

Lecture 2

Mechanics Revisited

Purpose of Today's Lecture

- Give a review of formal **design elements**
 - Not everyone here has had the Intro Games course
 - And for the rest of you, it has been over a year
- Develop a deeper understanding of **mechanics**
 - Understand the important of interactions
 - Understand the *analysis* challenges
- Set us up for the **later lectures** on mechanics
 - Mobile game design and monetization

Reminder: Aspects of a Game

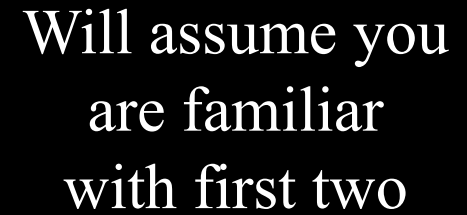
- **Players:** How do humans affect the game?
- **Goals:** What is the player trying to do?
- **Rules:** How can the player achieve the goal?
- **Challenges:** What obstacles block the goal?

Formal Design Elements

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

Formal Design Elements

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



Will assume you
are familiar
with first two

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)

Action
Platformer

- Does not need to be attached to an avatar

- Build (RTS or simulation)
- Swap (Bejeweled clones)
- Rotate (Stacking games)

Evaluating Your Actions



- How important are they?
 - Do they help achieve goal
 - If not, why are they there?
- **Example:** Platformers
 - **Goal:** reach exit location
 - Killing enemies is *optional*
 - Other actions are *secondary*
- **Goal:** Minimize verbs
 - More verbs lead to **bloat**
 - Leverage **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



