

Project Preparation

DR. ROBERT ZUBEK, SOMASIM LLC

EECS-397/497: GAME DEVELOPMENT STUDIO
WINTER QUARTER 2018
NORTHWESTERN UNIVERSITY



NORTHWESTERN
UNIVERSITY

Overview

1. How to get started on designing your game
2. How to prepare for your project proposal

Let's recap our model so far

Mechanics

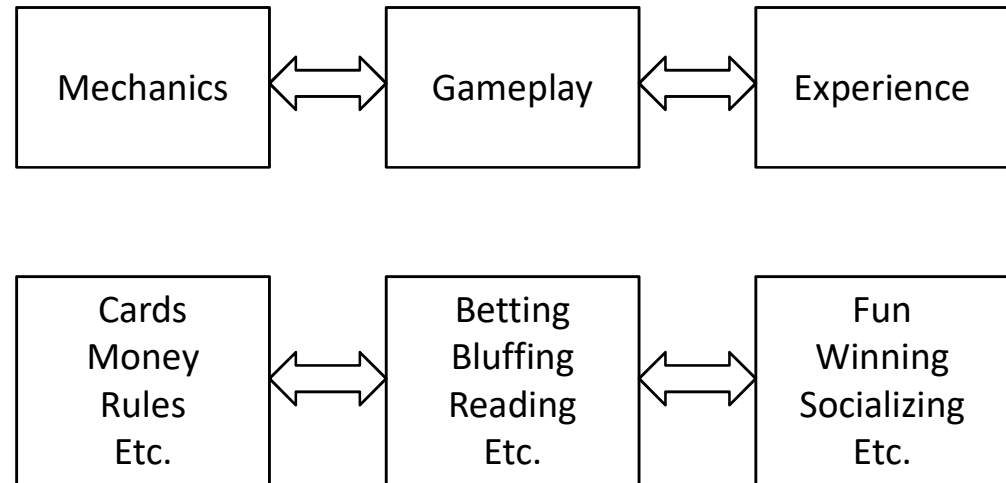
- Game elements, rules, code
- Player's inputs and outputs

Gameplay

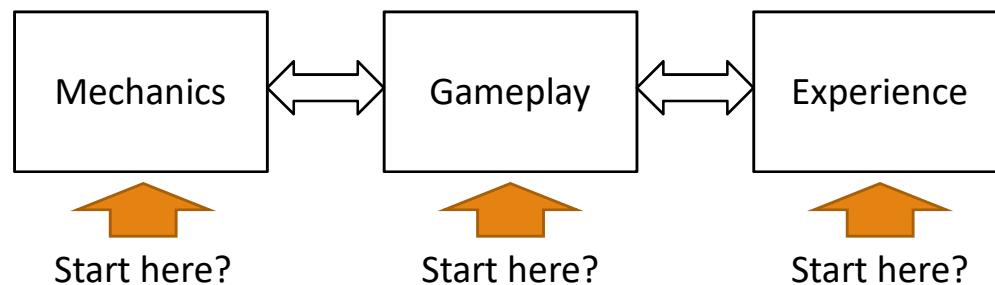
- How the game unfolds over time
- Activity / behavior/ patterns of play

Experience

- The feels / the fun



How do we start designing a new game?



Product design

Design is a **high-dimensional search problem**

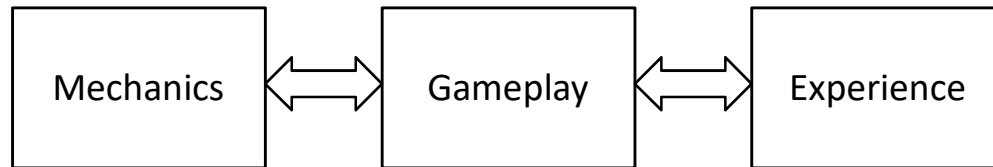
- We need to find a good solution to a set of constraints
- Not just in games!

Practicing designers pursue **exploration of the space of possible designs** using **prototypes** and **experience** as guides

Two broad families of approaches:

- Top down design
- Bottom up design

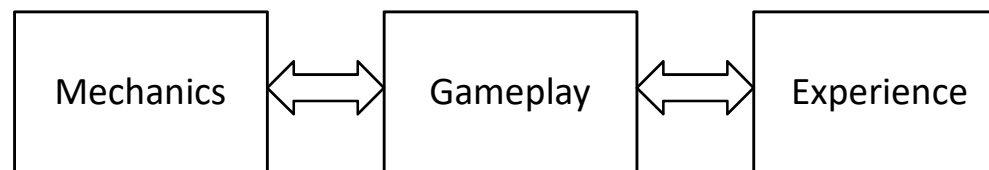
Product design



Top down

Start with desired experience, player archetypes, market analysis, etc. Figure out how to make a good experience for them.

