Moving Platforms, Scrolling Backgrounds in Game Maker

Foundations of Interactive Game Design
Professor Jim Whitehead
February 25, 2008
Upcoming

• Second exam is this Wednesday, February 27
  ‣ List of potential questions in a few minutes

• No class Monday, March 3
  ‣ Jim is attending a workshop

• Final gamelog assignment
  ‣ Due next week on Wednesday, March 5
  ‣ New due date, was Monday, March 3

• Final project due Monday, March 10
Game Design Workshops

• **Game Maker**
  ‣ Wednesdays, 6-8pm
  ‣ Engineering 2, room 180 (Simularium)
    ❖ Enter on plaza level between E2 and JBE

• **RPG Maker**
  ‣ Wednesdays, 5-7:15pm, Engineering 2, room 280
  ‣ **NEW:** Thursdays, 5-7:15pm, Engineering 2, room 215

• **CS 20/C# and XNA Game Studio Express**
  ‣ Thursdays, 4:30-7pm
  ‣ Engineering 2, room 399 (third floor, by elevators)
Exam #2

• The second midterm exam is this Wednesday
  ‣ Will primarily cover material between first and second exam
  ‣ May draw upon concepts and material from the entire class
  ‣ No Game Maker questions

• Will give list of potential questions on following slides

• Exam #2 will be similar in difficulty and format to the first exam
  ‣ Mostly short answer questions
Potential Exam Topics

• As Univ. of California students, you are expected to be able to assess complex material and make judgments concerning its relative importance.
• That said, it can be helpful to have some input from the Professor to help focus studying activity.
• The following are questions/material that are likely, but not guaranteed to appear on the exam.
• Anything covered in class or in the assigned readings since the last exam may appear, even if not explicitly mentioned today.
Potential Exam #2 Topics and Questions

• What is a core game mechanic?
  ‣ Be able to define, and identify in a game of choice, or demonstrated in class

• What are the core elements of a platform game?

• What is the core mechanic of a platformer?

• What benefit does scrolling provide for game design?

• What were some of the design challenges of moving platformers from 2D to 3D?

• What was the first platform game?
  ‣ Donkey Kong

• What were the first scrolling platform games?
  ‣ Pac-Land, Wonder Boy, Mario Bros.
Potential Exam #2 Topics and Questions

• What is the difference between challenge and conflict?

• Know Crawford’s dimensions of challenge
  ‣ Cerebellar, Sensorimotor, Spatial Reasoning, Pattern Recognition, Sequential Reasoning, Numerical Reasoning, Resource Management, Social Reasoning
  ‣ Be able to give an example of each.
  ‣ Given a game, be able to identify the kinds of challenge it provides.

• Be able to identify several forms of conflict
  ‣ Given a game, be able to identify the types of conflict the game supports

• What is Crawford’s definition of interactivity?
  ‣ What are the four important elements of his definition?

• Given a game example, be able to describe whether it exhibits high or low interactivity (according to Crawford)
Potential Exam #2 Topics and Questions

• What two seductions must a designer accomplish in creating their game?
• How does the act of submitting to game rules lead to player pleasure?
• What is the lusory attitude?
• What is autotelic play?
  ‣ What are some of the implications of games being mostly autotelic?
• What are the elements of a game’s reward structure?
  ‣ Types and frequency of rewards
  ‣ Types and frequency of punishments
• Be able to give examples of types of rewards in a game
  ‣ Given a game, be able to identify the kinds of rewards and punishments it provides.
• Be able to give examples of short and long term goals.
Potential Exam #2 Topics and Questions

• What was the first widely played computer game
  ‣ Pong

• Who were the key designers of Pong?
  ‣ Al Alcorn, Nolan Bushnell

• What company did they work for?
  ‣ Atari

• What is the narrative of a game?

