the gamedesigninitiative at cornell university

Lecture 3

Mobile Gameplay

Focus of Today's Talk

iPhone/iPod Touch

iPad

Focus of Today's Talk

Android Phone

Android Tablet

Not The Focus of Today's Talk

Why Not?

- "Traditional" input schemes
  - They all have gamepads
  - DS has stylus, which acts like a mouse pointer
- Supports classic gameplay
  - Games often "miniaturized" versions of console/PC titles
  - Market is very hardcore
- Not modern mobile market

Challenge: Input Modality

- Don't have standard gamepad controls
  - Add-on hardware is in infancy
  - Not standard, few games use

- Loss of a lot of functionality
  - D-Pads, joysticks for avatar control
  - Buttons for performing core actions
- Have to **rethink game input**

The Cheap Way Out

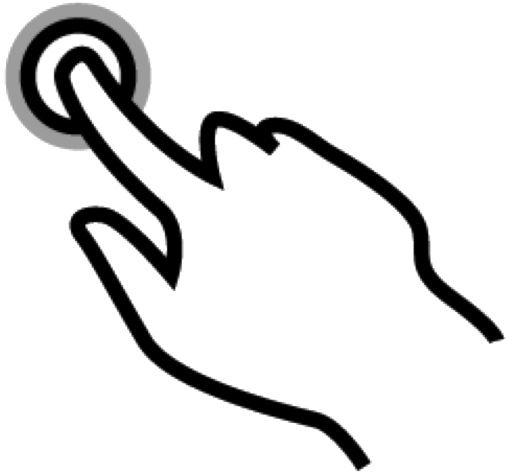
The Cheap Way Out

So What Can We Do?

- (Multi) Touch Controls
  - Pointing, dragging
  - Clicking, selecting
  - More advanced gestures
- Accelerometer Support
  - Tilting
  - Rotating

Touch: Basic Approach

- Can use touch interface like a **mouse**
  - Touch to click on a point,
  - Trace from touch to drag
- Port mouse-heavy PC/Mac games
  - Particularly strategy games/RPGs
- Keyboard exists, but is limited
  - Have to obscure screen to pull up keyboard
  - Use very sparingly (e.g. save file)



Example: *Plants vs. Zombies*

4152 Example: *Gathering Sky*

Balancing Multitouch

- Mouse games are "balanced" for a single pointer
  - Multitasking requires a lot of back and forth
  - Challenge is to do actions in an efficient order
- Multitouch eliminates this challenge
  - Can quickly move fingers anywhere
  - Can use multiple fingers at once
  - **Example**: Whack-a-Zombie
- Need to rethink gameplay

Size Matters

- Small screen makes multitouch *hard*
  - True multitouch only on a tablet
  - Phones are largely limited to gestures
- Fingers are **fatter** than mouse pointers
  - I did not mean to click that!
  - Also, fingers cover up the screen
  - Touch needs to be very forgiving

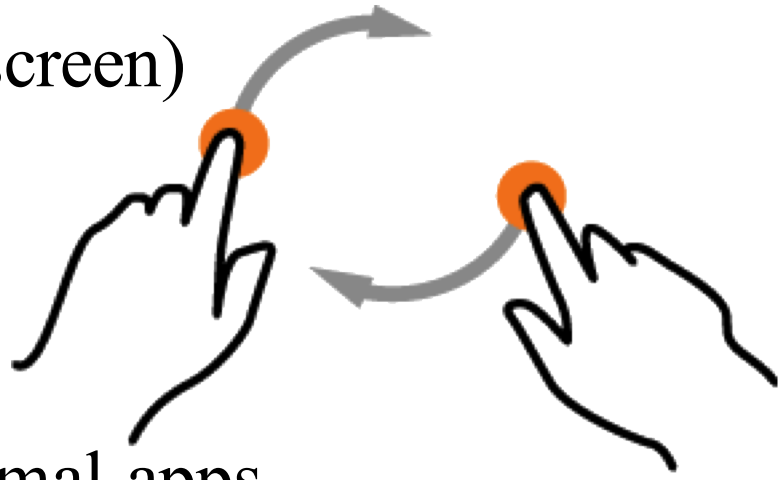
Click versus Pointing

- In a PC game, can hover mouse above location
  - Gives pop-up menus for gathering information
  - Key feature in RPGs, strategy games
- How can we distinguish point from click?
  - Could make clicking more cumbersome (e.g. verification dialogs)
  - But slows down the game
  - Any better solutions?

