

# Game Design



# Welcome to CS-GY 6553 / CS-UY 4553

Game Design

Spring 2020 - 17299 / 16050

Wednesday - 12:25 - 2:55 - 2MTC 811

Instructor: Carmine T. Guida  
Email: [cguida@nyu.edu](mailto:cguida@nyu.edu)

# Prof. Carmine T. Guida



# CS 6553/4553: GAME DESIGN

All Games > Free to Play Games > Quintet

## Quintet

Community Hub



A cooperative, cross-platform, multiplayer scifi game where players control the same ship through 5 roles — Captain, Helm, Tactical, Engineering & Science. Battle with other player crews over the Internet.

ALL REVIEWS: [Mostly Positive \(365\)](#)

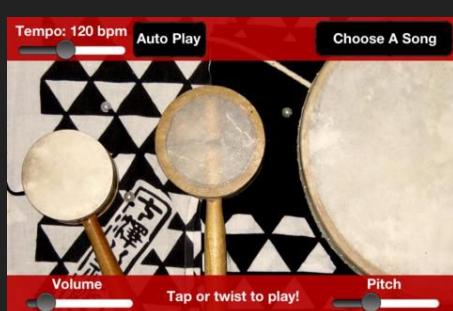
RELEASE DATE: Feb 1, 2013

DEVELOPER: Carmine T. Guida  
PUBLISHER: Carmine T. Guida

Popular user-defined tags for this product:

[Free to Play](#) [Indie](#) [Action](#) [Space](#) [Multiplayer](#) [+](#)

# CS 6553/4553: GAME DESIGN



Add a **leaderboard** to your game **right now**

- **no php or database**
- **easy to implement**
- **works on several platforms**

**dreamlo**

[Package contents](#) 17.8 KB

[Releases](#) current ver. 2.0

CARMINE T. GUIDA

**dreamlo.com - Free Instant Leaderboards and Promocode System**

FREE

[Add to My Assets](#)

★★★★★ 40 user reviews

Add a leaderboard and promocode system to your game right now!

Uses simple HTTP GET (WWW) requests.  
Works on several platforms.  
No php required.  
No SQL required.

The system stores name, high score, time in seconds and an extra string.

You can also create a simple promo code system for giving people codes to upgrade your games.

# The Class

## Prerequisites

CS-GY 6553

CS-GY 6533 or OART-UT 1600 and OART-UT 1605

CS-UY 4553

CS-UY 3113 , CS-UY 4533; OART-UT 1600 and OART-UT 1605

You must have instructor permission otherwise!

# Syllabus

## Not a Programming Course!

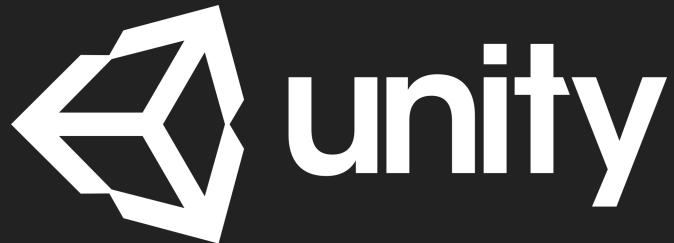
You should know how to program.

You can whatever game engine / tools / programming language you are most comfortable with.

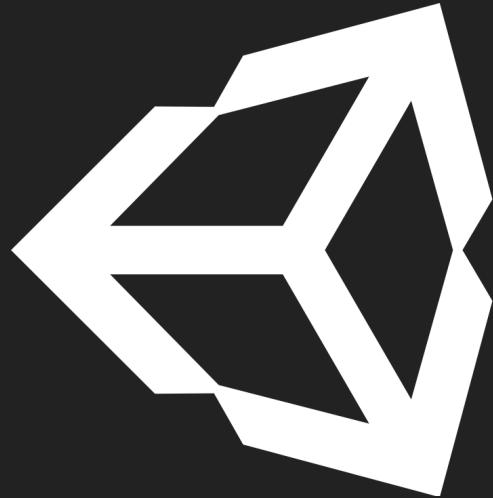
If you are not familiar with a game engine, this week and next week we will be working with Unity.

If you do not have Unity installed, get started now!

## Game Engines and Tools



# Unity



[unity3d.com](http://unity3d.com)  
Language: C#

# Processing



[processing.org](https://processing.org)  
Language: Java (simplified)

## p5.js and p5play



[p5js.org](https://p5js.org)  
[molleindustria.github.io/p5.play](https://molleindustria.github.io/p5.play)  
Language: Javascript

## Phaser



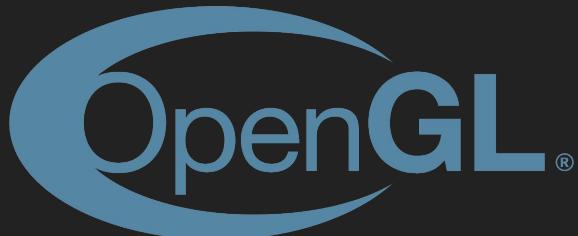
[phaser.io](https://phaser.io)

Language: Javascript

## SDL2 and OpenGL



[libsdl.org](http://libsdl.org)  
Language: C++



# Other Game Engines Welcome

Check with me today if you want to use something else!

