Announcements

• Read Chapter 6 Tips and Techniques for next class
• Reading quiz due next time
• New groups today
• Assignment 5 out
  – Part 1 due Thurs. Oct 21 and Part 2 due Oct 26
• Today
  – Interactive programming, event handlers
  – Create billboards
Control of Flow

• Control of flow – how the sequence of actions in a program is controlled
  – What action happens first, second, third, …. 

• In movie-style programs (Chaps 1-4) the sequence of actions is determined by the programmer
  – Creating a storyboard design
  – Writing program methods to carry out the designed sequence
Interactive Animations

• In interactive programs, the sequence of actions is determined at runtime, when the user provides **input**
  – Clicks the mouse
  – Presses a key on the keyboard

• Other sources of input are possible
Interactive Games

• In a video game where the user is guiding a spaceship, the sequence of actions …
  – Depends on what direction the user guides the ship
  – How fast the user presses the controls
• Each time the program runs, user input may cause a different sequence of actions
• Control of flow is “in the hands of the user”
Event Handlers

• An event may
  – Trigger a response, or
  – Move objects into positions that create some condition (e.g. a collision) that triggers a response

• An event handler is a method that is called to carry out the response.

• When an event is linked to an event handler, a behavior is created.
How does this effect your program?

• Our goal is to write interactive programs.
• The approach is the same as before, but the difference is now must be concerned with behaviors.
  – input from the user (events)
  – How objects respond to an event (event handler methods)
Example 1

- Build an air show flight simulator. The pilot (user) uses the biplane controls to perform acrobatic stunts.

- Problem: How do we write program code to provide a guidance system that allows the user to be the pilot?
Solution

- Use keyboard input
  - “F” key to move the biplane forward
  - Spacebar to make the biplane do a barrel roll
  - Note: other keys could be chosen

- Write event handler methods that respond to each key press

- Storyboards (next slide) and DEMO building world
Event Storyboards

- Since two keys are used, two events are possible – so two storyboard scenes

**Event**: Spacebar press

**Response**: Do together
- roll biplane a full revolution
- play biplane engine sound

**Event**: F-key press

**Response**: Do together
- move biplane forward
- play biplane engine sound

- Each storyboard outlines an event handler
  - Responds to a particular event
biplane.flyForward

- Do not modify the length of the sound
  - use “as is”
- Coordinate duration of move and play sound
  - Match duration of move to duration of sound
Events Editor - Linking

- Each event handler method must be linked to an event

1) Select “create new event”
 Then choose the type of event

2) A template linking is created
Events Editor – Linking (cont)

3) Select type of key for event

4) Select event handler method

Final result:
More Functionality

Events create new event

- When F is typed, do biplane.flyForward
- When Space is typed, do biplane.barrel
- When ↑ is typed, do biplane.flyDirection direction = up
- When ↓ is typed, do biplane.flyDirection direction = down
- When ← is typed, do biplane.flyDirection direction = left
- When → is typed, do biplane.flyDirection direction = right
Add a Billboard with Instructions

- Add an event “I” to make the instructions hide or show (create with paint, powerpoint or some tool and save as an image)
To drop in a Billboard

- “Make Billboard, import, picture appears in object tree”
Mouse Clicks

- Interactive programs – allow user to mouse click an object
  - Buttons in an interface
  - Targets in a game
  - Checklist of items on a form
- Will see how to pass information about a mouse clicked object to an event handler
Example 2

- Burning Building
- People are trapped in a burning building
- Select which person will be rescued
- See firetruck.savePerson
Storyboard

- Three people are to be rescued
- Could write 3 different methods

**Event:** click on guy1
**Responding Method:**
Save guy on first floor

**Event:** click on girl2
**Responding Method:**
Save girl on second floor

**Event:** click on girl3
**Responding Method:**
Save girl on third floor
A Better Solution

- Write one event handler
- Send in information needed for action

firetruck.savePerson:

**parameters:** whichFloor, whichPerson, howFar

Do in order
- point ladder at **whichFloor**
- extend ladder **howFar** meters
- **whichPerson** slides down ladder to fire truck
- pull ladder back **howFar** meters

What type are the parameters?
Three Events

- The argument sent to parameters depends on which person is mouse clicked.

- Note - we positioned fire truck so distance from floor X is X meters (to floor 3 is 3 meters)
Example 3 – put events in

- Zeus was a powerful god in Greek mythology. When Zeus was angry, he would shoot a thunderbolt out of the heavens to strike anyone who got in the way.

- The user will choose the philosopher who will be the next target of Zeus’ anger.
Storyboard

- Possible design – method with Object parameter named *who*, for object clicked

| Event: an object is mouse-clicked |
| Event handler: *shootBolt* |
| Parameter: *who* – object clicked |

Do in order

- prepare to strike object that was clicked
- thunder plays and lightning strikes object clicked
- lightning is repositioned for next strike

- The actions in storyboard are complex
- Break actions into simpler steps using stepwise refinement
A Driver

- shootBolt method - top level of our design
- It calls other methods and controls the overall action of the program – we call this a **driver**
One Link

- In the fire rescue example, we used three links – one for each person in the burning building. In this example, we use only one link by selecting “object under mouse cursor” as the argument.
Demo

- Test run of Zeus world – (this version doesn’t have the if statements from Chap 6 added in)
- When parameters are used in interactive programming – especially important to test that all possible parameter values work as expected
  - What happens if you click on each philosopher, one at a time?
- Also try things that shouldn’t work
  - What happens if you click on a column?
  - What happens if you click on a philosopher twice?
  - What happens if you click on Zeus?
Classwork today

• Create 2 worlds (or can combine them in one)
  – Problem 14, page 163
  – Problem 15, page 164 (can use any person)
  – Include instructions in both…(see handout)