Product design



Top down



Bottom up

Start with gameplay and the experience
it generates, let it “find” an audience

Product design

In this class, you will

- Start with a **broad top-down vision**, and then
- Work mainly **bottom up to prototype it**
 - Something small mechanically, iterate on it, and build it up

Top down and **bottom up** design often happens *at the same time*

- Decision about mechanics → affects gameplay → affects experience
- Decision about player experience → affects gameplay → affects mechanics
 - ... and don't get me started about *product* constraints like budgets, tech limitations, etc

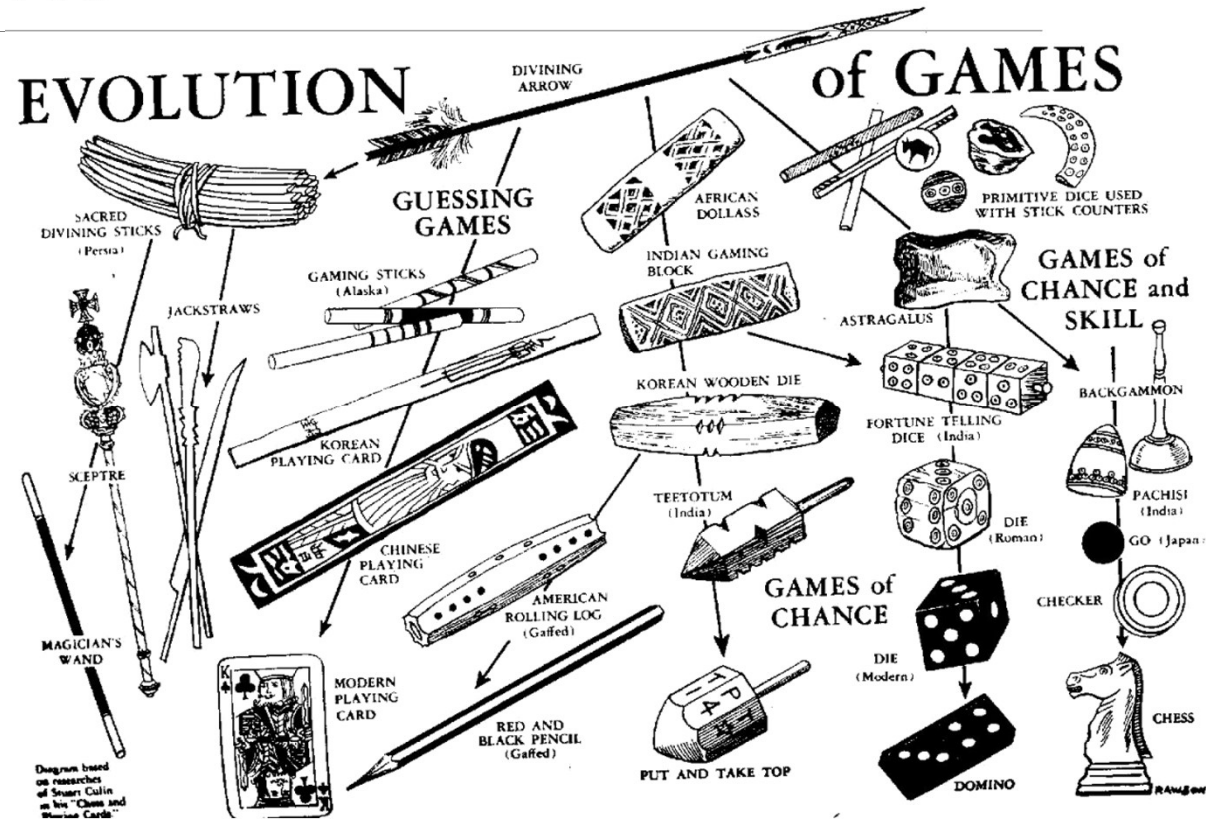
Evolving mechanics

Most new games start out inspired by what we already know **and that's okay :)**

- Chess is a masterpiece that evolved over centuries

So start from a known game and evolve it

- But in a smart way :)



Hierarchy

A game

A reskin or clone

A variant

A type

A genre

(The following examples courtesy of Raph Koster)

