Lecture 5

Rules and Mechanics
Today’s Lecture

• Reading is from Unit 2 of Rules of Play
  • Available from library as e-book
  • Linked to from the lecture page

• Not required, but excellent resource
  • Important for the serious designer
  • And ignore the Amazon reviews…

• The “Bible of Game Mechanics”
What are Rules?

- Definition from *Rules of Play*:
  - Rules are *formal schemas*

- But what does this really mean?

- Is it different for digital games?
Challenge of Defining Rules

- They do not need to be *fixed*
  - *Example*: *Nomic* (simulates democratic voting)
  - But are changed in structured ways

- They can *ignored*
  - House-rules that add or remove rules
  - Rule relaxation (e.g. playing with a young child)

- They are not always *explicit*
  - *Example*: does *Halo* have rules on camping?
Implicit Rules

- Rules beyond the explicitly stated ones
- Implicit rules for Tic-Tac-Toe
  - Must move in a “reasonable” amount of time
  - If loss is inevitable, must move or forfeit
- Often have to do with social conventions
  - If violate them, no one wants to play with you
  - Encapsulate being a “good sport”
Much more rigidly defined (in the software)

Possible to change in very structured ways
- Difficulty settings have a list of rule “alternatives”
- But arbitrary house rules are difficult (mods?)

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Friendly Fire (PC)</th>
<th>Friendly Fire (PS3 Xbox)</th>
<th>Flat Attack Bonus</th>
<th>Flat Defense Bonus</th>
<th>Flat Damage Bonus</th>
<th>Healing Effects Multiplier</th>
<th>Damage Threshold</th>
<th>Enemy Resist Bonus</th>
<th>Player Resist Bonus</th>
<th>Potion Cap</th>
<th>Trap Damage Multiplier</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casual</td>
<td>None</td>
<td>None</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>1.5</td>
<td>3</td>
<td>-10%</td>
<td>10%</td>
<td>20</td>
<td>0.5</td>
<td>Easy AI</td>
</tr>
<tr>
<td>Normal</td>
<td>50%</td>
<td>None</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>1.5%</td>
<td>0%</td>
<td>12</td>
<td>1.0</td>
<td>Moderate AI</td>
</tr>
<tr>
<td>Hard</td>
<td>100%</td>
<td>50%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.85</td>
<td>9</td>
<td>2.5%</td>
<td>0%</td>
<td>8</td>
<td>1.25</td>
<td>Full AI</td>
</tr>
<tr>
<td>Nightmare</td>
<td>100%</td>
<td>100%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.85</td>
<td>16</td>
<td>5%</td>
<td>0%</td>
<td>0</td>
<td>1.50</td>
<td>Full AI</td>
</tr>
</tbody>
</table>
Implicit Rules in Digital Games

- Implicit rules exist in digital games too
  - Camping in shooters
  - Juggle combos in fighters

- Depend upon context, and can change
  - Casual vs. core gamers in network play
  - Official vs. private game server

- Exist because cannot specify everything

- Commonly implemented via “terms of service”
(Formal) Rules in a Digital Game

- **Game State**: collection of values that represent the game world at a specific moment in time
  - Location, type of all the game objects
  - Non-spatial values (e.g. health) of these objects
  - Global non-spatial values (e.g. difficulty)
  - A high-dimensional tuple \( \mathbf{v} = (v_1, v_2, v_3, \ldots) \)

- **Possibility Space**: collection of all game states that are allowable (via the rules) in the game
  - Formally, we denote this space \( \mathbb{S} \)

- A *rule* is a function \( f: \mathbb{S} \rightarrow \mathbb{S} \)

But this is less helpful than you might think
How to Design Good Rules

- Player must have *meaningful choices*
  - Player must be able to make decisions
  - System must respond in significant way

**Bad Rules**: Guess heads or tails to pick a winner
- All you can do is guess the answer
- Has no significant effect on the outcome

**Bad Rules**: Move pieces on board with no interaction
- Actions have no meaning since pieces don’t interact
- There are no victory conditions or even challenges
Mechanics versus Rules

**Mechanics**
- *Informal* design concept
- Can represent activity over multiple animation frames

**Rules**
- *Formal* implementation
- Corresponds to code
- Typically at the level of an animation frame
- Though can have multiple rules per animation frame
- Goal is to match design
  - Is behavior correct?
  - Is behavior expected?
Formalizing Mechanics

- **Actions** take player input $I$ and current state $S$
  - Express as a function $g: I \times S \rightarrow S$
  - But could simplify; have input part of state

- **Interaction**: function between game states
  - Just like a rule, $f: S \rightarrow S$

- **Order** is another important consideration
  - Multiple actions, interactions possible per frame
  - How does order affect them?

