In any conversation, we need someone to talk with. Without a player, a game is just a set of instructions, whether executed by a computer or human beings who learn what cards to draw on their turn. An unplayed game is like a piece of sheet music: you can see its potential and imagine what it might be like brought to life. You can grasp from notation or rules that it’s complex and maybe glimpse its nature. Instructions need someone to carry them out to leap from untapped potential into a living, changing experience. To deepen our practice of playing games, we have to think about our own role in shaping what happens—and understand how our role as game designers intersects and tangles with the choices of players.
Players

My first real player was my little sister. I was around 12 years old when I discovered a digital game that let you design and play your own levels: the Macintosh version of *Lode Runner* (1984). It boasted a straightforward but deep system of climbing up ladders and racing across platforms, collecting bags of gold, running from nebulously defined enemies, and digging holes for them to fall into (see Figure 5.1).

![Figure 5.1](image)

Figure 5.1 A typical level in *Lode Runner*, with the player at the bottom, three enemies, and six bags of gold to collect.

I found the real magic of *Lode Runner* to be in the level-editing mode, which put the dozen or so objects of *Lode Runner* at my disposal. All of a sudden I was experimenting, creating scenes where the hero would be overwhelmed instantly by a horde of implacable enemies, or clamber and fall into a treasure chamber with hundreds of coins. I could create new scenarios that were completely unlike anything that came with the game; I could tell simple stories that played out in a series of twisty, challenging corridors.

When the player has collected every bag of gold in a level of *Lode Runner*, a new object often appears: a ladder that reaches to the top of the screen, allowing exit to the next level. In my own levels, I came up with new ways of using this suddenly appearing ladder. The space of the level would suddenly rearrange, and it would become clear that completing it required getting back across the dangerous level, being chased by enemies, to reach a previously invisible path. Suddenly I was creating plots with turning points!

Even though I could play those levels myself to see how they unfolded, there was something missing: a player, someone else who could experience the dangers and surprises I was crafting. I wanted to express something to someone, through this game. I wanted to see how another player would respond and if what I’d done would be clear. So I started using my 10-year-old sister as a guinea pig.
My sister knew how to play *Lode Runner*, and I’d make her sit in my well-warmed chair once I had finished creating a level. I’d tell her, “Go on, see if you can beat it!” She could beat my easier levels without much trouble, and although she had a big smile when she did, I felt disappointed somehow. I could tell that she was smiling in part because she’d beaten me somehow—as if I’d asked her a riddle and she’d managed to outwit me and find the solution with no help.

Before long I started creating fiendishly difficult levels for her to play: they required precise timing and exact knowledge of how to manipulate the movements of each enemy in order to win. These scenarios had lots of hidden trapdoors that looked like ordinary sections of floor but dropped the player right through them into certain death. I orchestrated the behavior of the enemies so that they’d start chasing the player at exactly the moment I wanted.

My sister would insist that these levels were impossible, and I’d smugly show her that they weren’t... well, as long as you had exactly the right skill, the correct strategy, if you knew the right path through the scene. As the designer, I possessed all the above, of course. I was thinking more like a player competing with a sibling, though, rather than crafting something for her. I wanted to beat her and see her admit defeat. That’s a natural impulse that I’ve seen play out many times since in games and levels made by kids for each other to play. But creating a system that’s practically impossible for anyone but the creator is just a tiny, tantalizing fraction of what we can do when we create games and ask others to play.

I was trying to create a harrowing experience for my sister, something with narrow escapes, unanticipated secrets, and perfect moments where a choice to run left or right made for an instant life-or-death difference. All the pieces were there, but with these fiendishly difficult levels, I hadn’t succeeded in engaging my sister, in showing her the magic I was trying to conjure. Eventually, when faced with a level full of tricks that were impossible to understand ahead of time, she rolled her eyes and refused to play.

**Creating Conversation**

So far, this book has talked extensively about the elements of vocabulary: verbs and objects, the pieces of context that aid in understanding those elements, and the ways those elements combine into scenes that develop verbs and create pacing. In the second part of the book, we’ll look at some broader questions: why might you want to pace the development of a particular verb? What kind of story is conveyed when contextual elements, objects, and verbs work together... or against each other? What might you try to say with all that vocabulary? And how might you invite players to say something in response? Do you want to invite players to put their own stamp on your game, or are you trying to convey something that’s best understood if a player primarily absorbs and listens to what your game has to say?
We use the vocabulary of written and spoken language to communicate with other people. The vocabulary of games allows us to express ourselves in tremendously powerful ways, saying things with systems in ways that words can’t. It lets us create different kinds of dialogue with each other. We’re lucky to live in a time when expressive systems—another way of thinking about games—are being explored by creators and players in all sorts of new ways, to converse about and reflect on our every idea.

It’s compelling to think of a game as a conversation: players make choices and use verbs within a system. In multiplayer games, these choices can communicate with other players. A single press of a button or move of a chess piece can convey aggression or uncertainty or less obvious concepts that are specific to a particular game. Players who are highly conversant in a system can read the moves of an opponent, whether human or computer controlled, and understand what’s being said even without words.

As the creator of a game, you also participate in the conversation, but in an unusual and special way. Unlike the times I peered over my sister’s shoulder and watched her play *Lode Runner*, you’re usually not there to watch your players. Instead, you’ve facilitated a conversation by deciding many aspects of how it will work beforehand. As a game creator, you craft the particular vocabulary of its conversation, deciding how verbs will develop and shaping the space of possibilities in which the conversation will happen. As creators, we try to shape a space where a good conversation with or between players could happen; we hope that players won’t throw their hands up in frustration and leave or get bored and drift away.

During a play session of a single-player game—the kind of game that’s the primary focus of this book—all the conversation is happening between the creators of the game and the player. It’s a tricky kind of conversation to have. As the creator, you have to hope that what you’re saying in the conversation—through the rules and shaping of the experience as well as the words, images, or sounds you’ve added to the mix—gets across and finds a player, somewhere out there, who responds with choices, thoughts, and maybe even interesting strategies and emotional engagement.

This challenge can feel like a gamble, like sealing a letter in a bottle and hoping someone figures out how to open that bottle and understands what you wrote. If you’re drawn to creating games—if you’ve ever felt the spark of excitement that I did when I started making *Lode Runner* levels for my sister—then maybe you have things to say which can’t simply be expressed in words, but which could find a compelling form in the systems of a game. Take the gamble! The good news is that in recent decades, many others have gone before you. We’ve tried, failed, succeeded, and tried again. Despite the fact that we’re all still learning exactly how to talk about games, finding words to use and models to think with, creators of games have found a lot of techniques and tricks to get our “letters in a bottle” read.
Iterating to Fun and Beyond

When I first started making levels in *Lode Runner*, I intuitively discovered one of the most pervasively used techniques for refining a game and fine-tuning the conversations that can emerge from it: I got someone to play it, went back and changed it, and made her play it again. Games need players, and as the participants in the conversation who might not be there when our games are played, we need to see people play and hear about their experience. Playtesting and iteration—the process of changing a game based on what you see and hear from the player during play—are the cornerstone of many creators’ process. After all, very few composers could create great works of music without ever being able to hear them; Beethoven, who lost his hearing, is the astonishing exception.

We playtest because we want to see a response to determine whether we’ve succeeded in eliciting the kinds of responses we were hoping for. Usually, the response a game creator is looking for is a smile, a look of intense concentration, the raised hands and lifted eyes that accompany a feeling of victory—all the hallmarks of someone who’s really into what’s going on and having fun. Playtesting lets us spot the barriers to reaching that place and then think about ways around those barriers. The barriers might include confusion about how to use a verb or pacing that’s too difficult for the kinds of players you’re hoping will play your game.

“Fun” is the most popular and traditional goal that game designers try to reach, however. Think about the metaphor of conversations again: talking with others, especially your circle of friends or other like-minded people, has often been described as one of the most consistently engaging and pleasurable things in life. That doesn’t mean that all conversations are fun. Some are deadly serious, even if they’re hard work to stay engaged with, and some conversations are necessary to convey important ideas. More and more, game designers are finding that fun is just the traditional role that games have played in society. We have to remember that it’s what most players expect of games still, but there’s a huge variety of other kinds of system-driven conversations that remain to be explored.

*Papers, Please* (2013) by Lucas Pope doesn’t try to present itself as a straightforwardly fun game. It tells you that you’re going to work: you play an immigration inspector, checking and stamping the documents of hundreds of would-be border-crossers. You’re employed by a harsh, totalitarian regime that tramples on rights and demands your diligent and detail-oriented assistance in exchange for a meager stipend to keep your family alive. The scenario is grim and mind-numbing, and so is the gameplay: you’re literally inspecting paperwork for discrepancy, expiration, and forgery and stamping it APPROVED or REJECTED, over and over. For each mistake you make, you’re penalized, which could make a life-or-death difference for your inspector’s family.
Figure 5.2  Stamping a seemingly endless series of border documents in Papers, Please.

This may not sound fun at all, on the surface—but Papers, Please manages to thoroughly express the workings of an unjust system that you find yourself trapped in when you play. You’ve got to decide whether to prioritize helping mistreated and threatened border-crossers or preserve your own family’s health and wealth. The shape of the game—the difficulty and balance of costs and payment—always holds out the possibility that if you’re good enough at your job, you can get away with some purposeful “slip-ups” to help people. Just as surely, your power to act is limited by the fact that you’re only one cog in the machine.

Lucas Pope playtested Papers, Please extensively to fine-tune the workings of the game’s fictional injustices. As one of the participants in the web forum where he posted early versions of the game, I took part in that process and saw the game get better at eliciting the kinds of feelings and experiences he was aiming for. Do all games benefit from playtesting, though? There’s an argument that can be made that the goal of some games is less about persuading the player to respond, feel particular things, or make certain kinds of choices, and more about expressing something that the creator wants to say—regardless of whether a particular player is willing to hear it.