# 3 Minute Meet & Greet

Unity, Processing, p5, Phaser, SDL + OpenGL

Find someone you do not know who uses the same Game Engine as you.  
Talk about a project you are proud of.

The second half of the semester is a team project, it may be helpful to find  
someone with complimentary skills.

# Class Roadmap

## Class Roadmap

Intro, Tools and Game Prototyping  
Systems, Rules and Mechanics  
Characteristics of Games  
Balance, Difficulty and Fairness  
Game Feel  
Board/Level Design  
Procedural Generation  
Narrative Mechanics  
From Prototype to Finished Game



# Systems, Rules and Mechanics

Elements, Relationships, Behaviors

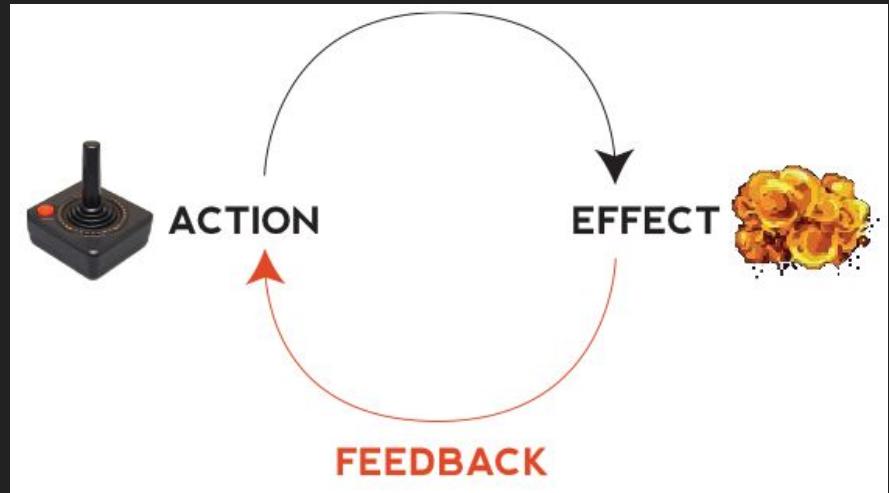
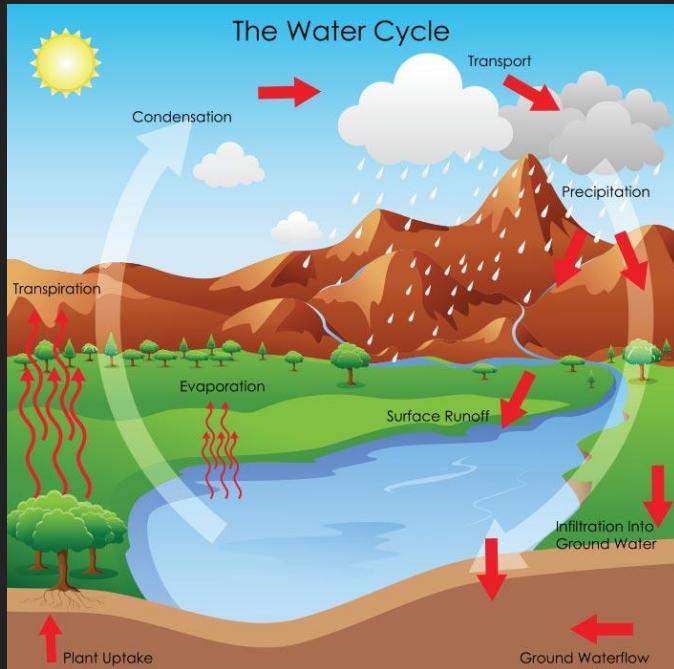
## Systems, Rules and Mechanics

“A system is a set of things—people, cells, molecules, or whatever—interconnected in such a way that they produce their own pattern of behavior over time.”