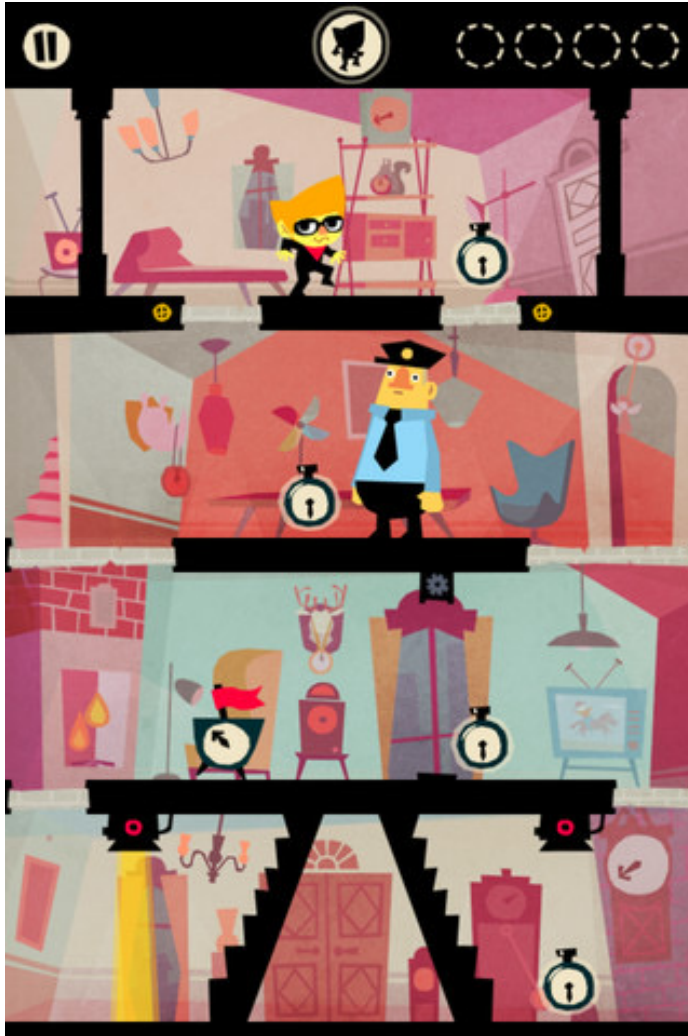
Game Mechanics

- **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*

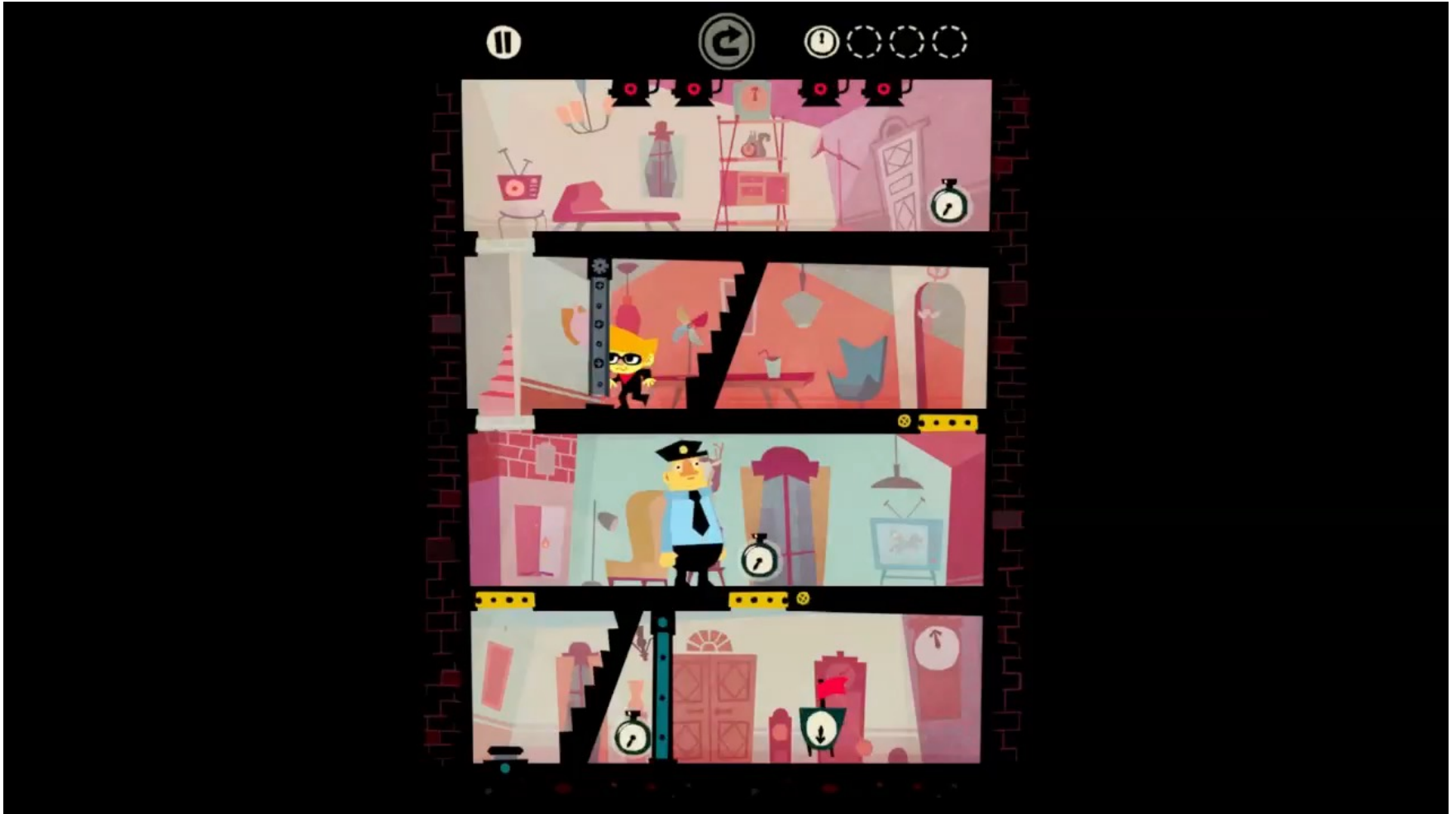


Early Mobile Games: Just One Verb



- Can we limit to **one** verb?
 - 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

Beat Sneak Bandit



Avoid Verb Proxies

- **Proxy**: verb that activates another verb
 - “Use an item” (what does the item do?)
 - “Shoot” (what does the weapon do?)
- Make the **outcome** of your verbs clear
 - Fire standard projectile (effects have “travel time”)
 - Fire continuous beam (effects are instantaneous)
- Important questions to ask
 - How does help reach the goal?
 - How is it outcome challenged?



Avoid Verb Proxies

- **Proxy**: verb that activates another verb
 - “Use an item”
 - “Shoot”
- Make the **outcome** of the verb clear
 - Fire standard projectile (effects have “travel time”)
 - Fire continuous beam (effects are instantaneous)
- Important questions to ask
 - How does help reach the goal?
 - How is it outcome challenged?

Behavior is defined by the *interaction* of projectile/beam (them do?) (weapon do?)



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**

- Can only pick one
- “Correct” choice

Generally necessary
for “deep” gameplay

“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

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...

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

Resources and the Game Economy

- **Sources:** How a resource can increase
 - **Examples:** ammunition clips, health packs
- **Drains:** How a resource is decreased
 - **Example:** health packs, armor
- **Converted:** Turns one resource to another
 - **Example:** vendors, *Starcraft* barracks
- **Traders:** Exchange resources between entities
 - Mainly (but not always) in multiplayer games

Will explore in more detail with monetization

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

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**

Key Word

- 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

Examples of Emergent 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

Examples of Emergent Actions

Interaction(?)

ump

- 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

Interaction

- Based on “real n 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

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

Common Spatial Interactions

Collisions

- Can effect *resources*
 - Player takes damage
 - Player gains power-up
 - Player-NPC transfer gold
- Can effect *spatial values*
 - Bounce off collision point
 - Swing from attached rope
 - Attraction to magnet/charge

Detection

- Examples:
 - Line-of-sight (w/ obstacles)
 - Spatial proximity
- Can have *direct* effects
 - Alarms in a stealth game
- Can have *indirect* effects
 - Tower defense targeting
 - Adjust NPC reactions

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

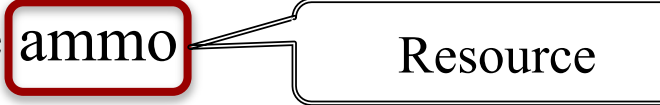
Spatial Affects Resources

- 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*

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

Next Time: Mobile Gaming