacks, then performing a decay, as follows:

- **Fuse**: Move a stack along a straight line of empty cells to fuse on top of another.
- **Annihilate**: Move a stack along a straight line of empty cells to a corresponding anti-stack; both stacks are annihilated and put aside in a ‘light’ reserve.
- **Decay**: Decay an opponent’s unstable stack by replacing a neutron with a proton.

The game ends when a player cannot fuse or annihilate a stack. A victory token is then awarded for each of:

- Producing the most **light** (annihilated pieces).
- Building the most **helium**.
- Producing the most powerful **star** (linked group of own stacks containing the most hydrogen).

The game is won by the player with the majority of victory tokens.

### 5.4 The Link with Physics Laws

The subject of the article is about gamifying physics. With all the different compromises we have met in order to keep Big*Bang interesting as a game, have we lost the link with physics?

The huge matter-antimatter clash occurred in the first fractions of a second after the Big Bang, and only later did helium start forming, while in the game both processes occur simultaneously. Furthermore, our universe seems to consist mostly of matter only, while the game has equal amounts of matter and antimatter.

So is the game totally science fiction? Maybe not. We assume that only one of matter or antimatter could survive the initial annihilation, and, since we live in a matter world, we assume that only matter did. But what if the rapid expansion that followed the Big Bang pushed antimatter-dominated regions far enough away from matter-dominated ones?

In that case, annihilation would have halted due to the physical separation of both populations, and the universe would also contain antimatter galaxies. However, since the chemistry of antimatter is identical, those galaxies would look exactly like matter ones. Our only chance to spot them would be their collision with a matter galaxy, through the gigantic annihilation flash that would follow. In fact space missions are searching for the characteristic signals of such a clash, or for antinuclei produced in ‘antistars’ [10], but no evidence has been found yet. Leaving this hypothetical matter-antimatter coexistence aside, what about the other physics laws?

The spirit of the primordial nucleosynthesis is captured reasonably well. The first fusion step is $^2$H, the only stack of height 2. By forming $^2$H, players shield against annihilation by the more abundant individual pieces (while threatening the formation of the opponent’s $^2$H nearby) and prepare the way to helium. Depending on the annihilation rate chosen, neutrons start disappearing sooner or later, adding angst to the race since you may end up with proton-dominated regions that, without neutrons, will be doomed fusion-wise. However, the third victory condition gives sense to these hydrogen areas too, since they represent the future; stars that will live long enough to generate the ingredients of life.

These simplifications only allow the spirit to be captured, but this was the intended goal. Some add-ons could make the physics more explicit, but once a critical balance between complexity, playability and theme has been met, adding rules should be avoided. We can still propose variants, though, e.g. electric repulsion could be incorporated by allowing the fusion of adjacent stacks with protons, or the formation of carbon-12 could be introduced as an automatic victory condition if a player succeeds in linking three helium-4 stacks (Figure 8 shows one such stack).

**Figure 8.** A helium-4 stack with neighbouring neutron, antineutrons and antiproton.
We have seen several examples of games dealing within complex environments. Even when those with a physical phenomenon: Triplanetary, about physics case was too vast, letting us explore the physical laws is not enough to make a good game; per started with a question: why are (good) board patterns have already been identified and the requires the same; look for simple patterns within that takes us from the physics to the game. The allowed us to discuss in addition the process of a problem. In that respect, the case study closing the design loop.

As I said in the introduction, when we look into the unknown we look for simple patterns within complex environments. Even when those patterns have already been identified and the laws derived, making a game out of them requires the same; look for simple patterns within those laws. Which ones would make it too fiddly or play too small a role, and which ones are significant for the phenomenon we wish to highlight in our game? In Triplanetary, it was vector movement modified by a simplified gravitational law, acting at 1-hex distance only; in Gauss, the attraction and repulsion along the axes of a hexagonal grid; in Momentum, the transmission of momentum along the axes of an orthogonal grid.

The design process of a game about physics is a closed loop. First, we analyse the physics case; then, we translate a law into rules; then we use them to build the game; and finally, we check that the result successfully evokes the physics. The paper started with a question: why are (good) board games about physics so rare? Closely mimicking physical laws is not enough to make a good game; this is where mere simulations differ from games. For example, even though the pieces in Triplanetary give the impression of moving as real space ships would, the result is somewhat fiddly in its implementation. Gauss and Momentum, on the other hand, are examples in which the simulation and the game work well; these games evoke souvenirs from a science classroom when played, closing the design loop.

The previous examples showed us the solution of a problem. In that respect, the case study allowed us to discuss in addition the process that takes us from the physics to the game. The physics case was too vast, letting us explore the parts of it that exhibit ‘simple patterns’, the possible pieces and rules that would translate them, and the games that they would make. Sometimes the physical process itself is not well suited for an interesting game, sometimes clear rules lack a competitive and fun dimension. We discarded first a game about the ‘Race to Carbon’, then another one about a ‘Race to Helium’.

In the end, it was the introduction of antimatter that solved the playability issues – quite unexpectedly, to be honest – to make Big*Bang work well as a game. In this sense, the game design process mimics the scientific method: trying things that mostly lead to dead ends but that sometimes lead to a happy end. Answering our original question, maybe dead ends are more common in games about physics because the laws are what they are. We cannot tweak them beyond reality, as we may do with rules from other themes, and sometimes the laws do not lend themselves to make a game work, however they are implemented.

The previous examples showed us the solution of a problem. In that respect, the case study allowed us to discuss in addition the process that takes us from the physics to the game. The physics case was too vast, letting us explore the parts of it that exhibit ‘simple patterns’, the possible pieces and rules that would translate them, and the games that they would make. Sometimes the physical process itself is not well suited for an interesting game, sometimes clear rules lack a competitive and fun dimension. We discarded first a game about the ‘Race to Carbon’, then another one about a ‘Race to Helium’.

When trying to use laws of physics as game rules, the first consideration must be the purpose of the game. Teaching tools do not necessarily need to be interesting as games, but should demonstrate a real phenomenon to the audience with almost no need of rules. If the aim is an enjoyable game, on the other hand, then we must select the degree of simulation that we want it to model.

The balance between the realistic simulation of actual physics laws and playability is a difficult one to achieve. Even the most simple laws lead to a complex ensemble full of details, and the simulation of all those details should be left to computer and video games. Even when dealing with only a few simple laws, board games must make an additional abstraction effort. The challenge for the designer is pointing out first the most characteristic law, and then finding a rule that at the same time is simple, intuitive, and that lets players feel as if the game pieces actually obey that law.

This case study of Big*Bang also showed other important aspects of the process, such as the importance of defining a suitable frame for the laws to be modelled and the choice of mechanisms that can lead to an interesting game without betraying those laws. We have seen that, even in a subject as complex as the birth of our universe, it is possible to design for effect. But this is not usually the case. Maybe good games about physics are rare because the laws of physics are not made for interesting gameplay, but for making our world work.
Acknowledgements

Thanks to the reviewers for their suggestions, the editors for formatting and improving this piece, and to Gunther Schmidl for permission to use his image shown in Figure[1].

References


Miguel Marqués was born in Spain and works as a nuclear physicist at the French CNRS. He is interested in science, sports and history, and designs games based on these topics.

Address: LPC (ENSICAEN), 14050 Caen cedex, France. Email: marques@lpccaen.in2p3.fr

Gloop Challenges #5 and #6

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.

Gloop Challenge #5

Gloop Challenge #6
Make the Design do the Work

Cameron Browne, Queensland University of Technology (QUT)

Improving the clarity of games allows players to spend more of their mental effort on strategic planning rather than the mundane bookkeeping of calculating legal moves. This article discusses techniques for achieving this, by making the design do the work rather than the player, and demonstrates this concept through example. Such techniques include visual design, simplifying rules, clarifying rules, harnessing emergent strategies, and minimising mental bookkeeping.

1 Introduction

This issue’s game design pattern deals with the notion of making the design do the work rather than the player. The aim is to free the player from the mundane bookkeeping of move-making, so that they can focus on the more interesting task of deciding which moves to make. This paper outlines relevant principles, then demonstrates these in relation to numerous examples and counterexamples from well-known – and some lesser-known – games and puzzles.

1.1 Transparency of Rules

The related concept of embedding the rules of a game to improve its design is treated in an earlier paper [7]. While there is overlap between the two concepts, embedding the rules is actually a subset of the broader aim of making the design do the work, which can take other forms such as increasing the complexity of rule sets in order to benefit the player. Embedding the rules aims to minimise the number of rules that players must learn, while making the design do the work aims to minimise the mental effort that players must expend in order to play the game. This is the difference between the clarity of a rule set (form) and the clarity of moves in action and their implications (function) [19].

The assumption here is that the rules of a game should be as transparent as possible, so as not to distract players from strategic planning. We want the mechanisms of play to be as clear as possible so that players can see far down the game tree [5]. However, this is not true for all types of games; e.g. many war games are measured by the complexity and quality of the simulated battle experience rather than their strategic depth. War gamers may recognise this as the distinction between design for cause – focussing on the detail – and design for effect – abstracting away the detail in favour of higher-level control.

1.2 Design for Cause and Effect

This distinction between design for cause and design for effect was first described in the seminal 1978 article ‘Game Design: Art or Science’ [3] in relation to two popular board war games of the time, and the merits of each side have been debated ever since. Game designer Alan Emrich later defined these terms as follows:

Design for Cause: When a game’s design has players follow all of the logical steps and procedures to obtain an outcome, when players experience a methodology and must consider its many facets. This can often lead to systems that are over-engineered. That is, when the players are doing all the work and the designer is having all the fun.

Design for Effect: When a game abstracts complex procedures for simplicity’s sake so that the players can get straight to the ‘boom’. That is, when the designer does all the work so the players can have all the fun.

Both philosophies have their proponents, although I personally find the latter more compelling and believe that it has broader relevance to many more types of games and puzzles, so will focus on that approach here. This paper could just as well be called ‘Designing for Effect’.

1.3 Perceived Affordance

Design researcher Don Norman identifies three basic principles for the design of effective user controls [5]:

1. Visibility: It should be obvious what a control is used for.
2. Affordance: It should be obvious how a control is used.
3. Feedback: It should be obvious when a control has been used.

---

The relevance of these principles to video game design has been observed [6]. In the context of board games and puzzles, the second concept of affordance is probably the most useful. The concepts of visibility and feedback are less relevant, as game components tend to be clearly visible and it is generally obvious when they have been used, as they have moved, rotated, flipped, promoted, etc.

The term affordance comes from psychology [7] where it refers to the actionable properties between the world and an actor. These properties are fundamental to the actor’s nature and do not need to be visible, known or even desirable [3].

Norman points out that when it comes to design, we are really interested in the perceived affordance of objects, i.e. how the user understands and uses them rather than what is actually true. There is an obvious correlation between perceived affordance and making the design do the work in games; we want the player to intuitively know what actions are available in any given situation, rather than having to expend undue mental effort to derive this knowledge.

2 Visual Design

A game’s appearance is what shapes a player’s first impressions, before they learn its mechanisms and behaviour through play. This offers considerable scope for perceived affordance as the visual features of a game are typically its most obvious. This section gives examples of effective visual design.

2.1 Chess

The chequered 8 × 8 square board, shown in Figure 1 will be familiar to most readers as the Chess or Draughts board. But what happens if we replace this board with a plain lined grid?

Figure 2 shows the result of transposing this same Chess position to such a uniform grid. Functionally there is no difference; the geometries of the two boards are identical, the technical aspects of play are not affected, and the same rules of movement still apply.

However, the reader might find the position harder to interpret on the unadorned lined grid. If one studies each piece in isolation then its moves are still obvious, but the overall position takes longer to understand at a single glance.

This is more than just a side-effect of seeing familiar pieces transposed to an unfamiliar board. The alternating cell colours make it easier to see at a glance which pieces on the far side of the board are under diagonal attack, which are the 50% of cells potentially threatened by a single bishop, and which squares each knight can potentially move to (they will be squares of the other colour). The chequered board colouring is an example of a design feature that is independent of the game’s rules and exists only to help the player.

2.2 Vault

This principle is extended in the recent game Vault, by designer Néstor Romeral Andrés, in which pieces can either move to an adjacent cell or ‘vault’ over a pivot piece of the same colour to the symmetrically situated cell beyond it, possi-
bly to capture an enemy piece. Figure 3 shows a typical jump move by White.

Figure 3. A jump move in Vault.

Pieces may move to a neighbouring cell of either colour, but will always jump to a cell of the same colour. This reduces the mental burden on the player by making it easier to identify the correct target cell for long distance vaults, as only five out of the nine cells in the 3×3 landing window (i.e. the destination cell and its immediate neighbours) will be the jump colour.

Game designer Ken Shoda took this idea a step further to suggest the colouring scheme shown in Figure 4. Only one of the nine cells in the 3×3 landing window will be the jump colour using this design, making jump targets even easier to locate. Players can spend less time working out where their jumps will land and more time on planning.

Figure 4. A jump on the improved Vault board.

2.3 Heptalion

The domino-like game Heptalion, described in a previous paper [7], demonstrates how the visual design of pieces can help the player. Figure 5 shows a partially solved Heptalion challenge, in which the (single) player must place the tiles on the right to cover the pattern shown on the left, such that all symbols match. Two tiles have been played so far in this example.

Figure 5. A partially solved Heptalion puzzle.

Note that tiles are placed face-down and that their backs are blank. This is not coincidence or cost-saving on paint; it is an effective design feature to occlude completed parts of the pattern from view. The game was originally played by placing tiles onto the pattern face-up so that the symbols still showed, but this led to confusion between which tiles were covered and which were not, interfering with the solver’s planning. The simple insight to place tiles face-down led to a dramatic improvement for the player.

This case is unusual in that the design is improved by removing information from pieces rather than adding information to them. For comparison, see [7] for examples of effective design based on embedding visual cues or instructions in the equipment.

3 Simplifying Rules

Moving from the visual to the conceptual, careful design can also simplify a game’s rule set, to reduce the number of rules and the resulting cognitive burden on the player.

3.1 Pentalath

The board game Pentalath, invented by the computer program LUDI in 2007 [12], is played on the unusual hexagon-based board shown in Figure 6. Players take turns placing a piece of their colour with the aim of forming 5-in-a-row of their colour, while obeying the surround capture rule of Go.

Surround capture on a square grid suffers from the problem of mutually supporting patterns that create infinite cycles of play in which pieces are captured and recaptured indefinitely,
unless special \(ko\) or \(superko\) rules are invoked to stop such repetitions from occurring. However, such \(ko\) cycles do not occur on the hexagonal grid, as discussed in \([9]\) and shown in Figure \(6\), where it is obvious that a white piece played at cell \(x\) cannot immediately be recaptured by Black.

![Figure 6. No \(ko\) on the hexagonal grid.](image)

**LUDI** had inadvertently (re)discovered this useful feature of the hexagonal grid through trial and error, allowing it to present a simplified rule set derived from Go that did not involve any form of \(ko\) rule. This made move planning easier for players, as it removed the need to constantly check for repetitions of previous board states. This feature was apparently known amongst Go players who had experimented with hexagonal versions of the game, but had not previously been documented in the literature to my knowledge.

### 3.2 XOXO

XOXO, shown in Figure \(7\), is a tile placement puzzle from Smart Games designed by Raf Peeters.\(^4\) Players are given a number of challenges, each showing a number of preplaced hint pieces, and must pack the remaining pieces into the rectangular grid.

![Figure 7. An XOXO challenge in progress.](image)

This style of puzzle has a long history and will be well known to most readers, but XOXO features a couple of interesting innovations. Each piece is a \(polyomino\) made from five connected squares, each with a prominent O ring on one side and X-shaped protrusion on the other, in an alternating pattern. The board has prominent bumps on alternating cells – as can be seen in the figure – over which the O rings fit snugly but the X protrusions cannot be placed. Pieces can therefore only be placed on the board in precise orientations and locations, to make an alternating ‘XOXO...’ pattern, and pieces can not accidentally be placed at an angle.

This design adds some interesting deductive aspects to the game to help players. First, pieces must be placed to fit not only the other pieces but also the bumps on the board, giving players extra information to work with. Second, the fact that each piece has an odd number of squares means that each piece has a different number of O and X shapes on each side. Players can therefore often deduce which side up the remaining pieces must be placed, based on the number of exposed bumps remaining on the board; if there are \(N\) bumps visible, then the remaining pieces must have \(N\) X shapes uppermost. These implicit constraints allow challenges to be constructed with very few predefined hint pieces but which are still fully deducible.

XOXO shows how clever design can bring with it a number of useful constraints implicit in the equipment, without the need to explicitly state additional rules and strategies.

### 3.3 Unico

Figure \(8\) shows a mockup of a smartphone puzzle game currently under development called Unico. The rules are decidedly simple:

*Swap and rotate tiles to create one path of a single colour.*

Figure \(9\) shows the last two moves in the solution of a Unico challenge. Tiles \(a\) and \(b\) and swapped (left), then tile \(c\) is rotated \(180^\circ\) to complete a path of a single colour.

At the lower (mechanical) level, the program makes things easier for the player by updating path segment colours as tiles are moved. Path segments moved to connect to one or more neighbouring segments are set to the same colour, and path segments whose connections are broken take on new distinct colours. Thus, every connected subpath always has its own distinct colour.

\(^4\)http://www.smartgames.eu/en/smartgames/iq-xoxo
At the higher (conceptual) level, colour coding connected subpaths in this way makes it easier for the player to understand game states than trying to follow the convoluted paths by shape alone. This makes the game more intuitive, as it is natural for the player to seek moves that create fewer colours, which create fewer (and longer) subpaths, which progresses towards the goal.

This is perceived affordance in action; the player need not understand that they are pursuing the implicit topological goal of completing a single connected path in order to play the game effectively. This saves the need to explicitly state this as a rule, simplifying things for the player.

Note that two of the cells in each state are ‘flat’ and do not have a tile drawn behind them. This is another visual cue to the player that these are fixed cells that cannot be swapped or rotated, without the need to explicitly state this in a rule.

3.4 Gloop

Sometimes, the design of a game can decide certain quantities or limits that might otherwise be based on arbitrary choices: How big should the board be? How many pieces should each player start with? Which tiling is best? And so on.

Consider the puzzle game Gloop by renowned Dutch designer Fred Horn. Figure 9 (left) shows a square tile with two equidistant points on each side, and Figure 9 (right) shows one way that paths can be drawn within the tile to connect different points such that no paths intersect. The Gloop set consists of 91 such tiles, representing all topologically unique ways that such paths can be drawn within a square tile (including the empty tile). The complete set of 91 tiles can be found in the official rule book.\(^5\)

A number of puzzles can be played with the Gloop tiles, the most interesting of which I find to be the task of creating patterns that form a single closed contour. For example, Figure 10 shows a 3 × 3 group of tiles that form a single closed contour. This task gets increasingly difficult for increasing numbers of tiles, and has in fact been proven impossible for 88 tiles or more.\(^{27}\)

---

\(^5\)http://nestorgames.com/rulebooks/GLOOP_EN.pdf
Even the size of the playing area was predefined for Horn. Most of his challenges involve placing all 91 tiles in a rectangle to satisfy certain criteria. Of the two discrete rectangles that can be constructed with 91 squares – 1×91 and 7×13 – a moment’s reflection should reveal that 1×91 is an unsuitable number (unless we allow open path ends) hence 7×13 is the standard size. This is a case of the design making things easier for the designer as much as for the player.

3.5 Xats

Xats pieces are the unusual shapes shown in Figures 11 and 12, formed by circles with up to six spikes radiating in each of the six hexagonal directions. They are produced as off-cuts from the hexagonal tile game Stax and have themselves been developed into their own game. The rules are given in [14] and on the Xats web site.

Figure 11. Which piece is white pointing at?

An inherent problem with Xats pieces is that they have a hexagonal basis but are difficult to line up in a hexagonal formation, even if placed adjacent to each other; the slightest misalignment or error in rotation leads to ambiguity. For example, the pieces shown in Figure 11 are placed in an implied hexagonal grid such that the white piece points to one of the blue pieces, but it is not clear which one. Such ambiguity can only confuse players and detract from the playing experience.

For this reason, designer Néstor Romeral Andrés abandoned the obvious 2D grid format – as used in the parent game Stax – and instead proposed using independent 3D stacks of pieces, as shown in Figure 12. Each turn, players choose a piece from their hand (top and bottom rows) to add to a stack in the shared play area (centre row), such that the piece’s spikes cover the spikes of existing pieces in that stack.

This not only side-steps the problem of ambiguous grid alignment, but also enforces accuracy in rotation, as players must align each piece with existing pieces in the stack. The game’s rules were simplified by exploiting the unique piece design to allow intuitive and error-free moves that are clear to players.

Figure 12. A Xats game in action.

However, this is not the only way in which the design was made to do the work in this game. A subsequent rule was added that allows players to discard one of their pieces and destroy a stack of the same height as the piece’s spike count, returning the stack pieces to their owners. This rule made games longer and more interesting, fit with the game’s general theme of recycling, and was carefully implemented to avoid cycles in play.

4 Clarifying Indicators

The previous Xats example demonstrated not only how the design can be used to simplify the rules, but conversely how additional rules can be added to clarify matters for the player. This section contains examples of the related notion of clarifying indicators, which are non-essential rules or design elements added to a game in order to improve the experience for players.

4.1 Catchup

In the game Catchup, by Nick Bentley, the number of pieces to be played each turn depends on both players’ largest group sizes [12]. This has the potential to distract players from the core task of move planning, if they must constantly monitor and recount group sizes just to determine the legal moves each turn.

For this reason, a scoring track was added around the board, as shown in Figure 13, on which players advance pieces of their colour to indicate their largest current group sizes. This enhancement is not essential to the game, but provides this key information to players at glance, simplifying the task of calculating legal moves.

Such scoring tracks are known as Kramerleisten, after Wolfgang Kramer, who used them in many of his designs.
4.2 Amazons

Sometimes explicitly restating an obvious rule can help clarify matters for the player. For example, in the game Amazons, by Argentinian designer Walter Zamkauskas, shown in Figure 14, players move one of their Amazons each turn like a queen in Chess, then fire an arrow from the new location in any of the eight cardinal directions to land as far as desired in its direction of travel, in another queenlike move. There is no capture and Amazons can not move onto any square already occupied by any other piece. The game is won by the last player able to move.

The first thing to note in the context of this discussion is that the firing of arrows creates walls that introduce an aspect of territorial connection and ensure that every game converges steadily to a resolution. The design brings to the game implicit mathematical complexity in a simple and intuitive manner.

The second thing to note is that the rules almost always include the clarification that the arrow can be fired back over the cell just vacated by its firer. This rule is superfluous – it is implied by the fact that the arrow can be fired in any of the eight cardinal directions – but it is still useful in removing any doubt in the player’s mind. If the player had to constantly deduce for themselves ‘yes, arrows can move back over the cell just vacated’ with every move then even that minor niggle has the potential to interfere with their move planning thought processes.

While I personally prefer elegant minimal rule sets, and avoid restating obvious rules where possible, this is warranted in cases where such clarifications could avoid potential misinterpretations by players. I have been surprised over the years by the number of ways in which players have found to misunderstand the rules of my own games, making me reevaluate my assumptions about what can and cannot be obviously deduced from a given set of rules. Clarifying such potential ambiguities is a simple way to make the rules do the work rather than the player.

4.3 Meaningful Names

German mathematician Ingo Althöfer suggests that even a carefully chosen name can make a game easier for players to understand and play, if it mentally prepares them for the concepts involved. For example, players might guess that the game Knight’s Tour involves moving a familiar knight piece on a tour of a chessboard, before even seeing the equipment and rules.

This preference for meaningful names applies to many well-known games. The traditional game Snakes and Ladders was released by Milton Bradley as Chutes and Ladders, making the principles involved even more obvious to players; you get what it says on the box. Players do not need reminding that the aim of the card game Five Hundred is to score 500 points. And there can be no doubt regarding the aim and general theme of Monopoly.

5 Emergent Strategies

While clarifying indicators explicitly help players, strategies that emerge from the play itself can implicitly help by providing clear plans of action to follow. Players are motivated to search for and follow such strategies for their own benefit rather than due to any expressed rules, and a game’s design will ideally provide ample scope for such implicit strategies to emerge naturally.
5.1 Omega

In the game Omega, each player’s score is given by the product of the sizes of each group of their colour [14]. For example, Figure 15 shows a four-player game on a size-5 board, in which the current scores are:

- **White**: \(1 \times 2 \times 2 \times 3 \times 4 = 48\)
- **Black**: \(1 \times 4 \times 7 = 28\)
- **Red**: \(1 \times 2 \times 4 \times 5 = 40\)
- **Blue**: \(1 \times 2 \times 3 \times 6 = 36\)

Omega originally suffered from a lack of clarity, as the need to multiply lists of group sizes after each move imposed a computational burden on the players that made move planning difficult, especially on larger boards. This led to unsatisfactory games in which players made moves without fully understanding their implications, and often nobody would know who was winning until the final score calculation at the game’s end.

These problems were neatly solved following an observation by designer Néstor Romeral Andrés that the optimal group size for maximising a player’s score is 3 [14]. This led to a simple strategy that suddenly threw the game wide open and introduced subtle tactics of connection and disconnection, as players sought to make their own groups of size 3 while preventing their opponents from doing the same. The design produced an emergent strategy that saved the game.

5.2 Slitherlink

Solution strategies are essential for logic puzzles and elevate them above mere exercises in combinatorial exhaustion. For example, Figure 16 shows one of the local solution patterns useful for the deduction puzzle Slitherlink [6], in which a simple closed path must be traced through orthogonal vertices of a square grid to visit the number of sides indicated on each numbered cell.

Slitherlink players will soon learn that such adjacent 03 pairs always produce the pattern of cuts and edges shown, which can be used as building blocks – along with other patterns – to aid solution for a more engaging playing experience. If the player must instead laboriously recreate the deductions that led to this pattern every time they see a 03 pair, then this will become tiresome very quickly. The designer should do the work of encoding such patterns in their challenges, so the player can have the more entertaining task of recognising and exploiting them.
5.3 Sokoban

Figure [17] shows the well-known puzzle game Sokoban, in which the player pushes boxes around a maze to cover target holes [5]. This is another example of how not only the design of the game, but the design of each level, can be made to work for the player’s benefit.

Sokoban’s premise and basic mechanism – pushing boxes – is as simple as it gets. However, clever level design can bring out surprising interactions that require some creativity to solve. For example, the solution shown in Figure [17] (right) starts with the box as close as possible to its target destination, but with the player on the wrong side of it; the box must be pushed as far away from its target as possible then back again.

The design of this challenge is efficient in making the box travel as far as possible and requiring the use of every cell, and is also rather perverse in making the player go to all that effort just to return the box to its starting position for the final push. This level design exploits the game’s characteristics to produce an engaging experience for the player.

6 Mental Bookkeeping

A common theme throughout this piece is to minimise the mental bookkeeping that players must perform. This section discusses ways that this can be achieved.

6.1 Lines of Action

Claude Soucie’s classic board game Lines of Action [17], shown in Figure [18] contains a number of movement rules:

- Pieces must move in an orthogonal or diagonal line a distance equal to the number of pieces in that line.
- Pieces can jump over friendly pieces but not land on them.
- Pieces can not jump over enemy pieces but can land on them (to capture).

This rule set has the potential to confuse. To determine the available moves, the mover must count the number of pieces along every line through every friendly pieces, then remember whether friendly pieces can jump over enemy ones or whether it is the other way around. So why has this game increased in popularity to become a classic and be well respected and widely played today?

I believe that this is because the movement rules exhibit an internal logical consistency that collapses to reduce their true complexity to almost nothing. Since pieces are only counted along the line of travel in each direction, then players can count them at the same time as evaluating potential landing sites along those lines with little additional mental work; with practice, counting pieces and determining landing sites becomes part of the same habitual action. To see this, consider the opening moves shown in Figure [18] and how it becomes obvious that there are two pieces along each potential direction of travel.

Further, it is standard in board games for pieces to land on enemy pieces to capture them, which gives the player a reference point for mentally consolidating the complementary jump and landing rules; you can kill an enemy but not a friend, so can jump over a friend but not an enemy. The movement rules thus reduce to two basic mental checks for players to perform. The game also benefits from the chequered board design, which makes diagonal lines easier to follow.

6.2 Ultima

Robert Abbott’s game Epaminondas [5] introduces a similar type of linear piece movement, in which phalanxes – consecutive lines of friendly pieces – move along their line as a unit, up to a number of squares equal to their length. Figure [19] shows the game in its starting position.

As is the case with Lines of Action, moves in Epaminondas are easy to visualise as the distance count and direction calculations occur along the same axis and can be done at a glance. In fact, moves in Epaminondas are easier for players to visualise than moves in Lines of Action, as short lines of consecutive pieces are easier to count in a glance than counting all pieces along a given axis. Epaminondas is particularly interesting for the
purposes of this discussion, as it was invented by Abbott in direct response to a perceived lack of clarity in his earlier game Ultima [17].

Ultima is a Chess variant in which the pieces have unusual and reasonably complex behaviours, which players found confusing:

... the resulting game is so complex that it is difficult to see more than one or two moves ahead, and too many pieces are captured simply by surprise attack. [5, p. 85]

Attempts were made to simplify the game for players by illustrating each piece with its movement details to ‘embed the rules’, as described in [7], but without much success. Ultima has since been disowned by Abbott, who found a solution to the problem of clarity by using uniform pieces for each player and instead using local relationships among pieces to define their movement in Epaminondas.

This further illustrates the difference between ‘embedding the rules’ and ‘making the design do the work’. Complicating the pieces with explicit movement instructions did not help as much as a complete redesign that simplified the pieces down to uniform stones; embedding the rules still made the player do the work in this case.

6.3 Quantum Leap

The recently invented game of Quantum Leap [10] is played on the hexagonal grid shown in Figure 20. Players take turns jumping one of their pieces along any of the six diagonal axes, by a distance equal to its number of adjacent friendly pieces, to land on an enemy piece (which is captured). The movement of pieces is dictated by their relationships with other pieces, as in Lines of Action and Epaminondas. However, there is a key difference; in these previous games, the distance and direction are both determined along the same axis (line of travel), whereas in Quantum Leap they are determined on independent axes (adjacent neighbourhood and line of travel). This added layer of complexity can make moves harder to visualise, as move processing involves a two-step process for each piece: 1) count its friendly neighbours, then 2) check this distance along each axis.

To get a feel for this process, identify the legal moves from the white piece marked ‘?’ in Figure 20.[11] This is not an overly taxing task for a single piece, but does require some concentration to get right on the somewhat confusing hexagonal grid. But consider the mental effort required to apply this process to all white pieces, and to visualise future moves from the resulting board states, and one starts to appreciate the difficulty of planning ahead in this game.

To address this problem, a variant was tested in which pieces were instead pushed a distance equal to the number of consecutive friendly pieces lined up behind them along the direction of travel. This placed the distance and direction calculations along the same axis to make moves easier for the player to calculate, but unfortunately led to a more restricted game; edge pieces were restricted to the edges and corner pieces had no possible moves.

11There are three.
6.4 Iqishiqi

The game of Iqishiqi, designed by Jo˜ao Pedro Neto and Bill Taylor, uses another push mechanism based on group connectivity. Players take turns placing a piece of the same colour, to expand a connected group of pieces that is in line-of-sight of a single neutral piece of another colour along any of the six cardinal directions. That neutral piece is then pushed along that line a number of cells equal to the group size; the mover chooses which direction if there is more than one line-of-sight.

For example, Figure 3 shows a board state in which the mover has just placed the piece marked ‘+’ to make a group of size 4, to push the neutral red piece four cells in either of the directions shown. If White is making the move, they will choose the upwards line that pushes the neutral piece to the white board edge, to win the game.

This movement mechanism is close in spirit to the Quantum Leap mechanism but is less confusing to apply:

• Moves are limited to those cells adjacent to groups in line-of-sight of the neutral piece, rather than all friendly pieces.
• Uniform group size is easier to calculate than differentiated neighbour counts.
• Sharing a single piece colour is simpler than differentiated colours in general.

These factors are subtle – and debatable – but combine to reduce the mental workload when identifying legal moves.

7 Counterexamples

A good way to highlight the benefit of making the design do the work is to study some counterexamples in which this principle is violated.

7.1 Quoridor

In the game Quoridor, players aim to move their pawn to the far side of a square 9×9 board, taking turns either moving their pawn a step or adding one of 20 wall segments to block the opponent. The game contains a special rule: players can not place a wall segment that would cut off all of the opponent’s remaining paths to victory.

This special rule presumably provides a more balanced game, prolonging the contest and avoiding trivial wins in which players cut the opponent off prematurely, and avoids the undesirable situation of one player having no chance of winning while the opponent plays out their remaining moves. However, it also adds strategic depth to the game.

Consider the situation shown in Figure 22 with White to play and win. White has exactly one winning move: placing a wall at a allows a win in three moves. If White plays anywhere else, then Black can place a wall at b to force White to backtrack through a much longer path, safe in the knowledge that the Black piece can not be blocked and will win in three moves.
The special ‘must allow an enemy path’ rule is an example of making the player do the work, as it is an additional rule that players must remember and apply when planning moves. However, visualising paths is an intuitive task that the human brain is adept at – except for particularly complex situations – and this additional rule does not represent an undue amount of mental calculation in practice. The advantages of including this rule obviously outweighed the disadvantages, and it is in fact one of the most interesting aspects of the game.

### 7.2 Red

In the game of Red, players add square pieces in two different sizes, with differently coloured backgrounds and foreground shapes. Players aim to connect groups of tiles with their background colour and with their foreground colour.

Red also contains a special rule: tiles can only be placed if they are adjacent to at least one existing small tile. This rule is necessary to stop players ‘hiding’ their pieces behind larger tiles for easy points. For example, the move shown in Figure 23 (left) cannot be played, as the tile being placed would not lie adjacent to any small tiles.

![Figure 23. Red problem and a possible solution.](image)

This special rule is another example of making the player do the work. However, there is an additional problem, as the rule is not intuitive and players keep forgetting it. This has necessitated revising the Red rule book to highlight this special rule, but this non-intuitive rule still troubles players and affects the game’s otherwise excellent clarity.

Is there a simple solution to this problem? Figure 23 (right) shows one option; marking small tiles with a ‘+’ sign as a visual cue that these are tiles that can be added to. However, this would detract from the visual clarity of the game, which is at odds with its designer’s aesthetic vision.

### 7.3 Rithmomachy

I conclude with a notable case for making the design do the work. There has recently been something of a revival in historical board games… but not this one! There is a reason why most readers will not have heard of Rithmomachy.

![Figure 24. The game of Rithmomachy.](image)

The mere sight of the imposing board and starting configuration of pieces shown in Figure 24 would be enough to give pause to even the most hardcore gamer. To quote noted game historian H. J. R. Murray [19, p. 227]:

*It was the dullness and difficulty of this game that killed it.*

To appreciate the extent of the problem, consider the following fragments from just a few of the game’s various capture rules [19]:

---

Addition: Two or more men can both be played by a legal move to a cell occupied by an enemy man, and the sum of the numbers on these men equals the number on the enemy man...

Multiplication: A man numbered \( n \) is \( x \) cells distant along an orthogonal or diagonal direction from an enemy man with number \( x \times n \) and no man intervenes between them...

Subtraction: Two men can be played by a legal move to a cell occupied by an enemy man, and the difference of their numbers equals the number of the enemy man...

Division: A man numbered \( n \) is \( x \) cells distant along an orthogonal or diagonal direction from an enemy man with number \( x / n \)...

Every turn would have been an exercise in laborious mental number crunching just to determine which moves were allowed, resulting in perhaps the worst clarity of any known abstract board game. Planning ahead even a move or two would have been beyond the skill of all but the most gifted and dedicated of players.

This game was evidently designed as a tool for teaching elementary mathematics to players, with little regard for notions such as elegance, clarity or fun. The design made the player do the work – and a lot of work – every turn.

Murray describes Rithmomachy as a ‘highly artificial arithmetical game’ and the only known game of the Middle Ages to have perished entirely, despite being played for hundreds of years. However, Ingo Althöfer points out that this is no longer strictly true, as a small number of players are now interested in the game from a historical perspective. Modern sets are typically accompanied with cards showing the important numerical relationships, to ease the burden on the player. The game would need to be redesigned for a modern audience – with a smaller board and simpler rules – if it were widely released today.

8 Computational Approach

Computer scientist Spyridon Samothrakis points out that another way of viewing this concept of making the design do the work, or designing for effect, is to understand it as designing games that provide features that are easy for the human brain to interpret. This is related to the brain’s capacity for chunking raw information into more useful higher-level conceptual units. If the player is able to mentally chunk trivial aspects of a game into more meaningful features, which can then be used for strategic planning, then this should make the game easier to play and more interesting for the player.

Samothrakis suggests that such an approach could lead to formal notions of design for effect, and proposes the following categories of features as a starting point:

- **Reward**: Does this action score a point?
- **Transition**: How will this action affect the game state?
- **State**: Given a state, where am I in the game?
- **Action**: What actions are available?
- **Static value**: How good is this state?
- **Dynamic value**: Will this action improve my position?

9 Conclusion

I hope that I have demonstrated through this discussion, and the related examples, the importance of making the design do the work rather than the player, especially for strategy games. This is more a mindset than a practical procedure as there are no hard and fast guidelines: sometimes rules need to be removed and sometimes added; sometimes equipment needs to be simplified and sometimes embellished, and so on. The designer is removing themself from the equation and letting the design speak for itself.

The main purpose of this piece is to raise awareness of such issues in the designer’s mind. Making the design do the work – rather than the player – has many benefits. It allows more elegant games that are easier to play, in which players can focus on the strategic battle at hand rather than the inner mental battle of simply interpreting and applying the game’s rules.

Acknowledgements

Thanks to Russ Williams, Richard Reilly and the reviewers for their helpful suggestions, and to John Farrell for pointing out the relevance of affordance in this context. The Chess board images for Figures 1 and 2 were created using the Apronus Online Interactive Chessboard. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.
References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.
Email: c.browne@qut.edu.au

Gloop Challenge #7

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.
How to Make a Better $3 \times 3 \times 3 \times 3$

Carl Hoff, Applied Materials

A Rubik’s Cube is a $3 \times 3 \times 3$ twisty puzzle. This puzzle can conceptually be pictured as an array of 27 cubies. However the standard Rubik’s Cube only has 26 of these cubies stickered. Not all of the 26 stickered cubies have a fixed position and orientation in the solved state. Similar limitations are also true for the standard model of the $3 \times 3 \times 3 \times 3$. This paper discusses the methods used to overcome these limitations of the standard Rubik’s Cube, the advantages and disadvantages of these methods, and how to apply one of the methods to the $3 \times 3 \times 3 \times 3$.

1 Introduction

In 1974, Ernő Rubik invented the $3 \times 3 \times 3$ twisty puzzle. Ideal Toys produced and marketed his creation as the Rubik’s Cube in 1980, and it took the world by storm. Not only were kids and adults fascinated by the puzzle, but it also caught the eye of mathematicians who saw the puzzle as an excellent representation of an algebraic group. The immense popularity of the Rubik’s Cube sparked the creation of many variations: the $4 \times 4 \times 4$, the $5 \times 5 \times 5$, and the application of the basic principles to tetrahedral, dodecahedral, and other geometries as the population clamoured for more.

But mathematicians did not stop there; they saw the Rubik’s Cube as a subject of mathematical investigation. Books were written that used the Rubik’s Cube as an introduction to group theory. Other mathematicians went on to consider Rubik’s Cube variations which could not be placed on store shelves. The $3 \times 3 \times 3 \times 3$ was independently studied by Dan Velleman, H. R. Kamack and T. R. Keane, and Joe Buhler, Brad Jackson and Dave Sibley. While two of these papers are officially unpublished, they are currently available online and were directly available from the authors at the time they were written. So they are heavily referenced in other works. A prime example is the July 1982 column of ‘Metamagical Themas’ in Scientific American, which was likely the first public exposure of the $3 \times 3 \times 3 \times 3$ in a mainstream publication.

The advent of 3D printing has led to a resurgence in the innovation of twisty puzzles. New puzzles and new designs appear weekly and are discussed in online forums. But many of these new innovations have not yet been carried over into the applications that allow one to play with the higher dimensional puzzles like the $3 \times 3 \times 3 \times 3$. For example, there are many methods used to turn a normal $3 \times 3 \times 3$, or Rubik’s Cube, into a Super Multi $3 \times 3 \times 3$. In this context, super means that all stickered pieces of the puzzle have been given a fixed position and orientation in the solved state and multi means that all volumes created by the cut planes which define the puzzle are included as stickered pieces in the puzzle. The normal $3 \times 3 \times 3$ fails in both of these areas as the face centres are stickered but the four orientations that they can reach are indistinguishable and the 27th or core cubie is not stickered at all. This paper will examine several of the common ways used to overcome these limitations and show how one of the most innovative methods can be extended to the $3 \times 3 \times 3 \times 3$.

Figure 1. Line, square, cube and hypercube.

---

1http://www.twistypuzzles.com/
2http://twistypedia.oskarvandeventer.nl/index.php/Super-(puzzle_name)
3http://twistypedia.oskarvandeventer.nl/index.php/Multi-(puzzle_name)

2 \( n \)-Cubes

Before getting too far into the discussion of the \( 3 \times 3 \times 3 \times 3 \), let us review some basics about \( n \)-dimensional cubes. A 0-dimensional cube is a simple point. To create a 1-dimensional cube, we sweep the point along some unit distance forming a line segment. To create a 2-dimensional cube, we sweep this line in a perpendicular direction forming a square. If we take this square and again sweep it along a direction perpendicular to its surface, we form a 3-dimensional cube. This is the same process one follows for creating a 4-dimensional cube, or hypercube: one sweeps a point along some unit distance forming a line. To create a 2-dimensional cube, we sweep this line in a perpendicular direction to its surface, we form a 3-dimensional cube. This is the same process one follows for creating a 4-dimensional cube, or hypercube: one sweeps a point along some unit distance forming a line. These steps are shown in Figure 1.

Table 1 summarises some of the terminology commonly associated with these \( n \)-dimensional objects. Note that the boundary which terminates the extent of the \( n \)-cube along a given cardinal direction is an \((n-1)\)-dimensional cube. So these boundaries occur in pairs. A line segment ends in two points, one at the line’s maximum extent and the other at the line’s minimum extent along the single dimension. Similarly, a square has four sides (e.g. top/bottom and left/right). A cube has six faces, and a hypercube has eight cells. The number of translational and rotational degrees of freedom (DoF) and the centre of rotation are also shown for each \( n \)-cube.

### 3 Cutting and Rotating \( n \)-Cubes

Since we will be cutting up these \( n \)-cubes to create twisty puzzles, we also need to discuss what they can be cut with and how they rotate. An \( n \)-cube can be cut with an \((n-1)\)-dimensional cube, but more generally it can be cut with any \((n-1)\)-dimensional surface. In 3D, these 2-dimensional objects are called planes and surfaces. In 4D, these correspond to the following 3D objects: hyperplanes and hypersurfaces. The notion of 4D rotation is one that many find difficult to grasp, as it involves rotation about a plane. So let us step back to 2D.

3.1 Two Dimensions

In 2D, a square just has three degrees of freedom. This is the number of parameters needed to give it a unique position and orientation in two dimensions. Those three parameters are two translations (an \( x \) and a \( y \) displacement from the origin) and an angle specifying its orientation relative to one of the cardinal directions. This rotation angle specifies a rotation about a point, a zero-dimensional object.

Along a 2-dimensional line you can have only translation, not rotation. The notion of rotation about a line would be meaningless to someone confined to a 2-dimensional world.

3.2 Three Dimensions

In 3D, a cube has six degrees of freedom. These are three translations, needed to specify the displacement in the \( x \), \( y \), and \( z \) directions, and three angles of rotation. In 3D, these three angles are called yaw, pitch, and roll, and specify rotation about one of the three cardinal directions.

In moving from two to three dimensions, the centre of rotation went from a 0-dimensional point to a 1-dimensional axis. Similarly, another dimension must be added to the centre of rotation when going to four dimensions.

3.3 Four Dimensions

In 4D, a hypercube has ten degrees of freedom. Four are the translations or displacements along each of the four cardinal directions. The other six are the rotations about the \( xy \), \( xz \), \( xw \), \( yz \), \(yw \), and \( zw \) planes.

To develop a mental picture of each of these rotations, let us go back and look at each of the \( n \)-cubes. The line in 1-dimensional space cannot be rotated so we will start with \( n = 2 \). A square has a single invariant centre of rotation. This is the point at the square’s centre, which does not touch any of the square’s four boundaries. So a 90° rotation results in the four sides orbiting about this centre of rotation.

In 3D, a cube allows invariant 90° rotations about three axes. Note that each axis passes
through the centres of two of the cube’s faces and does not touch the other four faces. Rotation about any one of these axes causes the two faces touching the axis to simply rotate in place. The other four faces orbit the centre of rotation, just as the sides of the square do in 2D.

In 4D, the hypercube has six centres of rotation allowing invariant 90° rotations. Each of these six centres of rotation is a plane which passes through four of the hypercube’s eight cells. These four cells simply rotate in place, just as the two squares which are intersected by the axis of rotation in the cube. The other four cells rotate in an orbit about the centre of rotation, as just seen with the sides of a 2D square, or with four of a 3D cube’s six faces. Union College hosts some very nice animations illustrating these rotations.

### 3.4 Unfolding Cubes and Hypercubes

The applications that allow one to play with a $3 \times 3 \times 3 \times 3$ typically display an *unfolding* of the hypercube into 3D space. So we should also review what the rotation of a hypercube about each of these six orthogonal planes looks like in this unfolded projection. These rotations will be used to perform global rotations of the entire puzzle; they play no part in the actual scrambling of the puzzle. Before we look at the global rotations of the unfolded hypercube, we will look at the three basic rotations of an unfolded cube in 2D space as seen in Figure 2 as follows:

- (a) An unfolded cube centred on the origin.
- (b) Rotation about the $x$-axis.
- (c) Rotation about the $y$-axis.
- (d) Rotation about the $z$-axis.

![Figure 2. Rotations on an unfolded cube.](http://www.math.union.edu/~dpvc/math/4d/folding/)

In all three cases, the squares which are intersected by the axis of rotation simply rotate in place about their centres. For these projections, we have chosen to unfold the cube into the $xy$ plane. The rotations about the $x$ and $y$ axes may appear odd, since they show an apparent translation of four squares in which a square pushed off one side of the image reappears on the other side.

This complexity is introduced because the $x$ and $y$ axes lie in the $xy$ plane, and in 2D it is impossible to rotate about a line. This complexity is avoided in the projection of the $z$-axis rotation. There the axis is perpendicular to the $xy$ plane, so it is reduced to a single point. In Figure 2(d), we see five of the squares are exhibiting normal 2D rotation about this point. In fact, all six squares could be shown as rotating about this point and the representation would still be valid.

Figure 3 shows an unfolded hypercube, adding a $w$-axis to the same $xy$ plane used previously, as follows:

- (a) A hypercube unfolded into the 3D space $\{x, y, w\}$.
- (b) Rotation about the $xy$ plane.
- (c) Rotation about the $xz$ plane.
- (d) Rotation about the $xw$ plane.
- (e) Rotation about the $yz$ plane.
- (f) Rotation about the $yw$ plane.
- (g) Rotation about the $zw$ plane.

Notice that any plane containing the $z$-axis is perpendicular to this 3D space, so the projections of the $xz$, $yz$, and $zw$ planes are just the $x$, $y$, and $w$ axes respectively. This results in rotations about these planes appearing as normal 3D rotations. The other three rotation planes, the $xy$, $xw$, and $yw$ planes, all suffer from the same complexity encountered with the 2D projections of the cube. Namely, we get a circuit of four cubes that appear to exit one side and reappear on the other, while the four cubes that are intersected by the plane of rotation simply rotate in place.

4http://www.math.union.edu/~dpvc/math/4d/rotation/
5http://www.math.union.edu/~dpvc/math/4d/folding/
The $3 \times 3 \times 3 \times 3$ is created by taking the hypercube and cutting it with two equally spaced hyperplanes along each of the four cardinal directions. These hyperplanes cut the original hypercube into 81 4D hypercubies, just as the planar cuts on the Rubik’s Cube divide it into 27 3D cubies.

The standard $3 \times 3 \times 3 \times 3$ representation is typically created by first looking at the nature of the $3 \times 3 \times 3$ and then extrapolating it into the $4^{th}$ dimension. The $3 \times 3 \times 3$ is a cube comprised of six faces. On each of these faces is a $3 \times 3$ array of stickers. In the solved state, all stickers on a given face are a single colour, as shown in Figure 4.

For this discussion, we define the positive $x$-direction as Right and denote this face with red stickers. The negative $x$-direction is Left, and it is denoted with orange stickers. The positive $y$-direction is Forward, and it is denoted with blue stickers. The negative $y$-direction is Backward, and it is denoted with green stickers. The positive $z$-direction is Up, and it is denoted with yellow stickers. And finally, the negative $z$-direction is Down, and it is denoted with white stickers.

<table>
<thead>
<tr>
<th>Axis</th>
<th>Direction</th>
<th>Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+x$</td>
<td>Right</td>
<td>Red</td>
</tr>
<tr>
<td>$-x$</td>
<td>Left</td>
<td>Orange</td>
</tr>
<tr>
<td>$+y$</td>
<td>Forward</td>
<td>Blue</td>
</tr>
<tr>
<td>$-y$</td>
<td>Backward</td>
<td>Green</td>
</tr>
<tr>
<td>$+z$</td>
<td>Up</td>
<td>Yellow</td>
</tr>
<tr>
<td>$-z$</td>
<td>Down</td>
<td>White</td>
</tr>
</tbody>
</table>

The faces and the stickers on a standard Rubik’s Cube are all squares, i.e. 2-dimensional objects. Since the six faces of a cube can be unfolded to create a 2-dimensional object, we can do the same for the Rubik’s Cube. Figure 5 shows this 2-dimensional representation, and Figure 6 shows what turning the Up face looks like in this representation.
4.1 Moving to 3D

With this 2D representation of the Rubik’s Cube in mind, we can now envision the 3D representation of the $3 \times 3 \times 3 \times 3$. In four dimensions, this object is a hypercube or tesseract. Just as the 2D surface of a 3D cube is made of six 2D squares which can be unfolded, the 3D surface of the 4D tesseract is made of eight 3D cubes which can be unfolded into 3D space. So to extrapolate our 2D representation of a Rubik’s Cube to the 3D representation of the $3 \times 3 \times 3 \times 3$, our $3 \times 3$ array of 2D stickers on each of six faces becomes a $3 \times 3 \times 3$ array of 3D stickers in each of the eight cells of the unfolded tesseract, as shown in Figure 7.

With the addition of a fourth spatial dimension, we need to define two additional directions. The positive $w$-direction is named ana, and it is denoted with cyan stickers. The negative $w$-direction is named kata, and it is denoted with magenta stickers. The names ‘ana’, ‘kata’ and ‘tesseract’ were coined by Charles Howard Hinton in his 1888 book *A New Era of Thought* [8].

In Figure 5, the white (Down) face is drawn in front of the blue (Forward) face. This face could also be drawn to the right of the red (Right) face, the left of the orange (Left) face, or behind the green (Backward) face. The same is true for the white (Down) cell seen in Figure 7. It could be drawn next to the red, green, orange, cyan, and magenta cells as well, as long as it is on the far side from the yellow cell. For this reason, the eighth cell is typically left out of the field of view in most $3 \times 3 \times 3 \times 3$ simulators. When performing a global rotation of the entire $3 \times 3 \times 3 \times 3$ which replaces the cell in the centre, then the eighth cell is pulled into view while another is then pushed.

Figure 5. A 2D representation of a Rubik’s Cube.

Each face cubie is represented by one sticker (square), each edge cubie by two differently coloured stickers, and each corner cubie by three differently coloured stickers. The centre cubie is not shown. Any of the six faces can be moved to the Up face location with a global rotation of the puzzle, making Figure 6 completely general.

Figure 6. A partial rotation of the Up face.

Figure 7. A 3D representation of a $3 \times 3 \times 3 \times 3$.
out of view into the eighth cell’s old location. Figure 8 shows the 3D representation with a partial rotation of the Up cell.

Figure 8. A partial rotation of the Up cell.

An excellent video introduction to the $3 \times 3 \times 3 \times 3$ by user ‘Mathologer’ can be found on YouTube, which is what prompted me to explore the $3 \times 3 \times 3 \times 3$. The most readily available application which allows one to play with the $3 \times 3 \times 3 \times 3$ is MAGIC CUBE 4D. This puzzle was created by Don Hatch and Melinda Green with development starting back in 1988. Jay Berkenbilt and Roice Nelson have also made major contributions. The best way to gain familiarity with the $3 \times 3 \times 3 \times 3$ is to play with this application for a while. There is also a Yahoo Group specifically dedicated to discussions about 4D Cubing. This group’s files section has a copy of the paper by Buhler, Jackson, and Sibley.

5 Limitations of the Model

To understand where the standard model of the $3 \times 3 \times 3 \times 3$ falls short, we need to first look at the standard Rubik’s Cube. The Rubik’s Cube represents a $3 \times 3 \times 3$ array of cubies that are stickered and can be scrambled with the goal of restoring them to their solved positions. However, an examination of the Rubik’s Cube reveals that only 26 cubies are stickered, of which six are only stickered on a single face.

The edges with two stickers and the corners with three stickers each have a given position and orientation in the solved state, but these only account for 20 of the 27 cubies. The six face centres, with only a single sticker, have nothing to mark their orientation, and there are four possible orientations in which they could exist. There is also the central cubie, or core, that is not stickered at all. In this case, the core’s orientation is known, as it is fixed relative to the face centres with screws. But in the general case, e.g. the $4 \times 4 \times 4$ or higher, there is new information that could be revealed if the interior cubies could be stickered.

The 3D representation of the $3 \times 3 \times 3 \times 3$ in Figure 7 shows 3D stickers. This is important: the $3 \times 3 \times 3 \times 3$ contains 81 hypercubies, just as the $3 \times 3 \times 3$ contains 27 cubies. And as with the standard Rubik’s Cube, these stickers appear on the various component hypercubies in different quantities, depending on their position. In the Rubik’s Cube, there are four types of cubies: one core with no stickers, six face centres with one sticker, twelve edge cubies with two stickers, and eight corner cubies with three stickers. In the $3 \times 3 \times 3 \times 3$ there are five types of hypercubies. By analysing how the puzzle twists, it is possible to determine which stickers belong to each of the hypercubies. Each cell is composed of a $3 \times 3 \times 3$ array of 3D stickers. The cell centre sticker never leaves the cell it starts in; it just rotates in place and can be in any one of 24 orientations. These cell centre stickers are the only sticker on their hypercubie, and there are eight of them, one in each cell. We will call these the eight cell centre hypercubies.

Now consider a sticker in a cell face position. This sticker is paired with the adjacent face of the neighboring cell and these two stickers are on the same hypercubie as they move in pairs. So the two stickers always move together with their hypercubie, analogous to the two square stickers on an edge cubie of a standard Rubik’s Cube. We call these hypercubies the cell faces, and there are 24 of them. They are similar to the edge cubies found in a normal $3 \times 3 \times 3$, which also have two stickers on adjacent faces.

Now consider a sticker in a cell edge position. This sticker is exposed on two faces, and it is paired with the stickers just opposite it across both of those faces. These hypercubies therefore have three stickers; we call these the cell edges, and there are 32 of them.

This leaves only the cell corner position sticker to explore. This sticker is exposed on three faces and it is paired with the sticker just opposite it across all three faces. These hypercubies have four stickers; we call these cell corners, and there are 16 of them.

Summing these hypercubies gives eight cell

---

6https://youtu.be/yhPH1369OWc
7http://superliminal.com/cube/cube.htm
8https://groups.yahoo.com/neo/groups/4D_Cubing/info
centres, 24 cell faces, 32 cell edges, and 16 cell corners, for a total of 80 hypercubes. However, \(3 \times 3 \times 3 \times 3 = 81\), so one must be missing. All the hypercubes we have counted so far are represented in the unfolded 3D surface of the tesseract. Just as in a Rubik’s Cube, there is a core hypercube that has no surface exposure and is currently unstickered. Table 3 summarises this enumeration of the \(3 \times 3 \times 3\) cubies and \(3 \times 3 \times 3 \times 3\) hypercubies.

<table>
<thead>
<tr>
<th>Name</th>
<th># Stickers</th>
<th>Name</th>
<th># Stickers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>1</td>
<td>Core</td>
<td>1</td>
</tr>
<tr>
<td>Face</td>
<td>6</td>
<td>Cell Centre</td>
<td>8</td>
</tr>
<tr>
<td>Edge</td>
<td>12</td>
<td>Cell Face</td>
<td>24</td>
</tr>
<tr>
<td>Corner</td>
<td>8</td>
<td>Cell Edge</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cell Corner</td>
<td>16</td>
</tr>
<tr>
<td>Total:</td>
<td>27</td>
<td></td>
<td>81</td>
</tr>
</tbody>
</table>

Table 3. Enumeration of cubies and hypercubies.

Just as with the Rubik’s Cube, if we describe the \(3 \times 3 \times 3 \times 3\) as the puzzle which allows these 81 hypercubies to be scrambled with the goal of restoring them to their solved positions, we note a few limitations.

- The core hypercube is unstickered. True, we know its position and orientation due to its relationship with the cell centres. While this new information will add nothing to the permutation count of the \(3 \times 3 \times 3 \times 3\), it still would be nice to reveal the interior with a general method that can be used for larger puzzles like the \(4 \times 4 \times 4 \times 4\) and \(5 \times 5 \times 5 \times 5\) where the interior does contain new information.

- Each of the eight cell centres has only one sticker, and this sticker does not reveal any information about which of the 24 possible orientations it is in. If we can add visible cell centre orientation to the puzzle, then we increase the permutation count of the \(3 \times 3 \times 3 \times 3\).

- The cell faces, which have only two stickers, are also missing some orientation information. In addition to being flipped like the 2-sticker edges on a Rubik’s Cube, these cell faces can be rotated about the axis connecting the centres of these two stickers. So even when they are restored to their solved position, there are eight possible orientations they could be in. But the stickers only reveal whether or not the hypercube is flipped. Even in the non-flipped state, the hypercube could appear solved but be in any one of the four states allowed by rotation about the axis connecting the sticker centres.

6 Overcoming the Limitations

To overcome these limitations, it is best to review several approaches to overcoming the limitations on a standard \(3 \times 3 \times 3\).

6.1 Exploded Views

One approach is to work with an exploded view of the 27 cubies, shown in Figure 9. Here all 27 cubies are seen, and all six faces of each cubie are exposed and stickerable. This view provides access to all the desired information and has been used in some applications. CUBIXPLAYER2 created by Per Kristen Fredlund is one such application. The main disadvantage of this approach is that it leaves no room for a physical mechanism, so one is limited to software simulation. However, the very nature of the \(3 \times 3 \times 3 \times 3\) itself precludes a physical model, so one is resigned to software simulations in any case. But it is unclear how best to visually display a \(3 \times 3 \times 3 \times 3\) array of hypercubies with all eight cells exposed and not create something far more confusing than intended.

![Figure 9. Exploded view of a 3×3×3.](image)

6.2 Pochmann Stickers

Another approach is to use Pochmann stickers. These are stickers named after Stefan Pochmann.

---

9 https://groups.yahoo.com/neo/groups/speedsolvingrubikscube/conversations/topics/10154
11 http://www.stefan-pochmann.info/spocc/other_stuff/super_5x5/
as he came up with this approach back in 2007 to turn normal $5 \times 5 \times 5$s into Super $5 \times 5 \times 5$s. Figure 10 shows Pochmann stickers applied to a Rubik’s Cube. The orientation of the centres is indeed made visible, adding to the available permutations of the puzzle. This same effect can also be obtained by adding a photo to each face of a $3 \times 3 \times 3$.

This approach can be applied to the $3 \times 3 \times 3 \times 3$. If the additional colours at the edges of the Pochmann stickers on a Rubik’s Cube are viewed as 1D stickers applied to the edges of the square 2D stickers, which identify a face, then these 1D stickers indicate a face’s orientation. On the $3 \times 3 \times 3 \times 3$, Pochmann stickers would be 2D stickers applied to the faces of the 3D stickers.

This approach has two main drawbacks.

1. The interior cubie or core is still not exposed. In the case of the standard Rubik’s Cube that is not a problem, since the interior cubie adds no new information, but for higher order cubes like the $4 \times 4 \times 4$ and $5 \times 5 \times 5$, the interior cubies certainly do carry additional information.

2. This approach results in multiple colours being present on each face. This gives the puzzle a busy look, making a solved state look less obvious. On the $3 \times 3 \times 3 \times 3$ this effect would be even worse, as the 2D stickers would partially cover the 3D stickers and obscure them.

### 6.3 Circle Cube Solution

Another way to overcome the limitations of the Rubik’s Cube is the approach employed by the Circle Cube. Figure 11 shows circles added to each face; they do not turn with the face layer.

These circles are like windows which display the sides of the cubies in the slice layer just below the face. Figure 12 shows an exploded view of a $3 \times 3 \times 3$ array of cubes with the Circle Cube stickers applied, to offer some insight into how these stickers are mapped back to the original 27 cubes.

Figure 13 shows how the core cubie in the Circle Cube has all six faces stickered. Figure 14 shows how each of the six face cubies has four of its faces stickered. Figure 15 shows how each of the 12 edge cubies also has four of its faces stickered. Figure 16 shows how each of the eight corner cubies has three of its faces stickered.

