Day 2: GreenFoot Afternoon

FlappyBird

Background, Falling Bird, User Input, Pipe Obstacles
New Scenario

- Open Greenfoot
- Open FlappyBird_TechGirls
Create World and Bird

- Create a New Subclass of World: FlappyBirdWorld
  - Select **Stride**
  - Use background image

- Create a New Subclass of Actor: Bird
  - Use Bird Image
Add Bird to World

- Edit FlappyBirdWorld
- Notice the dimensions of the world
- Add a Bird field variable (ctrl-v)
- Create a Bird Object in the constructor and add it to the World
- For the y position try using getHeight()/2 so it adjusts relative to the window height
Falling Bird

- **dy** is positive 2 because downward because coordinates
- This would be a constant rate of change, or constant velocity
- Inspect the Bird to see the how setLocation impacts the Y value.
Falling Bird

- Bird should fall according to the square of time
- After 1 sec: 1 ft
- 2 sec: 4 ft
- 3 sec: 9 ft
- Etc.
- Upside down parabola, speeding up and getting faster as it goes down
Falling Bird

- add a gravity variable $g$
- Update $dy$ by $g$ every act
- Run again... looks more realistic!
- Experiment with various values for $dy$ and $g$
KeyUpPresses

- Create a new method `checkForKeyPress()`
- Call the `checkForKeyPress()` from `act()`
- Run again!
- Experiment with values of `dy` and `g`
Some Tips

• An actor can call `getWorld()` and then can call methods on the world

• In the Greenfoot Help Menu reference:
  • “Greenfoot Class Documentation”
  • “Java Library Documentation”
Assignments – Match the Demo!

- Experiment with making gravity non-integer values, try .2 or .5. Change types and add type casts.
- Use 2 images for the bird, one for when it flaps up and one for when it is falling down.
- Use a different constructor for the world so that it can be unbounded (the bird can go beyond the upper and lower boundaries)
- When Flappy Bird reaches the bottom of the screen,
  - print “Game Over” to the console
  - then try printing the “Game Over” image to the screen
- Add pipes to fly over. The pipes will be a new class. The pipes should move from right to left on the screen. What happens when the bird hits a pipe?
- Automatically add new pipes at set intervals. Add them randomly?
- Extended:
  - Add pipes with spaces in between to fly through. Should the user be able to use the down arrow too?
  - Set angle of bird based on speed