When we playtest and iterate a game, we make changes that attempt to adapt the game’s form and the possible spaces that can emerge from it to the psychology and behavior of players. If we’re making a game that’s intended for young children, for example, we might change the controls so that they’re easier for players with less developed reflexes and motor skills, or we might adjust the difficulty of the game differently than we would for an experienced gamer. We move the game away from purely being about our own expression to adapt it for an audience.
That's not necessarily a bad thing, of course, but it means changing what we're saying or how we're saying it through game systems to attract, retain, or persuade players into hearing and engaging.

Your Conversation

What happens when game creators simply put their thoughts out there in an expressive system and ask players to listen without compromising or adapting? What if a game is trying to express something real about the creator's life? Anna's game *dys4ia* (2012) reveals her own experiences of taking hormones through dozens of small systems; it asks players to help unfold that story, piece by piece. *dys4ia* is a game that's less about players choosing what happens or expressing themselves and more about a kind of listening through interaction to understand a kind of life experience that most players don't share.

Telling and listening are part of conversations, too. Sometimes it makes sense to rest our active responses and simply hear what the person who's talking is trying to say and understand what they mean in the stories they tell—or the systems they build. Games can present us with overt choices and ask us what we think; they can also show us that in some circumstances and systems, choices are limited or don't necessarily make a difference. For example, as a single immigration inspector in *Papers, Please*, you can't help every single person cross the border. When you play *dys4ia*, you can't change the course of Anna's life or experiment with the system to see what would happen if she stopped taking hormones or reacted differently to emotionally trying circumstances. It's part of the story of her life, and it recounts through its systems what's already happened.

When you go into a conversation, you help shape how it'll evolve and turn out. Conversations can be polite and formal or raucous and free-wheeling; the same is true of games. As the creator of a game, even if you're not present when it's played, you'll make many choices that determine and limit what might happen in the conversation of play. Games can present us with overt choices and ask us what we think—like an interrogator demanding answers or a friend posing questions to help us understand how we feel. What would you do in a difficult situation? What kinds of choices would you make when faced with limited resources? We can also create wider spaces within games where we invite players to come up with their own strategies, reactions, and explorations into territories that we might never have anticipated as the creators of the game's vocabulary. Or we can limit those spaces and ask players to listen—to understand that not every system is open to being changed through the agency of players, not every story can be diverted toward a happy ending, and not every difficult challenge can be mastered and conquered.

These are all different ways of communicating through games, and they raise all sorts of questions. What kind of space do you want to shape? If you have something you want to say, how do
you get that across in a way that feels honest and true to players? How do you decide when to try to adapt to players’ expectations and psychology to try to elicit feelings of fun or persuasion, and when do you stop doing that in favor of holding on to your own expressions and just ask players to listen? If you’re inviting more open contributions to the conversation from players, how do you help them become conversant enough with our vocabulary to say something interesting in reply? Can we create space for a player to tell their own stories and express themselves in the space of a game, while also conveying what we have to say?

The brightest and most passionate game designers in the world continue to struggle with these questions because it’s exciting to explore a space with so much possibility that remains untapped. Although there are no definitive answers, the next few chapters share plenty of ideas about and around these questions. Maybe you’ll come up with some of your own answers.

Twenty years after I started experimenting with Lode Runner, I had a job designing games and another 10-year-old sister in my family. When I went home for the holidays one year, I brought my youngest sister one of the games I’d been working on. She was delighted and played it for weeks, mastering the intricacies of its system. She talked to me about it, asked me for help, and showed me her strategy. Inside the game, around it, and beyond it, we had a conversation.
Difficulty is one of the oldest ways to look at a player’s journey through a game. A novice player usually starts with simple challenges: learn to jump over this obstacle, understand that pushing the joystick to the right will move her avatar to the right. Even in multiplayer games like chess or golf, a new player will often take on easy opponents: other novice players, or skilled players who are “taking it easy” on the novice by playing with a handicap or deliberately playing below their level of skill. As the player masters some simple verbs, she’s faced with more difficult challenges.
Push and Pull

As a beginning designer making *Lode Runner* levels, I had a naive idea of difficulty: *harder is better*, and the ultimate challenge of playing any game is to *master the hardest challenges*. It’s an upward narrative of progress and increasing conflict, the same kind of story we find in many heroic narratives of literature or film. At the peak of difficulty, there’s an epic battle. On one side, there’s the player, with everything she’s learned. On the other side, there’s the game’s system at its utmost, wielding a climactic scene that the designer of the game has made to “throw the kitchen sink” of possible challenges at the player.

Difficulty can be compelling and dramatic: the player starts off easy, learns and deepens her understanding of the game’s possibilities, and climbs through increasing challenges to master the system. Overcoming difficulty is deeply appealing to us as human beings for good reason: it can give us confidence in our own ability to learn and even master difficult aspects of our lives.

In earlier chapters, we looked at how verbs can develop in relation to objects and other verbs. These elements of vocabulary are the building blocks of a conversation that players have with games we create, a conversation that we enable and shape by developing the game’s vocabulary. In this chapter, we discuss how ideas about pacing and development can be applied to the entire experience of a game, from the start through the middle and toward the end—assuming the game even has an end!

When we think about games as a conversation, we can discover many potential ways of looking at games. After all, not every conversation needs to be about challenging the participants, even if many important conversations *are* challenging. In a conversation, challenge can mix with pauses for reflection, times when we listen quietly, and statements of support and reassurance. Conversations are about *push and pull*: one person says something, and the other person listens and responds. At times we challenge each other, and at other times we allow another’s thoughts to explore and develop. A good conversation isn’t necessarily led by one person either; some or all of the participants have ways to voice their own input about the pace and goals of the conversation.

We can find ways to do all these things with games as well, in the unique ways that conversing through a system can create. As the creators of a game, we can shape the ways that the player can push and pull through the game’s system. Verbs are a great example of how a player can take an action and push into a game. We can share decisions with the player about how the push and pull of its conversation evolves—even the purpose of the conversation.

*Resistance* is another way of thinking about the push and pull of games. When a player uses the verbs at her disposal, she pushes against the game to see what will happen, and the game responds. As discussed in Chapter 3, “Scenes,” when the player of *Tombed* uses the “dig” verb against a metal section of floor, causing Danger Jane to hit it with her shovel, the game
responds with Jane’s digging animation and a metallic “ting” sound, but nothing else. The metal block does not give way but resists the verb. In that single moment of gameplay, the game has responded to the player’s push by pushing back and providing resistance.

In longer stretches of time than a single moment, the player may try many different ways to push into the system of the game: perhaps using the “dig” verb in different circumstances, or combining digging with left or right movement to drop Jane further into the vertical column that comprises the space of *Tombed*. The player may develop strategies to deal with the different scenes that follow in succession, developing her understanding of when and how to use verbs—including the “un-verb” of simply *waiting* for the ceiling to descend and destroy metal objects—so she can keep playing and reach the bottom.

The player of *Tombed* will also think about the goals presented by the game and her own goals in playing—the aspects of the game that pull her forward as she pursues them. She’ll have to reconsider how to reach those goals after she finds that Jane gets crushed by the descending spiked ceiling and falls off the bottom of the screen, followed in turn by the game resetting itself to an earlier state. This is a different kind of push from the game, declaring that the player won’t be allowed to proceed if the spiked ceiling contacts the top of Jane’s hat. The player must decide how to respond and if she wants to keep pushing. Does the player want to win? Then she has to find ways to push when the system pushes her back.

At each turn, the player pushes in different and increasingly complex ways, and *Tombed* pushes back: always applying pressure with the unstoppable descent of the spiked ceiling, but also with the changing objects that make up each scene, pushing the player to find new ways to use verbs and keep descending. Finally, *Tombed* stops pushing when the player reaches the bottom of the shaft. The oppressive ceiling disappears, the player uses the “Dig” verb one last time, and the game ends.

*Tombed* is a straightforward game in many ways. It has a few different verbs and can be played from beginning to end in under three minutes. Even so, the player must find many different ways to use those verbs and push to reach the end. *Tombed* was designed and paced to push back in different ways as well, sometimes giving the player a longer span of time to consider her decisions, and sometimes demanding that she act immediately. Sometimes she’s allowed many choices, and sometimes very few.

**Flow**

Back in Chapter 1, “Language,” we made fun of the word “flow.” It’s a term that’s often used by game designers to talk about difficulty, pacing, and challenge in games, but sometimes “flow” is tossed around so freely that it becomes a substitute for “fun” or “quality”—as if flow is a magical substance needed to keep players captivated by your game.
Flow is part of a psychological theory, first proposed by Mihály Csíkszentmihályi; it describes a state of focused motivation where someone’s so involved and energized by what she’s doing that she becomes completely absorbed and caught up in it. This state of flow is similar to colloquial ideas like “being in the zone.” It sounds like a wonderful thing; understandably, many game creators want as much flow as possible in their games. Flow doesn’t just come out of nowhere, though. Much has been written about flow, but most of what’s useful for making games can be summarized in three elements that Csíkszentmihályi says are necessary for flow to occur.

The first condition for flow is a situation with goals and a participant who can take action to make progress toward those goals. Luckily for us, both of these things are fairly common elements of games. The second condition for flow is feedback: the person experiencing flow has to see what happens as she tries to move toward her goal and be able to adjust her actions to respond to changing demands. If this sounds familiar, it’s because feedback is exactly what we’ve been referring to as resistance. Flow is just one way to talk about what happens when the objects, verbs, and resistance of a game develop at a particular pace that encourages a player to stick around for more of the conversation.

It’s not enough to simply give players feedback in response to their actions. The third element of flow is that demands on the player’s choices and actions must change and evolve over time. At first, figuring out how to use a game’s verbs to jump over a wall might be an interesting goal with feedback. The player figures out when to jump, and the game shows her that she made it over the wall. Now imagine repeating that action. If she had to jump over the same wall in a modded version of Super Mario Bros., at the same interval, for 10 minutes on end, it would become tedious. It would turn into a test of patience more than anything else, and it potentially would feel like a waste of time (see Figure 6.1).

![Figure 6.1](image.png)  What if Mario had to tediously jump over a long series of walls?