---

13 This animation shows how a Circle Cube turns: http://wwwmww.com/3x3x3x3/CircleCube.gif
Figure 13. The core cubie has six faces stickered.

Figure 14. A face cubie has four faces stickered.

Figure 15. An edge cubie has four faces stickered.

Figure 16. A corner cubie has three faces stickered.

Table 4 shows how the sticker count compares to a normal Rubik’s Cube for each of the four cubie types. In the Circle Cube, cubies now have enough stickered surfaces to give each one a fixed position and orientation in the solved state.

<table>
<thead>
<tr>
<th>Cubie</th>
<th>Count</th>
<th>Stickers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>1</td>
<td>0 Rubik’s 6 Circle</td>
</tr>
<tr>
<td>Face</td>
<td>6</td>
<td>1 Rubik’s 4 Circle</td>
</tr>
<tr>
<td>Edge</td>
<td>12</td>
<td>2 Rubik’s 4 Circle</td>
</tr>
<tr>
<td>Corner</td>
<td>8</td>
<td>3 Rubik’s 3 Circle</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>27</strong></td>
</tr>
</tbody>
</table>

Table 4. Sticker counts for Rubik’s and Circle Cubes.

The advantages of the Circle Cube method for overcoming the limitations of a Rubik’s Cube far outweigh the disadvantages. Each face in the solved state shows only a single solid colour, so it is easily seen at a glance to be solved. Face centre orientation is now an integral part of the puzzle. And even the 27th cubie, the core, is now stickered and a part of the puzzle. The core does not add to the permutation count of the Rubik’s Cube, but this method is general enough that it can be extrapolated to the 4×4×4 and 5×5×5, for which the interior cubies do add to the puzzle.

The first circle 4×4×4s were named Crazy 4×4×4s and were created by Daqing Bao in 2007. The first circle 5×5×5 to reveal all the interior cubies and to give each a unique position and orientation in the solved state was named the Double Circle Real 5×5×5 and was designed by Carl Hoff in 2010.

A physical Circle Cube’s disadvantage is requiring more components than a normal Rubik’s Cube, so its mechanism is slightly more complex. The mapping of the stickers back to the original 27 cubies is also non-trivial. Many people initially believe that far more than 27 pieces must be solved, until understanding that some of the physically distinct components move together as a group: their relationships with each other are locked and cannot be separated. But working out these relationships is a fun and satisfying exercise in itself, so arguably this is not a disadvantage.

7 The Sphere HyperCube

The Circle Cube solution is general enough that it can be applied in higher dimensions as well. The first public discussion of this generalisation was initiated by an online 2011 posting ‘4d Crazy Cubes’ by Krystian Wilisowski under the user name ‘madartilect’. Applying this generalisation to the $3\times3\times3\times3$, the circular windows become spherical windows in the fourth dimension, hence this puzzle is known as the ‘Sphere HyperCube’. These spheres offer a view into the previously hidden interior 4-volume of the puzzle where the 81st hypercubie, the core, is located.

Figure 17 shows a view of the Sphere HyperCube after a partial rotation of the yellow (Up) cell. The 2D analogue for the Circle Cube is inset into the upper right of the image. Just as the circle is fixed in the Circle Cube while the rest of the layer rotates about it, here the sphere is fixed while the rest of the cell is rotated about it.

Seven of the eight cells are shown; the eighth, the white (Down) cell, is just out of frame in one of the six directions. The 27 cubical stickers in each cell are cut with a spherical cut centred on each cell, giving 53 stickers per cell, which are all the same colour in the solved state.

---

16An animation of this complete turn can be found at: http://wwwmwww.com/3x3x3x3/Sphere3x3x3x3.gif

Figure 17. View of the Sphere Hypercube with the yellow (Up) cell partially rotated.
7.1 Mapping Stickers to Hypercubies

Now that we know what the Sphere HyperCube looks like, let us examine how these stickers map to the underlying 81 hypercubies. The core of the $3 \times 3 \times 3$ is totally unseen in the standard picture, but Figure 18 shows that it now has all eight cells stickered. The white sticker is in the out-of-frame eighth cell.

Figure 19 shows that cell centres in the Sphere HyperCube each have six of their eight possible stickers, as opposed to one in the standard $3 \times 3 \times 3$. Each cell centre now has a unique position and orientation in the solved state.

Just as a face centre on a Circle Cube is stickered with the colours of its adjacent faces and is missing the sticker of its own face’s colour, the same is true of a cell centre in this picture of the $3 \times 3 \times 3$. A cell centre hypercubie has a sticker in each of the adjacent cells, but no sticker in its own cell’s colour. The other missing sticker is from the opposite cell.

Figure 20 shows a cell face. Each cell face now has six of its eight possible stickers (as opposed to only two in the standard $3 \times 3 \times 3$). So now they have a unique position and orientation in the solved state.

Figure 21 shows that a cell edge now has five (instead of only three) of their eight possible stickers. Figure 22 shows that a cell corner has no more stickers in this representation: four stickers were already enough to give each corner a unique position and orientation in the solved state.

![Figure 18. Standard core with no stickers and Sphere HyperCube core with eight stickers.](image)

![Figure 19. Standard cell centre with one sticker and Sphere HyperCube cell centre with six stickers.](image)
Figure 20. Standard cell face with two stickers and Sphere HyperCube cell face with six stickers.

Figure 21. Standard cell edge with three stickers and Sphere HyperCube cell edge with five stickers.

Figure 22. Standard cell corner with four stickers and Sphere HyperCube cell corner with four stickers.
Table 5 shows the sticker counts by hypercube type for the standard 3×3×3×3 and the Sphere HyperCube.

<table>
<thead>
<tr>
<th>Hypercube</th>
<th>Stickers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>3×3×3×3</td>
</tr>
<tr>
<td>Cell Centre</td>
<td>0</td>
</tr>
<tr>
<td>Cell Face</td>
<td>2</td>
</tr>
<tr>
<td>Cell Edge</td>
<td>3</td>
</tr>
<tr>
<td>Cell Corner</td>
<td>16</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>81</strong></td>
</tr>
</tbody>
</table>

Table 5. Hypercube sticker counts.

8 Where to Next?

We now have the framework to make a playable Super 3×3×3×3 application. The author would love to be able to offer such a program to the online 4D puzzling community[17] and welcomes contact from programmers interested in helping.

This generalisation of the Circle Cube to the Sphere HyperCube has implications far beyond the 3×3×3×3 itself. It could be used to make a Super Multi 4×4×4×4. This would enable one to sticker the interior 2×2×2×2 to solve it as well as all the hypercubes on the 3D surface of the 4×4×4×4. This would give all of the 4×4×4×4 = 256 hypercubes a unique position and orientation in the solved state.

There are also other directions to explore. In 3D, the cube circle eventually resulted in a series of puzzles called the Crazy 3×3 Plus cubes which were designed by Daqing Bao[18] and released in 2010 by puzzle producer mf8. Figure 23 shows the eight different versions.[19]

9 Conclusion

It is time to take some of the innovation that has been applied to 3D puzzles with the advent of 3D printing and all the new designers it has attracted and apply some of the innovation to the higher dimensional puzzles. This paper presents an example, the application of the Circle Cube method of creating a Super Multi 3×3×3 and applies it to the 3×3×3×3. Table 6 shows the effects of making these hidden orientations visible on the 3×3×3×3.

The entries in this table have been verified with the Groups, Algorithms and Programming (GAP) programming language for computational discrete algebra[20]. The GAP code used to verify the figures can be found here.[21-24]

<table>
<thead>
<tr>
<th>3×3×3×3 Type</th>
<th>Corners</th>
<th>Edges</th>
<th>Faces</th>
<th>Centres</th>
<th>Permutations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>16! / 2 × 12! / 3 × 32! × 6! / 2 ×</td>
<td>24! / 2 × (8! / 4!) × 2! / 2 ×</td>
<td>1 × 24 / 2</td>
<td>1 × 24 / 2</td>
<td>1 × 24 / 2</td>
</tr>
<tr>
<td>Visible Cell Centre</td>
<td>16! / 2 × 12! / 3 × 32! × 6! / 2 ×</td>
<td>24! / 2 × (8! / 4!) × 2! / 2 ×</td>
<td>24 / 2</td>
<td>24 / 2</td>
<td>24 / 2</td>
</tr>
<tr>
<td>Visible Cell Face</td>
<td>16! / 2 × 12! / 3 × 32! × 6! / 2 ×</td>
<td>24! / 2 × 8! / 4 ×</td>
<td>4 / 2</td>
<td>24 / 2</td>
<td>24 / 2</td>
</tr>
<tr>
<td>Super</td>
<td>16! / 2 × 12! / 3 × 32! × 6! / 2 ×</td>
<td>24! / 2 × 8! / 4 ×</td>
<td>4 / 2</td>
<td>24 / 2</td>
<td>24 / 2</td>
</tr>
</tbody>
</table>

Table 6. Permutation counts for some 3×3×3×3 variants.

---

[17]https://groups.yahoo.com/neo/groups/4D_Cubing/info
[19]Image kindly provided by mf8.
[21]http://www.wwwmwww.com/3x3x3x3/Normal3x3x3x3.txt
[22]http://www.wwwmwww.com/3x3x3x3/3x3x3x3wCellCenter.txt
[23]http://www.wwwmwww.com/3x3x3x3/3x3x3x3wCellFace.txt
[24]http://www.wwwmwww.com/3x3x3x3/Super3x3x3x3.txt
Acknowledgements

Thanks to Brandon Enright and Jaap Scherphuis for the permutation calculations presented in this paper. Thanks to Mathologer for his excellent introductory video for the $3 \times 3 \times 3 \times 3$ on YouTube that sparked the author’s current interest in this puzzle. Thanks to Don Hatch and Melinda Green for their development of Magic Cube 4D. Most of the images in this paper were rendered using the 3D CAD program SOLIDWORKS.

References


Carl Hoff is an epitaxy process service engineer for Applied Materials currently supporting Global Foundries in Malta, New York. Research interests include twisty puzzle design and theory, packing puzzles, and puzzle rings. Address: 108 North Broadway, Malta, NY, 12020-5100, USA. Email: carl.n.hoff@gmail.com

Gloop Challenge #7

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.
Mitigating Non-Strategic Coalitions

Craig Duncan, Ithaca College

A common problem with three-player games is ‘kingmaking’, in which a player with no hope of winning is able to determine the eventual winner. I describe a known method for mitigating this problem and its modification for games that include final ranks. I also introduce the related term ‘princemaking’ to describe cases in which the leading player is able to determine the second place-getter, and strategies for mitigating this problem as well.

1 Introduction

COMBINATORIAL games are two-player games with no hidden information and no chance elements [1]. Games involving more than two players are susceptible to non-strategic coalitions [7] in which players may pursue personal agendas rather than playing strictly to win. A well-known example is the kingmaking problem, in which a player with no hope of winning is able to determine the eventual winner [3, 4], which can ruin a game for many players [11].

A lesser known type of non-strategic coalition problem is what I call princemaking, which occurs when the leading player is able to determine who comes second. This article explores the issues of kingmaking and princemaking and presents ways to mitigate these effects. Future references to ‘game’ in this article refer to combinatorial games.

2 The Kingmaking Problem

An interesting feature of three-player games is emergent temporary strategic alliances between two players, typically to thwart the current leader. Algorithms have been developed in an attempt to understand these social dynamics, but formal studies of these, such as [6], remain inconclusive, making it hard to say at a formal level what is ‘rational’ play in such games.

However, the intriguing social features of three-player games also create the potential for kingmaking. This is a problem for many players, as the chosen winner may consider the victory to be hollow, and the remaining player may resent having the chance of victory snatched away by the kingmaker rather then through their own strategic errors.

The very features of abstract strategy games that appeal to their devotees, namely the significant player control due to no hidden information or randomness, also exacerbate kingmaking in three-player games. A kingmaker has full knowledge of the game state (no hidden information) and more ability to manipulate the game (no randomness).

3 The Stop-Next Rule

One way to mitigate kingmaking is the Stop-Next (SN) rule:

Players may not let the next player win on the next turn, unless there is no other choice.

This rule was developed in 2002 to specifically address the problem of kingmaking in three-player games [23 pp. 161–165].

3.1 Yavalath

A well-known game featuring SN is Yavalath from 2007 [12 pp. 75–86].

Yavalath is played on a hexagonal grid of hexagons with five cells per side.

The board starts empty. Players take turns adding a piece of their colour to an empty cell.

Players win by making a line of four (or more) of their pieces, but lose by making a line of exactly three beforehand.

The game is tied if the board fills up before any player wins.

Yavalath was originally designed for two players, but later extended to three players by adding the following rules [12 pp. 75–86]:

A player must block the next player’s win if possible. Losing players leave the game but their pieces remain on the board. The winner is either the last surviving player or the first player to form a line of four (or more) of their pieces.
While Yavalath is most famous for having been designed by computer [9], it is also known among board gamers for working well with three players. A surprising amount of drama and strategy emerges from this simple rule set, and SN is an integral part of making the three-player version work. To appreciate why, consider Figure 2, which shows a game with turn order White/Black/Red with Black to move.

Both White and Red threaten to win next turn with a line of four (at the points marked +). If SN were not in place, then Black could decide who wins this game despite having no chance of winning themselves. If Black blocks the red line, then White will win the game (the triangle on the left gives White a winning advantage). Alternatively, if Black plays anywhere else then Red will complete a line of four to win.

The SN rule requires Black to block the red line, giving Black no choice in the matter. Note that SN is an exploitable strategy in its own right, which can be used to manipulate opponents into making unfavourable moves; the key difference being that these moves will be dictated predictably by the rules of the game and not the opponent’s personal agenda. The next game, Chromatix, shows such manipulation in action.

3.2 Chromatix

Chromatix, a game that I designed in 2015, clearly demonstrates this principle. The rules for Chromatix are as follows.

Chromatix is played by Pink, Grey and Maroon on a hexhex board with opposite sides bearing their colour. Players own stones that mix to form their colour:

- Pink owns red and white stones.
- Grey owns black and white stones.
- Maroon owns red and black stones.

Players take turns placing one of their stones on any empty cell.

A player wins by connecting their two opposite sides by a group of stones of their own colours. SN is in effect.

In an advanced version, a player can also win by connecting three non-adjacent sides.

Grey has no winning move, but Maroon is one move from winning, since a red or black stone at + will create a red/black connection between the maroon sides. SN requires Grey to stop Maroon from winning next turn, which can only be done by playing a white stone at +, as in Figure 3.

This move, however, creates a red/white connection between the pink sides. Thus SN forces Grey to give Pink the win. Pink was able to bring the game to a state in which both opponents were one move away from winning, knowing that SN would require Grey to give Pink the win.

---

1 https://www.boardgamegeek.com/boardgame/178059/chromatix
2 Note that corner hexes belong to both of the adjacent sides which they touch.
Thus SN can mitigate kingmaking in three-player abstract strategy games and at the same time add strategic interest. However, ‘mitigate’ is not ‘eliminate’; it is still legal for a player to intentionally play to set up the second player for a win on a later turn (beyond the next), and it is legal for a player to intentionally play in a way that sets up the third player for a win on the third player’s very next turn.

3.3 McCarthy’s Revenge Rule

An alternative to SN for mitigating kingmaking is provided by McCarthy’s Revenge Rule. This rule was proposed around 1950 by John McCarthy during gaming sessions with John Nash, Lloyd Shapley and Martin Shubik [10, p. 390]:

*If you cannot win the game, then hurt the player who has hurt you the most.*

While this revenge rule has psychological appeal, it may be unclear who is actually to blame for your predicament. The exact perpetrator may be hard to determine amidst the game’s complexity, or multiple opponents may be equally responsible, or perhaps a player only has themself to blame for poor play.

Another practical problem is that of enforcement. It may be unclear what moves will hurt your victim, and other players may disagree whether you are being appropriately vengeful. This can be hard to enforce, and is perhaps best left as a form of etiquette among players. SN has the advantage of mitigating the kingmaker problem in a transparent and unambiguous way. Further, Browne has recently demonstrated that hurting the opponent who has hurt you the least may be more effective in some cases [7].

4 Generalising Stop-Next

SN has a side effect which may be undesirable for some players in three-player games with final ranks, i.e., first place, second place, third place. It obliges the current player to prevent the next player’s immediate win, even when doing so worsens the current player’s own final rank. I present a version of SN which not only prevents non-strategically harming others, but also does not require non-strategically harming oneself.

4.1 Alaric

The following example from Bill Taylor’s game Alaric, which itself is a variant of Alak [11], demonstrates this problem with final ranks.

Board *a* in Figure 4 shows a sample Alaric endgame, with White to play next. If White plays on the empty cell on the left, then the result is board *b*. Now Red’s only move is to play on the remaining open cell, resulting in board *c*. Black has no legal move, so the game formally finishes at the end of Red’s turn. The final scores are 4/3/1 making Red the winner. White’s move *a* therefore led to an immediate win for the next player (Red).
Alaric is played on a circular ring of cells by White, Red and Black. Players take turns placing a stone of their own colour on an empty cell.

If the move causes a consecutive run of enemy stones (of either colour) with no adjacent empty cell(s), then those stones are captured and removed. The next player may not play onto any of the newly empty cells.

The game ends when the next player has no legal move. It is won by the player with the most territory, i.e. the number of that player’s stones in play plus the number of empty cells in continuous runs that touch only that player’s stones.

With SN in effect, White must play instead on the empty cell on the right to capture the single red piece, resulting in board $d$ in Figure 5. The newly vacated cell (marked $\times$) is unplayable for the next player (Red), whose only option is to make move $e$. The blocked cell is now playable again, so Black plays there, capturing all opponent stones (board $f$). There are no legal spaces where White can play, so the game ends with a clean 8/0/0 sweep by Black.

Thus without SN, White can achieve second place ahead of Black in third place, but with SN, White is forced to tie with Red. The following modification is proposed to avoid this problem.

4.2 General-Stop-Next

I propose the following general principle:

Players must block a win by the next player unless such a block causes self-harm.

Formalising this idea into a precise rule proved tricky, but Bill Taylor, João Neto and I eventually hit upon the following rule that I call General-Stop-Next (GSN):

The mover may not choose any move that gives the next player an immediate win, unless the mover’s only other moves would let the third player win on their next turn (regardless of the next player’s possible moves) in a way that worsens the mover’s own final rank.

GSN prevents self-harm in cases such as the Alaric example shown, but does not prevent all cases of self-harm. For instance, consider a case in which the third player does not have an immediate win next turn, but in which the mover will provably finish in a lower final rank by stopping the next player; GSN still requires the mover to stop the next player and accept the lower rank.

A more general rule that permits the mover to let the next player win whenever the mover can prove that stopping the next player will ultimately reduce the mover’s final rank would surely be impractical, requiring confusing proofs about future continuations, and probably resulting in disagreement among players. So a more permissive rule of this sort would potentially mire the game in intractable complexities of adjudication. The more restricted formulation of GSN, which requires looking ahead only two moves, is complicated enough. Note that GSN is equivalent to SN in games with no final ranks.

5 The Princemaking Problem

Just as a player whose loss is certain may find themselves in a position to determine which opponent wins, a player whose win is certain may find themselves in a position to determine which opponent gains second place. This problem, which I call ‘princemaking’, is undesirable for the same reasons as kingmaking.
5.1 Frozen-Irrelevance

One possible way to mitigate princemaking is to include a ‘catch-up mechanism’ to prevent a runaway leader in the first place. Another possible way (which can be combined with a catch-up mechanism) is to not use final ranks at all: simply end the game when it is provably certain that one player has won, and do not distinguish second and third place. A (less drastic) third option is to remove the certain winner from further play in the game, but let the two remaining opponents continue to battle for second place. This latter option, dubbed Frozen-Irrelevance (FI) by Bill Taylor, works as follows:

A player guaranteed of being the sole victor is frozen from further play, making that player irrelevant to the battle for second place.

This is similar in principle to the mechanism described by P. D. Straffin, Jr. for Three-Player Hex [10, p. 393], in which a player with no way to win is eliminated from the game. This suggests a more general form of FI in which a player is frozen as soon as a given rank is provably achieved – be it first, second or third – which could have potential benefits for both kingmaking and princemaking. However, I focus on the more specific formulation above to address princemaking in particular here.

5.2 Feedback Morro

Feedback Morro is a three-player game that uses FI and a catch-up mechanism to mitigate the problem of princemaking. It was developed by João Neto, Bill Taylor and me in 2015 as a three-player variant of Morro. Figure 6 shows a game of Feedback Morro after each player has played $n - 1$ turns.

Feedback Morro is played on a 9×9 grid by White, Red and Black. Players in turn place one or more of their stones on empty cells. Players try to form lines of their own stones, either horizontally, vertically or diagonally. Lines of the same length match. The player with the longest unmatched line at the end of the game wins.

The number of stones that players place each turn is equal to their current rank at the start of their turn:

- First place plays one stone.
- Second place plays two stones.
- Third place plays three stones.

FI: A player who creates an unbeatable line immediately wins and is frozen from further play, while the remaining players continue to battle for second place.

---

3 Catch-up mechanisms also help reduce kingmaking by making it harder for players to fall hopelessly behind.
Each player’s current score information is shown below the board, with the players listed in order of score. White is currently in first place having two 5-lines (i.e. lines of length 5), while the opponents each have only one 5-line. Red has four 4-lines compared to Black’s three 4-lines, so Red takes second place.

In the next game round \( n \), White will play one stone, Red (who will still be second) will play two stones, and Black (who will still be third) will play three stones. Suppose that White, Red and Black then play the marked stones in Figure 7. The table below the board shows the updated player ranks.

White’s single stone blocks Red’s diagonal 4-line from extending. This was the last remaining chance for any of White’s opponents to make a second 5-line. White’s current two 5-lines can therefore not be beaten – or even matched – making White the winner.

With FI in place, White must stop playing while Red and Black continue to play on for second place. Note that Black made another 4-line, so second place will be decided by shorter lines.

Figure 8 shows plausible moves by Red and Black in round \( n + 1 \) (White has been frozen so did not move).

Red created three more 3-lines by playing in the upper right. Black was entitled to three stones (since Red’s new 3-lines demoted Black to third place), but could only play in the two remaining empty cells, creating one more 3-line in the upper left. Black and Red are tied for 5-lines, 4-lines and 3-lines, but Black has more 2-lines, giving Black second place. The result with FI in place is: White, Black, then Red.

5.2.2 Continuations without FI

Let us consider the same game without FI in force. In this case, White will continue to play in round \( n + 1 \), despite having an invulnerable lead after turn \( n \), and White’s move will determine which opponent finishes in second place.

White can engineer second place for Black by playing in the top right cell. Red can then make only one more 3-line, as shown in Figure 9, allowing Black to retain second place.

Alternatively, White can engineer second place for Red by playing in the upper left cell. This blocks Black from making any more 3-lines, as shown in Figure 10 giving Red second place. This example demonstrates how the FI rule can act to prevent the winning player from choosing who comes second.
6 Conclusion

Abstract strategy games with three players enable interesting mechanisms not found in two-player games, such as temporary alliances, but also enable detrimental side effects such as kingmaking and princemaking. Kingmaking can be mitigated by using a simple Stop-Next rule, which can be generalised to also work for three-player games in which final positions are ranked.

Princemaking can be mitigated by catch-up mechanisms to keep scores close, and by the Frozen-Irrelevance rule, which requires a player with an invulnerable lead to cease further play. With these two means of mitigating the effects of non-strategic coalitions, many three-player abstract strategy games may become more viable and interesting games. This is good news for designers and players of abstract games; the design space of three-player abstractions is relatively unexplored compared to that of two-player games.

Acknowledgements

The author would like to thank Bill Taylor and João Neto for helpful discussions (and many games by email) that led to the ideas in this article. The author would also like to thank Russ Williams and Cameron Browne for extensive edits, and the anonymous reviewers for their helpful feedback.

References


Craig Duncan is Associate Professor of Philosophy and Religion at Ithaca College. His research interests include political philosophy, ethics, and the philosophy of religion.

Address: Dept. of Philosophy and Religion, Ithaca College, 953 Danby Rd, Ithaca, NY, USA.
Email: cduncan@ithaca.edu
The Development of a Tangram Family

Fred Horn, Freelance Game Designer

This article presents a family of Tangram-like puzzles, found by exploring other possible sets of pieces which are similar to the set of seven traditional Tangram pieces. The resulting Tangram variations are interesting in their own right, and can be combined into a set of twelve pieces to create additional puzzles.

1 Introduction

The well-known Tangram puzzle consists of the seven pieces shown in Figure 1 with the goal being to arrange the pieces to form a square or other specified silhouette shape. The standard set consists of five triangles (T), one square (S) and one parallelogram (P).

Figure 1. The traditional Tangram set.

Tangram has a long history, possibly originating in China a millennium ago [1] or even earlier [2], and became popular in Europe in the 19th century. Much has been written about the mathematics of the set [5, 6] and many collections of target shapes have been released [2, 3, 4].

Dekking [7] investigates the many different shapes that can be made with the traditional set. In 1976, this inspired my good friend Johan Siders and myself to look in the other direction, and experiment with differently sized pieces in the traditional shapes. Were the specific Tangram pieces special, or could some other combinations of shapes and sizes also produce similarly interesting puzzles? Would it be possible to improve on the original design?

Early results from these investigations were published in Dutch journal Natuur & Technik in 2002 [8] and a series of short pieces in Cubism for Fun in 2010 [9] [10] and 2011 [11] [12]. This paper summarises these earlier findings, to present the ‘complete family’, and adds some recent updates.

2 Sets with Five Triangles

We initially considered similar piece sets with distributions similar to the traditional set (i.e. five triangles, one square and one parallelogram) but with differently sized pieces. Our first observation was that the Tangram pieces can all be divided into isosceles or right-angled triangles of equal size, as shown in Figure 2. This gives a convenient unit of measure that we call basic triangles (BT) with which we can make calculations about the total area of given piece sets. For example, the pieces in the standard set add up to $1 + 1 + 2 + 2 + 2 + 4 + 4 = 16$ BTs.

Figure 2. The Tangram divided into triangles.

Pieces can be aligned two ways, with the short side of each BT aligned with either the grid or its diagonal, as shown in Figure 3. A square formed by two BTs aligned with the grid (left) will have a unit side length of $u$, while a square formed by four BTs aligned with the diagonal (right) will have a unit side length of $\sqrt{2}u$. Using BTs rather than Euclidean area simplifies area calculations.

2.1 Square Target

We were initially interested in finding piece sets that pack into the traditional square, rather than the many thousands of more exotic Tangram shapes that have been published. We therefore considered how many BTs will fit into squares of different sizes.
Figure 3. Size-2 and size-4 squares.

Table 1 shows the total area in BTs for squares of size 1 × 1 to 4 × 4 for both alignments. The task is now to find sets of pieces whose total BT area matches these totals.

<table>
<thead>
<tr>
<th>Alignment</th>
<th>1 × 1</th>
<th>2 × 2</th>
<th>3 × 3</th>
<th>4 × 4</th>
<th>5 × 5</th>
<th>6 × 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>2</td>
<td>8</td>
<td>18</td>
<td>32</td>
<td>50</td>
<td>72</td>
</tr>
<tr>
<td>√2u</td>
<td>4</td>
<td>16</td>
<td>36</td>
<td>64</td>
<td>100</td>
<td>144</td>
</tr>
</tbody>
</table>

Table 1. BT counts for square target areas.

In the spirit of the original Tangram, we want piece sets with five triangles, and at least one triangle of each size up to size 4. This allows the following combinations of triangles: {1, 1, 2, 2, 4}, {1, 1, 2, 4, 4}, and {1, 2, 2, 4, 4}. The leftmost column in Table 2 shows these combinations and the middle column sows their total area in BTs.

We now want to add two more pieces: either squares, parallelograms, or both, of size 2, 4 or 8. Adding these two pieces gives six additional combinations to consider: {2, 2}, {2, 4}, {4, 4}, {2, 8}, {4, 8}, {8, 8}. The rightmost column in Table 2 shows the total BT area when these combinations are added to an existing base of five triangles. Note that we exclude the size-8 triangle, as all five triangles are already accounted for.

<table>
<thead>
<tr>
<th>Triangles</th>
<th>Σ</th>
<th>Square, Parallelogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1,2,2,4</td>
<td>10</td>
<td>2,4,4,8,8</td>
</tr>
<tr>
<td>1,1,2,4,4</td>
<td>12</td>
<td>4,4</td>
</tr>
<tr>
<td>1,2,2,4,4</td>
<td>13</td>
<td>1,2,2,4</td>
</tr>
</tbody>
</table>

Table 2. Total numbers of basic triangles (BTs).

It can be seen from Table 2 that the only piece set areas that coincide with the square target areas from Table 1 are 16 and 18 (highlighted). This gives us the exact combinations of pieces that will pack into squares. Labelling the potential pieces up to size 8 as shown in Figure 4, we are therefore interested in the following six sets of pieces:

{\(T_1, T_1, T_2, T_2, T_4, S_2, P_4\)}
{\(T_1, T_1, T_2, T_2, T_4, P_2, S_4\)}
{\(T_1, T_1, T_2, T_2, T_4, S_4, P_4\)}
{\(T_1, T_1, T_2, T_2, T_4, S_2, P_2\)}
{\(T_1, T_1, T_2, T_4, S_2, P_2\)}
{\(T_1, T_1, T_2, T_4, P_2, S_4\)}

Figure 4. The basic shapes to size 8, colour coded by set. Note that \(T_8\) is not included in our analyses.
Figure 5 shows all solutions for these six piece sets, including reflections but not rotations, separated into those sets with a total area of 16 and those with a total area of 18. Each row shows a different piece set; you might recognise the first row as the traditional Tangram set, which has exactly two solutions for a square target.

This analysis produced a remarkable result: there are 16 solutions for the BT=16 sets and 18 solutions for the BT=18 sets! We leave it to the numerologists to make what they will of that.

2.2 Triangular Target

Given our success in finding alternative (but similar) piece sets that pack into a square target, we then turned to other primitive target shapes. The triangular target is a convenient choice, as cutting a square along two opposite corners gives a triangle of half the area. We can therefore re-use our calculations summarised in Tables 1 and 2 to find piece sets that will pack a triangular target shape, and indeed the following sets do:

\{T_1, T_2, T_2, T_4, T_4, S_4, P_8\}
\{T_1, T_2, T_2, T_4, T_4, P_4, S_8\}

Both of these sets have a total area of 25 BTs, which will pack a 5×5 square in the u alignment with a total area of 50 BTs (halved). Figure 6 shows two solutions using these piece sets, one including square S_8 and the other including parallelogram P_8.

Figure 6. Triangular solutions with a size-8 piece.

3 Unrestricted Piece Sets

We then considered what other Tangram variants might work using other than five triangles. Figure 7 shows the seven pieces to size 4. Can these be packed into a regular shape?

Figure 7. The seven pieces to size 4.
Unfortunately, their area sums to 26 BTs. The most regular rectangle that would allow this is $1 \times 13u^2$ (each unit square $u^2$ contains two BTs), which would not even allow some of the pieces to be placed.

However, the piece set shown in Figure 8 has a total area of 42 BTs, which fills a $3 \times 7u^2$ rectangle. This piece set, which includes representatives of all pieces up to size 8 (except for $T_8$), was laser cut in perspex for further testing (Figure 9).

### 4 Seven Pieces

Returning to the original Tangram’s count of seven pieces, we then took the pieces shown in Figures 8 and 9 and considered the question: which subsets of seven pieces from this set can form a square target? Further, and even more in keeping with the original Tangram, we only wanted subsets with two $T_1$ triangles.

As before, we knew that the completed square must consist of 16 or 18 BTs, and the seven selected pieces will be of size 1, 2 or 4. The large parallelogram $P_3$ will not even fit into the square area, and the large square $S_8$ does not leave sufficient space for the other six pieces, except for one rather boring simple piece set: $\{1, 1, 2, 2, 2, 2, 8\}$.

For the size 16 square, only one combination of sizes works: $\{1, 1, 2, 2, 2, 4, 4\}$. For the size 18 square, again only one combination of sizes works: $\{1, 1, 2, 2, 4, 4, 4\}$. Tables 3 and 4 show the valid seven-piece sets including two $T_1$ triangles that pack to fill a square target, of sizes 16 and 18, respectively. Note that the values shown are piece counts, not BT counts. Figures 10 and 11 show example packings of the six valid piece sets for size 16 and five valid piece sets for size 18.
4.1 Large Pieces

Turning now to piece sets with large size-8 pieces, Figure 12 shows the only seven-piece packing for a size-18 square target using large square $S_8$.

![Figure 12. Seven-piece packing at size 18 with $S_8$.](image)

With large square $S_8$ and large parallelogram $P_8$, two different seven-piece sets can pack a size-32 square target, as shown in Figure 13.

![Figure 13. Two size-32 packings with $S_8$ and $P_8$.](image)

4.2 Self-Referencing Shapes

Johan Sieders and I also explored a different way in which Tangram might be generalised. Given that the original puzzle includes a square that is the same shape as the default target, can we create non-square Tangram variants in which the target shape is similarly reflected in the shape of one of its pieces?

We discovered a pentagonal Tangram variant (with a pentagonal piece) and a hexagonal Tangram variant (with a hexagonal piece), as shown in Figure 14. The hexagonal variant has been published as Winter Glow [13], shown in Figure 15, after an earlier incarnation as Hextan.1

![Figure 14. Pentagonal and hexagonal variants.](image)

![Figure 15. Published hexagonal variant.](image)

Acknowledgements

Thanks to the editors for reformatting this piece from the notes provided, and redoing the figures.

References


1http://www.hongs.nl/index.asp?vi=li&s=Hextan
Fred Horn is a prolific Dutch amateur game and puzzle designer with hundreds of titles to his name. His published titles include Citadella, Cardo, Agilidade, Hextan and Gloop. Fred is a retired civil servant who divides his time between amateur jazz music and game design.

Address: The Hague, The Netherlands.
Email: wm.fred.horn@kpnmail.nl

---

**Gloop Challenges #11 and #12**

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.

**Gloop Challenge #11**

![Gloop Challenge #11](image1)

**Gloop Challenge #12**

![Gloop Challenge #12](image2)
Graph-Based Search for Game Design

Daniel Ashlock, University of Guelph

Cameron McGuinness, University of Guelph

This article is the second in a tutorial series on the relationship between games, puzzles, and combinatorial graphs. It introduces some classical and modern algorithms for permitting a computer to play games and design games and puzzles using search-based procedural content generation. These algorithms are applied in the evolutionary design of level maps.

1 Introduction

This article introduces algorithms that are useful both for creating artificial intelligence (AI) players for games and for procedural content generation (PCG). AI algorithms for playing games must search the space of possible future moves, which could require large amounts of time if the algorithms are not clever about how these searches are performed. The process of devising better search methods is the basis for much of the research in this area.

Procedural content generation covers a very wide range of topics. It includes any algorithmic method for designing a game or part of a game. This article introduces the topic and presents search-based procedural content generation (SB-PCG) as a motivating example. SB-PCG is demonstrated for creating maps for use in a fantasy role-playing game.

Graph algorithms are algorithms that operate on or are structured by graphs. This article introduces two types of graph algorithms that are especially useful in game play and puzzle design. The first, tree search algorithms, operate on a structure called a game tree. The second, path finding algorithms, are used to search a game board or map to find the most efficient set of moves to reach an objective.

1.1 Graph Theory

The first article in this series defined some basic graph theory terminology. We now introduce additional terms for this article. A tree is a connected graph with no closed paths. Figure 1 shows an example tree with nine leaves and five internal nodes. It is connected – it is possible to go from any node to any other node by following connections in the tree – but there are no closed loops in the structure.

A rooted tree is one that has a distinguished node called a root. Rooted trees are typically displayed in layers relative to the root. An example of a rooted tree is shown in Figure 2. Its root is the unique vertex on level 1. For both sorts of tree, a leaf is a vertex of degree 1 (with one adjacent neighbour) that is not the root. The vertices with degree greater than one are called internal nodes.

2 Tree Search

Consider a two-player game in which the players take turns moving. This situation can be naturally structured as a tree. The root of the tree is the starting (or current) position for the game. The second level of the tree contains all the positions that can be reached with one move by the

---

1For examples, see: https://en.wikipedia.org/wiki/Book:Graph Algorithms

first player. The third level of the tree are the
game positions that can be reached via a move of
the second player, and so on. A tree which thus
models a game’s play-space is called a game tree.
A layer of a game tree is called a ply of the tree.

When searching a game tree, the quality of
the board from the point of view of the player
making the first move is evaluated to guide the
search. The function used to evaluate the game
position is called the static evaluation function. Fig-
ure 3 shows a game search tree with nodes la-
belled with values of the static evaluation func-
tion. In this simple example, the static evaluation
function is used to compute only the values of
leaves. The internal nodes’ values are then com-
puted by assuming that each player will make
the best move they can. The first player strives to
maximise the evaluation value while the second
strives to minimise it; hence, this sort of search
tree is called a minimax search tree.

For example, the tree in Figure 3 has four
plies. The nodes of the tree are states of the game,
the connections represent individual moves. Red
(first and third level) and blue (second and fourth
level) nodes represent turns taken by different
players. The numbers on the nodes represent
how favourable the game position for root player.
Values on the leaves result from the static evalua-
tion function.

A major improvement on minimax tree search
is to use \(\alpha-\beta\) pruning, a technique that avoids
searching subtrees that cannot possibly change
the search outcome. \(\alpha-\beta\) pruning is well de-
scribed elsewhere in the literature [5], and typi-

\[\text{Figure 3. A minimax search tree.}\]

\[\begin{array}{ccccccc}
7 & 6 & 5 & 4 & 5 & 6 & 7 \\
6 & 5 & 4 & 3 & 4 & 5 & 6 \\
5 & 4 & 3 & 2 & 3 & 4 & 5 \\
4 & 3 & 2 & 1 & 2 & 3 & 4 \\
5 & 4 & 2 & 1 & 2 & 3 & 4 \\
6 & 5 & 3 & 2 & 3 & 4 & 5 \\
7 & 6 & 5 & 4 & 3 & 4 & 5 \\
\end{array}\]

\[\text{Figure 4. Dijkstra’s algorithm on a square grid.}\]

\[\text{Figure 5 shows Dijkstra’s algorithm in the}
\]

\[\text{presence of an obstruction, indicated by black}
\]

\[\text{squares. The shortest paths wrap around the ob-
struction, with the square farthest from the start-
}\]
ing square being near the middle of the board. Di-
jkstra’s algorithm can be implemented efficiently
to take a number of steps proportional to the num-
er of vertices in the graph. In Figure 5, that is the
number of unobstructed squares on the board.

![Graph with squares and numbers](image)

**Figure 5.** Distances incorporating obstructions.

While the graph that Dijkstra’s algorithm is
searching need not be a tree, the algorithm oper-
ates as a tree search algorithm. There is a node
of the search tree at each square the search visits.
The nodes below a given part of the tree are the
unvisited nodes that can be reached in one move
from the node currently being considered.

The interested reader can explore the $A^*$ al-
gorithm. This is a modification of Dijkstra’s al-
gorithm that uses heuristics to (usually) improve
performance.

## 4 Monte Carlo Tree Search

**Monte Carlo tree search** (MCTS) is a sampling-
based technique that builds a search tree based on
the outcomes of (semi) random playouts [12]. It
improves on minimax in that the static evaluation
function need only be applied at the end of each
playout, not at every node visited. Sampling the
search space, rather than completely exploring it,
has a number of benefits:

- An MCTS engine has an (ignorant) recommenda-
tion for what move to make after making only one playout. It is an *anytime* algorithm that can return its recommenda-
tion for a move whenever it is needed.
- While increasing the number of plies in a minimax
search causes a huge change in the amount of time needed to generate a
move, MCTS can just keep sampling. Its recommendation gets better the longer it
runs.
- An MCTS engine is driven by a node-
choice function that can be specialised to
particular sorts of games [13].

The MCTS algorithm works as follows. Start-
ing at the root of the game tree, the algorithm
uses a choice function to decide which chain of
play to follow downward through the tree until
reaching a node that is not fully expanded. From
that node, one or more leaf moves are added to
the tree below the selected node. A (semi) ran-
dom playout [2] is performed from each new leaf,
and its result backed up the sequence of moves
performed, updating each node in a process of
**back propagation**.

**Figure 6** shows an example MCTS tree under
construction. The round nodes are in the tree. The hexagonal nodes are (semi) random moves
of a playout from the node they descend from.

![MCTS Tree](image)

Each node stores the following information:

1. Fraction of wins in the playouts from that
   node down.
2. Number of playouts from that node down.

These numbers are used by the choice func-
tion to decide which node to go to next every
time a new path down from the root of the tree is
explored. For MCTS, the key component is this
choice function. When at a non-leaf of the tree,
selection of the next node to go to is based on the
fraction of victories at the next node, and the num-er of times the current and next node have been
visited. A number of different choice functions
have been proposed, as described in [12].

---

1. We say ‘(semi) random’ as strictly random playouts typically do not give very good results, and need to be
biased with some domain-dependent information to make better-than-random move choices.
4.1 Exploration vs Exploitation

A choice function can favour nodes with a high fraction of victories (exploitation) or nodes with relatively few playouts below them (exploration). The best balance of these two qualities is game-dependent and the focus of much MCTS research.

If $w$ is the number of victories at a node being considered, $n$ is the number of playouts below its parent node, and $m$ is the node’s own number of playouts, then the upper confidence bound (UCB) choice function \[10\] is used as follows:

$$UCB = \frac{w}{n} + 2C \sqrt{\frac{2\ln(n)}{m}}$$

The first term is the exploitation term, rewarding victories. The second term is the exploration term that rewards choosing nodes that have not been visited as often. The constant $C$ is used to fine-tune the balance between exploration and exploitation. In use, the node with the highest value for the choice function is chosen, breaking ties uniformly at random.

MCTS is an approximation to a minimax tree search. It has been proven that MCTS converges to minimax search as the number of samples increases. This convergence can be very slow, requiring a very large number of samples to get close. This means that MCTS is useful when minimax and its variants are impractical, but that minimax is more desirable when practical.

5 Designing Puzzles with MCTS

In addition to providing the AI for an opponent in a two-player game, MCTS may be used to solve puzzles. This means that it can also be used as part of a system to design puzzles, a type of search-based automatic content generation. In this case, each time the MCTS algorithm finds a ‘win’ (a solution to the puzzle), that solution is recorded in a database of solutions.

5.1 Ten-Yen Polyomino Puzzles

A polyomino is a shape formed by $N$ adjacent squares to form a connected set. Polyomino puzzles involves joining sets of polyominoes into target shapes. For example, Figure 7 shows two solutions of a polyomino puzzle called Ten-Yen that uses ten distinct polyominoes. There are a large number of ways to assemble the Ten-Yen pieces, and it is possible to declare secondary challenges such as requiring that all pieces that share a colour form a contiguous region (Figure 7 left), or surrounding one colour with another.

If a challenge of this sort has one solution then it typically has many, but the number of solutions varies substantially from shape to shape \[9\]. This makes MCTS sampling of solutions a suitable way to assess the difficulty of a given target shape for a puzzle.

The video game Tetris uses all five size-4 polyominoes. The pieces for Ten-Yen have a total area of $36 = 4 \times 9$. This suggests the idea of making the Tetris pieces out of $3 \times 3$ squares, giving several shapes which might be possible to fill with the Ten-Yen polyominoes. These five candidate goal shapes are shown in Figure 8.

We performed the following experiments. MCTS was used to check the relative difficulty of the five target shapes. The number of distinct solutions to the puzzle found in a fixed amount of time serves as a surrogate for the difficulty of a puzzle. The more solutions are found, the easier the puzzle. Example solutions for all target shapes are shown in Figures 9 to 13.

\[3\] Patented by Multiple Products Corporation in 1950: http://everything2.com/title/The+Ten-Yen+Puzzle
5.2 Polyominoes as a Search Problem

A human solving a polyomino puzzle will try various pieces, guided by their geometric intuition. Using MCTS to find solutions requires casting the puzzle as a tree-search problem. A polyomino can have from one to eight orientations, as shown in Figure 7.

A move in the search algorithm for solving a polyomino puzzle consists of selecting a shape, and an orientation of that shape, and then placing it as far up and left as possible. Correctly numbering the order of the pieces in any solution to the puzzle shows that this procedure for solving the puzzle can discover any solution. A numbering for the first solution to Ten-Yen shown in Figure 7 is given in Figure 15.

Any solution to the puzzle has a numbering of the pieces that enables rediscovery of the solution by the given algorithm. The full search tree, executed for these moves, would discover all possible solutions to the polyomino puzzle. The space of all solutions is combinatorially large, making the use of MCTS sampling a natural choice for estimating the relative number of solutions of different target shapes for puzzles using the same set of polyominoes.

Since the aim is to find many solutions rather than a single ‘best’ solution, MCTS is a good choice of search as it randomly samples the solution space, whereas an exhaustive search such as minimax could spend most of its time in a suboptimal part of the search space in which no solutions exist if we are unlucky. The correct balance of exploration versus exploitation in this case is an interesting research question.

5.3 MCTS Experiments

Since solving a puzzle does not result in a win, lose, or draw, the scoring function used was the fraction of the target shape filled. This choice is
potentially problematic as there are many partial solutions in which a single remaining piece does not fit, but it also avoids solutions that obstruct much of the board early on. Like the win/lose/draw scoring function, this technique returns values in the range $0 \leq s \leq 1$, and so is a minor modification of the MCTS technique.

The moves chosen were the pieces that fit starting from the upper-left-most empty square. If a point was reached where no piece could fit at all, the playout stops. The algorithm was allowed to run for 100,000 and 1,000,000 playouts. Thirty independent runs were performed for each of the five target shapes, with the average number of solutions recorded.

Figure 16 shows the resulting 95% confidence intervals on the number of distinct solutions found in 30 runs of the MCTS algorithm for each target shape. For each shape, the lower point is for 100,000 playouts and the higher point is for 1,000,000 playouts. The ‘L’ shape is evidently hardest, and increasing the number of playouts affected it and the Bar shape the least. The Square, ‘T’ and ‘Z’ shapes benefit substantially from more time spent solving the problem. These results suggest that MCTS is a viable method for evaluating the difficulty of target shapes.

6 Generating Level Maps

Dijkstra’s algorithm can be an effective search-based PCG technique for the production of maze-like level maps for games. For example, Dijkstra’s algorithm has been used to evaluate the quality of automatically generated level maps for a pencil-and-paper fantasy role-playing game [4]. Each map represented a scenario in which a Necromancer sends goblins to attack a village and the heroes (player characters) must find the evildoers and chastise them.

Figure 17. Level map generated for our dungeon-based game, using the ‘far’ quality metric.
The level maps were designed with an evolutionary algorithm, as described in [6]. Evolutionary algorithms are algorithms which use selection and variation, in imitation of biological evolution, as search strategies.

Any search algorithm must have some way of evaluating the quality of candidate solutions. For our dungeon-based game, we required certain distances to be long or short, and defined the fitness of a map to be:

$$fitness = \frac{\text{(sum of long distances)}}{1 + \text{(sum of short distances)}}$$

Two quality measures were tested. The ‘far’ metric specifies that goblin and Necromancer lairs should be far from the entrance. The ‘near’ metric specifies that goblin lairs should be near the entrance while Necromancer lairs should still be far from the entrance. In both cases, the armoury should be near a goblin lair.

Figure 17 shows an example level generated while applying the ‘far’ quality measure. The two Necromancer’s lairs (N) and three goblin lairs (G) are all far from the entrance (E), while the armoury (A) is close to a goblin lair. Figure 18 shows an example using the ‘near’ quality metric, in which the goblins’ lairs are much closer to the entrance. This produces a tenser level in which the player will engage in combat more quickly. The generator can therefore be calibrated to produce gentler levels that encourage exploration or harder levels that encourage combat, as desired.

Thus, the goals for which places should be near and far from one another are well satisfied. The ability of Dijkstra’s algorithm to efficiently measure distances in a graph – or in this case an obstructed grid of squares – allows the automatic production of maps with specified tactical properties. Such tools can both pregenerate published content and dynamically generate content during play. This can be useful for increasing replay value, or preventing players from exploiting advance knowledge of a level.

The cited paper outlines a pipeline for creating a combinatorial space of variations on the traditional ‘chastise the goblins’ adventure. It is practical to generate thousands of versions of the goblin adventure in a matter of minutes. This is a proof-of-concept for an adventure module on demand technology. Search algorithms such as minimax and MCTS are core technologies for creating such design pipelines.

Figure 18. Level map generated for our dungeon-based game, using the ‘near’ quality metric.
7 Conclusion

This article introduces a variety of tree-centred search algorithms and applies them to the task of measuring the difficulty of targets for a polyomino puzzle and to the problem of automatically generating level maps. A plethora of other applications exist in both game-play and in game and puzzle design.

It is easy to find articles on solvers for puzzles like Sudoku or Sokoban, but many researchers have not yet realised that a solver can be used as a quality measure for an automatic design system. MCTS is a powerful search algorithm, and while it has mostly been applied to playing games, many other uses are possible [12].

The simple example in this study – using MCTS for game design – demonstrates once such different type of application. MCTS can be used as a virtual play tester for assessing the quality of puzzles and, when partnered with an algorithm that can generate puzzles, used as a technology for automatically designing puzzles.

Acknowledgements

The authors thank the Canadian Natural Sciences and Engineering Research Council and the University of Guelph for their support of this work.

References


Dr Daniel Ashlock is a Professor of Mathematics at the University of Guelph, Canada. His research interests include applications of computational intelligence to both mathematical games and games played by people.

Address: Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada.

Email: dashlock@uoguelph.ca

Dr Cameron McGuinness is a consultant at Ashlock and McGuinness Consulting in Guelph, Canada. His research interests include Monte Carlo tree search, classification of games, and dynamic difficulty in games.

Address: Ashlock and McGuinness Consulting, 41 Lambert Cr., Guelph, Ontario, Canada.

Email: cameron@ashlockandmcguinness.com
A Game Design Approach to a Real-World Problem

Cameron Browne, Queensland University of Technology (QUT)

This short note phrases the game design process as an approach to solving more general real-world problems. I apply this process to the problem of Formula One (F1) point scoring, and the seemingly endless quest to devise a system that both rewards the best driver while maximising the drama for onlookers. A new point scoring system based on a simple linear model is suggested.

1 Introduction

For most followers of Formula One (F1) car racing, the world’s premier class of motor sport, the key point of interest is who will win the World Drivers’ Championship (WDC) each year. Unfortunately, the ultimate winner is decided before the final race of the season more often than not, making the final round(s) less interesting for viewers and resulting in reduced audiences.

To address this problem, the chief executive of the Formula One Group decided to try a new system for the 2014 season that awarded double points for the final race, in a transparent effort to keep the WDC battle alive as long as possible. This rule change achieved its desired result, with the WDC being decided on the final race that year, but was widely criticised by viewers and drivers, and was removed the following year.

There have been a total of thirty different points scoring systems used for the 66 F1 seasons to date. To a game designer, this behaviour is familiar; the owners of the sport are continually tweaking the rules, to see what works and what does not, in an effort find the fairest rule set that maximises the drama.

This is analogous to the play-testing process that a game designer would employ to improve a flawed design. So can a game design approach to the problem of F1 points scoring yield a workable alternative to this real-world problem?

This paper describes the problem, what I mean by a ‘game design approach’, and how it might be applied to offer a workable solution.

2 F1 Scoring Systems

Every year’s F1 scoring system allocates a number of points to the first $N$ drivers who finish each race, from a non-linear distribution that is inflated for the major place-getters. For example, Table 1 shows the non-linear distribution of 101 points used in the 2015 season, and how these were allocated to the first ten drivers to finish each race.

<table>
<thead>
<tr>
<th>Pos</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
<th>9th</th>
<th>10th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pts</td>
<td>25</td>
<td>18</td>
<td>15</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. 101-point system used for the 2015 season.

For the initial 1950s seasons, only the first five drivers were awarded points (with 1 bonus point for the fastest lap). This increased to the first six drivers during the 1960s, and has continued to increase to the first ten drivers from 2010 onwards.

Complicating matters, all systems used up to 1991 only counted a certain number of results for each driver per season. For the 1950–53 seasons only the best four results were counted per driver, for 1954–57 only the best five, in 1958 the best six, in 1959 the best five again, and so on.

Further complicating matters, for the 1967–1980 seasons, each drivers’ WDC points total was determined by combinations of their best and worst results. For example, the 1977 scoring system counted 15 results for each driver; their eight best results from their nine best races and their seven best results from their eight worst races.

For some seasons, points were shared between drivers on a team, or not awarded to drivers who failed to complete a certain distance (even if they finished in the top six), or halved for drivers who failed to complete 3/4 of the race distance, or doubled for the final round, and so on. The optimal point scoring system would:

1. reward the best driver each year
2. keep the WDC championship battle alive for as long as possible each year.

The fact that the system has been changed on average every two years suggests that this balance is hard to achieve, and that better systems

2Describing F1 scoring as a ‘real-world problem’ might be a bit of a stretch. However, it is a topic of interest to hundreds of millions of people.
3There is ongoing debate as to whether the term ‘best’ driver should refer to the most consistent or the most successful in terms of race wins.

are still being sought. We now consider how a game design approach might offer a workable alternative.

3 Game Design Approach

There are various ways to model the game design process [2]. However, at some point in most designs, the basic framework for the game will have been decided, including the equipment that it will be played with and some key mechanisms that it should involve. The task then becomes:

\[ \text{Given some equipment, and relevant mechanisms and constraints, what is the simplest rule set that allows the richest playing experience?} \]

For abstract and mathematical games, the aim is typically to find the simplest set of rules most compatible with the equipment, such that complexity emerges from their interaction rather than having to be specifically stated [9]. The designer may be striving to achieve a desired behaviour, or simply looking for emergent behaviour.

Such optimisation is typically achieved through an extended process of play-testing, iteratively trying different rule combinations to explore the design space through the available degrees of freedom [4]. It makes sense to start with simpler rule combinations, observe their result, then modify as desired.

4 Linear Point System

With these ideas in mind, let us phrase the problem as a game called ‘F1 Scoring’. F1 Scoring is played within the framework provided by the existing F1 establishment, and the basic mechanism involves a finite number of drivers competing in a finite number of races to score points, in order to decide an eventual champion. The desired behaviour is to reward the best driver while maximising the drama of the WDC battle. The simplest rule set to try initially is probably:

\[ \text{Award each driver in each race a number of points equal to their finishing position, in reverse order.} \]

For example, if sixteen cars enter the race, then the winner receives 16 points, second place 15 points, third place 14 points, and so on, down to 1 point for the last position. Cars that do not finish the race can still be assigned a position in the order in which they retired. The number of points awarded each round, and the number of drivers receiving them, will therefore vary each round depending on the number of entrants.

This is similar to the ‘incremental’ system used by the US NASCAR racing body [5], although it allocates points from a fixed base of 40 for the winner decreasing by 1 point per position, with bonus points for the winner, drivers that lead a lap and the driver who leads the most laps.

The proposed linear system is simpler than the NASCAR system, which was itself designed to simplify the scoring process for onlookers. While such simplicity may detract from the general sense of eliteness and mystery that F1 seems to encourage, it could have tangible benefits, as we shall soon see.

4.1 Implications

This simple linear system has the elegance of mathematical purity. It applies consistently in all cases and does not involve magic numbers, arbitrary cut-off points, special bonuses, or non-linear relationships to be fine-tuned. However, some implications must be considered.

4.1.1 Competitiveness

Would drivers still be motivated to achieve their best result each race if the leaders’ points are not inflated? Probably, given their competitive nature and the monetary rewards involved. Further, the linear system would actually increase the relative value of each WDC point, rather than de-value them through inflation, so that gaining one or two positions would be more important for every driver on the track rather than just the few leading the race.

4.1.2 Points for All

All scoring systems to date have only awarded points to the first \( N \) drivers, making WDC points difficult to achieve for minor teams and something of a milestone to aim for. The linear system would level the playing field so that all entrants receive points, which would make the ‘WDC points club’ much less exclusive. I do not know what impact this would have on drivers, teams, sponsors, viewers, etc., although simply getting a team and its car to the point of competing in F1 races must surely be an achievement in this sport.

---

\[4\] This is less relevant for other types of games, such as war games or role-playing games, in which the length and complexity of the rules themselves can be part of the immersive playing experience.

5 Results

As with any rule tweak, the proof is in the playing: how well would our revised game of F1 Scoring actually work? To evaluate the proposed system, I obtained the historical results of all 66 F1 seasons from 1950 to 2015 and re-calculated the WDC points table on a round-by-round basis using the linear system, determining the number of drivers who could still theoretically win the WDC at the end of each round.

Figure 1 shows a comparison between these linear results (dark blue, back row) and the actual scoring systems used (light blue, front row) for the last five rounds of each season. The height of each stack indicates the number of potential WDC winners following each round (capped at 10 for visual clarity). Using linear point scoring, all seasons produce a clear winner following the final round, except for the 1978 and 1990 seasons which produce ties.

5.1 Drama

For each round of each season, the higher the stack shown in Figure 1 the better, as higher stacks indicate less certainty of who will become eventual champion. This is analogous to the desirable quality of drama in adversarial games [19].

The results suggest that linear scoring would

---

6 From the Ergast database: http://ergast.com/mrd/

5.2 Fairness

The predominance of dark blue over light blue in Figure 1 suggests that, in general, many more drivers would indeed be in WDC contention for longer using simplified linear scoring. But would it also respect their relative order of merit to identify the best driver each year?

It would appear so. Both systems – actual and linear – did in fact produce the same winners for almost all seasons, and in the rare cases this did not occur the champion produced by the linear system was still a high place-getter in the actual results. The results for the top ten drivers were in most cases very consistent, with increasing deviations further down the list.

For example, Table 2 shows the WDC results for both the actual system used and the proposed linear system for the controversial 2014 season, in which double points were awarded in the final race to artificially extend the WDC battle. Both lists are identical in the top four positions, with the linear list showing smaller differences between positions.

<table>
<thead>
<tr>
<th></th>
<th>Actual Result</th>
<th>Linear System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>384 Hamilton</td>
<td>348 Hamilton</td>
</tr>
<tr>
<td>2nd</td>
<td>317 Rosberg</td>
<td>344 Rosberg</td>
</tr>
<tr>
<td>3rd</td>
<td>238 Ricciardo</td>
<td>=310 Ricciardo</td>
</tr>
<tr>
<td>4th</td>
<td>186 Bottas</td>
<td>=310 Bottas</td>
</tr>
<tr>
<td>5th</td>
<td>167 Vettel</td>
<td>291 Alonso</td>
</tr>
<tr>
<td>6th</td>
<td>161 Alonso</td>
<td>286 Vettel</td>
</tr>
<tr>
<td>7th</td>
<td>134 Massa</td>
<td>273 Button</td>
</tr>
<tr>
<td>8th</td>
<td>126 Button</td>
<td>259 Hülkenberg</td>
</tr>
<tr>
<td>9th</td>
<td>96 Hülkenberg</td>
<td>245 Massa</td>
</tr>
<tr>
<td>10th</td>
<td>59 Pérez</td>
<td>239 Magnusson</td>
</tr>
</tbody>
</table>

Table 2. WDC results for the 2014 season.

Figure 2 shows the total WDC contenders using both scoring systems for all 2014 rounds. It can be seen that the linear scoring system (dark blue, back row) could have naturally produced a more dramatic battle down to the final round, without the need for additional rules to artificially induce this behaviour.

5.3 Caveats

The results presented above must be taken with a grain of salt (a big one!), as the drivers’ behaviour is of course highly dependent on the point scoring system currently in use each year, and both drivers and teams will react to rule changes and use a variety of strategies in order to maximise their rewards. We cannot say that the alternative results shown would actually have occurred if the proposed linear system had been used.

It could be that drivers who might otherwise have pushed harder to achieve inflated leader points might be more content with a lesser position under the linear system. But I doubt that this would be the case, for the reasons given in Section 4.1.1.

Another concern is how the linear system might be prone to exploitation by drivers and teams. However, its simplicity makes it robust. The only way to manipulate the points awarded each round would be to affect the number of entrants, which is largely outside their control, or to affect the order of finishers, for which the governing F1 body already has a number of mechanisms in place.

An unexpected advantage of the linear scoring system is that the uncertainty in the WDC battle could be artificially increased simply by increasing the number of entrants in subsequent races. For example, additional cars might be added to later rounds to keep things interesting.

5.4 Variations

The basic rule set for linear point scoring outlined above appears to have some merit. But any good game designer should try obvious rule variations
to look for further improvement [4]. Even subtle
design changes can introduce nuances that affect
the way that a game plays, so multiple alterna-
tives should usually be considered. Can our game
of F1 Scoring be further improved?

One simple tweak would be to only award
points to the ten best finishers, as is the case with
the actual systems currently used. However, do-
ing so reduces the resulting drama until there
is no observable benefit. It appears that the in-
creased spread caused by the greater numbers
involved is central to the success of the linear sys-
tem. 24,490 drivers (±0.640 at the 95% confidence
level) started each race on average over all 935 F1
races.

Another simple tweak would be to use non-
linear point allocations, such as squaring points
or taking their square root. Neither of these ap-
proaches showed any observable improvement
over existing systems.

Linear systems enhanced with bonus points
for leading or winning races, such as that used
by NASCAR, improved over existing systems but
led to slightly less dramatic WDC battles in gen-
eral than the proposed simplified linear system.
The simplest rule set tried appears to achieve the
best results in this case.