Hierarchy

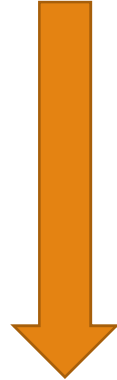
A game

A reskin or clone

A variant

A type

A genre



a unique construct

changes presentation or content

changes some rules

a set of related variants

a set of variants that share core mechanics

Hierarchy

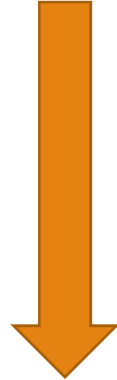
A game

A reskin or clone

A variant

A type

A genre



Straight poker

Wild cards

Stud poker

Poker

Hierarchy

A game

A reskin or clone

A variant

A type

A genre



Straight poker

Wild cards

Stud poker

Poker

Castle Wolfenstein

Spear of Destiny

Doom

Deathmatch FPS

First person shooters

Hierarchy

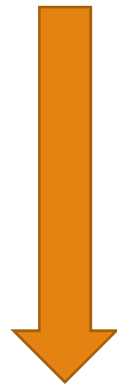
A game

A reskin or clone

A variant

A type

A genre



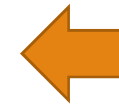
a unique construct

changes presentation or content

changes some rules

a set of related variants

a set of variants that share core mechanics



Lots of game innovation
happens on this level

... so, how do we do that?

Modeling gameplay



"Gone Home"

Walk around an empty house filled with mementos.

Read notes and examine clues as you find them.

You can only move on when you find a critical clue.

Modeling gameplay

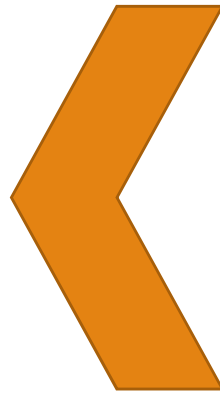
Card Game

Take a deck of cards, lay them face down in four groups.

Flip cards in a group in any order you like.

You can only move on to next group when you find an ace.

MORE ABSTRACT



"Gone Home"

Walk around an empty house filled with mementos.

Read notes and examine clues as you find them.

You can only move on when you find a critical clue.

MORE CONCRETE

Card Game

Take a deck of cards, lay them
face down in a row.

Flip them over in
any order you like.

You win
when you find a
critical clue.

MORE ABSTRACT

"Go
Wait
filled with mementos.

Read notes and examine clues
as you find them.

You can only move on
when you find a critical clue.

MORE CONCRETE

1

Start with an existing game

2

Identify mechanics abstractly

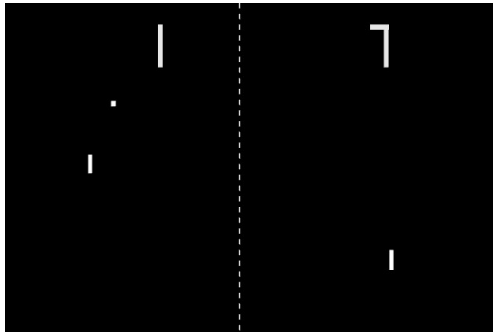
3

Modify mechanics in interesting ways

4

Apply to a concrete setting

Nothing starts *ex nihilo*



Pong (1972)

Player moves up-down

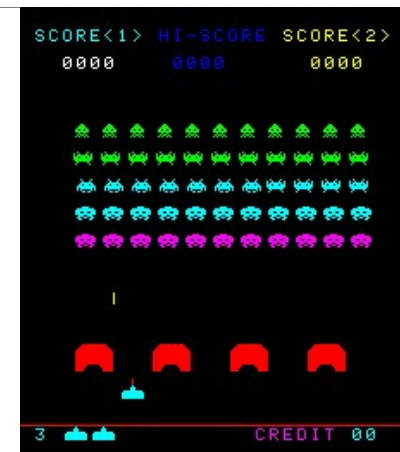
Bounces a ball at other player



Breakout (1976)