Example: *Avadon*

Touch: Gestures

- Can also leverage device **gestures**
  - Manipulation strokes common to device
  - **Example**: Pinching for zoom
  - **Example**: Rotating (object, screen)
- Natural for camera control
- **Design Approach**:
  - Think about how used in normal apps
  - How do you leverage this in a game?



Basic Gestures

Tap

Double Tap

Tap and Hold

Flick

Pinch

Spread

Rotate

Drag (Scroll)

Simple Multitouch Gestures

Two Fingers

Tap

Tap/Press

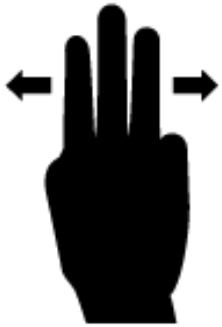
Double Tap

Drag

Three Fingers

Tap

Double Tap



Swipe

Drag

Simple Multitouch Gestures

Two Fingers

Tap

Sort of possible to get position

Tap/Press

Double Tap

Drag

Three Fingers

Tap

Getting position is largely hopeless

Double Tap

Swipe

Drag

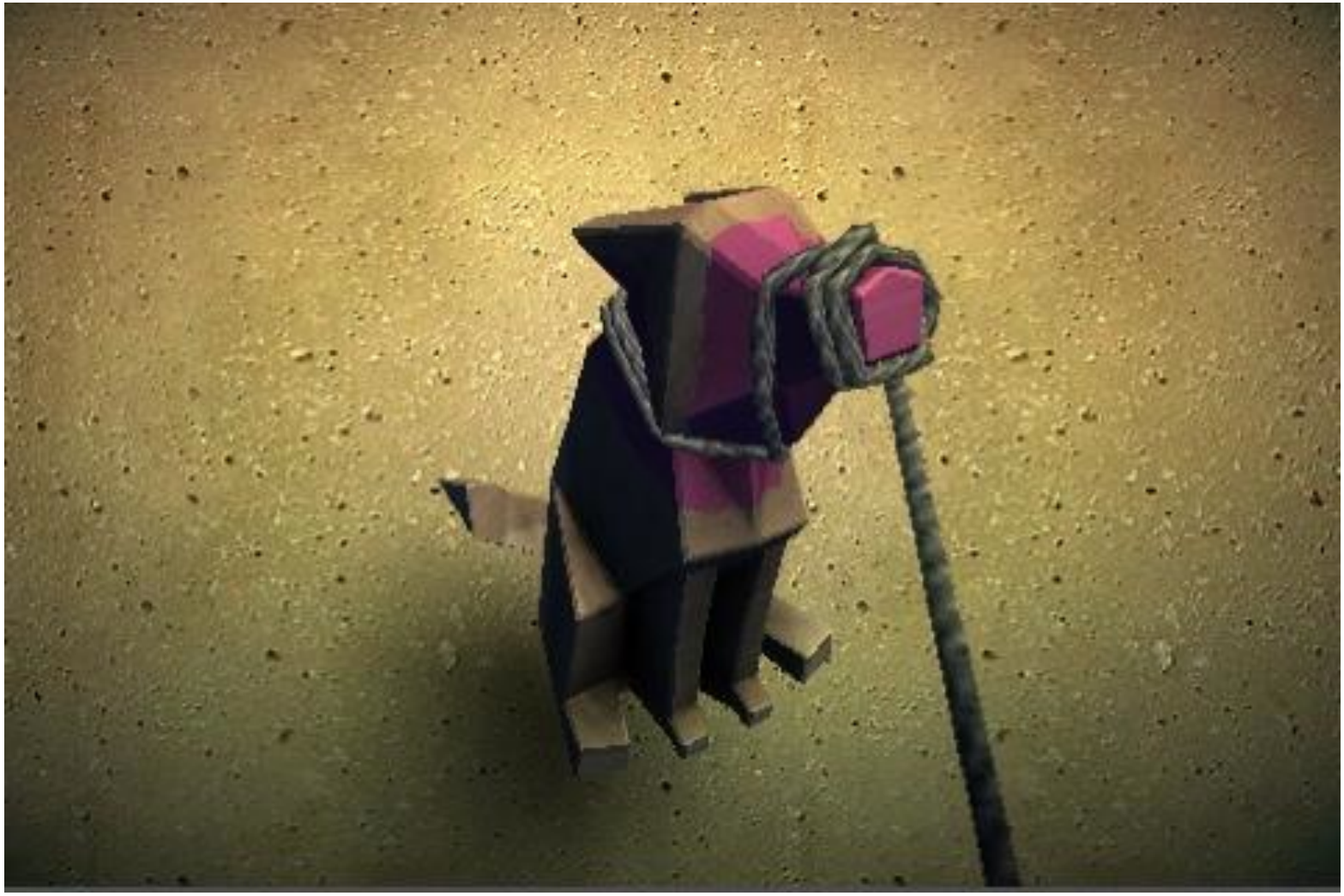
4152 Example: *Phantom Escape*