This last result has relevance to the debate
regarding whether the ‘best’ driver should be the
most consistent or the most successful. Awarding
bonus points to race winners might be an incen-
tive for drivers, but appears to reduce the overall
drama of the WDC battle, as the more successful
drivers outscore the rest of the field more quickly.
If prolonging the drama of the WDC battle is a
priority, then it appears that consistency is prefer-
able to success.

6 Conclusion

I do not actually know much about F1 racing, and
the above analysis may appear woefully naive to
experts in the field. This exercise was intended
more as a thought experiment about how a game
design approach might be applied to other types
of problems, to suggest new insights and (hope-
fully) workable solutions.

The simplest rule set tried for the given do-
main of F1 point scoring suggests attractive be-
aviour in terms of enhancing the drama in the
world championship battle each season, without
unduly impacting the fairness of the results. It
may not be a system for all seasons, but appears
to come close.

Acknowledgements

This work was supported by a QUT Vice-
Chancellor’s Research Fellowship. Thanks to
Russ Williams, Robin Barrett and Néstor Romeral
Andrés for useful comments that led to significant
improvements.

References

[1] Press Association, ‘F1’s Final Race Double
Points in GPs Scrapped After Only One
https://www.theguardian.com/sport/2014
dec/03/world-motor-sport-council-f1-
double-points
Creativity in a Closed Game System’, in Pro-
cedings of the IEEE Conference on Compu-
tational Intelligence and Games (CIG’12), Malaga,
Game & Puzzle Design, vol. 1, no. 1, 2015,
pp. 83–86.

Cameron Browne is a Vice-Chancellor’s Senior
Research Fellow at QUT, Brisbane, whose re-
search interests include artificial intelligence
and automated game design.
Address: School of EECS, Science and Engi-
eering Faculty, QUT, Brisbane, 4001, Australia.
Email: c.browne@qut.edu.au
What’s a Ludeme?

David Parlett, Gourmet Games

The word ‘ludeme’ does not (yet) appear in any dictionary and therefore has not established claim to any agreed definition. The purpose of this article is to explain my personal interest in it, to justify my use of it, and to offer some thoughts towards its definition. This is a revised version of the original article posted on my web site The Incompleat Gamester in 2007.

1 Introduction

A ludeme is an element of play, comparable to, but distinct from, a game component or instrument of play. Kings, queens, bishops, knights, rooks, pawns, and a chequered board, all constitute the instruments of play or the components of the game of Chess. Ludemes are the conceptual elements of the game, most typically equivalent to its ‘rules’ of play. For example, whereas the material piece shaped like a horse and designated ‘knight’ is a component of the game, the distinctively skewed move of a knight is a ludeme of the class ‘rule of movement’. But other types of ludemes also exist. For example, the name, referent and associated connotations of ‘knight’ – those of a chivalric courtier – may be said to constitute a thematic ludeme.

A characteristic property of ludemes is their propensity to propagate by passing not only from one game to another (the long diagonal move of the bishop is not unique to Chess but occurs also in continental Draughts) but even between games of entirely different classes. Thus the thematic ludeme of a knightly figure is not unique to Chess but also occurs in card games. Similarly, but perhaps more abstractly, it might be said that whereas an actual chequered board is an instrument of play, or game component, the idea of chequering a board so as to render diagonal moves more comprehensible is itself a conceptual component or ludeme.

2 Memes and Ludemes

The passage of ludemes from one game or game type to another brings to mind the definition of a meme, such as this from the Oxford English Dictionary:

A cultural element or behavioural trait whose transmission and consequent persistence in a population, although occurring by non-genetic means (esp. imitation), is considered as analogous to the inheritance of a gene.

The term was coined by Richard Dawkins in The Selfish Gene (1976). In his own words:

The new soup is the soup of human culture. We need a name for the new replicator, a noun which conveys the idea of a unit of cultural transmission, or a unit of imitation. ‘Mimeme’ comes from a suitable Greek root, but I want a monosyllable that sounds a bit like ‘gene’. I hope my classicist friends will forgive me if I abbreviate mimeme to meme… Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches.’

If Dawkins proposed the term in 1976, Alain Borvo could well have first published the term ‘ludeme’ in 1977, as Pierre Berloquin implies. But it doesn’t follow that he was directly influenced by the Dawkins ‘meme’. In this connection, Thierry Depaulis notes:

Now, whoever invented the word, it must have been coined in the 1960s or early 1970s when structuralism reigned in France and put into fashion words like phoneme, morpheme, grapheme, mytheme, lexeme, and the like. (Whence ‘mimeme’, certainly influenced by the same philosophical background.) I am not in error in saying that the man who invented ‘ludeme’ thought of it as a ‘logic micro-structure’ or ‘basic component of a game structure’.

3 Phonemes and Ludemes

In retrospect, my understanding of the term ludeme as expressed in the Oxford Guide to Card
Games [12] seems surprisingly apt, considering that I had not encountered the word ‘meme’ at the time. Rather, since my main subject is language, the word ‘ludeme’ sounded to me more like a construction analogical to that of the term ‘phoneme’, which dates from the end of the 19th century and is defined by the Oxford English Dictionary [2] thus:

A phonological unit of language that cannot be analysed into smaller linear units and that in any particular language is realized in non-contrastive variants... Although its exact nature is disputed, [...] the phoneme remains a standard taxonomic unit in the description of speech.

The bit about ‘non-contrastive variants’ may be explained as follows. The sound represented by the letter ‘T’ in English may vary from region to region, speaker to speaker, or even from one position to another in a word. For example, it may or may not be aspirated, as it most noticeably is in English spoken with an Irish accent. Any distinctive way of pronouncing it may be represented by a distinctive international phonetic symbol. But all are ‘allophones’ of a single phoneme /t/. What makes it a single phoneme is that if you pronounce ‘cat’ with any allophone of /t/ the word will always be perceived as denoting a cat. If, however, you voice /t/ it becomes /d/, and anyone pronouncing ‘cat’ in such a way will be perceived as referring to a cad. In other words, /t/ and /d/ are contrastive variants: change the phoneme and you change the meaning of the word.

Similarly, game elements may be contrastive or non-contrastive, and are ludemes only if they are contrastive. For example, many card games involve a hierarchy of suits for bidding purposes. In Bridge, spades > hearts > diamonds > clubs, in Preference it is hearts > diamonds > clubs > spades. The concept of suit hierarchy, as such, is a ludeme. It contrasts with a game in which one suit is always trump, such as the eponymous Spades. Here, the idea of fixing on a suit as permanent trump is a different ludeme. If you introduced it into Bridge or Preference, it would no longer be the same game, at least as far as bidding is concerned. By contrast, however, the actual order hearts > diamonds > clubs > spades, or any other of 24 possibilities, is not a ludeme because it is not contrastive. If you switched the order around as between Bridge and Preference, you would make no difference to any aspect of the play. The thinking involved would not be altered.

The same would apply if you simply reversed the order of ranks, so that in each suit Deuce was high and Ace low. Such variations are merely transformations of the same ludeme. If, however, you decreed that the ranking of cards should be upside down in trumps but not in side suits, then you would have introduced a novel ludeme and thereby significantly changed the game, at least as far as assessing your hand and conducting an auction are concerned (albeit not in the ensuing play of tricks).

To summarise, a ludeme or ‘ludic meme’ is a fundamental unit of play, often equivalent to a ‘rule’ of play; the conceptual equivalent of a material component of a game. A notable characteristic is its mimetic property – that is, its ability and propensity to pass from one game or class of game to another.

4 Origins of the Term

My interest in the term ‘ludeme’ arises from the fact that, if you enter it into a search engine such as Google, one of its earliest mentions occurs in a page called The Ludemic Game Generator [3] which is introduced as follows:

Inspired by passing mention of the word ‘ludeme’ in David Parlett’s ‘The Penguin Book of Word Games’ [5], The Ludemic Game Generator randomly combines categories and mechanics from those at BoardGameGeek to create new (and largely useless) game ideas...


Interesting that they use the term ludeme, which I had only seen in two places before – in David Pritchard’s book, and on my domain name!

Two comments. First, I do not recollect using the term in my Penguin Book of Word Games (1982) [5], but I certainly did use it, I think for the first time, in my Oxford Guide to Card Games (1990) [12], where I (rather disparagingly) reported an ascription of its coinage to Pierre Berloquin. Second, I do not think the late David Pritchard ever used it, but he and I were often amused by examples of confusion between our two names, and invariably corresponded with each other in terms of ‘To David P. from David

3http://kevan.org/ludeme
4https://boardgamegeek.com/thread/60120/grammar-gameplay
P.’ So it is possible that Sumo was referring to me rather than David Pritchard.

This brings us to A Grammar of Gameplay\[3\], subtitled ‘Game Design Atoms, Can Games Be Diagrammed?’, notes for a talk given by Raphael Koster, former Chief Creative Officer of Sony Online Entertainment, at a Game Developers Conference held in Austin, Texas, in 2005. I say ‘notes for a talk’ rather than the text or transcript of a talk, as the relevant item, also characterised as a ‘Quest for a universal notation of game play and design’, is presented on the web site as a series of notes and diagrams rendered as images, making referencing and word-searching virtually impossible. It consists of a series of cartoon-illustrated gnomic statements rather than connected argument. Typical are:

This talk owes a lot to the concept of ‘ludemes’. Ben renamed them ‘primary elements’. They are similar to ‘choice molecules’, but Ben likes empiricism, not theory.’

The most basic ludemes involve a user interface interaction.
The pieces in chess. They aren’t content, they are verbs.
The most basic sort of variable feedback is your opponent’s move.
Is there skill and risk involved in using an ability? If not, is this an atomic unit of gameplay?

Further reaction to Koster’s talk emanates from Stéphane Bura (Creative Director, Elsewhere Entertainment, 10Tacle Studios, Belgium), whose online article ‘A Game Grammar’ begins:

The goal of this document is to present a grammar enabling game designers to describe games in a useful manner. By useful, I mean that this grammar should allow us to:

- Communicate the underlying principles of a game: How are the different parts of the game linked? What kind of interaction is there between the players? Are there winning strategies or exploits? Etc.
- Doodle a game on a napkin, as Raph puts it.

After working on the problem a bit, I came to the conclusion – as Raph posits – that a useful game grammar would mostly describe constitutive rules, as defined by Katie Salen and Eric Zimmerman in ‘Rules of Play’\[4\]: ‘The constitutive rules of a game are the underlying formal structures that exist ‘below the surface’ of the rules presented to players. These formal structures are logical and mathematical’. Working at this level also requires some assumptions, such as assuming that all the players follow the implicit rules (etiquette). For instance, in a competitive game, all players must play to the best of their abilities to win.

So far, it would appear that ludeme means a ‘primary element’ or ‘atomic unit’ of play. Bura uses it further down on the same page, but it is not entirely clear to me exactly in what sense.

5 Conclusion

I do not claim to have invented the word ‘ludeme’; I not only credited it to its presumed inventor when I first used it but have since made enquiries as to its origin through other contacts in France (albeit without success); and I make no claim to its universal indispensability. I use it because I find it useful, and its meaning has always seemed plain enough to me from its context. What more can one ask of a word?

Acknowledgements

This article has previously been posted on Parlett’s web site The Incompleat Gamester in 2007 under the title ‘What’s a Ludeme? – and who really invented it?’\[7\]. The editors would like to point out that the piece is mostly reprinted verbatim here, but has been reformatted for publication. This includes reordering the content so that the definition of the term ‘ludeme’ comes first, adding appropriate references, and removing some asides by the author that were not central to the discussion.

References


\[6\]http://users.skynet.be/bura/diagrams/
**Gloop Challenges #13 and #14**

Pack the tiles on the right into the grid to form a single closed contour. Gloop is described on pp. 31–32.

**Gloop Challenge #13**

![Gloop Challenge #13](image1)

**Gloop Challenge #14**

![Gloop Challenge #14](image2)
Vol. 3, no. 1

Editorial
Talking About Other People’s Games ............................................ C. Browne 3

Nikoli Logic Puzzles
Herugolf ................................................................. Y. Anpuku 5

Article
Semantic Hashing for Video Games ....................... A. Isaksen, C. Holmgard and J. Togelius 10

Maths in Games
Limping Boards for Games ................................................. C. Browne 17

Case Study
The Complex $3 \times 3 \times 3$ ........................................ C. Hoff 25

Short Note
From Mathematical Proof to Puzzle ......................... N. Romeral Andrés 39

Maths in Games
CliqueR: A Graph Theory Game .................. D. Ashlock and A. McEachern 42

Game Design Patterns
Reinvent the Wheel .................................................. C. Browne 51

Article
Games for Three Players ............................................. J. Neto and W. Taylor 63

From the Archive
Some Random Thoughts On Chance and Skill ..................... D. Parlett 74

The front cover shows a nested Gadeiro design by Néstor Romeral Andrés. See pages 39–41 for details.
Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com

Editor-in-Chief
CAMERON BROWNE
Department of Knowledge Engineering (DKE)
Maastricht University
Maastricht, 6200 MD
Netherlands
cameron.browne@maastrichtuniversity.nl

Liaison Officer
NÉSTOR ROMERAL ANDRÉS
Nestorgames
Spain

Copy Editor
RUSS WILLIAMS
Green Star
Poland

Editorial Panel
INGO ALTHÖFER, Univ. Jena, Germany
DANIEL ASHLOCK, Univ. Guelph, Canada
KRIS BURM, Don & Co., Belgium
TRISTAN CAZENAVE, Univ. Paris-Dauph., Paris
BEN COUSINS, Scattered Entertainment, UK
KERRY HANDSCOMB, Vanc. Sch. Board, Canada
RYAN HAYWARD, Univ. Alberta, Canada
CARL HOFF, Applied Materials, USA
KATE JONES, Kadon, USA
MAKI KAJI, Nikoli, Japan
DON KIRKBY, BC HIV/AIDS, Canada
SIMON M. LUCAS, Univ. Essex, UK
DAVID MOLNAR, Univ. Puget Sound, USA
ANDY NEALEN, NYU Poly, USA
JOÃO PEDRO NETO, Univ. Lisbon, Portugal
DAVID PARLETT, Parlett Games, UK
RAF PEETERS, Smart, Belgium
RICHARD REILLY, Blinn College, USA
SPYRIDON SAMOTHRAKIS, Univ. Essex, UK
GREG SCHMIDT, Axiom, USA
SANDRA SNAN, Idiomdrottning, Netherlands
DIETER STEIN, Spielstein, Germany
STEPHEN TAVENER, Scat Ltd, UK
BRUCE WHITEHILL, Big Game Hunter, Germany
MARK WINANDS, Univ. Maastricht, Netherlands
I-CHEN WU, Chiao-Yung Univ., China

Game & Puzzle Design is an independent publication prepared and published with the kind help of volunteers. Set in Palatino using LaTeX.

Published content © 2018, Game & Puzzle Design.

Authors retain the rights to their source material and proof copies, but by submitting it to the journal agree to its publication in a form deemed suitable by the editors. Views expressed in the published content are those of the submitting authors and not necessarily shared by the editors. Authors are referred to the submission guidelines: http://www.gapdjournal.com/submit
Talking About Other People’s Games

Cameron Browne, Queensland University of Technology (QUT)

ANY designers like to talk about their games. There is an element of vanity and self-interest in this, as designers are usually proud of their creations and happy to show them off, and in many cases if the designer does not promote their game then nobody will. But there is also a more pragmatic reason: these are the games that the designer knows the best. In fact, the designer is typically the only person in the world who knows the full story behind each game, from inception to release.

At the same time, many designers are reluctant to reveal the underlying principles that make their games successful. This may be for professional reasons, a desire not to spoon-feed others, or simply an inability to articulate the finer details of the processes involved. In any event, the people who know their games the best are often reluctant to talk about them. This can be a problem for people like me who like to analyse games.

Design Constraints

Looking at other people’s games from the outside, we can make some educated guesses about certain design decisions, but these are just guesses. We can of course not deduce what the designer was thinking or aiming to achieve, or know what constraints they were working under.

Such constraints include the desire to work with predefined equipment, such as in Schmittberger’s classic New Rules for Classic Games, which is referred to several times throughout this issue, or to find games for given equipment for economic or ecological reasons, e.g. to reuse offcuts from the manufacture of other games.

Other designers constrain themselves by working towards certain game behaviours, although both approaches – equipment-based and behaviour-based – can be seen as two aspects of the same overall design process. Designing under such constraints can focus the search and actually inspire greater levels of creativity, but it can also lead to unaccountable design decisions that would not otherwise have been taken.

What Can We Learn?

So what can we learn by studying other people’s games? One of the beauties of many types of games is that they are self-contained entities that exist apart from their creators. In a Platonic view of the world, games are like mathematical truths just waiting to be discovered, and stand apart from their process of discovery.

The role of the games scholar can be likened to that of the art critic or film reviewer, or commentator on any art form. And like a good piece of art, a good game provides room for study and interpretation. We can deduce certain aspects of other people’s games from the qualities they exhibit. But it is almost always interesting to hear designers talk about their own games, to get the real inside story.

This Issue

This issue opens with the welcome return of the Nikoli Logic Puzzles column, after the recent retirement of long-time Nikoli correspondent Jimmy Goto. We thank Nikoli chief editor Yoshinao Anpuku and translator Ken Shoda for their efforts in continuing the series. This instalment describes ‘Herugolf’, a pure logic puzzle that cleverly captures the flavour of golf, in a rare case of a successful logical abstraction of a physical game. The designer outlines the process that led from the initial conception of the idea to the final game.

Isaken, Holmgård and Togelius then describe the use of a deep learning approach for the generation of levels for two recent classic video games – The Legend of Zelda and Super Mario Bros – in their article ‘Semantic Hashing for Video Game Levels’. This is the first Game & Puzzle Design paper to focus on video games since the first issue, which we hope marks an increase of submissions in this direction.

My paper ‘Limping Boards for Games’ describes the generalisation of a simple mathematical concept, and how this principle can be usefully applied to the design of game boards. I present several examples of games that demonstrate the benefits of this principle. Contrary to the above Editorial’s key point, none of these are games of my own design. But most are games that I have played and studied for many years, and obtained the designer’s inside comments where possible.

Carl Hoff continues his investigations into the computer-assisted design of (physical) mechanical puzzles in ‘The Complex 3×3×3’. This paper is Carl’s most – dare I say it – complex yet, and demonstrates the extraordinary degree of analysis and problem-solving that this particular puz-
Game & Puzzle Design

Vol. 3, no. 1, 2017

zle has required over many years. There are few other designers who could look at Carl’s design from the outside and deduce the steps that led to it, or appreciate the amount of work that has gone into it and continues to go into it.

In ‘From Mathematical Proof to Puzzle’, Néstor Romeral Andrés describes the genesis of his Gadeiro puzzle from a simple mathematical observation, showing that game designs can occur where we least expect them. This short note proved quite fruitful, providing both this issue’s cover image of an infinite Gadeiro series and its ‘feature puzzle’ (shown below). Each challenge was handcrafted by the designer, and chosen for symmetry and visual interest.

Daniel Ashlock and Andrew McEachern describe their family of CliqueR games in ‘CliqueR: A Graph Theory Game’. These were designed to introduce high school and university students to basic graph theory, so have a very mathematical basis. The authors describe a very focussed design process that is typical of developing such ‘Games With A Purpose’.

My article ‘Reinvent the Wheel’ investigates the notion that modern game designers have a limited pool of core mechanisms to work with, and since it is likely that the optimal form has already been found for most of these, then it can be more fruitful to start with known mechanisms and work away from them. I offer dozens of examples of games demonstrating this principle, including some of my own.

João Pedro Neto and William Taylor describe different ways to mitigate problems with non-strategic coalitions in ‘Games for Three Players’, using several of their own excellent board games as examples. This paper complements several other papers on the topic in previous issues by various authors, making this the most popular/urgent design issue among our contributors.

This issue concludes with a revised version of David Parlett’s observations on the relationship between chance and skill in games, in ‘Some Random Thoughts On Chance and Skill’. Games can involve many different types of uncertainty, and require many different types of skill, and Parlett sheds some interesting light on these distinctions.

References

Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, Australia, whose research interests include artificial intelligence and automated game design. Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia. Email: c.browne@qut.edu.au

Gadeiro Challenge #1

Pack the pieces on the right to fill the shape on the left. Gadeiro is described on pages 39–41.
Herugolf

Yoshinao Anpuku, Nikoli

Herugolf is a logic puzzle in which the solver must hit a number of golf balls from their tees to matching holes, while avoiding water hazards. It is a simple puzzle with a strong visual aspect.

1 Introduction

Herugolf is a Japanese logic puzzle [6] by designer Alkali-Kojo-Cho, based on a golf theme. It first appeared in 2013 in Puzzle Communication Nikoli issue 142, where it was originally called ‘Pro-Golfer Maru’ after popular Japanese comic series Pro Golfer Saru which featured a champion monkey golfer.

Few would have thought that a physical sport as nuanced as golf could successfully be turned into an abstract logic puzzle. However, Pro-Golfer Maru was well-received from the start, probably due to its originality and the quirkiness of its name. It has been especially popular with young men, perhaps due to the sporting theme and the fact that manoeuvring the balls neatly from tee to hole gives a feeling of ‘nice shot’

A year and a half after its first appearance, the puzzle became a regular item in Puzzle Communication Nikoli from issue 148 onwards, where its name was changed to Herugolf (the prefix ‘heru’ means ‘decrease’ in Japanese). Nikoli has recently published a book of Herugolf [2].

2 Rules

The aim in Herugolf is to hit each ball from a tee with a given number into a corresponding hole, while avoiding obstacles including water hazards.

The number defines the distance of each shot, and the player must deduce which hole each ball must go in. Figure 1 shows an example challenge (left) and its solution (right) from [2].

Rules of Herugolf

1. Hit each ball from its tee (circled) one or more times to reach a hole (H). Every ball must reach a different hole.
2. Each hit is shown by an arrow to its destination cell. Arrows can not cross tees, holes, lines of other arrows, or backtrack.
3. The first hit travels as many cells as the circled number, and each successive hit travels one less cell than the previous hit. Balls can only travel orthogonally, but may change direction after each hit.
4. Balls cannot leave the grid or stop in any Ike-Pocha (water hazard, shaded).

3 Worked Example

Herugolf has more rules than most Japanese logic puzzles, but can still be solved using basic deduction and does not require overly sophisticated solution techniques. This worked example of the challenge shown in Figure 1 demonstrates the basic strategies.

Figure 1. A Herugolf challenge (left) and its solution (right).
Firstly, identify those tees with only one possible shot. The 5 tee in the top left corner only has one possible shot (downwards), as a shot to the right would land in a water hazard. Similarly, the 3 tee towards the bottom right corner only has one valid shot (upwards). Figure 2 shows these forced shots.

The shot from 5 must then turn to the right (shots cannot reverse to backtrack themselves), then turn upwards to avoid crossing the other arrow, then turn left to avoid crossing the 3 tee or landing in a water hazard. The shot from 2 tee in the bottom left is now also forced (Figure 3).

In a well-designed challenge, each shot will reveal information that can be used to make further progress. For example, we are now able to identify forced shots from the other two 3 tees and make them, and extend the existing shots, as shown in Figure 4.

This example demonstrates a strong sense of dependency. Most of the information required to solve the challenge is hidden in plain sight, and it is up to the solver to find the initial forced moves, which reveal further forced moves, which reveal further forced moves, and so on.

## 4 Design

The designer of Herugolf, Alkali-Kojo-Cho, was initially thinking about new non-pencil-and-paper puzzles for Nikoli when the idea for Herugolf occurred to him. It was inspired by the board game Ricochet Robots by Alex Randolph, in which players move pieces to reach a goal, when Mr Alkali made the connection between this basic mechanism and the theme of golf. The question then became how to model as many aspects of golf as possible in a simple logic puzzle, for

---

which Mr Alkali also took inspiration from the video game Sid Meier’s SimGolf.

Herugolf incorporates many aspects of golf to capture the spirit of the game:

- Players hit balls from tees.
- Every tee has a hole (and vice versa).
- Balls should not land in water, cross other fairways or greens, or whizz over the heads of other golfers at other tees.
- Each tee has a distinct fairway leading to its hole, separated from all others.

The 1:1 mapping between tees and holes makes each challenge a self-contained golf course. For example, Figures 1-5 show a 7-hole golf course packed into an 8 x 8 grid, in which the balls played on each fairway do not interfere with each other.

The fact that shot distances decrease by one with each hit is the key mechanism that makes the game work. It is also a nice abstract of real golf, in which players typically hit the longest shot from the tee, then shorter and shorter more refined shots as they approach the hole.

The designer also considered including other elements of golf into the puzzle to deepen the ‘golf experience’, including sand banks, roughs, trees, and other obstacles. But in the end, he found the base constraints sufficient to allow interesting challenges to be devised, so omitted all obstacles except water hazards for the sake of simplicity.

Whenever Nikoli releases a new puzzle, there is usually a period in which other designers submit slightly modified forms of their own. Mr Alkali expected that some other designer would devise a version that at least included sand banks, but none did, so his rules became the final design.

4.1 Symmetry

As with all Nikoli puzzles, each challenge exhibits a strong sense of dependency and has a unique solution, and also shows a strong sense of symmetry or pattern in its design. This is more an aesthetic concern than a practical one, but is still important to us, as it demonstrates skill and flair, and elevates our puzzles from mathematical exercises to an art form. A well designed challenge imparts some of the personality of its designer, and can be very attractive to players.

Symmetry in Herugolf applies strictly to the water hazards, as these are the visually dominating features that define the shape of each golf course, rather than the tees and holes which are the finer detail. Herugolf is not strategically heavy and does not require sophisticated methods to solve, so designers typically focus on the visual interest of their challenges rather than their complexity.

The two examples shown in the Appendix demonstrate this visual emphasis. Figure 8 shows a symmetrical and highly unusual golf course, while Figure 9 spells out the relevant message ‘HOLE in ONE’.

While another of our puzzles, Tentai Show, reveals dot-pictures after it is solved, Herugolf provides such pictures before it is solved. Some Nikoli readers report that they enjoy simply seeing Herugolf challenges as much as solving them.

5 Conclusion

Herugolf is a simple logic puzzle that captures the essence of golf through a minimum of rules. Its focus on visual design rather than complexity makes it – like golf – a leisurely pastime.

Acknowledgements

Thanks to independent translator and game designer Ken Shoda for translating the original notes from Japanese, and the Editor-in-Chief for editing, formatting and illustrating this piece.

References


Yoshinao Anpuku is the chief editor of Japanese quarterly magazine Puzzle Communication Nikoli. Address: Nikoli Co. Ltd., 3-36-5 Nihonbashi-hamacho, Chuo-ku, Tokyo 103-0007, Japan. Email: an@nikoli.co.jp

Appendix: Large Examples

The following two challenges demonstrate the beauty of Herugolf. Figure 8 shows a design by Alkali-Kojo-Cho from our Herugolf book, which highlights our preference for symmetrical hint layouts in this unusual golf course with many parallel canals.
Figure 6. A symmetrical $24 \times 14$ Herugolf challenge by Alkali-Kojo-Cho [2].

Figure 9 by The Ax and Sword, also from [2], demonstrates a bit of humour on the part of the designer by encoding a message in the golf course design. Is there actually a hole-in-one in this challenge?
Figure 7. A 36 × 20 Herugolf challenge by The Ax and Sword, with a particular message [2].
We use semantic hashing, an unsupervised machine learning technique based on a type of deep neural network called autoencoders, to categorise video game levels. We show how this technique can be used on dungeon room maps from The Legend of Zelda and segments of 2D-platformer levels from Super Mario Bros. We also discuss how this technique can be useful for game designers, AI-assisted game design, and procedural content generation.

1 Introduction

Semantic hashing is an automated, non-linear method for finding similarities between instances of content using pretrained neural networks. The technique was originally developed for determining document similarity for searching through text documents to find related examples [1]. In this paper, we show how a variant of the semantic hashing technique [2] based on autoencoders [3, 4] can be used to find similarities in video game content.

Semantic hashing uses unsupervised learning to perform clustering of similar content. Unsupervised means the data does not need to be labelled by a game designer or player before processing. Instead, the system can take in level data – segmented into regions such as rooms in a dungeon or slices of a linear level – and determine which regions are similar given no additional knowledge of what the level data means.

Once a semantic hashing neural network is trained, it is simple to input new level data and have the network return the category which the new data is closest to. Thus, the network also allows us to classify unseen future data using the same structure.

We demonstrate the use of semantic hashing on two types of classic video game levels: top-down dungeon RPG maps from The Legend of Zelda [5] and side-scrolling platformer levels from Super Mario Bros. [6]. Level data is obtained from the Video Game Level Corpus [7], which takes level maps and preprocesses them to load into machine learning algorithms.

1.1 Application to Game Design

While this paper demonstrates a proof of concept to show that semantic hashing can effectively classify and cluster new game content automatically and quickly, we believe this method has a number of potentially useful applications in game design, procedural content generation, and user-generated content.

To begin, the statistical analysis of game content can help a designer understand which game content is common, unique, and under/over-represented in their design. For example, a designer may want to know how much variety and repetition there is in the game. Similarly, multiple designers working on the same game might wish to compare their content to see how much of it is similar. In a mixed-initiative system, an AI agent may offer suggestions to a human designer based on the type of work they are currently creating – to enable this, the machine must be able to identify what makes content similar or unique.

Procedural content generation allows a machine to generate new game assets [8], and a large category of such methods use machine learning to produce new content from existing game data [9]. This enables new methods in autonomous generation, co-creation, mixed-initiative design, and repair of content for games. With automated content classification, one can control the output of a procedural content generation system; e.g. if we wanted to generate levels with many bridges, we could use the classifier to check for this. Or, more generally, we could use an autoencoder to ensure that machine generated content has a certain typological distribution, ensuring variations or patterns in the output.

In games where players contribute content, semantic hashing could inform the game what the players are creating by assigning the new content to the most likely category already in the game. The autoencoder could learn these categories automatically from the existing game content, and can be retrained as more is created.
2 Semantic Hashing

Semantic hashing is based on autoencoders, which are a type of neural network. Neural networks are trainable non-linear mathematical models; most neural networks take an input vector (Figure 1a) and generate an output vector (Figure 1g). By providing input data and matching output data, the back-propagation algorithm can learn the parameters of the neural network to create a non-linear function. Neural networks are typically structured in layers, where each layer is a simple linear function with an input and output, and the layers are linked together to compose a deep neural network implementing a more complex non-linear function. Each layer can be of a different size, and the input and output vectors can have different sizes as well.

We show an autoencoder neural network in Figure 1. An autoencoder neural network is trained to reproduce as output a vector identical to the input. The input data is fed into the first layer of the neural network (Figure 1b) which is fully connected to successive layers of smaller and smaller hidden layers (Figure 1c). By fully connected, we mean that the entire output vector from layer $i$ is given as input to layer $i + 1$.

In the middle of the network is a narrow choke-point, the binary encoding layer (Figure 1d). This narrow layer forces the network to represent the input in an abstract and compressed form, so that it can come up with a more general model of the data. From this point, the network expands again through more hidden layers (Figure 1e) to the full-sized output layer (Figure 1f).

The network tries to produce the same output as input, which is difficult because the data must pass through the lossy narrow encoding layer. If we keep the size of this layer small, and encourage the values that pass through this layer to be 0.0 or 1.0, we can treat this layer as a hash value that categorises the data. If the encoding layer is $k$ bits wide, we can represent $2^k$ different categories. Because the training tries to reduce the difference between the input vector and output vector, the network will eventually produce a best-fit classification on the data it is trained on.

Between each layer, we also pass the data through what is called an activation function. It takes the output of each neuron and performs a simple non-linear transformation on it. We commonly use the “rectified linear unit” (ReLU) activation, which is simply $f(x) = \max(0, x)$, such that positive values are unmodified and negative values are clamped to 0. We also use a sigmoid function, which maps input values from $-\infty$ to $\infty$ to output values 0.0 to 1.0, in a smoothed step-like function. This allows us to model binary values in a way appropriate for neural networks.

By injecting Gaussian noise into the training in the output of the binary encoding layer, the network is forced to learn how to ignore noise and become less sensitive to small changes. This makes the data leaving the binary encoding layer either highly negative or highly positive. We then pass this through a sigmoid activation function, making the data either 0.0 or 1.0. This gives us the 0/1 bits we need for semantic hashing.

In Figure 2, we see level maps using one-hot encoding. The floor is represented by a solid color, the door by a white line, the wall by a black line, and the water by an open space.
In order to represent different classes of tiles in the game maps, we use a one-hot encoding [12, p. 129], as shown in Figure 2. We start with an input level, such as the example in Figure 2a from The Legend of Zelda. Each tile is given a semantic tile value (e.g. floor, water, door, wall), with \( N \) different possible tile values on the \( W \times H \) game map (Figure 2b). This encoding is split up into \( N \) binary planes each of size \( W \times H \), where for each position \( x, y \) only one of the \( N \) planes has a 1 and all others have a 0 (Fig 2c). We then flatten these planes into a single vector of length \( N \times W \times H \), with each element in the vector either a 0 or a 1 (Figure 2d).

One interesting and useful feature of semantic hashing is that the number of bit flips between two categories gives us a sense of distance between two categories. That is, the category representing 0000 is closer to 0001, 0010, 0100, and 1000 than it is to 1111. Thus not only do we automatically categorise and cluster the level maps, but we can also use the hash value to get a sense of which maps are closer to others.

### 3 Experiment and Results

In this section, we present our experimental results of using semantic hashing on dungeon maps from The Legend of Zelda and platformer levels from Super Mario Bros. Our networks are implemented in Keras\(^1\) using the TensorFlow\(^2\) backend running with 2 \( \times \) NVIDIA 1080 GPUs.

We use the same network topology for both games, as shown in Figure 3. The input and output layers are of size \( n = N \times W \times H \), where \( N \) is the number of tile types, \( W \) is the width of the content, and \( H \) is the height of the content. These values are defined in the subsections for each game.

When training, we inject Gaussian noise of standard deviation 1.0 into the encoding layer and output layer, followed by a sigmoid activation that encourages output values of 0.0 and 1.0. While full details are beyond the scope of this paper, for completeness we use a binary-cross entropy to determine error between input and output, and RMSprop for gradient descent.

#### 3.1 The Legend of Zelda

The Legend of Zelda is the first in a series of popular RPG video games for Nintendo game systems. The game contains two quests, each of which has nine multi-room dungeons. In total, there are 459 rooms, each of size 11 \( \times \) 16, but only 264 of these are unique.

---

\(^1\)https://keras.io/
\(^2\)https://www.tensorflow.org/
3.2 Super Mario Bros

Super Mario Bros is the first in a long series of popular 2D side-scrolling platformers for Nintendo game systems. The VGLC level corpus for Super Mario Bros. contains 15 platforming levels of varying length, each 14 tiles high and between 149 and 373 wide (average of approx. 195 width). The corpus contains the above ground, below ground, and tree levels. Instead of processing the entire level at once, we split each level into sliding 4-tile wide vertical windows. This gives us 2,923 windows of size $14 \times 4$, which we reduce to 1,317 windows by eliminating duplicate rows for training the autoencoder, so that common windows are not overrepresented. Each tile is put into one of four classes: BLOCK, MONSTER, COIN, or VOID. Therefore, the input vector is of size $14 \times 4 \times 4 = 224$ for each window.

We keep the number of training iterations to 1,000 epochs, with a mini-batch size of 32, and $k = 3$ bits of encoding for eight possible categories. As with the Zelda example, this only takes a couple of minutes to train on our system.

The results of the semantic hashing are shown in Figure 5. Yellow represents coins, red represents monsters, black represents empty space, and grey represents blocks, bricks, or ground. Some categories stand out to us: Category 0 generally represents underground levels without tunnels to climb through; Category 1 contains high obstacles to climb; Category 2 represents regions where something is over the player’s head (coins or ceiling); Category 3 represents drops; Category 4 includes level segments without a ground plane; Category 5 indicates small obstacles to hop over; Category 6 consists of gaps without monsters or coins; and Category 7 includes open spaces.

4 Conclusion

This paper has shown how machine learning and semantic hashing can be used to categorise game content without human knowledge. The focus of this paper has been to explain how semantic hashing works (with a minimum of mathematics) and to demonstrate the results of classification on two popular video games. While it can categorise individual pieces of game content, there are many additional aspects of level design that we do not focus on here, such as connectivity between rooms, dynamics, or narrative. The examples here are also all in the realm of two-dimensional levels, and we do not know how much additional work will be needed to make this method work well for other types of content.

While the training and clustering proceeds without human intervention, hyperparameters such as the number of training epochs and the structure of the neural network have a large impact on the effectiveness of semantic hashing. The proof-of-concept method we used here does require us to adjust and tune the hyperparameters to find the best results for a particular game, but there exist methods for using random search [13] and genetic algorithms to discover hyperparameters for neural networks [14].

While we have not yet applied this algorithm to an actual game design process, we believe the semantic hashing technique can someday prove to be valuable for applications of automated and AI-assisted game design. These potential applications include: understanding how much of each type of content is present in a game; generating new levels in a procedural content generation system; suggesting similar or different designs in a mixed-initiative system; and recognising what an end user might be creating for user-generated content. Iterative design requires a solid analysis and understanding of what has been created in each iteration in order to improve it, and these computer-assisted techniques are a promising way to improve our understanding of design and our ability to execute good design.

Acknowledgements

Thanks to Philip Bontrager, Daniel Gopstein, Scott Lee, Rishabh Jain, and Andy Nealen for their discussions and training on neural networks, deep learning, and autoencoders. Thank you to NVIDIA for providing GPUs resources for this research project.

Aaron Isaksen received his PhD from New York University, is a long-time game designer, co-founder of Indie Fund, and Chairman of IndieBox. Address: New York City, New York, USA. Email: aisaksen@appabove.com

Christoffer Holmggaard received his PhD from the IT University of Copenhagen, did his postdoc at New York University, and is a co-owner of Die Gute Fabrik. Address: Olfert Fischers Gade 45-2, 1311 Copenhagen K, Denmark. Email: christoffer@holmgard.org
Figure 4. Semantic hashing of Zelda maps into $27 \leq 2^5$ categories.

Julian Togelius is an Associate Professor at New York University. He co-directs the Game Innovation Lab and researches AI and games.

Address: New York University, 2 MetroTech Center, 11201 Brooklyn, NY, USA.

Email: julian@togelius.com
Figure 5. Semantic hashing of Mario level windows into $8 = 2^3$ categories.

References


Gadeiro Challenges #2 and #3

Pack the pieces on the right to fill the shapes on the left. Gadeiro is described on pages 39–41.

Challenge #2

Challenge #3
Limping Boards for Games

Cameron Browne, Queensland University of Technology (QUT)

This paper describes how the simple mathematical notion of the limping triangle may be extended to a broader class of limping polygons, and the implications this has for the design of game boards. Limping rectangular boards (with a square basis) and limping hexagonal boards (with a regular hexagonal basis) are shown to give the designer control over some useful game design parameters.

1 Introduction

A limping triangle is a right triangle in which the two short sides (i.e. not the hypotenuse) differ in length by exactly one unit [1, p. 116]. Figure 1 shows a limping triangle with short sides of length n and n+1.

![Figure 1. A limping triangle.](image)

This paper describes how this concept can be extended to describe certain subsets of game boards. While the limping property does not confer any particularly interesting mathematical properties to the triangle, its use in the context of board shape can give certain mathematical guarantees that are useful for game design.

2 Limping Polygons

Firstly, we extend the concept of the limping triangle to limping polygons, for which a similar n:n+1 property holds. Figure 2 (left) shows the 3:4:5 triangle which is both a Pythagorean triangle and a limping triangle.

![Figure 2. 3:4:5 triangle and 3:4 limping rectangle.](image)

Placing two copies of this triangle together forms the rectangle with sides {3, 4, 3, 4} (right). We denote this as a 3:4 limping rectangle, as its sides alternate between 3 and 4 units in length, and shall adopt the notation n:n+1 to describe such polygons.

2.1 Concavity and Convexity

The examples shown so far – triangle and rectangle – are convex polygons, as each turn is in the same direction. However, it is also possible to have concave limping polygons.

Figure 3 (left) shows a concave limping polygon with sides of length {1, 2, 1, 2, 1, 2, 1, 2}, in which turns alternate left and right with each segment. The polygon shown in Figure 3 (right) with sides of length {1, 2, 3, 2, 1, 2, 3, 2} is not a limping polygon in the true sense, as each pair of adjacent sides satisfies the n:n+1 property but the overall figure includes sides of length n+2. The segment lengths of true limping polygons must alternate only between n and n+1.

![Figure 3. A concave limping polygon (left) and a locally but not globally limping polygon (right).](image)

2.2 Right and Non-Right Polygons

If every turn in a limping polygon is a right angle, then the figure will cover an area that is an even number of square units. This can be seen by observing that side lengths along one axis must all be even while side lengths along the other axis

must all be odd. Any cross-section along the even axis must pass through an even number of units, and any sum of even numbers will be even, hence the total area of any right-angled limping polygon must be an even number of square units. This can be seen in Figure 4.

Figure 4. Right limping figures have even area.

Figure 5 shows the first four shapes in the series of minimal (1:2) convex limping polygons, which all have an even number of sides, but in which only the first (rectangular) case has a right-angled basis. Limping polygons must have an even number of sides in order to satisfy the \( n:n+1 \) constraint. Hence limping triangles – which inspired the idea – are ironically excluded from the broader umbrella of limping polygons.

Figure 5. Minimal (1:2) convex limping polygons.

3 Limping Boards

In terms of board design, we are most interested in convex polygons, as they are the most efficient and widely used board shape. More specifically, we are most interested in the first two convex cases shown in Figure 5 – rectangular and irregular hexagonal – as these are the two fundamental shapes that align naturally with the cells of regular square and hexagonal tilings (Figure 6).

Figure 6. Square and hexagonal bases.

3.1 Square Basis

Limping boards with a square basis are simply \( n \times (n+1) \) rectangles. For example, Figure 7 shows a 7:8 limping rectangular board.

Figure 7. A 7:8 limping rectangular board.

For limping rectangular boards, the number of edge cells \( C_{er} \) will always be even and is given by:

\[
C_{er} = 2(2n - 1)
\]

The total number of cells \( C_{tr} \) will also always be even and is given by:

\[
C_{tr} = n(n + 1)
\]

Limping rectangular boards are reasonably common, and the BoardGameGeek forum ‘Games played on an N x N+1 grid’ lists dozens of examples.\(^1\) Note that the terminology \( n \times (n+1) \) implies a rectangular shape, whereas the new \( n:n+1 \) terminology suggested for limping polygons applies to a wider range of shapes.

3.2 Hexagonal Basis

Limping hexagonal boards, with alternating sides of \( n \) and \( n+1 \) hexagonal cells, are more interesting for a number of reasons. Figure 8 shows the three smallest limping hexagonal boards, of size 1:2, 2:3 and 3:4. The figures are coloured to show how each can be decomposed into three rhombi of size \( n^2 \).

Figure 8. Limping hexagonal boards: 1:2, 2:3, 3:4.

---

\(^1\)https://boardgamegeek.com/geeklist/23237/games-played-n-n1-grid/
For limping hexagonal boards, the number of edge cells $C_{eh}$ will always be odd and is given by:

$$C_{eh} = 3(2n - 1)$$  \hspace{1cm} (3)

The total number of cells $C_{th}$ will have the same parity (i.e. even or odd) as its base size $n$, and is given by:

$$C_{th} = 3n^2$$  \hspace{1cm} (4)

This guaranteed divisibility by three, in both the number of edge cells and total number of cells, makes limping hexagonal boards a promising option for three-player games. It may also have benefits for publishers, if boards can be disassembled into three equal pieces for storage in a game box.

Further, limping hexagonal boards do not have a single central cell, as is the case with the more standard hexhex board (i.e. a regular hexagon tessellated by hexagons), but are centred on the intersection of three equally central cells. This is already an interesting design feature which can reduce any inherent first move advantage.

Figure 9 shows the limping rhombus board, which is an unusual hybrid. This board has a hexagonal basis but its shape has more in common with an $n \times (n+1)$ rectangle, giving an even number of edge cells and an even number of total cells. I know of only one game – $n \times (n+1)$ Hex, discussed below – which uses this board shape.

<table>
<thead>
<tr>
<th></th>
<th>Regular</th>
<th>Hex.</th>
<th>Limping</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_e$</td>
<td>Even</td>
<td>Even</td>
<td>Odd</td>
</tr>
<tr>
<td>$P_t$</td>
<td>As $n$</td>
<td>Odd</td>
<td>As $n$</td>
</tr>
</tbody>
</table>

Table 1. Cell parity of various board types ($n > 1$).

It can be seen that the limping hexagonal board is the only combination that guarantees an even number of total cells (for even sizes of $n$) but an odd number of edge cells. This, combined with the guaranteed divisibility by 3 and lack of a single central cell, make the limping hexagonal board a perhaps underused resource.

### 4 Examples

The following examples show instances of games that use limping boards as a basic design feature.

#### 4.1 Clobber

Clobber is a two-player combinatorial game, in which players take turns moving one of their pieces to capture (i.e. clobber) an adjacent enemy piece. The last player to move wins. Figure 10 shows the starting position for a 5:6 game.

Figure 9. A 5:6 limping rhombus board.

Figure 10. A 5:6 game of Clobber about to start.

Clobber is often played on a 5:6 or other $n:n+1$ limping rectangular board. This guarantees an even number of board cells regardless of $n$, hence theoretically offers each player an equal number of moves; an important consideration in combinatorial games.

---

2This is one reason why the 14×14 Hex board is often preferred over Piet Hein’s original 11×11 board.

3https://boardgamegeek.com/boardgame/23864/clobber
4.2 Silverman’s Minichess

Figure 11 (left) shows the $4 \times 4$ game of Minichess invented by mathematician David Silverman.\footnote{https://en.wikipedia.org/wiki/Minichess}

![Minichess board](image)

When a trivial win for White was pointed out, Silverman added an opening rule – Black nominates which pawn White must play first – but this was found to lead to a trivial win for Black. Finally, Silverman solved the trivial win problem by extending the board to a $4:5$ limping rectangle with a row of empty cells separating the players’ forces, as shown in Figure 11 (right). This demonstrates a common application of the limping rectangle, as a way of sizing boards between the more standard square sizes. The following considerations often result in limping rectangular boards, although the fact that they are limping is somewhat coincidental in these cases:

- Does the game need a different number of pieces? Add or remove a row/column of pieces.
- Do the opposing forces need more separation? Add a row/column of empty cells.
- Do the opposing forces need to engage more quickly? Remove a row/column of empty cells.

4.3 Bridg-It

The game of Bridg-It, by mathematician David Gale, is shown in Figure 12. Players take turns placing a bridge of their colour between available pegs of their colour, in an effort to connect their sides of the board.

The board consists of two orthogonally overlapping $5:6$ limping rectangular grids of pegs. The overall board is ostensibly a sparse $11 \times 11$ grid of pegs, but each player’s moves are constrained to their own $5:6$ limping rectangle. Limping rectangles provide the most square non-square rectangular shape, allowing this neat form of overlap.

![Bridg-It board](image)

The layout of pegs on the Bridg-It board can also be seen in the starting position of connection game Ayu,\footnote{https://boardgamegeek.com/boardgame/114484/ayu}[Figure 13] as a way to initially separate all pieces, and in the squares of the semiregular truncated square tiling 4.8.8 (Figure 14), although these occurrences are coincidental.

![Ayu board](image)

![Truncated square tiling](image)
4.4 Star

Turning now to games on limping hexagonal boards, the game of Star demonstrates their design benefits clearly. Star was designed in 1983 by mathematician Craige Schensted (now known as Ea Ea) [4]. Figure 15 shows a completed 4:5 game won by Black.

In Star, two players add a piece of their colour each turn, and win by achieving the highest score at the end of the game. A star is a connected chain of same-coloured pieces that contains edge pieces adjacent to at least three external dots (see Figure 15). Each star’s score is the number of external dots that its edge pieces are adjacent to minus 2. Each player’s score is the total score of their stars.

While the scoring system may initially sound confusing, it is extremely intuitive in play, and the game basically boils down to connecting as many edge cells of your colour into as few groups as possible, to maximise the external neighbour count while minimising the -2 group penalty.

Black has won the game shown with star scores of $11 + 2 = 13$ points, compared to White’s $3 + 1 + 4 + 4 = 12$ points. Black wins despite only having nine pieces on edge (scoring) cells while White has twelve pieces on edge cells; Black’s stronger connection reduces the number of black groups and makes the difference.

Star boards with even $n$ (e.g. size 4:5) will have an even number of cells, guaranteeing an equal number of moves for both players if the board fills up. However, the more fundamental reason that Projex is played on a limping hexagonal board is due to the geometry of projective edge connections, as edges of length $n$ nestle into neighbouring edges of length $n+1$ on the hexagonal grid, and vice versa. To illustrate this point, Figure 17 shows three adjacent limping hexagonal boards packed around a common intersection. Note that the boards are centred relative to each other.

4.5 Projex

Figure 16 shows the game of Projex, invented by mathematicians Lloyd Shapley and Bill Taylor in 1994. The board ‘wraps around’ at the edges as shown to form a projective plane, e.g. edge cell $a$ is considered adjacent to edge cells $l$ and $k$, etc.

Players win at Projex by making a chain of their pieces that forms a non-trivial loop encircling the projective plane. Explaining exactly what this means is beyond the scope of this paper, suffice it to say that players win by forming a chain of their pieces with an odd number of edge crossings, such as the white chain shown in Figure 16 which has a single edge crossing.

Again, even board sizes $n$ give an even number of board cells, guaranteeing an equal number of moves for both players if the board fills up. However, the more fundamental reason that Projex is played on a limping hexagonal board is due to the geometry of projective edge connections, as edges of length $n$ nestle into neighbouring edges of length $n+1$ on the hexagonal grid, and vice versa. To illustrate this point, Figure 17 shows three adjacent limping hexagonal boards packed around a common intersection. Note that the boards are centred relative to each other.

Figure 18 shows three hexhex boards packed together in a similar way. Notice in this case, however, that this packing is oriented and can progress either clockwise (left) or anticlockwise (right) around the intersection, which would

6https://boardgamegeek.com/boardgame/31805/projex
make wraparound projective adjacency ambiguous. The centred packing allowed by limping hexagonal boards avoids this ambiguity.

These properties combined harmoniously to benefit the game. The limping board design allowed an intuitive starting arrangement for each player’s twelve pieces as shown, and solved other problems that had been plaguing the game.

4.6 Shoulder to Shoulder

Figure 19 shows the starting position for Shoulder to Shoulder, a game for three players designed by David Parlett in 1975. The game was originally designed on a size 7 hexhex board with 127 cells, but it was found that the central cell gave the player who occupied it a huge advantage. A play-tester then suggested reducing the board to that shown – a 6:7 limping hexagonal board – which provided the following benefits:

- No more single central cell.
- Number of edge cells $C_{eh}$ divisible by 3.
- Total number of cells $C_{th}$ divisible by 3.
- Number of cells reduced from 127 to 108.

Horn explains that the Tres design was originally a size-5 hexhex layout, but was reduced to a 4:5 limping hexagon in order to make games

4.7 Tres

The game of Tres, designed by Fred Horn in 2011 and shown in Figure 20, involves a grid of pegs in a 4:5 limping hexagon. Players slot a three-holed triangle of their colour over three pegs each turn, and aim to surround the greatest number of pins with pieces of their colour. Pieces may stack up to three triangles high.

7 http://www.parlettgames.uk/gamepie/shoulder.html
8 Personal correspondence from Horn including unpublished Tres specification and rule sheet.
a bit shorter and reduce the amount of material required. An added benefit of the limping shape is that it reduces the board from 6-fold to 3-fold rotational symmetry, which Horn points out provides greater scope for strategy and allows more strategic play earlier in games.

4.8 Zertz

Figure 21 shows the starting layout for Zertz, a GIPF game designed by Kris Burm in 1999 [5]. The 37 black discs show the layout for the standard game, while dotted positions are for additional pieces that can be added for longer games.

If all 48 discs are used in the extended game, then the starting layout forms a 4:5 limping hexagon. This is another case of extending a standard board in one direction to coincidentally produce a limping board, as per Minichess.

4.9 \( n \times (n+1) \) Hex

The game of Hex is typically played on an \( n \times n \) rhombus of hexagons [2]. However, Martin Gardner describes an interesting variant played on an \( n \times (n+1) \) rhombus of hexagons [6], shown in Figure 22 with the first player aiming to connect the two sides farthest apart and the second player aiming to connect the two sides closest together. This is the only game played on an \( n \times (n+1) \) limping rhombus that I am aware of.

The interesting thing about this version of Hex is that it provides a guaranteed win for the second player, due to a point-pairing symmetry strategy. Each time the opponent plays, then playing in the matching cell with the same letter ensures that the closest board sides are connected before the farthest sides [6].

The partition of the \( n \times (n+1) \) limping rhombus into two mirrored triangles, shown in Figure 22, recalls the doubling of limping triangles discussed in Section 2 that started this discussion.

This raises the intriguing possibility of similar point-tripling strategies for some three-player games on hexagonal limping boards, e.g. based on the partitions shown in Figure 8. But such strategies would have to take into account intervening moves by both opponents so are unlikely.

5 Key Properties for Design

Some key properties in terms of game design, for the three types of limping board discussed above, are as follows:

1. **Limping Rectangular Boards**:
   - Even number of edge cells.
   - Even number of total cells.
   - 2-fold symmetry (rather than 4-fold).
   - Extend a row to separate forces.
   - Remove a row to encourage conflict.

2. **Limping Hexagonal Boards**:
   - Odd number of edge cells.
   - Odd or even number of total cells (as per \( n \)).
   - Number of edge cells divisible by 3.
   - Number of total cells divisible by 3.
   - No single central cell.
   - Projective edge crossings are centred.
   - 3-fold symmetry (rather than 6-fold).

3. **Limping Rhombus Boards**:
   - Even number of edge cells.
   - Even number of total cells.
   - Subject to symmetry strategies.

In all cases, the limping property guarantees that certain board sides will be longer or shorter than others, and closer or further from the opposite board sides than others. This asymmetry could be exploited to handicap games as needed. For example, in a connection game, the weaker player – or second player if there is a first move advantage – could aim to connect the longer...
board edges which are closer together, while their opponent could aim to connect the shorter board edges which are further apart. The combination of more target cells, closer together, gives the weaker player a double advantage for balance.

6 Conclusion

The extension of the mathematical idea behind limping triangles to limping polygons defines a class of shapes with some useful properties relevant to the design of game boards. Limping rectangular boards (with a square basis) provide useful board sizes in between standard square sizes, and guarantee an even number of board cells regardless of size. Limping hexagonal boards (with a regular hexagonal basis) guarantee an even number of cells when \( n \) is even, and an odd number of edge cells regardless of size. They also pack with projected neighbours to provide unambiguous wraparound adjacencies on the hexagonal grid. Other useful properties are listed above.

When developing games, it is worth keeping in mind the properties of the various board types summarised in Table 1. These constraints provide mathematical guarantees that can be exploited to elegantly solve certain game design problems. It is no coincidence that most of the designers mentioned above, who have used limping boards in their games, are mathematicians.

Acknowledgements

This work was supported by a QUT Vice-Chancellor’s Research Fellowship. Thanks to Russ Williams, Néstor Romeral Andrés, John Farrell, Sandra Snan and Fred Horn for examples of limping boards and their benefits.

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au

Gadeiro Challenge #4

Pack the pieces on the right to fill the shape on the left. Gadeiro is described on pages 39–41.
A Rubik’s Cube is a $3 \times 3 \times 3$ twisty puzzle. This puzzle can be conceptualised as an array of 27 cubies. The 27 cubies consist of four types of pieces: eight corners, twelve edges, six faces and one core. But is this really the complete picture? Have we really accounted for all possible piece types? $3 \times 3 \times 3 = 27$; thus, one is tempted to say ‘yes’, but this paper will argue that we are only seeing part of the picture. It will claim that the complete picture contains 64 cubies across ten piece types. These new cubies are defined, and a method for making them visible is detailed. This paper also explains why the resulting puzzle has been named the Complex $3 \times 3 \times 3$.

1 Introduction

The $3 \times 3 \times 3$ twisty puzzle was invented by Ernő Rubik in 1974 and marketed by Ideal Toys as the Rubik’s Cube in 1980. This kicked off the exploration of twisty puzzles in general, and many different variations resulted. The $4 \times 4 \times 4$ and $5 \times 5 \times 5$ soon followed, as did the application of the same principles to many other geometries.

Previously we have seen how the $3 \times 3 \times 3$ can be adapted to higher dimensions and shown how pieces normally hidden within the interior volume of the puzzle can be made visible. Here the intent is to go back and take a closer look at the $3 \times 3 \times 3$ itself.

The $3 \times 3 \times 3$ is defined by six cut planes. If the $3 \times 3 \times 3$ is centred on the origin, these planes are all an equal distance from the origin. It is valid to think of these planes as the $x = 1$, $x = -1$, $y = 1$, $y = -1$, $z = 1$ and $z = -1$ planes. These planes cut all of 3-space up into 27 separate volumes. So it is apparent that if there is more to a $3 \times 3 \times 3$, those pieces are not hiding inside or even outside the current surface of the puzzle. These 27 cubies constitute the set of real pieces. But this is just the beginning of this investigation. This paper outlines several new piece types that represent new classes of cubies, their operation, and how they can be visualised.

2 Nortmann’s Twistability

In 2009, Andreas Nortmann started the thread ‘Analysis of Twistability and Virtual Pieces’ on TwistyPuzzles.com, in which he introduced his scheme for analysing twisty puzzles. To see how his approach can find additional pieces, we will apply it to the Skewb, shown in Figure 1.

The Skewb, first produced by Mefferts in 1982, is defined by four cut planes. Each plane perpendicularly bisects one of the four body diagonals of a cube, so all four planes cross at the origin. These four planes cut 3-space up into 14 volumes which map to the eight corners and six face pieces seen on the Skewb. These 14 pieces are all the real pieces possible in a Skewb. A corner rotation of a Skewb moves half of these 14 pieces (four corners and three faces). A single corner piece cannot be rotated by itself.

Andreas’s methodology looks at the number of linearly independent layers per axis. It then defines a holding point for each possible set of linearly independent layers for a given puzzle. This is the point that is invariant, or stationary, under all rotations of the linearly independent layers. In some cases, it can be considered the intersection of all the dependent layers.

For the Skewb, let us name the six faces of the puzzle: $R=$right, $L=$left, $U=$up, $D=$down, $F=$front, and $B=$back. The RUF corner would then be the
corner at the intersection of the right, up, and front faces. On each axis, which allows rotations, the Skewb only has a single cut plane so there is only a single independent layer on each axis. In other words, if we define the RUF layer as the independent layer then the LDB layer is dependent. The latter differs from a RUF rotation only by a global rotation of the entire puzzle.

Each axis presents a binary choice as to which layer is called the independent one. There are four axes, giving \(2^4 = 16\) possible sets of linearly independent layers for a Skewb. This produces 16 possible holding points, yet there are only 14 pieces in a Skewb. These two additional holding points equate to two new pieces which exist outside the set of the real pieces.

Each of these 16 possible sets of linearly independent layers is called a **signature**, and to identify a piece with its signature you just look for the piece that remains stationary under rotation of all four independent layers. For example, if we define the linearly independent layers as LDB, LDF, LUB, and LUF we will note that none of these rotations move the R face piece. To easily see this, hold a Shewb and make a rotation about each of the four corners on the left face. The R face piece’s position and orientation are unaffected by all four turns. This is illustrated in the renderings shown in Figure 2. In this case, the R face piece is our holding point.

### Table 1. Nortmann’s twistability for the Skewb.

<table>
<thead>
<tr>
<th>Piece</th>
<th>Type</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Face</td>
<td>Real</td>
</tr>
<tr>
<td>RUF</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>RUB</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>U</td>
<td>Face</td>
<td>Real</td>
</tr>
<tr>
<td>RDF</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>F</td>
<td>Face</td>
<td>Virtual</td>
</tr>
<tr>
<td>LUF</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>RDB</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>B</td>
<td>Face</td>
<td>Real</td>
</tr>
<tr>
<td>LUB</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>D</td>
<td>Face</td>
<td>Real</td>
</tr>
<tr>
<td>LDF</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>LDB</td>
<td>Corner</td>
<td>Real</td>
</tr>
<tr>
<td>L</td>
<td>Face</td>
<td>Real</td>
</tr>
</tbody>
</table>

### 3 Virtual Pieces

What do these virtual pieces mean physically? Can they be given volume and appear on an actual puzzle? To answer these questions we take a closer look at the signatures of the two virtual pieces discovered in a Skewb. [LDB,RUB,RDF,LUF] is the signature of the first virtual piece but these also correspond to the four corners of a tetrahedron inscribed inside the cube. When we look for a piece that is unaffected by all four of these rotations we find none, but we note that there are four corner pieces that just rotate in position. These pieces never leave their solved position, it is only their orientation that changes. So we can imagine each of these four corners being connected to an axis which allows rotation and those four axes connected at the centre of the puzzle.

Figure 3. The core of a Skewb is a virtual piece.

Table 1 lists the combinations obtained with Nortmann’s twistability approach, where 0 = ‘dependent layer’ and 1 = ‘linearly independent layer’. New pieces discovered with Nortmann’s method of analysis are named **virtual** pieces.

