Creating a Shmup in Game Maker

Foundations of Interactive Game Design
Professor Jim Whitehead
February 5, 2007
Upcoming Events and Assignments

• **Game Concept Document**
  ‣ Due Wednesday
  ‣ More details in a few slides
  ‣ Description, Template, Evaluation criteria all on website

• **Gamelog**
  ‣ Due Friday
  ‣ Game of your choice

• **Work Breakdown and Schedule**
  ‣ Due next week, on Wednesday

• **Review Session**
  ‣ Tonight! Here in Media Theater
  ‣ 7:30PM start time
  ‣ Bring your review materials!
RPG Maker/Game Maker/Art help

- **Wednesday:** RPG Maker design session
  - Led by Nate Emond <llama971@gmail.com>
  - 5:30PM, E2 215
  - Note different room for this week

- **Thursday:** Game Maker help/design session
  - Earth & Marine Sciences, room B214
  - 4-5:10PM

- **Want help with your game art?**
  - Jemima Thomas <luperis@yahoo.co.uk>
    - Took 80K last year
    - Created *excellent* sprites for her game
    - Is interested in working with a few teams to help them with game artwork
    - Contact her via email to make arrangements

- **Game Maker reduce cost license keys - see me**
Game Concept Document

• Setting
  ‣ You have 5 minutes with an executive at a game publisher
  ‣ Want them to provide you funding to develop your game

• Goal
  ‣ Write a convincing document that gives highlights of the game
  ‣ Must be concise, punchy, to the point
  ‣ Describe important aspects of
    ❖ Gameplay - what does the player do?
    ❖ Challenges - what are important challenges faced by the player?
    ❖ Fictional setting - what is the background for the game’s story?
    ❖ Levels - briefly, what will the levels look like?
    ❖ Art element - what is the artistic feel of the game?
    ❖ Team - why are your team members the right ones for this project?
    ❖ Audience - who is the audience for the game?
Game Concept Document Examples

• On document camera, examples of game concept documents
Creating a Shmup in Game Maker

- Show brief demo of Gradius
- Important mechanics
  ‣ Steady movement through level
  ‣ Sensation of movement
  ‣ Spaceship cannot move offscreen
  ‣ Spaceship fires and destroys enemies
  ‣ Varied enemies
  ‣ Powerups
Backgrounds

• 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
Rooms, Views, and Ports

- **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 in 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
  ‣ Uh, oh, sounds computer sciency
  ‣ This is easy, really
• **Variable**
  ‣ A named value
  ‣ Can be **read**: look up the value
  ‣ Can be **written**: change the value to something new

• **Examples**
  ‣ The horizontal location of the player in the room is **named** “x”
    ✤ Its **value** is a number between 0 and the width of the room
    ✤ For the in-class demonstration, it varies between 0 and 3000
  ‣ The vertical location of the player in the room is **named** “y”
    ✤ Its **value** is a number between 0 and the height of the room
    ✤ For the in-class demonstration, it varies between 0 and 640
  ‣ Need names to identify the value we want
Variables (2)

• Full disclosure
  ‣ There appear to be two kinds of variables
  ‣ Variables defined on object instances
    ❖ x and y giving player position are defined on the player object instance
    ❖ Most of the dialog boxes in Game Maker default to instance variables, so don’t have to worry about this point usually
  ‣ Global variables
    ❖ We’ll see these in a few slides, things like view information
    ❖ Seem to be able to access these anywhere

• In Game Maker
  ‣ Can set the value of a variable using square “Var” action
    ❖ Under “Control” tab in Object window
    ❖ That is, setting a variable is a kind of action
      • Set a variable in response to an event
• If .. then behavior

• Example
  ‣ **If** the player horizontal position is outside the view...
  ‣ **Then** make sure the player stays inside the view

• Conditionals in Game Maker are represented by octagon shapes
  ‣ Found under “Control” tab
  ‣ Are an action you can take in response to an event

• Often want to check condition all the time
  ‣ Place these in the step event
  ‣ Step event is called every game tick (1/30th of a second)
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 \texttt{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: \texttt{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]

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

• 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