Touch: Natural Controls

- Successful games strive for **natural controls**
  - Verb controlled by a single movement/gesture
  - Gesture has a very natural physical feel to it
  - Maps naturally on to the action in the game
- **Examples**
  - Cutting (Cut the Rope)
  - Tracing (Flight Control)
  - Dragging (Nitali)
  - Pulling (Angry Birds)

4152 Example: *Flick Ship Spaceship*

Example: *Zen Bound*

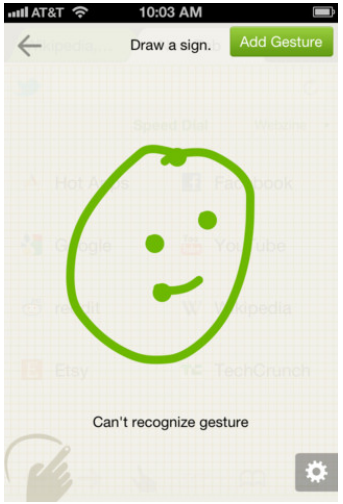


Example: *The Room*

Example: *Monument Valley*

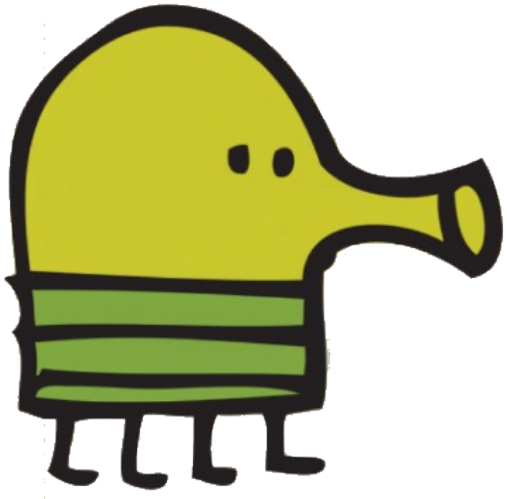
Custom Gestures: A Warning

- Leverage built-in gestures APIs
  - iOS/Android have libraries
  - Easy to use and stable
  - Custom gestures can slow production, *derail the game*
- But not all gestures supported
  - Circle around a target
  - 3 finger support only in OS X
  - Complex "spell wards"
- Borrow free libraries!



