

# Computer Games Laboratory

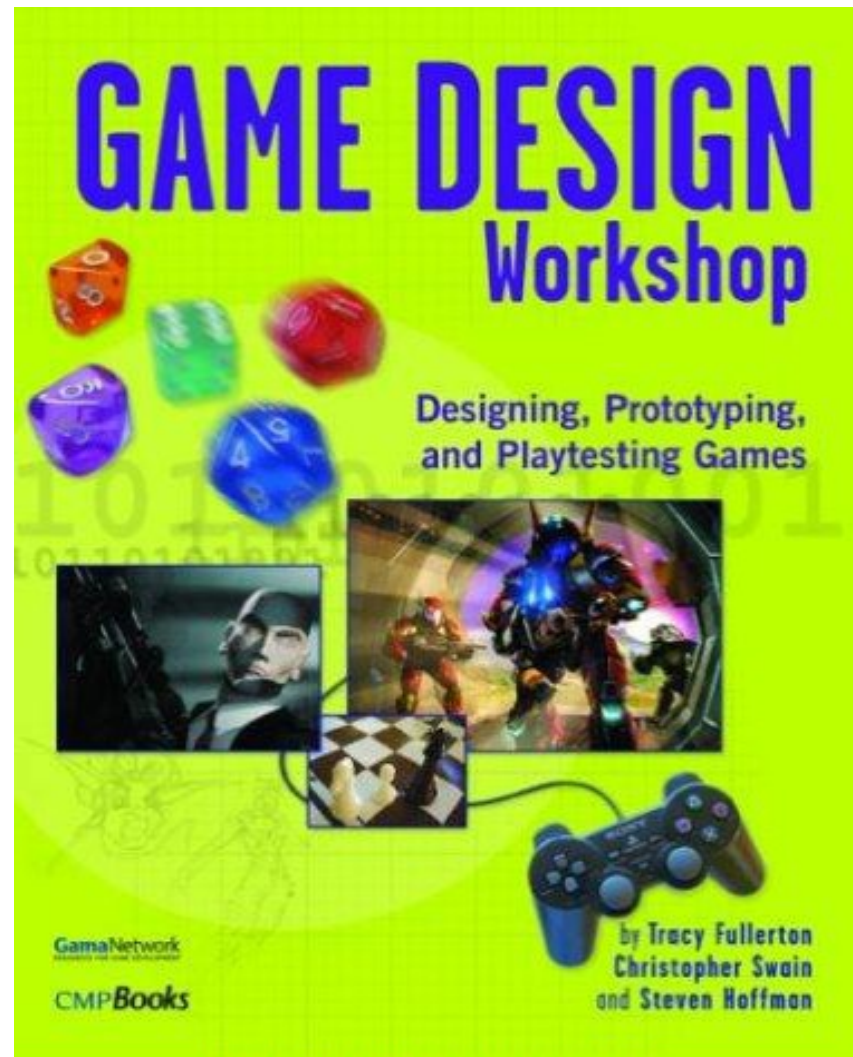
## Prototyping



# Prototyping

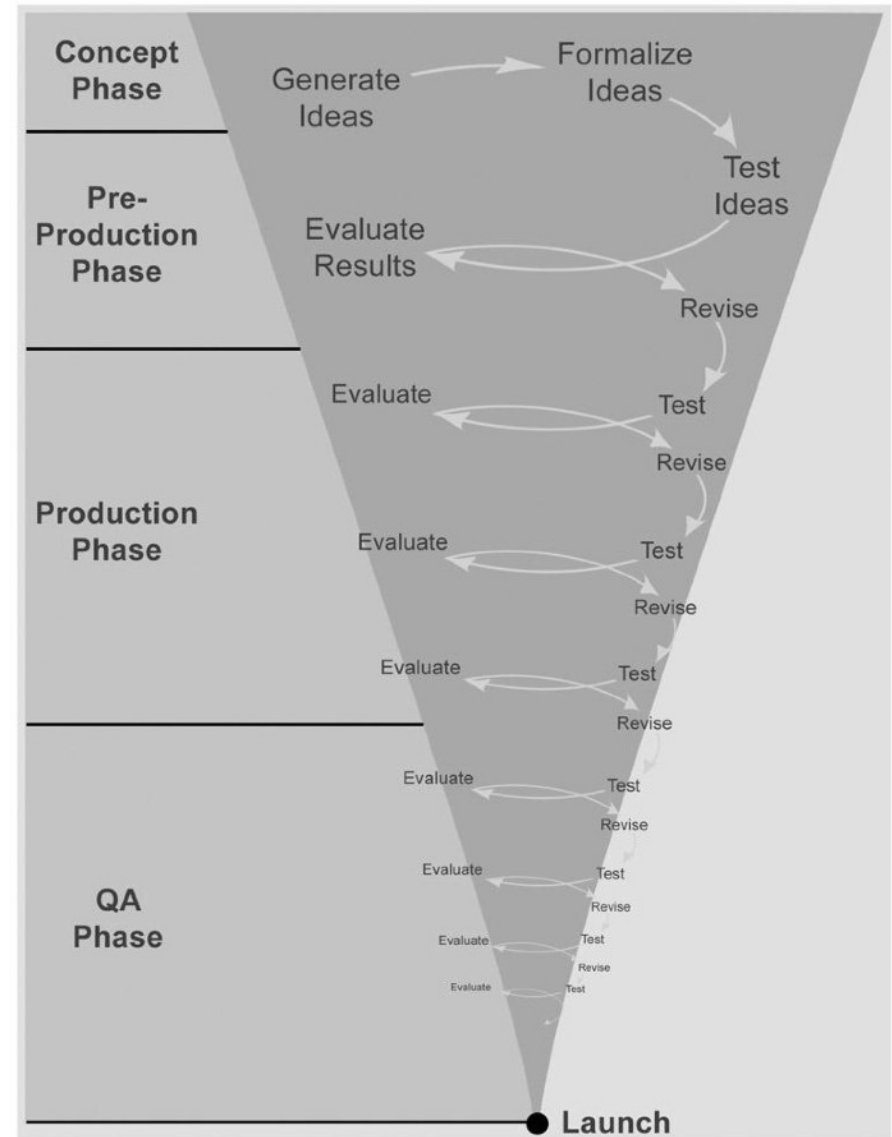
Recommended Reading:

Chapter on Prototyping



# Prototyping

- Why a prototype?
- “Creating a game without a prototype is like shooting a movie without a script.”
- A prototype adds more to a game than a script or doc can do:
  - Interactivity
  - Test and exploration
  - Modification



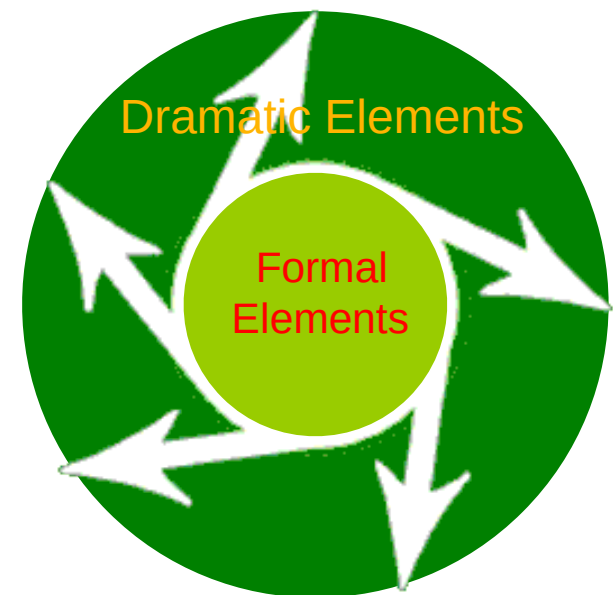
# Prototyping

- Prototype

- Something fast and cheap that allows you to answer a specific question about your game: “Is your game idea fun?”
  - Not something that eventually morphs into a game
  - Not something using the same technology as the production code
  - Not meant to impress others
  - Rather some form of scientific exploration

# Prototyping

- Prototype
  - Only fundamental mechanics (formal elements)
  - Considers player(s) and the computer
  - Rough approximation of artwork and features
  - Focus on gameplay, abstract from production process
  - Extensible, instrument for radical changes



# Prototyping

- The purpose of a prototype
  - Do not create a prototype to show something about the game
    - Rather validate or disprove some concept
    - Low effort
  - Test idea early on, before spending lots of effort on implementation
  - Prototypes don't generate ideas, they **validate** them
  - Find upsides and downsides
  - Experiment, persuade and inspire ...