Player moves left-right

Bounces a ball at bricks arranged in various patterns

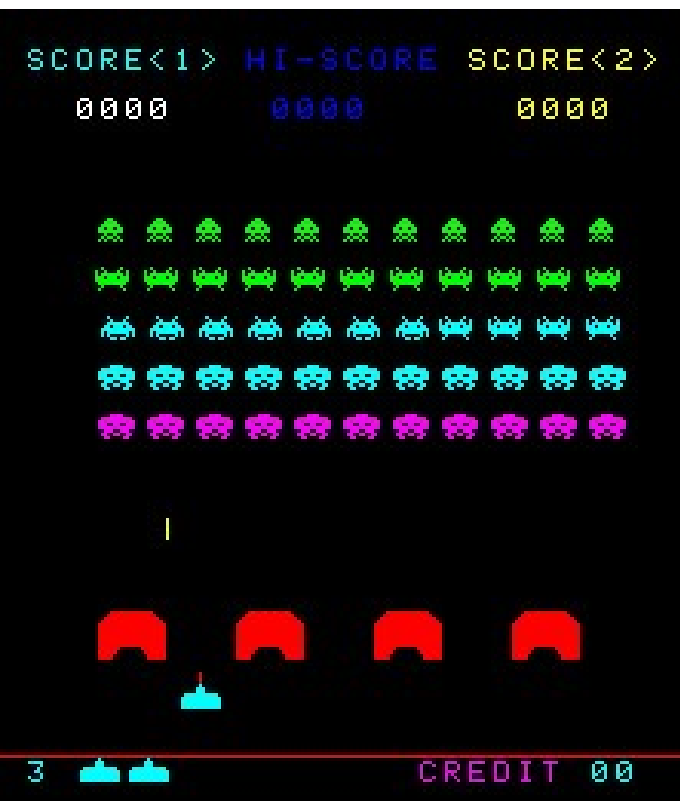


Space Invaders (1978)

Player moves left-right

Shoots at baddies, baddies shoot back and come down

Modeling gameplay



Space Invaders (1978) – shoot down all baddies before they shoot or touch you.

Player moves left-right

Baddies follow a path (zig-zag down)

Shoot upwards

Modeling gameplay



Galaga (1981) – shoot down all baddies before they shoot or touch you.

Player moves left-right

Baddies follow a variety of paths!

- zig-zag down as a group
- break out and fly at the player
- streams of baddies in a formation

Shoot upwards

Modeling gameplay

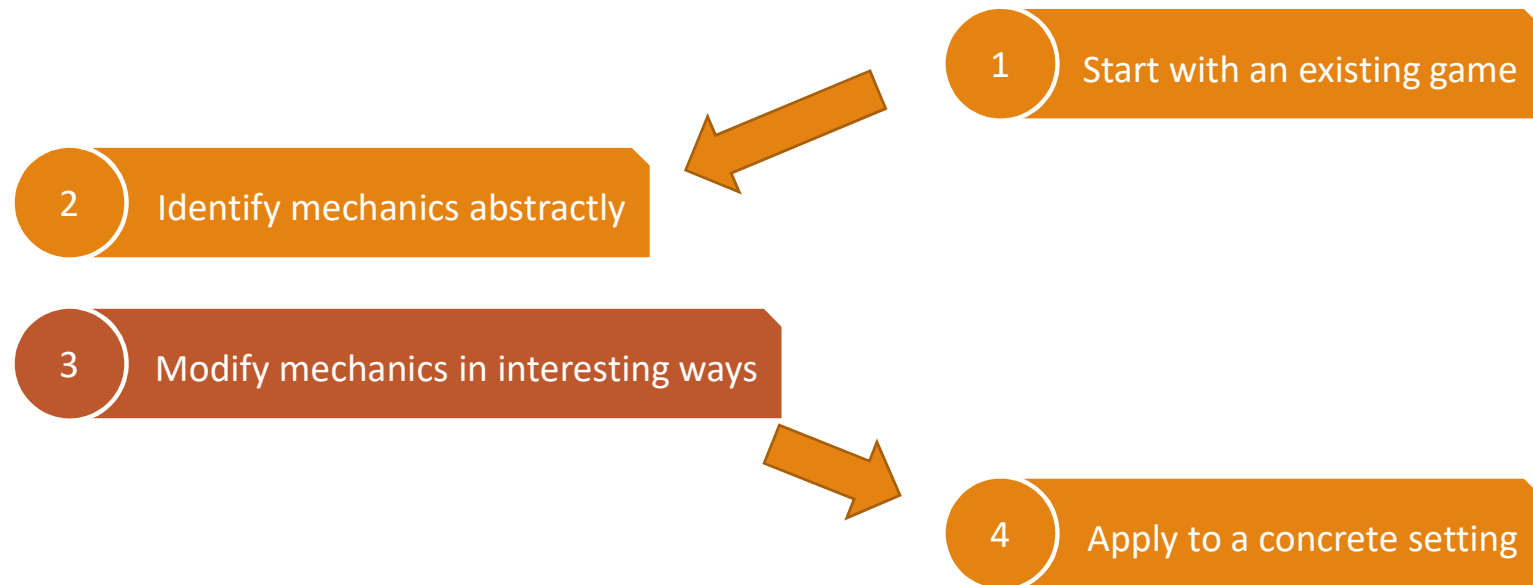


Gyruss (1983) – shoot down all baddies before they shoot or touch you.

Player moves in a circle around

Baddies fly out from the center
- spiraling out in formations
- individual bombs etc.

