Lecture 3

Design Elements
Reminder: Aspects of a Game

- **Players**: How do humans affect the game?
- **Goals**: What is the player trying to do?
- **Rules**: How can the player achieve the goal?
- **Challenges**: What obstacles block the goal?
Formal Design Elements

- **Players**: Player Mode Sketches
- **Goals**: Objectives
- **Rules**: Actions and Interactions
- **Challenges**: Obstacles and Opponents
Player Mode Sketches

- Game may have several *player modes*
  - Ways in which player interacts with a game
  - **Example**: Inventory screen vs. combat screen

- You should *storyboard* all of your modes
  - Sketches of each of the major player modes
  - May have action (like movie storyboard)
  - Illustrate how player interacts with game
Dragon Age: Standard Mode
Dragon Age: Inventory Mode
Lifted: Player Mode Sketch

Indicating Action
Lifted: Completed Game

Earth-snails make a delightful meal!

Time 00:07:88

Level 7
Diagramming Action

RISK

REWARD

EASY

HARD
Objectives

- Anything a player might strive for
- May be a primary game objective
  - Progressing the story
  - “Completing” the game
- May be an auxiliary game objective
  - Side missions/quests
  - Unusual achievements
- Sometimes player-directed
  - Reward structure in sandbox games
Objectives

- **Primary** objectives reflect vision
  - Wish fulfillment: I want to _________
  - Help player realize the dream

- **Auxiliary** objectives address player style
  - Achievements for achievers
  - Easter eggs for explorers
  - Online resources for socializers

- **Player-driven** objectives require a different focus
  - Start with a toy, and layer dramatic elements on it
Some Objective Categories

- **Capture**: take or destroy something of value
  - Includes “kill all enemies of type X”

- **Race**: reach a goal within time

- **Chase**: catch or elude an opponent
  - Race with a dynamic goal/destination

- **Rescue/Escape**: Get someone to safety

- **Exploration**: Locate something in game world
Some Objective Categories

- **Solution**: solve a problem or puzzle
- **Alignment**: arrange in a specific configuration
  - Color matching games!
- **Construction**: build, maintain, manage objects
  
  multiplayer games

- **Forbidden Act**: force opponent to do something
- **Outwit**: gain knowledge that gives an advantage
Some Objective Categories

- **Solution**: solve a problem or puzzle
- **Alignment**: arrange in a specific configuration
  - Color matching games!
- **Construction**: build, maintain, manage objects
- **Forbidden Act**: force opponent to do something
- **Outwit**: gain knowledge that gives an advantage

Design Elements
Actions

• **Verbs** that describe what the player can **do**
  • Walk  *(left or right)*
  • Run    *(walk, but faster!)*
  • Jump   *(up; jump/run for left or right)*
  • Shoot  *(left or right)*

• **Does not need to be attached to an avatar**
  • Build  *(RTS or simulation)*
  • Swap   *(Bejeweled clones)*
  • Rotate *(Stacking games)*

Action Platformer
Designing Actions

- Starts with brainstorming the verbs
  - Define the types of verbs
  - Define the scope of the verbs

- Design Goals
  - Enough verbs to avoid being too simple
  - But not so much to be confusing (verb bloat)
  - Do the verbs *directly* achieve the goal?

- Each verb maps to a single input
Primary Actions

- How do verbs, goals relate?
  - Imagine there no challenges
  - What verbs *must* you have?

- **Example**: Platformers
  - **Goal**: reach exit location
  - Only need movement verbs
  - Killing enemies is *optional*
  - Other actions are *secondary*

- **Design Goal**: Primary only
  - Secondary verbs lead to bloat
  - Add features with interactions
Secondary Actions are Acceptable

- Often in **puzzle platformers**
  - Platformer verbs + something
  - “Innovation on the cheap”
- Verb that alters “geography”
  - Access hard-to-reach areas
  - Directly overcome *challenges*
  - Not directly needed for goal
- But do this sparingly!
  - Indies have one new verb!
  - Other features are *interactions*
Interactions

- Not a *direct* action of player
  - Outcome of the *game state*
  - Can happen without controller

- **Example**: collisions
  - Accidental or player forced
  - May be bad (*take damage*)
  - May be good (*gain power-up*)

- **Other Examples**:
  - Spatial proximity
  - Line-of-sight
  - Resource acquisition
Game Mechanics

- **Game mechanic**
  - Relationship between verbs and interactions
  - Often call this relationship the “rules”
  - **Gameplay** is manifestation of these rules

- **Example**: Joust
  - **Verbs**: Flap; go left or right
  - **Interaction**: Collision with opponent
  - **Rule**: If hit opponent, lower player dies
Gameplay Example: *Joust*
Verb Minimalism

- Keep verbs at a minimum
- Mechanics are all interactions
- Common in mobile, tablet
- Due to lack of input modes

**Example:** Sneak Beat Bandit
- Has only one verb: *move*
- Rhythm game; move to beat
- All movement on rails
- If obstacle in way, turn
- Line-of-sight mechanics
Avoid Verb Proxies

• **Proxy**: verb that activates another verb
  • “Use an item” (what does the item do?)
  • “Shoot” (what does the weapon do?)