# Prototyping

- The purpose of a prototype
  - Define core gameplay elements in purest form
  - Learn whether core mechanics hold interest of players
  - Test e.g.:
    - Game mechanics
    - Balance of rules (too restrictive, too loose, too few, too many)
    - User experience
    - Embedded technologies
  - Discover play patterns and emergent behavior specific to your game

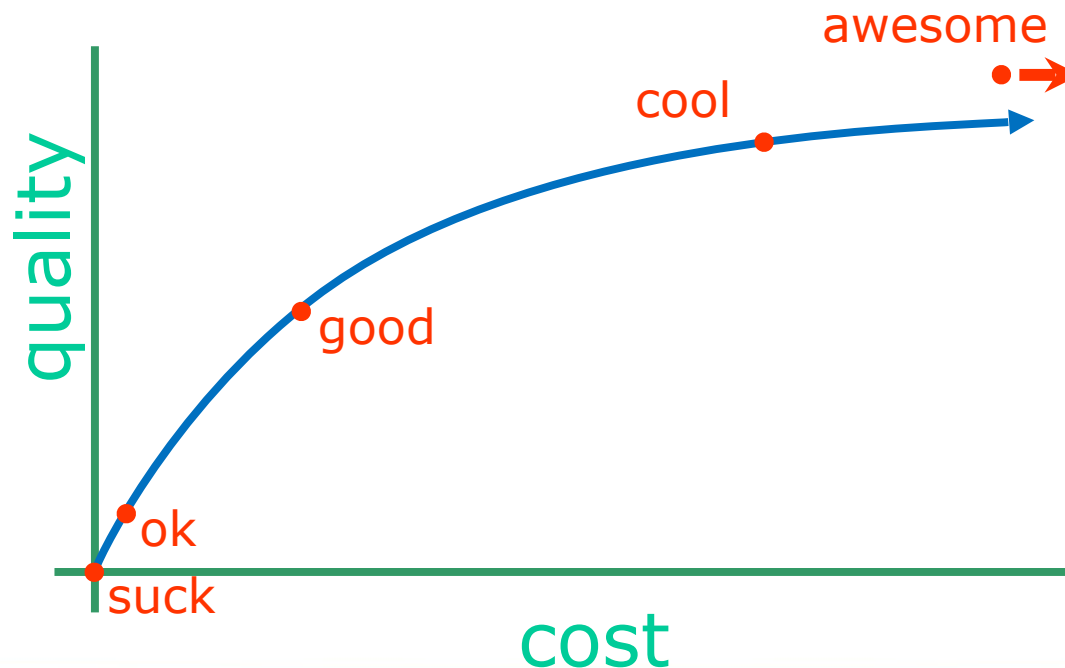
# Prototyping

- Question to be answered by the prototype
  - A good question is concise and can be answered in a fairly unambiguous way
  - Ask yourself where you need understanding
    - Not necessarily full game
    - Focus on central gameplay component(s)
  - Can be about game design, or any other aspect of the game
    - How to make things look heavy by coloring
    - Can I control a pen via my iPhone
    - Does this sound bring me into the mood, etc.
  - Do not try to ask for a good game idea via a prototype
  - Decompose a big problems into smaller tractable ones



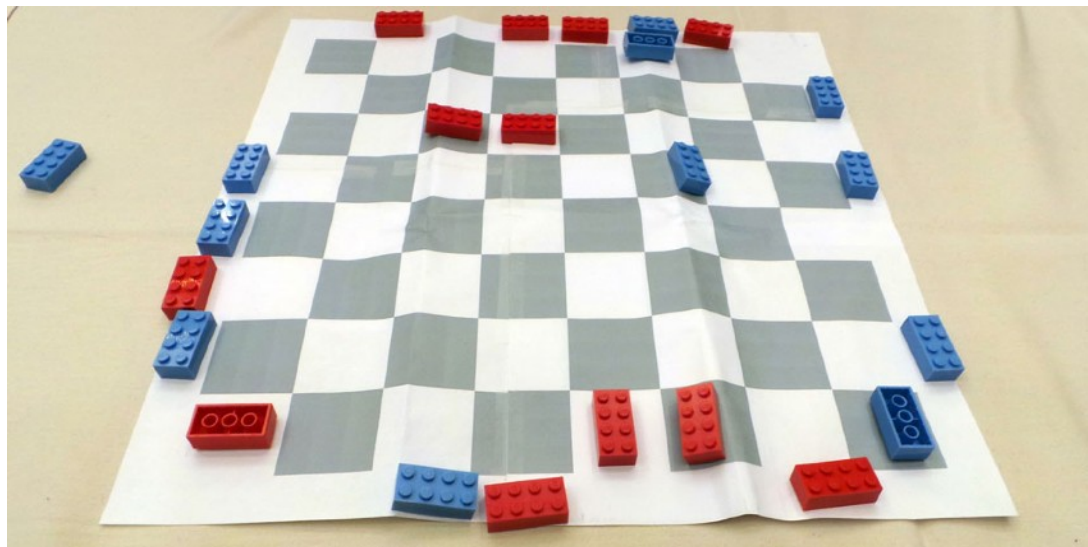
## Prototyping

- Measure the quality of a prototype
  - Find relevant characteristics; interactivity, robustness, usability, beauty, performance, agility, and many more
  - Evaluate per characteristic quality/cost