- Donella Meadows



# Systems, Rules and Mechanics



# Characteristics of Games

Playtime, Players, Heuristics



## Characteristics of Games

Length of Playtime

Number of Players

Heuristics

Interactivity

Politics

Depth

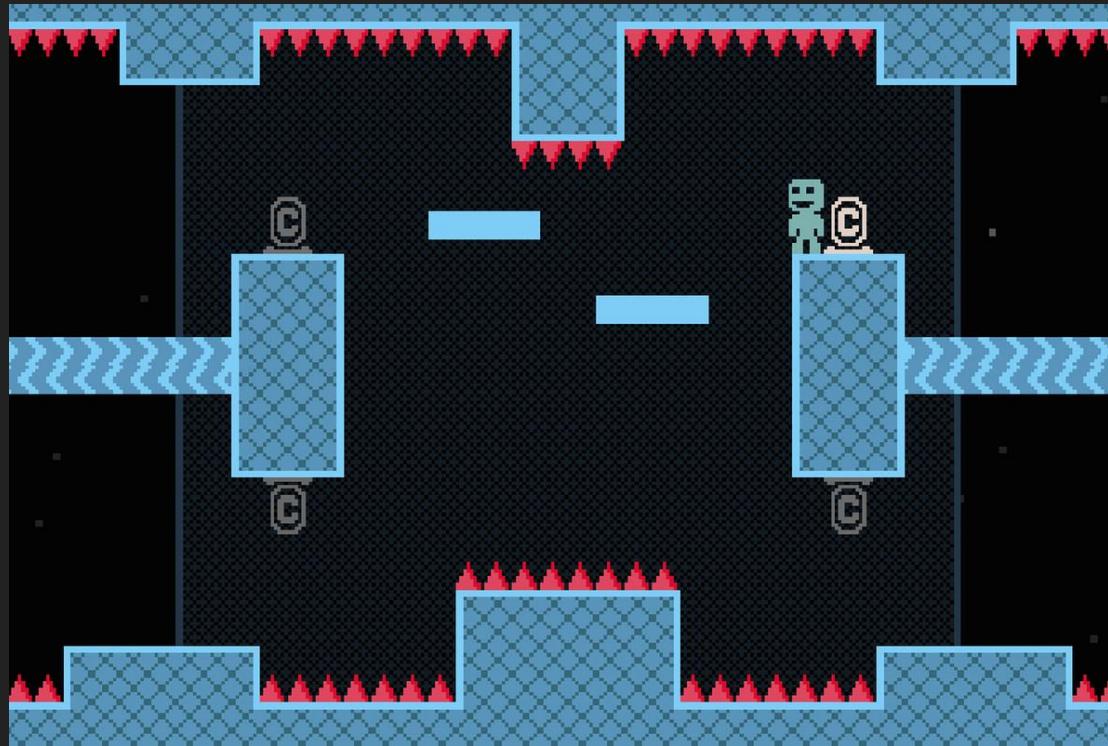
Standards

Ending/Victory Conditions



# Difficulty

## Difficulty



## Difficulty

How tough are you?

Can I play, Daddy?  
Don't hurt me.  
Bring 'em on!

🔫 I am Death incarnate!



## Difficulty



# Game Feel

## Game Feel

Virtual Sensation

Input / Response

Spatial Context

Signaling Information

Visual Feedback

Audio Feedback

Polish



## Game Feel



Spatial Context

## Game Feel



Visual and Audio

## Game Feel



Game Polish

# Board / Level Design

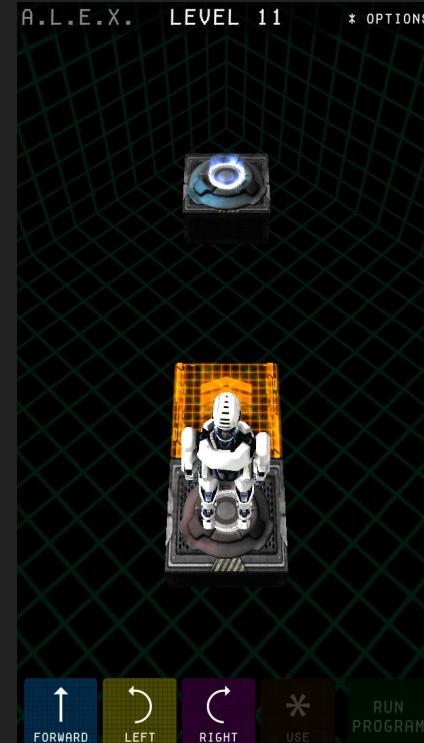
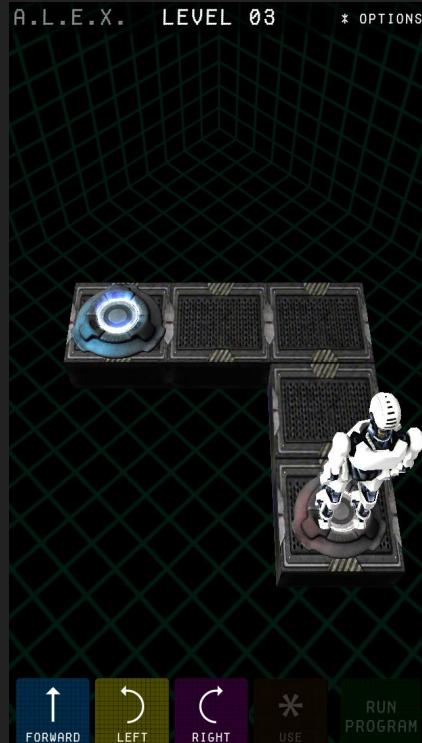
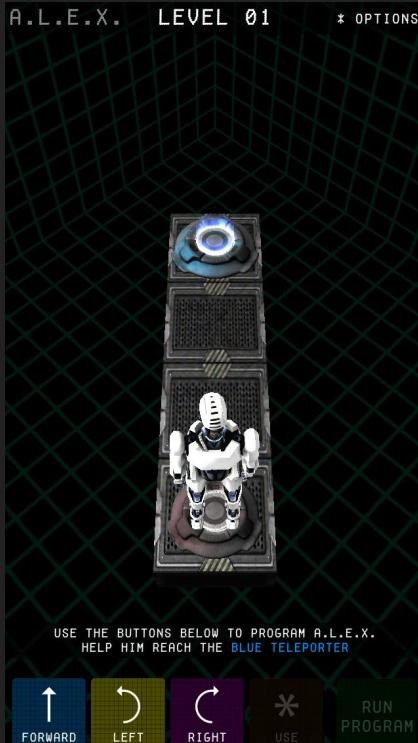
## Board / Level Design