• **Make your verbs outcome oriented**
  • Fire standard projectile (like shoot, but says what it shoots)
  • Fire freezing beam (what does and how it is applied)

• **Important questions to ask**
  • Does it help me reach a goal?
  • Does it overcome a challenge?
Avoid Verb Proxies

- **Proxy**: verb that activates another verb
  - “Use an item”
  - “Shoot”
- Make your verbs **outcome oriented**
  - Fire standard projectile (like shoot, but says what it shoots)
  - Fire freezing beam (what is does and how it is applied)
- Important questions to ask
  - Does it help me reach a goal?
  - Does it overcome a challenge?

Behavior is defined by interaction of projectile with the environment.

Design Elements
Combining Actions

• Verbs can combine in interesting ways
  • Run and jump in a platformer
  • Strafing fire in a shooter

• Typically result of the interactions
  • Each verb interacts with environment in different way
  • Combination of two give extra feature for “free”
  • This is an example of emergent behavior

• Not all combinations are emergent
  • Example: Double jump is not a feature of interactions
  • This type of verb combination is a distinct action
Combining Actions

### Running Jump
- Can move while in midair
  - Just horizontal movement
  - Not realistic; it is a game
  - Many platformer challenges assume this type of control
- Different than a long jump
  - Less height than reg. jump
  - No control once in the air
  - Would be a distinct action

### Strafing Fire
- Based on “real life” property
  - Bullets travel in straight line
  - Movement changes origin
  - Walking side-side makes a spray (used in covering fire)
- But some features are gamy
  - Bullets slower than life
  - Character faster than life
  - Creates interesting effects
Combining Actions

Jump

- Can move while in midair
  - Just horizontal movement
  - Not realistic; it is a game
  - Many platformer challenges assume this type of control
- Different than a long jump
  - Less height than reg. jump
  - No control once in the air
  - Would be a distinct action

Strafing Fire

- Based on “real life” property
  - Bullets travel in straight line
  - Movement changes origin
  - Walking side-side makes a spray (used in covering fire)
- But some features are gamy
  - Bullets slower than life
  - Character faster than life
  - Creates interesting effects
Combining Actions

Is this an example?

Why or why not?
Challenges

• **Obstacles**
  - Prevent progress towards goal
  - Have to be “overcome”

• **Opponents**
  - Players or bots with their own goals
  - May or may not need to be overcome

• **Dilemmas**
  - Can only perform one of several actions
  - “Correct” choice not immediately clear
Challenges

• Obstacles
  • Prevent progress towards goal
  • Have to be “overcome”

• Opponents
  • Players or bots with their own goals
  • May or may not need to be overcome

• Dilemmas
  • Cannot choose one of several actions
  • “Correct” choice not immediately clear

See Text for Specific Examples
Challenges: Limitations

- You **cannot** always perform an action
  - Shooting may require ammo
  - Cannot (always) jump in mid air

- **Limitation**: requirement to perform action
  - Boolean test (like an *if-then*)
  - Checked at time of user input

- **Only one** limitation per verb
  - If more than one, split into more verbs
  - Reason double-jump is distinct
Challenges: Limitations

- You **cannot** always perform an action
  - Shooting may require **ammo**
  - Cannot (always) jump in mid air

- **Limitation**: requirement to perform action
  - Boolean test (like an *if–then*)
  - Checked at time of user input

- Only **one** limitation per verb
  - If more than one, split into more verbs
  - Reason double-jump is distinct
Challenges: Resources

• **Game State**: numeric and symbolic values that represent the game world at a specific moment

• **Numeric** quantities are often called **resources**
  • Examples (player): ammunition, health points
  • Examples (external): monster spawns

• **Symbolic** values are “yes-or-no” quantities
  • Used for “lock-and-key” challenges
  • Typically create shallower gameplay
Challenges: Resources

- **Game State**: numeric and symbolic values that represent the game world at a specific moment.
  - **Numeric** quantities are often called resources.
    - Examples (player): ammunition, health points.
    - Examples (external): monster spawns.
  - **Symbolic** values are “yes-or-no” quantities.
    - Used for “lock-and-key” challenges.
    - Typically create shallower gameplay.

Will cover in more detail later.
Putting It All Together

- Start with your **vision**
  - I want to __________
  - This creates setting and player goals

- Create a (partial) list of the following:
  - **Objectives**
  - **Actions**
  - **Interactions**
  - **Challenges**

Sketch **player modes** to show them in action