The simplicity and lack of evolution in repetitive, already-mastered tasks results in boredom, one of the two pitfalls that disrupt flow. On the flip side, flow can also be disrupted if challenges are too difficult before the player has enough understanding and mastery of verbs to overcome them. If the next challenge after jumping over a simple wall involves a highly developed use of the verb that requires a lot of timing, the player may fail over and over again. She may end up feeling like her attempts are futile. This results in frustration and, like boredom, it can feel like a waste of time. The player feels stuck “doing nothing” rather than continuing to move through a flow-inducing series of evolving choices, actions, and challenges.
In terms of resistance, boredom is what happens when a player isn’t being pushed by the game system to do anything except repeat an action she already knows how to push with. Frustration can be similarly repetitive, such as a player pushing into the conversation of the game and being told, “No, that’s not it, try again” over and over again. Although resistance is happening, it’s stuck in a loop.

A commonly expressed idea about flow and games is that as designers, we should try to stick carefully to a channel between boredom and frustration, like a shark swimming between dangerous rocks on either side. Also like a shark, the challenge of a game in this model has to keep moving, so that repetition of actions that the player’s starting to master doesn’t get boring. Get the difficulty exactly right, and the player will stick with your game, developing more and more skill. The game then needs to respond in new ways, pushing back by providing the player with ever-greater challenge. This upward ascent resembles a slope toward the maximum possible challenge (see Figure 6.2). It’s similar to the narrative of difficulty mentioned earlier, an uphill battle toward an epic conflict. Unlike the simple idea of “the most difficult is the best,” however, thinking in terms of flow lets us focus more on the process of this journey. All along the way, the game must keep evolving the system to provide more difficulty so that the player will stay engaged until she reaches that pinnacle.

![Figure 6.2](image_url)

**Figure 6.2** For some game creators, the ideal experience involves staying in the zone between boredom and frustration as the player’s skills improve.
The channel between boredom and frustration is an ideal path, like a perfect model that many games strive for. In a game with perfect flow, the player would push and be pushed back but would be so engaged in what’s going on that it would all feel seamless, natural. Some games are good at finding this channel—even if they don’t start there at the beginning of a player’s experience!

*Super Hexagon* (2012) by Terry Cavanagh is an interesting example. To play, you simply use the verbs “rotate clockwise” and “rotate counterclockwise” to keep the arrow you control from colliding with a series of walls closing in from the outside of the screen (see Figure 6.3). The player has to rotate the triangle to go through the gaps. At the beginning, this is an incredibly difficult task, and a player is likely to die by colliding with a wall almost immediately, making game sessions last less than 10 seconds. At first, this seems like a clear violation of the “perfect model” of flow, but *Super Hexagon* uses a simple enough system that it doesn’t need to start off slow and easy. The player learns what to do by colliding with walls, over and over again. Because these early sessions are so short, it’s easy for the player to jump in again, grasp the patterns of walls that close in on her, and hone her reflexes.

![Image of Super Hexagon game](image)

**Figure 6.3** *Super Hexagon* dares to start off super-challenging.

Before long, many players will improve—and notice that they’ve improved, since their game sessions (and “longest time” records) will be getting longer. This kind of motivating feedback is essential for flow, but it’s worth noting that *Super Hexagon* doesn’t start off at the bottom-left
corner of a flow diagram—the kind of very easy, no-skill-required experience that often involves a tutorial that holds your hand or practice levels that go easy on you. Instead, it drops the player into the frustration of the game like a skier descending a steep slope (see Figure 6.4) and lets her figure out through short bursts of intense play that once she starts to get the hang of it, the challenge will become manageable. That steep slope may even be part of why getting better at this game feels so exciting. *Super Hexagon* shows us that not all games have to adhere to or strive for one model of flow. Following the “ideal” channel from bottom left to top right is just an idea that’s become traditional for many game creators.

![Flow Diagram](image)

**Figure 6.4** If a player isn’t put off by the difficult beginning, finding the flow of *Super Hexagon* can be a thrilling ride.

Instead of a straight line running from bottom left to top right on the flow diagram, the experience of many games involves a zigzag path. A game will present a new challenge, like a more difficult kind of jump, a new verb like “shoot” that has to be used in a different way (for instance, timing your shots so that they don’t miss), or a combination of verbs, like jumping and shooting at the same.

The player has to figure out how to master this new challenge. It’s a process that often feels frustrating at first as the player learns how to deal with it, especially if she doesn’t get it right on
the first try. As she masters the new challenge, the push of frustration lessens. Repeating the same action again and again drifts toward boredom, creating a zigzag. Of course, not all players are the same: some might master a verb or combination of verbs quickly, especially if they have experience from other games, while others may spend longer being frustrated. The purple line in Figure 6.5 shows the traditional ideal of flow, with a frustrated player following the red line and a player who masters challenges easily following the blue line.

In shaping the conversation of their game, game designers have figured out how to make this zigzag pattern part of a story that’s told through play. A moment of intense challenge that requires the player to use the verbs they’ve been practicing in previous sections might involve fighting a boss, for example. The visual and audio cues that accompany this moment might include a larger graphic to represent this dangerous obstacle, with music or sound effects that convey an ominous or climactic feel. Before and after this moment, the context isn’t as intense, and neither is the challenge: the player can relax and prepare for the next big moment, following an arc that builds up to the next conflict (see Figure 6.6). We’ll discuss more ways to create these kinds of pauses and plateaus (where the line of flow becomes more horizontal) throughout this chapter.
**Figure 6.6** A zigzag flow diagram with boss fights dipping into the frustration zone.

### Adjusting Difficulty

Games like *Super Hexagon* require the player to deal with frustration and failure and commit the time to overcome hard challenges. Competitive multiplayer games have a long legacy of putting this responsibility in the hands of players—in these games, not just a single player but a pair of competitors, or even a group or community of players who play together. Players who enjoy sport-like digital games such as *Hokra* (2011) or *BaraBariBall* (2012) have to teach newcomers how to master the challenges of the game, growing a community of players so they have more opponents to face. The difficulty in these games comes largely from how good your opponent is. Players can take it easy on beginners or play with deliberate limitations (or handicaps) to help them learn.

Single-player games face a different kind of problem because the player is alone in conversation with a system that can only say as much as its designer has allowed it to. Even so, it’s possible for the creators of a game to reveal some of what’s going on in the system and give players control over whether it offers more risk of frustration or boredom.
One common way to provide this control is to have the player select a difficulty mode at the beginning of the game. The player chooses whether she wants an experience that starts off challenging and evolves to become even more difficult, or one that’s easier—potentially to the point of boredom. Creating more than one way to pace the same game system can be difficult, however; one mode often ends up being perceived by players as the “real game.” Often this is the most difficult mode, especially for players who value skill and mastery. In addition, giving players this choice at the beginning of the game, before they understand the kind of resistance that the game offers, asks them to guess which difficulty setting will be most satisfying for them. What if a player is good at one aspect of the game but not others? Some games offer detailed controls for adjusting many aspects of difficulty, but there’s still a paradox: to understand how all those controls will affect a player’s experience, the player first has to learn how to play the game well enough to grasp her options.

In the early decades of digital games, the audience of players was relatively limited. Not only were most self-identified gamers white, male, and well-off enough to have steady access to computer technology (or at least quarters to dump into an arcade machine), but gamer culture and the systems it produced were focused on difficulty, challenge, and mastery. By the turn of the millennium, things had already changed a lot, and the game industry launched a new wave of “casual” games. These games were targeted toward players outside the usual suspects, many of them women, girls, and older people who weren’t part of the earlier eras of game enthusiasm. Casual games were known for being much less punishing and intensely difficult than games of earlier decades, and for bringing a much larger segment of the population to gaming. Gamers who had less experience with and fewer preconceptions about a particular kind of system are even less likely to grasp intuitively whether they want to play the “Hard” or “Easy” setting. Ever since the “casual revolution,” game creators are more likely to ask, “Who is this game for?” and sometimes, “How can we make this game more fun for more people?”

Game developers have been searching for many years for ways to seamlessly mold the resistance of a game to match each player’s abilities rather than create a kind of flow experience that works for some players but frustrates or bores others. Many of these attempts fall under the concept of Dynamic Difficulty Adjustment (DDA): methods of adjusting the rules and resources of a game to help players who are struggling with a game’s resistance, and increasing the challenge for players who are doing very well. When you’re playing the first-person shooter Half-Life 2 (2004), you’ll occasionally come across crates that contain helpful items to replenish your health points or ammunition (see Figure 6.7). If you’re well stocked with these resources, you’ll find fewer items inside a particular crate, but if you’re doing poorly and running low on ammo or health, that same crate is more likely to contain items that will help you replenish those resources. Many players of Half-Life 2 never notice that they just happened to get a more powerful healing item when they were running low on health; designers who use DDA extensively, helping players who struggle while increasing challenge for others, often try to do so subtly.
If you’re running low on ammunition in *Half-Life 2*, this crate is likely to contain some to help you out.

Much more egregious examples abound in games. In many racing games, for example, your opponents will actually drive faster if you’re in the lead and slower if you’re trying to catch up to them. It’s not hard to understand how this kind of adjustment keeps the game interesting in the service of flow, but becoming aware of how strangely fluid the behavior of the competition is can be jarring. It might even make the player feel like her own abilities and struggles don’t really matter, because the reality of the game world will be adjusted based on the player’s situation.

Subtlety is necessary in DDA because of how it changes and manipulates the conversation between player and game. When the player pushes against the game and doesn’t manage to make a difference or she meets an expected goal, the game pulls back its own resistance; when the player pushes forward successfully, the game’s resistance increases as well. Learning a game through your own ongoing conversation with it is a process of exploration. Exploring a system with an intense amount of DDA is like having a conversation with someone who’s changing her mind constantly based on what you’re expressing.

Used bluntly, DDA can give the resistance of a game a mushy feeling, as if there’s no fixed structure that the player can meaningfully encounter and push against. Used subtly, DDA may go unnoticed by players, but it’s still quietly manipulating the shape of resistance to create the
smoothest experience of flow, rather than presenting some unmoving challenges to the player and letting her decide how to overcome it—or simply stop playing. It’s no wonder that many creators of smaller games in recent years avoid DDA and simply let their game systems function without constant adjustments and modifications. Would Super Hexagon be a better game if Terry Cavanagh had designed it to get easier when the player inevitably fails at its extreme challenge? The shape of that game’s relentless resistance to player effort would become different, more malleable—and perhaps less meaningful for players who are willing to throw themselves again and again to build their skills in a hard kind of fun.