## Board / Level Design



## Board / Level Design



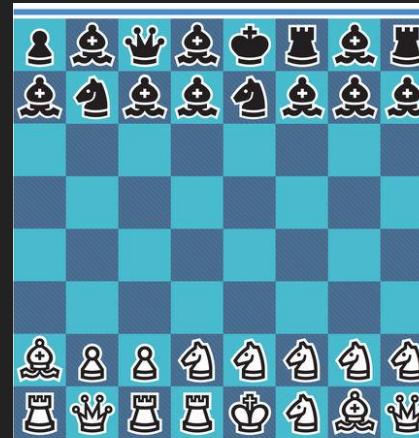
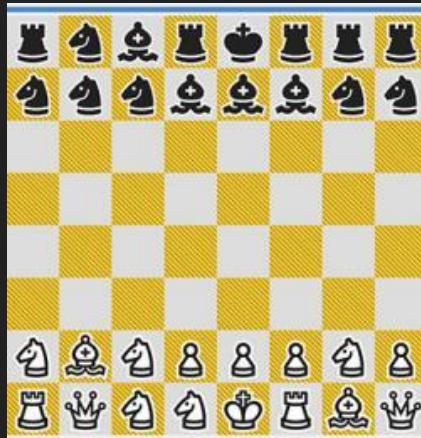
# Procedural Generation

And Randomness

## Procedural Generation



## Procedural Generation



Really Bad Chess

## Procedural Generation



# Narrative Mechanics

## Narrative Mechanics

### Missile Command

No win state - There is no winning in a global nuclear war.

Limited counter missiles - Have to make difficult choices of who to save and who to sacrifice.

Small number of cities - personal connection with what you are defending (author based it on the California coastline)



## Narrative Mechanics

### Monopoly

Lizzie Magie created the game in 1904 (originally called the "Landlord's Game") to teach the evils of wealth at the expense of others.

"In a short time, I hope a very short time, men and women will discover that they are poor because Carnegie and Rockefeller, maybe, have more than they know what to do with."



