Day3: GreenFoot Morning
Penny

Key controlled movement, Relationships between actors  Platforms, Object detection
Recap

• Remember what you have learned:
  • Fields/Variables (constructors, scope, getters and setter)
  • Loops
  • Superclass and subclasses (super and this)
  • Methods (and parameters)
  • Randomizing
  • Conditional if statements
  • Movement and turns
  • Objects and Worlds

• Remember resources
  • Command list
  • Ctrl-space for list of methods
Primitive Types and Objects

• Primitive types
  • int
  • double
  • boolean

• Objects
  • String (defined by Java) “in double quotes”
  • Actor (defined by Greenfoot)
  • Bird (defined by you)
Variables

1. Declare variables
   - Declare variables as fields, or
   - Declare variables locally within methods, or
   - Declare variables as parameters

2. Instantiate objects
   - Call new to get memory
   - Call constructor to initialize object

3. Assignment statements
   - literal values
   - expressions
   - method calls
Greenfoot CoderGirl Class

- Fields/Variables (constructors, scope, getters and setter)
- Loops
- Superclass and subclasses (super and this)
- Methods (and parameters)
- Randomizing
- Conditional if statements
- Movement and turns
- Objects and Worlds
- Types
Break for VT Alum Interview
DEMO
Game

DEMO
First
Version
CliffWorld

- Open Greenfoot
- Open Penny Scenario
- Create a New Subclass of World: **CliffWorld**
  - Select **Stride**!
  - Use cloud background image
- Create 2 New Subclasses of Actor: **Mover** and **Platform**
Mover Class Constants

- Create constants for the mover for vertical acceleration and sideways speed.
  - Use command \texttt{v} to add a variable
  - Use the commands \texttt{ctrl-shift-n} and \texttt{ctrl-shift-s} to make it static and final
- Use command `/` to comment your constants
Mover Class Field Variable

- Create a field variable for vertical speed
- Write a setter method for vertical speed
- Change sideways speed constant name to sSpeed
Mover methods side to side

• Move right
• Mover left is symmetric
SubClasses

- Create a subclass of Mover: Penguin
- Create a subclass of Platform: Cliff
- Use the addObject method to add a new Cliff and Penguin to CliffWorld
Subclasses Check Keys

• Penguin method to check for left and right arrow key input
• call method from act()
• Run it
Mover methods

- Need to check if mover is still on the platform
- Write onGround method
- How to use it?
  - Experiment with having Penguin class call the onGround method
  - Experiment with penguin placement and constant value
  - Consider how the Penguin will fall
  - Add a getter for vSpeed
How did you use onGround()? 

- One possibility for checkFall() method. Remember to call it from act()
- How does it work?
- How can we adjust it?
- How can we improve it?
Mover fall method

• Add fall method to the Mover class
  • Remember how FlappyBird fell?
• Call it from the onGround() method of the Penguin class
  • Other Mover subclass may fall for different reasons
• What about hitting the bottom boundary?
Cloud

- What are the clouds behaviors?
- How do we make the cloud move?
- Consider methods and fields available to Cloud.
- Experiment and Run
Cloud Motion

- Does the cloud move back and forth between boundaries?
- \( \|\| \) is the OR operator (arabic keyboard?)
- How to deal with bouncing off boundaries?
Cloud Behavior

- What else does a cloud do?
  - It can catch penguins!
- Experiment and Run
Assignments

• Make your game match the demo
  • Make the Penguin jump when the space is pressed
    (hint: you can use the shift key to select and copy entire methods from previous projects)
  • Add another cliff

• Look for code that can be placed into reusable methods
  • atBottom(), gameEnd()

• Experiment with clouds and cliffs

• Add charms for Penguin to grab when jumps

• What about trying these things?
  • Different scenery and actors
  • Arrows accelerate and break
  • Keyboard to control clouds