---

3 All renderings are made using SOLIDWORKS computer-aided design software: http://www.solidworks.com
construction of most Skewbs as seen in Figure 3. This core of the Skewb is not created when purely planar cuts are used so it is not one of the 14 real pieces. This core is one of the two virtual pieces. It remains stationary while the four corners it is attached to are allowed to rotate.

The second virtual piece has this signature [RUF,LDF,LUB,RDB] and these too are the corners of a tetrahedron inscribed inside the cube. So this too could serve as a core if it were attached to the other four corners using the same logic. These parts are not found together inside any physical Skewb as they would interfere with each other. However, they can be made to coexist. A puzzle which contains all the real and all the virtual pieces is called augmented (real + virtual = augmented). A possible construction for the Augmented Skewb was presented on TwistyPuzzles.com back in 2011.

Figure 4 shows how this would appear in cubic form. It makes use of circles to expose the interior core on the puzzle’s surface and makes use of bridges to construct the second core on the exterior of the puzzle where it cannot interfere with the first one. Note that if one of these cores is viewed as a holding point, the other core moves with every turn.

To fit all 16 pieces into the Augmented Skewb mechanically, the two cores could not be identical. The two cores are seen in Figure 5(a) and (b). Note the second core is composed of two disconnected parts. Each core behaves as if it is affixed to four corners.

Since the two cores are different in design, this necessitated two types of corners. Figure 5(c) shows an example of a single corner which acts like one of the four corners attached to the core in Figure 5(a). It is constructed of three discontinuous parts. Figure 5(d) shows an example of a single corner which acts like one of the four corners attached to the core in Figure 5(b). This corner is constructed of just a single part and is physically attached to the core in Figure 5(b) with a screw. Figure 5(e) shows a single copy of a face piece. All six faces are mechanically identical.

3.1 Virtual Pieces and the 3×3×3

Nortmann’s methodology can be applied to the 3×3×3. The 3×3×3 has 3 axes of rotation, the x-axis, y-axis, and z-axis. On each axis there are 3 layers which allow rotation. On the x-axis exists the right layer (R), the left layer (L) and a slice layer (S_x). On the y-axis exists the front layer (F), the back layer (B) and a slice layer (S_y). On the z-axis exists the up layer (U), the down layer (D), and a slice layer (S_z).

These 3 layers on each axis are not independent. For example, a rotation of the R layer is the same as a rotation of the L and S_x layers in the opposite direction combined with a global rotation of the entire puzzle. So for each of these three axes we can fix one dependent layer and still have a completely valid representation of the 3×3×3. There are three axes, each with a choice between three layers as to which to pick as the fixed or dependent layer, giving 3^3 = 27 possible choices. Table 2 shows how each choice matches up to one of the 27 already known pieces, where 0 = ‘dependent layer’ and 1 = ‘linearly independent layer’. So this means that the 3×3×3 does not contain any virtual pieces. But the investigation does not stop there.

---

To see how a part of the picture is still missing, consider the $3 \times 3 \times 3$ with curved cuts shown in Figure 6. Note the appearance of a new piece type, highlighted in red.

Figure 6. A $3 \times 3 \times 3$ created with curved cuts.

Table 2. Nortmann’s twistability for the $3 \times 3 \times 3$.

Table 3. Galla’s twistability for the $3 \times 3 \times 3$.
4 Galla’s Twistability

Fortunately, in the same TwistyPuzzles.com thread started by Nortmann [7], Matt Galla introduced his methodology for performing a similar analysis. Whereas Nortmann considered all possible ways for a piece to remain stationary, Galla took the approach of looking for all possible ways a piece could move. For his analysis, one only needs to be concerned about the independent layers, so we will pick the core as our holding point. This keeps the slice layers from turning and we only need to consider the R, L, F, B, U, and D turns. For each, we simply need to ask if the part is in or outside of this layer, a binary choice. There are six layers so there are $2^6 = 64$ possible pieces. These are listed in Table 3, where 0 = ‘part is outside this layer’ and 1 = ‘part is in this layer’.

Table 4 names the new piece types and lists some key properties. The four types that we are all familiar with are the core, which does not turn with any layers as it is typically defined as the holding point; a face, which just rotates in place; an edge, which rotates with two adjacent layers; and a corner, which rotates with three all mutually adjacent layers. The new piece types cover the additional possibilities.

A **bifloor** rotates with two opposite layers. A **triwall** rotates with three layers, two of which are opposite each other. An **inverse edge**, the new piece type shown in Figure 6, turns with all layers except two adjacent layers, which is the opposite signature of a normal edge. An **inverse bifloor** turns with all layers except two opposite faces. The **inverse face** turns with all layers except one. And, finally, the **inverse core** turns with all layers.

<table>
<thead>
<tr>
<th>Type</th>
<th>Signature</th>
<th>Qty</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>No layers</td>
<td>1</td>
<td>Real</td>
</tr>
<tr>
<td>Face</td>
<td>1 layer</td>
<td>6</td>
<td>Real</td>
</tr>
<tr>
<td>Edge</td>
<td>2 adjacent layers</td>
<td>12</td>
<td>Real</td>
</tr>
<tr>
<td>Bifloor</td>
<td>2 opposite layers</td>
<td>3</td>
<td>Imag.</td>
</tr>
<tr>
<td>Corner</td>
<td>3 mutually adjac. layers</td>
<td>8</td>
<td>Real</td>
</tr>
<tr>
<td>Triwall</td>
<td>3 layers with 2 opposite</td>
<td>12</td>
<td>Imag.</td>
</tr>
<tr>
<td>Inverse Edge</td>
<td>All but 2 adjacent layers</td>
<td>127</td>
<td>Imag.</td>
</tr>
<tr>
<td>Inverse Bifloor</td>
<td>All but 2 opposite layers</td>
<td>3</td>
<td>Imag.</td>
</tr>
<tr>
<td>Inverse Face</td>
<td>5 Layers (all but 1 layer)</td>
<td>6</td>
<td>Imag.</td>
</tr>
<tr>
<td>Inverse Core</td>
<td>All 6 Layers</td>
<td>1</td>
<td>Imag.</td>
</tr>
</tbody>
</table>

Table 4. The ten Complex $3 \times 3 \times 3$ piece types.

Like the new piece types identified via Nortmann’s methodology, the new piece types identified with Galla’s method were also given a name. They are called imaginary. The full set identified with Galla’s methodology is the superset of all the real, virtual, and imaginary pieces. Just as the set of the virtual pieces excludes the real pieces, the set of imaginary pieces does not include the real or virtual pieces.

So where do all these new pieces fit into the mechanism? How does one interpret them physically? Figure 6 showed one way to introduce the inverse edge, but, upon closer inspection, there is no room in the puzzle to add a mechanism for this piece, aside from maybe the use of magnets. But there is an even easier way to introduce several of these parts, and it also highlights the nature of the relationship between a given type and the inverse of that type.

Consider the pair of $3 \times 3 \times 3$ twisty puzzles shown in Figure 7. The puzzle on the left shows a face turn one layer deep, while the puzzle on the right shows a face turn two layers deep. The red cubies constitute a single face layer in both cases.

Both are typical $3 \times 3 \times 3$s and both share identical mechanisms. For the $3 \times 3 \times 3$ on the left we define the core as a holding point and the R, L, F, B, U, and D layers are defined as they usually are as the rotation of the nine cubies on that face. This left $3 \times 3 \times 3$ contains the four types we are familiar with: a core, faces, edges, and corners. For the $3 \times 3 \times 3$ on the right we define things a bit differently. Now a face turn is considered to be a rotation of the 18 cubies that are closest to that face and we do this for all six faces.
Physically we have the same puzzle but when looked at from this perspective we now have a puzzle that contains an inverse core, inverse faces, inverse edges, and the same corners we had before. In fact, the corners are their own inverse. From a solving and construction perspective, these two puzzles are identical.

The virtual pieces may not appear that interesting or to add anything new. But that is only true in this case as we were simply trading out one part for its inverse. Things get much more interesting if the inverse can be added while retaining the original part, as hinted at in Figure 6.

5 Imaginary $3 \times 3 \times 3$ Parts

Presented next is a method that can be used to add all the imaginary parts to a $3 \times 3 \times 3$ while retaining all the real parts. This method was first presented by Carl Hoff in December 2009 on the TwistyPuzzle forums. This puzzle is called the Complex $3 \times 3 \times 3$. In general, a puzzle which contains all the real and all the imaginary pieces is called complex ($\text{real} + \text{imaginary} = \text{complex}$).

To create the picture of the Complex $3 \times 3 \times 3$, we start with an array of $5 \times 5 \times 5$ cubies, as shown in Figure 8. The Complex $3 \times 3 \times 3$ is created by applying a new definition of a face turn to the $5 \times 5 \times 5$ array of cubies.

And just as we did in the example above we play with the definition of a face-turn. There are actually a couple of definitions we could use, but the nicest is to define a face turn as the rotation of all the cubies in the layer closest to that face plus the rotation of the cubies in the central slice layer on that axis as well. If we use this same definition for all faces we see we have now created all ten piece types, 64 pieces in all, in one puzzle. All ten piece types are highlighted in Figure 9.

If this $5 \times 5 \times 5$ array of cubies was a Multi $5 \times 5 \times 5$ it would also have ten piece types but they would be spread out across 125 pieces. To see how these two puzzles are related we will map the ten piece types of the Multi $5 \times 5 \times 5$ to the Complex $3 \times 3 \times 3$.

Using our definition of a face-layer turn for the Complex $3 \times 3 \times 3$ and applying it to a Multi $5 \times 5 \times 5$ we see the following:

1. The eight $3 \times 3 \times 3$ corners never move so these eight cubies combine to form a single part, the core, in the Complex $3 \times 3 \times 3$.

2. The four X-centres on a given face are only moved when that face rotates. These four parts combine to become a new face centre for the Complex $3 \times 3 \times 3$.

3. The two wings on a given edge only move with those two adjacent faces. These two parts combine to become a new edge for the Complex $3 \times 3 \times 3$.

4. The $5 \times 5 \times 5$ corners are moved by three mutually adjacent faces so they become the corners of the Complex $3 \times 3 \times 3$.

5. The four $3 \times 3 \times 3$ edges in a given slice layer only turn with the two opposite faces on that same axis so these four parts merge to become a bifloor in the Complex $3 \times 3 \times 3$.

6. Two opposite T-centres on a given face move with that face and two of the neighbouring faces, so these parts combine to become a triwall in the Complex $3 \times 3 \times 3$.

7. The middle $5 \times 5 \times 5$ edges are moved by all faces except for two adjacent faces so these are the inverse edges of the Complex $3 \times 3 \times 3$.

8. Two opposite $3 \times 3 \times 3$ face centres are moved by all the faces except the two opposite faces on that same axis. These two parts combine to form a single inverted bifloor of the Complex $3 \times 3 \times 3$.

9. The $5 \times 5 \times 5$ face centres are moved by all but the opposite face turn so these become the inverted faces of the Complex $3 \times 3 \times 3$.

10. The core of the Multi $5 \times 5 \times 5$ is moved with all of the defined layers so it is the inverse core of the Complex $3 \times 3 \times 3$.

---

A new definition of a face turn can be used to map the pieces of a Multi $5\times5\times5$ to the Complex $3\times3\times3$, as summarised in Table 5.

It is still an open question whether a physical mechanism exists that would allow the construction of a purely mechanical Complex $3\times3\times3$. Initial steps have been taken with the design of a physical Multi $5\times5\times5$ that allows all piece types to be stickered and solved. The resulting puzzle, called the Double Circle Real $5\times5\times5$ [6] and shown in Figure [10], is a necessary step in the process of designing a physical Complex $3\times3\times3$. 

Figure 9. Piece types found in the Complex $3\times3\times3$: (a) core, (b) face, (c) edge, (d) corner, (e) bifloor, (f) triwall, (g) inverse edge, (h) inverse bifloor, (i) inverse face, and (j) inverse core.
It is hoped that some combination of gears and latches could be worked into the design of the Double Circle Real $5 \times 5 \times 5$ to turn it into a Complex $3 \times 3 \times 3$, but that remains elusive. Electronics may need to be incorporated into the design to enable this functionality. Another option would be a computer program that simulated the puzzle and allowed it to be manipulated virtually.

Landon Kryger has presented one such application, simply called COMPLEX. It allows one to play with a completely different visual representation of this Complex $3 \times 3 \times 3$. His application simply presents the orientation of each of the 64 cubies and graphically shows how those orientations change with the face turns. Figure 11 shows the puzzle in its solved state, as rendered by Kryger’s COMPLEX program. Figure 12 shows the puzzle after a turn of the U (White) face.

### 6 Are Imaginary Pieces Redundant?

As seen in Figure 7, some real parts could be swapped out for the addition of some imaginary parts and nothing changed. So do these new pieces actually add value to the puzzle? One way to check whether the imaginary pieces actually add any new information is to calculate the number of permutations for the Complex $3 \times 3 \times 3$.

The first person to do a complete enumeration of the permutations of the Complex $3 \times 3 \times 3$ was Andreas Nortmann. Nortmann’s results, combined with some further analysis by Brandon Enright, are presented in Table 6.

These results clearly show that the imaginary pieces do add to the number of permutations in a rather significant way. But just as the core adds no new information to the Multi $3 \times 3 \times 3$, the faces give away their orientation, there are some redundant piece types in the Complex $3 \times 3 \times 3$. In Table 5, Enright’s analysis adds each of the new piece types to a Super $3 \times 3 \times 3$ individually.

---


When the bifloors are added to the Super 3×3×3 the permutation count is unchanged. As long as the face centres contain orientation information, they make the core and the bifloors completely redundant. The triwalls, when added to a Super 3×3×3, increase the permutation count by a factor of 490,497,638,400. Adding the inverse bifloors increases the permutation count by a factor of 384. Inverse edges increase the count by a factor of 490,497,638,400, the same as the triwalls. Including inverse faces increases the permutation count by a factor of 12,288, and finally, adding the inverse core increases it by a factor of 12.

However, if these numbers are multiplied together, then the actual permutation count of the Complex 3×3×3 will be overstated by a factor of 384 × 12 = 4,608. The reason for this is the inverse faces share the same relationship with the inverse bifloors and the inverse core as that present between the faces, bifloors, and core. In other words, the orientation information from the inverse faces makes the inverse bifloors and inverse core completely redundant.

7 Is It Solvable?

The question is not just whether the Complex 3×3×3 has a solution, but whether it is solvable by an individual in a reasonable amount of time. Fortunately, the answer is: yes.

Brandon Enright was the first to solve the Complex 3×3×3 in January of 2012, calling it ‘the hardest puzzle I have ever solved’. He used Kryger’s COMPLEX program together with several puzzles from Gelatinbrain’s Virtual Magic Polyhedra to find the solution. Nan Ma became the second person to solve the Complex 3×3×3, in December 2012.

8 Symmetry

The cube has 48 symmetry elements, which can be categorised as follows:

- 1 – Identity (I).
- 3 – Rotation by 180° about an axis connecting two opposite face centres (F_{180\ R_e}).
- 6 – Rotation by 90° about an axis connecting two opposite face centres (F_{90\ R_e}).
- 8 – Rotation by 120° about an axis connecting two opposite corners (C_{R_e}).
- 6 – Rotation by 180° about an axis connecting two opposite edges (E_{R_e}).
- 1 – Point reflection or inversion about the cube centre (P).
- 3 – Reflection in a plane midway between two opposite faces (F_{180\ R_e}).
- 6 – Rotation of a mirrored image (also called rotoreflection) by 90° about an axis connecting two opposite face centres (F_{90\ R_e}).
- 8 – Rotoreflection by 60° about an axis connecting two opposite corners (C_{R_e}).
- 6 – Reflection in a plane coincident with two opposite edges (E_{R_e}).

The symmetry group of the cube contains 98 different subgroups which fall into 33 different conjugacy classes. Schoenflies notation reduces this number further to only 25 subgroups in the context of point groups. This is because without the reference frame of a cube some of these 33 conjugacy classes are isomorphic.

When discussing the symmetry groups in the context of cubic puzzles, it pays to keep the 33 conjugacy classes distinct and without unique names for them both Jaap Scherphuis and Andreas Nortmann independently developed their own naming schemes for these 33 conjugacy classes.

All of the piece types that appear in any Complex N×N×N can be sorted by symmetry. This is done by examining the symmetry of a single piece of each type. Of these 33 conjugacy groups, only ten ever appear in the examination of the Complex N×N×N puzzles. These are listed in Table 7 using the three relevant naming schemes.

Several of these do not yet appear in the Complex 3×3×3 but are listed here for completeness. It should be noted that each piece belongs to the same family as its inverse. For example, a face and an inverse face will always belong to the same conjugacy class. This is easiest to see by changing the point of reference from the core to the inverse core. This transformation, which will turn faces into inverse faces and vice versa, works for all piece types.

Also note that X-centres and Wings are indistinguishable by their symmetry. These are in essence two copies of the same type of piece, just as a Multi 5×5×5 contains two sets of corners, the 3×3×3 corners and the 5×5×5 corners. In other complex puzzles, it is even possible for an

---

9http://users.skynet.be/moz071262/Applets/MagicPolyhedra
11https://www.jaapsch.net/puzzles/symmetr1.htm
**Table 6.** A list of all symmetry conjugacy classes that can be found as piece type symmetries in the Complex $N \times N \times N$ puzzles.

<table>
<thead>
<tr>
<th>First Found</th>
<th>Complex $N \times N \times N$</th>
<th>Super Complex $3 \times 3 \times 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 = N</td>
<td>6 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
<tr>
<td>5 = N</td>
<td>5 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
<tr>
<td>4 = N</td>
<td>4 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
<tr>
<td>3 = N</td>
<td>3 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
<tr>
<td>2 = N</td>
<td>2 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
<tr>
<td>1 = N</td>
<td>1 = N</td>
<td>$12 \times$ Super $3 \times 3 \times 3$</td>
</tr>
</tbody>
</table>

Table 6: Comparison of permutation counts between the Super $3 \times 3 \times 3$ and Complex $3 \times 3 \times 3$.
X-centre and a Wing to merge to create a new piece type. So the distinction between X-centre and Wing can become rather ambiguous. Finally, for puzzles with cubical symmetry the maximal orbit size is 24 positions. This is exceeded only by the obliques, so they always come in sets which consist of two orbits.

Also seen in Table 7 is the first Multi $N \times N \times N$ and Complex $N \times N \times N$ where each family type first appears. The bifloors, quadwalls, and triwalls only ever appear in the Complex $N \times N \times N$ as imaginary parts and have no counterparts in normal $N \times N \times N$ puzzles.

9 Putting It All Together

In May 2017, shortly before the publication of this article, I conceived of a possible mechanism for the Complex $3 \times 3 \times 3$. A rendering of this potential mechanism, modelled in SOLIDWORKS, is shown in Figure 13. Note that parts were still being 3D printed at the time of publication, so this is currently an unproven design.

![Figure 13. A rendering of a potential mechanism for a physical Complex $3 \times 3 \times 3$.](image)

The core of this design is a Slice $3 \times 3 \times 3$ which makes use of circles to reveal the central cubie, which is itself a puzzle that has never been made before. I stated in a 2013 TwistyPuzzles.com post that: ‘I feel something like the Complex $3 \times 3 \times 3$ is going to have to be solved in stages. The first stepping stone is the SuperMulti-$5 \times 5 \times 5$. The Slice-turn-only Circle $3 \times 3 \times 3$ is another.’

If the design shown in Figure 13 actually works, then I will have realised a dream that I have been working towards for a very long time.

Further, its design will have followed the predicted path of stepping stones with surprising accuracy.

10 Conclusion

A thorough investigation of the $3 \times 3 \times 3$, a puzzle with three orthogonal axes and two independent layers of rotation per axis, does reveal a much richer area of exploration than initially assumed based solely on the visual appearance of a Rubik’s Cube. We have explored two methods used to analyse twisty puzzles and seen how both have been used to reveal the existence of new piece types, virtual and imaginary. Also, we have applied these methods to the $3 \times 3 \times 3$, creating the Complex $3 \times 3 \times 3$, a puzzle with 64 pieces across ten different piece types.

These methods are general enough that they can be applied to other puzzles and other geometries. For example, the Complex $3 \times 3 \times 3$ has $2^8 = 256$ pieces across 15 different piece types and it too can be viewed as a subset of the Multi $5 \times 5 \times 5 \times 5$ with a new definition of a cell turn. The Complex $5 \times 5 \times 5$ contains $2^{12} = 4096$ pieces across $(\binom{12}{3}) = 220$ piece types.

The challenge of understanding all these new pieces and trying to develop mechanisms which will bring them into physical puzzles is a very rewarding process. The Augmented Skewb is now available in physical form and the hunt for a viable mechanism for the Complex $3 \times 3 \times 3$ continues. If you have suggestions on how this might be accomplished, then please contact me. I would be happy to assist with the development of a program that simulates the operation of the Complex $3 \times 3 \times 3$.

11 Acknowledgements

I would like to thank Konrad Sassen for supplying photos of a Skewb. I also wish to thank all of the following for their roles in the development of the Complex $3 \times 3 \times 3$ over the years since it was first gleaned from theory: Matt Galla, Andreas Nortmann, Landon Kryger, Brandon Enright, Nan Ma and everyone else who has posted or asked questions about the Complex $3 \times 3 \times 3$. I also wish to thank Jaap Scherphuis, whose rigorous computational analyses of puzzles in general inspired me long before the Complex $3 \times 3 \times 3$ was discovered.

---

References


Carl Hoff is an epitaxy process service engineer for Applied Materials currently supporting Global Foundries in Malta, New York, United States. Research interests include twisty puzzle design and theory, packing puzzles, and puzzle rings. Address: 108 North Broadway, Malta, NY, 12020-5200, USA. Email: carl.n.hoff@gmail.com

Glossary

This Glossary defines key terms used throughout the paper. See\(^1\)\(^2\) for more details. The conjugacy class of key elements is shown in Table 7.

**Augmented** A puzzle containing all of the real and virtual pieces possible for a given geometry.

**Bifloor** A piece type. In the Complex \(N\times N\times N\) puzzles, these appear exclusively as imaginary pieces. An example is shown in Figure\(^9\).

**Complex** Puzzle containing all possible real, virtual and imaginary pieces for a given geometry.

**Core** A piece type. In the Complex \(N\times N\times N\) puzzles, these can appear as real or imaginary pieces. An example is shown in Figure\(^9\).

**Corner** A piece type. In the Complex \(N\times N\times N\) puzzles, these can appear as real or imaginary pieces. An example is shown in Figure\(^9\).

**Dependent Layer** A twisty puzzle is cut into layers via cut plains or surfaces along a given axis. One layer is picked to be the dependent layer. All possible rotations of this layer can be expressed as some linear combination of the rotations of the other layers on that axis plus a global rotation of the entire puzzle about that same axis.

**Edge** A piece type. In the Complex \(N\times N\times N\) puzzles, these can appear as real or imaginary pieces. An example is shown in Figure\(^9\).

**Face** A piece type. In the Complex \(N\times N\times N\) puzzles, these can appear as real or imaginary pieces. An example is shown in Figure\(^9\).

**Face Layer** The collection of outermost surface pieces in a twisty puzzle which rotates about an axis coming out of a face of the puzzle.

**Grip** A collection of pieces that rotates about an axis in a twisty puzzle. A grip can be located at a face, vertex, edge, slice, etc. A collection of grips and the positions they are allowed to rotate to defines a given puzzle geometry.

**Holding Point** The piece in a puzzle which remains stationary under rotation of all the independent layers of rotation.

\(^{15}\)http://twistypedia.oskarvandeventer.nl

\(^{16}\)https://www.speedsolving.com/wiki/index.php/Category:Puzzle_Parts
**Imaginary Piece** A new piece identified by Matt Galla’s Analysis of Twistability that has not already been defined as a real or virtual piece.

**Layer** A collection of pieces which rotates about an axis in a twisty puzzle.

**Linearly Independent Layer** A layer which remains on a given axis after the dependent layer has been assigned.

**Multi** A puzzle containing all the real pieces possible for a given geometry.

**Oblique** A piece type. In the Complex $N \times N \times N$ puzzles, these can be real or imaginary pieces. An oblique is highlighted in the $7 \times 7 \times 7$ in Figure 14.

**Orbit** The set of positions that a piece can be sent to via layer rotations. This excludes global rotations of the entire puzzle.

**Permutation Count** The number of reachable states of a twisty puzzle.

**Quadwall** A piece type. In the Complex $N \times N \times N$ puzzles, these appear exclusively as imaginary pieces. The first examples of quadwalls appear in the Complex $5 \times 5 \times 5$. The simplest subset of the Complex $5 \times 5 \times 5$ to contain a quadwall is the Slice $5 \times 5 \times 5$. A quadwall is highlighted in Figure 15.

**Real Piece** A piece identified by a non-zero volume of 3-space that is created by the cut planes of the twisty puzzle.

**Signature** In Nortmann’s Analysis of Twistability, this equates to the set of dependent layers of rotation that leaves a given piece stationary. In Galla’s Analysis of Twistability, this equates to the set of linearly independent layers of rotation for a twisty puzzle which moves a given piece.

---

17 Quadwalls are discussed in detail at: http://www.twistypuzzles.com/forum/viewtopic.php?p=363792#p363792
**T-Centre** A piece type. In the Complex $N \times N \times N$ puzzles, these can be real or imaginary pieces. A single T-centre is highlighted in Figure 17.

![Figure 17. A T-centre.](image)

**Triwall** A piece type. In the Complex $N \times N \times N$ puzzles, these appears exclusively as imaginary pieces. An example is shown in Figure 9.

**Virtual Piece** A new piece identified by Andreas Nortmann’s Analysis of Twistability that has not already been defined as a real piece.

**Wing** A piece type. In the Complex $N \times N \times N$ puzzles, these can be real or imaginary pieces. A single wing is highlighted in Figure 18.

![Figure 18. A wing.](image)

**X-centre** A piece type. In the Complex $N \times N \times N$ puzzles, these can be real or imaginary pieces. A single X-centre is highlighted in Figure 19.

![Figure 19. An X-centre.](image)

**Gadeiro Challenge #5**

Pack the pieces on the right to fill the shape on the left. Gadeiro is described next, on pages 39–41.
From Mathematical Proof to Puzzle

Néstor Romeral Andrés, Nestorgames

This short note describes how an investigation into the construction of a geometric figure inadvertently led to the design of a published game, demonstrating that inspiration for games can occur where you least expect it.

1 Introduction

The Al-Quds Star, shown in Figure 1, is a symbol found on a number of emblems and flags, based on the Muslim symbol Rub El Hizb. In Arabic, rub means one fourth or quarter, while hizb means a group or party. This figure is also called Solomon’s Star or Gadeiro. In my home country of Spain, it can be found on pavements, walls and ornaments in many regions, and it is the official symbol for the city of Teruel.

I was fascinated by this shape as soon as I saw it, and wanted to explore its geometry. This paper describes how this exploration led to the creation of new puzzle game.

2 Geometric Construction

Figure 2 shows the construction of the Gadeiro. It is based on two square frames, one of which is rotated 45° to the other, then their combined inner area is removed to give the final shape. Each square frame is defined by an inner and outer square, and the width of the frames is defined by the shape itself, as the corners of the smaller inner squares coincide with the edge midpoints of the larger outer squares (marked a in the Figure).

2.1 Proof by Geometry

Once I understood this construction, an interesting question arose: is the area of the inner (white) shape equal to the area of the outer (red) shape?

First, note that the ratio of distances of rotated to non-rotated corners is $\sqrt{2}:1$, as shown in Figure 3.
Since \((\sqrt{2})^2 = 2\), this suggests that the total area enclosed by the outer shape is twice the area of the inner shape, hence the inner (white) and outer (red) shapes both have the same area. I prove this in a more detailed document elsewhere\(^2\).

Another geometrical proof is achieved by observing that the Gadeiro shape fits exactly inside a copy of itself when shrunk to half its area. Continuing this halving series, as per Figure 4, gives the series \(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \ldots\) which approaches 1 as the number of halvings approaches infinity:

\[
\sum_{n=1}^{\infty} \frac{1}{2^n} = 1
\]

This manifestation of Zeno’s famous Dichotomy Paradox \(^1\) demonstrates that the areas of the inner and outer shapes are equal, but is very much a theoretical proof as such a series can never be physically constructed. Could there be a more direct and practical way to prove this equality?

### 2.2 Proof by Game

I then considered a more direct approach: could the outer shape be dissected into pieces and reassembled to fill its interior? Figure 5 shows the best dissection that I have come up with, which I believe could be optimal for this problem. The best dissection will have the fewest number of cuts giving the fewest number of pieces, and I liked the symmetry of the pieces produced.

![Figure 5. Dissection of the Gadeiro into pieces.](image)

3. A Game is Born

Apart from providing a nice proof of inner/outer area equality, these pieces have found a life of their own as a puzzle game called Gadeiro, which has now been published \(^2\). This is a form of Tangram puzzle \(^3\) in which players use the pieces to form given silhouette shapes, such as those shown in Figure 7. Figure 2 shows a solution for one of these challenges.

This exercise shows how games can emerge from the most unlikely inspirations!

---

\(^2\)http://www.gapdjournal.com/issues/issue-3-1/gadeiro.pdf

\(^1\)http://www.gapdjournal.com/issues/issue-3-1/gadeiro.pdf

\(^3\)http://www.gapdjournal.com/issues/issue-3-1/gadeiro.pdf
Figure 7. Some Gadeiro challenge shapes.

Figure 8. Solution of a Gadeiro challenge.

Acknowledgements

Thanks to the Editor-in-Chief for compiling and formatting this piece.

References


Néstor Romeral Andrés is the owner and director of independent game publisher Nestorgames. He has designed over 70 board games and was awarded Spanish Game Designer of the Year for 2009–10. Address: c/Mar de las Antillas 6, 3-2, Zaragoza, 50012, Spain. Email: info@nestorgames.com

Gadeiro Challenge #6

Pack the pieces on the right to fill the shape on the left.
1 Introduction

The game CliqueR was designed to familiarise students with certain concepts from graph theory. Once designed, the game showed a surprising strategic depth and also was an excellent domain for search-based procedural content generation [1, 2]. This article will give the rules of CliqueR and the motives for its initial development, explain the graph theory that underlies the design of CliqueR boards, and sketch a system for automatically generating CliqueR boards with controlled properties.

1. Decide who moves first (e.g., by coin toss).
2. Players take turns moving if possible.
3. A move consists of drawing a line between two unconnected dots.
4. The line must not cross another line.
5. Completing a clique scores: 2-clique = 1 pt, 3-clique = 3 pts, 4-clique = 5 pts.
6. All cliques completed per move count.

Notice that drawing a line scores one point, since that is completing a 2-clique. Rule 6 means that if you complete a 3-clique, you get one point for the 2-clique (the line) and three more for completing the 3-clique. Similarly, completing a 4-clique requires you not only to draw the line that completes it but also to complete two 3-cliques as part of completing the four clique for a total of 1+3+3+5=12 points. Some sequences of play for the board shown in Figure 2 permit one move to complete two four-cliques for a total score of 20 points – this happened during play-testing.

An example of a CliqueR board appears in Figure 2. A board consists of a placement of dots on a square with the property that no three of the dots lie on the same line. In order to explain the rules of CliqueR, we define the term clique. A clique is a set of dots such that every pair of dots is connected by an edge. Examples of two, three, and four vertex cliques are shown in Figure 1.

A mathematical fact that informs the design of CliqueR is that it is impossible to draw a 5-clique (five vertices with all possible connections) without at least one pair of edges crossing. The rules are given below.
A relatively small number of games based on cliques in graphs appear in the literature. In [4], several clique-based games, none of them similar to CliqueR, are defined. None of these games involve non-crossing edge rules, giving them a very different character.

2 Mathematics of CliqueR Design

The key to understanding that there is a non-trivial design problem for CliqueR boards is the fact that there are exactly two possible topological states for four points in the plane [3], at least as long as no three of the points fall on the same line. Versions of these two possible states are shown in Figure 3. The number of sets of four points on a CliqueR board that have each of these configurations varies substantially with the placement of the points. In general, the more tetrahedral configurations present, the deeper the strategy required to play the board and the higher the maximum possible score is.

The way to tell which of the two configurations a set of four points exemplifies is to consider the following: a configuration is tetrahedral if one point is in the interior of the triangle formed by the other three. If none of the points are in the interior of the triangle formed by the other three then the configuration is quadrilateral. Tetrahedral configurations are also sometimes called non-crossing configurations while quadrilateral ones are called crossing configurations, after the way the segments joining the points do or do not cross.

Consider the CliqueR board shown in Figure 4. The vertices are arranged on a regular polygon. This configuration is known to maximise the number of quadrilateral configurations – in fact, it is easy to see, using the left hand image, that the interior of three points in this configuration will not contain any of the others in its interior. On other CliqueR boards, the strategy is to set up your ability to complete a 4-clique; this board has none. This means this regular polygon, or in fact any convex polygonal layout, yields the strategically simplest CliqueR board.

An example of a CliqueR board with many tetrahedral configurations appears in Figure 5. This board was constructed using the following greedy algorithm. Start with a triangle. For the remaining points, place a point, avoiding putting three points on a line, inside the largest number of triangles you can. Tracing through the construction, we start with \( \triangle 123 \). Point A can go anywhere inside the first triangle, placing it off-centre makes it easier to make later placements.

At this stage, every point inside the original triangle – other than one that would violate the ‘no three points on a line’ rule – is inside two triangles. This makes point B as good or as bad a choice as any other. Once B is placed there are now points inside two triangles and points inside three triangles. Point C is inside \( \triangle 123, \triangle 12A, \) and \( \triangle 1AB \), letting it satisfy the placement inside the largest number of triangles criterion. In general, when we place a point, it forms three new triangles. The interior points of these are inside the largest number of triangles possible, and so the point D inside \( \triangle ABC \) is a correct choice.

The configurations in Figures 4 and 5 demonstrate that there can be a large variation in the number of tetrahedral configurations. The greedy algorithm is also a nice algorithm for quick hand design of a CliqueR board with the potential for lots of 4-clique scores. The extreme values for having zero tetrahedral configurations occurs whenever the points are arranged as a regular polygon. It turns out that the other extreme, finding configurations with a maximal number of tetrahedral
configurations, is a famous unsolved problem in mathematics, called Sylvester’s Four-Point Problem [5]. The exact answer is known up to 27 points and the values are available in the Online Encyclopedia of Integer Sequences (OEIS) [1].

The OEIS was founded in 1964 by N. J. A. Sloane, as a printed book, listing known sequences that count something interesting. The sequences in the OEIS come up in many different contexts and the authors of this paper have often found them relevant for designing combinatorial games – including CliqueR. At the time of writing this article, some 284,776 sequences are in the OEIS, which has an excellent search function. Searching a sequence returns the sequence, other related sequences, citations of papers about the sequence, the formula and generating function if they are known, and an explanation of known meaning(s) of the sequence. Mathematical papers have sometimes arisen from unexpected connections generated by the use of the OEIS.

This process of discovery relies on the fact that the same sequence may count many different things. The Fibonacci numbers, for example, count both the number of different ways to fill a $2 \times N$ rectangle with $1 \times 2$ rectangles and the number of petals at a given distance from the centre of some types of flowers. Sequences in the OEIS are assigned identity numbers. The sequence giving the minimum possible numbers of quadrilateral configurations in any arrangement of $n$ points (with no three points on a line) has identifier A014540. This is the sequence used to determine the maximum number of possible tetrahedral configurations possible in CliqueR.

3 Automatic Design of Boards

For aesthetic reasons, we force an odd number of points laid out in a bilaterally symmetric fashion for the automatically designed CliqueR boards. The smallest number of points that yields non-trivial CliqueR play is $n = 7$ and so we design boards with 7, 9, 11, 13 and 15 points. When $n < 6$ points are used it is possible for an experienced human player to look ahead to the end of the game and so the game assumes a Tic-Tac-Toe-like character. Figure 2 shows an automatically designed board with seven points.

The design process uses evolutionary algorithms [6] for search of good boards. Evolutionary algorithms are algorithms based on Darwin’s theory of evolution [7]. The algorithm operates with the following steps.

1. Generate 1000 symmetric CliqueR Boards.
2. Evaluate the quality of these boards.
3. Repeat 40,000 times:
   4. Pick seven boards at random.
   5. Delete the two worst.
   6. Copy the two best.
   7. Mix and match points in the copies.
   8. Replace a point randomly.
   9. Evaluate the quality of the new boards.
10. End of Repeat

The critical thing here is the steps that say ‘Evaluate the quality’. This quality measure is called the fitness function of the evolutionary algorithm. It is based on being able to compute, in a board, the number of tetrahedral configurations while knowing the maximum number of tetrahedral configurations possible. In each run of the evolutionary algorithm a fraction of the maximum possible number of tetrahedral configurations was picked as a target. One board was judged as better than another if it came closer to having this target number of tetrahedral configurations. In order to get a board with relatively simple strategy, the fraction of possible tetrahedral configurations was set low; near zero – to find a board with deeper strategy it was set high; near one.

Using a target fraction of the maximum number of possible tetrahedral configurations permits us to tune the degree of challenge posed by the automatically located CliqueR boards. This is one of the nicest features of evolutionary computation as a design tool. It permits the designer to change the request to the search software by relatively simple editing of the fitness function.

3.1 A Technical Challenge

The description of the evolutionary algorithm is fairly high level and the first author will be happy to supply code on request. There is a technical aspect of the algorithm that is worth a warning to the wise.

The core step of evaluating the fitness of boards to drive the evolutionary search for the desired type of boards is being able to sort out if each of the sets of four points on a board are a tetrahedral or a quadrilateral configuration. For any four given points, there are three ways to pick two sets of two points. To test a set of points, consider the three sets of two pairs of points individually. Each defines two lines that intersect somewhere or are parallel – this latter case we think of as intersecting at infinity. A pair of lines

[1]https://www.oeis.org

404 Game & Puzzle Design Vol. 3, no. 1, 2017
cross in the interior of the configuration, indicating a quadrilateral configuration, if and only if the intersection of the two lines is in the interior of both the line segments defined by the pair of points. This is a bit dense, but actually involves only high school algebra.

Here is the warning: do not use general point positions and real arithmetic to do these computations. Write the code to compare integers and use only points with integer coordinates. The evolutionary algorithm used in this work was able to exploit round-off error in real arithmetic to make a quadrilateral configuration appear tetrahedral. Worse, general point positions often produce pairs of lines that intersect at such small angles as to appear to be the same line. It is not hard to show that each such awful pair of lines on a CliqueR board have a topologically equivalent board without awful pairs of lines, and in fact where the points have integer coordinates. Code for performing the tetrahedral/quadrilateral determination on sets of four points is part of the code available from the first author.

3.2 Automating Game Design

The concept of automating the design of games, with evolution or other computational techniques, is not a novel one. Examples of this sort of work include the following. The authors of [8] used what was then a novel approach of using genetic algorithms to search the space of board games rules in order to design balanced board games. It was shown that with equal computational effort, the genetic algorithms produced what the author define as better board games than a random search strategy.

In [9], the authors investigate automatic evaluation of game quality relative to what a human being would find interesting, and use evolution to search the space of combinatorial games. Automatic techniques have also been applied outside the combinatorial game design field. The authors of [10] created a formalisation and a context-free grammar for general card games, then used stochastic processes to generate novel card games based on rules from existing games like Texas Hold’Em and UNO. More recently, the authors in [11] focussed on formalising game mechanics, and then applying automatic processes to search the space of game mechanisms, to create a role-playing platformer video game.

4 Automatically Designed Boards

In this section we exhibit some of the automatically designed boards with comments about their character. We start with Figures 6, 7 and 8 which show CliqueR boards of 15 points, designed to have 10%, 30%, and 80% of the possible number of tetrahedral groups of four points. These target fractions of available tetrahedral configurations were chosen to illustrate some of the characteristics of the designed boards.

Figure 6. A size 15 board with target fraction 10%.

Figure 7. Size 15 board with target fraction 30%.

Figure 8. Size 15 board with target fraction 80%.
Notice that the board shown in Figure 6 which targets having a low fraction of the possible tetrahedral configurations, is geometrically similar to the regular polygon configuration. In fact, this board is essentially a convex 13-gon with two points displaced inside. The board shown in Figure 7 with a tetrahedral configuration target of 30% consists of an outer septagon with an inner octagon. Remember that all the groups of four points that occur in a CliqueR board that is laid out as a convex polygon form quadrilateral, rather than tetrahedral, configurations. The 15-point board designed with a 80% tetrahedral target fraction does not have a simple polygonal decomposition and possesses far more tetrahedral configurations as a result.

These three examples demonstrate that the design algorithm is working in a nominal fashion, getting close to its target percentages, which in turn means that the strategy for being able to control the strategy depth of automatically designed boards is working. We now turn to CliqueR boards with fewer points.

4.1 Smaller Boards

Figures 9, 10, and 11 show boards designed with 9, 11 and 13 points for targets of 40%, 90%, and 20% of the maximum number of tetrahedral configurations. The nine-point board has a moderate target fraction and produces three triangles that have fairly obvious tetrahedral configurations. This board might make a good one for introducing strategy for capturing 4-cliques.

The board shown in Figure 10 was designed with a high target fraction, but exhibited an annoying property: some of the tetrahedral conditions contain lines that are very close to one another. This situation is common both in boards with high target fractions and with more points. Section 6 suggests a possible way of getting the automatic design software to avoid this problem.

The board with 13 points shown in Figure 11 was created with a relatively low target fraction of 20%. We see that the phenomenon of designing a polygon with a few points inside occurs again in this case. This suggests a good hand design strategy: start with a convex polygon and place a few points inside it.

5 CliqueR Variants

A natural variant of CliqueR is to play the game as a solitaire – but only some boards are interesting for solitaire play. Examine the completed solitaire shown in Figure 12. This board uses the regular polygon layout, meaning that there are no 4-cliques possible. This means that any completed solitaire will be a triangulation of the polygon – that is a partition of the polygon into triangles. The number of triangles (3-cliques) in such a completed solitaire is always two less than the number of points and the number of edges is always twice the number of points less three.
This means that for such a board the solitaire score is $3(n - 2) + 2n - 3 = 5n - 9$ which may be computed before the start of play. This score is obtained by following the rules of CliqueR without worrying about how to play. The corollary of this is that interesting solitaires require that many tetrahedral configurations be possible.

When there are tetrahedral configurations, obtaining some 4-cliques block others, yielding a solitaire game with strategy.

5.1 Expert CliqueR: Curved Lines

This variation of CliqueR makes both the two player and solitaire versions more challenging: relax the constraint that players must use straight lines. When this constraint is relaxed and the players can use curved lines, then the formerly crossing configurations can be completed to a 4-clique in the fashion shown in Figure 13.

While curved lines are allowed in Expert CliqueR, the lines must still not cross one another. In this situation, 5-cliques remain impossible, and so the scoring system remains the same. The design process for Expert CliqueR is different. Instead of designing to the number of tetrahedral configurations, a more sophisticated, simulation based fitness function would be required to evolve boards for Expert CliqueR.

5.2 Limited CliqueR

Another possibility for varying the play of CliqueR would be to draw in light grey guidelines and only permit play using those lines. This would permit far finer control over the strategic possibilities. For smaller numbers of points this would probably not enhance playability or enjoyment of the game, but for large numbers of points – 13 or more – this might restore playability in what might otherwise be an extremely messy game. This limiting technique could be applied to the standard or expert versions of CliqueR.

5.3 Chromatic CliqueR

In Chromatic CliqueR, the moves made by the players are drawn in distinct colours. It is still the case that only one line may be drawn between two vertices, but lines of different colours are permitted to cross one another. This variation of CliqueR would permit larger scoring cliques. The rules are amended as followed.

2. Players move if possible, each using their own colour.

4. The line must not cross another line of the same colour.

5. Completing a clique scores: 2-clique = 1 pt, 3-clique = 3 pts, 4-clique = 5 pts, 5-clique = 7 pts, etc.

This variation of CliqueR may be applied to all of the versions of CliqueR given previously. It is an excellent target for limited CliqueR; the scoring of this version can become somewhat challenging.

6 Next Steps

For each of the numbers of points and target tetrahedral fractions, the evolutionary algorithm was run 30 times, yielding a large selection of possible CliqueR boards for the parameter set. The board shown in Figure 14 is problematic and illustrates a problem with the automatic design strategy. When playing CliqueR on paper with pencils, it is problematic if it is difficult to separate lines: the illustrative board has a large problem with close lines. All possible lines that might be drawn during play are shown. Another board from the same collection of runs of the evolutionary algorithm is shown in Figure 15. This board has far
fewer pairs of lines with a small angle between them.

Figure 14. Size 15 board with target fraction 80%.

As noted before, one of the nice properties of evolutionary algorithms as design tools is that it is not difficult to revise the fitness function to encode additional desirable properties. What differentiates the two examples of a good and a bad CliqueR board? The number of angles between play lines that are below a critical ‘annoying’ value. It is easy to compute these angles and so a second fitness criterion could be added. When two configurations have the same number of tetrahedral configurations, break the tie by minimising the number of angles below, say, $15^\circ$.

Figure 15. Size 15 board with target fraction 80%.

Some work will be required to implement this good angular behaviour criterion – for example, it may be that small angles on long lines are less important, because long lines tend to spoil many potential tetrahedral scoring configurations and so will not, in practice, arise during play. Another concern is that using the angular character of boards to break ties might have little effect.

The primary fitness function – in effect the number of tetrahedral configurations – are a discrete function and so ties are possible. In an evolving population, members of the population tend to have fitness values near the current best which also ensures that there will be ties and use of the tie-breaking fitness function.

Another factor that might be valuable to control for smaller games is to ensure that the points are dispersed so that the board occupies much of the design space. The board in Figure 16 illustrates this problem. This failure to effectively use the design space could be corrected by modifying the fitness function, but there is a better way. Expanding or contracting a board in the vertical or horizontal number does not change the tetrahedral or quadrilateral character of the sets of 4 points in the board. This means that a simple affine deformation of the board can solve this problem – and may also be able to help to a limited degree with the small angle problem. This sort of adjustment can be performed before the automatically designed configurations are saved.

Figure 16. Size 11 board with target fraction 50%.

Each of the variations of CliqueR defined poses a distinct design challenge. An interesting question is to determine if boards designed for standard CliqueR are good boards for expert CliqueR. The design strategy for Chromatic CliqueR is most like that for standard CliqueR. The other variants will all probably require simulation based automatic design, in effect using simple artificial intelligence agents (AIs) that play the game as fitness functions. A natural choice for this AI would be Monte-Carlo Tree Search [12, 13].

The automatically designed board in this article all have bilateral symmetry and an odd number of points. This was done both because it improved the appearance of the boards over those with fully random point placement and probably makes the game somewhat easier to play. Relaxing or replacing this constraint is a good target for
future work. Boards with rotational, rather than bilateral, symmetry might have very interesting appearances.

Another topic for the future is playing CliqueR with more than two players. It seems intuitive that this would substantially increase the strategic complexity. It might also be interesting to attempt a full strategic analysis of CliqueR on a convex polygon. The strategy for these boards – those with no tetrahedral configurations – may be simple enough to completely solve the game.

7 Conclusion

The game CliqueR was originally designed as a way of familiarising undergraduate students with some of the mental processes needed to think about graph theory. An enthusiastic response to the game from students in a class taught by the second author – who invented the game – suggests that it is useful for this purpose. The mathematics of topological graph theory contains a good deal of knowledge, reported in this article, that informs the character and design of CliqueR boards. The game serves as a wonderful application of the, somewhat abstract, graph theory. Sylvester’s Four-Point Problem is not obviously useful in game and puzzle design, but it is at the core of the design principles for CliqueR design.

A topic treated lightly in this article is the hand design of CliqueR boards. The greedy algorithm for finding boards with many tetrahedral configurations is a starting point. Hand design is especially likely to be useful in Limited CliqueR.

7.1 Code Availability

During the completion of this article, a number of possible improvements to the CliqueR code were devised but have not yet been implemented. Requests to the first author for CliqueR design code will be honoured as soon as the code has been properly revised and commented.

Acknowledgements

The authors thank the Canadian Natural Sciences and Engineering Research Council and the University of Guelph for their support of this work.

References

Gadeiro Challenges #7 and #8

Pack the pieces on the right to fill the shapes on the left. Gadeiro is described on pages 39–41.
Reinvent the Wheel

Cameron Browne, Queensland University of Technology (QUT)

This article presents the case that most new games are created from novel variations or combinations of a limited core of fundamental mechanisms, and that reinventing even well-known mechanisms – not always in optimal or expected ways – can provide a rich source for new designs. Mutation and crossover are identified as two common modes of reinvention, applied to rules and equipment.

1 Introduction

My editorial for the previous issue of Game & Puzzle Design, titled ‘Nothing New Under the Sun’ [9], explored the notion that most new games are created from novel combinations of a relatively small core of known mechanisms. If you study any game, you will probably recognise the elements that make it up from previous games, but applied in a new context.

Original ideas that truly transform the gaming landscape are rare [2]. Board game historian Irving Finkel states that the idea of renting out a square (as in Monopoly) was the last ‘momentous’ innovation in board games [2]. However, I would add to that the concept of connection, which has led to hundreds of new games over recent decades that constitute the relatively new genre of connection games [3], and other innovations including cooperative games, deck-building card games, simulation games, and so on.

There are now thousands of game designers worldwide producing hundreds of thousands of new games per year of all types. The design space has been so closely examined and so well trodden by so many inventive minds that it can sometimes feel that the good ideas have already been discovered. This is obviously not the case – good new games continue to be produced – but it is likely that the most useful core mechanisms have already been discovered.

One way to progress is to identify the key mechanisms and refine them to their fundamental forms, applying the typical game designer’s urge for elegance and simplicity [4]. But it is hard to think of simpler forms for the more well-used mechanisms, such as N-in-a-row, piece hopping, capture by replacement, trading a card, etc.

A more fruitful way to progress may be to start with mechanisms known to work and to build away from them, to find interesting variations nearby in the design space. This process of ‘complexifying’ ideas may run counter to the designer’s desire for simplicity. However, the trick is to look at more complex variations to find something original and interesting, then find the simplest valuable form of these ‘complexified’ variants. In the end, game design is the art of creating complexity from nothing: a Draughts set is just a piece of cardboard and a handful of plastic disks until the rules are added.

1.1 Reinventing the Wheel

Figure 1 shows a favourite comic strip from my childhood. It is an episode of B.C. by Johnny Hart, in which Thor the inventor presents his new triangular wheel, and describes it as an improvement on the square wheel as ‘it eliminates one bump’ [5].

Figure 1. Reinventing the wheel. Reprinted with permission of Ita Hart Trust (all rights reserved).
This comic strip is cited in design theory as an example of a mindset ‘going the wrong way’ [6]. But I think there is more to it than that, and see it as a brilliant demonstration that even suboptimal designs can have unexpected benefits when viewed in the right light. I don’t know how many games have resulted from designers trying something ridiculous only to find that it worked much better than expected.

This article promotes the practice of looking for inspiration by modifying known ideas away from their fundamental forms, as that is a more realistic way to go. In this context, the ‘wheels’ to be reinvented are fundamental game mechanisms and pieces of equipment.

This article complements previous articles in this series as follows. ‘Explore the Design Space’ [7] discussed how to effectively explore the design space, ‘Bug or Feature’ [10] discussed what to look for in new designs, and ‘Reinvent the Wheel’ focusses on where to start looking in the design space.

2 Modes of Reinvention

Given a fundamental mechanism that we wish to reinvent, it is useful to identify ways in which it can be modified, so that the resulting variants can be classified into meaningful categories.

The most obvious ways to reinvent a mechanism or piece of equipment – modifying its parameters and combining it with others – have obvious similarity to the standard genetic operators of mutation and crossover [9]. This suggests four basic modes of reinvention:

1. **Mutation:**
   (a) **Rules:** Conceptual variation in the way that a rule is applied.
   (b) **Equipment:** Variation in the way that equipment is used.

2. **Crossover:**
   (a) **Rules:** Reuse of rules in conjunction with rules from other contexts.
   (b) **Equipment:** Reuse of equipment (and the rules they imply) in other contexts.

All of these modes can be seen as applications of combinatorial creativity [10]; given a mechanism or piece of equipment, what degrees of freedom can be modified, and what other contexts can it be combined with? The steps for achieving this have been covered elsewhere [7], so the following examples are more to demonstrate the range of reinvented mechanisms and the prevalence of reinvention in games of many types.

3 Examples

The following examples demonstrate the reinvention of well-known mechanisms and equipment in a range of games, through mutation and crossover. Many are taken from Schmittberger’s classic book New Rules for Classic Games [11], which is a rich source of reinvented game ideas.

Finding examples of reinvention in games is easy; pick any game at random and it is likely to contain something familiar. The following examples below have been carefully selected to clearly delineate the four modes of reinvention.

3.1 Mutation of Rules

The mutation of rules is the most prevalent mode of reinvention. It is easier – and cheaper! – to modify and test variations of ideas than equipment, and the mutation of individual elements allows more freedom than having to find compatible elements with which to cross them over.

3.1.1 N-in-a-Row

The concept of forming N-in-a-row pieces is one of the most fundamental and widely known game mechanisms. The BoardGameGeek geeklist ‘the Tic Tac Toe Family Reunion’ lists hundreds of examples of games that use it. Figure 2 shows one such mutation, called Yavalath, which was invented by the computer program LUDI and has proven popular with players [12].

![Figure 2. Black cannot block White in Yavalath.](https://www.boardgamegeek.com/geeklist/7522/tic-tac-toe-family-reunion)

Yavalath employs the familiar N-in-a-row rule, but adds the key enhancement that players lose if they make (N-1)-in-a-row beforehand. Players immediately understand the basic mechanism but are usually surprised by the resulting
emergent strategies. For example, Black would like to block White’s potential 4-in-a-row by playing at the dotted cell in the example above, but cannot as that would create a losing line of 3. Players can manipulate their opponent’s moves through judicious play.

This *misère* version \[13\] of the basic mechanism, i.e. lose with \(N\)-in-a-row, does not work very well in itself, and results in cold games in which the board fills up until a player is forced to make a losing move. This modification would therefore seem of little value, and a step in the wrong direction for the reinvention of the \(N\)-in-a-row ‘wheel’. Perhaps this is why it took a computer program, that does not have such biases, to find the successful combination of ‘win with \(N\)-in-a-row but lose with \((N-1)\)-in-a-row’.

A more recent reinvention of the \(N\)-in-a-row mechanism is found in the game LOT (Line of Three) by Néstor Romeral Andrés \[14\]. In this game, making a line of three pieces does not immediately win the game, but allows the mover to promote one of the line pieces into a double-piece stack, and players win by making three such stacks in a row. For example, Dark wins the game shown in Figure 3.

![Figure 3. Dark wins this game of LOT.](image)

This use of a known goal mechanism to trigger some behaviour other than an outright win adds a layer of indirection that provides a deeper game with little additional cost. A similar idea can be seen in the GIPF game Yinsh \[15\], in which achieving 5-in-a-row does not win the game immediately but triggers the removal of a different type of piece, which is a step towards winning but also reduces movement options.

### 3.2 Piece Hopping

Hopping over pieces to capture them, as in Draughts, is another fundamental mechanism that will be familiar to most players. One of the most elegant and prominent variations on this idea can be found in the Draughts variant Laska, designed by World Chess Champion Emanuel Lasker in 1911 \[16\] pp. 271–272].

![Figure 4. Laska jumps take a jumpee with them.](image)

In this variant, the topmost piece jumped over is not captured, but is instead placed under the jumping piece where it lands to form a column. Figure 4 shows such a jumping move. This mutation is not only an elegant mechanism in itself, but brilliantly introduces the concept of stacking to Draughts to add another strategic dimension.

Super Chinese Checkers, proposed by R. Wayne Schmittberger \[11\] pp. 8–9, demonstrates another simple mutation of the basic hop mechanism, but based on the dimension of length rather than height. Players can hop in a line over a single piece of any colour provided that the piece jumped over lies at the exact midpoint of the jump. For example, Figure 5 shows a red piece performing three long jumps.

![Figure 5. Triple jump in Super Chinese Checkers.](image)

Schmittberger reports that the modified rule speeds up play and adds many interesting tactical possibilities. A similar extended hop mechanism is also used in the more recent game Vault, al-

\[3\] http://www.nestorgames.com/rulebooks/VAULT_EN.pdf
though they are not constrained to the cardinal directions of the board\footnote{https://www.boardgamegeek.com/boardgame/27938/tortuga}

The game Tortuga by Vincent Everaert demonstrates yet another dimension in which the hop mechanism can be modified: by toggling the state of pieces hopped over. Figure\ref{fig:6} shows a game of Tortuga in progress, with two turtle pieces flipped to show that they have been hopped over; flipped pieces remain inactive until they are hopped over again to unflip them.

The flip/unflip mechanism is an elegant and natural mutation of the hop, and rarely does a mechanism complement the theme of a game so beautifully. The turtle pieces already have a certain zoomorphic charm, and most of us know from experience that a turtle is helpless when flipped onto its back, adding to the thematic consistency\footnote{If one doesn’t think too much about the idea of turtles hopping over each other.} and further charm of the game.

![Figure 6. Tortuga pieces flipped and unflipped.](image)

3.2.1 Chess Variants

The world of Chess variants provides a rich source of fundamental mechanism mutations. The popularity of Chess and the number and variety of piece types has allowed designers to come up with thousands of distinct Chess variants over the years\cite{17}.

One of the simplest Chess mutations is Extinction Chess\cite{11} pp. 19–20], which transposes the winning condition from checkmating the king to eliminating any of the opponents’ piece types, i.e. king, queen, rooks, bishops, knights or pawns (checkmating the king is effectively a subset of this goal). This simple change has a profound effect on the game: sacrificing a queen is an immediate loss; forking say the opponent’s last pawn and last bishop threatens an imminent win; and the fact that pawns can also promote to kings means that a king can be captured and the game continue.

Another simple Chess mutation is to constrain the movement of pieces with dice rolls, as in Dice Chess\cite{11} p. 189], in which players roll a die each turn that dictates which type of piece they must move that turn. This harks back to the Chess-like game Chaturaji from 1030AD in which the movement of pieces was determined by rolling two dice\cite{18}.

Similar mutations include the die rolls dictating instead which player moves, or how many moves the roller gets that turn. These variants could also be seen as the crossover of equipment types (dice with Chess), although dice are a rather ubiquitous method for incorporating nondeterminism in a game.

3.2.2 Nonlinear Scoring

In the game Boggle, players score points according to the length of words they can form from a random set of letters. Schmittberger suggests a simple mutation he calls Big Boggle\cite{11} p. 9] in which only words with five or more letters score points, and these are scored according to the following exponential progression:

\[
\text{Points} = 2^{n-5}
\]

For each word of \(n\) letters, this gives the value shown in Table 1. This change makes players look harder for longer words and makes the game more dramatic\cite{19} by giving trailing players a greater chance of catching up with an especially good move.

<table>
<thead>
<tr>
<th>Letters</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>64</td>
</tr>
<tr>
<td>12</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 1. Exponential Big Boggle scoring.

Backgammon players will recognise this progression from the doubling cube with which the stakes are raised as Backgammon games progress. The doubling cube starts each game by showing the value 1, then is turned to double the value when players feel that they have an advantage and invite the opponent to concede. The game’s final result (win=1, gammon=2, backgammon=3) is then multiplied by the final doubling cube score.
The doubling cube may seem to be a simple scoring mutation but has a huge impact on the game, making games faster, more exciting and more suitable for gambling. Jacoby and Crawford dedicate *The Backgammon Book* (1970) to ‘the genius who invented the doubling cube and made backgammon the game it is’ [20, p. 4], and Schmitberger describes it as ‘the twentieth century’s contribution to the evolution of backgammon’ [11, p. 160]. Each doubling action poses a possibly critical decision for the opponent – whether to accept the double or concede – and it is surprising that this mechanism has not been adopted by more games.

The doubling cube is included in this category, Mutation of Rules, as it is really a modification to the scoring rules of the game rather than the reuse of a standard die, i.e. it is not rolled but used to track the multiplier.

### 3.2.3 Theme Extrapolation

Rule modifications that naturally extend the theme of a game can be natural mutations. For example, Schmittberger describes a Risk variant called Risk with Retreats, in which players have the option of allowing armies that are attacked to retreat to an adjacent country and concede their ground, rather than fighting to the bitter end [11, p. 49]. This modification fits naturally with the game’s theme of conquest and battle, offers additional strategic possibilities, and became Schmittberger’s usual way of playing the game.

A more recent example is the game Piece o’ Cake [6], which does something of the reverse. It takes the well-known pie rule, used as a first move equaliser in many games [3], and extends this theme literally to produce a game about cutting cakes.

### 3.2.4 Geometric Transformations

Simple transformations of relevant geometric patterns can add complexity to a game. For example, Richard Moxham’s game Morelli [21] takes the idea found in many games of forming squares or rectangles with pieces, and extends it by allowing players to form squares at any rotation and not just aligned with the board’s grid. Figure 7 shows an example of a valid white frame formed by a central white piece and four other white pieces defining a (slightly rotated) square around it.

Care must be taken with such mutations, as even simple geometric transformations can be difficult for players to visualise, let alone incorporate in their strategic move planning. Interpreting such transformations can require more mental effort from the player, which is the opposite of making the design do the work [22].