• What is the difference between embedded and emergent narrative?
CS 20 Exam Questions

• Be able to understand the operation of a simple code example using C# and XNA GSE
• What is a spritebatch?
• What is a texture2D?
• Basic knowledge of controller input
• Different types of controller input
• Bounding box collision detection approach
Creating a Moving Platform
Making a Moving Platform

• Two main issues in creating a moving platform
  ‣ Need to make platform move along a consistent path
  ‣ Once the player has landed on the platform, the player’s speed must match that of the platform

• Approach
  ‣ Use a path to control movement of the platform
  ‣ Add a check for a collision with the player object to the platform
    ❖ Update player movement if true
• An advanced feature
  ‣ Have I mentioned you really want the advanced version?

• A predefined pathway enemies can follow
  ‣ Pattern of movement of enemies in Shmup
  ‣ Pattern of movement of enemies through a level in a Platformer
    ❖ Think about barrels in Donkey Kong: always follow one of small set of paths

• To create
  ‣ Add.. Add Path
  ‣ Define set of points on the path
  ‣ Can have different speeds at different points on path

• To make work
  ‣ Connect with an object
    ❖ In create event, use action to start path (on Move tab)
  ‣ Can specify repeating/one-time, speed, relative or absolute coords
Paths and Movement

• An object following a path has:
  ‣ No x or y velocity
    ❖ hspeed and vspeed are 0
    ❖ Attempts to set the speed are ignored
  ‣ No acceleration
    ❖ Gravity does not affect an object following a path

• Position is changed between Step and End Step events
  ‣ Attempts to manually alter position of object are ignored in Begin Step and Step Event
  ‣ Changing position in End Step event leads to unusual behavior, as it reverts back to its path movement during Begin Step and Step events, but then goes to assigned position in End Step
    ❖ Don’t recommend this
Paths for Platforms

• When making a path for a platform
  ‣ Recommend making a closed path
    ❖ That is, go out some distance, then come back to where you started
  ‣ In Create event for moving platform object
    ❖ Use “Set path for the instance”
    ❖ At end: continue from start
      • “reverse” appears to be buggy
    ❖ Relative: relative
  ‣ If you use “reverse” I have seen platforms start oscillating wildly after about 5-6 back-and-forth movements
Updating Player Position: X

• Need to adopt two different approaches for x & y axes
  ‣ X-axis movement is not affected by gravity
  ‣ Y-axis movement is affected by gravity

• We’re using the all-in-one collision detection approach in the player object
  ‣ Would like to avoid adding additional complexity to this
    ❖ But end up having to do so for y-axis – x-axis is fine, though
  ‣ So, add collision detection logic to moving platform
  ‣ A bit backwards: have platform determine if player has landed on it
    ❖ Usually think of the player colliding with the platform
• Add to “End Step” event
  ‣ Needs to be End Step to come later than the player object’s position updates
    ❖ Part of all-in-one platform collision detection, which is in End Step
    ❖ Also needs to come after path update, which is between Step and End Step
  ‣ Use “If there is an object at a position”
    ❖ Blue ball in Octagon on Control tab of MovingPlatform object
    ❖ Object: player object (Ball)
    ❖ x: 0
    ❖ y: -5 (not larger than height of ball)
    ❖ Relative: yes
    ❖ NOT: no
  ‣ Checks to see if the player is in any of the pixels immediately above the platform
Updating Player Position - X (3)

- If yes
  - Use “Set the value of a variable” (grey square with “Var” on Control)
  - Object: self (platform)
  - Variable: ball.x (this is the Ball’s x value we’re updating)
  - Value: x - xprevious
    - Add this step’s movement for the moving platform to the x value of the player
    - Causes the player avatar to move with the platform
    - Note that path following does not change the speed of the platform object
      - So, cannot just add MovingPlatform.hspeed, even though that may seem reasonable.
    - This behavior can be inherited for multiple platforms
  - Relative: yes
Updating Player Position - Y

• We want to keep gravity active
  ‣ Makes falling off a platform realistic

• Implications for all-in-one
  ‣ Two cases
   ❖ Platform moving up
     • Platform has moved into ball by start of End Step (all-in-one)
     • Need to move ball out of platform
     • Horizontal collision check is before vertical collision check
     • Since ball is in platform, no horizontal movement occurs, since there is immediate collision
     • Need to fix this
   ❖ Platform moving down
     • Platform has moved below ball by start of End Step
     • If gravity large enough, will keep player on platform due to normal all-in-one behavior
     • Horizontal movement works fine
• Simple fix for being inside platform
  ‣ Move above platform
  ‣ Set a downward velocity
  ‣ Let normal downward movement logic occur
    ❖ Move to collision
    ❖ Causes ball to move down to platform
  ‣ Have to add this into all-in-one
Updating Player Position - Y (3)