## Narrative Mechanics

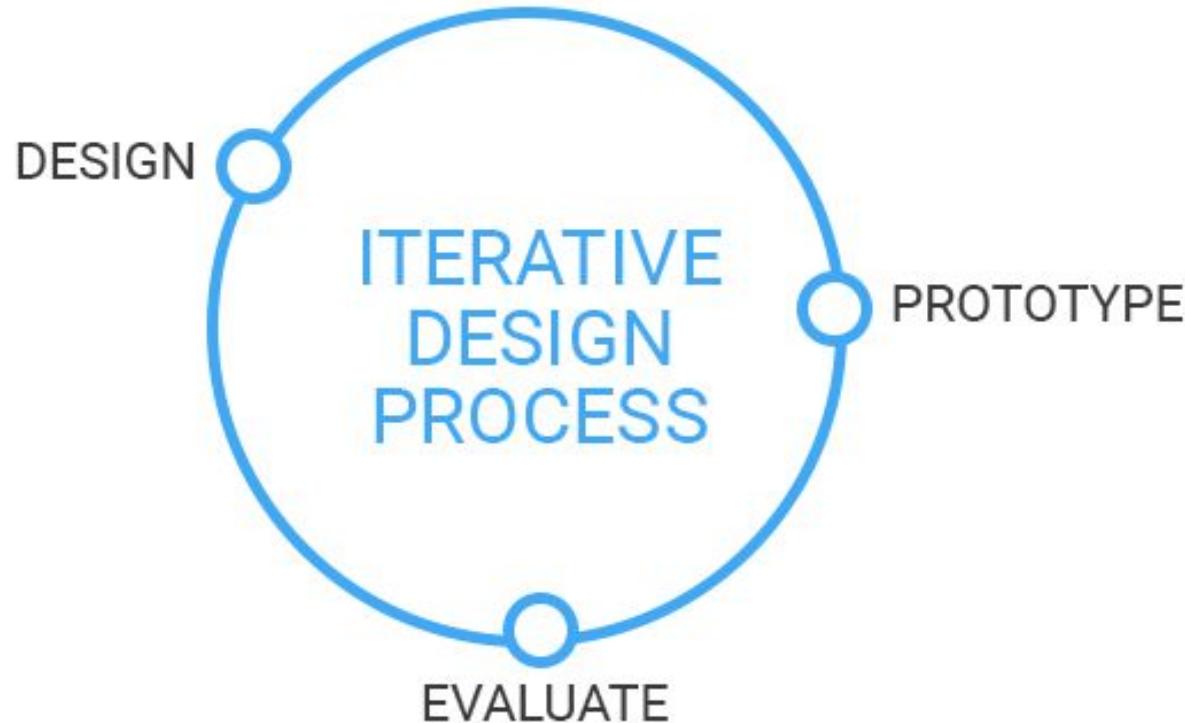


## Narrative Mechanics



Time  
for a  
Break!





# Prototyping

## Prototyping Goals

Proving that your basic game mechanic is interesting.

Changing the basic game mechanic if it isn't.

Finding new mechanics that emerge from your game's systems.

Exploring your concept from all possible sides.

## From Prototype to Finished Game



## From Prototype to Finished Game



## From Prototype to Finished Game



## From Prototype to Finished Game

Overwatch was released May 2016

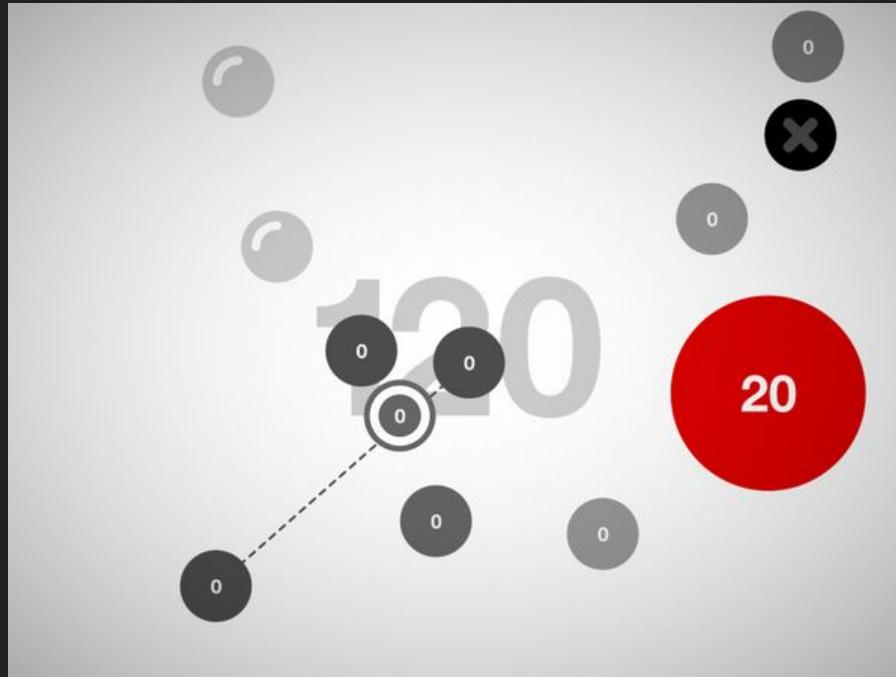


## Final game does not always have to look much different than the prototype!



This would not do. John needed room to show off his exceptional skills. As it was, he was trapped, on the wrong side of these little dot things.

**Final game does not always have to look much different than the prototype!**



# Getting Started with Prototyping a Game

## Getting Started With Prototyping a Game

1. Start with a Theme.

## Getting Started With Prototyping a Game

1. Start with a Theme.
2. Pick a few core elements iconic to that theme. Don't get complicated!

## Getting Started With Prototyping a Game

1. Start with a Theme.
2. Pick a few core elements iconic to that theme. Don't get complicated!
3. Define their relationships as rules in the game space.

## Getting Started With Prototyping a Game

1. Start with a Theme.
2. Pick a **few core elements** iconic to that theme. Don't get complicated!
3. Define their **relationships** as rules in the game space.
4. Define the player's role in these relationships.

## Getting Started With Prototyping a Game

1. Start with a Theme.
2. Pick a **few core elements** iconic to that theme. Don't get complicated!
3. Define their **relationships** as rules in the game space.
4. Define the player's role in these relationships.
5. What decisions does the player have to make at all times. Does this create a deep and compelling decision space for the game?

## Getting Started With Prototyping a Game

1. Start with a Theme.
2. Pick a **few core elements** iconic to that theme. Don't get complicated!
3. Define their **relationships** as rules in the game space.
4. Define the player's role in these relationships.
5. What decisions does the player have to make at all times. Does this create a deep and compelling decision space for the game?

