Lecture 13

Level Design
What is Level Design?

• Understanding of **player capabilities**
  • Abilities, mechanics available to the player
  • Assumptions of current player skill level

• Layout of **game geography**
  • Location and relationship of challenges
  • Movement of dynamic features (e.g. NPCs)

• Layout of **player progression**
  • How the player should move through the game
  • How the player visualizes this progression
Aspects of Game Design

- Games as **Education**
  - Train player skill and understanding
  - Focuses primarily on player capabilities

- Games as **Exploration**
  - Focuses on the game geography
  - Typically involves heavy storyboarding

- Games as **Storytelling**
  - Focuses on player progression
  - Most challenging element of game design
Aspects of Game Design

- **Games as Education**
  - Train player skill and understanding
  - Focuses primarily on player capabilities

- **Games as Exploration**
  - Focuses on the game geography
  - Typically involves heavy storyboarding

- **Games as Storytelling**
  - Focuses on player progression
  - Most challenging element of game design
Learning How to Play

- Mechanics are (often) new and unfamiliar
  - Players have to learn how to interact with them
  - Aside: why innovation is not always popular

- Players could learn by reading the *manual*
  - This is boring! Let me play already

- Tutorial levels allow the player to...
  - Get started playing immediately
  - Learn the mechanics while playing
Classic Approach: Restrict the Player

• Start with your **gameplay specification**
  • Remove all but the barest mechanics
  • Remove verbs by disabling controls
  • Remove interactions by omitting "board elements"

• Levels add new mechanics back one at a time
  • **Example**: Platformer with a "no-jump" level

• Do not need to add a new mechanic each level
  • "Deep" mechanics allow many levels per mechanic
  • This can influence game geography (e.g. worlds)
Example: Starcraft Campaign
Explicit Restrictions

- Mechanics are unavailable for current level
  - Controls for actions are explicitly disabled
  - Interactions disabled, even if elements present

- **Motivation**: Prevents player confusion
  - Do not waste time on useless mechanics
  - Key in the casual and young audience

- **Examples**: Many AAA commercial games
  - *Starcraft* single-player campaign
  - *Portal* (integrated into story)
Implicit Restrictions

- Mechanics are always available, but not needed
  - Challenges designed for an explicit mechanic
  - Other mechanics may succeed, but they are harder
  - Level has hints to guide player to right mechanic

- **Motivation**: Allow replay in tutorial levels
  - Players go back and try optional approaches
  - Achievements are structured to encourage this

- **Example**: Many amateur Flash games
  - *My First Quantum Translocator*
The Tyranny of Choice

- Too much choice can make us unhappy
  - We are often paralyzed by what to do
  - Studied by Myers & Lane; popularized by Barry Schwartz

- But games are about **meaningful choice**
  - Problem is when choices are too similar
  - Good choices must be *significantly* different
  - **Example**: Dagger adds +1 bonus to a stat of 102

- Players use rough heuristics for making choices
  - Pattern match current situation to determine action
Portal 2 Mechanics

Level Mechanic
Reinforcement

How long to “dwell” on mechanic before a new one?

Actions:

A = jump  B = dash

A B vs. A A A A B
Recombination

How often to combine with other mechanics

Actions:
A = jump    B = dash    C = shoot fireball

A   B   C   vs.   A   AB   ABC
Reinforcement vs. Recombination

A A A B B B

Reinforcement

A A B B AB AB

Recumbination

A B C D E

A AB ABC

ABCD ABCDE
Robot Unicorn Attack
Robot Unicorn Attack Progression

Mechanics:

A = jump  B = dash

A A A A B B A A A B

High reinforcement, low recombination
game design initiative at cornell university

A B C

PILLARS

TIME: 0
PAR TIME: 60
SPEED TIME: 30

COINS: 0/19
STARS: 12
POINTS: 1203

COMBO(C) REWIND(Y) QUIT RESTART(R)
Hello Worlds

**Mechanics:**
A = move  B = two worlds  C = close world

---

Moderate reinforcement, high recombination
Starcraft
Starcraft

Low reinforcement, high recombination
Aspects of Game Design

• Games as **Education**
  • Train player skill and understanding
  • Focuses primarily on player capabilities

• Games as **Exploration**
  • Focuses on the game geography
  • Typically involves heavy storyboarding

• Games as **Storytelling**
  • Focuses on player progression
  • Most challenging element of game design
Players Want to Explore the World

- Exploring the **physical space**
  - What happens when I go here?
  - **Example**: Any western RPG
  - But does not require complex game world

- Exploring the **ludic space**
  - What happens when do this action?
  - Requires deep, complex interactions
  - **Example**: Buckets in Skyrim
Players Want to Explore the World

- Exploring the **physical space**
  - What happens when I go here?
  - **Example:** Any western RPG
  - But does not require complex game world

- Exploring the **ludic space**
  - What happens when do this action?
  - Requires deep, complex interactions
  - **Example:** Buckets in Skyrim

Essentially covered this already
Storyboarding

- Diagrams player action throughout level
  - Different from film storyboarding
  - Currently a bunch of *informal practices*

- **Embodied Action**
  - Action that is tied to a character/avatar
  - Typically maps player movement in level