Touch: Avatar Controls

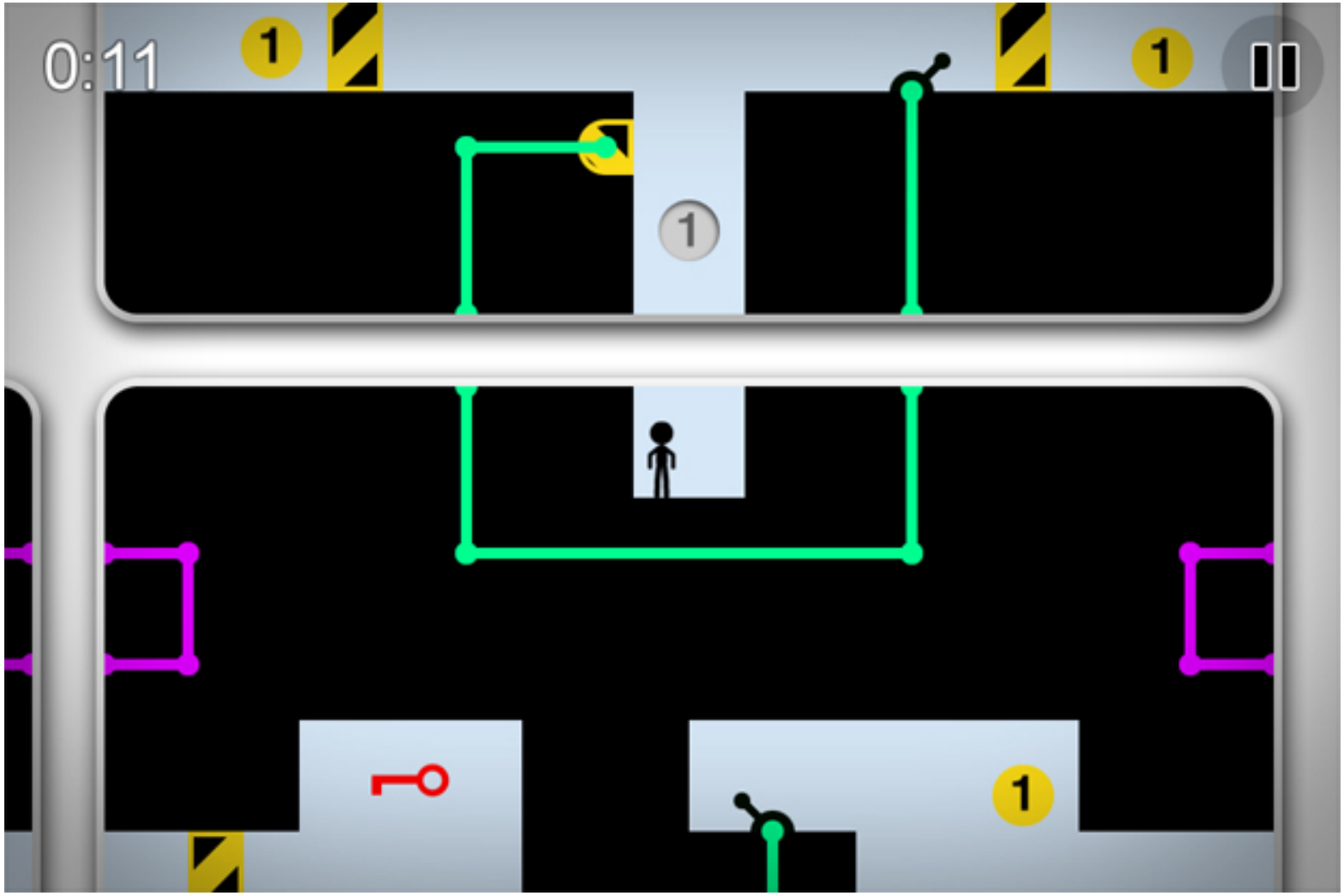
- Several (non-joystick) options for movement
  - Drag the character
  - Point to a waypoint
  - Point to direction



- But how to indicate avatar actions?
  - Want to move and act at same time
- **One Solution**: put actions into movement modes
  - Drag versus waypoint
  - Press+hold drag versus drag