You want to have the **smallest possible space** with the least amount of rules that creates a **compelling experience** for the player. Use time restrictions and rule coupling to make smaller rule sets more interesting.

# Prototyping in This Class

Minimalist Game Design

# Minimalist Game Design Methodology

## Towards Minimalist Game Design

Andy Nealen<sup>\*</sup>  
Rutgers University  
Hemisphere Games

Adam Saltsman<sup>†</sup>  
Semi Secret Software

Eddy Boxerman<sup>‡</sup>  
Hemisphere Games

### ABSTRACT

In this paper, we describe a design methodology that we have termed *Minimalist Game Design*. Minimalist games have small rulesets, narrow decision spaces, and abstract audiovisual representations, yet they do not compromise on depth of play or possibility space. We begin with a motivation for and definition of minimalist games, including terms such as "rules," "mechanics," "control," and "interface," and illustrate the importance of artificial design constraints. Using a number of examples, we show the strengths of minimalist game elements in systems, controls, visuals, and audio. Adhering to these constraints, these games feature a small set of mechanics and one core mechanic, while still being sufficiently deep and allowing for player exploration and performance. This depth comes from procedural methods, combinatorial complexity, probability, obfuscation, challenge, or any combination thereof. Our methodology embraces principles of holistic design, where there is no "filler," and where every element of the game contributes to the play experience in some meaningful, deliberate way.

### Categories and Subject Descriptors

K.8.0 [Personal Computing]: General – Games; H.5.2 [Information Interfaces and Presentation]: User Interfaces – User-centered design



Figure 1: The one-button game *Canabalt*

developers. We believe that self-imposed, deliberate constraints on both the design process and game are an important component to exploring new types of games and play. Even established developers seem to occasionally adhere to this philosophy (e.g. *WarioWare* and *Rhythm Heaven*).

To this end, we introduce the notion of *Minimalist Game Design* as a general design principle, and we attempt to define the cultural space of minimalist games. But first we must ask: why minimalism? Why exploratory game design under minimalist constraints? What makes this model compelling to both designers and players? The following list is

## Minimalist Games

Small Rulesets

Narrow decision spaces

Abstract Visual Representation

Abstract Audio Representation



...yet they do not compromise on depth of play or possibility space!

## Game Design Vocabulary

**State** - The space, entities and variables of our game.

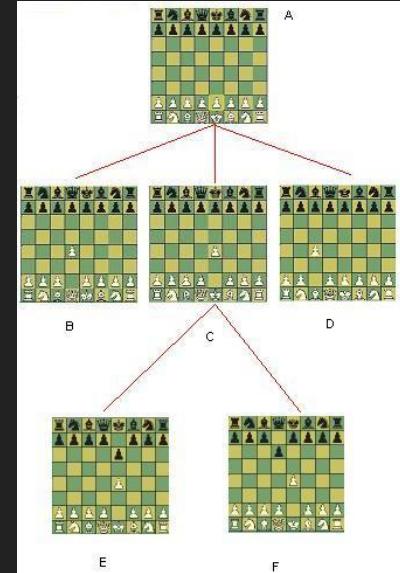
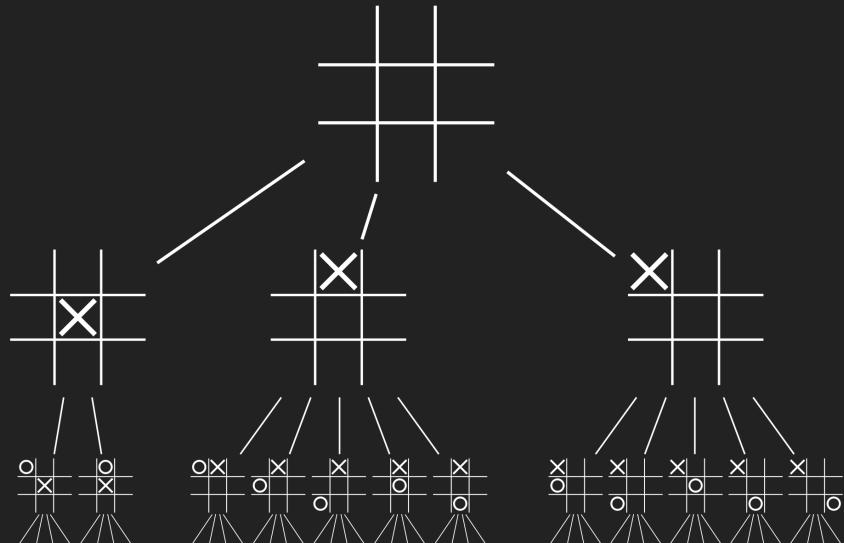
**Rules** - Define all changes of state in the game.

**Mechanics** - Rules that can be invoked by the player.

**Controls** - The way by which the player invokes the mechanics.

**Interface** - The full input/output of the game; Controls combined with audio-visual feedback.

## Small Ruleset, Narrow Decisions, Abstract Visuals, Maintains Depth



# Time Constraints

Time constraints are one way to make narrow decision spaces more interesting.

