Announcements

- Read Chapter 4, Section 2 for next time
- Assignment 3 storyboards due Tuesday
  – World is due next Thursday night, Sept 18!

What we will do today

- Lecture on Chap 4, Sec 1
- Classwork
  – Create three animations
    • Snowpeople mods including flipping hats
    • Helicopter
    • Cameras moving
  – Get checked off today and for last time

Larger Programs

- Programs start to increase in size – many lines of code
- Games and “real world” applications have thousands, even millions of lines of code
- Want to organize large programs into small manageable pieces
Classes, Objects and Methods

- Object-oriented programming uses classes, objects and methods as basic components.
- These components help you:
  - Organize large program into small pieces
  - Design and think about an intricate program
  - Find and remove errors (bugs)

In your programs, you’ve used

- Classes
  - In Alice, classes are predefined as 3D models

- Objects
  - An object is an instance of a class
    - Class: Chicken
    - Objects: Chicken, Chicken2

In your programs, you’ve also used

- Built-in (predefined) methods
  - Examples: move, turn to face, say

- World.my first method
  - Example: robot on the moon from chapter 2, wrote code where an alien surprised the robot
  - All the code in World.my first method

Modifying the Program

- Modify program to get robot to try twice to move toward the alien or the alien go up and down twice.
- To make modification, add more lines of code
  - makes the program code longer and more difficult to read and think about
- Show alien world from last time
A Solution

• A solution to the problem is to
  – Define our own method
  – Name the new method `surprise`

• Then, can drag-and-drop the `surprise` method into the edit box, just like the built-in methods

World-level method

• `surprise` is a world-level method because it
  – Is defined as a method for `World`
  – Has instructions that involve more than one object (`robot, alienOnWheels`)

Using the surprise method

• This method is executed by calling (invoking) the method from my first method

  ![Diagram of world-level method](image)

• For testing, invoke temporarily when world starts

  ![Diagram of testing](image)
Why write our own Methods?

- Saves time – can call method again and again without rewriting code
- Reduces code size – call method instead of rewriting same code
- Allows us to think at higher level
  - Think “surprise” instead of “alien moves up, alien says something, robot turns around…”
  - Technical term for “think at a higher level” is abstraction

World.myFirstMethod now

- Move robot forward twice as far by invoking “investigate” twice
Classwork today

- Modify snowpeople to add two methods
  - catchAttention
  - Flip Hats
- Move the camera with an object
  - skyride – download from CompSci 4 page
- Create airport/helicopter world with new method
  - circleTower