DDA doesn’t have to leave players’ choices about how much resistance they encounter by the wayside. flOw (2006) was one of the first games designed by Jenova Chen, who chose that name because part of what he and his collaborators were influenced by and seeking to explore was the idea of flow in games. Rather than adjusting the system’s resistance purely based on the player’s performance, flOw tries to give the player concrete choices about pacing the game. In flOw, you control a fish-like creature swimming in an area with other creatures that can be eaten—and that will sometimes try to eat you. If you successfully maneuver your fish’s mouth onto one of these creatures, they burst into white food pellets that can be eaten to make your own fish’s tail longer and capable of withstanding more bites from other creatures. Each area also has a red food pellet your fish can eat that lets you dive deeper into waters with more dangerous, challenging enemies, and a blue food pellet that takes you in the opposite direction, to safer areas. Every player’s journey through the shallows and depths of flOw is slightly different because players can retreat and advance based on how much challenge they’re seeking. Also, losing all your health doesn’t result in the game ending; you instead bump up one level to an easier area.

The organic, player-controlled difficulty of flOw is more integrated into the course of playing the game than asking the player to choose “Hard” or “Easy” before the game starts or via a settings control panel. Structurally, it has similarities to early digital games like NetHack (1987), where the player learns that travelling deeper into a dungeon via a staircase she’s discovered will lead to greater challenge. In NetHack, your goal as a player is stated from the beginning: reach level 100 and claim the ultimate prize, the “Amulet of Yendor,” before you succumb to various threats and enemies that end the game.

flOw, like Chen’s other games, is much less explicit about the player’s purpose and whether she should be trying to dive as deep as possible at all. Although reaching the bottom layer of flOw does give the player the opportunity to unlock more varieties of fish to play as, it’s possible to play and enjoy the game while simply wandering through higher layers and surviving and eating like a simple oceanic organism, content with its lot. Many players, especially those trained to think of games as challenges to overcome, can play flOw as a game of increasing challenge, much like they might play NetHack, but flOw avoids stating overtly what a player must do to win.
There’s no single correct way to shape the difficulty of a game into exactly the right kind of resistance for every player. The right decision for your game depends on its goals and what it’s trying to say in a conversation with players: do you want a highly flexible push-and-pull game that changes shape depending on how the player approaches it? Or will you establish a firm structure, making a hard declaration of what your system requires, and let players figure out how to handle it—even if it means some of them may leave the game before finishing it or miss the perfect flow by a wide margin? Do you intend to involve players in deciding how the game’s resistance evolves? If games are conversations, they’re ones where, as designers, we have to choose what we say carefully and know what we’re going to say in advance, even though we’re often unable to anticipate how all the unique players will react. When we create spaces for players to make their own choices and determine their own approaches to a system, all sorts of things can happen—but that may mean that our own ideas of how the conversation will unfold have to play less of a role as well.

Alternatives to Flow

So far, our discussion of flow has revolved around the idea that games ought to try to adapt to players, avoiding frustration or boredom for too long, and sometimes including players in decisions about how the games’ resistance evolves. Seeking flow states means “meeting players where they are” and ceding some degree of authorial control to foster feelings of engagement and, gradually, mastery through skill building.

Striving for a game with ideal flow that always moves perfectly between frustration and boredom isn’t the only way to make a game, however. It’s possible to create interesting games that don’t seek out a perfect flow state. For example, what would happen if a game didn’t start out slow and easy and didn’t get harder?

Three Body Problem (2012) by Robin Burkinshaw doesn’t change at all as the player continues to interact with it (see Figure 6.8). The system starts off as hard as it’s ever going to get, but with simple rules: the player has to maneuver a square to collect points that appear, while two other squares try to collide with and kill it. Just as with Super Hexagon, the first time you play Three Body Problem you’re likely to die very quickly, because the other squares are relentlessly chasing you. It’s not an impossibly frustrating problem, however; you can quickly learn to survive longer by watching and learning how the other two squares move.

With practice, a player of Three Body Problem can close the gap between her abilities and the challenge, making the game easier. This model puts all the responsibility for creating flow into the player’s hands: she has to accept that she’s a long way from mastery and keep working at it of her own accord. Once she can handle the challenge, the task becomes to survive as long as possible to collect more points, challenging both endurance and skill. If we made a diagram of a player’s experience of flow in this game, it would look very different for each player depending
on how each dealt with the challenge of the game’s simple system. Rather than trying to meet players where they are, it’s up to an individual to decide where to meet Three Body Problem.

Games like this demand more from a player than games that hold the player’s hand, but for players who are willing to start in a frustrated place and learn their way out of it, powerful feelings of flow can still emerge.

There’s another reason to consider alternatives to traditional flow and require players to meet the game, rather than the other way around: although it’s often strategic in a conversation to try to adapt how you speak to your listeners, sometimes that’s not enough. Sometimes you have to ask the other participants to hear exactly what you’re saying—and as we discussed in the previous chapter, some games are more about asking players to listen. Gone Home (2013) is a game in which players enter a house seemingly abandoned by its family, taking on the role of the eldest daughter who’s returned from abroad. At first, Gone Home seems to play with some conventions of horror games—you explore dark rooms, looking for secrets, and are startled by creepy noises (see Figure 6.9).

The revelation of Gone Home is that it’s not a game about facing undead horrors or even about a mounting arc of difficulty and mastery. Instead, you find clues as to the recent and long-buried history of the house of the protagonist’s family, uncovering the truth about why nobody’s home through diary entries, letters, bills, notes hastily left on the kitchen table, and
the mundane details of household life. There’s challenge and problem-solving in Gone Home as you piece together clues and search for secret passages, but it’s not an experience that needs to grow more challenging or gradually build the player’s skills. Instead, the player comes to understand the systems at play—the relationships of characters, the ways that different members of the family inhabit and use various parts of the house—by uncovering new information, some of it in the form of words or diagrams, some of it ingrained in the spatial arrangement and visual representation of a home.

Figure 6.9  Gone Home subverts expectations with unnerving experiences that can’t be conquered with typical game verbs.

Gone Home is set in a world that closely mirrors our own—it could be drawn from the experiences of real people. Of course, some games are overtly autobiographical, like dys4ia, which we’ve already discussed, or Mainichi (2012) by Mattie Brice, a game that represents a single day within the author’s life. It doesn’t necessarily make sense to create a traditional journey of flow through a game that recounts actual events—after all, real people’s lives don’t always progress from easier to more challenging. They can’t necessarily be conquered by building skills and systemic understanding, but they can be represented through systems. The shape of resistance in these games plays a role—showing players where the systems represented can or can’t be pushed—but the experiences that result aren’t necessarily about players overcoming resistance or finding strategies to plant their own flag of victory at the top of a mountain. Instead, they offer players an opportunity to listen and understand systems that they might not otherwise have considered.
Opening Up Space

So far in this chapter, we’ve discussed resistance in ways very similar to traditional ideas of difficulty. Games that try to create flow require players to push into the system with increasing skill met by increasing challenge, often involving the development of a verb. Other games eschew flow in favor of other kinds of experience and understanding. Let’s think about yet another way of looking at resistance: opening and narrowing the space of choice for the player. In a game like Tombed, a lot of the resistance comes from the constant “push” of the descending spiked ceiling. The player has to figure out how to act, right now, or face certain death, and the game grows more difficult as the player masters more ways to use its verbs.

Anna’s game REDDER might seem thematically similar to Tombed at first: the player travels deeper and deeper into the ground through a series of tunnels and chambers, albeit as an astronaut exploring a seemingly alien landscape of mysterious and dangerous technology. REDDER, however, has a much more open feeling compared to the tight, constrained feeling of Tombed, because the player can wander in many different directions (see Figure 6.10). Even though some of the challenges are similar, sometimes involving dangerous force fields moving toward the player’s avatar and requiring quick timing to move past them, the feeling of resistance is very different. Unlike Tombed, the player can decide to retreat from a dangerous-looking room and explore in another direction, since REDDER allows the player to move off the screen to other areas, traveling left, right, or upward as well as downward. This puts some of the shaping of the game’s resistance into the hands of the player.
Games also open the shape of resistance to player choice by offering a larger palette of verbs. At certain points in Super Mario Bros, the player finds blocks that sprout fire flowers; if the player chooses to touch the flower, suddenly a new verb is available, and pressing the button which previously allowed Mario to run now also shoots a bouncing fireball that can eliminate enemies. It’s up to the player whether she wants to find and accept power-ups like the fire flower. Furthermore, there are trade-offs in these choices. For example, when Mario becomes larger after touching a mushroom, he can break bricks with his head, but he can’t fit into small spaces. Like the decision to move toward a more difficult or easier area or level of a game, these choices often shape what happens next and what kind of resistance the player encounters—but in these strategic choices, what the player prefers often depends on her way of playing and how she prefers to use verbs. As small Mario, the player is more vulnerable to dying because large Mario can run into an enemy and survive without losing a life. However, many expert players of Super Mario Bros prefer to stay small because it’s easier to avoid deadly obstacles and enemies when you’re half the size!

Whenever a player decides to use a verb and pushes into the system, we can see choices being made—even the fundamental choice to keep engaging with the game to see what happens, to try to overcome challenges, to accomplish some kind of goal. When the player decides when to use the verb “jump” in Redder to avoid a dangerous force field, she’s made a certain kind of choice: jumping at the right time may allow her to continue, while using that verb at the wrong time could kill her avatar, sending her back to the last checkpoint she passed. Deciding which direction to explore in, or retreating out of a room because it looks difficult, is a different kind of choice, much like the choice in Flow to dive to deeper, more difficult waters or surface to easier ones. These kinds of choices affect the resistance that the player encounters, letting her choose her own pace or avoid encountering certain kinds of resistance altogether.