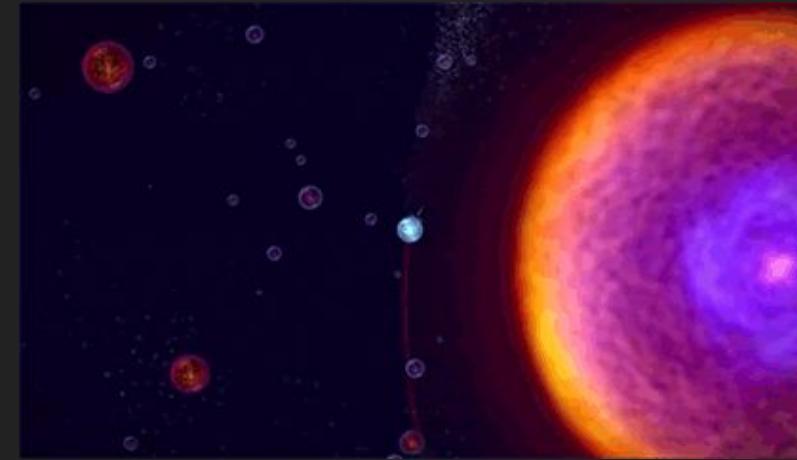
## Time Constraints - Canabalt



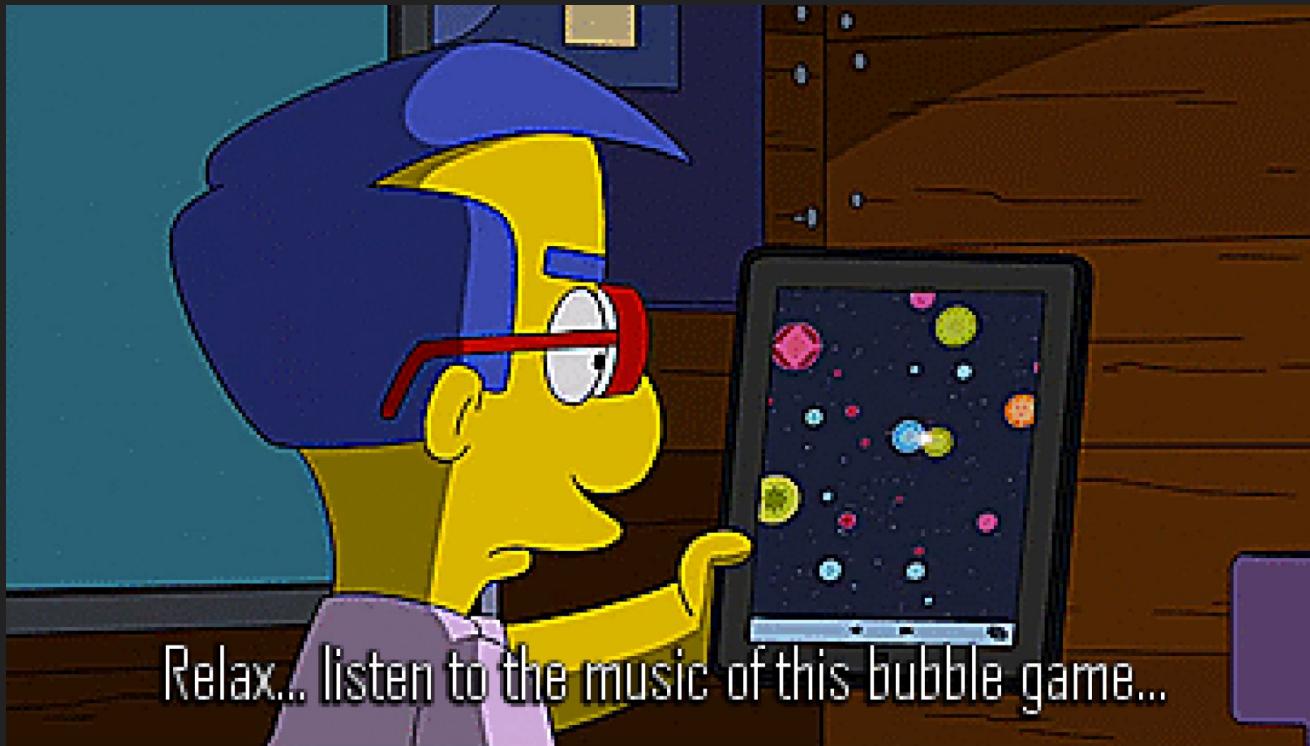
# Rule Coupling

Combining rules is one way to  
make small rule sets more interesting.