# Prototyping

- How to generate a prototype
  - Look for the cheapest way to get it
  - Steal it or fake it
  - Use available tools



# Prototyping

- How to generate a prototype
  - Let the designer set the problem, let the programmer understand the goal, work together to solve the problem
  - Which prototyping technique? Paper, storyboard, physical, software
  - Code is expensive, reduce cost & effort
  - Only spend code where you need understanding;
  - Don't restrict your mental capabilities by code
  - Don't care about robustness, code optimality, software engineering
  - If software, choose between rapid prototype and using a real engine

# Prototyping

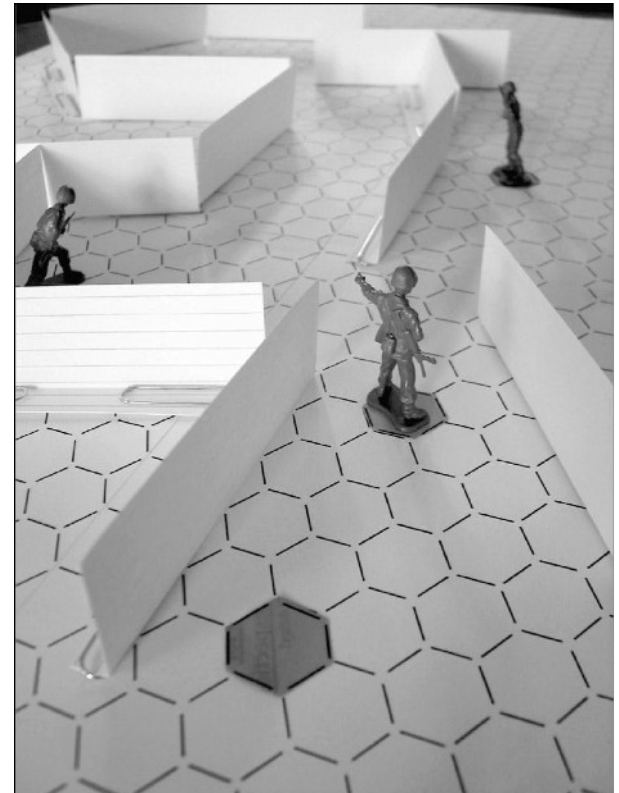
- Prototyping techniques
  - “Paper” / Physical prototypes
    - Good for testing game mechanics, quick to produce, but cannot convey game experience and action
  - Storyboard and animatics
    - Captures user experience, useful for communicating ideas
  - Software prototypes
  - Hardware prototypes (i.e., manufacturing processes)

# Prototyping

- Physical prototypes - Step 1: Foundation
  - Design the basic game objects and mechanics
    - Cards, paper, pens, toys etc.
  - Test your game idea
  - Restrict yourself to the fundamentals; Don't try to be too detailed

# Prototyping

- Physical prototypes - Step 1: Foundation
  - Example: first-person shooter
  - Core gameplay: simultaneous action
  - Accomplish with action cards



# Prototyping

- Physical prototypes - Step 2: Skeleton
  - Prioritize what is most essential and refine
    - Number of spaces player can move
    - Procedures for turning
    - Hit and miss rules for shooting
  - Build upon foundation with structure to support essential parts of game
    - Scoring system
    - Hit points

# Prototyping

- Physical prototypes - Step 3: Formal Details
  - Add rules and features for a fully functional fun game
  - Focus on most important formal elements
    - Is objective interesting and achievable
    - Is player interaction ideal
    - Are there missing rules
  - Test each rule individually to determine if it is critical or not
    - Hit percentage, health scoring,...



# Prototyping

- Physical prototypes - Step 4: Refinement
  - You have a playable system
  - Play, tweak, play, tweak, play, tweak,...
  - Question smaller and smaller details
  - Especially: Is the game fun?
  - Add new features one at a time

## Look ahead: next steps

- Mutual project critiques on the wiki until Friday (May 15th)
  - Each team member separately for all other game ideas
  - *What is your favorite aspect of the proposed game? Why?*
  - *What is your least favorite aspect? Why?*
  - *Which single change/addition would you suggest to most improve the game?*
- Build a paper prototype for the next milestone (May 23rd)
  - Present changes to your game idea after the feedback
  - Explain your paper prototype and show a demo of its gameplay
- Afterwards implementation phases start!