![Figure 7. A (slightly rotated) frame in Morelli.](image)

### 3.2.5 Play Order

The order of play is an implicit rule so fundamental to most games that it is not always stated, beyond comments such as ‘play passes to the left’. My three-player board game Triad [23, pp. 175–177] subverts this fundamental rule with a simple mutation to play order. The hexagonal board cells are shaded in three alternating colours, as shown in Figure 8 and the player to move each turn is the player with the same colour as the cell just landed on by the previous player’s move.

![Figure 8. Colour dictates move order in Triad.](image)

---


7[https://www.mayfairgames.com/products/patchwork](https://www.mayfairgames.com/products/patchwork)
The game Patchwork, by Uwe Rosenberg,
shows a more recent example of variable move
order. Players move tokens around a scoring
track, shown in Figure 9, and the player furthest
behind on the track is the next to move each turn.
This is an elegant and transparent way to keep
the game balanced and prolong the contest.

![Figure 9. The Patchwork scoring track.](image)

Constraints concerning which pieces a player
can move each turn are another obvious aspect of
game control to mutate. For example, Kamisado
players control which piece the opponent must
move next turn according to the colour of the
cell they land on, while Quarto players explicitly
hand their opponent the piece they want them to
play next turn.

![Figure 10. Dice as pieces in Chase.](image)

3.3 Mutation of Equipment

Mutation of equipment involves the reuse of (pos-
sibly modified) equipment for different purposes.

3.3.1 Dice

A common reinvention of dice is as playing pieces
for games, utilising the face values shown to con-
strain movement. One of the original games to
do this was Chase from 1986, shown in Figure 10.
In this game, the uppermost face of each
die shows the exact number of spaces that it can
move.

![Figure 11. A territory card from Risk.](image)

3.3.2 Cards

Cards can contain a wealth of information and
constitute gaming systems in themselves, so
are a rich source of reinvention. For example,
Schmittberger describes the Risk variant Nuclear
Risk, which repurposes the territory cards whose main purpose is for players to collect and trade in for reinforcements.

Nuclear Risk adds another use: if a player
rolls double 4, 5 or 6, then the topmost territory
card is selected and the country shown suffers
a nuclear accident, wiping out all armies occu-
pying it. Armies moving into nuked countries
are halved for the remainder of the game, and
nuked countries do not count toward the control
of continents.

A further mutation of this idea, called Tacti-
cal Nuclear Risk, allows players to play territory
cards to nuke the country shown at will. This has
an even greater impact on the character of the
game, encouraging players to decentralise their
forces, and is probably a more realistic extension thematically.

---

8https://www.boardgamegeek.com/boardgame/38545/kamisado
9http://en.gigamic.com/game/quarto-classic
10https://www.boardgamegeek.com/boardgame/316/chase
Mutations can affect the way in which cards are played. For example, in Bohnanza [24], shown in Figure 12, players are not allowed to reorder the cards in their hands but must play them in the order they are drawn. This adds an interesting new dynamic to the game.\footnote{The fixed card order in Bohnanza is really a rule mutation, but is included here as it is specific to cards.}

![Figure 12. A hand of Bohnanza cards.](image)

In the card game San Juan, shown in Figure 13, cards take on multiple roles, as currency, buildings and resources.\footnote{https://www.boardgamegeek.com/boardgame/8217/san-juan} Seldom is a single component used to such good effect.

![Figure 13. A game of San Juan in progress.](image)

The Combat Commander system of card-based war games is another case in point [25]. Figure 14 shows a typical Combat Commander card and its wealth of information. Cards can:

1. Execute an order.
2. React to another order.
3. Determine which event is triggered.
4. Perform a dice roll.
5. Randomly determine a map location.

3.3.3 Pieces

The recent board game Tak, shown in Figure 15, makes clever reuse of the shape of playing pieces.\footnote{http://www.cheapass.com/tak} Pieces placed on their sides to become standing stones block the cell they occupy from further stacking, but do not count towards their owner’s goal. A precursor of this idea may be found in Shogi (or Japanese Chess), in which the direction in which pieces are placed relative to the players becomes important [26].

![Figure 14. A Combat Commander card [25, p. 3].](image)

![Figure 15. Tak pieces in various orientations.](image)
The gems involve capture rules that are too detailed to describe here, suffice to say that they add a tremendous amount of strategic complexity to the game. Players can be faced with hundreds of possible moves per turn, instead of the usual handful of moves in standard Mancala games.

3.3.4 Tiles

Path-based tile games offer many opportunities for mutation and reuse in the geometry of the tiles and patterns involved. My earlier paper ‘Explore the Design Space’ [7] covers many developments of this family of games, so I will only give one example here.

The tile-based puzzle game Gloop by Fred Horn [27], shown in Figure 17, uses a set of 91 square tiles with two vertices per side, that represent all unique combinations of non-overlapping paths between different vertices. The game consists of a number of specific challenges for players, but it is also a rewarding exercise to just make free-form shapes with the tiles, as shown.

3.3.5 Board Geometry

The mutation from square to hexagonal bases for playing grids (Figure 18) occurred last century in two main contexts for board games: in connection games and tabletop war games. The prevalence of hexagonal grids in connection games stems not only from the influence of the great game Hex from the 1940s, but because the trivalent nature of the hexagonal grid eliminates potential problems with deadlocks in many games [3].

The modern move from square to hexagonal grids in tabletop war games is often attributed to designer Charles S. Roberts, who pioneered the use of the hexagonal grid in many of his designs in the 1960s [14]. The hexagonal grid eliminates corner cases (there are no corner-sharing ‘diagonal’ neighbours) and the difference in coordinates between two cells more closely matches their Euclidean distance than on the square grid, which is important for war game simulations.

The hexagonal grid is of course used in other types of games for other purposes as well. Historical examples include Chinese Checkers from 1892 and Agon (or Queen’s Guard) from the 18th century [16]. Modern examples include Yavalath (Figure 2), which works well with three axes along which lines can form, whereas the two axes of a square grid are not enough and the four axes of a square grid with diagonals are too many.

3.4 Crossover of Rules

The crossover of rules involves the collision of ideas, or more precisely the reuse of rules in conjunction with rules from other contexts. Crossover is typically harder to achieve successfully than mutation and therefore rarer, as two sets of colliding assumptions and constraints must merge harmoniously. The following examples are hence presented on a case-by-case basis.

3.4.1 Gonnect

Gonnect, designed by João Pedro Neto in 2000 [28], combines the rules of Go with the rules of Hex. Players place stones and perform surround captures as per Go, but win by connecting their sides of the board with a chain of their pieces. It is not only my favourite game crossover, but is one of my favourite games of any type.

In order to avoid deadlocks on the square grid, as shown in Figure 19, the base rules of Go are actually simplified by disallowing pass moves. The next player to move in this example will be forced to fill in one of their eyes, allowing the opponent to capture that group and go on to win the game.

14https://www.boardgamegeek.com/thread/1762462/hexes-wargames-history
Gonnect also greatly simplifies the Go winning condition from a rather complex scoring system to an intuitive connective goal. The result is a beautifully elegant but deep game.

### 3.4.2 Guerrilla Checkers

Similarly, Guerrilla Checkers, designed by Brian R. Train in 2010, is a combination of the rules of Go and Draughts. One player starts with six Draughts pieces on the board that move and capture as Draughts pieces (i.e. by hopping over enemy stones), the other player starts with 66 Go stones which are added to the board and capture as Go stones (i.e. by surrounding enemy stones). Players win by eliminating all enemy pieces.

The rules of the two base games have been modified as follows for balance, but Guerrilla Checkers retains the spirit of both:

- Number of pieces.
- Starting position of Draughts pieces.
- Two Go stones placed per turn.

The Draughts-like goal – eliminate all enemy pieces – is different to that of Go, but consistent with the overall aim of capture and dominance.

### 3.4.3 Racing Kings

Racing Kings, invented by Vernon R. Parton in 1961 [11 pp. 98–99], is a Chess variant in which the pieces start as shown in Figure 20 and a player wins by being the first to reach the far side of the board with their king. It is not allowed to put or move either king into check at any time.

Racing Kings is included in this category, Crossover of Rules, as it is more than a simple rule change applied to a standard Chess set. According to Parlett’s classification of board games [16], Chess belongs to the Displace category while Racing Kings would belong to the Race category. Racing Kings places known pieces from one class of game, with their movement rules intact, into a totally different category of game.

### 3.4.4 Hive

Hive, shown in Figure 21, is a popular tile-based game by John Yianni [29]. Players place hexagonal tiles showing insect figures, and move them in ways reminiscent of how the insect shown would move. The relevance of Hive to this discussion is that it combines two types of tile-based games: it starts as a tile placement game, then progresses as a combination of tile placement and tile movement game.
As with Tortuga (Section 3.2), the zoomorphic aspect of the pieces adds a certain charm to the game, which is reinforced by the natural and intuitive nature of their movements. The hexagonal basis, reminiscent of the internal architecture of bee hives, also fits the insect theme nicely. The design elements all click in Hive, making it more popular than many deeper and more strategically sophisticated games.

3.4.5 Kakuro

The crossover of rules is also prevalent in the design of solitaire logic puzzles. For example, the Japanese logic puzzle Kakuro [30], shown in Figure 22, is essentially a numerical Crossword with the no-repetition constraint from Sudoku.

![Figure 22. A Kakuro challenge.](https://example.com/kakuro.png)

The ‘clues’ here are the hint numbers shown for each horizontal and vertical run of spaces. Players must deduce which digit 1 to 9 goes in each space, such that no digit is repeated in a run, and each run’s digits add to the hint value shown.

3.4.6 War Game Mashups

There exists a subculture of tactical wargamers who enjoy ‘mashing up’ elements from different tactical tabletop wargames, e.g. playing with Band of Brothers rules and units on a Combat Commander map, or converting scenarios from one system entirely to another system [16]. This very explicit case of crossover lies somewhere between Crossover of Rules and Crossover of Equipment as it can involve aspects of either or both.

Dread is often praised for its innovation and the amount of suspense that this novel reuse of an imported mechanism achieves. It won the 2006 Ennie Award for Innovation, and was also nominated in the Best Game and Best Rules categories.

3.5 Crossover of Equipment

Crossover of equipment involves the reuse of equipment, and the rules they imply, in other contexts. This category is the most limited and difficult to achieve, as finding two distinct sets of equipment and their rules that combine harmoniously is rare.

3.5.1 Dread

As a game that transcends genres to successfully reuse equipment, it is hard to think of a clearer example than Dread [31]. Dread is a system of adult horror role-playing games in which the success of player actions is decided not by the usual dice throws, but by making a move in a sub-game such as the children’s dexterity game Jenga [32]; if the player loses the subgame then their in-game character dies. Figure 23 shows a player deciding the fate of his character.

![Figure 23. A Dread player hard at battle.](https://example.com/dread.png)

Dread is often praised for its innovation and the amount of suspense that this novel reuse of an imported mechanism achieves. It won the 2006 Ennie Award for Innovation, and was also nominated in the Best Game and Best Rules categories.
3.5.2 For the Crown

For the Crown, shown in Figure 24, is a deck-building card game played using a chessboard. The cards show Chess-like moves that can be applied to special units on the board.

![Figure 24. A hand of For the Crown cards.](image)

This example shows crossover of some equipment from Chess – the board – but is more a crossover of genres.

3.5.3 Triple Tournaments

A very literal way to combine distinct games, with their equipment and rules intact, is suggested by Schmittberger [11, pp. 39–40]. Players set up any three games of their choice, and each turn must choose which of the three games to make their move in, with the aim of winning at least two of the games overall. This is a simple and effective way of adding complexity and tension to existing games without modifying them in any way. Schmittberger suggests the following groupings:

- Chess/Checkers/Chinese Checkers.
- Backgammon/Parcheesi/Monopoly.

4 Conclusion

These examples demonstrate the prevalence of the reinvention of fundamental mechanisms and equipment in a range of games. With respect to modes of reinvention, mutation is easier to apply than crossover (only one set of assumptions and constraints needs to be taken into account) and rules are easier to modify than equipment. This is borne out in the above survey of games, as there is no shortage of examples for the Mutation of Rules (Section 3.1), but many fewer examples of Crossover of Equipment (Section 3.5).

Note that only successful mutations and crossovers are shown above. The number of failures will be many times greater, but still interesting parts of the design space to explore; it is hard to tell whether a given reinvention will work – or at least provide some unexpected benefit – until it is tried. However, an experienced designer will have an intuition for what is likely to work or not work, and which degrees of freedom will be most promising to explore.

Acknowledgements

Thanks to Stephen Tavener, Russ Williams, Richard Reilly, John Farrell and Néstor Romeral Andrés for discussions on this topic. This work was supported by a QUT Vice-Chancellor’s Research Fellowship.

Many thanks to the Ita Hart Trust for permission to reprint the B.C. comic strip by Johnny Hart shown in Figure 1. Also thanks to the following for image permissions: BoardGameGeek.com user GeoMan for the Tortuga image (Figure 6); Carthoris Pyramidos for the Chase image (Figure 10); Jeremy Kalgreen for his photo of Risk territory cards (Figure 11); Jason Allen for his image of Bohnanza cards (Figure 12); Andrew Brooks for his photo of the Patchwork scoring track (Figure 9); BoardGameGeek.com reviewer Enders-Game for the San Juan (Figure 13) image; Charles King for his Tak image (Figure 15); Matthew Marquand for his Hive image (Figure 21); Tomer for the Dread image (Figure 23); and Dušan Vit for the For the Crown image (Figure 24).

References


Cameron Browne is a Vice-Chancellor’s Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.

Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.

Email: c.browne@qut.edu.au
Creating suitable abstract games for three players is a hard design problem. When one player gains a lead, the other players tend to cooperate until this lead is pegged back and a new balance is achieved. This article explores the dangers of this behaviour, some simple ways to minimise it in games, and provides examples of abstract board games that exploit the proposed design strategies.

1 Introduction

Among family and friends, it is common to make a group around a table to play board games. In this age of e-mail and online servers of various sorts, it is becoming increasingly easier and more common for many players to join to play games via electronic media. But considering abstract games, most are for two players. Some, especially chess variants, are extended to four players. Very few have good three-player versions. Why is that?

Moving from two-player games to multi-player games creates a social environment which allows alliances, threats, betrayal, and a raft of group behaviours that go well beyond the mere dictates of rules. Some games, which would have no interest with just two players, may flourish with several players; however, the opposite is also true for some games.

1.1 Petty Diplomacy

We are referring to games in which players interact with each other, whether it be directly or through the pieces on the board. This excludes games like Snakes & Ladders where pieces share a board but do not interact. In these games, the main problem is the management of alliances. Multi-player scenarios are subject to a phenomenon that can ruin many promising games, called the petty diplomacy problem (PDP) [1]. In its form known as the ‘tall poppy effect’, as soon as a player gains a lead, the other players cooperate to cut them down, then shift alliances when a new leader develops.

One way to address PDP in multi-player games is to design the game with an even number of players – typically four – and assign predesignated teams (e.g. many card games and four-player Chess). Another is to increase the number of players – typically to an odd number such as five or seven – and allow multiple alliances to occur (e.g. Risk). But the problem is exacerbated for three-player games.

1.2 The Kingmaker Effect

Another problem with multi-player games is the kingmaker effect, in which a player about to lose is able to dictate which opponent will win the game. If the number of players is three, then defeating one player means granting victory to the other. So in a three-player game, kingmaking is a powerful bargaining tool. Also, with such a small set of players, the suicide strategy may not be entirely effective, and the attacked player may become so weak as to attempt to inflict the suicide/murder pact on another player. The cycle continues and the game dynamic is compromised.

1.3 Revenge Rules

Multi-player games are also subject to problems of spite and revenge. Straffin, in his article on a three-player version of Hex [2], states the McCarthy Revenge Rule to handle such cases: If I am about to lose, I will inflict as much damage as possible on the player who put me in this position. This balancing mechanism is widely used and has a rough justice about it, and may be relevant to simulation board games such as Diplomacy and Monopoly.

However, it is difficult to enforce in practice, and sometimes both opponents can be equally to blame for a player’s disastrous position. Further, Browne demonstrates by example that actually enforcing the opposite rule – hurt the player who has hurt you the least – can actually work much better in reducing non-strategic coalition problems in some cases [11].

We feel strongly that such off-the-board bargaining should have no place in abstract board games. Friendly reminders from one player to another, that the third is about to become unsailable unless a certain action is taken, are probably unavoidable. But the open use of threats and revenge to induce such action is unpalatable.
1.4 Possible Solutions

We have thought about this subject and tried several ways to inhibit PDP with augmented rule sets, keeping in mind that it may be impossible to eliminate it completely. It is possible that complete elimination of PDP may not be achieved and not even desirable (e.g. pure race games do not have PDP but tend to be intellectually boring games). So PDP may well add new layers and worthwhile complexity to the play. A good rule of thumb seems to be ‘tall poppy good, threats and revenge bad’.

In other words, the ability to form micro-alliances (a PDP effect) is not only a side effect of three-player games; it may well be crucial to their success. Some designers such as R. Wayne Schmittberger [1] go to great pains to reduce micro-alliances in games. But perhaps this is not a good thing, and such micro-alliances should be encouraged! But it is a thin line. Too much PDP and the game might cycle unless it has a specific length. If not, as in many games, it may never end or just produce endless stalemates, as can occur in the game Diplomacy.

The basic idea that we adopt is that no player may allow the next player to make an immediate win on their next move, unless no other option is available. This does not, of course, prevent the player from making a more subtle move that virtually guarantees one of the others will win later on, but does at least remove the grosser forms of PDP and spitefulness. This rule, which we call the Stop-Next rule, is discussed by Duncan in his article ‘Mitigating Non-Strategic Coalitions’ published in the previous issue of Game & Puzzle Design [4]. We examine its application here to a different and broader range of games.

The idea of an immediate win (that is, the fact that the next turn may be the last one legally played) is more clearcut in some games than others. For some games, we have found it wise to extend this rule a little (e.g. if the following series of moves is virtually forced) or even reduce it a little (e.g. if it is hard to tell at a glance that a won position is indeed won). But usually there is no problem. We assume this rule is in place in all of our following games, with further clarification provided where needed.

We adopt the following definitions:

- **Mover**: The player currently moving.
- **Next**: The next player to play.
- **Other**: The remaining player.

2 Shared Pieces

One way to reduce PDP is to use shared pieces. Here, in a sense, everyone is helping everyone else, even in the two-vs-one scenarios. Some examples follow.

2.1 Triskelion

The game of Triskelion, by Bill Taylor, uses two types of stones for three players in a hexagonal connection game [5 p. 165]. The board is shown in Figure 1.

Three players (White, Black and Grey) share 30 white and 30 black stones. Each player owns the opposite pair of sides of their colour. White starts by placing a black stone on an empty cell. Thereafter, players take turns placing a stone – of a different colour to the last stone played – on an empty cell, i.e. stone colours alternate.

Players win by connecting their sides with connected chain of pieces of a single colour. If two such connections are made, then the Mover wins, otherwise Next wins. If a Y pattern is made connecting three non-adjacent edges with one colour, then the Mover wins.

Notice how the number of piece types (two) and number of players (three) are out of phase every second turn. This means that a player who
plays a black stone at turn \( n \) will only play another black stone at turn \( n + 2 \). Thus, the players must constantly focus on chains of different colours. This simple method of alternating both three players and two piece colours seems remarkably effective and interesting for three-person play. When playing Triskelion, it is common to reach interesting end-games, full of tension and desperate tactics, that produce the winner only when the board is almost full.

Most stones played provide potential paths to victory for the other players as well. Due to the possibility of making paths of both colours, it is rare for a player to be in a totally lost position. Using PDP is only possible very near the end-game, and is limited by the fact that each player must play a specific colour on the board. Therefore, players can create positions in which neither opponent can attack (if either colour will complete their connection), but in trying to prevent this win both opponents are still contributing towards the player’s victory.

For example, Figure 2 shows a typical Triskelion position, with Black to play a black stone. Grey has two realistic paths to victory; an upper path with white stones and a middle path with black stones. White can only win with black stones and shares some of that path with Grey. Black must now play a black stone, which is useless to them but may be helpful to their adversaries, giving them time to prevent Black’s vertical white path. Grey can therefore profit from both opponents’ efforts.

### 2.2 Three-Player Iqishiqi

Another game with shared pieces is the three-player version of Iqishiqi, designed by João Pedro Neto [5, p. 89, 167], played on the board shown in Figure 3.

![Figure 3. Iqishiqi board.](image)

**Three-Player Iqishiqi**

The three players are Black, Grey and White. The game starts with a black stone in the centre.

Players take turns placing a white stone on an empty cell from where it must push the black stone. The black stone is only pushed if the dropped white piece belongs to a group where at least one of its pieces (including the dropped piece) is in the clear line-of-sight from the black stone, i.e. there are no stones between it and the black stone. The black stone is pushed away from the group a number of steps equal to the group size. Pushes off the board are not allowed. If the group allows for more than one direction to push the black stone, the player can choose which direction.

If the black stone reaches an edge, the owner of that edge wins. Mover wins if the stone reaches a corner owned by them, but if the stone reaches another corner then Next wins. Mover wins if Next cannot make a legal move on their turn.

Note that in Three-Player Iqishiqi, the Stop-Next rule should be applied only to reaching an edge, not to making a stalemating move. The game is like building a maze of shared stones.
while trying to move the black stone to a dead end or making the adversaries push the black stone near one of your edges, where you might score a ‘goal’ by landing it right on the edge.

For example, Figure 4 shows a typical game after 23 moves, in which the black stone has been shuttled around the board, which happens when the local density of pieces reaches a certain threshold. Move 23, which formed a group of size two, pushed the black stone two cells down.

Figure 4. A typical game after 23 moves...

Figure 5. ...and after 29 moves.

...and after 29 moves. White can now win by playing a stone at the cell marked ‘x’, and pushing the black stone one step to a White edge. Note that Grey’s move 29 must have been forced, otherwise it would have been illegal due to the Stop-Next rule.

3 Unequal Tasks

One of the intriguing things about three-player games is that it is not necessarily a bad thing if the players have somewhat different tasks, perhaps of differing difficulty. Though this would usually be fatal in a two-player game, the tall poppy effect ensures that with three, the easier tasks will not automatically lead to easier (or any) wins. The following examples demonstrate this.

3.1 Porous Torus

Three-player Porous Torus is a game adapted by Bill Taylor from one of his two-player inventions. Here the players have tasks differing noticeably in difficulty, but not so much that it might make it particularly easy for any of them to maintain the upper hand for long.

Porous Torus is played on a rectangular array of hexagons representing a torus, as shown in Figure 6 with normal local hexagonal connectivity. The top and bottom rows are considered adjacent, as are the leftmost and rightmost columns.

It is a fascinating fact that on a torus there are three kinds of ‘global’ loop that can be made, that is, a loop going ‘right around’ the torus and not smoothly contractible to a point. Even better, the three types are mutually exclusive and exhaustive, meaning that one of the three must eventually be made, and it will not be possible to make two in different colours. This property, which is called \textit{win-loss complementarity}, is a key property of good connection games (as seen in Hex and Y). The three types of global loop, from easiest to hardest, are vertical, horizontal and spiral, as shown in Figures 7, 8 and 9.

Figure 6. Porous Torus board.
Three players (Diagonal, Horizontal and Vertical) take turns placing a stone in an empty cell. The first stone played is black, then stone colours must alternate, as per Triskelion (see Section 2.1). The Stop-Next rule applies.

If a global loop is formed in a single colour, then Diagonal wins if the loop is diagonal (i.e. a spiral), Horizontal wins if the loop connects left-right, and Vertical wins if the loop connects top-bottom. If two types of loop are made simultaneously, then the Mover wins if they own one of the loops, otherwise Next wins.

Figure 7. A vertical loop.

Figure 8. A horizontal loop.

Figure 9. A diagonal loop (spiral).

It is still possible that loops of two types in the same colour could be made with a single move. In such cases, the Mover wins if they own one of the loop types, else Next wins.

Figure 10 shows a game about to be won by Diagonal. A black spiral will be formed if the two cells marked ‘1’ are filled with black stones, and a white spiral will be formed if the two cells marked ‘2’ are filled with white stones. The other players, even acting in concert, cannot stop all of these connections.

Figure 10. A winning position for Diagonal.

4 Freezing Players

In certain types of abstract games, namely connection games, the position may be such that one or more players are left with no winning options. In these situations, their only role is to be king-makers. One way to prevent this is to insert a mechanism of player freezing. Those players that are unable to win stop playing altogether. The decision to remove stones or leave them on the board is an important one, and must be considered on a game-by-game basis. The following section shows a working example of this mechanism.

4.1 Three-Player Hex

It is not easy to extend the classic board game Hex [6] to a hexagonally shaped (or hexhex) board, as shown in Figure 11, as such boards allow Y connections that can result in unwinnable deadlocks. But this problem with potential deadlocks can be solved by simply removing any player who can no longer make a winning connection. The resulting game of Three-Player Hex by Philip Straffin [2] is described below.
Three-Player Hex

Three players (Grey, White and Black) each own the opposite board sides of their colour.

Players take turns placing a stone of their colour on an empty cell. A player wins by connecting their board sides with a connected chain of their pieces.

Any player who can no longer possibly make a winning connection is removed from the game, but their pieces remain on the board. A player also wins if both opponents are removed.

For example, Figure 12 shows a game won by White, as neither Black nor Grey can make any possible winning connection so have been removed from the game. Grey was eliminated by Black on move 33, which permanently blocked the grey sides from being connected, then Black was eliminated by White on move 34, which permanently blocked the black sides from being connected. White wins, Black comes second and Grey comes last.

Three-Player Hex is a mixture of connection game and blockage game. Players tend to concentrate on leaving as many as possible open paths to victory instead of racing to connect, as this would cause a sequence of blocking replies from the others, and one player cannot force a way through two others in concert. The enjoyable twist of Three-Player Hex is the use of deadlocks as an essential part of the game dynamics, with the player removals.

5 Variable Turn Order

Sometimes players in three-player games may feel annoyance at the fixed order of moves, in that they may feel that one is a bad player. So, it is an advantage to be following that player in the move order, to explore their weak moves. One possible solution is to allow the Mover to decide who plays Next. However, this could have the perverse effect of letting two players dictate that each other moves next, effectively starving the other one.

One remedy is to let players grow in strength with each turn that they are neither the nominator nor the nominee. For instance, if A nominates B to move next, then C gets to place an extra piece on the board after B’s move. This strikes a balance between the benefits of moving (mobility) and the benefits of not moving (additional resources). The nomination should not be completely arbitrary, of course, but should be dependent on the kind of move made.

5.1 Triad

With these ideas in mind, Cameron Browne designed the game of Triad [5, pp. 175–176], which is played on the tri-coloured hexhex board shown in Figure 13. This game clearly demonstrates the PDP phenomenon, in the way that micro-alliances form and evaporate between competing players with each passing turn due to the tall poppy effect, but allows clever players to control play through a forced move mechanism.
Triad

Three players (Red, Green and Blue) start with nine pieces each of their colour positioned as shown. Red moves first, then players take turns moving in three parts as follows.

1. The Mover must move a piece of their colour in a straight line along any of the six hexagonal directions to land on an empty foreign cell, i.e. a cell of a different colour. All intervening cells must also be empty. The opponent who owns the landing cell becomes the Candidate, and the other opponent becomes the Bunny.

2. All opponents’ pieces of either colour immediately adjacent to the landing cell are captured and removed from the board. The Mover must make the move that captures the most pieces but may choose amongst equals (Max-Capture rule).

3. The Mover must then add a Bunny piece to any empty cell, unless the Bunny has just been eliminated. The Candidate becomes the Next player to move.

Play stops as soon as any player is eliminated. The game is won by the player with the most pieces remaining on the board, otherwise is a tie between the two remaining players if both have the same number of pieces remaining.

The Max-Capture rule allows an interesting twist on this theme. A clever player can manipulate an opponent into performing certain captures and then returning control of the move, or even eliminating the third player, by judicious placement of the Bunny piece.

For example, Figure 14 shows a typical board position with Red to play. Say that Red makes the move indicated to the green cell, which captures the two green pieces marked ‘X’ and delegates control of play to Green, and adds a blue (Bunny) piece at the cell marked ‘+’. This gives the position shown in Figure 15.

Green is now forced to make the move indicated, as this captures the most pieces (three). This move kills three blue stones and returns control of play to Red, who then has several move options to choose from, regardless of where Green places the Bunny piece, thus Red retains control of the game.

6 Other Extensions to Three

This section presents abstract board games that work nicely in a three-player setting. Most of these examples use the Stop-Next rule, showing how this is a meta-rule that can mutate two player games into workable multi-player variants.
6.1 Three-Player Reversi

Reversi (aka Othello) can be successfully adapted to three players on a hexhex board. We suggest the initial setup shown in Figure 16.

![Figure 16. Initial setup for Reversi for three.](image)

The Stop-Next rule is not used in this game, but it is advisable to play with an additional rule: no player can eliminate either opponent (unless this is unavoidable). This prevents two players cooperating to single out the third and remove them from the game. A player with no legal move merely passes, as in normal Reversi.

6.2 Three-Player Gomoku

Figure 17 shows a three-player version of the classic game Gomoku played on an $11 \times 11$ board.

![Figure 17. White exploits Stop-Next to win.](image)

The Stop-Next rule makes the three-player version of this game problematic, but has both enabling and restrictive good effects when used. A hindering effect is that it becomes very dangerous for a player to make a single one-ended line of three. Next will ignore it and make their own open-ended line of three if possible, forcing Other to block the original Mover’s threat. This leaves the Mover to try to block both ends of Next’s open three, as shown in Figure 17. A player can often use Stop-Next to keep the Next – and maybe both – opponent busy, while they develop their own position.

Figures 18, 19 and 20 show the aggressive nature of this game due to the Stop-Next rule. For example, Figure 18 shows forced moves by White and Grey from move 10 onwards, due to Stop-Next. After this forced sequence, White should play move 20 to prevent a double open three that would give the game to Grey, shown in Figure 19.

![Figure 18. Stop-Next creates many forced moves.](image)
From this position, Grey must play move 27 as shown in Figure 20 due to the Stop-Next rule, to avoid an immediate win to Black. Grey move 30 forces White to reply 32, allowing Black to make the killer 31. This triggers a sequence of forced moves 32 .. 39 that allows Black to apply the coup de grâce with move 40, which sets up an unbeatable triple threat (cells marked ‘x’).

6.3 Three-Player Gonnect

The board game Gonnect, designed by the first author, is a mixture of the territorial play of Go with a connection goal. The following box summarises the rules for the standard two-player game.

**Gonnect**

Two players (Black and White) take turns placing a piece of their colour on an empty intersection.

The rules of Go apply (surrounded enemy groups with no freedom are captured and removed; no previous board position can be repeated; suicide is not allowed) expect that players cannot pass.

Pie Rule: Black places a stone and then White may elect to swap colours instead of moving.

Players win by forming a chain of orthogonally adjacent stones of their colour connecting two opposite board edges, either left-right or top-bottom. A player with no legal moves loses.

The three-player version of Gonnect applies the same territorial and connective concerns in a multi-player scenario. This requires further clarification of the surround capture rule: all stones without any liberty are captured, i.e. the Mover may capture pieces from both opponents with a single move. For example, a White move 1 shown in Figure 21 would capture the black and two grey stones.

To inhibit PDP, Three-Player Gonnect is played with an additional Mandatory Prioritised Capture (MPC) rule, which states that it is mandatory to capture Next player stones if possible, else to capture Other player stones if possible, else to move freely.

Figure 23 shows a game of Three-Player Gonnect played on a 13×13 board that ended in a remarkable sequence of forced moves. Black
can still win with a horizontal connection while Grey threatens a vertical connection, but White’s position is hopeless. The move sequence is Grey ⇒ White ⇒ Black, with Grey to move.

**Figure 22.** A Three-Player Gonnnect position.

Move 3 captures stone 2 and move 6 captures stone 5. White moves 2 and 5 appear futile, but forced the Next player to capture them, allowing Grey to produce an unusual sequence of forced captures due to the MPC rule.

**Figure 23.** A sequence of forced captures.

Play continues as shown in Figure 24. Grey extends their main group with move 1, then White move 2 forces Black to immediately capture it with move 3 due to MPC. Grey then plays move 4 at the point vacated by stone 2 to capture grey stone 3, and White is forced to make move 5 to capture the black stone according to MPC. Grey has exploited the MPC rule to achieve a winning position.

**Figure 24.** Grey forces a winning position.

7 Conclusion

This article shows possible ways to extend two-player abstract board games into the difficult design realm of three-player games. A fruitful mechanism is to restrict the immediate moves of the next player to gain some control of other player’s actions. The article explores several ways to implement the Stop-Next rule in the context of different game types (connection games, territory games, etc.). The examples presented here are all successful implementations of Stop-Next, and it is our hope that they help the reader to design new three-player games.

Some games, of course, are designed specifically for three players and are virtually impossible to extend further. Triskelion and Triad are of this type, as is Hex. Some seem to have no hindrance at all in extending almost without limit, providing only that the board is big enough, like Gomoku, Gonnnect, and Reversi. Then there are some that may extend only a little; Porous Torus can extend to four players and Iqishiqi to six. However, we leave it to a further generation of gamers to investigate these matters more fully.

The games presented in this article, and many others, can be extended to four or more players with some success using the same techniques. Many commercial games are specifically designed for groups of four people or more. We believe that abstract board games, such as those shown above, can also have a place among such groups of players.
Acknowledgements

João Pedro Neto is a BioISI member and has the support of UID/Multi/04046/2013 strategic project. Thanks to the Editor-in-Chief for extensive edits to this piece, and for redoing the figures in colour. This article is based on a book chapter of the same title by J. Neto, C. Browne and W. Taylor in *Mathematical Games, Abstract Games*. [23]

References


João Pedro Neto is Professor of Computer Science at the University of Lisbon. He is interested in the study, design, collection, publication and play of abstract games, and helps organise a yearly national event at Portugal for the dissemination of new abstract games between young students. **Address**: University of Lisboa, BioISI – Biosystems & Integrative Sciences Institute, Lisboa, Portugal. **Email**: jpmneto@fc.ul.pt

Bill Taylor is a former Lecturer in Mathematics at the University of Canterbury, New Zealand. With research interests in probability theory and discrete mathematics, he has given seminars on abstract games at Board Games Studies and other venues. **Address**: University of Canterbury, Christchurch, New Zealand. **Email**: W.Taylor@math.canterbury.ac.nz

Gadeiro Challenge #10

Pack the pieces on the right to fill the shape on the left. Gadeiro is described on pages 39–41.
Some Random Thoughts on Chance and Skill

David Parlett, Gourmet Games

The roles of chance and skill in strategy games are often confused. I argue that chance is not the same as luck nor skill the same as intelligence, and explore some ideas along these lines. This discussion is an updated version of a paper presented at the 2008 Board Game Studies Colloquium held in Lisbon.

1 Introduction

Like Gaul, games are anciently and popularly divided into three parts: games of skill such as Chess and Go, games of chance such as Snakes & Ladders and Roulette, and games of mixed chance and skill such as Backgammon and Bridge. Such categorisation is patently inadequate. It is slightly more adequate to demolish the divisions and regard chance and skill as polar opposites of a single continuum, so that any given game – or any given instance of one – may be regarded as involving X percent skill and (100 – X) percent chance.

But then skill and chance are themselves inadequate terms. Games involve many different forms of chance, some of which are perceived rather than real. A more appropriate term for this end of the spectrum is uncertainty, or unpredictability as to the outcome of a game. All games by definition involve a degree of uncertainty, for if the outcome of a game were ever entirely certain or predictable there would be no point in playing it.

At the opposite end of the spectrum lies the antidote or counter to uncertainty, which is the degree, if any, to which you can control or at least influence the outcome of a game. The opposite of uncertainty is better characterised as controllability rather than skill, as skill itself is not an atomic property: there is no such thing as a single, universal ‘skill at games’ but rather many different types of skill. People tend to play those games for which their particular talents suit them, or, if their talent is not one of controllability, to which they are most attuned by temperament.

I am interested in exploring the elements of uncertainty or types of chance that may be encountered in games, and the corresponding elements of skill or types of controllability that may be employed to counter them. This exploration takes me into specialised disciplines, such as mathematics, psychology, and pedagogy, in which I have absolutely no qualifications or expertise. I write purely as a games enthusiast and inventor, and can only hope that my comments might be found to have some bearing on: a) the classification of games; b) inventing games; and c) games appreciation.

2 Uncertainty (Chance)

2.1 Randomisation

The primary, most fundamental and oldest embodiment of uncertainty is the occurrence of randomising events such as the cast of lots or dice, for which reason many suppose games to have originated in the practice of divination. Equally fundamental, but historically more recent, is the randomisation of an initial position, which is classically embodied in the dealing of playing-cards from a shuffled pack.

2.2 Compulsion (vs Choice)

An element of uncertainty occurs in randomising games that give you no choice of play. A classic example is the Indian ancestor of our Snakes (or Chutes) & Ladders. Why such a game should continue to exist is well explored by Salen & Zimmerman in Rules of Play [2, p. 179]. Briefly, it may be ascribed to the quality of drama it displays, that is, the possibility for a player to recover from a weaker position. I would paraphrase their argument by suggesting that Snakes & Ladders may be regarded as an overlap between, on one hand, the playing of games, and, on the other, the performance of plays, a point which I think would have appealed to historian Johan Huizinga. Compulsion also overlaps with divination, in that it is an essential property of Fate. The opposite of compulsion is choice, or free will, which provides an essential opportunity for the exercise of skill.

1Huizinga is the author of the famous treatise Homo Ludens: A Study of the Play-Element in Culture [3].
2.3 Information: (Im)perfect, (In)complete, (Un)certain

Careless commentators tend to lump randomisation together with imperfect information, but in fact there is a significant difference between the types of informational imperfection. Let us take Backgammon and Bridge as exemplars:

- Backgammon starts from a predetermined opening array and all subsequent moves are visible to both players. To this extent (only) the game is one of perfect information at least as to the present and the past. The element of randomness is introduced by the unpredictable roll of dice, so the type of uncertainty it involves may be characterised as ‘future imperfect information’.

- Bridge, on the other hand, starts from a randomised opening array. Thereafter, however, it is entirely free from uncontrollable randomising eventualities. It is, therefore, a game of ‘past imperfect information’.

The differences are significant in that they call for distinctive skills in order to exert some degree of control over their outcome.

- In Backgammon, skill consists in gradually setting up positions in which you can benefit from a greater proportion of possible future casts than your opponent, whom you try to manoeuvre into such a position that very few possible casts are favourable.

- In Bridge, skill consists in deducing or inferring the lie of cards in other players’ hands, initially by means of the auction, and subsequently by playing your cards in such a way as to uncover existing information in time to take advantage of it. Bridge and other intelligent card games are therefore not so much games of imperfect information as games of perfecting information.

Further distinctions may be drawn between perfect and imperfect information, certain and uncertain information, and complete and incomplete information; but let this suffice for the moment.

2.4 Inequality and Indeterminacy

Inequality is associated with a random and therefore indeterminate opening array such as the initial distribution of cards. Its effect is reduced by sufficiently increasing the number of deals that constitute a whole game. It is also inherent in asymmetric board games like Hnefatafl and Fox & Geese, where the two players have different forces and different objectives; but here, too, it is easily overcome by playing an even number of games and alternating the positions. You might say that it is a feature of most combinatorial games, at least in so far as it is usually advantageous to move first.

Here it might be relevant to note that one of the ways in which games evolve is by deliberate alteration of the rules in order to reduce potential inequalities of players’ experience. If you subscribe to the online newsgroup rec.games.abstract you will be aware of a current interest in devising forms of Chess with indeterminate openings, such as Fischer Chess (aka Chess960). In these variants, the opening array is unknown in advance, but is fully open before play begins so there is no lack of information. What is lacking here is an experience of playing with a particular configuration of pieces. On these grounds one might propose ‘novelty’ as a carrier of uncertainty.

2.5 Opacity vs Clarity

An element of uncertainty, or at least uncontrollability, is induced by the opacity of a game. By opacity I mean the opposite of clarity, a property I think Robert Abbott was the first to point out in a 1976 edition of Games & Puzzles magazine. In an article entitled ‘Under the Strategy Tree’ Abbott writes:

Clarity is essentially the ease with which a player can see what is going on in a game... [It] has nothing to do with simplicity, or even with elegance. Edward de Bono’s L-Game is elegantly minimal – it uses only four pieces and is played on a board of only $4 \times 4$ squares. It is not, however, clear. I find it very hard to picture what the board will look like when I turn my ‘L’ over, I find it harder still to visualise my opponent’s responses, and it’s impossible for me to look ahead to my next move. A game can be simple yet lack clarity, and conversely a game can be complicated but still clear. Playing a game soon reveals its degree of clarity. The greater the clarity of a game, the further you can see into it, and therefore the greater its depth for you.

2.6 Perception + Complexity = Perplexity

Chance and skill may differ according to whether they are perceived from the inside, subjectively
by a player, or from the outside, objectively by an observer. As Salen and Zimmerman put it, ‘the feeling of randomness is more important than randomness itself’ [2, p. 3]. They quote the example of Chinese Checkers with four or more players:

As the game unfolds..., the centre... becomes crowded with a seemingly random arrangement of pieces... even though every single move... is the result of a player making a strategic choice about where to play next. If you closed your eyes and opened them only when it is your turn to move, it might seem like the board is merely reshuffling itself, particularly in the middle... game, when the centre area is most crowded. This feeling of randomness is only an illusion, however, as there is no formal chance mechanism in the game. [Hypothetically] logical players... wouldn’t feel any randomness: they could look at the board and immediately trace every move back to a series of strategic decisions. However, for human players, this feeling of randomness is an important part of what makes the game fun to play... [T]he feeling of randomness creates a sense of open-ended possibility and players are rewarded for taking advantage of chance configurations on the board... Seeing a pattern emerge out the chaos that allows you to jump a piece back and forth all the way across the entire length of the game board is a moment of wonderfully meaningful play. [2, pp. 2–3]

Another subjective perception of chance comes into play when a Chess master plays a novice. Both have perfect information as to the moves and changing positions, but whatever the weaker player does, the stronger sees through it to the motivation of that move, to what their opponent has in mind, what situation it might ultimately lead to, and how to circumvent or take advantage of it. The weaker, however, will often be baffled by the master’s unforeseen move with what are apparently unforeseeable implications. As far as a novice is concerned, they might as well be confronting a completely random move determined by the roll of a die, and exhibiting no perceptible past cause or future effect. As Chess expert and scientist Philip Ross puts it:

The feats of chess masters have long been ascribed to nearly magical mental powers. This magic shines brightest in the so-called blindfold games in which the players are not allowed to see the board... [6]

Ross’s reference to magic reminds me of the converse of perceived but unreal chance, namely that of perceived but unreal skill. An example is provided by ethnographer Dennis Tedlock in his Introduction to Stuart Culin’s Games of the North American Indians:

We might call it a ‘game of chance’, which is what Culin calls similar games in this book, but that expresses the point of view of an observer. Meanwhile, the participants constantly think in terms of strategy, pitting their wishes against chance in momentary acts of magic, which is what we all find ourselves doing when we throw dice. [A] paradox of the Zuñi game of wooden dice is that, technically, it is not what Culin calls a ‘game of dexterity’, and yet the players do try slightly different ways of handling the sticks, as if they could influence the outcome of a throw. [...] So if there is any dexterity here, it must remain on the side of magic [...] [7, p. 23]

We might mention, for completeness, the existence of chance factors that are extrinsic to the game. I first became aware of these in my school-days, when my friend’s mother raised some objection to our playing Chess on a Sunday. ‘It’s a game of skill’, we protested, ‘not a game of luck’. ‘Well’, she replied, ‘You’re lucky if you win, aren’t you?’.

It was at this point that I realised that: a) luck and chance are not the same thing; and b) you are indeed lucky if in a Chess tournament you happen to be drawn against a weaker player, or one whose opening you have just been mugging up, or one who happens to be feeling unwell at the time. Of such chance factors, no more need be said.

3 Controllability (Skill)

3.1 Cheating

The best way to exercise control over the outcome of a game is to cheat. This pits maximum controllability on your behalf against baffling uncertainty on the part of your victim, who, if sufficiently gullible, may look upon your constant success as a form of magic. And why not – when you consider that cheating and magic are little more than differing interpretations of the same conjuring trick?
3.2 Memory – Past and Future

The importance of memory is obvious, but its ramifications are subtle and it would be impossible to outline them here without going into disproportionate detail.

In his book *The Chess Mind*, player and analyst Gerald Abrahams holds that memory is easily overestimated, especially if it is taken to imply remembering a number of standard openings [8]. Very long retentiveness, indeed, is ‘often a concomitant of minds lacking in originality’. More important than the consciously recollected ‘is that set of mental habits which smooths the action of the mind’, a capability best described as ‘technique’, and most relevant in the endgame. Of greatest significance, however, is what he refers to as ‘holding in [one’s] mind a clear conception that is in part constituted by the memory of what will have happened, i.e. what has already happened as a mental event’.

Here we find ourselves talking about the forward visualisation involved in combinatorial games like Chess. What do we mean by forward visualisation? At first sight, it means looking ahead to our next move and to the sequence of moves likely to result from it. This is described as examining the branches of the strategy tree, and is something that computers are very good at. In human terms, it seems like a form of memory, only in reverse, in that we are following a sequence forward into the future rather than backward into the past.

I have always described this ability as mental ‘projection’, in that we are projecting ourselves into the future. Abrahams refers to it simply as ‘vision’ [8]. But in fact ‘future recall’ or ‘reverse memory’ is a pretty good term for it, as experiments have shown that exactly the same parts of the brain light up as when it is engaged in tracing memories backwards [9].

3.3 Intuition

Against this, however, must be set the discovery that Chess masters do not normally go down this analytical route on a step-by-step basis. World Chess champion José Raúl Capablanca, on being asked ‘How many moves do you see ahead?’, is said to have replied ‘Only one – but it’s always the best one’. As Philip Ross observes:

> He thus put in a nutshell what a century of psychological research has subsequently established: much of the chess master’s advantage over the novice derives from the first few seconds of thought. This rapid, knowledge-guided perception, sometimes called apperception, can be seen in experts in other fields as well. Just as a master can recall all the moves in a game he has played, so can an accomplished musician often reconstruct the score to a sonata heard just once. And just as the chess master often finds the best move in a flash, an expert physician can sometimes make an accurate diagnosis within moments of laying eyes on a patient. [6]

How do they do this? They see total situations as in a photographic memory, to such an extent that it seems to an outside observer – or to their hapless opponents – more like a stroke of intuition than a process of cerebral analysis and future recall. Ross notes that it was in 1894 that Alfred Binet, co-inventor of the first intelligence test, hypothesised that Chess masters achieved an almost photographic image of the board, but he soon concluded that the visualisation was much more abstract, resembling, rather, the same kind of implicit knowledge that the commuter has of the stops on a metro line [6].

The expert relies not so much on an intrinsically stronger power of analysis as on a store of structured knowledge, enabling them to reconstruct any particular detail at will by tapping a well-organised system of connections. A weaker player, confronted with a difficult position, may calculate for half an hour, often looking many moves ahead, yet miss the right continuation, whereas a grandmaster sees the move immediately, without consciously analysing anything at all. In brief, experts rely more on structured knowledge than on analysis.

Ross goes on to invoke the theory of information *chunking* developed by Chase and Simon of Carnegie Mellon University [10]. Simon explained the masters’ reconstruction of Chess positions with the aid of a model based on meaningful patterns called chunks, enabling them to manipulate vast amounts of stored information that would be expected to strain the working memory beyond its normal capacity to contemplate more than seven items at a time. An alternative explanation is offered by biologist Lois D. Isenman in her paper on intuition:

> Intuition as a bridging function brings the power of the unconscious into conscious thought. Through intuition, the unconscious with its vast memory banks, its associative accessing system, its speed, and its ability to process multiple items in parallel, greatly enriches the ability of conscious mental activity to manipulate logic and construct empirical tests... As dreams demonstrate, the unconscious frequently communicates in the language of symbols. In
symbol formation, each object can be represented by multiple associative categories. In symbolic expression, any one of these aspects can stand for the whole item; however, symbols often simultaneously encode a number of different levels, presenting a richly textured and very often surprising understanding of the object under consideration. In the unconscious, in effect, each item is categorised by all its different component parts, as well as its descriptive, situational, and affective associations. Intuitions very frequently come through to awareness in symbolic form and tend to share in the rich and unexpected quality that characterises unconscious mental processes... Associative processing, an important component of symbol formation, plays a central role in intuition whether or not intuition is expressed in consciousness in symbolic form.

Intuition may be closely allied to the skills of deduction and inference required of intelligent card games. Does it also shade into elements of extra-sensory perception as might be the case in the card game Pelmanism, also known as ‘memory’? If so, are we also encroaching on the borders of what some might categorise as ‘magic’?

3.4 Inference and Deduction

When I was a teacher in my early twenties I played a lot of Chess. At one school most of the staff played Kriegspiel even those members who did not normally play Chess. I soon discovered, to my surprise, that I was better at Kriegspiel than at standard Chess.

Then I noticed that the annual Kriegspiel tournament was invariably won not by the strongest Chess players but by the strongest Bridge players. It was obvious that one of the skills demanded by Kriegspiel was that of deduction or inference as to the positions of the playing pieces; that this was a skill particularly demanded of card-players; and that it might therefore be appropriate for me go in for card games rather board games.

In reading the literature on games I also became aware that Chess players tend to look down on card-playing on the grounds that cards are not games of perfect information, with the further implication that games of perfect information require more skill than games of imperfect information, which are therefore to be equated with games of chance. Poet E. J. Mortimer Collins, in Attic Salt, writes:

There are two classes of men, those who are content to yield to circumstances, and who play Whist; and those who aim to control circumstances, and play Chess.

But of course this is nonsense. Information is not absent from strategic card games: rather, it is released gradually as cards are played or announcements made, and much of the information that has not yet been revealed is to be deduced or inferred – or even ‘intuited’ – from that which has.

The acquisition of information is as much the goal of strategy in strategic card games as the positional moves made as a result of the knowledge acquired. Indeed, in the higher trick-taking games positional moves may be made specifically for the purpose of acquiring information, even at the expense of loss of material – a device equivalent to the gambit at Chess.

3.5 Specific and Peculiar Skills

Another question arises from personal experience. One of my favourite abstract board games is the game of Pentominoes, originally proposed by mathematician Solomon Golomb and sometimes referred to as Golomb’s Game.

Why do I enjoy this game so much more than games of the Chess/Draughts variety?

I would say that it involves what I call the packing skill. It is interesting that I happen to be very good at efficiently packing suitcases, loading excessive amounts of luggage into the car when going on holiday, and finding new ways of rearranging my expanding collection of books and videos without taking up much more space than they did when I started.

In what way does Pentominoes differ from (say) Chess or Draughts? My first observation is that it is a game of placement rather than movement. It starts with an empty board and play proceeds with each in turning placing a piece but not thereafter moving it. The same applies to another game I enjoy, namely Reversi (Othello). On these grounds, I often wonder whether I might have had some aptitude for Go, also a game of placement rather than movement, had I only discovered it early enough in life.

This is one example of how the classification of games may relate to a classification of different types of skill involved in playing them, and perhaps also taste and temperament. It is not surprising that some people will play only word games, and some only war games or fantasy games.
3.6 Types of Intelligence

I have long been seeking a classification of mental skills that might be applicable to games classification. In other words, can we classify games by reference to the skills required in playing them rather than directly by reference to the contents of the games themselves? Eric Solomon tells me that this was the basis of a booklet he was planning some years ago, but found too difficult to follow through. The most promising line of enquiry that I have discovered derives from Gardner’s concept of ‘multiple intelligences’ [15]. Gardner originally distinguished seven types of intelligence, then later added an eighth, as follows:

1. Linguistic.
2. Logical-mathematical.
5. Spatial.
6. Interpersonal.
7. Intrapersonal.
8. Naturalistic.

I will comment on these in a different order, which I think more relevant to their application to games.

1. **Spatial intelligence**: the potential to recognise and manipulate the patterns of wide space (those used, for instance, by navigators and pilots) as well as the patterns of more confined areas (such as those of importance to sculptors, surgeons, chess players, graphic artists, or architects). Obviously in most board games, with the possible exception of Mancala; irrelevant to most card games (even Patience) except those designed to imitate board game activity.

2. **Logical-mathematical intelligence**: the capacity to analyse problems logically, carry out mathematical operations, and investigate issues scientifically. Is this involved in the type of forward thinking relevant to most abstract board games? Does it relate to the deduction/inference skill of most card games? If not, how does deduction/inference fit into Gardner’s schema? Should we not also posit some kind of creative intelligence?

3. **Musical intelligence**: skill in the performance, composition and appreciation of musical patterns. I am not aware of any musical games, but it is worth noting the frequent association of skill at Chess with advanced musical skills. Many good musicians are good Chess players, and vice versa. In mediaeval universities, music and mathematics formed, with astronomy and geometry, a ‘quadrivium’ of subjects regarded (paradoxically, to modern minds!) as lower than the ‘trivial’ ones of grammar, logic and rhetoric.

4. **Linguistic intelligence**: sensitivity to spoken and written language, the ability to learn languages, and the capacity to use language to accomplish certain goals. At first sight, this would appear to be relevant only to word games. But consider what the essence of language is: it is the ability to encode our multi-dimensional experience of the world and our interaction with it into a linear stream of a limited number of discrete sounds, from 20 to 50 according to the language we use. This seems to me clearly related to both the logical-mathematical and the musical intelligence.

5. **Interpersonal intelligence**: a capacity to understand the intentions, motivations and desires of other people and, consequently, to work effectively with others; in other words, a theory of mind. If we convert the phrase ‘work effectively with others’ into ‘work effectively against others’, we find this obviously fundamental to the skills required to play any strategic game against a live opponent. An interesting sidetrack in this day and age, of course, is how relevant this ability is to playing against computer software. Interpersonal intelligence is especially relevant to all intelligent card games.

6. **Intrapersonal intelligence**: a capacity to understand oneself, to have an effective working model of oneself – including one’s own desires, fears and capacities – and to use such information effectively in regulating one’s own life. Poker; need I say more?

7. **Bodily-kinaesthetic intelligence**: the capacity to use one’s whole body or parts of the body (like the hand or the mouth) to solve problems or to fashion products. Obviously relevant to outdoor sports and to games of dexterity, manipulation, and hand-eye coordination. It would appear to be related to spatial intelligence.
8. **Naturalist intelligence**: relates to observing, understanding and organising patterns in the natural environment. A naturalist is someone who shows expertise in the recognition and classification of plants and animals. I have no suggestions to make in this regard.

9. **Creative intelligence?** There must be such a thing as creative intelligence, but I am not quite sure where it fits into Gardner’s scheme of things. I would take it to be at least related to, if not a form of, the skill of inference already mentioned. Abrahams remarks, in *Brains in Bridge*:

> What distinguishes the player of any of the best-known card games from the player of one of the main board games is that the former frequently analyses, whereas the latter always synthesises...[16]

Synthesising is a form of creativity, and inference a form of inductive reasoning, classically tested in Abbott’s celebrated game of Eleusis[17] – which, be it noted, is not a board game but a card game. For more on this point see Robert Harris’s 1998 essay ‘An Introduction to Creative Thinking’.[3]

### 3.7 Intelligence is not Skill

As interesting as the concept of multiple intelligences may be, I feel that it does not entirely answer my enquiry into the skills involved in game-playing; for a skill is not the same as an intelligence; rather, it is the application of an intelligence, which is not only a skill in itself but also introduces another potential element of differentiation into the subject. In brief, I have still not been able to track down a classification of mental skills as distinct from the mere intelligences on which they may be based. Further, ‘application’ may itself be regarded as a property essential to increasing one’s controllability of a game, in the sense that the more you apply yourself to a game the better your skills become. In other words, practice makes perfect, or as Gerald Abrahams describes:

> In the course of a simultaneous display... I [once] said to one of my opponents, ‘Tell me, Mr McMahon, how long did it take you to learn to play Chess so badly?’ He replied, ‘Sir, it’s been nights of study and self-denial’. [16 Preface]

### 4 Conclusion

Arthur C. Clarke’s Third Law of Prediction states that: ‘Any sufficiently advanced technology is indistinguishable from magic’[18]. We may well say the same of any sufficiently advanced intelligence.

### Acknowledgements

This material was first presented as a paper at the 2008 Board Game Studies Colloquium held in Lisbon titled ‘On Chance and Skill in Games’ then posted on the author’s web site *The Incompleat Gamester* under the title ‘On Chance and Skill in Table Games: How Strategy Outwits Uncertainty’ [1]. The piece has been reprinted here complete, with updates by the author, and formatted to fit the journal style with current references added.

### References


David Parlett is an inventor, author, journalist and consultant in the field of tabletop games and puzzles. He is the author of several books on games including The Oxford History of Board Games and The Oxford Guide to Card Games.

Address: South London, UK.
Email: games@parlett.eu

Gadeiro Challenges #11 and #12

Pack the pieces on the right to fill the shapes on the left. Gadeiro is described on pages 39–41.
Vol. 3, no. 2 2017

Editorial
Back to Basics. .............................................................. C. Browne 3

Nikoli Logic Puzzles
Stained Glass ................................................................. Y. Anpuku 5

Case Study
Disthex: A New Twist on Hex ........................................ R. Ishii and C. Browne 10

Maths in Games
Groups in Circle Puzzles .................................................. W. Kretschmer 15

Case Study
From Untouchable 11 to Hazmat Cargo .............................. C. Hoff 27

Game Design Patterns
Ludoku: A Game Design Experiment ............................... C. Browne 35

Maths in Games
Edit Games .............................................................. D. Ashlock and A. McEachern 47

Article
Measuring Drama in Snakes & Ladders .............................. J. Neto and J. N. Silva 56

Article
Dice Design Deserves Discourse ................................. S. Yermolaieva and J. A. Brown 64

Article
Tension in Puzzles .......................................................... C. Browne 71

From the Archive
Games and Politics ......................................................... R. Garfield 79

The front cover shows circle packing groups by William Kretschmer. See pages 15–26 for details.

All content © 2017, Game & Puzzle Design.
Game & Puzzle Design (G&PD) is a peer-reviewed research journal publishing high-quality work on all aspects of game and puzzle design.

Published through Lulu Press Inc., 3101 Hillsborough St, Raleigh, NC 27607, USA.

ISSN: 2376–5097
ISBN: 978-0-244-72393-4

http://www.gapdjournal.com
Back to Basics

Cameron Browne, RIKEN Institute

THERE seems to be a trend in recent years for new games to be bigger and brighter than their predecessors. This is especially obvious in the board game industry, where new titles released each year try out out-do each other in terms of production and appeal, but it also seems to be happening in the video game industry. Big production is in. Games are coming in bigger boxes, with more components, more detailed boards, and longer rule books to cope. And they certainly look good! A big, glossy, colourful box is eye-catching on the shelf. But what about the games themselves? Progress in this respect is not as encouraging, as the design of the underlying games appears to be taking second priority to marketing concerns.

Market Pressure

Industry contacts – both designers and publishers – are worried about this trend, but there is little incentive to arrest it. Consider the main publicity outlets for board games these days: large trade shows such as Essen SPIELE where thousands of games must compete for the attention of the crowds filing past, and online funding campaigns such as Kickstarter where presentation and unique selling points are paramount. Kickstarter stretch goals – bonus features to be added if certain funding milestones are met – further exacerbate the problem. Designers are obliged to list speculative features that look and sound good but which are not guaranteed to eventuate; these are by definition optional and not essential to the core game and may not even have been tested extensively. And in the event that the stretch goals are actually reached, then even if the game does end up being playable – which is not guaranteed! – those bonus features must be added whether they truly add value or not.

Form Over Function

A friend recently pointed out that there appears to be some confusion now between good game design and simply piling up mechanisms. ‘Wow, 40 pages of rules,’ he joked, ‘it must be good!’

I experienced this phenomenon first hand when another friend introduced me to one of last year’s most popular board games, and I literally fell asleep at the table at least twice in the 15 minutes it took him to explain the rules (sorry Ken!).

And the experience of playing that game was not much better; this popular design from a popular designer piled complexity upon complexity, until every action felt micromanaged under the sheer weight of rules that I could barely remember. These rules were probably carefully play-tested, fine-tuned and balanced, but the simple enjoyment of play seemed to have been lost along the way. I am not rushing to play that one again.

Getting Back on Track

This trend towards ‘bigger is better’ may be attributed to a cult of the new, and the emergence of a new generation of gamers for whom the classics of the past are quaint antiquities that lack the ‘wow’ factor and instant gratification of the latest blockbusters, and do not look as impressive on the shelf. So what can be done about this trend?

Designers have the power to refocus their attention on good design, and to resist market-driven temptations such as planned obsolescence in games. Players have the power of their wallet. If a game is underwhelming, do not buy its expansions, or the Star Wars-themed version, or next year’s bloated offering; they will probably not be any better.

My own preference is for abstract strategy games, which allow deep play with simple, elegant rule sets, although much of any game’s complexity can be hidden in the rules and equipment if it is well designed. But this Editorial is not necessarily a call for a return to simplicity, but rather a return to the principles of good design; even complex games can be attractive, elegant and engaging if they are carefully thought through and implemented.

Designers should concentrate on making games that players will enjoy for years rather than games that fly off the store shelves but are quickly forgotten. This is a call to restore games as works of art rather than consumer content.

This Issue

This issue’s contributions highlight this theme of going back to the basics (of design). While this is an undercurrent of all previous issues, it is especially prevalent here.