Shoot inwards to the center



Modeling gameplay

Games are made of mechanics

We can change existing mechanics...

We can bring in new ones...

We can rearrange them in unexpected ways...

(The following examples courtesy of Raph Koster)

Changing mechanics

1. Simplest change: **add a stat**, and some **rules** that use this stat

- *Action cooldown → requires planning and defense*
- *Number of moves → efficiency and planning ahead*
- *Different attack / defense ratings → possibly divergent strategies*

1b. Remove a stat! More difficult :)

Changing mechanics

2. Add new goals

Take a platformer

- Add “best time wins” goal. You get speedruns.

Take a side-scrolling shooter

- Add “rescuing” goal. You get the classic Defender.

It gets interesting when the goals are contradictory...



Changing mechanics

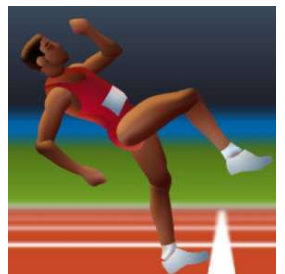
3. Change / extend verbs

<i>Game</i>	<i>Actions</i>	<i>Movement</i>
<i>Rock paper scissors</i>	<i>3 attacks</i>	<i>None</i>
<i>Karate Champ</i>	<i>13 attacks, 4 defense</i>	<i>Left/right</i>
<i>Karateka</i>	<i>6 attacks, 2 stances, bow</i>	<i>Left/right + scrolling</i>
<i>Battle Arena Toshinden</i>	<i>> 50 attacks, 2 stances</i>	<i>Left/right (2 independent)</i>
<i>Bushido Blade</i>	<i>8 weapons, many moves</i>	<i>Left/right + depth</i>

Changing mechanics

4. Change input mappings

- Switch from analog (continuous) input to binary one → Asteroid-style thrust
- Switch from instant input to timed input → “Charge up” actions
- Switch from simple input to complex motions → Wii motion sensors
- Split simple input to several inputs → QWOP



Changing mechanics

Several places to mess with input mappings:



Changing mechanics

5. Mash together games with similar systems

- Mario meets Joust
- Breakout meets Space Invaders
- Twister meets Tic Tac Toe

Do you pick mechanics from one and add to the other?

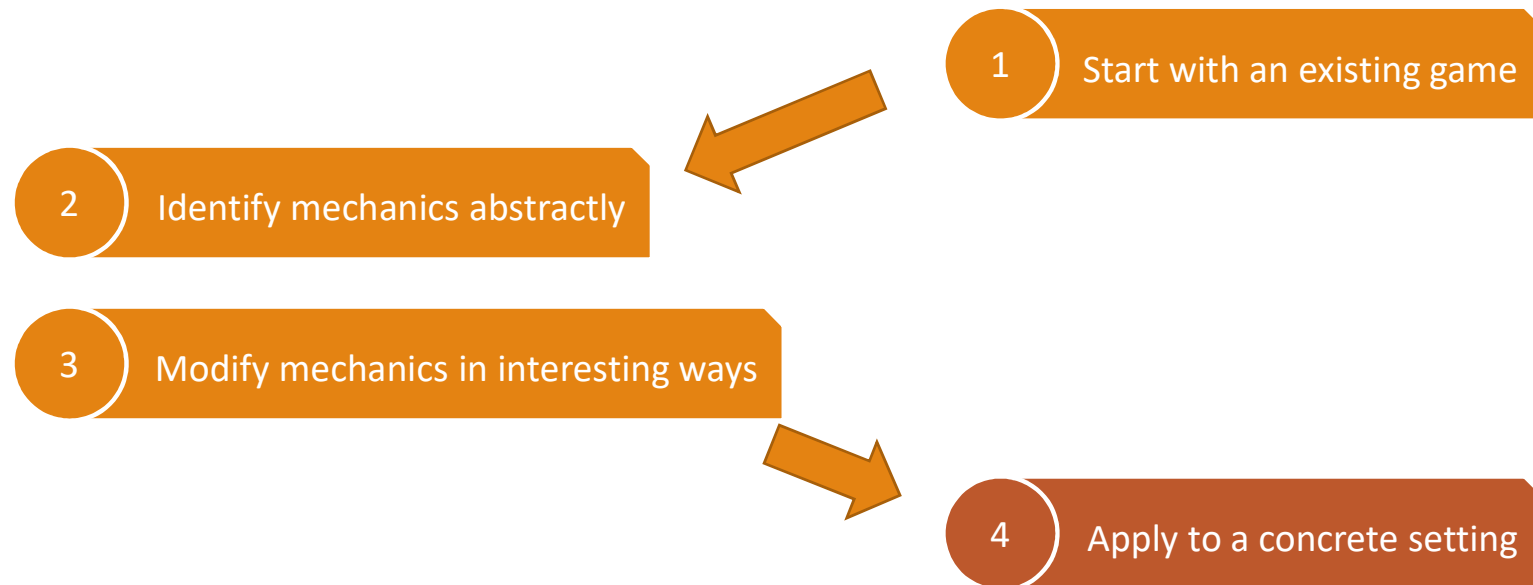
(And how will it change the game?)