In Shadow of the Colossus (2005), the player explores a huge, mountainous landscape, searching for crumbling giants of stone and metal that must be conquered to move on to the next challenge. The player’s primary verbs involve climbing and jumping, hanging onto the giant’s enormous body, and stabbing it to gradually defeat it. There’s another verb that plays a role: riding a horse. Riding can be useful in battles, but the most frequent use of riding in the game involves traveling between various areas of the world, each containing a different colossus.

After each victory over a colossus, the player returns to the central location where the game began: a huge tower in the midst of a large, empty plain surrounded by mountains. To find another colossus, the player must travel across the plain to find another area, and the distances traversed are long enough that it makes sense to ride. There’s not much to see on the central plain. Shadow of the Colossus takes place in an area with no towns or points of interest, just some crumbling ruins that seem to have been abandoned long ago. Riding across these areas isn’t difficult at all and doesn’t involve much strategy or decision-making: you simply get on your horse and ride in one direction or another, sometimes urging your horse to gallop faster.
with the press of a button. The player can also use an amulet to create a beam of light that points toward the location of the next encounter with a colossus, so there aren’t even significant challenges of exploration to figure out which way to go.

Why does Shadow of the Colossus require these lengthy travel times, long enough that some players (and reviewers writing about the game) think the ride could even become boring? The contrast between battle and travel feels deliberate: riding across the plains is a relaxation in the resistance of the system, an opportunity for the player to set the pace—or simply enjoy the sights and sounds of the world. For some players, this absence of resistance might even offer a chance to pause and reflect, much as transit times often do in the real world, and perhaps to consider the larger questions of the game, like why you are hunting these gigantic, solitary, often peaceful-seeming creatures. When the player is done with her moment of pause, the goals of the game are right there waiting to be picked up again.

### Opening Up Purpose

Games can open up the space of resistance by letting the player decide not only how to play, but toward what end she’s striving—that is, what goal she is trying to achieve in the game. A game like Tombed has a single, straightforward goal: survive by descending. REDDER requires players to pursue a few different goal objects in the form of diamond-shaped objects scattered throughout the game. As a creator, you can add many goals to your game and let players decide which to pursue first—or at all. On the other hand, even with a long list of goals, a player’s choice of what to pursue is limited to selecting from what’s on that list. In games with complex systems that can produce unexpected possibilities, players can come up with their own goals—things you may never have dreamed of as the creator.

Open world games like Fallout: New Vegas (2010) take the first route: they open up the space of play by giving the player many different goals to select from. New Vegas is a huge world, one that’s chock-full of things to find, computer-controlled characters to meet, dangerous encounters to overcome. As the player wanders across the deserts and highways of this game, numerous points of interest appear, visible both in the player’s view of the world and on a map that fills itself with more and more icons as the player passes nearby or hears about them from other characters. Games like Fallout: New Vegas can end up with an almost dizzying array of goals as the player continues to push into and explore the world, accumulating dozens of potential missions to pursue, and spotting even more potentially interesting places to visit from a distance. Each location has its own challenges and particular shape of resistance. In one spot, you might encounter a gas station full of murderous raiders who’ll try to kill you, while at another you might meet a friendly merchant who wants to exchange gossip and trade. Exactly what will happen is uncertain for any player who’s not playing with a gigantic guidebook to accompany the game, and that’s part of the experience of openness and choice: you never know what’s going to happen next.
Simulation games, on the other hand, often avoid giving the player any overt goals. They’re sometimes described as “sandbox” games because of how they allow a player to explore the game’s system, an unfolding array of verbs and objects, to figure out their own objectives. In *The Sims* (2000), the player can build homes for virtual characters (Sims) that can interact with objects in the home, cooking a meal at a stove or using a toilet. The player decides what objects to put in this virtual dollhouse and can command the Sims to interact with the objects, although the Sims have their own simulated needs and will wander around looking for food to eat if they’re hungry or someone to talk to if they want social companionship. The needs of the Sims aren’t directly under the player’s control. They grow hungry or sleepy over time, and the growing needs of each Sim changes the feeling of resistance in the game: the player might be getting two Sims to interact and have a conversation, but a third Sim nearby will be growing frantic if he can’t find anything to eat.

The goal of playing *The Sims* is supposedly left up to the player: there’s no “game over” or indication of failure if the player chooses to torture her Sims, refusing to place beds or toilets until the Sims are collapsing on the floor in exhaustion, surrounded by pools of their own urine. Like many simulation games, *The Sims* doesn’t give the player explicit missions or objectives to check off. It’s up to the player to explore what happens if she creates a house full of starving, filthy Sims or a happier home of well-fed virtual money-earners. At the same time, the game still feels like it has opinions about how your virtual home turns out. *The Sims* doesn’t punish the player directly to convey the idea that dirty, exhausted Sims might not be an ideal state of affairs. The Sims themselves express unhappiness and don’t perform as well in generating resources for the player to use when they leave the house to go offscreen to a job and return with money that the player can spend.

*The Sims* does have implicit goals, which should be clear even from reading this description: if the player cares for their Sims well, more money will be available to buy better amenities and furnishings for the home, which in turn provide for the Sims’ needs more efficiently. The fanciest-looking beds and computers in the game are also the most effective and most expensive, requiring the player to run an efficient household. Even though it’s not explicitly stated, the implicit goal of *The Sims* is to perform well at the cycle of labor, income, and fulfilling basic needs that most of us are familiar with from day-to-day life. This implicit goal is reinforced through rewards, which we’ll discuss later on in this chapter. If you want the best stuff, you need to care for your Sims.

There’s no explicit way to “win” *The Sims*, but it’s clear that to give your Sims what most of us would consider a nice life, you have to become good at a simple capitalist system of earning and buying, then earning more to buy better stuff. Still, the game doesn’t push back with much resistance if the player chooses not to pursue this goal and opts for a life of simplicity or squalor instead. Rather than the hard resistance of losing points or forcing the player to start over, *The Sims* has a softer form of resistance, embodied mostly in the crying faces of unhappy Sims.
Nothing happens to change the system or state of the game if you let your Sims cry, but it’s still an emotional signal: you made your little people sad! It’s a form of resistance that’s up to the player to interpret: maybe she sees herself as a cruel torturer of virtual beings!

Some sandbox games go even further to create a wide-open form of game conversation where players are the ones coming up with the topics at play. Unlike *The Sims, Minecraft* (2009) doesn’t have an economic system that involves going to a job and earning money. Instead, the player has to harvest raw materials from an enormous landscape composed entirely of cubical blocks in dozens of different types and then figure out not only what kind of objects to populate their sandbox world with, but also how to craft those objects from raw materials. Harvest 16 units of wood with one tool, use another tool to turn the wood into four boards, and you’ll finally be able to turn those boards into a chair. As in *The Sims, Minecraft* has more expensive and difficult-to-procure items in its huge catalog of objects, although the expense is in rare materials and multistep processes rather than a currency. More interesting still, especially for our discussion of resistance, *Minecraft* allows the player to craft many kinds of components and tools, then put the components together to create new kinds of devices. Enterprising players have built elevators and even calculating machines out of the simple mechanical and electrical components that the game system provides.

*Minecraft’s* system of parts is complex enough that players can explore it to come up with possibilities that the game’s creators never dreamed would be possible. Players aren’t just making choices about how to deal with a difficult challenge or use a verb; nor are they deciding which goals to pursue in a system or at what pace to pursue them. Instead, they’re fashioning their own kinds of resistance—new challenges and objectives made possible through recombining the existing structure of the game such as building an entire house with light switches. Instead of selecting from options preordained by the creator, the possibilities come from their own imagination or from what other players have dreamt up.

It’s worth remembering that even an open-ended game like *Minecraft* is not the same thing as a blank canvas upon which participants can draw anything they choose. The shape of possibilities in a game is fashioned out of the structures, rules, and building blocks that creators like you put there, even if there are potential outcomes that you don’t anticipate. This is part of what makes creating a game such a uniquely rewarding pursuit. As a creator, you produce structures that serve as the groundwork for all sorts of player choices, including ways that players can make the game their own and surprise you with their ingenuity and choices. The resistance inherent in *Minecraft’s* structure comes from a set of rules that determine what happens. The player can’t simply build a platform next to a switch and turn it into an elevator. The system is more complex than that and requires deeper understanding of how its objects interact. Games that are open in this way aren’t necessarily better or more enjoyable than games that push the player with tighter forms of resistance, however. They’re just different kinds of conversations, and as with verbal conversations, the world would be a less interesting place if everything worked the same way.
By opening up or narrowing the space of player choice in your game, you can let your players affect the shape of resistance. At the narrowest end, players can experiment or choose when to use the verbs you’ve provided them: Is it the right moment to maneuver left or right in Super Hexagon? Which is the best spot to dig to descend further in Tombed? As you open the shape of resistance, the choices change: How much resistance will there be, and how soon will it appear? Will the player swim toward more challenging depths in flOw or spend time riding through quiet meadows in Shadow of the Colossus before the next battle? You may want to open up the choice of which verb the player will use, whether that’s represented as a selection of weapons in a dueling game or deciding whether to grab a Super Shroom to play as big Mario. Finally, if you pull back your game’s shape of resistance to the point where the player’s making choices on her own, you can allow her to decide which goals to pursue. You can even ask the player to come up with her own goals, leading to the big question of why she wants to play and to what end.

The Pull of Rewards

The flow of difficulty and challenge is a kind of resistance that pushes the player to varying degrees, requiring her to push back into the system of the game, try to overcome obstacles, and deepen her understanding of the game. A player also pushes into a system and finds resistance when exploring the possibilities of a system, whether it’s figuring out what a system can do or coming up with her own goals and strategies within the structured rules of a system. If resistance is a way of looking at the push and pull of the ongoing dialogue between player and game, then we need to find ways to pull players as well as push them, to encourage and lead them forward.

Because the games we create are having these conversations with players in our absence, we set up systems to give that signal to players: “Yes! More of that! That’s right!” This kind of positive feedback is often thought of as a reward. The tradition of rewarding players who are “doing the right thing” goes back a long way through the history of various kinds of games: gambling for money, accolades and cash prizes for sports tournaments, and so on.