The opening article ‘Stained Glass’, from Nikoli’s Yoshinuo Anpuku and translator Ken Shoda, describes a simple Japanese logic puzzle

---

1http://www.gamesradar.com/video-games-have-become-too-complex-and-need-regress/

inspired by the design of stained glass windows, with carefully pared-down rules.

The article ‘Disthex: A New Twist on Hex’, co-authored with Japanese game designer Ryoji Ishii, describes the design of an elegant new Hex variant called Disthex. This game shows a good understanding of Hex, as a simple rule change complements the existing rules while fundamentally changing the character of the game.

William Kretschmer then dives into the complex world of group theory with ‘Groups in Circle Puzzles’, to describe a family of mechanical puzzles with an unashamedly mathematical basis. This piece demonstrates how fundamental mathematical principles can translate directly to interesting games and puzzles. Carl Hoff follows this with his back-story of the development of another maths-based mechanical puzzle in ‘From Untouchable 11 to Hazmat Cargo’.

My piece ‘Ludoku: A Game Design Experiment’ describes the somewhat deliberate design process behind a new Sudoku variant called Ludoku, in an effort to create a simplified version of the original game that is still strategically deep. You can judge the result for yourself, as Ludoku is this issue’s ‘feature puzzle’, with challenges printed throughout the issue were space permits.

‘Edit Games’ by Daniel Ashlock and Andrew McEachern then describes how the well-known concept of ‘edit games’ can be extended to describe families of existing games and puzzles, as well as used to create new ones.

João Pedro Neto and colleague Jorge Nuno Silva return to one of the most basic games of all in ‘Measuring Drama in Snakes & Ladders’. They show how simple metrics can be used evaluate variants, to help find interesting new twists on even the simplest of games.

Sofiia Yermolaieva and Joseph Brown examine one of the most basic pieces of game-playing equipment in ‘Dice Design Deserves Discourse’. They show that players’ preferences are not always the most practical when it comes to design.

My article ‘Tension in Puzzles’ aims to define the somewhat nebulous notion of ‘tension’ in terms of games, and extend this concept to puzzle design. The issue concludes with a reprint of Richard Garfield’s classic piece on ‘Games and Politics’, which includes the first known description of the ‘kingmaker effect’ in print.

References

Cameron Browne is a Research Scientist at the RIKEN Institute’s Advanced Intelligence Project (AIP) in Tokyo. His research interests include artificial intelligence and automated game design. Address: Mitsui Building, 15th floor, 1-4-1 Nihonbashi, Tokyo 103-0027, Japan. Email: cameron.browne@riken.jp

Ludoku Challenges #1 and #2
Fill the grid with numbers 1..9 such that no number is repeated in any row or column, and the diagonal neighbours of a number do not repeat that number or each other. Ludoku is described on pages 35–46.
Stained-Glass

Yoshinao Anpuku, Nikoli

Stained-Glass is a logic puzzle in which the solver must colour certain regions within a pattern to reveal a hidden target image. It is a puzzle that produces a work of art as it is solved.

1 Introduction

Stained-Glass is a Japanese logic puzzle [6], invented by Japanese designer Mitsuyuki Okuyama, which first appeared in Puzzle Communication Nikoli Issue 102 in 2003. Stained-Glass involves deducing which regions of a design must be coloured to reveal a target image. As with Ten-tai Show, another Nikoli puzzle [2], its appeal lies in producing a work of art as well as the satisfaction of finding a solution for each challenge.

A characteristic of Stained-Glass is that each challenge is made from free-form regions that can be of any shape (even curves are allowed). This makes it an attractive alternative for players wanting something different to the regular grids of most other geometric logic puzzles. Stained-Glass is unusual in that challenges do not necessarily involve any form of symmetry, as is the case with all other Nikoli puzzles.

2 Rules

Each Stained-Glass challenge consists of a field marked with lines or curves that define regions of arbitrary shape. Some circles in three colours (black, white and grey) are placed at boundaries where regions meet. The rules are as follows.

Rules of Stained-Glass

1. Black circles touch more shaded regions than unshaded regions.
2. White circles touch more unshaded regions than shaded regions.
3. Grey circles touch an equal number of shaded and unshaded regions.

For example, Figure 1 shows a typical challenge and Figure 2 shows its solution. Note that it is the number of pieces rather than the size of pieces that is important here; large and small pieces count equally.

Solutions are shown in multiple colours here, due to the artistic inspiration behind Stained-Glass. However, players will typically use a single colour in practice, as they solve each challenge with a pen or pencil.

3 Worked Example

The following figures show a worked example of solving a simple Stained-Glass challenge, to illustrate some of the basic strategies involved. Starting with the challenge shown in Figure 3, the first step is to look for circles that only cover one of two regions, as these are immediately solvable.

For example, the upper left white circle must cover two empty regions (marked with small dots as a reminder), while the middle black circle must cover two coloured regions, as shown in Figure 4.

Now, the leftmost black circle must cover more coloured regions than empty ones, and already covers two empty and two coloured ones, so the remaining region must be coloured, as shown in Figure 5. Similarly, the last unknown region covered by the topmost white circle must be empty, as it is known to already cover one coloured and one empty region, so it is marked with a dot. The remaining unknown region covered by the rightmost black dot must be coloured, to maintain its coloured majority.

Figure 5. Steps involved in solving a simple Stained-Glass challenge.

Figure 6 shows the final steps in the solution process. Two of the regions covered by the rightmost grey circle have been coloured, so the remaining two regions must be empty. Conversely, two of the regions covered by the lower grey circle are known to be empty, so its remaining unknown region must be coloured. Finally, the last unknown region covered by the lower left white circle must be empty, to maintain its majority.

Figure 6. Steps involved in solving a simple Stained-Glass challenge.

The key solution strategies can be summarised as follows:

1. White or black circles bordered by one or two shapes result in all bordering shapes being of the type specified by the circle.

2. Once at least half of the bordering shapes on a grey circle are known to be of one type, the other half should also be known.

4 Design

Stained-Glass was inspired by the ‘Tea Break Puzzle’ that designer Mitsuyuki Okuyama used to solve in the margins of mathematics texts at elementary school. These involved colouring specified regions within a field, such as those labelled with even numbers or marked as solid (such as the example shown in Figure 7), to reveal a hidden image. Mr Okuyama was attracted by the fact that these puzzles were not constrained to the square grid, and he looked for ways to introduce logical deductions into the basic idea.

![Figure 7. A Tea Break Puzzle-style challenge.](image)

An initial version of the puzzle that became Stained-Glass used numbers in the circles to indicate the number of regions to be coloured. Other versions included numbers on the pieces themselves, or rock/paper/scissors symbols such that the ‘winning’ region around each circle was coloured, and so on.

However, as chief editor of Nikoli, I pointed Mr Okuyama to Tentai Show [2] to show the basic principle that puzzles which create images should be simple and without numbers. We have found that Nikoli readers who enjoy finding images in puzzles do not enjoy solving difficult puzzles with various traps. Mr Okuyama took this comment on board, and produced the final version of Stained-Glass presented above.

4.1 Fine Tuning

One subtlety of Stained-Glass design is that the circles, which are necessarily placed where regions intersect, can interfere with the outline of the final coloured image when the challenge is solved. Tentai Show is superior in this respect, as its guiding black circles occur inside its regions and disappear when their regions are coloured.

However, this problem can be largely solved when designing Stained-Glass challenges by following a simple principle: avoid black circles on acute corners of coloured pieces. Also, it is best to minimise the number of circles in a challenge as much as possible. This not only yields neater final images, but adds to the challenge and can surprise players with the complexity of the image hidden by so few hints.

Mr Okuyama’s unpublished work also includes challenges in which the borders form part of the final image. Stained-Glass allows a variety of ways in which designers can express their individual tastes.

Unfortunately, there are not many designers willing to create Stained-Glass challenges at present. This could be due to the free-form nature of the puzzle, as defining arbitrary regions requires more skill than choosing points on a fixed grid [I can confirm that! – Ed.]. Another problem is that regions must be carefully judged if they are to hide the final image; too much detail can lead to uniquely shaped regions that broadcast the solution before the challenge is solved.

5 Conclusion

Stained-Glass is a simple logic puzzle, based on free-form shapes rather than a regular grid, that rewards the player by revealing a hidden image in its solution. I invite readers who like to draw to try their hand at devising their own Stained-Glass challenges.

Acknowledgements

Thanks to independent translator and game designer Ken Shoda for translating the original notes from Japanese, and the Editor-in-Chief for editing, formatting and illustrating this piece.

References


Yoshinao Anpuku is the chief editor of Japanese quarterly magazine *Puzzle Communication Nikoli*. Address: Nikoli Co. Ltd., 3-36-5 Nihonbashi-hamacho, Chuo-ku, Tokyo 103-0007, Japan. Email: an@nikoli.co.jp
Appendix: Large Examples

The following challenges demonstrate larger examples of Stained-Glass. The challenge shown in Figure 8 is by designer Mitsuyuki Okuyama (from Nikoli 106), while the full-page example shown in Figure 9 is by 3-ji Kansu (from Puzzle the Giant 26). Both of these magazines are published by Nikoli.

Figure 8. A large Stained-Glass challenge (by designer Mitsuyuki Okuyama).
Figure 9. A larger Stained-Glass challenge that includes curved region boundaries (by 3-ji Kansu).
Disthex: A New Twist on Hex

Ryoji Ishii, Independent
Cameron Browne, RIKEN Institute

Disthex is a variant of the classic connection game Hex, with one key difference that transforms the game. This article describes Disthex, its derivation, and the unexpected impact of a simple rule change, which resonates with the nature of the game and introduces significant new strategies.

1 Introduction

A common practice in game design is to take successful existing game ideas and look for ways in which they might be fruitfully modified or recombined [3]. This article describes one such case of a simple modification to the classic connection game Hex [2] that has a surprising impact on the nature of the game.

2 Hex and Disthex

Hex is the quintessential connection game [3]. Its rules are summarised below.

\begin{itemize}
\item Hex is played on an \(n \times n\) rhombus tessellated by hexagons, with two opposite sides coloured white and two opposite sides coloured black.
\item 1. Two players, White and Black, take turns placing a piece of their colour at an empty cell.
\item 2. Players win by forming a connected chain of their pieces between their sides of the board.
\end{itemize}

Hex is usually played with an additional swap rule to stop the first player making an overly strong opening move (towards the board centre):

\textbf{Swap Rule}: Instead of making their first move, the second player can elect to swap colours and steal the opening move.

For example, Figure 1 shows a \(7 \times 7\) game of Hex won by White, who has connected the white sides of the board with a connected chain of white pieces. Note that only a small part of this board is filled before the game ends, which is typical of Hex. Hex is typically played on larger boards (usually \(11 \times 11\) or \(14 \times 14\) [2]) and has in fact been solved for size \(7 \times 7\) [4].

2.1 Disthex

Disthex was designed by the first author in 2017, and is played as follows:

\textbf{Disthex} is played on a \(7 \times 7\) Hex board, as per Hex, except that players cannot play adjacent to the opponent’s last move. The mover must pass if no legal move is available.

For example, Figure 2 shows a game of Disthex won by Black, who has connected the black sides of the board with a chain of black pieces. Note that much more of the board has been filled compared to Hex, making Disthex more amenable to smaller boards. The basic mechanism that allows this, due to the novel ‘no-last-
adjacent’ rule, is described in the following sections. Disthex is also played with the swap rule.

In the figures, each player’s last move is indicated by a dot. In actual games, each player’s last move can be readily indicated by a second piece of the same colour stacked on top.

3 Derivation

Yavalanchor [5, pp. 17–18] is a variant of the abstract board game Yavalath [5, pp. 13–14]. The defining rule of Yavalanchor is that each turn the current player must either:

1. Place a neutral (red) piece, or
2. Place a piece of their colour adjacent to an existing neutral piece.

Figure 3 shows a game of Yavalanchor in progress. Crosses indicate cells adjacent to neutral (red) pieces at which the current player can place a piece of their colour.

![Figure 3. A game of Yavalanchor in progress.](image)

The first author was intrigued by this concept of anchor pieces, and decided to see whether this metarule could be successfully applied to other board games. The classic connection Hex came to mind, as its rules are as simple and elegant as Yavalath’s, and also played on a hexagonal grid.

Unfortunately, the anchor rule did not work very well when applied directly to Hex. The fact that both players could play adjacent to the neutral anchors, and that the anchors themselves counted in the connections of both players, made connection too easy and actually reduced the tension of the game. The anchor rule was therefore inverted so that players could not play adjacent to anchors, and anchors were redefined simply as the opponent’s last move.

This modified anchor rule worked well with Hex and produced surprising and interesting situations. The resulting game was named Distant Hex, which was abbreviated to Disthex.

3.1 Stalemates

It is a well-known property of Hex that a winning connection by one player precludes any possible winning connection by the opponent; every game must have exactly one winner. However, a problem soon emerged with Disthex that jeopardised this attractive property, as situations can arise in which the current mover cannot play at any of the remaining empty cells as they are all adjacent to the opponent’s last move. Figure 4 shows such a situation, in which Black has just moved.

![Figure 4. White must pass and lose.](image)

There are three obvious ways to handle such situations:

1. The mover must pass if there is no legal move (in which case White must pass and Black will win the game shown in Figure 4).
2. The mover may play next to the opponent’s last move if there are no other legal moves (in which case White will win the game shown in Figure 4).
3. The game is a draw (neither player wins).

Option 2 creates a cumbersome exception to the ‘no-last-adjacent’ rule and Option 3 loses the elegant ‘one-player-must-win’ property of Hex. We therefore decided on Option 1 – the mover must pass if there is no legal move – as the best way to handle such situations. However, this situation has not yet occurred in practice (to our knowledge) and is unlikely to occur in general play.

3.2 Board Size

We believe that the ‘no-last-adjacent’ rule works so well for Disthex because Hex is a game that involves regular passages of close combinatorial play, in which players trade turns close to the opponent’s last move – more so than any almost other game that we are aware of – hence forbidding adjacent replies subverts the character of the
game at a fundamental level. This constraint adds a lot of tension to the game, as the best moves that players really want to make are typically forbidden by the ‘no-last-adjacent’. Luckily, this tension produces surprising and interesting situations rather than simply being frustrating.

An added benefit of the new rule is that games of Disthex are harder to win. In Hex, a player will usually resign as soon as their opponent has established a series of virtual connection[1] between their pieces that guarantees a winning connection between their sides. For example, Figure 5 shows the most important virtual connection, called the bridge, that guarantees connection between two pieces; if the opponent intrudes with move 1 then the owner can restore the connection with move 2.

![Figure 5. Bridges in Hex are virtually connected.](image1)

But in Disthex, such virtual connections that would be guaranteed in Hex are no longer guaranteed and subject to attack, as illustrated below in Section 4 hence the battle continues beyond a point that would be considered ‘game over’ in Hex. Games of Disthex therefore tend to fill up much more of the board before the game is won than in Hex, making Disthex suitable for smaller boards. 7×7 was therefore chosen as the standard size, with games comparable in length and depth to Hex games played on an 11×11 board.

### 4 Strategies

This section outlines some basic strategies particular to Disthex, which highlight fundamental ways in which it differs from Hex.

#### 4.1 Bridge Intrusions

Bridges in Disthex can still be defended in isolation. For example, consider the white bridge shown in Figure 6 (left). If Black intrudes, then White can play the response shown to safeguard the connection (middle). Black is forced to play elsewhere, allowing White to complete the connection next turn (right).

![Figure 6. Isolated bridges can be defended.](image2)

However, bridges in Disthex are sensitive to the context of nearby pieces. For example, Figure 7 shows a white bridge with a nearby black piece (left). If Black intrudes as shown, then White cannot play the optimal response but must instead choose which side to defend (middle). Black can then play on the other side and threaten to cut White’s connection next turn (right).

![Figure 7. Nearby pieces can affect bridges.](image3)

Even a player’s own pieces can impede a bridge connection. For example, Figure 8 shows a white bridge with nearby black and white pieces (left). If Black intrudes as shown, then White’s own piece blocks the optimal response and White must instead choose which side to defend (middle). Black can then play on the other side to threaten to cut White’s connection as before (right).

![Figure 8. Friendly pieces can impede bridges.](image4)

#### 4.2 Decoy Moves

Consider now the case of multiple bridges, as shown in Figure 9 (left). If Black intrudes into one of the bridges, then White can make the optimal response shown above to protect it... at least temporarily (middle).

---

1. A virtual connection[1] consists of two pieces (or groups of pieces) A and B belonging to a player, and a set of empty cells, such that the opponent cannot stop the player from connecting A and B.
However, if Black then intrudes into the other bridge (right), then this constitutes a high-priority threat that White cannot ignore. White must choose between completing the first bridge (a) or protecting the second one (b) and in so doing give up their protection of the first; in either case, Black can cut one of the bridges next turn.

Black’s second move in this example is described as a decoy move, as it encourages the opponent to shift their last played piece, freeing up its previous location for a move. Such exchanges are reminiscent of ko battles in Go, in which players seek to play high-priority moves elsewhere, in order to change the board state and allow a capture that would otherwise be forbidden by ko. Decoy moves in Disthex empty much the same logic, but are much easier to maneuver in practice as players do not have to remember the previous board state (or all previous board states in the case of superko).

Decoy move exchanges introduce a Nim-like aspect of decoy move counting, in which players must ensure that they have enough high-priority decoy moves on the board in order to cover the number of moves that the opponent can play in a particularly desirable part of the board. Just as the ‘not-last-adjacent’ rule is the mechanism that constrains moves, decoy moves are the mechanism that free the board up again.

4.3 Strategy Stealing

The fact that having additional pieces on the board is not always beneficial for a player, as shown in Figure 8, has an even more fundamental impact on the game from a theoretical perspective. It means that the famous strategy-stealing argument proposed by John Nash [3, pp. 379–380] (see below), which proves that there is a winning strategy for the first player in Hex – even if that strategy is not actually known for all board sizes – does not apply to Disthex.

This proof relies on the fact that having extra pieces on the board can never harm a player’s position. This is not the case in Disthex, which invalidates the proof for this game and further highlights its fundamental difference to Hex.

5 Example

Figure 10 shows a simple Disthex puzzle: Black to play and win. If Black makes move 1 shown in Figure 11, then White can reply as shown to guarantee a winning white connection next turn.

Black must instead make move 1 shown in Figure 12 in order to win. White can attempt to block with a reply similar to that shown in Figure 7 but Black can respond as shown to guarantee a winning black connection next turn. White has no higher-priority decoy moves that might cause Black to relocate their last piece elsewhere, so that White could then play at the critical crossing point to win instead.

---

2The ko rule states that no moves can be made that would return the board to the same state it was last turn.
6 Conclusion

The ‘no-last-adjacent’ metarule extends the classic connection game Hex to produce the surprising and interesting variant Disthex that packs more punch into smaller boards. We believe that the success of this metarule in this case is due to the fact that it nicely complements Hex’s inherent tendency towards close combinatorial play.

Acknowledgements

Thanks to Ken Shoda for introducing the authors and helping to make this article happen.

References


Ryoshi Ishii is a freelance Japanese application/system programmer. Address: Kamisaginomiya, Nakano, Tokyo, Japan. Email: airtoxin@icloud.com

Cameron Browne is a Research Scientist at the RIKEN Institute’s Advanced Intelligence Project (AIP) in Tokyo. His research interests include artificial intelligence and automated game design. Address: Mitsui Building, 15th floor, 1-4-1 Nihonbash, Tokyo 103-0027, Japan. Email: cameron.browne@riken.jp
Groups in Circle Puzzles

William Kretschmer, Massachusetts Institute of Technology

Circle puzzles are sequential move puzzles that consist of intersecting rotating disks in the plane. Previous authors recognise the possibilities for diverse group-theoretic structure within this class of puzzles, but few have made progress in classifying all groups that can appear in circle puzzles. This paper discusses several methods that could help reach such a classification, including canonical decompositions into smaller groups, computational search for empirical classification, and a new algorithm for finding circle puzzle representations of particular sets of permutations.

1 Introduction

Circle puzzles are a class of sequential move puzzles in the plane that consist of intersecting disks. Each move involves rotating a disk by some fixed increment. As with other sequential move puzzles, the goal is to use these moves to restore a scrambled puzzle to a canonical starting state. An example circle puzzle is shown in Figure 1. The left disk has two stops, and the right disk has three stops. Available moves are to rotate these disks in increments of $\pi$ and $\frac{2\pi}{3}$ radians, respectively. The goal of the puzzle is to restore the current state with three separately coloured segments. Engel [1] offers a more detailed overview, highlighting the history and geometric properties of circle puzzles.

Figure 1. An example circle puzzle.

Like other sequential move puzzles, one can associate with each circle puzzle a group of permutations on moved pieces to precisely describe the set of reachable positions. Nevertheless, relative to other classes of puzzles, the group-theoretic nature of circle puzzles is poorly understood. For example, Wilson [2] showed that with a small number of exceptions, a class of sliding permutation puzzles embedded in graphs can only generate symmetric groups or alternating groups. This class of puzzles includes the famous 15 Puzzle. Scherphuis [3] and Yang [4] extended Wilson’s result to another type of graph-embedded puzzles that loosely correspond to circle puzzles with small overlap between disks.

Montenego et al. [5] analysed the group structure of puzzles based on rotating squares arranged in a grid. A move on such a puzzle consists of rotating a block of squares by 90 degrees. The authors showed that with some small exceptions, most groups in these puzzles also decompose into symmetric or alternating groups.

In contrast, little is known about the more general circle puzzles. Eidswick [6] observes that circle puzzle groups are generated by permutations of equal-length cycles, and that cycles belonging to two different generators share at most two moved pieces. However, he notes that these conditions are not sufficient for a particular set of generators to have a corresponding circle puzzle, and also notes that there is little research on permutation groups satisfying these properties.

This article presents several separate but complementary paths for studying circle puzzle groups. First, it discusses canonical ways of classifying circle puzzle groups by decomposing them into smaller groups in Section 2. In Section 3 the article discusses how to identify groups in a particular circle puzzle by computing a permutation representation, and notes several patterns that might lead to a general classification. Finally, the article introduces in Section 4 an algorithm for deciding whether a particular group can be realised by a (slightly more general) circle puzzle, strengthening Eidswick’s results.

This article assumes some basic familiarity with group theory; a glossary of the terms used is provided in the appendix.

2 Decomposing Groups

2.1 Simple Groups

As with many other mathematical objects, groups are often best studied by breaking them up into...
smaller pieces. One way to break up any finite group is to take a decomposition into so-called *simple groups*. Intuitively, simple groups act like prime numbers: just as every positive integer can be expressed uniquely as a product of prime numbers, so every finite group can be expressed as a ‘product’ of simple groups. There are some caveats, e.g. two inequivalent groups can decompose into the same set of simple groups. Nevertheless, the Jordan-Hölder theorem guarantees that the set of simple groups that make up a finite group are unique up to reordering.

### 2.2 Primitive Groups

For groups of permutations, sometimes it makes more sense break apart a group by looking at how it acts on smaller sets of pieces. Perhaps the easiest way to see this is through a canonical example: the 3×3 Rubik’s cube. Anyone who has solved a 3×3 Rubik’s Cube knows that stickers on corner and edge pieces are inexchangeable. This motivates decomposition by piece type: we can consider separately the actions on each set of mutually exchangeable pieces or stickers. For the 3×3×3 cube, this means that we consider the edge pieces and corner pieces separately (we usually ignore fixed pieces, which in this case include the centre pieces).

Within a single piece type, we can decompose further. For example, the 24 corner stickers on the Rubik’s Cube are grouped into eight pieces of three stickers each. One can use this observation to extract two groups: (1) the action on the eight corner pieces, and (2) the action on the three stickers of a correctly positioned (but perhaps incorrectly oriented) corner piece. Because every permutation of the corner pieces is reachable, the first group is the symmetric group $S_8$. The second group is the cyclic group $C_3$ because there are three ways to orient a corner, and because this group of orientations is generated by either a single clockwise or counterclockwise rotation of the corner.

This sort of decomposition generalises: it is known as a *block decomposition* or sometimes a decomposition into *primitive groups*. In decomposing the Rubik’s Cube, we partitioned the elements into separate *orbits*: sets of mutually exchangeable elements. Within a single orbit, we found *blocks of imprimitivity*: a partition of the elements that is itself acted upon by the group. In general, one can iterate the process of partitioning into orbits and searching for blocks of imprimitivity until the permutation group is fully decomposed into *primitive groups*. A rigorous treatment of this subject is beyond the scope of this article, but formal definitions and a proof of this are given in e.g. [7].

### 2.3 Decompositions in Circle Puzzles

An important observation is that the block decomposition often reflects a viable solving method. Indeed, many Rubik’s-type twisty puzzles can be solved by permuting and then orienting one piece type at a time. This article is not the first to make note of this; Eidswick [6] also observed the usefulness of this decomposition for puzzle solvers. For this reason, the block decomposition is arguably the most sensible way to study groups in permutation puzzles.

Nevertheless, block decompositions are not always so structured. In general, blocks need not correspond to physically attached stickers or pieces. Figure 2 shows an example: this circle puzzle has a set of twelve exchangeable pieces that splits into four blocks of three pieces each, but pieces within a single block are not connected.

![Figure 2](image_url)

*Figure 2. A single orbit of pieces, partitioned into four blocks of imprimitivity.*

Block decompositions of physically disconnected pieces can sometimes identify other common structure. For example, the block decomposition highlighted in Figure 3 reveals blocks that would behave as separate orbits if, instead of a move of order 2 and a move of order 3, the puzzle consisted of three moves of order 2.

Block decompositions also have some purely group-theoretic limitations. In general, block decompositions are not unique (i.e. some permutation groups can be partitioned into blocks in multiple ways). Additionally, block decompositions may fail to capture other interesting group-theoretic properties. For example, the orbit highlighted in Figure 2 is surprisingly equivalent to the so-called projective general linear group $\text{PGL}_2(\mathbb{Z}/9\mathbb{Z})$, a group constructed from $2 \times 2$ matrices whose entries are integers (mod 9). However, the block decomposition breaks only into cyclic and symmetric groups.
3 Groups from Circle Puzzles

If an ultimate goal is a complete classification of circle puzzle groups, then a first place to start is to examine the structure of some examples. Modern computer algebra systems, such as GAP [9], make it easy to automatically perform the permutation group decompositions described in the previous section. Thus, all that is needed is a procedure for computing permutation representations of arbitrary circle puzzles. The section that follows describes some of the challenges that arise in implementing such a procedure. Following that is a discussion of some patterns observed in a brief (but not exhaustive) computational search.

3.1 Computing Groups in Circle Puzzles

A circle puzzle is defined by its set of moves. Each move involves rotation of a disk by some fixed increment. Thus, each move can be specified by the $(x, y)$ coordinates of the disk centre, a radius $R$, and a turning increment $N$ that defines the angle of rotation $\frac{2\pi N}{x}$.

Given such a set of moves, computing the permutation representation of the corresponding circle puzzle requires first computing the shape of each moving part. Once parts have been identified, one obtains a set of permutations by computing the image of each part under each move. Associating each part with a unique identifier yields a permutation for each move on a shared set of numbered indices.

Geometrically, parts on circle puzzles are formed by intersections and differences of disks, so the shape of a part can be specified by the set of arcs on its boundary. As a result, one algorithm for computing the shape of each part is to first compute the complete set of cuts that separate parts, without initially associating each cut with its adjacent parts. This involves recursively computing the image of the boundary of each disk under rotations by other moves. Once a complete set of cuts have been computed, one can group arcs into closed loops to obtain part boundaries.

Of course, this brief description overlooks many important implementation details:

- One must choose a numerical representation that maintains sufficiently high precision, and that allows for comparisons (and ideally, hashing). Another option is to perform exact computations on algebraic numbers in a computer algebra system.
- Several choices for the representations of geometric objects are available. As an example, arcs can be stored independently; however, a data structure that groups together arcs around the same circle is often desirable for combining new arcs with previously computed arcs that overlap.
- An essential part of the algorithm is computing the set of points where cuts intersect. The naïve algorithm takes quadratic time (i.e. by testing every pair of arcs for intersection), but better algorithms are known; see e.g. [10].
- The procedure for computing part boundaries from the complete set of cuts is non-trivial, but is comparable to the process of computing the faces (i.e. regions bounded by edges) of a planar graph. It is often easier to assume that part boundaries are
simply connected; i.e. parts have no holes in them. This is always true if no part completely encloses another move. In any case, an orbit of enclosed parts can be block-decomposed by enclosing parts, so such puzzles cannot introduce new groups.

- This algorithm does not address part orientation: parts with certain symmetry may be rotated multiple ways to fit into the same location. Of course, this can only introduce cyclic groups using the block decomposition by part (compare to orientable parts on the Rubik’s Cube), so for the purpose of finding ‘interesting’ groups, orientation can be safely ignored.

- The algorithm fails to terminate on infinite puzzles, i.e. puzzles wherein the cut patterns become infinitely dense. It is an open problem whether a set of moves yields a finite (‘doctrinaire’) or infinite (‘jumbling’) circle puzzle. This has been discussed on the Twisty Puzzles forum.

These details are addressed in an open-source Scala library available on GitHub.

### 3.2 Search and Classification

Searches using the previously described algorithm have led to the discovery of several groups that were never before identified in twisty puzzles (shown in the appendix), and has also revealed some patterns. Of note, no primitive circle puzzle group acting on more than thirteen points has been identified that is not a cyclic, dihedral, symmetric, or alternating group. Indeed, by taking disks with large overlap, it is easy to construct circle puzzles with thousands of parts for which every piece type decomposes into the aforementioned groups; see Figure 4 for an example.

Why is it difficult to find other groups acting on large numbers of pieces? For one, it is hard to perform a systematic search of circle puzzles containing many pieces. Among circle puzzles with rotating disks of large overlap, it is easy to construct circle puzzles with thousands of parts for which every piece type decomposes into the aforementioned groups; see Figure 4 for an example.

Why is it difficult to find other groups acting on large numbers of pieces? For one, it is hard to perform a systematic search of circle puzzles containing many pieces. Among circle puzzles with rotating disks of large overlap, it is easy to construct circle puzzles with thousands of parts for which every piece type decomposes into the aforementioned groups; see Figure 4 for an example.

Taking a random pair of generators on \( n \) points yields a symmetric or alternating group with high probability that asymptotically approaches \( 1 - \frac{1}{n!} \). Thus, one could believe heuristically that no such groups are known only because they become much rarer and more difficult to find when the number of pieces gets large.

### Conjecture 3.0.1

With finitely many exceptions, every primitive group acting on \( n \) points obtained from a circle puzzle by orbit and block decomposition is one of the following:

1. The cyclic group \( C_n \), with \( n \) prime.
2. The dihedral group \( D_n \), with \( n \) prime.
3. The alternating group \( A_n \).
4. The symmetric group \( S_n \).

All known exceptions are shown in the appendix.

### 4 Circle Puzzles from Groups

Another way to better understand circle puzzle groups is to answer the question: when can one realise a particular permutation group with a circle puzzle? A classification in the style of Conjecture 3.0.1 might answer this question based...
on the structure of the group, but perhaps one would like to know whether a particular set of generators for such a group can be realised physically. For example, one might want to choose a set of permutations that maximises the number of moves needed to solve a puzzle, as this would yield a maximally difficult puzzle.

This question is not entirely new. Indeed, in designing a physical puzzle to realise the Mathieu group $M_{12}$, van Deventer and Kriz [13] asked whether their approach is general: can one always construct a mechanical puzzle to realise a specific set of permutations? A new algorithm presented here solves this problem for a class of circle puzzle variants wherein pieces are imagined to be infinitely small points.

4.1 Thin Circle Puzzles

The puzzles considered previously are defined by rotating disks: around each centre of rotation, all points within a certain radius rotate together to complete a move. Here, we instead imagine that moves only rotate points at a discrete set of radii around the centre of rotation. This article refers to puzzles of this type as thin circle puzzles; the circles introduced previously might be called disk circle puzzles to avoid ambiguity. On thin circle puzzles, the intersection of two moves will be a finite set of points, so one only considers the action on these points of intersection and their images under each of the moves. Figure 5 shows an example of such a puzzle and its moved points. Each move is a simultaneous rotation about a set of concentric circles, and correspond to the permutations $(1, 2, 3)(4, 5, 6)(7, 8, 9)$ and $(1, 8, 12, 6)(2, 11, 3, 10)$, which together generate the Mathieu group $M_{12}$.

Circle puzzles of this type were implicitly introduced previously on the Twisty Puzzles forum[1] but to the author’s knowledge, discussion of these puzzles in the literature is new.

Formally, a thin circle puzzle consists of a set of labelled points in the plane and a set of moves. Each move is associated with a turning increment $N$ and a set of concentric circles in the plane. Moves act on labelled points by counterclockwise rotation along the concentric circles in increments of $\frac{2\pi}{N}$ radians. Crucially, concentric circles belonging to the same move rotate simultaneously, and all labelled points lying on a rotated circle are rotated. Thus, the set of labelled points must be closed under rotation by each of the moves. As with other sequential move puzzles, the goal of the puzzle is to restore labelled points to their initial positions by rotation.

Thin circle puzzles have the useful property that every disk circle puzzle corresponds to a thin circle puzzle, obtained by choosing a representative point from each piece. Figure 6 shows an example.

Figure 5. Thin circle puzzle with two moves.

Figure 6. Top: one piece type in a disk circle puzzle. Bottom: its corresponding thin circle puzzle.

Consequently, if one can show that a particular group cannot be realised by a thin circle puzzle, then it certainly cannot be realised by a disk circle puzzle either.

Of course, not every thin circle puzzle corresponds to disk circle puzzle. The thin circle puzzle in Figure 5 has no corresponding disk circle puzzle because disks for both available moves would have to enclose points that should be fixed by those moves.

Figure 5 also highlights how thin circle puzzles might offer simpler embeddings of large groups. In particular, thin circle puzzles can isolate particular piece types, eliminating the dozens of small and 'uninteresting' pieces in a puzzle like the first one shown. Thus, thin circle puzzles might actually represent a more practical model for construction, assuming one can devise a mechanism to force concentric circles to rotate simultaneously.

4.2 Circle Puzzle Linear Algebra

To find thin circle puzzles for a set of generators \( G \) acting on a set of indices \( M \), it helps to first consider a related problem: given a set of labelled points in the plane, decide if they correctly realise \( G \) as a thin circle puzzle. As it turns out, this procedure becomes simpler if the coordinates of points are given as numbers in the complex plane, so one can suppose that the labelled points are specified by a vector \( v \) where \( v_a \) is the \( C \)-coordinate corresponding to a label \( a \in M \). It will also sometimes be helpful to suppose that the coordinates of rotation centres are given. For a generators \( g \in G \), we will similarly use \( v_g \) to denote the corresponding centre of rotation. Overall, one can view \( v \) a vector of \(|M| + |G| \) complex numbers.

Using this notation, there are just 3 conditions that \( v \) should satisfy:

- **Consistency**: given a cycle of \( N \) labels permuted by \( g \in G \), the corresponding points should form the vertices of a regular \( N \)-gon in the plane, with labels going counterclockwise. Furthermore, the centre of this polygon should be precisely \( v_g \), the centre of rotation by \( g \).

- **Point non-degeneracy**: no two moved points should have the same coordinate. In symbols, for each \( a, b \in M \) with \( a \neq b \), we must have \( v_a \neq v_b \).

- **Generator non-degeneracy**: recall that the definition of a thin circle puzzle requires that if some move rotates a labelled point in a circle, then all other points in that circle must also rotate. Thus, for a generator \( g \in G \), if \( b \in M \) is not rotated by \( g \), then \( b \) should not lie on any of the circles rotated by \( g \). In symbols, we use \( g(b) \) to denote the new location of \( b \) after application of \( g \). Thus, \( b \) is fixed by \( g \) if and only if \( g(b) = b \). Using this notation, the desired condition is: for each \( a, b \in M \) with \( g(a) \neq a \) and \( g(b) = b \), we must have \(|v_b - v_g| \neq |v_a - v_g| \).

A surprising observation is that the consistency constraint is linear in the coordinates of \( v \). Indeed, suppose \( g \in G \) cycles the points \( \{x_1, \ldots, x_N\} \subset M \) as \( g(x_n) = x_{n+1} \) for \( 1 \leq n < N \), and \( g(x_N) = x_1 \). Then, one can encode the consistency constraint for this cycle using the following equation:

\[
e^{\frac{2\pi}{N}}(v_{x_n} - v_g) = v_{x_{n+1}} - v_g \quad \forall 1 \leq n < N \tag{1}
\]

Essentially, Equation 1 says that rotating the point corresponding to \( x_n \) about the centre of \( g \) by \( \frac{2\pi}{N} \) radians should yield the point corresponding to \( x_{n+1} \). This intuition is depicted in Figure 7, in which \( v_1 - v_g \) and \( v_2 - v_g \) are drawn as pointed arrows. Multiplication by \( e^{\frac{2\pi}{N}} \) rotates counterclockwise by \( \frac{2\pi}{N} \) radians in the complex plane. This equation also shows why complex numbers come in handy: multiplication by a root of unity corresponds to a rotation. Note that only \( N - 1 \) constraints are needed because rotating \( N \) times by \( \frac{2\pi}{N} \) radians about any point is equivalent to performing no rotation at all.

![Figure 7. Visualising the consistency condition.](image)

Why is linearity useful? It means that with a little bit of matrix algebra, it is possible to succinctly represent (and compute!) the set of all possible consistent circle puzzles. In particular, it means that the set of consistent circle puzzles for a particular set of generators form a vector space over the complex numbers. Furthermore,
we can compute a basis for this vector space: a set of vectors that can be added together to uniquely represent every element of the vector space. This reduces the problem of finding circle puzzles to the problem of finding vectors in our vector space that satisfy the criteria for point and generator non-degeneracy, as described previously.

One can decide if there are any non-degenerate circle puzzles by inspecting an arbitrary basis. The formal statement of this follows:

**Theorem 4.1:** Let B be a basis for V, the k-dimensional C-vector space of consistent thin circle puzzles for a set G of generators acting on a set of labels M (k ≤ |M| + |G|). Consider the following conditions:

1. For all \(a, b \in M\), there exists \(v \in B\) with \(v_a \neq v_b\).
2. For all \(g \in G\), for each \(a, b \in M\) with \(g(a) \neq a\) and \(g(b) = b\), either there exists \(v \in B\) with \(v_a \neq v_b\) and \(|\frac{v_a - v_b}{v_b - v_a}| \neq 1\), or there exist \(v, w \in B\) with \(v_a \neq v_g, w_a \neq w_g\) and \(|\frac{v_a - v_b}{v_b - v_a} - \frac{w_a - w_b}{w_b - w_a}| \neq 0\).

If these conditions hold, then the set of degenerate circle puzzle representations in V is a null set with respect to \(2k\)-dimensional real Lebesgue measure (i.e. it has \(2k\)-dimensional volume 0 when V is identified with \(\mathbb{R}^{2k}\)). Otherwise, if any conditions fail to hold, then all circle puzzles in V are degenerate. □

Evidently, the statement of this theorem is rather technical, and the proof of this theorem (given in the appendix) is even more dense. Nevertheless, the key takeaways from this mathematical result are quite intuitive.

First of all, this theorem gives a deterministic procedure for deciding if there exist circle puzzles that satisfy the non-degeneracy criteria. In particular, both of the conditions in the statement of the theorem can be checked efficiently given a basis for the vector space of consistent thin circle puzzles. The theorem also states that these conditions are both sufficient and necessary, which is the most that one could hope for.

Second, this theorem precisely quantifies the space of valid thin circle puzzles for a particular set of generators. Indeed, the theorem shows that either all consistent circle puzzles have some degeneracy, or almost all consistent circle puzzles have no degeneracies! It is quite remarkable that one can only wind up with one of two extremes, and nowhere in between.

Lastly, this theorem gives rise to a practical procedure for actually finding non-degenerate circle puzzles: simply choose a random element of the vector space. Why does this work? In the previous paragraph, ‘almost all’ carries with it a strong measure-theoretic condition. Without going into too much detail, it essentially says that for any ‘reasonable’ random distribution over our vector space, the probability of selecting a degenerate circle puzzle will be 0. So, if non-degenerate circle puzzles exist, one should expect to find them with high probability.

### 4.3 Implementation and Discussion

If one is interested turning the previously derived results into an algorithm for finding thin circle puzzles, then in principle one only needs a tool that can perform linear algebra over the complex numbers. Unfortunately, one has to be a bit more careful than that because given the interest in finding non-degenerate circle puzzles, numerical stability is key.

In fact, issues of numerical stability can be avoided completely. Notice that the only irrational coefficients needed to encode the constraints are roots of unity (i.e. complex numbers of the form \(e^{\frac{2\pi i}{N}}\) where \(N\) is an integer). This implies that all of the linear algebra can be performed over a cyclotomic field: a subset of the complex numbers containing sums and products of rational numbers and roots of unity.

Thankfully, modern computer algebra systems can represent elements of cyclotomic fields exactly in terms of rational numbers, and as a result floating point and fixed precision arithmetic are unnecessary. Indeed, a reference implementation of this algorithm in Sage [14] is offered, available through the CoCalc platform.

![Figure 8. A thin circle puzzle acting on 28 points.](https://cocalc.com/projects/5b48c192-de77-4d90-989b-f3e03481678c/files/Circle puzzles from groups.sagews)
Use of this implementation has already led to the discovery of new thin circle puzzle representations for particular groups. For example, Figure 5 shows the only known thin circle puzzle with two moves that generates the Mathieu group $M_{12}$. Figure 8 shows a thin circle puzzle that acts primitively on 28 points, isomorphic to the symmetric group $S_{28}$, suggesting that the set of exceptional thin circle puzzles (in the sense of Conjecture 3.0.1) may be larger than the set of exceptional disk circle puzzles.

It remains largely unclear whether one should expect similarity between the classifications of thin circle puzzle groups and disk circle puzzle groups, in part because it is much more challenging to empirically explore the space of thin circle puzzles. Whereas disk circle puzzles are specified by just a few parameters (location, size, and turning increment of each disk), this algorithm for thin circle puzzles can take as input an arbitrary set of permutations. Furthermore, if one wants to represent a particular permutation group by e.g. selecting generating permutations at random, then perhaps the chance of finding one of the few generating sets that yields a non-degenerate circle puzzle becomes much smaller as the group gets larger. Thus, new techniques are needed to make searching for larger thin circle puzzles practical.

One can easily extend the previously described algorithm to several thin circle puzzle variants, including puzzles that allow generators consisting of multiple cycle lengths, or puzzles that allow non-concentric rotations for a single generator. Nevertheless this algorithm also has some limitations. For example, one might want to consider circle puzzles wherein a particular generator is realised by either a clockwise or a counterclockwise turn. This algorithm only considers one rotation type at a time, so testing all possibilities could require exponentially many (in the number of generators) calls to the algorithm.

5 Conclusion

The group-theoretic structure of circle puzzles remains only partially understood, but this paper opens several new avenues for research. The circle puzzle decision algorithm gives the strongest known mathematical restrictions on circle puzzle groups; the hope is that these conditions will help to produce a classification in the style of Conjecture 3.0.1.

The newly found circle puzzle groups should also be of interest to puzzle designers. Kriz and Siegel [15] offered a challenge to implement a physical puzzle containing the Mathieu Group $M_{12}$; the circle puzzle seen in Figure 5 might well yield one of the simplest possible mechanical realisations. At first glance, it seems more elegant than e.g. van Deventer’s Topsy Turvy and Number Planet [13].

In fact, the newly discovered groups have already lead to the creation of new puzzles, such as the author’s recently released Modular Cube shown in Figure 9. The four piece types are exceptions to Conjecture 3.0.1, and several other piece types have surprising group-theoretic properties.

Before closing, the author suggests several directions and open questions for future research:

- Extend the results in this paper to circle puzzle equivalents on the sphere. Is there an algorithm similar to the one described in Section 4 for deciding if a particular group can be realised by a spherical circle puzzle? Which groups can be realised on the sphere but not in the plane, or vice-versa?
- Make the algorithm for thin circle puzzles more useful in practice. Is there an algorithm that decides whether a thin circle puzzle can be modified into a disk circle puzzle? Are there practical ways to find generators for large groups that produce non-degenerate thin circle puzzles?
- Classify the group-theoretic structure of finitely-generated infinite/jumbling circle puzzles. Do all infinite circle puzzles contain a permutation that can be applied infinitely many times without returning to the starting state? Some finitely generated groups can be described in terms of a finite presentation: a finite set of relations in terms

http://www.twistypuzzles.com/cgi-bin/puzzle.cgi?pkey=6303
of the generators that completely describe the group. Do any or all infinite circle puzzles admit finite presentations?

Acknowledgements

The author would like to thank Bram Cohen and Brandon Enright for helpful discussions regarding the discovery and classification of the puzzles shown in the appendix. The author would also like to thank Keaton Stubis for helping to formalise the proof of Theorem 4.1, and the anonymous reviewers for constructive feedback.

References


William Kretschmer is an undergraduate student of mathematics and computer science at MIT, with research interests in theoretical computer science.

Address: Department of Mathematics, MIT, Cambridge, MA 02139, USA
Email: kretsch@mit.edu

Appendix

A.1 Glossary of Group Theory Terms

This article uses the following terms from group theory:

Alternating Group: The group \( A_n \) of \( \frac{n!}{2} \) even permutations on \( n \) points. An even permutation is a permutation that can be expressed in terms of an even number of swaps (cycles of length 2).

Cycle: Also called a cyclic permutation, a cycle of order \( n \) permutes \( n \) elements \( \{x_1, \ldots, x_n\} \) in a way that sends \( x_i \) to \( x_{i+1} \) for \( 1 \leq i < n \), and sends \( x_n \) to \( x_1 \).

Cyclic Group: The group \( C_n \) of order \( n \) generated by a single element of order \( n \). Some authors denote this as \( Z_n \) or \( \mathbb{Z}/n\mathbb{Z} \), because this group is equivalent to the additive group of the integers \( \text{mod } n \).

Dihedral Group: The group \( D_n \) of \( 2n \) symmetries (rotations and reflections) of a regular \( n \)-sided polygon. Some authors write \( D_{2n} \) instead.

Generator: A set of group elements are generators (i.e. they generate the group) if every group element can be expressed as a product of these generators. In the context of permutation puzzles, generators almost always refer to the set of available moves. For example, the \( 3 \times 3 \times 3 \)
Rubik’s Cube is generated by quarter turns of the six faces.

**Orbit:** The possible images of a single point in a permutation group. In permutation puzzles, the orbit of a piece is the set of reachable positions.

**Order:** The order of a group $G$ is the number of elements in $G$. The order of group element $g \in G$ is the smallest integer $n$ such that $g^n$ is the identity element of $G$. On the $3 \times 3 \times 3$ Rubik’s Cube, for example, the order of the group is roughly $4.3 \cdot 10^{19}$, the order of a quarter-turn is 4, and the order of a half-turn is 2. Groups of infinite order can have elements of infinite order.

**Permutation:** A bijection from a set to itself. Most commonly, this set is a set of positive integers. Permutations on such numbered indices are often written in cycle notation. For example, $(1, 2)(3, 4, 5)$ denotes the permutation that sends 1 to 2, 2 to 1, 3 to 4, 4 to 5, and 5 to 3.

**Symmetric Group:** The group $S_n$ of $n!$ permutations on $n$ points.

### A.2 Exceptional Circle Puzzle Groups

The figures that follow show circle puzzles containing all of the known exceptional groups suggested in Conjecture 3.0.1. In general, similar piece types can be realised over a range of puzzles with varying radii. The configurations shown here were chosen only to make clearly visible the relevant piece types. Some groups have multiple embeddings in circle puzzles. In such cases, all known non-identical embeddings are shown, and geometrically equivalent configurations with different generators are stated. All known examples involve circles placed symmetrically about a single point, and most have just two circles.

The following notation is used for these puzzles. For a puzzle with $k$ symmetrically placed circles, circle $j$ is assumed to be centred at $(\cos(\frac{2\pi j}{k}), \sin(\frac{2\pi j}{k}))$. $R_j$ denotes the radius of circle $j$. $N_j$ indicates the turning increment: circle $j$ turns in increments of $\frac{2\pi}{N_j}$ radians.

The notation for group names used here is roughly consistent with the Group Properties Wiki, e.g. in GL, SL, PGL, PSL, PTL, etc. Notice that Figure 13 is not identical to the orbit in Figure 14. In fact, this orbit is harder to solve, requiring up to 20 moves compared to 14 for the previous puzzle.

---

https://groupprops.subwiki.org/wiki/Main_Page
https://www.jaapsch.net/puzzles/pgl25.htm
Figure 13. $GL_3(F_2) \cong SL_3(F_2) \cong PGL_3(F_2) \cong PSL_3(F_2) \cong PSL_2(F_7)$ of order 168 on eight pieces. $N_1 = N_2 = 3, R_1 = R_2 = 4.5$. Equivalently realised in $N_1 = 2, N_2 = 3, R_1 = R_2 = 10$.

Figure 14. $PGL_2(F_7)$ of order 336 on eight pieces. $N_1 = 2, N_2 = 3, R_1 = R_2 = 3.2$.

Figure 15. $PGL_2(F_7)$ of order 336 on eight pieces. $N_1 = 2, N_2 = 3, R_1 = R_2 = 3.2$. Equivalently realised in $N_1 = 2, N_2 = 6, R_1 = R_2 = 3.2$.

Figure 16. $\text{Aut}(A_6) \cong \text{Aut}(S_6) \cong PTL_2(F_9)$ of order 1,440 on ten pieces. $N_1 = 2, N_2 = 4, R_1 = R_2 = 2\sqrt{2}$. This puzzle is the mass-produced Eliac.\[http://twistypuzzles.com/cgi-bin/puzzle.cgi?pkey=6009\]

Figure 17. $\text{Ree}(3) \cong PTL_2(F_8)$ of order 1,512 on nine pieces. $N_1 = N_2 = 3, R_1 = R_2 = 2.4$. Equivalently to an orbit on the Trapentrix.\[http://twistypuzzles.com/cgi-bin/puzzle.cgi?pkey=3346\]

Figure 18. $\text{Ree}(3) \cong PTL_2(F_8)$ of order 1,512 on nine pieces. $N_1 = 2, N_2 = 3, R_1 = R_2 = 3.8$.\[http://twistypuzzles.com/cgi-bin/puzzle.cgi?pkey=6009\]
forms a vector space. Also note that

\[ |v| = \begin{cases} 1 & \text{if } v \neq \mathbf{0} \\ \text{undefined} & \text{if } v = \mathbf{0} \end{cases} \]

in the case of the second bullet, hence \(|c| \neq 1\). Conversely, \(c\) must be finite otherwise one would have \(v_a = v_b = v_{g(a)}\) for every \(v \in B\), again violating the non-degeneracy criterion. Thus, the vectors with \(|r| = 1\) or undefined are just the representations with \(r\) undefined. Because at least one basis vector must have \(r(v) = c\), the set of vectors with \(r\) undefined is a proper subspace, which again is a null set.

Otherwise, \(r(v)\) takes on multiple defined values for \(v \in B\), necessarily putting us in the second case of the second bullet. Because \(V\) is closed under translations, without loss of generality suppose that \(v_G = 0\) for all \(v \in B\) (formally, this can be done by a projection). Consider the further projection onto \(C^2\) that sends \(v \mapsto (v_a, v_b)\). Notice that \(r\) remains well-defined and is invariant under the projection. Also note that \(r\) maintains the same subspace property as above. That the projected space contains two vectors with distinct values of \(r\) immediately implies that the projected space is at least two-dimensional; thus it is in fact the entirety of \(C^2\). The subset of \(C^2\) with \(|r| = 1\) or undefined is smoothly parameterised by \((x + yi, (x + yi) \cdot e^{i\theta})\) for real parameters \(x, y, \theta\). Thus, this subset is parameterised by 3 real parameters in a vector space of dimension 4 over \(\mathbb{R}\). Extend this parameterisation into a smooth parameterisation of the vectors in \(V\) with \(|r| = 1\) or undefined by choosing a basis for the kernel of the projection onto \(C^2\), and extending it to a basis for \(V\). This smoothly parameterises the set of degenerate representations in \(V\) for \(g, a, b\) by \(2k - 1\) real parameters. By Sard’s theorem \([16]\), this set is a null set.

To complete the proof of the theorem, note that the set of all degenerate representations is the union of sets containing a point degeneracy or a generator degeneracy for particular sets of points and generators. A finite union of null sets is a null set.

**Proof of Theorem 4.1.**

Given a permutation \(g \in G\) and points \(a, b \in M\), define the relative position of \(b\) with respect to \(a\) and \(g\) in a thin circle puzzle representation \(v \in V\) as \(r_{g,a,b}(v)\) or simply \(r(v) = \frac{v_b - v_a}{v_0 - v_g}\). This number can be thought of as the \(C\)-coordinate \(v_B\) when \(v\) is scaled and translated such that \(v_G = 0\) and \(v_B = 1\). Formally, \(r(v)\) can be infinite, but is undefined when \(v_G = v_a = v_B\). \(r\) has the useful property that a set of vectors with \(r(v)\) constant (or undefined) forms a vector space. Also note that \(v\) has a generator degeneracy if and only if \(|r(v)| = 1\) or is undefined.

Using this definition, the conditions in Theorem 4.1 can be restated equivalently as follows:

1. For all \(a, b \in M\), there exists \(v \in B\) with \(v_a \neq v_b\).
2. For all \(g \in G\), for each \(a, b \in M\) with \(g(a) \neq a\) and \(g(b) = b\), either there exists \(v \in B\) with \(r(v)\) defined and \(|r(v)| \neq 1\), or there exists \(v, w \in B\) with \(r(v), r(w)\) defined and \(r(v) \neq r(w)\).

First, verify that if either condition is not satisfied, then every representation in \(V\) is degenerate. If the first non-degeneracy condition is not satisfied, then there exist \(a, b \in M\) with \(v_a = v_B\) for all \(v \in B\). This property is preserved under linear combinations, implying that all representations \(v \in V\) have \(v_a = v_B\) and are therefore degenerate. Similarly, if the second non-degeneracy condition is not satisfied, then there exists \(g \in G\) and \(a, b \in M\) with \(g(a) \neq a\) and \(g(b) = b\) such that for some \(c \in C\) with \(|c| = 1\), \(r(v) = c\) or is undefined for all \(v \in B\). Likewise, this is because \(r\) is preserved under linear combinations.

Completing the proof in the other direction requires the following lemmas:

**Lemma:** Fix \(a, b \in M\). By Theorem 4.1, the set of representations \(v \in V\) with \(v_a = v_b\) is a null set.

**Proof:** If there exists \(v \in B\) with \(v_a \neq v_B\), then the set of vectors \(w\) with \(w_a = w_B\) must be a proper \(C\)-subspace of \(V\), and thus may also be viewed as a proper \(\mathbb{R}\)-subspace of \(V\). Proper subspaces of a 2\(k\)-dimensional \(\mathbb{R}\)-vector space are null sets with respect to 2\(k\)-dimensional measure.

**Lemma:** Fix \(g \in G\) and \(a, b \in M\) with \(g(a) \neq a\) and \(g(b) = b\). Under the conditions of Theorem 4.1, the set of representations \(v \in V\) with \(|r_{g,a,b}(v)| = 1\) or undefined is a null set.

**Proof:** First, note that there must exist at least one basis vector \(v\) with \(r(v)\) defined, as otherwise \(V\) would fail to meet the point non-degeneracy criterion for \(a\) and \(b\).

Suppose there exists some \(c \in C\) such that \(r(v) = c\) or is undefined for all basis vectors \(v \in B\). This puts \(B\) in the first case of the second bullet, hence \(|c| \neq 1\). Conversely, \(c\) must be finite otherwise one would have \(v_a = v_b = v_{g(a)}\) for every \(v \in B\), again violating the non-degeneracy criterion. Thus, the vectors with \(|r| = 1\) or undefined are just the representations with \(r\) undefined.

To complete the proof of the theorem, note that the set of all degenerate representations is the union of sets containing a point degeneracy or a generator degeneracy for particular sets of points and generators. A finite union of null sets is a null set.
From Untouchable 11 to Hazmat Cargo

Carl Hoff, Applied Materials

Untouchable 11 is a packing puzzle designed by Peter Grabarchuk. This paper describes Untouchable 11 and its ‘untouchable’ concept, and explores applying this concept to other hexomino packing puzzles. Every untouchable packing puzzle can be mapped to an equivalent conventional packing puzzle (in which pieces can touch), enabling the use of existing software tools for analysis. Exploring this puzzle space led to the creation of a new puzzle, Hazmat Cargo.

1 Introduction

Untouchable 11 is a packing puzzle consisting of eleven pieces based on the eleven possible unfoldings of a cube, which themselves are a subset of the 35 hexominoes\(^1\). The goal is to place all eleven pieces onto a board such that no pieces touch, even diagonally at corners. The pieces can be rotated and flipped, but must be placed orthogonally onto the grid of the board. The puzzle offers three challenges:

1. Easy (9×17 board).
2. Medium (10×15 board, Figure 1).
3. Hard (12×12 board).

This paper describes how this idea of ‘untouchable’ packings has spread to other puzzles, and ultimately led to a new design of mine, described in a later section.

1.1 History

Untouchable 11, designed by Peter Grabarchuk\(^2\), first appeared on the gaming website SmartKit.com\(^3\) which sponsored the development of the associated app. In October 2008, it was launched with a contest\(^4\) which gave a Smartkit t-shirt and the book Puzzles’ Express 3\(^5\) to the first person to solve all three challenges.

Figure 1. Screenshot of the medium (10×15) Untouchable 11 challenge.

---

\(^1\)http://mathworld.wolfram.com/Polyomino.html
\(^2\)http://www.grabarchukpuzzles.com
\(^3\)http://smart-kit.com
\(^4\)http://smart-kit.com/s1512
The concept of a polyomino packing puzzle, in which no two pieces can touch even at a corner, appears to be original to the Grabarchuk family. In his book *Polyominoes* [2], Solomon Golomb asks what is the minimum number of pentominoes that can be placed on an $8 \times 8$ checkerboard such that none of the remaining ones can be added. The answer is five, and Figure 2 shows one such configuration. This sparse covering of the board seems to be a precursor to Grabarchuk’s untouchable concept.

Kadon Enterprises, Inc. also has a few games using similar concepts. Squint, a logic game played on a $9 \times 12$ grid, using their Quintillions set (1980). The goal is to make the last move by leaving no space on the grid for the opponent to place another quint (their brand name for pentomino).

Players in turn select a quint from the common pool and place it on the grid. The first quint must cover one of the board’s corner squares. Later quints must be placed so that at least one of their corner points touches a corner point of any of the quints already on the board, and no sides may touch. Figure 3 shows such an arrangement.

This rule that corners must touch and sides may not touch results in a similarly sparse covering of the board. It also appears in the well-known game Blokus (2000) as a restriction on each player’s own pieces.

Cornered is a similar logic game played using the Sextillions set. In that game, the pieces (the 35 hexominoes plus one duplicate) are divided between two players. In turn, players select one of their own pieces and place it on a $15 \times 15$ grid. A player’s own pieces may not touch each other, not even diagonally at corners. A piece may touch opponent’s pieces only at corners (no sides), but are not required to touch. The last player to put a piece on the board wins.

The only other puzzle I am aware of which uses the eleven unfoldings of a cube is a puzzle Kate Jones presented as her exchange gift at the 11th Gathering for Gardner. She named this puzzle 11 Magic Cubes. Other than using the same pieces, it bears little resemblance to Unravelable 11.

## 2 Solving

In 2008, I solved the easy and medium challenges by hand. After days of struggling with the hard challenge, the closest I came to solving it is shown in Figure 4.

At this point, Peter was contacted and asked if the solution was unique. It turned out that the initial challenges were solved by Grabarchuk family members without the aid of computer algorithms. Peter knew of only two solutions to the hard challenge, and the total number of solutions was an unknown at that time. So now there were two puzzles to solve: I still needed to solve the

---

5http://www.gamepuzzles.com

6http://www.gamepuzzles.com/g4g11cubes.pdf
hard challenge, and — more interestingly — to count the total number of solutions!

Unable to find a solver capable of solving these untouchable packing problems, I created my own, shown in Figure 5. Algorithms for solving packing puzzles typically use a recursive backtracking search [3]. Knuth describes how to efficiently implement this type of search in his paper ‘Dancing Links’ [4]. Matt Busche also has an article suggesting how to combine a number of relevant strategies and ideas, including those developed by de Bruijn [5] and Fletcher [6].

Figure 5. The author’s Untouchable 11 solver.

My Untouchable 11 solver uses several of these strategies. The source code is in Quick Basic 4.5 and is available. The code works and found all seven solutions to the hard challenge of Untouchable 11, but it took over 24 days to complete its search. The output of that initial search is available, but be warned that it contains solutions.

However, before the 24-day search was completed, it became apparent that the puzzle could be mapped to a conventional (touching) packing puzzle. This would allow the use of many other existing solvers which are much more efficient.

The fastest of the polyomino solvers that were readily available in 2008 was Gerard Putter’s Polyomino Solver [10]. Once the hard challenge was mapped to its conventional touching equivalent and fed into this solver, the seven solutions were all found in under an hour. This work was completed before my 24-day search finished running.

The idea is to map each original piece to a new piece defined by squares centred at vertices of the original piece, and increasing the width and height of the playing grid by one square. Figure 6 shows the original 12×12 challenge viewed this way: an equivalent task is to place the vertices onto the 13×13 grid of vertices. This results in exactly fifteen empty vertices.