Will return to this
Understanding Game State

- Many game state values are **spatial**
  - Represent location of a game **entity**
  - Also physical values like velocity, acceleration

- Entities act as containers for non-spatial values
  - Values that never change: **attributes**
  - Values that can change: **resources**

- Attributes, resources can be global as well
  - Though most mechanics are at entity level…
Actions Affecting Spatial State

• Typically we what we would call *movement*
  • Present in all but the most abstract games

• But there are many ways to implement
  • **Direct** movement of avatar (e.g. WASD)
  • **Indirect** movement of avatar (e.g. pathfinding)
  • Alter the **environment** (e.g. removing platforms)

• Area of much potential *innovation*
Altering the Environment

- Found in “physics” games
  - No direct control of avatar
  - Can only remove/add/move obstacles in environment
  - Movement is “natural”

- **Example**: *Screw the Nut*

- Physics is a rule system
  - Interaction, not action
  - Takes one state to another
  - Also one that is complex to understand/model
Innovating Avatar Movement

- 2D games move on 2-axes
  - Classic: left-right/up-down
  - Unless top-down game, one of these axes is restricted
- Is jump the only option?
  - Launcher/trajectory verbs
  - (Limited) teleportation
- **Example**: *Knightmare Tower*
  - Launcher-style game
  - Vertical movement is boosts gained from killing enemies
Environment **AND** Avatar

- Possible to split the verbs
  - Some for avatar movement
  - Others for environment
- Found in “drawing” games
  - Draw missing platforms
  - Avatar walks on platforms
  - **Ex:** Max & Magic Marker
- Innovate by limiting avatar
  - Move on single axis
  - Combine with environment
  - **Example:** Swindler
“Deep Gameplay”

- Want many ways to overcome challenges
  - **Example**: kill enemy or sneak past
  - If just one way, gameplay is “shallow”

- Shallow challenges hurt replayability
  - “Twitch” challenges become boring fast
  - Cerebral challenges solved by the walkthrough

- All games should have a **strategic** element
Strategy

• **Definition**: an elaborate sequence of steps
  - Action is the culmination of all the steps
  - Changing steps or order changes action

• Still allows for puzzle gameplay
  - Allow some *flexibility* in these solution steps
  - **Example**: Multiple solutions to Rubik’s Cube
  - **Example**: Time-rewind in *Braid*

• *Resources* are a common way to implement
Resources and Gameplay

- Resources are crucial to “combat” mechanics
  - Entities have resource values (e.g. health, ammo)
  - Expend resources to affect others (e.g. attack)
  - May change resources of that entity (e.g. damage)

- Three basic categories of resource combat
  - **Tug-Of-War**: entities take from each other
  - **Dot Eating**: entities race to gather *limited* resource
  - **Flower Picking**: race to gather *unlimited* resource
Resources and the Game Economy

- **Sources**: How a resource can increase
  - **Examples**: ammunition clips, health packs

- **Drains**: How a resource can decrease
  - **Examples**: firing weapon, player damage

- **Converters**: Changes one resource to another
  - **Example**: vendors, *Starcraft* barracks

- **Traders**: Exchange resources between entities
  - Mainly (but not always) in multiplayer games
Economic Challenges

- You can use resources to
  - Control player progression (hinder or advance)
  - Modify player abilities (limit or enhance)
  - Create a large possibility space (for replay value)
  - Create strategic gameplay

- Do not need a lot of resources
  - Not every game is a strategy game
  - But **almost all** games have some economy
Resources as Dilemma

- Players perform cost-benefit analyses
  - **Cost**: resource change not beneficial to player
  - **Benefit**: resource change beneficial to player

- **Example**: Survival Horror
  - Use ammo to shoot zombie (Cost: ammo)
  - Use knife to stab zombie (Cost: health)
  - Benefit the same in each case

- Players act with least cost for benefit
Resources and Monetization

- Most resources are gathered in-game
- But some games allow external sources
  - Get resources from a friend on Facebook
  - Pay for resources with a credit card
  - Known as resource monetization
- Free-to-play, pay-for-stuff
  - Modern business model for online games
  - Subscription model is (mostly) dead
Example: *Free Realms*
Complexity in Games