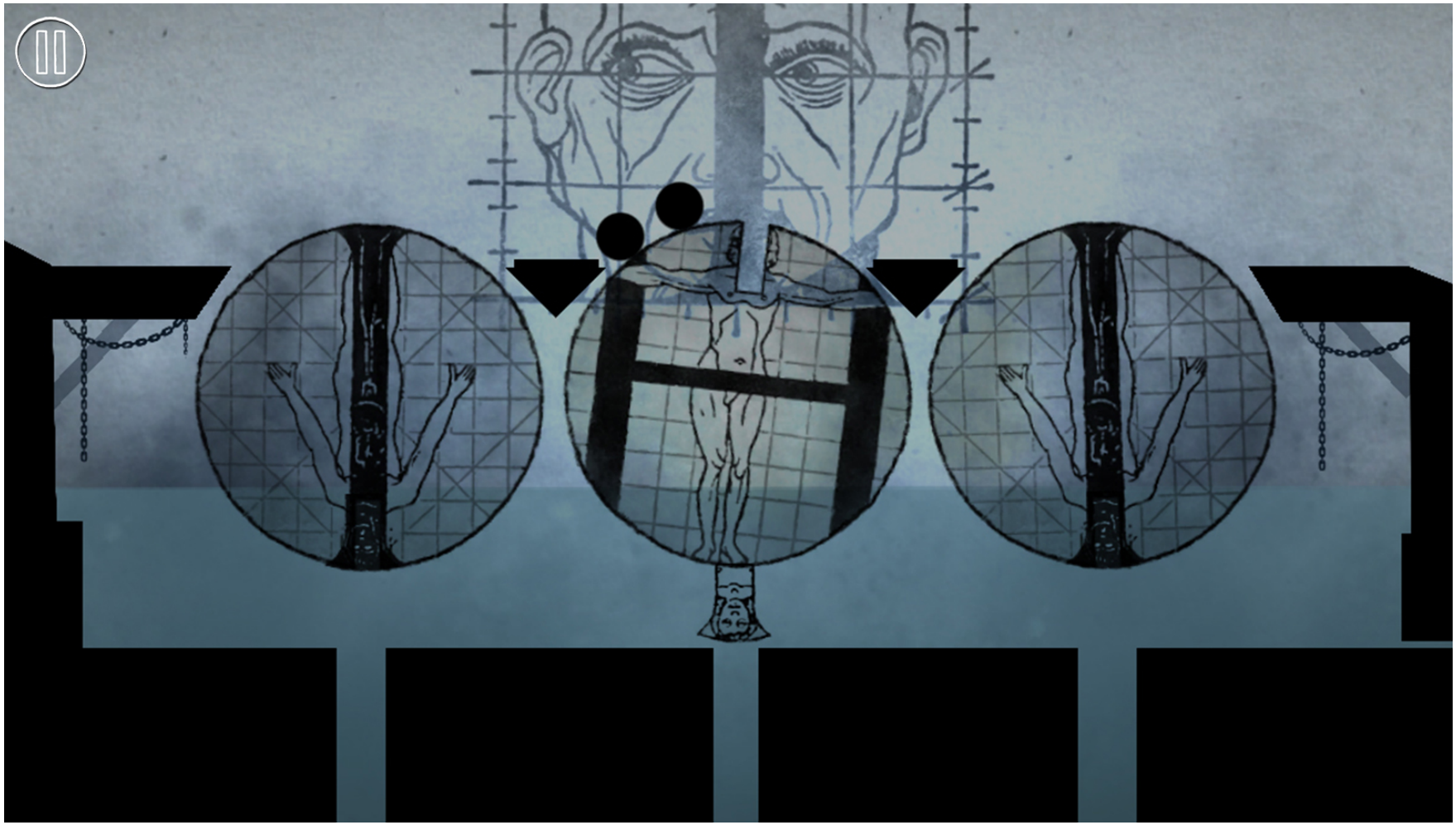
Example: *Spider*

Example: *Continuity 2*

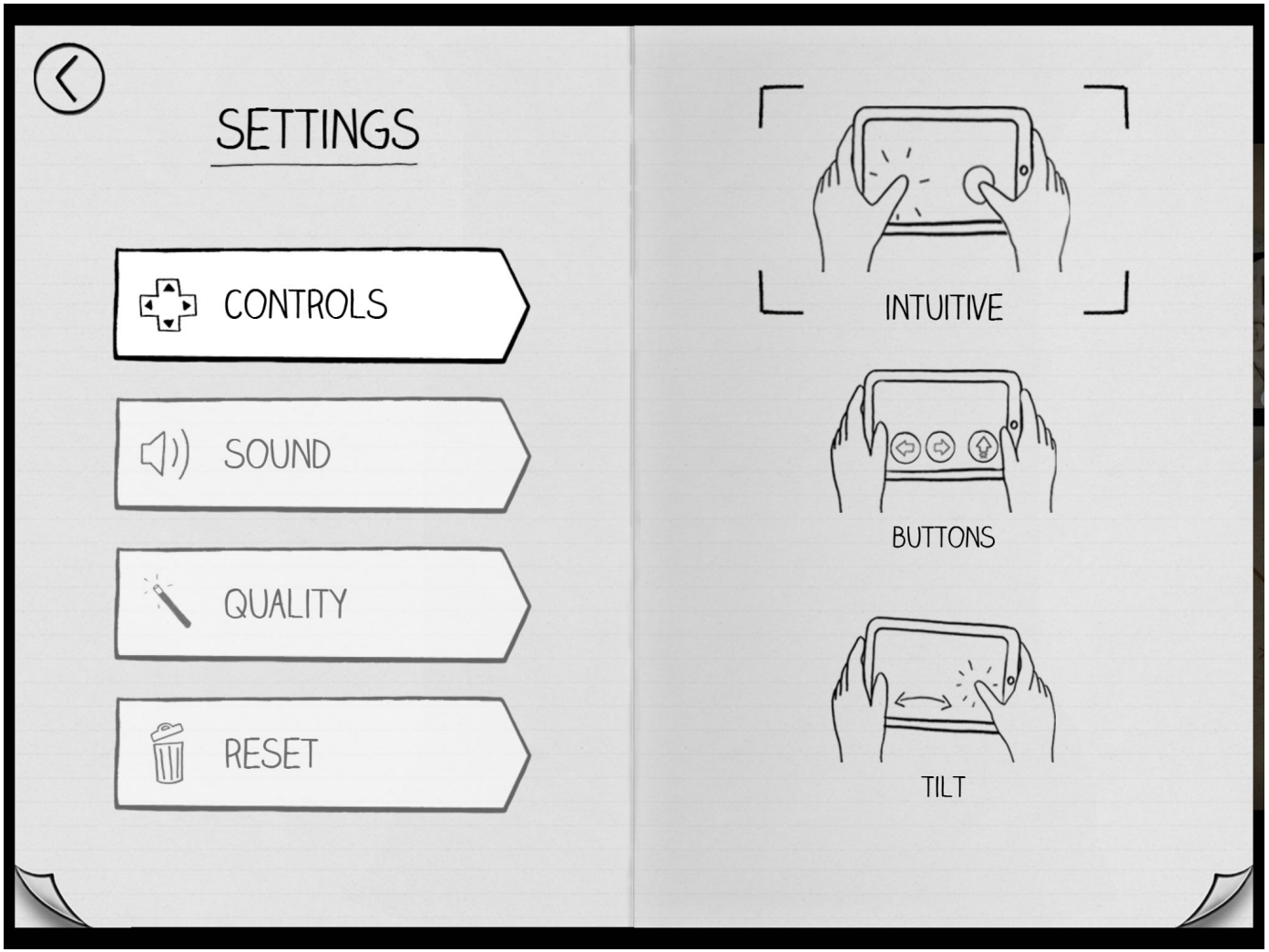


Example: *Night Sky*

Example: *Type:Rider*