- **Disembodied Action**
  - Action corresponding to UI elements
  - **Example**: Buttons, menus
Embodied Action: Single Scene
Embodied Action: Multiple Scenes

- Point light
- Move, plug in
- Move, grab lights
- Point light
- Walk
- Win
Disembodied Action: Cause and Effect

- **Draw the initial scene**
  - Could be the entire level
  - Zoomed in portion of screen
  - Must capture area that will be affected by the action

- **Indicate the action**
  - Draw mouse pointer
  - Indicate gamepad button
  - Annotate with a “tool tip”

- **Draw the action effect**
  - Change in initial scene
Game Geography

- Relations of game challenges
  - Multiple challenges in a level
  - Flow of level progression
- Easiest to design *discretely*
  - Well defined player paths
  - Some deviation allowed
  - Storyboard indicates paths
- Ensure *meaningful choice*
  - More than one path successful
  - Balance the risk vs. reward
Risk versus Reward

[Edmund McMillen, edmundm.com]
Challenges with Geography

- World may be too open
  - Difficult to storyboard
  - Tyranny of Choice?

- World may be dynamic
  - Geography includes NPCs
  - They react to player actions
  - Again, how storyboard?

- Discretization is hard
  - Need to set boundaries
  - Must define building blocks
Boundaries

- **Explicit Boundaries**
  - Player is not allowed to go somewhere
  - **Example**: Platform layout in platformers
  - **Example**: Corridor layout in top-down or 3D

- **Soft Boundaries**
  - Player will face *unreasonable* challenges there
  - Player is at most "warned" about challenges
  - **Example**: Most open world RPGs
Level Design: Then and Now

1993

2010

FPS map design
But Actually…

[refugeinaudacity.wordpress.com]
Building Blocks

• Design game as **individual challenges**
  • Single obstacle or mechanic usage
  • Single interaction with a single NPC

• Corresponds to a **scene or encounter**
  • Single frame in your storyboard
  • Much less that a whole level

• **Assemble** these together to get a level
Example: **Blush**

Blush - Easy Level Storyboard
Team Motmot
Nora Ng-Quinn, Jie Ren, Ben Liu, Jeran Fox, Matt Slemon

**KEY**
- Yellow = Far Jump
- Green = Sticky
- Blue = Vertical Move

**Goal color**
**Initial color**

1. **Yellow**
2. **White**
3. **Green**
4. **Yellow**
5. **Green**
6. **Yellow**

**See whole level first**

**Get ready!**

- Yellow long jump to other platform, turn it yellow
- Jump in blue bucket & jump up, turning the upper platform top, & jump to turn the platform blue
Example: Hollow Wood

- Wood layouts:
  - Straight
  - Turn
  - Sparse
  - Fork
  - U-turn
  - Open space

- Level #1:
  - Fork
  - Open space
  - Sparse

- Map:
  - Path
  - House

---

Level Design
Working with Design Patterns

• Good building blocks become design patterns
  • An established mechanic/challenge pair
  • A design decision good enough it was copied

• Design patterns depend on genres
  • Shooter: Arenas, Strongholds
  • RPG: Waved minions, Inaccessible archers
  • Platformer: Forced jumps, Rhythm breakers

• Learn design by playing more games!
Aspects of Game Design

• Games as Education
  • Train player skill and understanding
  • Focuses primarily on player capabilities

• Games as Exploration
  • Focuses on the game geography
  • Typically involves heavy storyboarding

• Games as Storytelling
  • Focuses on player progression
  • Most challenging element of game design
Players Need Discrete Progression

- **Analogy**: hiking to a mountain peak
  - Constantly making progress to top of mountain
  - Psychologically hard to recognize this progress
  - **Result**: take pride in reaching concrete landmarks

- Players need this discrete progress as well
  - Individual levels
  - Game worlds
  - Narrative or *storytelling*
  - **Games** without story
Players Need Discrete Progression

- **Analogy**: hiking to a mountain peak
  - Constantly making progress to top of mountain
  - Psychologically hard to recognize this progress
  - **Result**: take pride in reaching concrete landmarks

- Players need this discrete progress as well
  - Individual levels
  - Game worlds
  - Narrative

Games
without story

In a Previous Lecture
Game as a Travelogue

Level Design
Travelogues are a Kind of Story

- Focus is on **journey**, not the characters
- Many examples in **literature**
  - *Gulliver’s Travels*
  - Classic heroic myths (e.g. *Odyssey*)
  - Epic fantasy (e.g. Tolkien)
- Problem is the **emotional impact**
  - One landmark is as good as any other
  - Can we achieve this without having narrative?
Emotional Progression: *Apsis*
Emotional Progression: *Apsis*

1. **Beginning**: Open and enjoyable
2. **Action**: Wind currents
3. **Harder**: Obstacles introduced
4. **Increasing**: Faster wind
5. **Increasing**: More obstacles
Emotional Progression: *Apsis*

6. **Relaxing**: Open up again
7. **Harder**: Hawk appears
8. **Joy**: Wind escapes hawk
9. **Rest**: Journey reaches end
Working on Your Level Design

• Create a **player tutorial**
  • Represents early levels in your games
  • Focuses on educational aspects of level design

• Create a **prototypical level**
  • Assume mastery of basic mechanics
  • Complex level with many components
  • Show how building blocks fit together

• Outline your **player progression**