Key Abstractions in Game Maker

Foundations of Interactive Game Design
Prof. Jim Whitehead
January 24, 2008
Upcoming Assignments

• Today: Second gamelog assignment
  ‣ Must write your gamelogs about a game from classics list
  ‣ Have until midnight today to complete
• Next Wednesday: Game Concept Document
  ‣ A compelling document that sells your game concept
  ‣ Title page
    ❖ Title of game, name of group, name of team members, sample artwork
  ‣ Overview page
    ❖ Table at top: game genre, platform (PC/GameMaker, PC/RPG Maker, etc.), team size
    ❖ Key points section
      • Bulleted list of important elements of gameplay
      • Goal of game, what makes game unique, main characters, main fictional elements
      • Sample artwork image to give feel of the game
  ‣ Biographies
    ❖ True, pocket biographies of each team member (1-2 paragraphs each) stressing experience that makes you a strong game designer
  ‣ 1-3 pages giving a textual description of the game
    ❖ Fictional background, brief description of characters, goal of player in game, how does player interact with the game, brief description of levels, game audience, other important elements as needed.
  ‣ 1-2 pages of sample conceptual artwork
    ❖ Hand-drawn sketches are fine
• Start early!
Announcements

• www.soe.ucsc.edu/classes/cmps080k/Winter08/
  ▸ Has syllabus, assignment descriptions, exam days, final exam time, link to gamelog site, links to tool descriptions, and more…

• Porter Video Games Tournament
  ▸ Fifth Annual Video Game Tourney
  ▸ Saturday, Jan. 26, 2008, 3pm-3am, Porter Dining Hall
  ▸ Register/details: portervgt5.0@gmail.com
  ▸ Facebook group: “PorterVGT 5.0”
  ▸ They need game machines

• Game club announcement
  ▸ Meet Thursdays, 2:15pm
  ▸ Conference room A
    ❖ Go in entrance next to “Express It”
• Help session for making your Mac dual boot
  ‣ Today, 3pm-4pm, Engineering 2, room 392
    ❖ E2 is the glass-fronted building, not the big concrete one
    ❖ Third floor, use hallway by elevators
  ‣ If you can’t make this time, contact Ian Rickard to arrange help time
    ❖ inio@soe.ucsc.edu

• RPG Maker for ITS labs
  ‣ Will have a few full copies available, but may take a few weeks
  ‣ Request has gone in

• Weekly help session for CS 20 (C# and XNA Game Studio Express)
  ‣ Thursday, 4:30-7pm
  ‣ Engineering 2, room 399 (third floor, by elevators)
  ‣ Starts next week
What is Game Maker?

• Game Maker is a tool mostly for creating 2D games
  ‣ (though there is limited 3D support in the latest version)

• It uses a property-sheet approach to create a game
  ‣ Create a set of sprites, objects, rooms, sounds
  ‣ Each one of these has a series of properties that you can set – these affect its behavior

• Game Maker does not require programming
  ‣ It does have a scripting language (Game Maker Language, GML) you can use for more advanced features
  ‣ It’s possible to make very interesting games without knowing this language

• Though, of course, if you do learn the language, you’ll be able to make more complex games
  ‣ Creating games is somewhat less tedious using GML
Simple Ball Example

• In-class demonstration of using Game Maker to create the simple ball game
  ‣ This example is described on pages 12-14 of the Game Maker manual
  ‣ It is highly recommended that you also follow this example and create the simple game
  ‣ Provides a relatively gentle introduction to how to use Game Maker
Sprites

• Sprites are the visual representation of objects in the game
  • A sprite can be a single image that never changes
  • You can also animate a sprite by having a sequence of images display in a row
  • Show Pacman sprites in Game Maker

• To add a sprite:
  • Menu Add -> Add Sprite

Source: www.molotov.nu
Sprite Issues

• The “Transparent” choice when creating a sprite indicates whether the background color of the sprite should be transparent
  ‣ Most of the time, this is the correct choice
  ‣ Only want your item to show, not a square with the item inside it
• Exception might be when you have a wall
  ‣ Even then, might take advantage of transparency to create interesting effects
• Background color is the leftmost, bottommost pixel of the image

Source: www.molotov.nu
Objects (Briefly)

- Objects add behavior to Sprites
  - A Sprite just tells you what an item looks like
  - A Sprite conveys no information on how it works in the game world
  - Does it move, bounce, explode, etc? A Sprite doesn’t say anything about these issues – Objects do

- Every character, monster, wall, item, ball in the game has its visual representation made with a Sprite, and its behavior determined by an Object

- **Sprite**: visual appearance only
- **Object**: behavior
Basic Model of an Object

• **An Object** reacts to **Events** by performing one or more **Actions**
  ‣ For example, a *ball* reacts to a *collision* with a *wall* by *bouncing*
    ❖ Ball and wall are objects
    ❖ A collision is one kind of event
    ❖ Bouncing is one kind of action

• **Event**: a notable occurrence in the game
  ‣ An object hitting another object
  ‣ A mouse click on an object
  ‣ An object being created
  ‣ A clock tick

• **Action**: an activity that can occur in the game
  ‣ Bouncing off a wall
  ‣ Starting to move in a particular direction
Important Events

• **Create**
  ‣ When an object is created
  ‣ Useful for setting the initial motion of an object

• **Collision**
  ‣ When two objects collide, each object receives this event
  ‣ Useful for collisions with walls, enemies, bullets, etc.

• **Step**
  ‣ The game progresses in “ticks”
  ‣ Each tick is 1/30th of a second
  ‣ Receive a step event each tick
  ‣ Useful for updating game state, checking for whether the player is close to another object, and so on
Rooms

- A room is where the action takes place in a game
- Can represent many things:
  - Opening screen
  - Fields of play
    - Levels, dungeons, rooms, villages, outdoors, etc.
  - Help screens
  - Cut scenes
- Typically each level is a separate room in Game Maker
- Class demos have been taking place in default room
  - Automatically created by Game Maker
Gathering Keypresses

• Three events control key input

• **Keyboard Event**
  ‣ Generates an event as soon as a key is pressed
  ‣ Also continues to generate events while the key is kept pressed
  ‣ Good for repeated firing of a weapon
  ‣ Generally not good for movement, or jumping

• **Key Press Event**
  ‣ Generates one and only one event for a key being pressed down
  ‣ Good for changing direction of a player agent
  ‣ Good for starting some action (movement) of the player agent

• **Key Release Event**
  ‣ Generates one and only one event for an already down key being released
  ‣ Good for ending something started with a Key Press event
Demonstration of Key Press Differences

• *Demonstration with Game Maker, keypresses, and sounds*
  ‣ Keyboard event will cause sound to be played repeatedly, while key is held down
    ❖ Challenging to get just a single sound instance to play, as you typically get many keyboard events for pressing a key
  ‣ Key Press event
    ❖ Will cause “down” sound to play just once
  ‣ Key Release event
    ❖ Will cause “up” sound to play just once
Moving a simple sprite

- Movement occurs when a movement action occurs in response to a user input event.

Example:
  - On an object, create a keypress event (up).
  - Associated action start moving in a direction
    - Applies to self
    - Click up arrow
    - Set speed to 5
    - Click “relative”
  - Repeat to create keypress events for down, right, and left
    - Change clicked arrow each time
    - Keep other settings the same
  - Place object in a room
  - Start game (Run -> Run Normally, or F5)