Note: thanks to Wanda Dann and Steve Cooper for slide ideas
Announcements

• Assignment 4 is due Thursday
  – Must include your code on Alice web page
    • Done automatically if you select the option

• Check consulting hours, lots of modifications this week and next!
What we will do today

• Lecture on Chap 4, Sec 3
  – Class-level Methods and Inheritance – Part 1
• Classwork
Animated Actions

• Some actions are more naturally associated with a specific class of objects rather than the overall world

• Examples
  – A person walking
  – A wheel rolling
  – A fish swimming
Class-level Methods

• Write a method to add abilities/functions to a specific class of objects
  – Class-level method
  – NOT world-level method

• Now show how to build class-level method
An Example

• How can we create a skate method for ice skater objects?

We need to:
1) Associate the new method with an ice skater
2) Write the new method to animate the ice skater

NOTE: ice skater in the book and in Alice don’t quite match up in the names of body parts.
The solution

To associate the animation with the ice skater

- Select iceSkater tile in Object Tree
- Select methods tab in details area
- Click on “create new method” button
Storyboard for *skate*

**Skate:**
Do Together
move skater forward 2 meters
Do in order
slide on left leg
slide on right leg

- **The slide actions**
  - Require several motion instructions
  - We’ll break these two actions into smaller pieces
  - Technique is *stepwise refinement*
Refined storyboard for skate

Skate:
Do Together
  move forward 2 meters
Do in order
  slideLeft
  slideRight

slideLeft:
  Do in order
  lift right leg and turn upper body forward
  lower right leg and return body upright

slideRight:
  Do in order
  lift left leg and turn upper body forward
  lower left leg and return body upright
Writing the code

• Next step – translate design into code
• For *slideLeft*, possible translation is:

<table>
<thead>
<tr>
<th>Design Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lift the right leg</td>
<td>Turn the right leg forward</td>
</tr>
<tr>
<td>Turn upper body forward</td>
<td>Turn upper body forward</td>
</tr>
<tr>
<td>Lower the right leg</td>
<td>Turn the right leg backward</td>
</tr>
<tr>
<td>Return the body upright</td>
<td>Turn the upper body backward</td>
</tr>
</tbody>
</table>
SlideLeft and Demo
Correspondence of design to code

Skate:
Do Together
  move skater forward 2 meters
Do in order
  slide on left leg
  slide on right leg
Question

• Writing methods to make ice skater perform a skating motion – intricate task
• Would like to reuse these new methods in another world
• How can you make skate method available for an ice skater in a different world?
Answer: Save out as a new class

1) Rename iceSkater as cleverSkater

2) Save out as a new class. Alice saves the new class as CleverSkater.a2c
Inheritance

• The CleverSkater class
  – **inherits** all the properties and methods from the original IceSkater class
  – has newly defined methods (**skate**, **slideLeft**, **slideRight**)

• In other programming languages, the concept of creating a new class based on a previously defined class is called **inheritance**
Using CleverSkater

• An instance of the CleverSkater class can be added to a new world
Benefits of Inheritance

• Inheritance supports
  – Reuse of program code
    • Programmer writes code once
    • Use code later in different programs
  – Sharing code with others on a team project
Classwork today

- Create a new class
  - Inherit from another class that has 4 limbs
  - Create at least four new methods
  - One of the new methods should invoke one of the other new methods
- Repeat with another class
- Save out new classes and read into another world
- See handout for more details
Classwork today
create a new class example

crouching
crouching

walking
turn and smile