In effect, this thickens each piece by wrapping it in an additional half-square wide layer. This additional part of each piece neatly fits into the required gaps between pieces in the original version of the puzzle. Each resulting piece is one square higher and one square wider. Figure 7 shows how two original pieces become two touching thicker pieces under this mapping.

Figure 6. Mapping to a touching packing puzzle.

Figure 7. Half-unit thickening of pieces.

The fastest of the polyomino solvers that were readily available in 2008 was Gerard Putter’s Polyomino Solver [10]. Once the hard challenge was mapped to its conventional touching equivalent and fed into this solver, the seven solutions were all found in under an hour. This work was completed before my 24-day search finished running.

Figure 8. Result from Gerard’s Polyomino Solver.

---

7http://www.mattbusche.org/blog/article/polycube
8http://wwwmww.com/gapd/Untouch.TXT
9http://wwwmww.com/gapd/SOLUTION_Finished.TXT
10https://gp.home.xs4all.nl/PolyominoSolver/downloadsolver.htm
The latter results confirmed the count and solutions found with Gerard’s solver. Figure 8 shows output from Gerard’s solver for the medium challenge. (We will not spoil the solution to the hard challenge here!) It found 482,482 solutions in 104,334 seconds (roughly 29 hours).

3 New Challenges

With a general solver, the first space to explore was additional rectangular boards as new challenges for these eleven original pieces. Table 1 shows these results. The ‘Empty’ column gives the number of empty cells in the mapped version, i.e. number of untouched vertices in the original version.

<table>
<thead>
<tr>
<th>Board</th>
<th>Solutions</th>
<th>Name</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>12x12</td>
<td>7</td>
<td>Hard</td>
<td>15</td>
</tr>
<tr>
<td>11x13</td>
<td>33</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>10x15</td>
<td>482,482</td>
<td>Medium</td>
<td>22</td>
</tr>
<tr>
<td>9x16</td>
<td>174</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>9x17</td>
<td>65,516,235</td>
<td>Easy</td>
<td>26</td>
</tr>
<tr>
<td>8x18</td>
<td>15</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>7x21</td>
<td>60,327</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>6x24</td>
<td>8</td>
<td></td>
<td>21</td>
</tr>
</tbody>
</table>

Table 1. Solution counts for Untouchable 11 challenges.

Five new challenges were found that all fall between the medium and hard challenges in terms of difficulty. It was also proven that one entire row of the easy challenge, the 9x17 board, could be left empty, because the 9x16 board is solvable. Untouchable 11 now consisted of eight total challenges and received the Gamepuzzles Annual Polyomino Excellence Award for 2015. Figure 9 shows the trophy.

A physical version of Untouchable 11 was created as the author’s exchange puzzle for the 2017 International Puzzle Party (IPP37) in Paris, France. This puzzle included all eight challenges. The pieces were made of laser-cut acrylic by Sculpteo. The board was 3D printed in Polyamide using selective laser sintering, SLS, by i.Materialise. Figure 10 shows the final product.

Figure 10 does not show a solution, as two pieces touch at corners. A state with a single corner touch is known as a near-solution. These were counted for the original Untouchable 11 hard challenge in November, 2016, and 3,092 near solutions were found. This count was later verified by Landon Kryger in December 2016.

4 Widening the Search

The search for a set of eleven hexominoes which can be placed on a 12x12 board with a single unique solution was started in 2012. That work was done by creating modified code for each subset and running it through Gerard Putter’s Polyomino Solver.
As each subset had to be coded by hand, this was slow tedious work, and the work was put on hold when a set with just two solutions was found. That set uses one hexomino which is not an unfolding of the cube. It was shared with Peter Grabarchuk and resulted in the release of Untouchable 11: Master Challenge in March 2012, shown in Figure 11. This work was initially prompted by the need for an exchange gift for the 10th Gathering for Gardner, G4G10.

The search resumed late in 2016 with the assistance of programmers Brandon Enright and Landon Kryger. Landon had created a new, efficient solver which could test all possible subsets of a given size from a master set on a given board, to find puzzles with unique solutions.

---

Figure 11. Untouchable 11: Master Challenge.

Figure 12. The 35 hexominoes and their vertex duals.

14http://www.puzzles.com/PuzzleClub/Untouchable11MasterChallenge
The first thing to decide on was the master set that would be used: as shall be shown, there is no reason to include all 35 hexominoes, and a smaller set of candidates would mean a shorter search time. Figure 12 shows the complete set of 35 hexominoes and their vertex duals, created by mapping each vertex to a square, i.e. the thicker versions of each piece. 27 vertex duals have fourteen squares (shown in blue), but seven have thirteen squares (shown in green), and one has only twelve (yellow). We decided to use only the first 21 hexominoes as the master set. The hexominoes 22 through 35 were removed from consideration for the following reasons:

Hexominoes 22-35 have fewer than fourteen squares in their dual versions, so they seem easier to place. Hexominoes 22-35 can all be contained in a 3 × 3 or a 4 × 2 box. These are all more compact than the original eleven unfoldings of a cube, so they seem easier to place.

Hexominoes 22 and 23 both map to the same vertex dual polyomino. Any set containing both could never have a single solution, since those two pieces could always swap positions, so at least one must be excluded. Hexominoes 24 and 25 also both map to the same vertex dual polyomino. Hexomino 26 is unsuitable, as no vertex dual has a protruding square which could fit in the small gap on its right side. Therefore, any solution containing this piece produces a second solution with this piece rotated 180°.

After we selected the master set, the results shown in Table 2 were generated after many months of CPU time. We found seven sets of eleven hexominoes with unique solutions on the 12 × 12 board. Also note that there are seven sets of twelve hexominoes which also have unique solutions on the 12 × 12 board.

Table 3 shows all 11-piece and 12-piece sets with unique solutions. These are excellent puzzles left for the reader to solve. It may seem counter-intuitive, but the 12-piece sets are much easier to solve than the 11-piece sets. This is due to the availability of only a single empty node, which allows one to backtrack much sooner, thus simplifying the search.

Table 2. Summary of search results. N indicates number of pieces.

<table>
<thead>
<tr>
<th>Board</th>
<th>N Empty</th>
<th>Subsets Tested</th>
<th>Search%</th>
<th>Single%</th>
<th>0 Solns</th>
<th>1 Soln</th>
<th>&gt;1 Soln</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 × 6</td>
<td>2 8</td>
<td>210 210</td>
<td>100.0%</td>
<td>18.5714%</td>
<td>5</td>
<td>124</td>
<td>39</td>
</tr>
<tr>
<td>6 × 6</td>
<td>2 21</td>
<td>210 210</td>
<td>100.0%</td>
<td>0.0000%</td>
<td>0</td>
<td>0</td>
<td>210</td>
</tr>
<tr>
<td>6 × 6</td>
<td>3 7</td>
<td>1,330 1,330</td>
<td>100.0%</td>
<td>6.0902%</td>
<td>1,074</td>
<td>81</td>
<td>175</td>
</tr>
<tr>
<td>7 × 7</td>
<td>3 22</td>
<td>1,330 1,330</td>
<td>100.0%</td>
<td>0.0000%</td>
<td>0</td>
<td>0</td>
<td>1,330</td>
</tr>
<tr>
<td>7 × 7</td>
<td>4 8</td>
<td>5,985 5,985</td>
<td>100.0%</td>
<td>8.6717%</td>
<td>4,365</td>
<td>519</td>
<td>1,101</td>
</tr>
<tr>
<td>8 × 8</td>
<td>4 25</td>
<td>5,985 5,985</td>
<td>100.0%</td>
<td>0.0000%</td>
<td>0</td>
<td>0</td>
<td>5,985</td>
</tr>
<tr>
<td>8 × 8</td>
<td>5 11</td>
<td>20,349 20,349</td>
<td>100.0%</td>
<td>4.2017%</td>
<td>4,750</td>
<td>855</td>
<td>14,744</td>
</tr>
<tr>
<td>9 × 9</td>
<td>6 16</td>
<td>54,264 54,264</td>
<td>100.0%</td>
<td>0.0792%</td>
<td>199</td>
<td>43</td>
<td>54,022</td>
</tr>
<tr>
<td>9 × 9</td>
<td>7 2</td>
<td>116,280 116,280</td>
<td>100.0%</td>
<td>0.0439%</td>
<td>116,213</td>
<td>51</td>
<td>16</td>
</tr>
<tr>
<td>10 × 10</td>
<td>10 10</td>
<td>203,490 203,490</td>
<td>100.0%</td>
<td>5.0980%</td>
<td>79,601</td>
<td>10,374</td>
<td>113,515</td>
</tr>
<tr>
<td>11 × 11</td>
<td>11 11</td>
<td>293,930 293,930</td>
<td>100.0%</td>
<td>0.0003%</td>
<td>0</td>
<td>1</td>
<td>293,929</td>
</tr>
<tr>
<td>11 × 11</td>
<td>10 14</td>
<td>352,716 107,010</td>
<td>30.3%</td>
<td>1.9325%</td>
<td>100,236</td>
<td>2,068</td>
<td>4,706</td>
</tr>
<tr>
<td>12 × 12</td>
<td>12 12</td>
<td>352,716 352,716</td>
<td>100.0%</td>
<td>0.0020%</td>
<td>49</td>
<td>7</td>
<td>352,660</td>
</tr>
<tr>
<td>12 × 12</td>
<td>12 12</td>
<td>293,930 293,930</td>
<td>100.0%</td>
<td>0.0024%</td>
<td>293,920</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>13 × 13</td>
<td>13 13</td>
<td>116,280 116,280</td>
<td>100.0%</td>
<td>0.0000%</td>
<td>116,280</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14 × 14</td>
<td>14 14</td>
<td>20,349 20,349</td>
<td>100.0%</td>
<td>0.0000%</td>
<td>20,348</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. 11 (A–G) and 12 (a–g) piece hexomino sets.

<table>
<thead>
<tr>
<th>Set</th>
<th>Hexominoes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9 10 12 13 14 15 16 17 18 20 21</td>
</tr>
<tr>
<td>B</td>
<td>8 9 10 11 13 15 17 18 19 20 21</td>
</tr>
<tr>
<td>C</td>
<td>8 9 10 11 12 13 15 17 18 20 21</td>
</tr>
<tr>
<td>D</td>
<td>8 9 10 11 13 15 16 17 18 20 21</td>
</tr>
<tr>
<td>E</td>
<td>1 8 9 11 12 13 15 17 18 20 21</td>
</tr>
<tr>
<td>F</td>
<td>1 11 12 13 14 15 16 17 19 20 21</td>
</tr>
<tr>
<td>G</td>
<td>4 8 9 12 13 15 16 17 18 20 21</td>
</tr>
<tr>
<td>a</td>
<td>1 2 3 4 5 6 7 8 10 11 13 14</td>
</tr>
<tr>
<td>b</td>
<td>1 2 3 4 5 6 7 8 10 11 13 18 21</td>
</tr>
<tr>
<td>c</td>
<td>1 2 3 4 5 6 9 11 12 13 18 21</td>
</tr>
<tr>
<td>d</td>
<td>2 3 4 5 6 7 9 10 11 13 16 18</td>
</tr>
<tr>
<td>e</td>
<td>1 2 4 5 6 8 9 10 11 12 14 18</td>
</tr>
<tr>
<td>f</td>
<td>1 2 3 4 5 7 8 10 11 12 14 16</td>
</tr>
<tr>
<td>g</td>
<td>1 2 3 5 6 7 8 9 11 14 16 19</td>
</tr>
</tbody>
</table>

16http://www.mathpuzzle.com/eternity.html
17Figure derived from: http://www.archduke.org/eternity/solution/index.html
5 Open Questions

Here are two open hypotheses, neither of which have been proven:

1. The 9-piece set used in Hazmat Cargo is the only 9-piece subset of the hexominoes to have a single solution on the $11 \times 11$ board.

2. All other 9-piece subsets have multiple solutions on the $11 \times 11$ board; there are none with no solutions.

There are $\binom{35}{9} = 70,607,460$ possible 9-piece subsets of the 35 hexominoes. Of these, only 293,930 have been searched, i.e. only about 0.42%. The sets that have been searched contain the hardest-to-place pieces.

Since they all have solutions, it is believed that adding easier-to-place pieces to the mix will not result in sets without solutions, or other sets with just a single solution. Still, neither hypothesis can be asserted with certainty. Please contact the author if you are able to prove either hypothesis.

There is also the question of what fun and interesting puzzles may exist in the space of untouchable hexomino packing puzzles with rectangular boards. That is the next task slated for Kryger’s solver. If the piece sets are expanded to include other polyominoes and the board shapes are not restricted to just squares or rectangles, then there are even more possibilities.

6 Conclusion

While Hazmat Cargo did not win any awards at the design competition, it did receive numerous compliments, including the thematic barge and hazmat drums. Several commented that the physical design fit the untouchable concept perfectly. It was fun to design and took on a significantly different aesthetic than my previous designs.

Aside from the simple pleasure of designing a new puzzle, the lesson here is to take a new look at the puzzles you have enjoyed. In this case it was Peter Grabarchuk’s Untouchable 11, which introduced a new concept to polyomino packing puzzles. This concept proved to open a very vast and interesting area which proved worthy of exploration. Five new challenges were added to the original Untouchable 11 puzzle. The Untouchable 11: Master Challenge was created and resulted in a new app being released and enjoyed. And the exploration resulted in a very difficult 9-piece puzzle named Hazmat Cargo.

---

18http://www.solidworks.com/
19https://www.shapeways.com/
Acknowledgements

Thanks to Peter Grabarchuk for his efforts in creating Untouchable 11, his permission to use that puzzle as my exchange puzzle at IPP37, and his blessings on writing this article. Thanks to Brandon Enright and Landon Kryger for all the assistance they have provided. Using Gerard Putter’s Polyomino Solver alone, it would have taken me twenty years to search the 352,716 subsets needed for just the 11-piece 12 × 12 puzzle, while Kryger’s solver reduced that time to a couple of months. Thanks to Gerard Putter for sharing his polyomino solver and Jaap Scherphuis for sharing his polyform puzzle solver, both of which were used in this study, with Jaap adding new functionality to his solver at my request. Thanks to Kate Jones for providing the information on Squint and Cornered. Please contact me for solutions to challenges presented in this paper.

References


Carl Hoff is an epitaxy process service engineer for Applied Materials currently supporting Global Foundries. Research interests include twisty puzzle design and theory, packing puzzles, and puzzle rings.

Address: 108 North Broadway, Malta, NY, 12020-5200, USA.

Email: carl.n.hoff@gmail.com

https://www.jaapsch.net/puzzles/polysolver.htm
Ludoku: A Game Design Experiment

Cameron Browne, RIKEN Institute

This article provides a practical example of designing a game from scratch, using principles outlined in previous articles in this column: where to start, what to aim for, trouble-shooting the design, and how to evaluate the outcome. The resulting game, called Ludoku, is a Sudoku variant that simplifies the basic Sudoku design while introducing new strategies without adding undue rule complexity.

1 Introduction

In previous instalments of this Games Design Patterns column, I have tried to outline practices conducive to good game and puzzle design [3, 4, 9, 10, 11]. In this instalment, I put my own words into practice, to show how they may be applied to design a new puzzle game.

This article describes the game thus derived, called Ludoku, then goes on to summarise the design process, the game’s strengths and weaknesses, and the general success of the exercise.

1.1 Ludoku

Ludoku is a Japanese-style logic puzzle [6] derived from Sudoku [8]. Figure 1 shows a typical challenge with 17 starting hints. The complete rules for playing Ludoku are given in the following blue box.

Ludoku is played on a 9×9 square grid, with some hint values shown. The aim is to fill the grid with numbers 1..9 such that:

1. No number is repeated in any row.

2. No number is repeated in any column.

3. The diagonal neighbours of a number do not repeat that number or each other.

Rule 3, the local diagonal neighbourhood rule, is illustrated in Figure 2. Consider the region formed by the diagonal neighbours of the central cell with the value 4 (shaded). No other cell in this region can also contain a 4 (left), and no other cells in this region can contain repeated numbers of any value (right).

This exact Sudoku variant has not been proposed before to my knowledge. Nikoli, the proprietary owner of Sudoku and world’s foremost publisher of it and other Japanese logic puzzles, confirm that this design has no precedent that they know of.

2 Design Process

The design process that led to Ludoku followed the basic advice outlined in previous articles in the Game Design Patterns series, as follows.

---

1Private communication with Nikoli’s chief editor Yoshinao Anpuku.

2.1 Reinvent the Wheel

The article ‘Reinvent the Wheel’ suggests starting the design process with a known design that has proven to be good and then look for ways to modify it. This gives an entry point into the design space that is known to be fruitful.

I chose Sudoku as my starting point for this exercise, as Japanese-style logic puzzles are my favourite type of solitaire puzzle and Sudoku is the most widely known example of this genre. Figure 3 shows an example.

Figure 3. The ‘World’s Hardest Sudoku’.

The fact that so many Sudoku variants already exist suggests that this is a rich region of the game design search space, while the danger is that so many designers have already explored this region that it could be exhausted. Is there scope for yet another Sudoku variant with something new and meaningful to offer?

2.2 Explore the Design Space

Now that the starting point had been decided, I applied the strategy outlined in the article ‘Explore the Design Space’ and considered the degrees of freedom that could be modified.

The Sudoku Dragon web site lists some of the degrees of freedom that designers have modified over the years to create new Sudoku variants. In almost all cases, Sudoku variants add complexity to the rules (e.g. Diagonal Sudoku), grid design (e.g. Killer Sudoku) or both (e.g. Try), in order to introduce additional constraints and solution strategies. However, I wanted to go in the other direction and simplify the design.

I decided to explore variants on a plain 9×9 square grid with the usual 3×3 Sudoku sub-regions removed, as shown in Figure 4. This would simplify the design at least, and hark back to the puzzle’s origins as a Latin square.

![Figure 4. A plain 9×9 square grid.](http://www.sudokudragon.com/sudokuvariants.htm)

2.3 Make the Design Do the Work

So what Sudoku-based rules would such a simplified design support? According to the article ‘Make the Design do the Work’, the rules of a game should be as transparent and intuitive as possible, and flow naturally from the design of the equipment. Instead of giving the player many rules to remember, as few rules as possible should be defined and the design should enforce the rest.

There is not much to work with in a 9×9 square grid apart from orthogonal and diagonal adjacency. Sudoku makes good use of orthogonal adjacency in its row and column rules (Rules 1 and 2 above), so can diagonal adjacent be exploited in a similar way? The simplest such rule change, most in keeping with the existing rules, would be:

3. No number is repeated along any diagonal line.

However, it turns out that such fully diagonal Sudoku packings are not possible on a 9×9 square grid, as discussed in Appendix A. This is probably one reason that Diagonal Sudoku only involves two diagonal constraints across the full board between opposite pairs of corners. I therefore tried a reduced version of this rule:

3. The diagonal neighbours of a number do not repeat that number.

---

2http://www.sudokudragon.com/sudokuvariants.htm
This allows full packings on the $9 \times 9$ square grid, exploits diagonal adjacency and is consistent with the game’s other rules. Further, this rule effectively replaces the local $3 \times 3$ sub-grids in Sudoku with local $3 \times 3$ ‘X’-shaped regions, as highlighted in Figure 2, thus maintaining conceptual consistency with the original game at a more fundamental level. However, a problem with this new rule soon became apparent.

### 2.4 Bug or Feature?

The article ‘Bug or Feature?’ [10] promotes awareness of apparent bugs in designs that can be turned around to produce useful features. The problem with the rule listed immediately above is that it allows opposite diagonal neighbours of a cell to have the same number, such as the repeated value 5 in Figure 2 (right). This created some cognitive dissonance, as such repeated diagonals just looked wrong, and felt in violation of the spirit of the diagonal constraint. Each time it occurred, I had to mentally go back over the rules and confirm that it was indeed legal, disrupting the flow of the game.

The solution was to simply forbid such cases, as follows:

3. The diagonal neighbours of a number do not repeat that number or each other.

This rule change removed the problem in a consistent and elegant way without adding undue complexity, and introduced new strategies (see Appendices B.2.2, B.2.3 and B.2.5). Turning this bug into a feature was a clear improvement and gave the final rule set shown in Section 1.1.

### 2.5 Embed the Rules

The article ‘Embed the Rules’ [11] describes the benefits of having the design of a game’s equipment implicitly enforce its rules as much as possible, in order to simplify the rule set and make the design more poka-yoke (i.e. mistake-proof). It could be argued that this new Sudoku variant violates this principle by instead simplifying the equipment (by removing the sub-grids) and adapting the rules to suit.

However, note that little complexity is added to the game. The original Sudoku rule 3 (that no number is repeated in any $3 \times 3$ sub-grid) is simply replaced by the new rule 3 (that the diagonal neighbours of a number do not repeat that number or each other) and the original local Sudoku constraints ($3 \times 3$ sub-grids) are replaced by new local constraints ($3 \times 3$ ‘X’ regions). Further, given that the new diagonal rule implicitly exploits an additional property of the square grid – diagonal adjacency – I would argue that the new design embeds its rules in the equipment at least as much as the original Sudoku design.

Now that the equipment and rules of the new variant had been decided, the game required a name. I chose ‘Ludoku’ as a contraction of ‘Locally Diagonal Sudoku’, with the additional bonus that ludo is the Latin root for ‘play’.

### 3 Analysis

This section provides a brief analysis of Ludoku and how it differs from Sudoku.

#### 3.1 Distinguishing Features

The most obvious difference between Ludoku and Sudoku is the absence of the $3 \times 3$ sub-grids. These are instead effectively replaced with the implicit $3 \times 3$ ‘X’ regions due to the new diagonal neighbourhood rule.

Figure 5 shows the three basic region types in Ludoku: rows, columns and ‘X’ regions. It is worth distinguishing between global regions (i.e. rows and columns) that contain each of the numbers 1..9 when completed, and local regions (‘X’ regions) that will only contain five of the numbers 1..9 when completed.

![Figure 5. Row, column and ‘X’ regions.](image)

A key difference between Sudoku’s sub-grids and Ludoku’s ‘X’ regions is that no number may be repeated in a Sudoku sub-grid (Figure 6, left) while such formations do not necessarily violate the diagonal neighbourhood rule in Ludoku (Figure 6, right).
Ludoku’s ‘X’ regions provide weaker constraints for performing deductions than Sudoku’s sub-grids, which has implications for the game’s strategic depth. Note, however, that there are only nine 3×3 sub-grids in a Sudoku grid while there are 77 ‘X’ regions in a Ludoku grid, one centred on each cell minus the four corners (which are subsets of the ‘X’s at diagonally adjacent cells). Sudoku has $9 + 9 + 9 = 27$ constraint regions in total to work with while Ludoku has $9 + 9 + 77 = 95$. This far greater number of weaker constraints outweighs any potential loss.

Another feature that highlights the fundamental difference between these two games is that no Sudoku challenge can start with fewer than 17 hints and still remain uniquely deducible⁴ while there exist deducible 15-hint Ludoku challenges, as shown in Figure 7. There may exist deducible Ludoku challenges with even fewer hints; a complete search/analysis has not been done.

### 3.2 Strategic Depth and Deducibility

Ludoku allows most of the basic Sudoku strategies to be applied (except for those specific to the

3×3 sub-grids) plus the addition of several new strategies. Some of these are listed in Appendix sections B.1 and B.2, respectively. In terms of number of strategies, Ludoku could well be strategically deeper than Sudoku.

Importantly, Ludoku balances global constraints provided by the row and column regions and the local constraints provided by the ‘X’ regions. Such interaction between global and local constraints appears to be central to the success of many logic puzzles.

A logic puzzle is described as deducible if it can be solved by applying logical deductive steps to produce a unique solution [13]. Ludoku succeeds in allowing deducible challenges that are interesting to solve, much like Sudoku, using the strategies listed in Appendix B.

The greater number of regions – 95 as opposed to 27 – makes Ludoku harder than Sudoku in general, as players must remain vigilant over a greater number of potential deduction points throughout the game. This greater mental effort is reduced to a manageable level through the judicious use of relevant strategies that encapsulate the side-effects of the new constraints, but there is no denying that Ludoku is hard; the more difficult examples can take over an hour to solve.

Even the annotated 7×7 sample game listed in Appendix C requires knowledge of the relevant strategies and significant forward planning. For example, consider the sequence of deductive steps leading to the instantiation of the value 2 in Figure 36. This sequence relies on several different regions, both local and global, and apparently unrelated candidate values 4, 1 and 3 before the eventual 2 is instantiated.

This increased difficulty in Ludoku is both a blessing and a curse. Sudoku enthusiasts looking for a new challenge with novel strategies that will stretch their deductive skills might enjoy Ludoku, but it is unlikely that the average player looking for a mild diversion will persist with it.

### 3.3 Challenge Design

It is preferable to design Ludoku challenges with their starting hints in symmetrical patterns. Sudoku publisher Nikoli have long maintained that handcrafted challenges are superior to those generated algorithmically [14], and symmetric hint placement is an indicator of handcrafted design. Even when challenges are generated by computer, incorporating symmetry can help give the impression of handcrafted design [15]. Symmetric hint placement is especially important in Ludoku, as the absence of 3×3 sub-grids makes the starting hints the only way to give challenges structure.

³Proven by McQuire et al. [12] in a 7.1 million hour search performed over one year on a supercomputer cluster.
Figures 8, 9 and 10 show some templates for generating Ludoku challenges with symmetric hint patterns. Black cells indicate positions of starting hints, and the number in each empty cell shows the total number of starting hints that the cell shares a constraint region with, indicating the amount of deductive information available to each cell. If a cell shares constraint regions with eight or more starting hints, then its value can be immediately instantiated if those starting hints contain eight different numbers. Higher values indicate greater constraint, and in logic puzzles it is usually beneficial to focus on the point of most constraint at each step.

The different distributions of cell totals give each pattern a different character. The ‘diamond’ design shown in Figure 8 has a high-value cell at its centre with 12 shared starting hints that is likely to be immediately deducible, but the available information dissipates quickly the farther a cell is from the centre. Solving challenges based on this pattern would typically involve focusing on the centre then solving outwards.

The ‘asymptotes’ design shown in Figure 9, conversely, has a low-value centre cell surrounded by four high-value diagonal neighbours that would be sensible starting points.

The ‘circle’ design shown in Figure 10 is more rounded, so to speak, with a reasonably homogeneous distribution of cell totals over most of its area, apart from the outermost cells. This is the most pleasing design found so far, both aesthetically and in terms of deductive flow during solution of the challenges that it produces.

4 Generation

The Ludoku challenges shown in this paper, and printed throughout this issue, were generated algorithmically using the following approach:

1. Generate a random packing of numbers that satisfies the Ludoku region constraints.
2. Choose a starting hint set as follows (with equal probability):
   (a) A pre-defined pattern (Figure 1).
   (b) Iteratively removing hints in a symmetric pattern (Figure 24).
   (c) Iteratively removing single hints (Figure 7).
3. If the final hint pattern provides a deducible challenge, then:
   (a) Evaluate the challenge.
   (b) Store the challenge to file.
In steps 2(a) and 2(b), hint patterns were iteratively reduced as long as the challenge remained deducible using the strategies listed in Appendix B. This process generates around one deducible challenge per second per thread on a typical laptop computer.

Challenges were evaluated for quality by recording the sequence of strategies applied and applying the following calculation:

\[ \text{quality} = \text{variety} + \text{degree} - \text{help} \quad (1) \]

where \( \text{variety} \) is the number of times the player must apply a different strategy to progress in the solution (to encourage interplay between strategies), \( \text{degree} \) is the minimum number of times any one strategy is applied (to encourage the use of all strategies), and \( \text{help} \) is based on the number of starting hints (to reward fewer hints).

This quality estimate gives some indication of the strategic depth and difficulty of challenges, but does not always capture how truly difficult a challenge is for the human player. This measurement was used to indicate potentially interesting challenges, with high-scoring examples then being hand-tested for more accurate evaluation.

Note that the automated solution process was based entirely on the strategies listed in Appendix B and not the more thorough deductive search technique devised to solve and evaluate player difficulty for general deduction problems [13]. This is because the strategies implemented already encoded the key deductive steps that players could be expected to make for this game, and the challenges thus generated already proved difficult enough without considering higher levels of deductive embedding. ‘Easy’ 9\times9 challenges can still take 20-30 minutes while ‘hard’ challenges can take up to 1-2 hours to manually solve.

5 Conclusion

Previously described Game Design Patterns [3, 4, 10, 11] were applied to create the Ludoku deduction puzzle, an apparently novel Sudoku variant that simplifies the board design and introduces new strategies without adding undue rule complexity. Ludoku could be strategically deeper than Sudoku but its distribution of local constraints over the entire grid, rather than concentrated in just nine sub-grids, makes it harder work for players and more difficult to solve.

On the positive side, I find Ludoku to be an interesting puzzle that is absorbing and enjoyable – if challenging! – to solve. On the negative side, it will probably be too challenging for most players, and is in the end just another Sudoku variant. However, I am generally satisfied with the result of this game design experiment.

Acknowledgements

This work was conducted while the author was a member of the RIKEN Institute’s Advanced Intelligence Project (AIP).

References

Appendix A. There is No Fully Diagonal 9×9 Sudoku

The question raised in Section 2.3 is whether an n×n square grid can be filled with numbers 1...n such that no number is repeated in any row, column or along any diagonal line within the grid. Let us call such a packing a fully diagonal Sudoku packing. Two n×n square grids that allow such packings are sizes n = 5 and n = 7.

Figure 11. Fully diagonal 5×5 Sudoku packings.

Figure 12. Fully diagonal 7×7 Sudoku packings.

Appendix B. Solution Strategies

This appendix describes some key strategies for solving Ludoku challenges. A general rule of thumb for tackling logic problems can also save time: focus on the most constrained point at each step.

B.1 Regular Sudoku Strategies

The following basic Sudoku strategies also apply to Ludoku.

B.1.1 Eliminate by Region

When the value of a cell is known, then that value can be eliminated as a candidate from all other cells with which it shares a region. For example, the value 3 shown in Figure 13 can be eliminated from the other cells shown.

Figure 13. Remove candidate 3s from other cells.

This problem is equivalent to the n²-Queen Colouring Problem, for which it has been shown that it is not possible to superimpose nine differently coloured solutions to the n-Queens Problem on a 9×9 grid. Hence, it is not be possible to derive any fully diagonal 9×9 Sukoku challenges using the initial rule set outlined in Section 2.3.

---

4Vaˇsek Chv´atal, ‘Colouring the Queen Graphs’: http://users.encs.concordia.ca/chvatal/queengraphs.html
B.1.2 Instantiate by Cell
If the number of candidate values for a given cell has been reduced to a single possibility, then that value can be instantiated at that cell. For example, Figure 14 shows a cell that must be 9.

![Figure 14. The centre cell must be 9.](image)

B.1.3 Instantiate by Region
If a given value can only occur in one possible cell within a global region, i.e. row or column, then that value can be instantiated at that cell. For example, Figure 15 shows a case in which the cell marked ‘?’ within the row must take the value 4.

![Figure 15. The cell marked ‘?’ must be 4.](image)

B.1.4 Pairs of Pairs by Region
If the candidate sets for two cells within a region are reduced to the same two candidate values, then those values can be removed as candidates from all other cells within that region. For example, the row shown in Figure 16 has two cells reduced to candidates 4 and 5, hence these values can be eliminated from other cells in the region.

![Figure 16. 4 and 5 can be eliminated as shown.](image)

B.1.7 Cross-Elimination
If two candidate values can only occur at the same two positions in any two rows or columns, then that value can be eliminated from those positions in any other columns or rows.

For example, consider Figure 17 which shows the coverage of known 4s in this example. It may not seem that any other 4s can be immediately instantiated from here.

![Figure 17. Coverage of known 4s.](image)

However, 4s only occur on the same two rows (three and seven) of the two columns highlighted in Figure 18. The value 4 can therefore be eliminated from other cells along these two rows as shown, allowing another 4 to be instantiated.

![Figure 18. 4s can be eliminated from two rows.](image)
B.2 Ludoku-Specific Strategies

The following strategies, particular to Ludoku, are based on local diagonal relationships between cells. The rationale behind these is to eliminate candidate placements that would incorrectly eliminate neighbouring values from their respective row or column due to local diagonals.

B.2.1 1-Step Pairs

If a given value can only occur in two adjacent cells within a given row or column, then that value can be eliminated from diagonally adjacent cells, as shown in Figure 19.

![Figure 19. 5s can be eliminated.](image1)

B.2.2 2-Step Pairs

If a given value can only occur in two cells separated by one intervening cell within a given row or column, then that value can be eliminated from diagonally adjacent cells up to two steps away, as shown in Figure 20.

![Figure 20. 5s can be eliminated.](image2)

B.2.3 4-Step Pairs

If a given value can only occur in two cells separated by three intervening cells within a given row or column, then that value can be eliminated from diagonally adjacent cells exactly two steps away, as shown in Figure 21.

![Figure 21. 5s can be eliminated.](image3)

B.2.4 1-Step Triplets

If a given value can only occur in three consecutive cells within a given row or column, then that value can be eliminated from the common diagonally adjacent cells, as shown in Figure 22.

![Figure 22. 5s can be eliminated.](image4)

B.2.5 2-Step Triplets

If a given value can only occur in three cells within a given row or column, each separated by an intervening cell, then that value can be eliminated from diagonally adjacent cells exactly two steps away, as shown in Figure 23.

![Figure 23. 5s can be eliminated.](image5)
Appendix C. Worked Example

This appendix provides a worked example of a 7×7 Ludoku challenge (Figure 24) that shows most of the deductive strategies listed in Appendix B in action. Note that even though this challenge is smaller than the standard size, it is still quite difficult. Also note the constant interplay between local and global constraints in allowing deductions.

A 7 can immediately be instantiated by region, along the third row (Figure 25). Note that rows are numbered from bottom to top.

A 6 can then be instantiated along the seventh column (Figure 26).

The coverage of known 7s means that 7 can only occur in a 2-step pair along the fourth row (Figure 27). . .

. . . allowing two potential 7s to be eliminated from the sixth row and a further 7 to be instantiated (Figure 28).

A 6 can be instantiated once again along the seventh column (Figure 26).

Figure 24. Example 7×7 Ludoku challenge.

Figure 25. 7 can be instantiated.

Figure 26. 6 can be instantiated.

Figure 27. A 2-step pair of 7s.

Figure 28. 2-step pair elimination to give a 7.
This new 7 eliminates one of the 2-step pair to allow another 7 to be instantiated (Figure 29)... …which in turn allows a 7 to be instantiated on the fifth row (Figure 30). The final 7 can then be trivially instantiated on the seventh row.

Only two 5s can exist in the sixth column in a 1-step pair, allowing the elimination of candidate 5s from neighbouring cells (Figure 31). This produces another 1-step pair of 5, in the fifth column, which eliminates another neighbouring candidate 5 (Figure 32).

This allows a 5 to be instantiated on the sixth row (Figure 33). A similar process can be applied to deduce the positions of the remaining 5s (Figure 34).
Candidate 4s can then be reduced to a 2-step pair in the fifth row, which reduces candidate 4s to a 1-step pair in the third row, which eliminates a neighbouring candidate 4 above (Figure 35).

![Figure 35. 2-step pairs of 4s.](image)

The two cells circled in Figure 36 can then be reduced to candidates 1 and 3, and the cell thus triangulated must take the value 2.

![Figure 36. Instantiation due to pairs of pairs.](image)

This leads to the immediate instantiation of a nearby 2 (Figure 38)...

![Figure 37. Trivial instantiation of a 2.](image)

...and the deduction of another 2, through an elimination due to a 1-step pair (Figure 38).

![Figure 38. More involved deduction of a 2.](image)

And so on, until the final solution (Figure 39).

![Figure 39. Final solution.](image)
An ‘edit’ is a small, simple change in a configuration, such as changing a letter in a word or moving a toothpick in an arrangement of toothpicks. The goal of edit puzzles is to find a sequence of edits (possibly minimal) to change one configuration into another. This article examines various edit puzzles and defines a new type called Number Sentence Morphing. Dynamic programming is useful for solving edit puzzles and assessing their difficulty. We examine examples of the effects of placing additional conditions on the edits. Number Sentence Morphing puzzles can also be useful for elementary math courses, and they are used as an example of how to design an automatic puzzle synthesis system with algorithms based on combinatorial graphs.

1 Introduction

An edit puzzle is one in which the solver must change one configuration into another by a series of moves. A move in such puzzles is a minimal change, called an edit. A classic example of this is given in Figure 1, taken from the Toothpick Geometry website starting with a configuration of four $1 \times 1$ squares, the goal is to remove two toothpicks and leave only two squares.

Another well-known edit puzzle is Word Morphing, in which the player must transform a starting word into a goal word by repeatedly changing one letter, with the constraint that each intermediate step must itself be a word. For example, the player may be set the challenge of transforming the word ZOOS into CAGE. One possible solution is:

<table>
<thead>
<tr>
<th>ZOOS</th>
<th>MOOS</th>
<th>MOOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOOT</td>
<td>FORT</td>
<td>FART</td>
</tr>
<tr>
<td>CART</td>
<td>CARE</td>
<td>CAGE</td>
</tr>
</tbody>
</table>

A word morph is scored by the number of edits used, with a goal of finding a small number of edits. The example used eight moves – but someone with a good vocabulary might succeed with fewer. A teacher might also insist that the intermediate words be polite (disallowing ‘FART’), or require that all words come from a class vocabulary list (a severe limitation which would make creating an interesting puzzle more challenging).

In some edit puzzles the goal is to minimise the number of moves, while in others it is to find any successful edit sequence. The toothpick puzzle is an example of the latter. The puzzle statement requires that only two moves be used, so there is only success or failure, not a score.

This article connects the construction and solution of edit puzzles to earlier articles on graph algorithms and their place in puzzle design, presents examples of edit games, and introduces an edit game for elementary math students.

2 The Mathematics of Edit Games

Two things are needed to define a class of edit puzzles: a set of legal configurations and a set of edit operations. Specific constraints on configurations are part of the specification. For example, in the Word Morphing example, only English words are allowed, not arbitrary strings of letters. This

---

1http://math.sfsu.edu/cm2/papers/Toothpickgeometry.pdf

makes the puzzle more difficult, and it also tests the player’s vocabulary. For the toothpick puzzle, every configuration which you can reach by removing toothpicks from the original configuration is allowed. The edit moves are removals of one toothpick. Edit games can be represented and studied mathematically, using graph theory.

Figure 2. Part of word graph for ZOOS to CAGE.

An article on the use of graphs in game and puzzle design is available in [1] while a graph algorithm for solving or designing edit puzzles is explained in [2]. Both these articles appear in earlier editions of Game and Puzzle Design. Briefly, a graph \( G \) is a collection of objects or vertices, denoted \( V(G) \) (or simply \( V \)) and a set of connections between pairs of objects called edges, denoted \( E(G) \) (or \( E \)). For an edit game, the vertices are the allowed configurations, and the edges are pairs of configurations such that an edit transforms one into the other. A part of the graph for the word morph puzzle example is shown in Figure 2. Note that the complete graph would have hundreds or thousands of vertices, representing all 4-letter English words reachable from ZOOS.

The partial graph in Figure 2 shows that we can find another path to solve the example word morph puzzle, avoiding the rude word:  

\[
\text{ZOOS} \to \text{ZOO} \to \text{ZOT} \to \text{GOT} \to \text{GO} \to \text{AGO} \to \text{AGE} \to \text{CAGE}.
\]

This involved seven, rather than eight, edit moves. Adding new types of moves increases the number of edges in the graph and, on average, reduces the edge distance between pairs of vertices.

If we change the editing rule, then the graph changes considerably. In the Word Morphing example, the only edits are changing letters. Suppose, instead, that an edit could also insert or delete a letter. Then a shorter solution (with no rude words!) is:  

\[
\text{ZOOS} \to \text{ZOO} \to \text{ZOT} \to \text{GOT} \to \text{GO} \to \text{AGO} \to \text{AGE} \to \text{CAGE}.
\]

This involved seven, rather than eight, edit moves. Adding new types of moves increases the number of edges in the graph and, on average, reduces the edge distance between pairs of vertices.

In fact, if we define the distance between two vertices in a graph to be the number of edges in the shortest path leading from one vertex to another, then this makes a graph into a metric space [3]. A metric space is a set in which we have a reasonable notion of distance between any two points. The rules for a metric space are simple:

1. The distance from an object to itself is zero,
2. The distance between distinct objects is positive,
3. The distance from \( a \) to \( b \) and \( b \) to \( a \) is the same, and
4. The distance from \( a \) to \( b \) plus the distance from \( b \) to \( c \) is at least as far as the distance from \( a \) to \( c \).

In edit games, the use of shortest paths to define distance corresponds to the notion smallest number of moves, while a problem can be solved if there exists any path between them, no matter how long. The fact that edit puzzles like Word Morphing generate a metric space suffices to show that the graph algorithms in [2] work for designing and solving edit puzzles.

The distance metric for changing letters in words are important enough to have its own name: the Hamming distance. Similarly, the metric for adding, deleting, or substituting letters is called the edit distance or Levenshtein distance. These metrics are explored in detail in [4].
3 Further Examples of Edit Games

The toothpick puzzle in Figure 1 had removing toothpicks as its edit moves. The puzzle in Figure 3 requires the player to move toothpicks instead. This is a different edit move; a puzzle could also allow both types of edit. The goal is to move three toothpicks and leave exactly three squares. This puzzle is made slightly harder by a human desire to keep the outer edges polygonal. This is not a requirement of the puzzle – just a tendency embedded in many people's thinking – and solving the puzzle requires discarding this assumption.

One challenge is to remove three toothpicks and leave seven triangles. Another is to remove three toothpicks and leave six triangles. Another is to remove four toothpicks and leave five triangles. And can you remove four toothpicks and leave four triangles? This puzzle has the added constraint that the outside toothpicks (forming the single large triangle) cannot be removed. This takes the unconscious assumption which may hinder people solving Figure 3's puzzle and makes it an explicit requirement.

The solutions to the puzzles given in Figure 4 are not unique. Designing these puzzles is made easier by exploiting the incidence relation between toothpicks and triangles, which involves noting which toothpicks are part of (or incident on) each triangle. Identifying each toothpick by number as shown in Figure 5, we can explicitly list each triangle’s toothpicks. The small triangles are:

{1, 2, 3}, {3, 5, 6}, {4, 5, 8}, {6, 7, 9},
{8, 11, 12}, {9, 13, 14}, {10, 11, 16},
{12, 13, 17}, {14, 15, 18}.

The medium triangles are:

{1, 2, 4, 7, 8, 9}, {4, 5, 10, 13, 16, 17},
{6, 7, 12, 15, 17, 18}.

The large triangle is:

{1, 2, 4, 7, 10, 15, 16, 17, 18}.

As an example of using the incidence relation, start with that list of thirteen triangles. Remove toothpicks 3, 11, and 14 by crossing out each triangle in which they appear. The result is this:
Removing those three toothpicks caused six of the thirteen triangles to disappear, leaving seven triangles. We can also see that the remaining triangles include only three small triangles but all of the larger ones. Using incidence relations this way demonstrates that your solution is correct, and this is a natural way of coding this sort of puzzle for software exploration.

Because the nine outer toothpicks cannot be removed, only nine inner toothpicks need to be considered. For puzzle configurations this small, using incidence relations is a feasible strategy for practical manual exploration. While this demonstration uses toothpick-triangle incidence, the notion of incidence may be defined for many pairs of types of objects.

Incidence structures are common in combinatorial mathematics. Graphs, for instance, can be specified by saying which edges are incident on what vertices. The most extensive use of incidence structures is in the theory of statistical designs [5]. Incidence structures can often serve as underlying data that permit the construction of algorithms for designing puzzles; the toothpick puzzles serve as an example of this principle.

3.1 Coin Puzzles

Close cousins to toothpick-moving puzzles are coin moving puzzles [2]. These puzzles start with coins on a flat surface, and the coins are either slid or moved. Sliding requires that a coin be moved along the flat surface without moving other coins, while moving permits lifting a coin and putting it down somewhere else.

Coin moving puzzles often exploit the natural packing of circles into a hexagonal lattice [6]. Figure 6 shows an example. The goal is to move two coins so as to transform the pyramid into a ring. There is an additional constraint that each move must be to a place touching two other coins.

A square lattice is less efficient at packing coins, but also works for such puzzles. Figure 7 shows an example, in which the goal is to slide two coins and transform the triangle into a square. Figure 7's puzzle can be solved by sliding, but Figure 6's puzzle requires picking up the second coin as sliding it would displace another coin.

The difference between sliding and moving is similar to the difference between substitution and addition/deletion/substitution in word morph puzzles. The number of paths through the configuration space for the coin puzzles is similarly larger for moving than for sliding. There are puzzles which can be solved by sliding and moving, but which require more slides than moves. Is it possible to solve Figure 6's puzzle by sliding coins, with the constraint that each slid coin must touch two other coins?

3.2 The Towers of Hanoi

The Towers of Hanoi is a famous puzzle involving three posts and a number of rings of distinct sizes [7]. A configuration is any arrangement of the rings on the posts, with the restriction that larger rings may not rest on smaller ones. An edit
move is to move a ring from one post to another. The goal is to move all rings from the first pole to the last. Figure 8 shows an initial configuration with five disks. The standard puzzle uses 64 rings and requires $18,446,744,073,709,551,615$ moves to solve. For $n$ rings, the minimum solution requires $2^n - 1$ moves.

Figure 8. Towers of Hanoi puzzle with five disks.

Figure 9 shows all nine possible configurations of the two-disk version of the puzzle. Links are shown between pairs which are connected by a single editing move. A minimal solution moves directly from the top configuration (both disks on the left post) to the lower left configuration (both disks on the right post) in three steps.

Figure 10 shows the underlying beautifully regular graph structure. Compare this with Figure 11 which shows the underlying graph structure for the three-disk version. Each additional disk added to the puzzle in effect transforms every vertex in the graph into a new triangle!

Figure 9. The nine possible configurations for two-disk Towers of Hanoi.

Figure 10. Graph for two-disk Towers of Hanoi.

Figure 11. Graph for three-disk Towers of Hanoi.
4 Number Sentence Morphing

We now present a novel puzzle called Number Sentence Morphing as a further example of an edit puzzle. A number sentence is a name used in elementary education for an equation using only the operations addition, subtraction, multiplication, and division, whole numbers, and possibly parentheses. If the expressions on both sides of the equals sign have the same value the number sentence is true, while if they do not it is false. Here are examples of a true and a false number sentence:

\[
3 + 4 \times 5 - 3 = 20 \\
3 + 4 \times 5 + 3 = 20
\]

Multiplication has higher precedence than addition or subtraction, so the first sentence is true while the second one is false.

A number sentence is well-formed if the expressions on both sides of the equals sign alternate numbers and operations, beginning and ending with a number. Starting with a false number sentence, the goal of the Number Sentence Morphing game is for individuals or teams to competitively edit the number sentence, attempting to minimise the difference between the values represented on either side of the equals sign. A formal statement of the rules if given in the blue box.

There may be a pedagogical value in having number sentences that cannot be edited into true number sentences. On the other hand, starting with a true number sentence and editing it is an easy way to select a starting number sentence and operations, beginning and ending with a number. Starting with a false number sentence, the goal of the Number Sentence Morphing game is for individuals or teams to competitively edit the number sentence, attempting to minimise the difference between the values represented on either side of the equals sign. A formal statement of the rules if given in the blue box.

There may be a pedagogical value in having number sentences that cannot be edited into true number sentences. On the other hand, starting with a true number sentence and editing it is an easy way to select a starting number sentence and operations, beginning and ending with a number. Starting with a false number sentence, the goal of the Number Sentence Morphing game is for individuals or teams to competitively edit the number sentence, attempting to minimise the difference between the values represented on either side of the equals sign. A formal statement of the rules if given in the blue box.

Rule 5 creates one of the scoring mechanisms, but it is also included to cause students to engage in a greater degree of reflection while playing. As we will see in the examples of play below, the difference between the two sides of the equals sign can vary greatly and so it is often difficult to keep the disagreement between the sides of the equals sign small. Strategically this means that progress toward ending the game often requires extreme disagreements along the way.

Rule 7 is potentially a trap. A team with a low difference between the sides of the equals sign may want to loiter to collect more points. This rule is more likely to be a trap the larger the number of teams is.

Rule 9 gives a potentially large reward for finishing first – it is intended to encourage progress toward the end of the game.

Rule 10 intentionally discourages true but boring trivial solutions such as \(3 = 3\). It encourages students to use as many of their symbols as they can. Ideally, students will achieve a perfect result, arriving at the solution with no objects remaining in their symbol pool.

The following small example shows a simple game for which four edits suffice to conclude the game. This example shows that the difference between the sides of the equation \((\Delta)\) can go up and down through orders of magnitude.
Table 1. A sample game of Number Sentence Morphing.

<table>
<thead>
<tr>
<th>Turn</th>
<th>Sentence Pool</th>
<th>Δ</th>
<th>Final score: 4+2-6=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3+4^2+5^3=20</td>
<td>+</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3+4^2+5^3=20</td>
<td>+</td>
<td>195</td>
</tr>
<tr>
<td>3</td>
<td>3+4^2+5^3=20</td>
<td>+</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3+4^2+5^3=20</td>
<td>+</td>
<td>180</td>
</tr>
<tr>
<td>5</td>
<td>3+4^2+5^3=20</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

4.1 Example Play

Table 1 shows a full example game with three players and the starting number sentence \(1 + 2^* \ 5 + 4 = 3 \times 6\). Each player has a current number sentence, symbol pool, \(\Delta\), and score so far. The game lasts 6 turns. (Turn 0 is the starting state.) Player 1 and Player 2 happen to make the same moves during the first four turns, while Player 3 diverges immediately. With 11 on the left, Player 2 wants to change their 6 on the right to 56, then to 5 + 6, which would finish with a perfect result – both a minimal difference and an empty symbol table. However, Player 1 ends the game before player 2 can finish this sequence. Player 1’s strategy backfires, however, as it ignores the large size of their symbol pool.

Player 1 earned 4 ‘smallest difference’ points, +2 points because no one else had the smallest difference at the end of the game, and −6 points for having 6 leftover symbols in their pool, for a total score of 0. Player 2 earned 3 ‘smallest difference’ points and -2 points for 2 leftover symbols, a total of 1. Player 3, who is not very close to a game ending state, nonetheless won with 3 points, thanks to having an empty symbol pool.

4.2 Manipulation of Number Sentences

The examples above were chosen and solved by hand. Let us now discuss the potential for puzzle design software. Number Sentence Morphing clearly belongs to the class of edit puzzles. The space of permitted configurations is defined by the well-formed number sentences that use the symbols in the initial number sentence. The edit rules are deletion and insertion of available symbols, making this still another set of editing moves from those that yield the Hamming and Levenshtein distances – the edit rules here do not permit substitution at all and restrict the set of symbols available for insertion to those in the edit pool. The edit pool started empty in the example, but a puzzle could start with symbols in the pool.

If we can specify the graph structure for Number Sentence Morphing in an algorithmically friendly manner, this will permit the use of graph algorithms as a design tool for locating good morphing starting points. A dynamic programming search of the space of configurations reachable by legal edits can determine the smallest reachable absolute difference, and the shortest edit-paths to reach that goal, and the values of \(\Delta\) each turn on those paths. This yields a rich set of information to help a designer select interesting puzzle instances.

Every number sentence transformation is reversible: if we can edit sentence \(P\) to create sentence \(Q\), then we can apply the reverse of that edit to turn \(Q\) back into \(P\). Therefore the graph representing Number Sentence Morphing has undirected edges. The adjacency relation (rules which specify whether two vertices are connected by an edge) can be defined as follows:

1. Delete a matched pair of parentheses.
2. Delete a digit adjacent to another digit.
3. Delete an operator between two digits, i.e. not adjacent to a parenthesis.

Figure 12 shows an example portion of a graph defined by these rules. Each vertex has information in the format: number sentence, symbol pool, \(\Delta\). From the top node \((2 \times (1 + 3) = 7)\) the shortest game-ending state, reachable in five moves, uses the well-formed but perhaps surprising technique of enclosing a single number in parenthesis. Parentheses denote precedence in the equation, and in the case of \((number)\) simply equate to the value of the number.
The Number Sentence Morphing game satisfies a number of elementary curriculum goals, including practising with the definition of a well-formed number sentence, practising basic arithmetic operations, practising problem solving, and engaging in strategic thinking. The scoring rules were specifically designed to create a moderately complicated strategic landscape and so encourage students or teams to engage in reflection and strategic reasoning.

The difficulty of a given Number Sentence Morphing puzzle can be set using dynamic programming [2] which can rapidly determine the shortest path to a game-ending configuration. The longer the minimal solution, the more difficult the game. In addition, having the values of $\Delta$ permits a designer to evaluate the degree to which the students must use strategy to overcome desire for short term gain in terms of score for having the lowest $\Delta$ in the current turn.

**4.3 Sample Challenges**

We end this section with some Number Sentence Morphing problems for the reader. Each of them has a minimum difference of zero and so can be turned into true sentences. These puzzles can be used in the manner described for team play or as solitaire puzzles.

3 $\times$ 4 = (12 + 8)
13 + 23 = 20 – 1
2 + 2 $\times$ 2 + 2 = 2 $\times$ 2 + 2
5 $\times$ 3 + 6 = 2 $\times$ (1 + 3 + 4)
(1 + 1 + 1 + 1) * (1 + 1 + 1) = (66)

**5 Characteristics of Edit Games**

**5.1 Competition, Solitaire, and Cooperative Games**

The examples of edit puzzles in Section 3 and Section 4 are solitaire puzzles while Number Sentence Morphing is structured as a competitive game, in which players (or teams) independently compete to solve the same puzzle. Clearly these boundaries are flexible: a Number Sentence Morphing challenge can be solved alone as a puzzle, and any solitaire puzzle can be solved cooperatively by a team, or in competition with others, either as a simple race or with some kind of scoring rules added.

**5.2 The Choice of Edit Moves Matters**

It is worth revisiting the importance of the exact choice of edit moves to make available when posting an edit puzzle. A clear demonstration of this
is the minimum number of edits needed to transform the binary string \texttt{01010101} into \texttt{10101010}. If only substitution is allowed, then each digit must be changed, requiring 8 edits. If, on the other hand, we are working with the Levenshtein operations (letting us also remove and add digits), then the minimum path consists of only two edits:

\[
01010101 \rightarrow 10101010
\]

An important rule of thumb is that the more neighbours of a configuration there are in the corresponding graph, the shorter paths tend to be. Adding additional editing moves to a puzzle makes the entire space of configurations more compact. The collection of possible editing moves on strings, geometric configurations, and other natural puzzle spaces is huge and a good place to let your imagination run free.

6 Conclusion

A relevant issue not dealt with in this article is the aesthetics of the puzzles. Dynamic programming could be used to enumerate the configurations for each of the edit puzzles and games shown here, but it ignores the degree of engagement or surprise the puzzle engenders. The first toothpick puzzle in Figure 1 subverts expectations by having the two squares in the answer be the large one and a small one; many of those attempting the puzzle might waste some time trying to find a solution that yielded two small squares. While not as elegant, the toothpick puzzle in Figure 3 requires that some toothpicks remain which are not part of any square – here the player is potentially betrayed by an unconscious assumption which is not actually a rule for the puzzle.

The coin-sliding puzzle in Figure 7 is made more puzzling by having the solution be a square rotated diagonally from the usual orientation for a square. The underlying graph for the Towers of Hanoi shows a beautiful regularity in the overall problem space that may speak to why this puzzle is a classic. The Towers of Hanoi puzzle is routinely used to teach both mathematical proof and recursion in computer science.

Word Morphing and Number Sentence Morphing lack geometric elegance and have fewer expectations available to subvert the puzzle player. Here the fun comes from the challenge of having a very large space of configurations to search, something the human brain is remarkably good at doing. While you can create Number Sentence Morphing puzzles by starting with a true number sentence and editing away from it, there is no guarantee – unless your number of edits is pretty small – that there will not be a shorter path to a game-ending configuration. There may be a shorter path to the end of the game you intended or there may be a different game ending configuration nearer than you supposed. For Number Sentence Morphing, a dynamic programming tool is a useful aid for validation of a problem, or at least your own understanding of it.

Acknowledgements

The authors would like to thank the Canadian Natural Sciences and Engineering Research Council, the University of Guelph and the University of Victoria for their support of this work.

References


Daniel Ashlock is a Professor of Mathematics at the University of Guelph, Canada. His research interests include applications of computational intelligence to both mathematical games and games played by people. Address: Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada. Email: dashlock@uoguelph.ca

Andrew McEachern is a Teaching Professor in the Department of Mathematics and Statistics at the University of Victoria, Canada. His research interests include mathematical education and biology, and applications of computational intelligence to games. Address: Department of Mathematics and Statistics, University of Victoria, British Columbia, Canada. Email: amceachern6@gmail.com
Measuring Drama in Snakes & Ladders

João Pedro Neto, University of Lisbon
Jorge Nuno Silva, University of Lisbon

Board games can be judged by different characteristics, which may be subjective in nature but nonetheless useful for describing a game’s dynamics. One is drama, introduced by J. M. Thompson in his article ‘Defining the Abstract’ [1], which is connected to the ability to recover from a weaker position. This article proposes several criteria to measure drama in the context of race games, specifically for several historical and modern variants of the traditional game of Snakes & Ladders (known as ‘Chutes and Ladders’ in the USA). Given the probabilistic nature of race games, these criteria are measured by simulating millions of matches and statistically analysing the results. The article includes a discussion of this analysis. This methodology can be used to measure drama in other board games with strong random elements, and can thus be a useful tool in game design.

1 Introduction

Every board game player has preferences, finding some games more appealing than others. What makes a game successful is not completely clear. Even among 100% random games, some seem more attractive than others. This raised a natural question: among games in which players only throw dice and move their pieces accordingly, what makes some of these games more interesting than others?

Drama seems to be the right answer. This concept, introduced by Thompson [1] in the context of abstract games, has been adapted by the present authors to some race games in [2]. The present article contains our effort to assess how much drama we find in various forms of the traditional game Snakes & Ladders. Figure 1 shows an Indian set from the 19th century.

Mathematical methods have been applied to race games. Markov chains were used by Daykin [3] and Althoen [4] to estimate the length of a typical Snakes & Ladders game. Seville [5] used Monte Carlo and Markov chain methods to study the playful character of Snakes & Ladders and the Game of Goose.

2 The Games

Snakes & Ladders is a traditional board game of Indian origin. Players randomly move along a track on a board which traditionally represents a personal spiritual path. The path is normally along a square grid, with some ladders and snakes connecting non-adjacent pairs of squares, marking promotions and demotions. In traditional versions, the cells are inscribed with text.

According to Topsfield [7, 8, 9], some versions of Snakes & Ladders – namely Jain, Hindu and Muslim – were played in India during the 18th and 19th centuries, and are believed to be several centuries older. This family of games evolved from previous versions, adapting its didactic mission and perhaps becoming more playable.

Although the number of cells may range from 72 to several hundred, most variants have similarly structured rectangular boards. This article studies six historically significant variants of Snakes & Ladders. Figure 2 shows a modern Milton Bradley edition with 100 squares, similar to a 1943 edition studied by Althoen [4].

---

1http://www.giochidelloca.it/
2Similar forward and backward loops were already present in the game Fifty-Eight Holes from the second millennium BC, and can be found even in the modern Monopoly (see [6, p. 59]).
Figure 2. A modern Milton Bradley edition (author’s collection).

Figure 3 shows a clearer abstract representation of this board. The track begins at square 1 and finishes at square 100. Ladders: (1, 38), (4, 14), (9, 31), (21, 42), (28, 84), (36, 44), (51, 67), (71, 91), (80, 100). Snakes: (16, 6), (47, 26), (49, 11), (56, 53), (62, 19), (64, 60), (87, 24), (91, 25), (95, 49), (97, 59).

Figure 4 shows a version considered by Daykin [3]. Ladders: (6, 23), (8, 30), (13, 47), (20, 39), (33, 70), (37, 75), (41, 62), (57, 83), (66, 89), (77, 96). Snakes: (27, 10), (55, 16), (61, 14), (69, 50), (79, 5), (81, 44), (87, 31), (91, 25), (95, 49), (97, 59).

Figure 5 shows a version considered by Daykin [3]. Ladders: (6, 23), (8, 30), (13, 47), (20, 39), (33, 70), (37, 75), (41, 62), (57, 83), (66, 89), (77, 96). Snakes: (27, 10), (55, 16), (61, 14), (69, 50), (79, 5), (81, 44), (87, 31), (91, 25), (95, 49), (97, 59).

Figure 6 shows a modern ‘Tantra’ variant with 84 squares mentioned by Parlett [11, p. 94]. Ladders: (14, 44), (47, 83), (50, 84), (58, 79). Snakes: (78, 52), (76, 2), (67, 16), (61, 42), (48, 10), (45, 9), (13, 8), (17, 1). Figure 7 shows a board cited in Topsfield [8, p. 82]. It is the Vaiṣṇava variant, which was popular in Rajasthan, the Punjab Hills, Uttar Pradesh and Nepal. Ladders: (10, 23), (17, 69), (20, 32), (22, 60), (27, 41), (28, 59), (37, 66), (45, 67), (46, 62), (54, 58). Snakes: (12, 8), (16, 4), (24, 7), (29, 6), (44, 9), (52, 35), (55, 2), (61, 13), (63, 3), (72, 51). Figure 8 shows a second historical board cited in Topsfield [8, p. 78]. It was used by the Jains of western India in the 18th and 20th centuries. Ladders: (7, 44), (38, 80), (44, 61), (47, 84), (65, 68). Snakes: (17, 1), (13, 8), (45, 9), (48, 10), (58, 21), (60, 41), (67, 23), (75, 2), (76, 52).
3 Measuring Drama

Drama, as defined by Thompson [1], is a subjective way to access game quality. We use a set of objective criteria intended to measure a game’s drama. In this article, we will use the criteria presented in [2] for the Game of Goose. These criteria all have a statistical nature, suitable for the random nature of Snakes & Ladders. This randomness suggests the use of probabilistic simulation to statistically estimate how strong each criterion is for each Snakes & Ladders variant. These statistical criteria arguably correspond to the subjective sense of drama, such that more dramatic games will have higher statistical values.