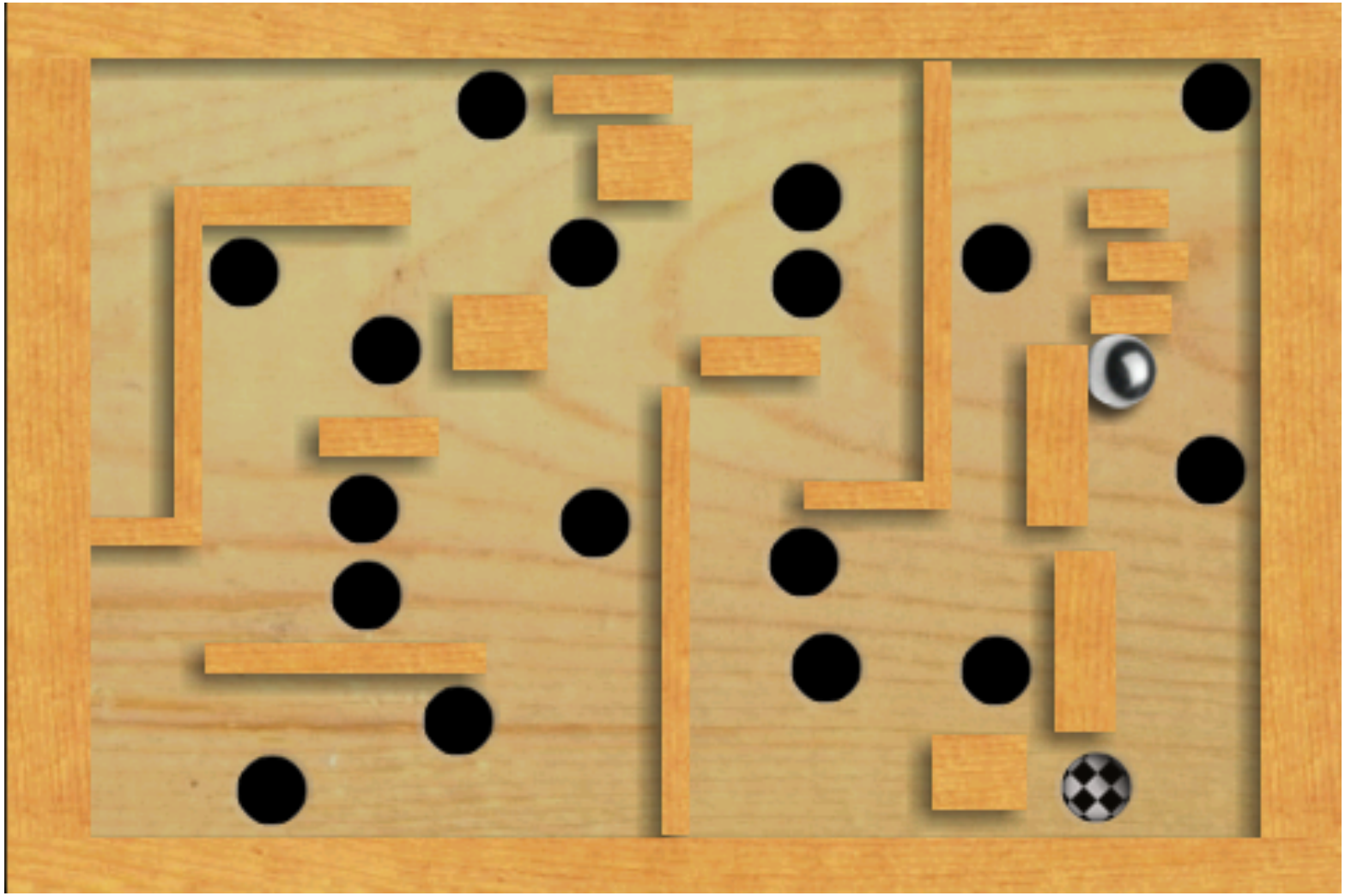
Example: *Type:Rider*



Accelerometer: Basics

- **Can** detect rotational movement
  - Rotate from flat plane
  - Rotate around edge
- **Cannot** detect other movement
  - Lateral movement of device
  - Absolute position of device
- Ideal mechanic for
  - Marble-style games
  - Steering/On-rails games