Changing mechanics

6. Replace mechanics with unexpected alternatives

- Can you do a territory based game... with cards?
- Can you do health... with territory?
- Can you do aim... with tokens?
- Can you do jumping... with dice?





Starting on your project

This week:

- Split up into groups (3 people each)
- Come up with project ideas
- Write up a project proposal

Next week:

- Present your project idea to the class
- Finalize project plan
- Get started!



Project proposals

Make a plan of where you're going,
before you take your first steps

Get your classmates excited about your work

Let me know what you're going to make



Communicating your design

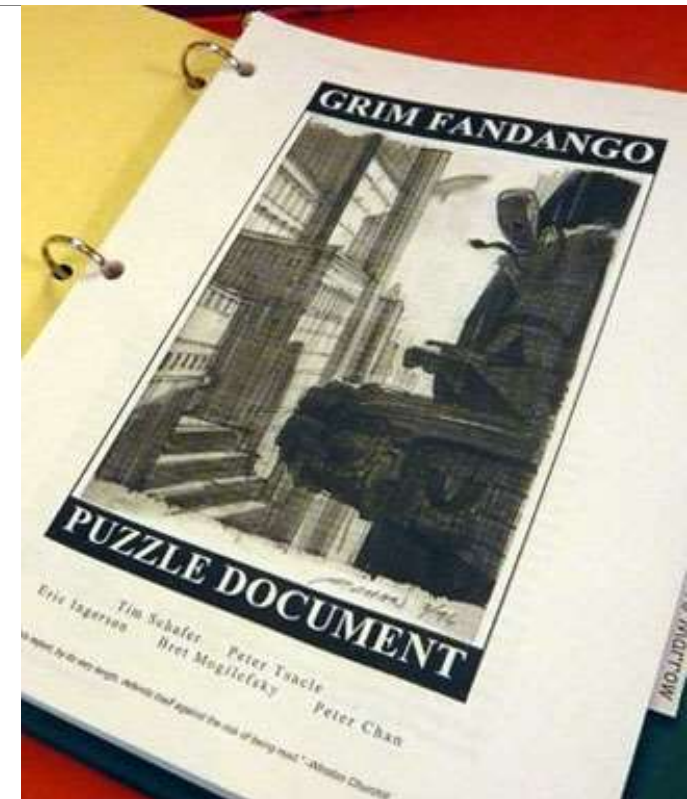
Design document

How do we communicate this design to others?

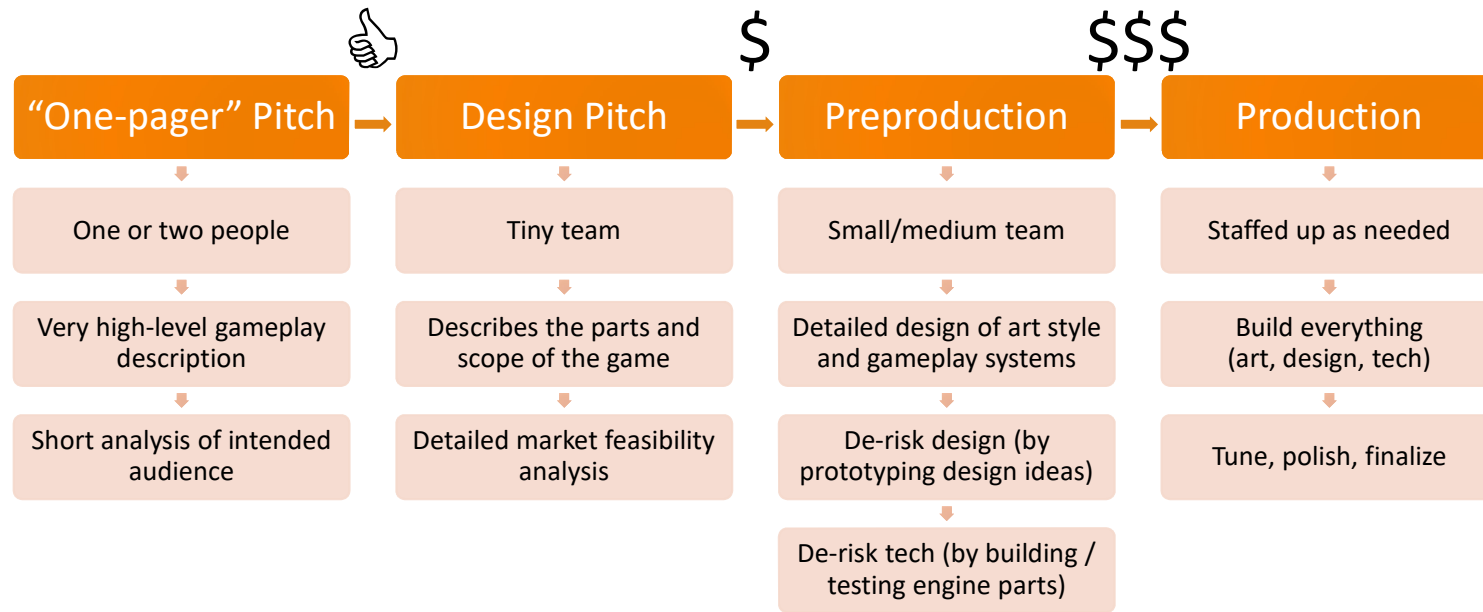
- We could write it all down...
- ... but design will evolve over time