• Why is Tic-Tac-Toe unpopular w/ adults?
  • Experienced players always draw
  • Very easy strategy to memorize

• The game is too simple; needs to be complex
  • But if game is too complex, no one will play

• Complexity best through *emergent behavior*
  • “Coupled, context-dependent interactions”
Emergent Behavior

- **Coupled Interactions**
  - Two mechanics that can happen at once
  - **Verbs**: jump AND run in a platformer
  - **Resources**: warrior AND archer in an RTS

- **Context-dependent Interactions**
  - Mechanics combine to give new behavior
  - **Verbs**: jump and run is new form of movement
  - **Resources**: warriors form wall to cover archers
Emergent Behavior

- **Coupled Interactions**
  - Two mechanics that can happen at once
  - **Verbs**: jump AND run in a platformer
  - **Resources**: warrior AND archer in an RTS

- **Context-dependent Interactions**
  - Mechanics combine to give new behavior
  - **Verbs**: jump and run is new form of movement
  - **Resources**: warriors form wall to cover archers
Common Spatial Interactions

<table>
<thead>
<tr>
<th>Collisions</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can effect <em>resources</em></td>
<td></td>
</tr>
<tr>
<td>• Player takes damage</td>
<td></td>
</tr>
<tr>
<td>• Player gains power-up</td>
<td></td>
</tr>
<tr>
<td>• Player-NPC transfer gold</td>
<td></td>
</tr>
<tr>
<td>• Can effect <em>spatial values</em></td>
<td></td>
</tr>
<tr>
<td>• Bounce off collision point</td>
<td></td>
</tr>
<tr>
<td>• Swing from attached rope</td>
<td></td>
</tr>
<tr>
<td>• Attraction to magnet/charge</td>
<td></td>
</tr>
<tr>
<td>• Examples:</td>
<td></td>
</tr>
<tr>
<td>• Line-of-sight (w/ obstacles)</td>
<td></td>
</tr>
<tr>
<td>• Spatial proximity</td>
<td></td>
</tr>
<tr>
<td>• Can have <em>direct</em> effects</td>
<td></td>
</tr>
<tr>
<td>• Alarms in a stealth game</td>
<td></td>
</tr>
<tr>
<td>• Can have <em>indirect</em> effects</td>
<td></td>
</tr>
<tr>
<td>• Tower defense targeting</td>
<td></td>
</tr>
<tr>
<td>• Adjust NPC reactions</td>
<td></td>
</tr>
</tbody>
</table>
# Resource-Spatial Interactions

## Resource Affects Spatial
- Resources can unlock areas
  - Keys are a trivial resource
  - Also use resource thresholds
  - **Ex:** Collect all tokens to pass
- Resources affect difficulty
  - Adjust input device sensitivity
  - **Ex:** Deadeye meter in *RDR*
  - **Ex:** Jet packs to increase jump

## Spatial Affects Resources
- Resources made by entities
  - Have a spatial location
  - **Ex:** Time to transfer resources
  - **Ex:** Sources be captured
- Resource values are entities
  - Take up physical volume
  - Need space to acquire
  - **Ex:** Inventory in *Deux Ex*
Resource-Spatial Interactions

Spatial Affects Resources

- Resources made by entities
  - Have a spatial location
  - **Ex:** Time to transfer resources
  - **Ex:** Sources be captured

- Resource values are entities
  - Take up physical volume
  - Need space to acquire
  - **Ex:** Inventory in *Deux Ex*
Coupling is not Enough

- Example of *trivial* coupling:
  - RTS with single unit type – warrior
  - Coupling can arise from multiple warriors
  - When attack, count number on each side

- Group of warriors *is* sum of its parts
  - Just make a single warrior stronger
  - Discover from *resource analysis*

- Emergent behavior must couple *nonlinearly*
  - If $n$ base mechanics, more than $O(n)$ behaviors
Example: Starcraft

- Basic units can
  - Attack in sky and/or land
  - Defend in sky and/or land
  - How can these combine?

- Further complexity:
  - “Buff” friendly units
  - “Control” enemy units
  - How does this affect game?

- Challenge: What is minimal complexity for a good RTS?
Summary

- Rules are **formal systems** defining your game
  - Take one game state and produce another
  - Implementation of mechanics (a design concept)

- Game state is broken into two categories
  - **Spatial values** are attached to game *entities*
  - **Resources** create economy and *strategic* gameplay

- Good rules should
  - Allow for *meaningful play*
  - Allow for *emergent behavior*