- Check for object at a position
  ‣ Applies to: self
  ‣ Object: MovingPlatform
  ‣ x:0, y:0
  ‣ Relative: yes

- Jump to position
  ‣ Applies to: self
  ‣ x:0, y: -5
    ❖ The y movement needs to be larger than the platform's movement in y each step.
  ‣ Relative: yes

- Set vertical speed
  ‣ Applies to: self
  ‣ Vert. Speed: 5 (same as abs(y offset) above)
Final result

• Platforms can move in complex patterns in x and y
  ‣ Player avatar will stay on the platform
  ‣ Gravity works
  ‣ Player avatar can move left/right and jump at any time

• Can have multiple moving platforms at once
  ‣ Place horizontal fix-up in MovingPlatform parent
    ❖ Make this parent solid, but invisible
  ‣ Have each moving platform inherit from MovingPlatform
    ❖ In create event, set unique path for this platform
Creating a Scrolling Background
• Background
  ‣ Represents a static background image
  ‣ Defined separately, then combined with a room
  ‣ Can move, or be motionless

• Many uses
  ‣ Moving starfield
    ❖ Create starfield bitmap image
    ❖ Define as background
    ❖ Have it move backwards: creates sensation of motion
  ‣ Reduce boundary objects
    ❖ In tile-based games, need many tiles to create an interesting level
    ❖ With objects, would need separate object for each tile type
      • Can slow a game down
    ❖ Instead, create background image
    ❖ Then, use single, invisible boundary object for collision detection
• **Game window**
  ‣ The window on screen that shows the game
  ‣ 640 pixels wide x 480 pixels high by default in Game Maker

• **Room**
  ‣ Defines a level
  ‣ May be larger than the game window

• **View**
  ‣ An interesting subset of a room
  ‣ Often, but not always, the size of the game window

• **Port**
  ‣ The dimensions of a view in the game window

• **Views and ports are advanced Game Maker features**
  ‣ You *really* do want the Advanced version
Rooms, Views, and Ports Example

Room (solid line): Holds entire level (not all visible on screen at once)

View (dotted line): Portion of room to make visible, usually portion of level visible to player

Port (light solid lines): Mapping of View to Window. Usually 1 to 1, but doesn’t have to be. Radar scope effect from Defender: map View to smaller area in Window.

Window (heavy solid line): What the player is actually seeing
Using Views to make a Shmup

• Desired behavior
  ‣ View slowly moves through level at constant speed
  ‣ Player must always be within the view
    ❖ That is, player cannot move offscreen
    ❖ Player is pushed along if they lag behind: must make forward progress
  ‣ Player bullet can only destroy enemies that are onscreen

• Need to use variables and conditionals to make this happen
Moving Through Level at Constant Speed

- Involves moving view slowly through room

**Approach**

- Every game tick, move left boundary of view forward by 2 pixels
- Left boundary is held in variable `view_xview[0]`
  - Note that the 0 means View 0: would need to change for other views
- So, put increment into Step Event for player object
  - “Set the value of a variable” - dark gray square box with “Var” in Control panel
  - Variable: `view_xview[0]`
  - Value: 2
  - Check “Relative” box - this means, add 2 to the current value
Keeping Player on Screen

- Need to check for player x less than left hand side
  - view_xview[0]
- Need to check for player x greater than right hand side
  - view_xview[0]+view_wview[0]
- Need to check for player y less than top
  - Less than 0
- Need to check for player y greater than bottom
Bullet Only Affects On Screen Enemies

• Need to check if the bullet has gone outside of the view
  ‣ That is, is the bullet x greater than right side of view
  ‣ Check for bullet x greater than view_xview[0]+view_wview[0]
  ‣ Outside view event did not work for me

• If bullet has gone too far
  ‣ Destroy the bullet
  ‣ If the bullet has been destroyed, cannot affect enemies