

# CIS 300

## Introduction to Computer Game Design

<http://www.cis.cornell.edu/courses/cis300/2005sp>  
<http://www.cs.cornell.edu/projects/game>

Lecture 7: The Parts of a Game  
Spring 2005

1

## Announcements

- Read Chapter 4 (and 3) of R&A for next week. We'll be discussing setting, story, and narrative for next week.
- The *Group Charter* assignment is going up soon, due on Tuesday.
- Draft of your *Concept Document* is also due next week.

2

## The Parts

It's often hard to break up a game into distinct parts, because there is usually too much overlap to separate them. But, here are four broad components:

- Game Engine
- Rules and Mechanics
- User Interface
- Content and Challenges

3

## Game Engines

Sometimes when a developer or player uses the term "engine" they really mean "graphics engine". But a game engine encompasses much more. Game engines:

- Power the graphics and sound
- Power the AI
- Power the physics and interactions in the game
- Describe the nature of the game space
- Define the parameters of game objects
- Define the space of possibilities in the game world

4

## Game Engines: Graphics

- Includes the low level computational instructions for how things are drawn on the screen.
- Contains routines for manipulating images.
- Defines the graphical capabilities of the game.
- For this class, our graphics engine is GameX.

5

## Game Engines: Physics

A (idealized) physics engine defines what physical attributes objects and the world itself can have, but not the precise values or effects of those attributes.

A physics engine may specify that:

- There is a gravity force.
- Objects have friction constants.
- The ways in which water can deform.
- The computation routines by which objects interact.

A physics engine does not (necessarily) specify that:

- Gravity is  $G$ , or  $g$ , or even inverse square.
- The specific friction constants of objects.
- The specific result of dropping a ball into water.
- The specific routines called by particular interactions.

6

## General Game Engines

A game engine specifies the *space of possibilities* for a game, but not the *specific parameters* of elements of that game.

Some components of the *Super Mario Bros.* “engine”:

- Levels are fixed height scrolling maps.
- Levels are populated by blocks and enemies.
- Mario (and Luigi) can be small, big, or fiery.
- Blocks are affected by being bumped from below.
- Enemies are affected by being stomped, bumped from below, or hit by enemies or projectiles.
- Enemies have different movement/AI schemes.
- Enemies can spawn projectiles or other enemies.

7

## The D&D Analogy: Engines

If we consider a Dungeons & Dragons campaign as a game, then the D&D engine:

- Details the d20 system.
- Details round-based time.
- Tells us about actions, partial actions.
- Indicates that entities all have certain parameters.
- Tells us the spell system and basic combat system.
- Encompasses highest level decisions in the rulebooks.

8

## Characteristics of an Engine

- Is broad, adaptable, and extensible.
- Firmly encodes all non-mutable design decisions.
- Allows parameters for all mutable design decisions.
- Should outline the gameplay and challenge possibilities.
- Determines the overall game architecture.
- Is coded so that new design decisions leave it unchanged.

9

## Rules and Mechanics

Specific decisions about game parameters, obstacles, and abilities determine the rules and mechanics of the game. This includes things like:

- Player abilities
- Enemy stats
- Enemy behaviour
- Spell details
- Jumping height
- Gravity strength
- Point values
- Interplay between game objects

While the overall challenges aren't determined here, the heart of gameplay is in mechanics.

10

## Rules: Super Mario Bros.

Some rules from Super Mario Bros:

- One kind of block is the “question” block. A question block, when bumped, yields either a coin, 10 coins, a power-up, or a star.
- If Mario triggers a power-up when small, it is a mushroom. When big or fiery, it is a fire flower.
- Goombas die when stomped.
- Turtles become shells when stomped or bumped.
- 100 coins yields an extra life.
- Spinys damage Mario when stomped.
- Piranha Plants aim fireballs towards Mario.

11

## Rules and Mechanics (cont'd)

- If we continue the D&D analogy, then engine + mechanics = core rulebooks.
- Engine and mechanics still doesn't make a whole game.
- AI is part of the mechanics.
- If you have the engine and the mechanics, you should be able to make a level editor or game toolset.
- Takes the space of possibilities, and makes *decisions* for all parameters

12

## Interfaces

- The engine and mechanics tells us what the player and other objects in the game can do.
- The interface tells us how the player does things, and how she knows what's happening in the game.
- Interfaces thus have two parts:
  - Player-to-Computer
  - Computer-to-Player
- The interface is the center of the user experience.
- In the D&D analogy, the interface is character sheets, maps, dice, pencils, and the voices of the players and the Dungeon Master.

13

## Interface Tips

- On the PC, your inputs are mouse and keyboard. This affects not just the interface, but the design itself.
- Carefully consider the depth and width of your interface.
- Details are best processed at the center of vision.
- Peripheral vision mostly detects motion.
- Enhance your interface with sounds.
- Familiarity is better than innovation in interface.
- Strive for an “invisible” interface, but metaphorically.

14

## Content and Challenges

Content is everything we haven't discussed yet. We can divide it into two types: gameplay and non-gameplay.

Non-gameplay content includes:

- Graphics
- Sound Effects
- Background Music
- Cut Scenes
- Story
- Flavor Text
- Dialogue

To be fair, many of these have deep gameplay implications, and should be considered at other stages.

15

## Gameplay Content

When developers speak of content, they often mean gameplay content:

- Goals and victory conditions
- Missions and quests
- Level design
  - Pacing and Atmosphere
  - Difficulty curves and Balance
  - Reward structure
  - Atmosphere and Harmony
- In the D&D analogy, “modules” (adventures), and the DM's imagination are the content.

16

## Why the division?

- These four components - *Engine*, *Mechanics*, *Interface*, and *Content* – are **not** created sequentially, or separately. But thinking about them will keep you organized.
- Understanding the *Engine* tells you what decisions must be made early, and what should be hard-coded.
- Understanding the *Mechanics* tells you what design decisions may need changing and should be mutable.
- Understanding the *Interface* allows you to shape the user experience to fit your game vision.
- Understanding the *Content* ensures that you create the right world and gameplay for your game.

17