Advanced Prototyping

Chaim Gingold

Chris Hecker
Have Idea
Ask Questions
Prototype
Develop
Sell It
Have Idea
Ask Questions
Prototype
Develop
Sell It
Why Prototype?

1. Answer questions
2. Find upside & downside
3. Persuade and inspire
Prototypes do not generate ideas from scratch!
Bigger Projects, Too
agile

cheap + fast

validation

problems

solutions

slow + expensive
Metrics
Cheap

agile

light
Falsifiable

make a claim

testable
tested

learn
Our Atmosphere is Collapsing!
Please help fix it!
Time is running out!

Right Away!

No.
// First hello.
group GreetFirstContact
{
  // Warm welcome
dialog
  {
    motive Default;
    text "Welcome!"
    button
      {
        text "I come in peace.";
        do RelationshipPlus;
        do SayDialogFirstGroup ReactToIComeInPeace;
      };
    button
      {
        text "Prepare to die.";
        do RelationshipMinus;
        do SayDialogFirstGroup ReactToThreat;
      };
    button Bye
  };
};
Relevant generalizable
Surprising feedback upside and downside inspiring
Persuasive

fun
tangible
clear
disruptive
Cheap + Persuasive =
Design Doc = Prototype?
What is your Question?
Q:

Can we make a fun social game between characters?
Q:

Is Leg UI concept user friendly, powerful, and hot?
Q:

Can rolling around with a sticky ball be compelling?
Q:

Here’s a design doc for a game. Is it going to be fun?
“So you want to write a prototype?”
Step 1:
don’t
steal it
fake it
rehash it
Step 2: 
permission vs. forgiveness

“if it takes less than two days, just do it”
Step 3: fail early
Step 4: gather reference material
Decomposition
What’s in?
What’s out?
Connections
creature editor

Creature Game

Economy

Object Composition

(Animator, Skinner, Painter, Stitcher)

Legs

Torso

Parts

Palette
prototyping:

- constraints
- +
- freedoms
- +
- assumptions

“what’s really important?”
Economics
where do you spend your resources?

interactivity
robustness
hotness
usability
fun
duration
beauty
performance
variety
agility
velocity
quality vs. cost curve is [very] nonlinear
quality vs. cost curve is [very] nonlinear
code vs. content
code

interactive

autistic

code

content

dead bits

emotional
Our Atmosphere is Collapsing!
Please help fix it!
Time is running out!

Right Away!
No.
you understand your problem

you are still clueless
Only spend code where you need understanding; throw content at the rest.
Programming
What’s important?

agility
&
velocity
What’s not important?

- robustness
- elegance
- optimal code
Be Lazy

Computers are *fast* and **HUGE**
Don’t commit to an abstraction!

Code influences your mental model. Must stay agile.
Don’t commit to an abstraction!

**toolkit vs. framework**

- recombinant
- compositional
- immediate
- delicatessen

- static
- controlling
- retained
- prix fixe dinner
Tower of Tuning

scripting
hotloading
data driving
interactive editor
recompiling
Collaboration
Designer-Programmer
Designer

and

Programmer
Feedback
Demo

Persuade
Buy-In
Harvest Good Ideas
Test

Validation
Quiet!
Observations
Archive It.
Repeat.
the end