GDD (game design doc), or more often a Design Wiki

- “Necessary evil” with large teams where everyone works on a tiny part of the whole
- Used for pitching to publishers etc.



Typical “corporate” greenlight process



Design document

GDD is complete overkill for small teams

Many indies start with “prototype first” and let design evolve naturally

- But that causes problems too: easy to lose direction and get lost

The Tadhg Kelly model

- GDD -----★----- Prototype First
- Do prototyping once you’ve created a very brief design pitch
- There are some specific questions he likes to see addressed

Tadgh Kelly's "Raw Game" approach

(From GDC Next 2014 talk)

Q1: Is the game's fiction urgent?

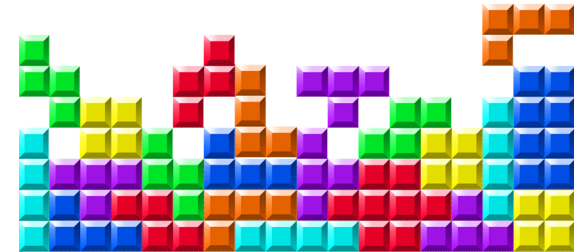
Not just the game story, but the game's *aesthetics* – the player's experience of playing:

Tetris: OMG THESE BLOX JUST WONT STOP FALLING

Minecraft: the world is dangerous, and you need to make shelter and everything else

The Sims: struggle to remain happy, be successful at work, and have friends

Note: these fictions can be realized with all sorts of mechanics!



Q2: What is the player's job?



Not the avatar, or the story character – the player!

Keep it short. One word. Maybe a short phrase.

Tetris: I'm a sorter

Minecraft: I'm an explorer and a builder

The Sims: I'm a working adult


Q2: What is the player's job?

Very helpful in defining boundaries

- If I'm a sorter, I'm probably not a soldier
- If I'm an explorer, I'm probably not a manager

Gives player clear context of what's expected of them

What if you have more than one “role”?

- Probably several games in one
- Proceed with caution! There be dragons 



(“The Movies”, Lionhead 2005)

Q3. What are the verbs?

By verbs we mean individual units of in-game action.

- not physical actions like swipe, touch, or click

Fighting games: punch, kick, block, jump

Tycoon games: build, collect rent, buy, sell

Minecraft likes: go, dig, collect, craft, attack



Q4: Do actions support each other?

If actions are completely independent, they'll get boring quickly.

It's better to make them support or chain together

Soul Calibur: combos like jump + kick

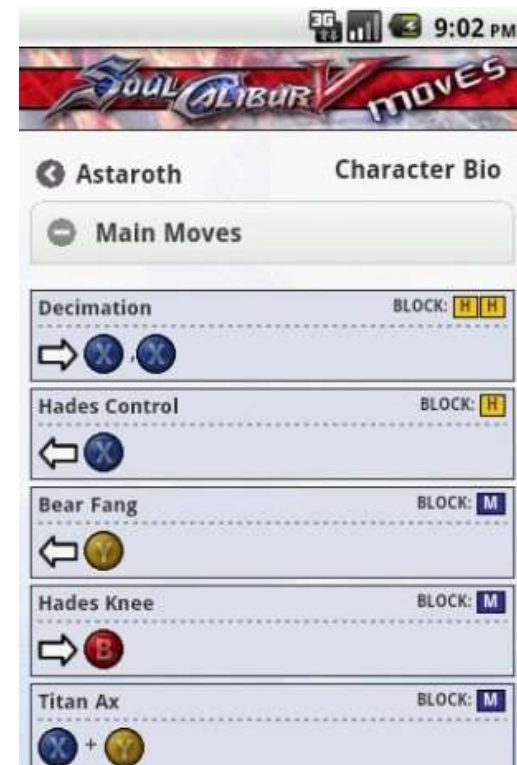
- Gives it extra strength, but takes longer

