Announcements

- Read Chapter 5.1 for next time
- Assignment 5 out today
- Prof Rodger – extra office hours friday, no office hours next week!

Guidelines

- To avoid misuse of class level methods
  - Avoid references to other objects
  - Avoid calls to world-level methods
  - Play a sound only if the sound has been imported and saved out as part of the new class
- If these guidelines are not followed and an instance of the new class is added to another world
  - Alice will open an Error dialog box to tell you something is wrong

What we will do today

- Lecture on Chap 4 Sec 3 (continued)
- Classwork
Problem

• What if you were convinced you needed to write a class-level method where another object is involved?
• For example, a method for ice skater to skate around another object – here a penguin

Solution

• Class-level method with object parameter

  cleverSkater.skateAround

  Parameter: whichObject

  Do in order
  Do together
  cleverSkater turn to face whichObject
  cleverSkater lift right leg
  cleverSkater move to whichObject
  cleverSkater turn around whichObject
Translating Design into Code

- Most of skateAround storyboard easy to code
- Last two steps, require more thought
  - cleverSkater move to whichObject
    - What distance should cleverSkater move?
  - cleverSkater turn around whichObject
    - How do we tell cleverSkater to turn (in a circle) around another object?

Built-in Functions (or questions)

- The built-in function `distance to`
  - used to determine the distance the skater must move

Calling the function

- Code to move skater to `whichObject`

Oops, skater will collide with penguin!

Distance between two objects is measured center-to-center

Expressions

- To avoid collision
  - Use math operator to create an expression that adjusts the distance
- Math operators in Alice
  - addition +
  - subtraction -
  - multiplication *
  - division /
- Example:
How to put in an Expression

Result:

```
cleverSkater.move.forward() (cleverSkater.distanceTo WHICHOBJECT -> - 1)
```

More on AsSeenBy

- Use invisible object (isShowing set to false) to have objects fly around in a circle

```
cleverSkater.move.forward() (cleverSkater.distanceTo WHICHOBJECT -> - 1)
```
Testing

• Each time you create a new class, test it!
  – Add an instance of new class to new world
  – Write a short test program
    • Test each new method
• Testing increases your confidence in the ability to reuse your code in other worlds

Classwork today

• Modify your two classes from last time
  – Add methods with parameters
  – Use asSeenBy
  – Use isShowing
  – Use math
• Create a new world with 3 objects and demo
• See handout for more requirements