Chances are that when you a play a game, you’re not doing it for the practical value of the rewards you might earn. You’re playing the game for its own sake, because hopefully it’s enjoyable, meaningful, or rewarding. If games ought to be inherently fulfilling in their own right, do we need rewards? It may make more sense to think of them as feedback in the game’s conversation with players. In single-player games, we leave rewards along the way to encourage, to tell players that they’re on the right track. Still, the visual language of rewards often has the feeling of a payoff, a prize that has some kind of value. At the surface level, you can probably recall some examples of what these rewards look and sound like: shiny coins, bouncing stars, an overflowing treasure chest, perhaps accompanied by uplifting, victorious music and large text that declares, “You Did It!” Peggle (2007) has one of the most well-known examples of a game’s contextual elements used to create a rewarding feel. When the player manages to bounce a ball
into the central opening at the bottom of the screen, the game lights up with fireworks, and Beethoven’s “Ode to Joy” plays triumphantly. It’s a real spectacle.

Just as with other kinds of feedback discussed earlier in the book, it’s important to let players know when they’ve pushed into the system and done something “right.” This is part of how players understand the shape of the system when they play. What’s the purpose of the conversation? How does the player evaluate whether she is going the right way? Just as Danger Jane has to be shown pushing uselessly against a wall, or hitting an undiggable metal block with a “ting” of the shovel, signifying the places where the game can’t be pushed, we want to show our players very clearly where the game can be pushed. Still, showering players with the audiovisual signals of reward can feel gratuitous or meaningless if we use them everywhere, as if we threw confetti and cheered every time a friend said something we understand in a conversation. Reward, the pull of our system-created conversation, has to come at a meaningful moment—and just as any celebration doesn’t feel joyful just because people are throwing streamers, the feeling of reward in games is based on what the player’s been doing up to that point.

Rewards are great for marking milestones during a player’s journey and separating one part of the game from the next. When you manage to navigate past a few minutes’ worth of enemies, pits, and navigational hazards in Super Mario Bros., you’ll reach a castle with a flagpole outside of it. This sight, familiar to many gamers, lets you know that it’s time for a pause in your journey. It’s a short pause, but it’s still a break, a release and relief from your efforts. When you enter the castle, fireworks go off. As the pace of activity suddenly slows for a moment, the visual elements of reward appear. All of this happens together to mark the event: you’re done with one part; now it’s time to take a breath before the next part begins.

Not all games are broken into discrete chunks like this, but these pauses are useful moments to signal players that they’re successfully pushing into the game. When a player finishes one section of a game, it could look like one level of Super Mario Bros, leveling up a character in a role-playing game, accomplishing a discrete mission objective, or fully exploring an area of the game world. The relief of having finished an identified section can be a release from the pressure of decision-making and responding to push-back from the game: a moment to stop and celebrate.

Cat Cat Watermelon (2010) from Lexaloffle Games is broken into 20 levels, each one posing a more difficult challenge than the last. The player stacks various objects (including cats, watermelons, beach balls, and more) on top of each other to create a tower that doesn’t fall over. If the player manages to stack all the objects for a given level, a sign appears with a fairly standard victory message, telling her that the level’s complete. Ironically and intentionally, the sign actually knocks over the tower that the player just built, sending all the carefully placed objects tumbling away. This is a great example of how the moment of success in a game can create a feeling of release, even from what you’ve just accomplished. Hooray! You did it! And now we sweep everything away. This is a kind of reward too, even though it doesn’t come with
fountains of gold coins and exploding sunbursts. It’s there as a marker, an acknowledgement that you’ve done it. By clearing out the remnants of the past (what you’ve just accomplished), it also pulls the player toward the next challenge.

Of course, this isn’t the only way to mark the occasion of a player successfully pushing into a game up to a particular point. It’s easy to imagine games that do the opposite. When you finish a section of the game, a certain set of tasks, or a particularly difficult challenge, you get to keep a memento of your success, such as a medal or souvenir. It all depends on how you want the conversation of your game to mark memories of the past and open the way toward the future, where the player will explore more of the game’s possibilities.

Previous chapters have talked a lot about what happens when the player’s introduced to a new verb and how verbs develop along with a player’s understanding of how to use them. Opening the conversation of a game to include more verbs is an exciting moment, and it’s often used to create a moment of reward. When a player has successfully pushed into a game and accomplished something she’s been striving for, she’s hopefully deepened her understanding of the game’s verbs and system. Introducing a new verb—or a new way to use a verb—can help keep the experience flowing interestingly. For similar reasons, introducing new objects to use verbs on is also a common form of reward, whether those objects come in the form of new areas to explore, new opponents to face, or new obstacles that have to be traversed. Unlocking new sections of the game may not seem like a reward in the same way that finding a gun that enables the shoot verb is, but both are significant rewards because they let the player push into new parts of the experience. These rewards connect the player’s past accomplishments to what’s coming next.

**Resources**

A chest full of gold coins is one of the most traditional forms of reward, signifying wealth. In a game, as in real life, the value of money depends on what you can buy with it. We can think of currency and other spendable resources as enabling particular kinds of verbs: spend and save. Currency is useful for game creators as a flexible kind of reward because it’s usually represented as a number. Players can make choices about how much to exchange, perhaps for a new verb or access to new objects, or they can hold onto it in anticipation of being able to afford something else later on. Not all spendable resources look like money, either; *skill points* are another type of currency, usually rewarded after the player reaches a certain point in a game.

You can think about any kind of number that the player can spend or save as a currency reward, the means by which other verbs are enabled. For example, *ammunition* is an expendable resource that enables the “shoot” verb in many games. *Health* is a special kind of expendable resource that the player tries to avoid losing in challenging situations and which she has to save enough of to avoid running out. Running empty on health leads to “death,” and although that means something very different in various games, it often involves starting over, dealing with a penalty, or experiencing a setback of some kind.
Players of role-playing games are familiar with experience points, yet another kind of resource that doesn't involve much choice. These points are automatically saved for you until you have a certain amount, and then they're spent to reach a new level, at which point many games give the player a reward in the form of spendable resources, new verbs, or access to new areas of the game. Experience points aren't really an expendable currency; they're basically just a marker of progress toward the next part of the game experience.

Dead End Rewards

Creators of games have invented far more ways to entice players into staying engaged with a game, from the bells and whistles of exciting audiovisual feedback to piles of spendable resources—sometimes in such huge amounts that decisions about spending and saving become meaningless. Many games also try to weave a story into the conversation of playing—a subject we'll be talking about more in Chapter 7, “Storytelling.” When you're playing a game, your experience of push and pull, tension and release has a narrative flow of its own.

Game creators who also want to tell a more authored story often do it in bits and pieces at a time, with a prewritten dialogue between two characters onscreen, or a cutscene that involves little to no involvement on your part. These dramatic interludes are often placed between sections of gameplay so they don't disrupt the flow of actual gameplay. As a result, they often happen around the same time as other rewards or pauses in the resistance of a game, because the pleasure involved in watching a story unfold is also a reward. Using story as a reward is a little troublesome, however. If your story is exciting enough that it feels rewarding just to reach the next scene, then experiencing that story could become the real motivation to play, at least for some players. We'll come back to talking about story in the next chapter, but for now it's good to understand how story rewards, divorced from the system and its shape of resistance, are like a dead end: they don't feed back into what the player's doing in the same way as giving the player a new verb does.

Achievements are another kind of dead end reward, popularized by mass-market online game networks like Xbox Live and PlayStation Online. Cheevos (as Anna and other critics like to call them) are deliberately outside the real systems of gameplay. Games can record achievements into an online network's system, but unless a game has its own way of tracking and using achievements, they never affect anything else in the game. Like story rewards that don't function as part of gameplay, cheevos sit outside the conversation; they have to be pursued purely for the sake of reward. Because game creators would have to duplicate the achievement systems inside their own game to avoid this dead end, it's no wonder that many cheevos seem to have little to do with what makes a game interesting.

The least interesting kinds of cheevos are either awarded simply for playing the game, often just duplicating the game's own system of reward and progress, or require the player to do repetitive actions, like killing large numbers of enemies or amassing lots of resources, in ways
that they wouldn’t be likely to pursue without the achievement. It’s possible to create interesting achievements that can feel genuinely rewarding but, like story, this tends to work best if a game’s creators have figured out how to integrate achievements into their own systems. That ends up making the online networks’ achievement system a little redundant for most players who don’t care about racking up Xbox Gamerscore points that simply show how much time someone’s put into playing Xbox games.

**Time and Punishment**

Rewards involve a shaping of resistance that pulls the player forward instead of pushing her back. Moments of reward encourage the player to push deeper into the game. Difficult challenges, as well as the very rules of the game, are the structures of resistance that the player pushes against. Of course, we can’t just talk about rewards and difficulty without discussing another way that games push the player: punishment. Moments of punishment happen when the player makes a mistake or fails at the game, whether by being crushed by a spiked ceiling (as in *Tombed*) or losing all health points while fighting an enemy (a system found in countless games). If rewards are the “carrots” that encourage the player to move forward, punishments are the “sticks” that signal when the player does something, or ends up in a situation, that the system doesn’t encourage.

Punishments don’t just block the player from doing something, as when a player encounters a rule like she can’t dig into a metal floor. Instead, they often send the player flying backward through the experience. When the spiked ceiling of *Tombed* reaches the top of Danger Jane’s helmet, Jane goes flying off the screen, disrupting the scene. The game then resets itself to an earlier moment, the last time that the game invisibly recorded that Jane had passed a certain depth in her descent. This moment is a *checkpoint*, sometimes marked by a little flag or other landmark, like the small white pillars that the player frequently passes while exploring the Martian landscape of *REDDER* (see Figure 6.11). Returning the player to a checkpoint is like repeating part of a conversation—perhaps to go over something a second or third time that one of the participants didn’t understand or that they need to revisit to respond differently.