Diablo: loot and economy

- Planning how to convert items and resources

FPS: weapons and upgrades

- Choice of weapon and add-on becomes strategic



Q5. What are your mechanics types?



Going back to our examples of mechanics:

- Board / space
- Resources
- Units
- Uncertainty
- Physical actions
- Progression
- Etc.

Q5. What are your mechanics types?

Some examples:

FPS: Guns, bullets, health

- Different guns have different speed and damage; you need to find bullets to shoot, etc.

Tennis: Ball, aim, court position

StarCraft: base, soldiers, time, board visibility

What are yours?

- Don't have to have all of them, but have a variety



Q6. What's the defining rule?

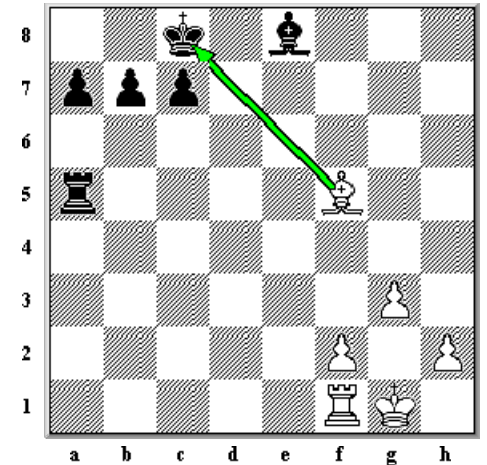
Is there one rule that makes the game unique, and without it, it wouldn't be the same?

Chess: putting the king in check

FarmVille: making the player wait

Roguelikes: risk of sudden death

("Rule" is deliberately vague in this context)



Q6. What's the defining rule?

You can make the game more interesting by adding player actions that support or frustrate the defining rule

Example: **Mario Kart**

- The defining rule is driving quickly and accurately
- The game is full of elements that mess with it (offense + powerups)

If you evolve your game from an existing game...

- See if you can identify some defining rule (or rules)
- See if it's interesting to add game elements that mess with it



Q7. What are you testing?

In the sense that the game is always
“testing” or “challenging” the player at something

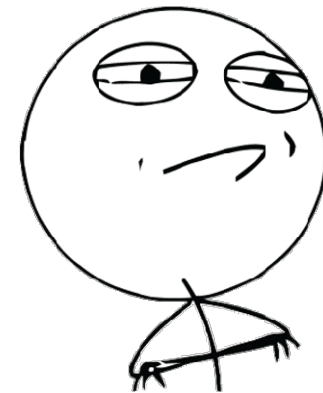
Quake: fast reflexes, accuracy

Tetris: sorting shapes quickly

StarCraft: managing a zoo of units

What is the challenge your player will be facing?

CHALLENGE ACCEPTED



The 7 Questions Wrap-Up

The goal is not to get them right immediately

The goal is to explicate your design assumptions – it's to make the designer think and plan

Q&A

Remainder of this week

Until end of this week:

Organize into teams of ~3

- Feel free to use Canvas / Piazza discussions to discuss ideas and recruit team members

Settle on the project idea

- Something that fits into one quarter :)
- Optimize for the first ~5 minutes of gameplay

Homework due EOD Sunday 1/28

1. One-pager pitch doc for your classmates

- Your team name
- Your team members
- What's your game idea
- What's cool about it
- Submit on Canvas
- I will share everyone's pitch docs with the whole class

2. Project details doc for me

- Your team name
- Your game idea
- Design details:
 - Identify your inspiration (other games etc)
 - Answer each of Kelly's "7 Questions" for your game
 - Identify player progression mechanics
 - Identify sources of uncertainty (see Costikyan)
- Also submit on Canvas

Next week

On Monday 1/29:

Each team will present their project in class

- One person per team does the presentation
- Quick pitch – 3 minutes max
- You can use the projector if you'd like

On Wed 1/31:

Lecture on “Game Production Process”

Start prototyping on the project!

Also I'll ask for an updated project proposal
(details next time)