The first criterion, denoted win, reflects how balanced the winning percentages are. A game with more balanced wins is more interesting than a game in which some positions tend to win much more often than others. We wish to understand where each variant lies on the spectrum between a perfectly balanced game in which each player wins equally often and a game in which e.g. the first player always wins. We denote this by the win criterion. The win value is the standard deviation of the estimated wins for all player positions. (I.e. we consider the winning percentages of each player as the outcome of the same random variable.) A perfectly balanced game would have

The pieces start off the board. Two important rules need to be clarified: how pieces interact and how to define movement past the last square.

For piece interaction, there are three traditional options:

- **None**: There is no piece interaction. All pieces move independently, and may occupy the same square.
- **Stop**: A square can only hold one piece at a time. A player’s turn is cancelled if their piece would land on an occupied square.
- **Swap**: A square can only hold one piece at a time. If a player’s piece lands on an occupied square, then the non-moving piece is moved to the starting square of the moving piece.

There are three possibilities for the situation when a player’s die result would move their piece beyond the last square:

- **Stop**: The piece does not move.
- **Goose**: The piece bounces back from the last square and moves backward the remaining steps (as in the Game of Goose).
- **Loose**: The player wins the game.

It is not well-known which interaction and ending rules were traditionally used for the six historical variants, so we consider all possible combinations. We also measure these combinations for each modern Snakes & Ladders variant.

We define special cells to be all squares linked to another square by a snake or ladder. Different Snakes & Ladders variants have different special cell configurations. A major goal of this article is to explore whether these configurations are important for dramatic purposes.
win = 0, while a more unbalanced game will have higher variance, and so higher win values.

The next criterion, denoted lead, is the average number of rounds the winner has the lead just before winning. A small lead value implies more dramatic endings: it can often happen that a player behind the current leader surges ahead and seizes victory at the last moment. A large lead value typically means that the winner was already ahead for a long time, so there was no surprise in the final outcome. Thus a more dramatic game will have a lower lead value.

A related criterion is ANP, the average number of players who take the lead during the match. A dramatic match should have many different leaders, which adds to the enjoyment and interest of the game for all players. Low ANP values indicate few leaders, which makes the game dull.

However, very high ANP values might show a lack of decisiveness (see [12, p. 23]) and make the game feel too random or chaotic, potentially spoiling the experience. For a game with \( n \) players, good ANP values seem to lie around \( n/2 \).

4 Empirical Analysis

This section presents the main results of computing the drama criteria by simulation. For control purposes, we added a trivially simple variant board which we call ‘Abstract’, which has no snakes or ladders, but the same rules for piece interaction and ending as the other variants. This variant is useful for comparing the impact of special cells generally.

Compared to [2], the present analysis is more demanding, given the combinatorial complexity of applying all 9 combinations of the 3 interaction rules and the 3 ending rules, for each of 7 variants, i.e. \( 9 \times 7 = 63 \) rule and board combinations. Additionally, we test each variant with 3 to 6 players, thus requiring \( 4 \times 63 = 252 \) simulation runs.

4.1 Experimental Setup

Each one of the 252 variants was run 50,000 times to reduce statistical variance. Thus over 12 million games were simulated. The drama criteria statistics were computed for each distinct variant. We used the programming language R to code, simulate, and analyse the results.

4.2 Measuring Win

Figure 9 shows the computed win values for each variant’s simulation. Each data point represents the 50,000 runs of a given quadruple (variant, interaction rule, ending rule, number of players).

Higher values mean more unbalanced results, i.e. some players won more often than they should under the hypothesis of perfect balance. The \( y \)-axis is in log scale.

![Figure 9. Win values for each of 252 variants.](image)

Variant Abstract is clearly the most unbalanced for every number of players. This is evidence that the special cells are useful by allowing more opportunities for other players to win. This makes a game more balanced and, we argue, more dramatic.

Swapping piece interaction provides more balance (square variants in Figure 9), which is not surprising because it causes leading pieces to sometimes get moved back, providing a balancing effect. The goose ending rule (red variants in Figure 9) is often more balanced than other ending options, but there are too many exceptions to consider this a general rule. But combining the swap and goose rules (red squares in Figure 9) provides the best combination, in most cases, according to win values. Unsurprisingly, preventing interaction via the stop rule (triangles in Figure 9) works against drama: the leading pieces tend to keep their lead, sometimes blocking the advancement of other pieces.

Different numbers of players do not seem to significantly impact this measure. Figure 10 shows an example of win values for a rule combination (stop interaction and loose ending) with higher variability; even here, the variation is very mild. For a given board, the values are almost constant across various numbers of players. The only exception is Abstract, which slowly becomes more balanced as the number of players increases. Special cells much more effectively provide bal-

---

3Code, data, and further analysis available at https://github.com/jpneto/Snakes-Ladders
ance, and they maintain it across various numbers of players.

Figure 10. Win values for stop interaction and loose ending.

If there is some imbalance, which player has the advantage? As expected, the first player tends to win more games. Figure 11 compares Abstract and Althoen for all nine rule combinations. (The results are similar for other variants compared to Abstract.) The dashed line shows the ideal values for a perfectly balanced game. In Abstract, the first player is always more likely to win, even though the different interaction and ending rules influence the final percentages. In Althoen, the first player advantage still exists but the game is much more balanced. Althoen’s balance is also more consistent across various rule combinations.

Figure 11. First player win percentages for Abstract and Althoen.

However, even if the six historical Snakes & Ladders variants are more balanced than Abstract, are they sufficiently balanced? That is, are these win values sufficiently close to zero that they can be explained by random noise? Let us call this the uniformity hypothesis. We performed a test to check whether the values favour the uniformity hypothesis or reject it. We tested, for each variant, if the simulation values (namely, the highest win percentage minus the lowest percentage) were surprising against 1,000 runs of a uniform sampling of 50,000 elements (the number of simulation runs performed). We then computed at what percentile the simulation values lie on this multiple sampling. A high percentile means that the result is surprising, suggesting that it is a result of a non-uniform distribution (i.e. not a perfectly balanced game). A low percentile, on the other hand, suggests the result is a result of a uniform distribution (i.e. it is evidence of a perfectly balanced game).

Table 1 shows the 16 best results sorted by their probability of being biased. All other combinations’ values were higher than 90%. There are 212 games at percentile 1, which is strong evidence for non-balanced variants. There are 21 games with percentile larger than 95%, which marks typical statistical significance. This means that only a few of these Snakes & Ladders variants can be considered to be balanced games.

Table 1. The 16 most balanced variants.

<table>
<thead>
<tr>
<th>Game</th>
<th>Plyrs</th>
<th>Int</th>
<th>End</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daykin</td>
<td>6</td>
<td>swap</td>
<td>goose</td>
<td>0.25</td>
</tr>
<tr>
<td>Historical72</td>
<td>3</td>
<td>swap</td>
<td>loose</td>
<td>0.36</td>
</tr>
<tr>
<td>Daykin</td>
<td>6</td>
<td>swap</td>
<td>stop</td>
<td>0.42</td>
</tr>
<tr>
<td>Ayres</td>
<td>4</td>
<td>swap</td>
<td>goose</td>
<td>0.47</td>
</tr>
<tr>
<td>Daykin</td>
<td>4</td>
<td>swap</td>
<td>goose</td>
<td>0.47</td>
</tr>
<tr>
<td>Daykin</td>
<td>3</td>
<td>swap</td>
<td>loose</td>
<td>0.50</td>
</tr>
<tr>
<td>Daykin</td>
<td>3</td>
<td>swap</td>
<td>loose</td>
<td>0.54</td>
</tr>
<tr>
<td>Historical72</td>
<td>4</td>
<td>swap</td>
<td>stop</td>
<td>0.59</td>
</tr>
<tr>
<td>Daykin</td>
<td>4</td>
<td>none</td>
<td>goose</td>
<td>0.77</td>
</tr>
<tr>
<td>Althoen</td>
<td>3</td>
<td>swap</td>
<td>goose</td>
<td>0.83</td>
</tr>
<tr>
<td>Historical72</td>
<td>4</td>
<td>swap</td>
<td>loose</td>
<td>0.84</td>
</tr>
<tr>
<td>Ayres</td>
<td>3</td>
<td>swap</td>
<td>stop</td>
<td>0.85</td>
</tr>
<tr>
<td>Ayres</td>
<td>3</td>
<td>swap</td>
<td>loose</td>
<td>0.86</td>
</tr>
<tr>
<td>Historical84</td>
<td>3</td>
<td>swap</td>
<td>goose</td>
<td>0.88</td>
</tr>
<tr>
<td>Althoen</td>
<td>4</td>
<td>swap</td>
<td>goose</td>
<td>0.89</td>
</tr>
<tr>
<td>Daykin</td>
<td>5</td>
<td>swap</td>
<td>stop</td>
<td>0.89</td>
</tr>
</tbody>
</table>

4.3 Measuring Lead

The lead criterion is based on the premise that a game is more dramatic if the player who wins gained the lead only a few turns before the end of the game. Here we record who leads after each round, i.e. after all players make one move. The x-axis is the number of rounds in which the winner led before winning. Figure 12 shows these lead values for all variants with six players using the swap interaction rule. This particular subset of combinations is representative: the remaining variants have similar results.
For the lead criterion, we notice that there are important differences between rule variants and the special cells’ configuration. Variant Abstract shows a good lead value, having most of its winners ahead in only the final two rounds (≈ 20,000 from a total of 50,000 runs). The historical variants are very similar to Abstract, and are even more robust with respect to the ending rule. The other variants have a somewhat less dramatic ending than Abstract, but in most results – around 80% – the winner was ahead in only the final five or fewer rounds.

Of the ending rules, the goose variant shows the best results: the chance that the leader will be bounced back from the last square makes it more possible for another piece to take the lead. This effect is larger in Tantra, Althoen, and Ayres. The worst ending rule by the lead criterion is loose. It makes the endings of Althoen, Ayres, Daykin and especially Tantra, more predictable. In the Tantra variant with loose ending, the winner usually has the lead for six rounds before winning, making the ending less surprising endings and thus less dramatic.

Overall, with few exceptions, all tested variants showed adequate lead values, having a small average of consecutive winning rounds.

4.4 Measuring ANP

The ANP (average number of players) criterion is based on the premise that a game is more dramatic if more players take the lead sometime. A good ANP value should be somewhere in the middle between the extremes. A low value, near 1, implies that most games only have one leader the whole time, which is clearly boring and predictable, while a high value near \( n \), the number of players, seems too random and indecisive.

Figure 13 presents ANP values for six-player variants using the goose ending rule. Each board’s graph shows the number of matches having one to six players in the lead, so the y-values sum to 50,000 matches for each variant. Other ending rules produce similar results.

We see that all variants, except Daykin, behave as desired. Around 70% of the matches have the number of leaders in the ideal middle values. Variant Abstract already has this behaviour, which means this is already present in a pure race with no special cells. However, adding special cells makes these values more stable with respect to the interaction rule.

5 Measuring Game Length

Another important game characteristic is game length, the number of rounds it takes to end. This is not a measure of drama per se, but it reflects the idea that very short matches lack the opportunity for dramatic episodes and emotional investment, and very long matches lead to boredom. Figure 14 shows the mean length of a 6-player match for each of the 63 variants.

Rules related to swap interaction increase game length somewhat, since these rules makes pieces move backward. However, the overall effect is not large, and they are arguably worth it for the dramatic piece switches that they enable.

Of the ending rules, loose usually provides faster matches, unsurprisingly, since it makes winning easier for a player close to the last square. The other two ending rules (stop and goose) require a single specific die roll result to win. Those two rules give similar average game lengths.
A curious result is that Abstract causes longer matches than the real variants with special cells. This implies that the backward effect of snakes is smaller than the forward effect of ladders. An interesting project would be to find a special cell configuration for which these two effects are similar, cancelling each other out.

If we plot the histogram of all 50,000 game lengths, for each variant, we find another interesting property. The empirical distribution is asymmetrical with a right heavy tail. This makes it more convenient to display histograms with a log-normal axis.

We computed this log-normal fit and the Akaike information criterion (AIC), an Information Theory measure to find the average out-of-sample quality-of-fit statistic for a model [13]. The lower the value, the higher the model fit. Figure 15 shows the combination with the best fit, which was Daykin with three players using the none interaction rule and the goose ending rule. Most variants we tested show similar values.

Figure 16 shows the worst fit, which looks more like a normal distribution. It is six-player Abstract with none interaction and loose ending. Figure 17 plots AIC values for the log-normal fit of several variants for varying numbers of players, using swapping interaction. Abstract does not show meaningful differences across the number of players, even when pieces interact by swapping. This seems to indicate that most matches do not have enough interaction to change these values, which is corroborated with most matches having just one leader. For all the other variants there is a clear pattern: the AIC decreases with player count. This means that increasing player interaction together with these special cells configurations makes the game more predictable with respect to the log-normal model fit.

Figure 15. Log-normal fit of a 3-player Daykin.

Figure 16. Log-normal fit of a 6-player Abstract.

Figure 17. AIC quality-of-fit with swap.

6 Conclusion

This article presents a methodology to produce and compare statistical measures for Snakes &
Ladders and similar race games with randomness as their main feature. This makes them perfect candidates for statistical analysis, since computer simulations can quickly produce enough data to allow discovery of patterns.

In this work, the use of a control game, Abstract, is crucial for determining the origin of certain behaviours, i.e. whether they are produced by the pure race element, or as a consequence of special (snake and ladder) cells. And by experimenting with all possible combinations of piece interaction and ending rules, we can study their impact on game dynamics.

By visualising the data, we were able to reach several conclusions. Special cells are necessary to stabilise the winning percentages, making the games more balanced, especially by using the Game of Goose interaction and ending rules. Game balance is invariant for various numbers of players (up to six players). Nonetheless, statistical analysis found that most variants (232 of the 252 tested) are not balanced: there is a subtle advantage for the first player (subtle enough to not be noticed in actual play). Matches in which the winner led in the final rounds are very common, but this is not a result of having special cells, because the Abstract version already shows this behaviour. The number of different leaders in a match is adequate (not too few, not too many) in all variants, but special cells do seem to improve on the control variant. Game length seems robust to different rule combinations. There is also empirical evidence that Snakes & Ladders game length follows a log-normal distribution. The existence of special cells does not perturb this pattern, and even seems to help according to the results of quality-of-fit tests.

This methodology is flexible enough to be applied to other board games with strong random elements, i.e. where luck is at least as influential as skill. This is important for the game designing process, when one wants to tune the game parameters to guarantee a balanced game and to increase the probability of dramatic last moment come-from-behind wins, thereby improving the players’ gaming experience.

Acknowledgements

João Pedro Neto is a BioISI member and is supported by project UID/Multi/04046/2013.

References


João Pedro Neto is Professor of Computer Science at the University of Lisbon. He is interested in the study, design, collection, publication and play of abstract games, and helps organise a yearly national event for the dissemination of new abstract games between young students. Address: University of Lisbon, BioISI – Biosystems & Integrative Sciences Institute, Lisbon, Portugal. Email: jpneto@fc.ul.pt

Jorge Nuno Silva is a Professor at the Department of History and Philosophy of Science of the University of Lisbon. He is interested in games, mainly board games, and their technique, history and culture. Address: ULisboa, CIUHCT-Centro Intenuniversitário de História das Ciências e da Tecnologia, Lisbon, Portugal. Email: jnsilva@berkeley.edu
Dice Design Deserves Discourse

Sofiia Yermolaieva, Innopolis University
Joseph Alexander Brown, Innopolis University

Dice are not just game components. Their aesthetic design can influence a player’s emotions as well as speed in accurately reading the dice results. In order to examine the factors which positively contribute to a player’s experience, a set of fifty players were tested with dice categorised by four criteria: numeral type (digits, pips, symbols), colour, shape and size. We explored players’ subjective preferences among the various dice, as well as objectively measuring the time needed to perceive and mentally process the results of the dice. The tests discovered a disconnection between a player’s subjective feeling of which die is ‘best’ and the objective time needed for a player to understand the die result.

1 Introduction

Dice have a long history of use not only in games but in religious divination and other mystical and psychological activities, appearing in a variety of physical materials (stone, bone, ivory, wood, plastic, etc.) and forms. Many people still have superstitions about ‘lucky’ dice and firmly entrenched rituals about how to resolve a misrolled die (cocked, out of the dice tray, off the table, etc.). Many board games include dice with custom art on the faces or unusual colours.

There is little standardisation in the physical design of dice beyond the realm of gambling, in which standards are necessary to satisfy legal requirements and to ensure fairness (and of course even in casinos dice are a frequent object of superstition and emotion). Dice are evidently more than a mere practical mechanical randomisation tool, all the more so in the modern world where digital forms of randomness are often more accessible, cheaper, and better suited to the task of producing stochasticity. Note that as we are more interested in the players’ perception of the dice for the purposes of this study, we ignore issues related to dice fairness.

The design of dice clearly serves an emotional role in addition to a functional role. Norman [1] examines the design of many objects for both their functional and non-functional characteristics. His anecdotal findings are that there is a disconnection between objects which are functional and those which users enjoy, and that emotion often takes precedence over usefulness in purchasing decisions. In designing introductory board games, Steenon states:

A good gateway game should last somewhere between 45 and 90 minutes. A

game that takes longer may lose newer players, while anything shorter than 30 minutes may leave its players feeling like their purchase was not a good value. [2]

In addition, Nephew advises:

make sure [the board game] has as little downtime as possible. [3]

This implies that a useful die allows for faster game-play. However, our hypothesis is that other factors of the design besides efficiency will influence the users.

In this study, a set of seven dice were tested for their usability in a simple game with fifty users. The goal was to determine which of the design characteristics have an effect on their choice of the most functional die, which we define to be the one which requires the least time from the roll to making a move in the game. In this study, we investigate whether the most objectively usable die is the die which users prefer.

2 Methodology

In order to investigate this question, we conducted the series of experiments outlined below, using students of Innopolis University as subjects.

2.1 Experimental Design

Tests were conducted during two days in a meeting room with two chairs and a glass table. In order to examine different dice from the user perspective, two participants played each other in a game of modified Knock Out. Each player selects and announces their own ‘knockout number’ from 1 to 6.
In turn, players roll the die, until one of them rolls the knock-out number of the rival. In this case, they ‘knock out’ the rival player and score a point. After a knock-out, both players may change their knock-out numbers. The player with the greatest score wins at the end of the specified number of rolls, in our case seventy.

Figure 1 shows the seven dice which were given to the participants in a random order for ten rolls of each die during the game. This reduces both boredom and practice bias. Their scores within the game were tracked, and in order to ensure no bias, they were not told of the objective of the study; they were only told to play the game.

For each roll by a participant, we measured the time from the start of the roll (dropping the die from the hand) to the moment when the participant declared their rolled value to the other player. In order to ensure the accuracy of this timing and to reduce human measurement errors, the sessions were taped and the values were extracted from the video.

At the end of the session, the players were given a short questionnaire about their experience with the dice. Players were asked to decide what die was the best to use during the game and explain what influenced their choice.

2.2 Pips, Numerals and Symbols

To analyse effects of value representation, the results of three dice were used: Large Symbol Die, Large Numeral Die, and Large Pip Die in Figure 1. These dice have the same size, colour, and shape. They differ only in how they show values (symbols, numerals or pips).

2.3 Contrast

For the contrast difference analysis, two dice were used: Large Numeral Die and Large Contrast Numeral Die (Figure 1). They both have the same size, shape, and Arabic numeral representation of values. They have different colour contrast: one die has black numerals on a white background, while the other has silver numerals on a black slightly transparent background.

2.4 Shape

To analyse the user perception of dice shape, two dice were used: Medium Pip Rounded Die and Medium Pip Square Die (Figure 1). Both have black pips on a white background. Both dice have the same size, but their shape is slightly different. The Medium Pip Square Die has sharp edges and corners. The Medium Pip Rounded Die has rounded edges and corners.

2.5 Size

To compare size differences, three dice were used: Large Pip Die, Medium Pip Square Die, and Small Pip Die (Figure 1). These dice have the same colour contrast, shape, and pip representation of values. They differ noticeably in size.

3 Results and Discussion

Table 1 shows the mean and standard deviation for each user rolling each die. Tables 2 and 3 show the mean for all users of these values for each die.
<table>
<thead>
<tr>
<th>User</th>
<th>Symbol</th>
<th>Numerical</th>
<th>Contrast</th>
<th>Large</th>
<th>Rounded</th>
<th>Medium</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.78</td>
<td>2.95</td>
<td>3.3</td>
<td>2.79</td>
<td>3.01</td>
<td>4.53</td>
<td>3.18</td>
</tr>
<tr>
<td>2</td>
<td>2.75</td>
<td>2.84</td>
<td>3.03</td>
<td>2.78</td>
<td>3.05</td>
<td>4.49</td>
<td>3.32</td>
</tr>
<tr>
<td>3</td>
<td>2.75</td>
<td>3.25</td>
<td>2.81</td>
<td>2.78</td>
<td>3.13</td>
<td>4.52</td>
<td>3.27</td>
</tr>
<tr>
<td>4</td>
<td>2.74</td>
<td>3.4</td>
<td>2.78</td>
<td>2.7</td>
<td>3.27</td>
<td>4.49</td>
<td>3.46</td>
</tr>
<tr>
<td>5</td>
<td>2.9</td>
<td>3.59</td>
<td>2.83</td>
<td>2.73</td>
<td>3.27</td>
<td>3.68</td>
<td>3.5</td>
</tr>
<tr>
<td>6</td>
<td>3.03</td>
<td>3.62</td>
<td>2.74</td>
<td>2.8</td>
<td>3.43</td>
<td>3.22</td>
<td>3.49</td>
</tr>
<tr>
<td>7</td>
<td>2.93</td>
<td>3.58</td>
<td>2.76</td>
<td>2.8</td>
<td>3.46</td>
<td>3.08</td>
<td>3.6</td>
</tr>
<tr>
<td>8</td>
<td>2.98</td>
<td>3.71</td>
<td>2.78</td>
<td>2.82</td>
<td>3.4</td>
<td>2.96</td>
<td>3.54</td>
</tr>
<tr>
<td>9</td>
<td>2.94</td>
<td>3.81</td>
<td>2.79</td>
<td>2.84</td>
<td>3.72</td>
<td>2.97</td>
<td>3.47</td>
</tr>
<tr>
<td>10</td>
<td>2.97</td>
<td>3.81</td>
<td>2.79</td>
<td>2.79</td>
<td>3.74</td>
<td>2.60</td>
<td>3.52</td>
</tr>
<tr>
<td>11</td>
<td>2.93</td>
<td>3.93</td>
<td>2.78</td>
<td>2.81</td>
<td>3.76</td>
<td>2.52</td>
<td>3.59</td>
</tr>
<tr>
<td>12</td>
<td>2.9</td>
<td>3.85</td>
<td>2.76</td>
<td>2.79</td>
<td>3.74</td>
<td>2.44</td>
<td>3.6</td>
</tr>
<tr>
<td>13</td>
<td>2.86</td>
<td>3.47</td>
<td>2.76</td>
<td>2.78</td>
<td>3.74</td>
<td>2.39</td>
<td>3.68</td>
</tr>
<tr>
<td>14</td>
<td>2.84</td>
<td>3.24</td>
<td>2.66</td>
<td>2.8</td>
<td>3.74</td>
<td>2.33</td>
<td>3.45</td>
</tr>
<tr>
<td>15</td>
<td>2.68</td>
<td>3.25</td>
<td>2.54</td>
<td>2.79</td>
<td>3.78</td>
<td>2.26</td>
<td>3.42</td>
</tr>
<tr>
<td>16</td>
<td>2.65</td>
<td>3.25</td>
<td>2.50</td>
<td>2.69</td>
<td>3.51</td>
<td>2.18</td>
<td>3.56</td>
</tr>
<tr>
<td>17</td>
<td>3.11</td>
<td>3.2</td>
<td>2.54</td>
<td>2.62</td>
<td>3.45</td>
<td>2.12</td>
<td>3.37</td>
</tr>
<tr>
<td>18</td>
<td>3.06</td>
<td>3.15</td>
<td>2.5</td>
<td>2.54</td>
<td>3.43</td>
<td>2.18</td>
<td>3.42</td>
</tr>
<tr>
<td>19</td>
<td>3.02</td>
<td>2.99</td>
<td>2.44</td>
<td>2.45</td>
<td>3.08</td>
<td>2.13</td>
<td>3.33</td>
</tr>
<tr>
<td>20</td>
<td>2.97</td>
<td>2.98</td>
<td>2.37</td>
<td>2.48</td>
<td>2.97</td>
<td>2.2</td>
<td>3.26</td>
</tr>
<tr>
<td>21</td>
<td>2.91</td>
<td>2.8</td>
<td>2.29</td>
<td>2.46</td>
<td>2.92</td>
<td>2.26</td>
<td>3.2</td>
</tr>
<tr>
<td>22</td>
<td>2.85</td>
<td>2.78</td>
<td>2.29</td>
<td>2.38</td>
<td>2.81</td>
<td>2.49</td>
<td>3.05</td>
</tr>
<tr>
<td>23</td>
<td>2.92</td>
<td>2.79</td>
<td>2.31</td>
<td>2.37</td>
<td>2.71</td>
<td>2.52</td>
<td>2.99</td>
</tr>
<tr>
<td>24</td>
<td>2.95</td>
<td>2.79</td>
<td>2.37</td>
<td>2.47</td>
<td>2.56</td>
<td>2.68</td>
<td>2.98</td>
</tr>
<tr>
<td>25</td>
<td>3.09</td>
<td>2.54</td>
<td>2.41</td>
<td>2.42</td>
<td>2.42</td>
<td>2.75</td>
<td>2.97</td>
</tr>
<tr>
<td>26</td>
<td>3.04</td>
<td>2.57</td>
<td>2.46</td>
<td>2.41</td>
<td>2.4</td>
<td>2.73</td>
<td>2.86</td>
</tr>
<tr>
<td>27</td>
<td>2.57</td>
<td>2.58</td>
<td>2.43</td>
<td>2.44</td>
<td>2.37</td>
<td>2.72</td>
<td>2.83</td>
</tr>
<tr>
<td>28</td>
<td>2.65</td>
<td>2.56</td>
<td>2.47</td>
<td>2.51</td>
<td>2.46</td>
<td>2.69</td>
<td>2.77</td>
</tr>
<tr>
<td>29</td>
<td>2.86</td>
<td>2.54</td>
<td>2.56</td>
<td>2.52</td>
<td>2.38</td>
<td>2.68</td>
<td>2.85</td>
</tr>
<tr>
<td>30</td>
<td>2.95</td>
<td>2.53</td>
<td>2.51</td>
<td>2.48</td>
<td>2.41</td>
<td>2.56</td>
<td>2.9</td>
</tr>
<tr>
<td>31</td>
<td>3.04</td>
<td>2.54</td>
<td>2.56</td>
<td>2.45</td>
<td>2.41</td>
<td>2.57</td>
<td>2.88</td>
</tr>
<tr>
<td>32</td>
<td>3.08</td>
<td>2.63</td>
<td>2.63</td>
<td>2.52</td>
<td>2.49</td>
<td>2.36</td>
<td>2.89</td>
</tr>
<tr>
<td>33</td>
<td>3.06</td>
<td>2.64</td>
<td>2.64</td>
<td>2.55</td>
<td>2.56</td>
<td>2.37</td>
<td>2.87</td>
</tr>
<tr>
<td>34</td>
<td>3.05</td>
<td>2.76</td>
<td>2.64</td>
<td>2.44</td>
<td>2.66</td>
<td>2.27</td>
<td>2.95</td>
</tr>
<tr>
<td>35</td>
<td>2.94</td>
<td>2.74</td>
<td>2.64</td>
<td>2.51</td>
<td>2.8</td>
<td>2.29</td>
<td>2.9</td>
</tr>
<tr>
<td>36</td>
<td>2.9</td>
<td>2.82</td>
<td>2.67</td>
<td>2.53</td>
<td>2.97</td>
<td>2.36</td>
<td>2.96</td>
</tr>
<tr>
<td>37</td>
<td>3.03</td>
<td>2.9</td>
<td>2.63</td>
<td>2.57</td>
<td>3.1</td>
<td>2.39</td>
<td>2.98</td>
</tr>
<tr>
<td>38</td>
<td>2.92</td>
<td>2.86</td>
<td>2.62</td>
<td>2.47</td>
<td>3.25</td>
<td>2.43</td>
<td>3.05</td>
</tr>
<tr>
<td>39</td>
<td>2.88</td>
<td>2.87</td>
<td>2.61</td>
<td>2.42</td>
<td>3.36</td>
<td>2.43</td>
<td>3.03</td>
</tr>
<tr>
<td>40</td>
<td>2.75</td>
<td>2.95</td>
<td>2.66</td>
<td>2.44</td>
<td>3.49</td>
<td>2.51</td>
<td>3.01</td>
</tr>
<tr>
<td>41</td>
<td>2.73</td>
<td>3.04</td>
<td>2.72</td>
<td>2.46</td>
<td>3.55</td>
<td>2.57</td>
<td>3.07</td>
</tr>
<tr>
<td>42</td>
<td>2.7</td>
<td>2.96</td>
<td>2.62</td>
<td>2.44</td>
<td>3.57</td>
<td>2.56</td>
<td>2.98</td>
</tr>
<tr>
<td>43</td>
<td>2.63</td>
<td>2.92</td>
<td>2.62</td>
<td>2.37</td>
<td>3.65</td>
<td>2.54</td>
<td>3.02</td>
</tr>
<tr>
<td>44</td>
<td>2.58</td>
<td>2.91</td>
<td>2.73</td>
<td>2.36</td>
<td>3.68</td>
<td>2.47</td>
<td>2.99</td>
</tr>
<tr>
<td>45</td>
<td>2.57</td>
<td>3.0</td>
<td>2.64</td>
<td>2.28</td>
<td>3.64</td>
<td>2.41</td>
<td>2.99</td>
</tr>
<tr>
<td>46</td>
<td>2.58</td>
<td>2.96</td>
<td>2.64</td>
<td>2.31</td>
<td>3.45</td>
<td>2.38</td>
<td>2.98</td>
</tr>
<tr>
<td>47</td>
<td>2.58</td>
<td>2.91</td>
<td>2.68</td>
<td>2.28</td>
<td>3.44</td>
<td>2.38</td>
<td>2.96</td>
</tr>
<tr>
<td>48</td>
<td>2.81</td>
<td>2.89</td>
<td>2.65</td>
<td>2.29</td>
<td>3.27</td>
<td>2.29</td>
<td>2.86</td>
</tr>
<tr>
<td>49</td>
<td>2.79</td>
<td>2.89</td>
<td>2.63</td>
<td>2.33</td>
<td>3.27</td>
<td>2.23</td>
<td>2.88</td>
</tr>
<tr>
<td>50</td>
<td>2.8</td>
<td>2.9</td>
<td>2.65</td>
<td>2.29</td>
<td>3.25</td>
<td>2.22</td>
<td>2.88</td>
</tr>
</tbody>
</table>

Table 1. Mean time (in seconds) and Standard Deviation to roll each die for each user.
To determine whether a bias exists in the enjoyment of a die due to winning more with it, a Pearson correlation coefficient was examined. The calculated correlation coefficient is equal to -0.429747095, which shows that there is no significant correlation between number of wins and number of votes for each of the dice.

Based on mean roll time, Repeated Analysis of Variance (RANOVA) was performed. It consists of repeated measures on each die as the condition in order to find differences between mean time under seven conditions [4]. Figure 2 shows the RANOVA result; the p-value of 0 (i.e. extremely significant) shows a power of 1.

Using processed data from the fifty users’ evaluation cards, we calculated statistics about users’ choices for each die (Table 4). Table 5 shows the total votes for each of the four design characteristics among all dice. Using the survey on the design parameters, we performed an analysis of parameter influence on the choice of each die as the best (Table 6).

### 3.1 Pips v. Numerals v. Symbols

The values for the Large Symbol Die were the hardest to recognise for the users. Users 3, 5, 12, 13, 17, 20, 24, 25, and 38 each made at least one mistake when announcing their rolled number. Users 3, 5, 12, 13, and 24 mistakenly saw the circle of 6 stars as 5. These users corrected their own mistake after a few seconds. However, users 20, 25, and 38 also misread 6 on this symbol die as 5, and neither player noticed the error.

![Figure 2. Repeated Measures ANOVA calculated marginal means for the seven dice as conditions.](image)
Table 6. Analysis of parameters influence on the choice of each die as the best one.

| Parameters (number of Votes and Ticks) |
|-------------------------------|----------------|----------------|----------------|----------------|----------------|
| Die              | Size | Shape | Symbols | Colour Contrast |
| Symbol           | 2 of 2 | 1 of 2 | 2 of 2 | 2 of 2 |
| Numeral          | 4 of 6 | 3 of 6 | 5 of 6 | 0 of 6 |
| Contrast         | 2 of 3 | 1 of 3 | 1 of 3 | 2 of 3 |
| Large            | 8 of 9 | 3 of 9 | 5 of 9 | 2 of 9 |
| Rounded          | 14 of 14 | 8 of 14 | 7 of 14 | 8 of 14 |
| Medium           | 6 of 6 | 4 of 6 | 6 of 6 | 2 of 6 |
| Small            | 10 of 10 | 1 of 10 | 5 of 10 | 1 of 10 |

Figure 3 shows the similar circular arrangements of the 6 stars and the familiar 5 pips. Figure 4 shows the dissimilar arrangements of the ‘circle’ of 6 stars and the familiar ‘two straight lines’ of 6 pips.

Figure 5. Misleading large single star and 5 pips.

These mistakes by players could be explained by optical or visual illusion when players try to quickly recognise the value on an unfamiliar die. As Kuhn and Land explain in [6], the user’s perception of a die’s values was influenced by their past experience with classical (pip) dice: the user expects the star arrangements to be the same as the familiar pip arrangements.

User 17 similarly erred by reading 1 (a large star) as 5, and neither player noticed. Figure 5 shows the large star’s similarity with the more familiar 5 pips.

Another explanation could be the description similarity slip, which is an error connected with making right actions on the wrong object [7]. For example, a user rolls a new type of die, having the conceptual model of the previously used die. 18% of the users had conceptual activation errors recognising the value represented by a symbol on a die’s face.

58% of the test participants voted for the ‘Symbol’ parameter, which means that it has a rather high influence on their choice of the best die in the test. Out of all dice represented in this set, the Large Pip Die has the greatest number of votes — nine, in contrast with Large Symbol Die, which has two votes, and Large Numeral Die, which has six votes. Thirteen users wrote on their cards that Large Symbol Die is the worst, because of the stars used for value representation, but two people voted for that die as the most enjoyable one because of its uniqueness. One person mentioned Large Numeral Die as the worst one, because it uses digits instead of pips.

User 36 mentioned that there was no reason to underline the digit 6 on the Large Numeral Die because the die has no 9. However, underlining the 6 could prevent confusion in a game requiring more than one type of die. For example, in a game with a player rolling and summing two dice, a 6-sided and a 10-sided (as in Figure 6), the underlined values will be consistent on both dice, helping the player’s visual perception.

In a $t$-test with two-sided significance of $p = 0$, it was found that the Large Pip Die, which has the most votes, has less mean time to roll than the Large Numeral Die. With two-sided significance of $p = 0.006$, it has less value than the Large Symbol Die. RANOVA statistics show that the Large Pip Die has the least value of the difference between mean roll time.

There is a positive correlation between symbols, which were most recognisable for test participants, and mean time to roll the die (including the time to recognise the value of the die). The die voted best (Medium Pip rounded die), also uses pips. Seven of the 14 users who picked this die as the best, marked the ‘Symbols’ column in Table 6, which indicates that pips played a crucial role in their choice.

### 3.2 Contrast

Colour contrast influenced the decision of 30% of the test participants regarding the best die. Large Numeral Die with black digits on a white background has six votes for best die, and one vote for worst die. Large Contrast Numeral Die with a silver numeral on a black background has three votes for best, and one vote for worst.

Table 6 suggests that users voting for the Large Numeral Die were not influenced by the ‘Colour contrast’ design characteristic. The choices of two of the three users voting for the Large Contrast Numeral Die were influenced by the colour contrast parameter. The mean time to roll the Large Numeral Die was 3, compared with to 2.6 for the Large Contrast Die. The RANOVA measurements in Figure 2 show that the Large Contrast Numeral Die has the second smallest differences between mean roll time out of all users, while Large Numeral Die is fifth. The Large Contrast Numeral Die has 78 wins, while the Large Numeral Die has 77 wins. So Large Contrast Numeral Die required less time for users to throw and recognise the value, and also had more wins, yet Large Numeral Die received more votes.

Medium Pip Rounded Die received more votes (14) as the best die. It also has classical black pips on white, and eight of those fourteen voters considered this characteristic relevant in their voting.

### 3.3 Shape

38% of the users considered shape when deciding which die was best. Medium Pip Rounded Die was most popular and 8 of its 14 voters considered the rounded shape of the die in their decision. The Medium Pip Square Die has six votes, four of whom said die shape influenced their choice.

The Medium Pip Rounded Die’s mean time to roll was 3.2 seconds, while Medium Pip Square Die’s mean time was 2.67. The Medium Pip Square Die had a higher number of total wins by players, 77, while the Medium Pip Rounded Die had 70 wins.

Medium Pip Round Die took longer to roll, because of its shape. Figure 2 shows that it had the highest value of the difference between mean roll time by all users. It had one of the lowest numbers of wins. Yet it still received more votes.

### 3.4 Size

The size of the die was the most influential parameter on the users’ voting. Size influenced 88% of the participants. All users voting for the Small Pip Die (ten people) and Medium Pip Die (six people) mentioned size as influential. Size was considered by eight of the nine participants who voted for the Large Pip Die. Among the voters for whom size mattered, Small Pip Die received the most votes as best. Out of all seven dice, it also received the most votes as most enjoyable.

Users 1, 12, 32, 43, and 48 called Small Pip Die the cutest one. Users 5, 12 and 40 described it as the funniest one. Users 8 and 26 wrote that its small size enables Small Pip Die to maximise the randomness of the result, and it is hard to cheat because you can not fully control the die. User 20 wrote that this die is easier to throw, and it takes less time compared to the other dice to roll it.

But in fact, a $t$-test with two-sided significance of $p$-value equals to 0.275 shows that the mean time to roll the Large Pip Die has no significant difference with the mean time to roll the Medium Pip Square Die. In contrast, a $t$-test with two-sided significance of $p = 0.$ shows that the mean time to roll the Small Pip Die is greater than those of the Large Pip Die and Medium Pip Die. The RANOVA results show that the Large Pip Die has the least value of mean time difference, Medium Pip Square Die is third smallest, while the Small Pip Die has one of the highest values.
The Large Pip Die had the fewest wins (66) of all dice, the Medium Pip Square Die had among the highest wins (77), and the Small Pip Die led with 82 wins. So more users voted for the die with a higher mean roll time (sixth place by RANOVA results), and with the highest number of wins — Small Pip Die. All fourteen users who voted Medium Pip Rounded die as best mentioned size as influential. Note that the above results are taken from a small sample size, so should be regarded with some skepticism.

4 Conclusion

This study analysed dice based on four design characteristics and mean time to roll the dice by 50 users. From the results of the user survey, it was found that the medium-sized round-cornered die with values shown as black pips on a white background received the most user votes for usability. Nonetheless, RANOVA results showed that it had the greatest difference between the mean time to roll. This suggests that design characteristics influenced users’ preference more than objectively practical matters like time to roll and time to recognise a roll’s result.

Users’ choices were also analysed for correlation with winning. Two of the most popular dice also had higher numbers of wins, but the third most popular die had the fewest wins. Thus the ‘luck’ of the die may somewhat influence a user’s preference, but users also consider a die’s physical design. The error rate in reading a die’s result does not necessarily influence a user’s preference: some users liked the Symbol Die, despite readability problems with its facing showing values 1, 5 and 6.

Further study should be performed with a greater amount of participants in order to build a decision tree to find a correlation between specific die characteristics and the mean time needed to roll them. The set of design characteristics considered in this article should be tested on a larger number of dice with a variety of value symbols, colours, sizes, and shapes. Furthermore, the set of die characteristics should be expanded with physical properties such as weight, the sound of the die falling on the table, and material, to investigate design characteristics in even more detail. The height from which users throw the die could be considered when analysing the speed of a die’s roll. Influence of design characteristics on a die’s randomisation should be considered in future work, because in this study none of the participants mentioned this factor in the ‘Your option’ section of the survey card. Many participants noted that the dice falling on the glass table made them fearful of breaking it, so the rolling surface might also have an effect.

Acknowledgements

Thanks to the participants at Innopolis University, IBM for providing a student license for IBM SPSS Statistics Software which helped with the data analysis, and Elizabeth Reading for her valuable, expert advice with editing the paper.

References


Sofiia Yermolaieva has a B.Sc. in Computer Science from Innopolis University, and is currently a Senior IT Engineer at Sberbank Technology. Address: Innopolis University, Republic of Tatarstan, Russian Federation, 420500. Email: s.yermolaieva@innopolis.ru

Joseph Alexander Brown is Assistant Professor and Head of the Artificial Intelligence in Games Development Lab at Innopolis University, Russia, and Adjunct Professor of Computer Science at Brock University, Canada. His research interests include applications of procedural content generation to games and human factors in board games. Address: Innopolis University, Republic of Tatarstan, Russian Federation, 420500. Email: j.brown@innopolis.ru
Tension in Puzzles

Cameron Browne, RIKEN Institute

While tension is generally understood to be an important design feature for games, its exact meaning in this context remains poorly defined. This paper explores the meaning of tension in games and, by extension, in solitaire puzzles. A simple classification scheme for puzzles is presented to clarify the discussion.

1 Introduction

Tension is one of those key qualities that most designers know is important in a game, but which has no precise definition and can mean different things to different people. Tension is a familiar concept that plays a regular part in many aspects of our daily lives, but has a specific meaning in the context of games. This article aims to narrow down exactly what tension means in this context, to define it in a more concrete and measurable – if not formal – way, and to extend this concept to the notion of tension in puzzles.

2 Tension in Games

A lot of designers talk about tension and its importance for games, but I have not yet seen a precise definition of this term in this context. Authors seem to assume that it is a universal concept that the reader will understand, and which does not require further elaboration.

In his classic article ‘What Makes a Game Good?’ [1], veteran board game designer Wolfgang Kramer lists tension as one of the key factors in designing good games. While he does not specifically define his understanding of tension, he introduces the notion of tension curves that chart tension of a game throughout its course, as shown in Figures 1 and 2.

Figure 1 shows two games in which tension increases linearly, but game ‘A’ is preferable as it begins with an initial level of tension. Figure 2 shows two games with multiple tension peaks, in which in which game ‘A’ is again preferable as it has fewer tension peaks and less pronounced tension, both of which make the game less chaotic.

There seems to be a consensus that a well-designed game will follow cycles of crisis and relaxation, as local battles come to a head, are resolved, then build up to the next battle. Too much tension can be exhausting, while too little can be boring.

1 Personal communication.

In terms of video games, Rauers describes a number of ‘tension mechanics’ used to create tension in his game Card Thief without defining exactly what ‘tension’ means in this context [2]. Rose discusses the importance of tension and release in games, using a number of video games as examples [3]. Again, Rose does not define these terms, but does observe that ‘tension is the state of mental or emotional strain’ and that ‘choices need to result in tangible consequences’.

I believe that this last observation gets to the heart of the matter, and propose the following informal description of tension in games:

*Tension is related to the amount of impact that players’ choices have on the outcome of the game.*

If a game state has many available moves, none of which are significantly better or worse than any other, then that state is in a low state of tension. Conversely, a game state with many losing moves but only one winning move is in a high state of tension – assuming that the player realises this! – and the harder that winning move is to identify then the tenser things are.

For pure strategy games, tension is correlated to the amount of thought that players must invest in order to win the game. If a player can make random moves throughout a game and still win against an intelligent opponent then that game will have low tension, but if they must carefully plan each move then tension will be higher.

As an extreme example, consider a variant of Chess in which a six-sided die is rolled when either king is mated, and White wins if the number of pips shown is odd while Black wins if the number of pips shown is even. Such a game would be fair and balanced but have no tension until the final die roll, as no move by either player would have any real impact on the outcome until then.

For non-deterministic games with a chance element, such as race games in which movement is based on dice rolls, tension is correlated to players’ probabilities of winning. A game is not very tense if one player is far ahead on rolls and almost certain to win, but can become much tenser if the opponent catches up with a sequence of unlikely dice rolls. Even Snakes & Ladders can be tense if the outcome of a game comes down to a pivotal dice roll.

This leads to a simple metric for estimating the tension $T$ for a given game state $S$ as follows:

$$T(S) = 1 - \frac{M_{Sw}}{M_S}$$  \hspace{1cm} (1)

where $M_S$ is the number of moves available in state $S$ and $M_{Sw}$ is the number of winning moves available in state $S$. This equation is a simplification; the notion of ‘winning move’ is somewhat nebulous and depends on the type of game being modelled, and will require a more involved continuous distribution for probabilistic games.

3 Tension in Puzzles

So how can this notion of tension be applied to puzzles? We understand puzzles to be single-player games in which a player attempts to solve given challenges posed by a setter [4].

Just as games can be won or lost, so can puzzles be won by finding the solution or lost by failing to do so. It is useful here to distinguish between dynamic and static puzzles, but in both cases tension is again related to the degree to which choices made by the player affect the outcome.

3.1 Dynamic Puzzles

Dynamic puzzles are those in which changes to the game state are not permanent but can be further modified. Such puzzles typically involve pieces that move or information that can be modified or deleted. Moves are not guaranteed to progress the game state towards a solution, and can lead to cycles if states are repeated.

![Figure 3. Sokoban example from [5] showing solution states. The winning action sequence is \{D, D, R, U\}.](image-url)
Figure 3 shows a typical dynamic puzzle called Sokoban (Japanese for ‘warehouse worker’) in which the titular worker must push a number of boxes to cover target cells ■. This challenge, from [5], is solved by moving the single box: Down, Down, Right and Up.

### 3.1.1 States and Transitions

Figure 4 shows the situation leading up to the final push move in the Sokoban example above. Note that the worker has several paths by which they can be positioned for the final push; the choice of path is irrelevant, it is the push move itself that counts.

![Figure 4. Multiple paths to reach the final push.](image)

There may be several ways to get from one state to another, but it is convenient to collapse these into a single transition between these two states, as shown in Figure 5. The important thing is to know which states are achievable from which other states, not how they are achieved.

![Figure 5. State transitions can be collapsed.](image)

This allows the derivation of game trees for puzzles that enumerate all possible states and the dependencies between them. For example, Figure 6 shows the complete game tree for the above Sokoban example, with transitions labelled by action (U, D, L, R) and states denoted as follows:

1. A double circle ◊ denotes a solution state.
2. A solid circle ● denotes a dead-end state.
3. A crossed circle ⊕ denotes a key state that must be visited in order to reach a solution.
4. An empty circle ○ denotes any other state.

![Figure 6. Game tree for the Sokoban example.](image)

The winning sequence of actions \{D, D, R, U\} is highlighted in red, leading to this example’s solution ◊. Note that every action in this example leads to a key state ⊕ that must be visited in order to solve the challenge. From each state there is exactly one winning action – all other actions are non-winning – hence this challenge represents a case of maximum tension, as any mistake will make the challenge unsolvable.

Key states ⊕ allow the setter to exert authorial control over the solution process [4], by creating states that the solver must visit along the solution path and possibly also the order in which these must be visited. The initial state \(S_0\) of a challenge is always a key state.

### 3.1.2 Dynamic Puzzle Types

This method of describing the state spaces of puzzles allows a simple scheme for classifying dynamic puzzles, based on their basic game tree shape, as shown in Figure 7.
Dynamic puzzle challenges may be classified as *interesting* or *uninteresting*. Interesting challenges are well-formed and have non-trivial solutions, while uninteresting ones may be:

- **Invalid**: No solution can ever be reached from the initial state $S_0$.
- **Trivial**: A solution can be immediately and trivially reached from the initial state $S_0$.

Interesting challenges may be *monotonic* or *non-monotonic* in nature. Monotonic challenges are those in which the player can backtrack or undo moves to return to previously visited states. The ‘Non-Monotonic’ case in Figure 7 shows an example in which there are no dead-end states, and the player can reach the solution from all other states.

Monotonic cases may be further classified as:

- **Structured**: Key states beyond $S_0$ dictate states that the player must visit on the solution path.
- **Strict**: Every state on the solution path is a key state.

---

**Figure 7.** Basic types of dynamic puzzles, based on game tree shape.
Non-monotonic challenges allow the player to recover from mistakes, so can be 'friendlier' in nature and are a common design choice for children's puzzles. Note, however, that non-monotonicity does not imply simplicity; consider the Rubik’s cube!

Structured challenges allow more difficulty and authorial control through the existence of key states that the player must visit in order to reach a solution, but with the possibility of alternate paths to those solutions. Strict challenges are unforgiving as there is exactly one correct move from any state and any error is non-recoverable. Figure 3 is an example of a strict challenge. Note that non-monotonic challenges can also be structured in some cases, e.g. when the player has some backtracking freedom between key states, but can never be strict.

Key states allow the designer to add a deductive element to their puzzles, by defining certain key states that must be achieved before others can occur. For example, piece A can only be removed if piece B is moved, which requires piece C to be moved, which requires...

Tension in dynamic puzzles can be inversely related to the ratio of winning moves to available moves from each given state, as it is for games. Strict challenges will typically be the most tense.

3.2 Static Puzzles

Static puzzles are those in which changes to the game state are permanent. Such puzzles typically have a single unique solution, achieved by iteratively revealing information with each move.

3.2.1 Deducible Static Puzzles

Deducible static puzzles are those in which the solution can be iteratively deduced from the available information. Japanese logic puzzles [6], such as Sudoku and Slitherlink, are classic examples. There are no losing moves unless the player makes a bad deduction; all correct moves reveal information that progresses the challenge towards its solution. So what does ‘tension’ mean for such puzzles?

Figures 8 and 9 show two Slitherlink challenges, in which the aim is to complete a single closed path that follows the grid, such that each cell containing a hint number is visited by the path on that many sides. Each figure shows: (a) the initial challenge; (b) chains of obvious deductions that would typically be made by an experienced player; (c) the starting point(s) from which these deductions would be made; and (d) the solution path(s) that these chains of deduction would follow. The numbers in red indicate the number of deductions that can be expected from each starting point. See [7] for details.

Figure 8. A handcrafted Slitherlink challenge, simplifications, deduction points and solution path.

Figure 9. A computer-generated Slitherlink challenge, simplifications, deduction points and solution paths.
Figure 10 is a handcrafted challenge that shows a high degree of dependency \cite{4}, as there is a single starting point whose deduction reveals further information, which leads to further deductions that reveal further information, etc. The solution path is like a trail of bread crumbs that is drip-fed to the player. The single starting point eventually reveals enough information to solve the challenge, making it narrow and deep.

By contrast, Figure 9 is a computer-generated challenge that contains many starting points, each of which reveal some information towards the solution but not the complete solution; more complex deductions are required to progress beyond the obvious deductions (b). This challenge could be characterised as broad and shallow.

Since all moves in such puzzles progress the challenge towards its solution, tension is based not on the ratio of winning to non-winning moves but on the amount of available information. The handcrafted Slitherlink example starts off in a high state of tension as it only has a single starting point, and maintains this throughout its solution. The computer-generated example, on the other hand, starts off in a low state of tension as there are many different starting points, but this tension increases as the obvious simplifications are made and the information dries up.

Tension in deductive puzzles can be measured using the deductive search (DS) method \cite{7}, which solves challenges by iteratively applying cycles of increasingly complex deduction and simplification as more information is revealed. For example, Figure 10 shows a difficult Slitherlink challenge from \cite{8} in the process of being solved by DS. Progress has stalled due to lack of information and requires the solver to step to the next level of search in order to make the key deduction at the cell circled in order to progress. This challenge is currently in a state of high tension, but making this key deduction will reveal new information that allows further deductions and relaxes the state of tension.

Figure 11 shows the tension profile for this challenge, based on the number of deductions made per DS cycle. Note the cycles of tension and relaxation as deductions are made and new information is revealed, reminiscent of Kramer’s tension graphs for games shown in Figure 2.
3.2.2 Non-Deducible Static Puzzles

Non-deducible static puzzles are those in which the solution cannot be iteratively deduced from the available information. Such puzzles may have multiple solutions, but determining whether a given move leads to a solution or not would require a full game tree expansion, which is infeasible in most cases due to their complexity.

The puzzle game BoxOff is a typical example [9]. In BoxOff, the player randomly fills a square grid with coloured pieces, then successively removes pairs of piece in order to clear the grid. Each pair must be the same colour and the rectangle they form cannot contain any other pieces. For example, the two white pieces shown in Figure 12 can be removed but the two black pieces cannot (unless the white pieces are first removed). Even though pieces are removed, this game is still ‘static’ as moves cannot be reversed.

Figure 12. The white pieces can be removed, but block the black pieces from being removed.

The puzzle game BoxOff is a typical example [9]. In BoxOff, the player randomly fills a square grid with coloured pieces, then successively removes pairs of piece in order to clear the grid. Each pair must be the same colour and the rectangle they form cannot contain any other pieces. For example, the two white pieces shown in Figure 12 can be removed but the two black pieces cannot (unless the white pieces are first removed). Even though pieces are removed, this game is still ‘static’ as moves cannot be reversed.

Figure 12. The white pieces can be removed, but block the black pieces from being removed.

The ratio of winning moves to total moves per turn gives a measure of tension, as per Equation (1), and games played on the standard 6 × 8 grid tend to show the expected peaks and troughs of tension reminiscent of Kramer’s tension graphs (Figure 2). However, a different character of the game emerges when tension is averaged over 1,000 standard games, as shown in Figure 14.

The average tension of games starts off low, building to a climax towards the middle-to-late game, before trailing off towards the end. The initial low tension is good in this case, as the player does not even know whether their random starting position has a valid solution or not. Higher initial tension would have the undesirable effect that early mistakes could make the challenge immediately unsolvable, but that the player would not find this out for many moves.

Figure 14. Tension profile averaged over 1,000 randomly set games of 6 × 8 three-colour BoxOff.

3 In practice, around 4,999 out of every 5,000 random BoxOff challenges will be solvable in the standard game [9].
4 Conclusion

The concept of tension in games is widely known and discussed, if not precisely defined in the literature. This article aims to provide a coherent understanding of the term in the context of games, and by extension to explore what this concept means in the context of solitaire puzzles.

Tension is shown, through example, to have different meanings for different types of puzzles, suggesting a simple classification scheme that may be useful in itself. Hopefully these observations will go some way towards a more precise formal definition of tension in games and puzzles, and methods for its empirical measurement.

Acknowledgements

This work was conducted while the author was a member of the RIKEN Institute’s Advanced Intelligence Project (AIP). Thanks to Russ Williams and Stephen Tavener for feedback.

References


Cameron Browne is a Research Scientist at the RIKEN Institute’s Advanced Intelligence Project (AIP) in Tokyo. His research interests include artificial intelligence and automated game design. Address: Mitsui Building, 15th floor, 1-4-1 Nihonbashi, Tokyo 103-0027, Japan. Email: cameron.browne@riken.jp

Ludoku Challenges #3 and #4

Fill the grid with numbers 1..9 such that no number is repeated in any row or column, and the diagonal neighbours of a number do not repeat that number or each other. Ludoku is described on pages 35–46.
Games and Politics

Richard Garfield, Wizards of the West Coast

In games with more than two players, politics invariably occur in one form or another. This paper describes an unfortunate side-effect of such games which I call the kingmaker effect. It was first published in 1997 in The Duelist magazine [1].

1 Introduction

In my experience critiquing games, the concept that has caused the most dispute is politics. I refer to a game as political if it has more than two players, or sides, and during a significant portion of the game the other players could agree to make you lose. Two-sided games, like Magic, Chess, Bridge, and Basketball, are never political.

Right now my study of political games is riddled with judgement calls, making it far from precise. For example, Yahtzee is a game for more than two players that isn’t particularly political: I could win even if everyone else decides I shouldn’t. Risk is highly political, however, since one person cannot expect to beat the rest of the players allied together unless they account for less than half the power in the game.

There are some good things about political games. Any player usually has a chance to catch up, no matter how far behind he or she might be. A political game is as deep as the players wish to make it: simple and straightforward, or convoluted and laden with conspiracy.

That said, I lean towards games where politics take a back seat. I haven’t always felt that way, but over the years I have found that when I played games with a strong political component, the game itself didn’t matter much.

2 Playing Nicely With Others

There is a wide array of opinions, often passionate, about the role of politics in games, with equally intelligent folk at all extremes. Most people who have played a lot have had some good experiences with political games. It is always hard to draw conclusions from past game experience, though, because good players can make any game fun. Similarly, it is hard to determine whether a political game is itself at fault or if the players aren’t playing well. When someone is always whining about being behind, is that a problem with the player or the game?

Players often increase their enjoyment of political games by establishing unwritten rules of conduct. I know circles where whining is punished by group attacks. Other groups forbid negotiations, or only allow players to exert limited influence. Players are commonly expected to maximise their personal position even when they have no chance of winning. Often it is difficult to figure out exactly what the rules are, and playing around on the boundary of what is acceptable is risking group displeasure. When the game depends on unwritten rules, I usually credit the players with creating a lot of the fun, rather than the game.

There is a lot of potential for abuse in games where players can trade resources freely, since two players who cannot win individually could flip a coin and give the winner all their pooled resources to create a single viable position. To prevent such abuse, groups sometimes outlaw coin flips or random decisions, but players can still circumvent such efforts by alternating the ‘winner’ between games or by developing understandings. For example, if John is out of the running in this game and gives me good trades or gifts, he will get reciprocal consideration in the future.

3 Bad Games and Good Politics

Many features crop up frequently in political games that I consider bad game elements. A major part of the strategy in a political game is to draw attention to other people’s positions and attempt to play them off against one another. One of the easiest ways to do this is to take a weak position. This may not immediately appear to be bad, but the implications are profound: if you choose a weak position, then it is not actually weak. And if weak positions really have the same power, then how you play the game doesn’t make much difference. What really matters is how you play the players, whether the game is Risk or Family Business.

One of the most unpleasant features of a political game is what I refer to as kingmaking. Kingmaking happens when a player who has no chance of winning can choose who does win. This holds some charm for beginners, because being a kingmaker allows revenge against irritating players, and justifies diplomacy – the winner is chosen
by someone else. The advanced player tends to
dislike kingmaking, though, because it trivialises
the time spent playing. The longer the game goes
on, the more irksome is such an ending.

Another depressing thing about many politi-
cal games is the way they encourage passive play.
If attacking another player costs me and my op-
ponent resources, then there is a strong incentive
to sit back and let other people fight. Games that
have this characteristic can be a lot of fun if some
of the players ignore this and attack anyway, but
are a real drag if everyone sees waiting as a dis-
advantage. How many times have you seen one
player get sick of doing nothing and say, ‘Well, I
have to be going, so I have to attack’? Boredom
should not be an incentive for conflict in a game.

4 Conclusion

It is a good exercise to evaluate the effect of poli-
tics on games involving more than two sides. This
can be quite a challenge, and people who meet it
often come out with a different perspective on the
games they play. The result for me was discover-
ing that most political games were, underneath
the veneer, the same game, and that I was tired
of playing that game.

Acknowledgements

Thanks to Richard Garfield for permission to re-
publish this piece, verbatim, from his 1997 ‘Lost
In the Shuffle’ column in The Duelist magazine [1].

References

[1] Garfield, R., ‘Lost in the Shuffle: Games and

Richard Garfield is a mathematician, inventor
and game designer, most well known for Magic:
The Gathering, the first modern collectible card
game.
Address: Philadelphia, USA.
Email: info@threedonkeys.com

Ludoku Challenges #5 and #6

Fill the grid with numbers 1..9 such that no number is repeated in any row or column, and the diagonal
neighbours of a number do not repeat that number or each other. Ludoku is described on pages 35–46.

<table>
<thead>
<tr>
<th>Ludoku #5 (Hard)</th>
<th>Ludoku #6 (Hard)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 4 3</td>
<td>1 8 7</td>
</tr>
<tr>
<td>5 1</td>
<td>5 2</td>
</tr>
<tr>
<td>4 1 2</td>
<td>3 1</td>
</tr>
<tr>
<td>1 3 9 6 5</td>
<td>5 9 8</td>
</tr>
<tr>
<td>6 7 8</td>
<td>2 7</td>
</tr>
<tr>
<td>4 6</td>
<td>8 3</td>
</tr>
<tr>
<td>8 6 2</td>
<td>4 5 6</td>
</tr>
</tbody>
</table>