Example: *Labyrinth*

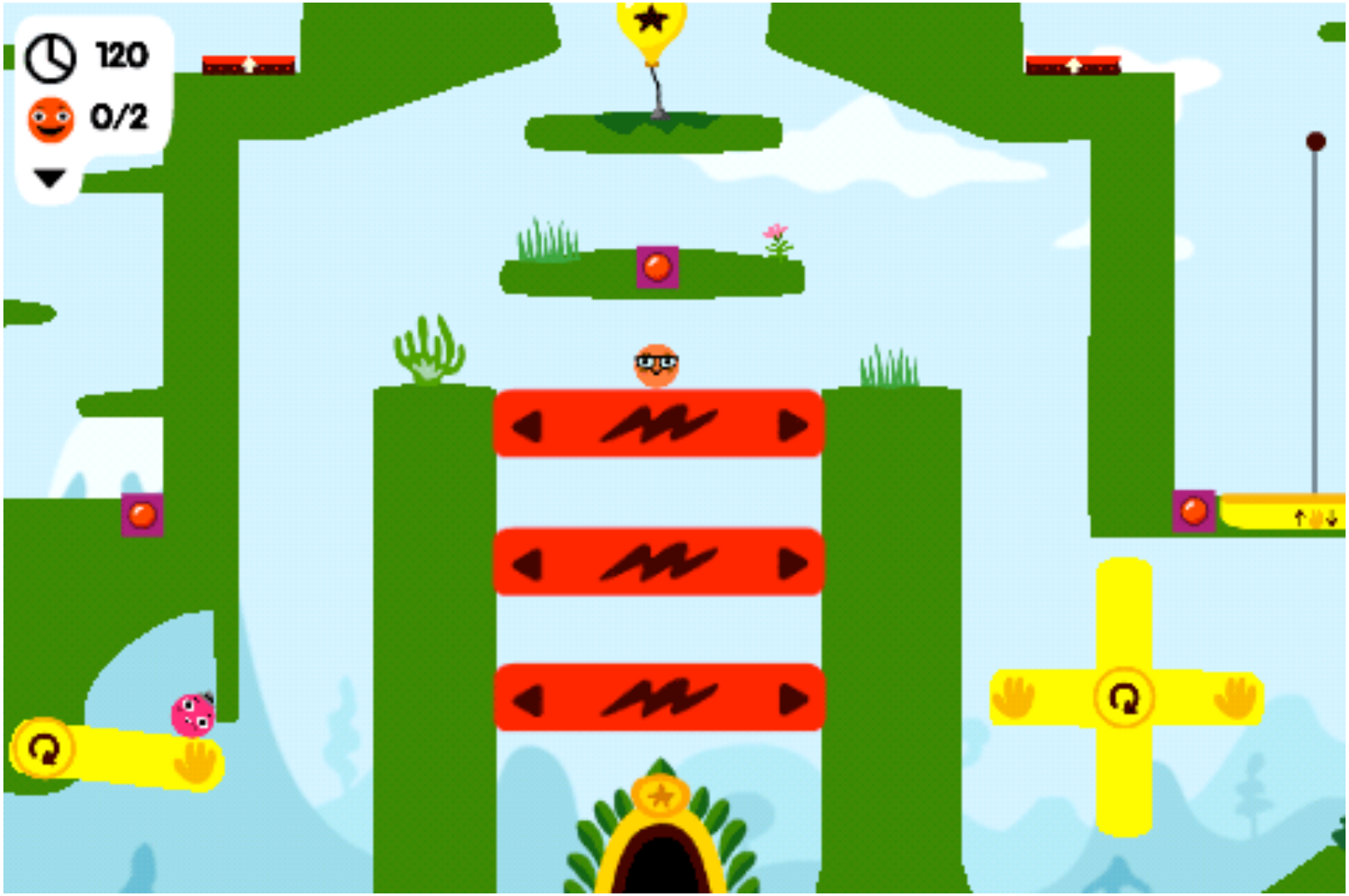


Accelerometer + Touch

- Solves the problem of actions
  - Use accelerometer for movement
  - Use touch for other actions
- But have to hold the device
  - Hard to gesture with hand that holds
- **Idea**: Keep actions unobtrusive
  - Avoid "button mashing" mechanics
  - Allow touch to use thumbs as much as possible

Example: *Mass Effect Galaxy*

Example: *Rolando*



Example: *Knightmare Tower*

Accelerometer: Challenges

- The control device is the **display**
  - Extreme controls make game hard to see
  - Even worse when combine with touch
- Even basic movement is a **challenge**
  - Hard to quickly change directions
  - Prone to overcorrection
  - **Example**: *Labrynth*

Final Word: Know Your Audience

- Phone games are meant for "quick play”
  - Must be able to start, play, and save in 2 minutes
  - Should be able to pick up where left off quickly
  - Controls should be (relatively) simple
- Tablet games can be more complex
  - Supports longer play units (why?)
  - Larger screen permits more complex controls
  - Games are closer to PC indie games
  - And can also cost more!

Final Word: Know Your Audience

- Phone games are meant for "quick play”
  - Must be able to start, play, and save in 2 minutes
  - Should be able to pick up where left off quickly
  - Controls should be (relatively) simple
- Tablet games can be more complex
  - Supports longer play units (why?)
  - Larger screen permits more complex controls
  - Games are closer to PC indie games
  - And can also cost more!