In many games, players can create their own checkpoints as well, by saving the game. If they encounter a punishment, perhaps one that terminates the experience like “dying” in a game usually does, they can return to one of the moments where they saved. Regardless of whether the system provides a way for players to manage their own checkpoints or includes checkpoints at predetermined points in the experience, the player is sent backward to repeat what she’s done. This form of punishment also exists in games that simply punish the player with a “GAME OVER” message, like *Three Body Problem* and *Super Hexagon* do—except in those cases, as with many classic arcade games, the player is returned to the beginning of the experience to start again. The entire system is reset.
In nondigital games such as sports or board games, a mistake can result in losing the game, which also effectively resets the system. If you want to play again after losing at chess or basketball, you have to start again. More gradual punishments in traditional games also exist. If you move a chess piece to a vulnerable position, your opponents might take it, costing you one of your pieces. In basketball, a punishment can happen if you violate one of the rules of the game, such as walking without dribbling the ball. In this case, the punishment is that the ball is given to the opposing team. This kind of punishment has the effect of tilting the game toward your opponent, possibly bringing the game closer to victory for your opponent and closer to a loss for you.
In single-player games, however, the only opponent is the system itself. Although there are gradual punishments (losing some of your health points), we don’t always want to reset the entire game if the player reaches a final punishment. Even if the whole game does reset, we want the player to have the choice to try again. In fact, if your goal is to provide the player with an experience of flow, balancing frustration and boredom, you may want your system to help the player out if she’s making mistakes, which is precisely the purpose of the Dynamic Difficulty Adjustment techniques that we discussed earlier.

Instead, single-player games often punish the player with repetition: die or make a significant mistake, and you’ll have to repeat a section that you already did, starting from the last checkpoint. Repetition may seem like a boring kind of punishment, since it may involve the player encountering familiar objects and scenes, even using verbs in the same ways that they already did. On the other hand, repetition can be useful in a couple different ways. First, if the space of the game is open enough that the player has a lot of different verbs at her disposal and choices of how to use those verbs, or there are different areas of the game to explore, the player may experience something very different the next time around. Second, even if the player does go through the same motions again, repetition can serve as practice: what was challenging the first time becomes slightly easier, and when the player reaches the point where she made a mistake, she has the opportunity to practice overcoming that challenge again.

In these two ways, punishment in the form of repetition has a purpose beyond just saying, “Bad player! No!” in our systematic conversation: it gives the player the opportunity to revisit earlier moments, to push into the game in another way and create a different result. You can imagine the same thing happening in a spoken conversation, if one participant doesn’t understand something that’s said: we repeat and revisit ideas and probably try to communicate in a slightly different way so that we can continue.

Anna’s game *Mighty Jill Off* (2008) gives the player plenty of opportunities to practice because of the way the game develops the “jump” verb, which Jill uses to climb toward the top of a tall tower. The first half of the game consists of a number of sections, each designed to teach the player about a particular way of jumping and color-coded to be easily identifiable. The green section near the beginning involves simple jumps from platform to platform, but the following blue section requires the player to use a more advanced kind of jumping. In *Mighty Jill Off*, rapidly tapping the jump button allows Jill to slow her descent. Combined with moving left or right, this allows her to float sideways to land on platforms. After the blue section, the orange section teaches the player how to run off platforms and jump at the same time, and the lime-green section introduces a new kind of hazard: spiders that chase the player.

Each of these sections starts off with a scene or two that introduces the new kind of jumping or hazard in a straightforward way that’s not too challenging or complex; the next part of the section often involves combining what’s just been learned with hazards or situations from earlier
in the game, elaborating on the new game mechanic and further developing each use of the verb.

The blue section starts off with a scene that can’t be passed without executing a sideways float, followed by a drop, no less than four times (see Figure 6.12). On top of this repetition, it’s likely that a new player who’s just learning *Mighty Jill Off*’s particular form of jumping will die repeatedly when she touches the torches. When Jill dies, she tumbles back toward the bottom of the tower whence she came—but a few moments later, she reappears at the last checkpoint the player passed, often at the beginning or halfway through one of the colored sections. There’s a checkpoint at this first blue scene, letting the player practice, die, and practice some more until she’s good enough to clear four jumps in a row.

![Figure 6.12](image-url) The player must master new skills to advance through *Mighty Jill Off*.

Practice is only useful up to a certain point, however. Once a player has thoroughly mastered a certain use of a particular verb, perhaps even multiple times with the same arrangements of objects in a scene, practice becomes rote instead. The player’s gains in understanding and skill to use those verbs start to level off, improving only a little bit. Rote activity is closer to second nature, done without a lot of thought and next to no challenge, and rote actions repeated in
the same scenes can rapidly move the player toward the boredom corner of the diagram of flow. Even so, rote plays a huge role in many digital games, and players are willing to put up with a lot of rote punishments, especially if they’ve become emotionally invested in moving through rote activities to tackle challenges again. Some players might be seeking the satisfaction of overcoming what they couldn’t before, while others might be eager to reach the moments of reward that they’re hoping lie beyond those challenges, and others still might just be so used to rote actions that they don’t mind the boredom.

Many digital games of recent decades use rote activity not just as a form of punishment, but as a basic element of the resistance of the game. Even if the player hasn’t made a mistake or died, there’s still a lot of rote activity—usually involving verbs that are easily mastered and challenges that are easily overcome—that the player has to push through before reaching a more intense experience of challenge or a moment of reward.

A certain breed of single-player digital role-playing game, exemplified by the many titles of the *Final Fantasy* series, uses rote activities throughout. Although these activities are presented as combat against a variety of monsters and enemies, most of the time the player can defeat those enemies just by pressing the Attack button over and over again. Even if that simple verb isn’t enough, the most the player has to do is use a slightly more advanced verb that involves a limited resource, or understand a system of “weaknesses”—for example, fire-type monsters can usually be defeated by using a water-type verb, which uses up an automatically replenishing resource often called *mana*. Although there are verb-object relationships to learn here, they quickly become rote, and the player is often called on to use them again and again.

Players refer to this kind of repeated rote activity as *grinding*: performing the same actions again and again in order for the sake of reward. One of the hallmarks of this kind of role-playing game is that the player can grind to accumulate resources that make her avatar more powerful—not because the player herself is understanding the system more deeply or learning how and when to use verbs, but simply because the numbers of resources are climbing. In some ways, the choice to grind for more resources can be an interesting way to open up the space of resistance in a game: if the player has a difficult challenge ahead of her in a role-playing game, she can often invest time into grinding to boost her resource numbers, which will make that challenge easier. For example, the player might accumulate enough experience points to level up and increase her health resource, allowing her avatar to take more damage before dying. She might also accumulate enough of a currency resource to purchase a new sword that uses the same attack verb, but which can remove more of a monster’s health.

By definition, because it involves rote activities, grinding takes time, not skill or understanding. It’s a player’s choice whether to invest this time, but many grind-based games require a certain amount of grinding just to get enough resources to proceed. Some grind games are all about investing time to get more resources, such as the wildly popular *Farmville* (2009). Like role-playing games, *Farmville* has multiple resources to increase and lets the player level up to access
more kinds of resources and new verbs and objects, but advancement involves planting various crops rather than fighting monsters (see Figure 6.13). Reach level 13, for example, and you can plant strawberries, which, like all the other crops, exist only as a way to invest more time. When you plant a crop, you must pay some of your currency and then wait for a certain length of time—between 30 seconds and 48 hours. When the time’s up, you receive an increase in your currency resource by clicking on the crop, which represents harvesting and selling the crop.

![Figure 6.13](image)

**Figure 6.13** Grinding in *Farmville* to earn more resources.

The skill involved in *Farmville* is more akin to deciding how to invest your resources and when; it’s still a game with a complex system of verbs and objects that the player has to push into, but nearly all the resistance of the system comes from waiting for your crops to grow. There’s punishment in *Farmville* as well, but it happens only if the player makes the mistake of waiting too long to return to the game and harvest—in which case the crops have withered, causing the player to lose her investment.

Whether it appears as a punishment for making a mistake or is simply required to complete actions, grinding and rote repetition are still forms of resistance: it’s difficult to keep at a task that feels boring, and it takes patience to wait. This kind of gameplay raises a question for players and creators alike, however: are these interesting forms of resistance that we want to partake in? They involve a smaller number of choices than figuring out where and when to jump, shoot, and dodge in more action-driven games. Although some players might be more patient or better at enduring rote tasks, it’s a different form of skill. Players who enjoy
games where the space of resistance consists of exploring complex rules and choices, pushing against challenges, and dealing with less rote punishments are used to a kind of resistance that comes from the *frustration* side of the flow diagram. Grind-based gameplay, on the other hand, involves resistance from the *boredom* side. And it’s not hard to understand why many players simply don’t want to invest their time in activities that primarily challenge patience, endurance, and the ability to sink many hours of time into a game.

Many game creators feel that using grinding is disrespectful of players’ time. The popularity of grinding in role-playing games, social games like *Farmville*, and increasingly in other game genres might be the result of pressure from business managers and marketing strategies that simply try to deliver more hours of gameplay or keep players around longer so that they’ll spend more money in free games where they can make purchases while playing. Along with the lure of rewards, grinding can be abused in this way to create filler that doesn’t do much to enhance the conversation of gameplay.

On the other hand, some players genuinely enjoy challenges of patience, efficiency, and optimization; there’s strategy involved in figuring out how to grind more effectively, wasting as little time as possible, or even managing your own patience as you push through a boring kind of resistance. If action-driven games sometimes resemble a complex obstacle course to navigate, then grinding games involve something more like the endurance and sustained concentration of a marathon. Different players enjoy different things—so beyond making a choice about what kind of conversation you want to facilitate as a creator, the most important thing is to be honest with players about what kind of resistance your game uses. Players who are looking for difficult challenges or who want to measure their skill against competitors have every right to feel betrayed if they find that a game’s been packed with a lot of grind simply to make it longer!