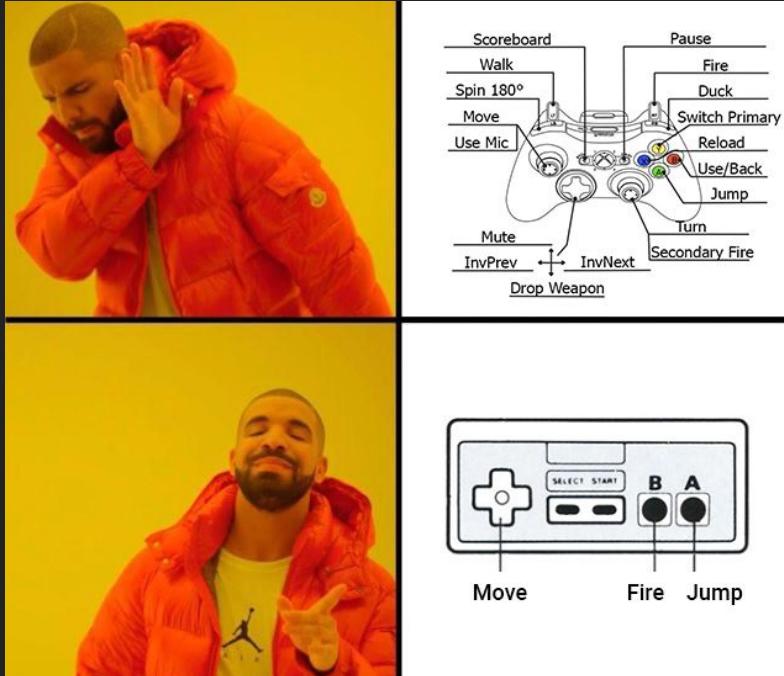
## Rule Coupling - Osmos



## Rule Coupling - Osmos



# Minimalist Controls



# Minimalist Visual Representation



# Prototyping in This Class

Small set of rules and thus a small state.

Small set of mechanics or only one core mechanic.

A simple abstract audio-visual representation and simple controls.

# Guida's Guide

# Guida's Guide

1. Make a list of all the features you want in your game.

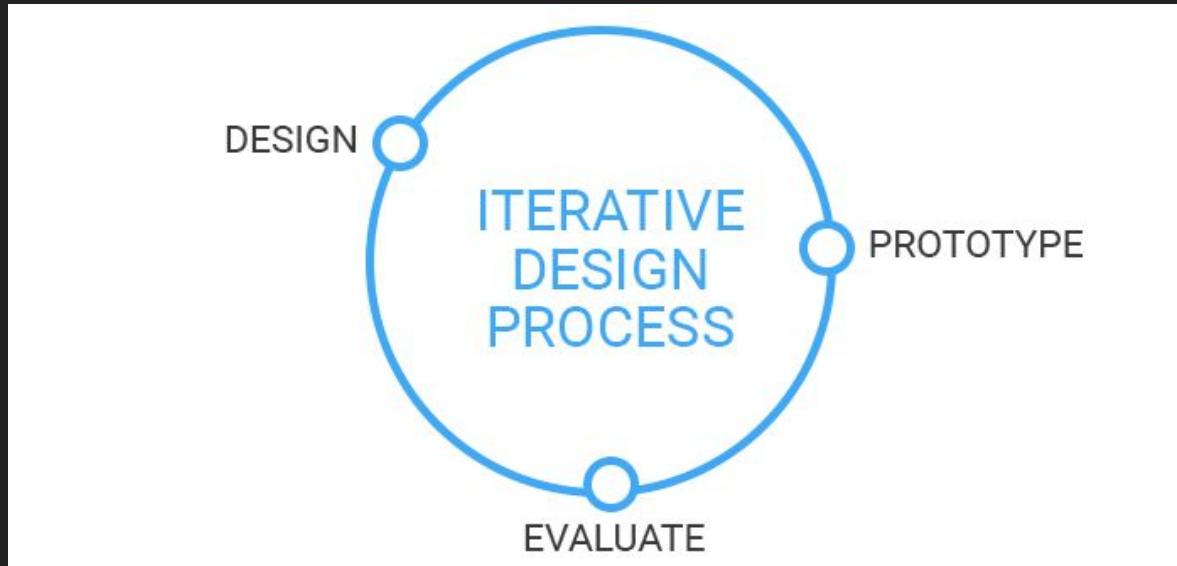
# Guida's Guide

1. Make a list of all the features you want in your game.
2. Take half that list.

# Guida's Guide

1. Make a list of all the features you want in your game.
2. Take half that list.
3. Throw that half away.

# Playtesting



## Playtesting

Open / Closed

Target Audience

Test the Test!

Testing the game not the player!

Take a Survey



<https://gamecenter.nyu.edu/events/playtest-thursdays/>

# Remember Prototyping Goals

Proving that your basic game mechanic is interesting.

Changing the basic game mechanic if it isn't.

Finding new mechanics that emerge from your game's systems.

Exploring your concept from all possible sides.

# Uno!

Let's add some mechanics to Uno and playtest!