**Scoring and Reflection**

One more shape of resistance deserves mention: scoring. It’s neither a punishment nor a reward, but a reflection of the player’s experience. A player’s score is sometimes revealed throughout a game, as in classic arcade games where the score is displayed in a corner of the screen, often going up steadily as play continues. As we discussed earlier, moments of punishment and reward often happen at significant milestones in the ongoing conversation of a game, after the player pushes through one chunk of the experience—or makes a fatal mistake that ends the game! These moments are often used to reveal the player’s score as well or focus the player’s attention on a score that’s been present all along. It’s the time in the conversation where the participants check in about what’s been said and determine how things have been going.
A score is an evaluation, and as the creator it’s up to you to decide what’s being evaluated. Like rewards, increasing score is a statement to the player that she’s done something you consider positive through the system of the game. Many games award higher scores for destroying more enemies, for example. This doesn’t shape resistance immediately in the way that in-game rewards like a new verb or the unlocking of new spaces would, but it does send a message: it’s good to destroy enemies! If that’s what you want your game to say, then it can be useful to reinforce the verb-object structures of your game by reflecting those goals with higher scores. Not every use of a verb or interaction with an object needs to result in a score; it’s up to you to decide what’s significant for raising or lowering the score.

Because a score is an evaluation, it can feel like either a reward or a punishment, depending on whether the score’s presented by the system and interpreted by the player as being high or low. A score usually involves numbers, or sometimes letter grades. In games with complex systems of choice and lots of ways to do well, scores can sometimes be broken down into many different measures of performance. Because scores turn some of the player’s actions into numbers or letters, they’re useful for comparing performance between players. This is, of course, how scoring has been used for centuries in sports and other competitive games. In more recent decades, scores have appeared on leaderboards that let players compare what they’ve done in a game to what their friends or vast populations of players online have done.

Scores in single-player games provide a special kind of comparison and feedback: the player can compare her own scores at different times or on successive playthroughs of the game. As a player, being able to track your own score over time provides deeper insight into how you’ve pushed into a game and how the creators of the game have evaluated your actions. Of course, all this depends on whether the system of scoring is clear enough for players to understand what the score means. If you want a player to actively think about score and how to affect her score, it’s useful to be as clear as possible about what’s going on. As with other rewards and contextual cues, visual feedback is crucial, but a straightforward explanation of scoring can be just as helpful.

Like some of the systems of reward we talked about earlier, score often exists outside of the system of verbs, objects, and choices that the player pushes through while playing the game. One of the things that makes score more interesting as part of the conversation, and not simply something external, is that it takes its meaning from players’ own understanding of how it’s important—or unimportant. In a competitive tournament where all players have agreed that the highest-scoring player is the victor, comparisons of score are crucial; the score is imbued with a lot of significance. In a single-player game, it’s up to the player herself whether she wants to pursue higher scores to beat her friends or her own past best efforts. Scores are an evaluation and not simply a reward that lures the player on. They’re useful as tools to help the player gain her own insights into how and why she’s playing.
Although giving higher scores for desirable actions and lowering scores for undesirable actions in a game is the norm, it’s even possible to create scoring systems that don’t necessarily favor one style of playing or action over another. In *Wonder City* (2013), a game I helped create, the player guides the actions of a character who’s just discovered she has superpowers. Through a series of decisions, the player decides how this superheroine will relate to other characters, protect her secret identity, and deal with tricky situations (see Figure 6.14). There’s no single score that determines how well a player did; instead, the game keeps track of the kinds of choices the player makes. Does the heroine use her powers at every opportunity or try to solve problems in other ways? Does she collaborate with her friends, or is she more of a loner? After each chapter of the game, the player sees icons that represent a style of heroism: direct or indirect, powerful or restrained, collaborative or independent, among others.

![Figure 6.14 Making decisions in Wonder City.](image)

Although the game’s system keeps track of these styles with a series of numerical scores, we actually opted not to show the numbers to the player. In *Wonder City*, we wanted the game’s conversation with the player to focus less on trying to affect the score and more on the player’s gut feeling about how she wanted her character to behave in each situation that requires a decision. The result is a scoring system that’s more like a personality test: there’s no better score, only different scores that reflect ideas back to the player about the style she’s created through a series of cumulative choices (see Figure 6.15). Like the other parts of a game’s resistance that we’ve discussed in this chapter, score can be shaped in many different ways, depending on what kind of conversation you’re trying to bring about.
Review

- Difficulty is a traditional way of talking about a player’s journey through a game: games start off easy, players deepen their skill and understanding through playing, and then they take on more difficult challenges. In a simplistic version of this journey, the most difficult experiences are the pinnacle of what a game can offer. But this is just one way of looking at the conversation of playing a game.

- When we talk about resistance, we mean the many ways that a player can push into a game by using verbs and making choices, the ways that a game pushes back by presenting challenges to overcome and consequences for actions, as well as the ways that a game can pull a player with rewards or open up wider spaces for the player to push into. Resistance shapes the conversation of a game via a system of rules that the player can push against.

- Flow is another popular way of talking about the way players encounter resistance in a game: an engaging experience that hovers between frustration from high levels of resistance or difficulty on one side, and boredom from the repetition of already-mastered tasks on the other. Different games have very different kinds of flow. Some game creators try to create perfect flow, never allowing frustration or boredom for long. There are many ways to make interesting games without perfect flow, such as games that challenge players from the beginning and ask them to overcome frustration.
Whether you’re thinking about games in terms of flow, difficulty, or resistance, you can shape the push and pull of your game’s conversation in many ways: by pacing the development of verbs as discussed in Chapter 3, “Scenes,” or by opening up the pacing to player input, such as by letting her select a difficulty setting.

Game designers who strive for perfect flow sometimes use Dynamic Difficulty Adjustment (DDA), techniques that help a player who is struggling with frustration or make things harder for a player who is getting bored. These techniques are tricky, however, because it’s hard to guess what a player’s experiencing. Even if you can guess, DDA can lead to the game feeling mushy, adjusting itself to everything the player does without providing a firm structure that the player can push against to understand its rules and challenges.

When you let players have a hand in shaping the resistance of your game’s conversation, they can be part of deciding how and when their experience gets more challenging. Games can let players choose how quickly to move to a more challenging space within the game or whether they use the verbs that require more skill. Games with very open spaces of resistance let players choose their own goals as well—or even make up goals to accomplish within a system, as in sandbox games.

Rewards are a way to shape resistance that pulls players forward, encouraging them to pursue goals or repeat actions. Some rewards operate within the system of a game, opening more opportunities for players to push. For example, rewards can unlock new verbs to use or new areas of the game to push into. Other rewards don’t feed back into the system, such as story cutscenes or traditional achievements that exist on a console platform. It’s worth thinking about whether the lure of this kind of reward can eclipse the pleasure of playing a game for its own sake.

Resources are a kind of reward that enables other verbs. For example, ammunition is a resource that lets the player use the verb shoot. Resources are flexible because they’re represented as a number that the player can manage through spending or saving. They can be linked to goals or punishments. In many games, if you run out of a resource called health, you experience a setback or have to start the game over.

Punishments push hard against the player when they do something the game deems wrong, often involving a penalty that increases resistance or a setback that requires the player to repeat some or all of what she’s done before. Repetition can be a useful form of punishment, especially for games that try to maintain flow, where a resistance-increasing penalty would make the game more frustrating. Repetition of a challenge that hasn’t been mastered can give the player a chance to practice, to experience the same situation again, make different choices, and seek a different outcome that leads to reward as opposed to punishment.

Repetition of tasks the player has already mastered is often called grinding and is sometimes used in a game for its own sake. Grinding over long periods of time creates a different kind of resistance that requires endurance and patience to overcome.
boredom, rather than challenges to overcome frustration. Some players enjoy this kind of resistance, but many games use it simply to increase the amount of time the player must invest to reach a goal. Many players resent this kind of tactic to keep them playing.

- Score is not inherently a reward or a punishment; it’s an evaluation where the player can reflect on what she’s done. In creating a score system for a game, you’re telling the player what kinds of actions are desirable or undesirable in the conversation. A player can compare a score to her own previous scores or to other players’ scores to understand how she’s doing. Scores can be open to player interpretation as well. Especially in single-player games, players can choose to disregard the evaluation of score. It’s possible to create scoring systems that don’t value one type of action or decision more highly than another but simply let the player reflect.

**Discussion Activities**

1. Think about experiences you’ve had playing games that were very challenging, frustrating, or difficult. Now think about some experiences that were very easy—even to the point of boredom. Compare your experiences with others. Are there particular kinds of gameplay that you find challenging or easy? Why?

2. Imagine that you’ve been asked to make a game based on something that happened to you, or some aspect of your life. What would you draw from your own experiences to make a game about? How could you express something about your life using systems as well as contextual elements like words and images?

3. Using the scenario you designed in Discussion Activity 5 of Chapter 3, discuss how you could change the rules of the scenario to increase the resistance and difficulty. Also discuss how you might raise and lower resistance to create different kinds of pacing from this list:

   - Pacing based on the classic idea of flow, where a game starts off easy, introduces the player to new verbs and objects, and builds in challenge to avoid boredom.
   - Pacing that starts off difficult, demanding a lot of the player, and only becomes less frustrating through practice.
   - Pacing without any flow at all. Think about other ways that a game’s shape might change over time that aren’t about building skill and rising challenge.

4. With that same scenario, are there additional rules you could add that let players have a hand in whether the game is harder or easier? How would you open the conversation of your game to include player choices about their own goals in the game?
Group Activity

For this activity, you’ll once again be using Knytt Stories, the game creation tool described at the end of Chapter 4, “Context.” Split into two teams (even a team of one is fine!) and design a level in Knytt Stories. When you’re done, have the other team play it, and watch them playing. Don’t say anything while they’re playing, but keep track of your own reactions as they move through the scenes you’ve created. How does watching other people play make you feel about what you created?

Using the preceding process—creating a level, and then watching someone else play it—try out the following ideas:

- Make an easy game that just introduces a new verb.
- Take the easy game and see if you can make it more challenging by increasing the resistance somehow. Can you increase the resistance by developing verbs? How about just with objects?
- Extend the game so that it has periods of high resistance (for example, a difficult section) and periods of low resistance (a section where the player can pause and relax or just engage in rote activities she’s already mastered.
- Make an extremely high-resistance game—not impossible, but one that you’re not sure if the other team could actually beat. See what happens!