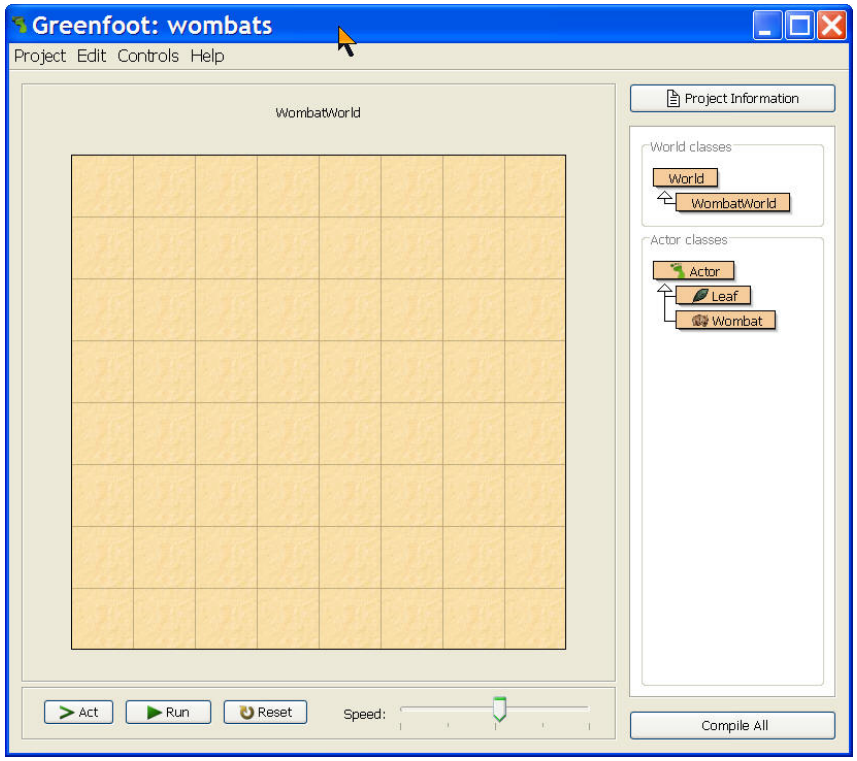
Wombats

Creating Games with Greenfoot

www.devoxx4kids.com

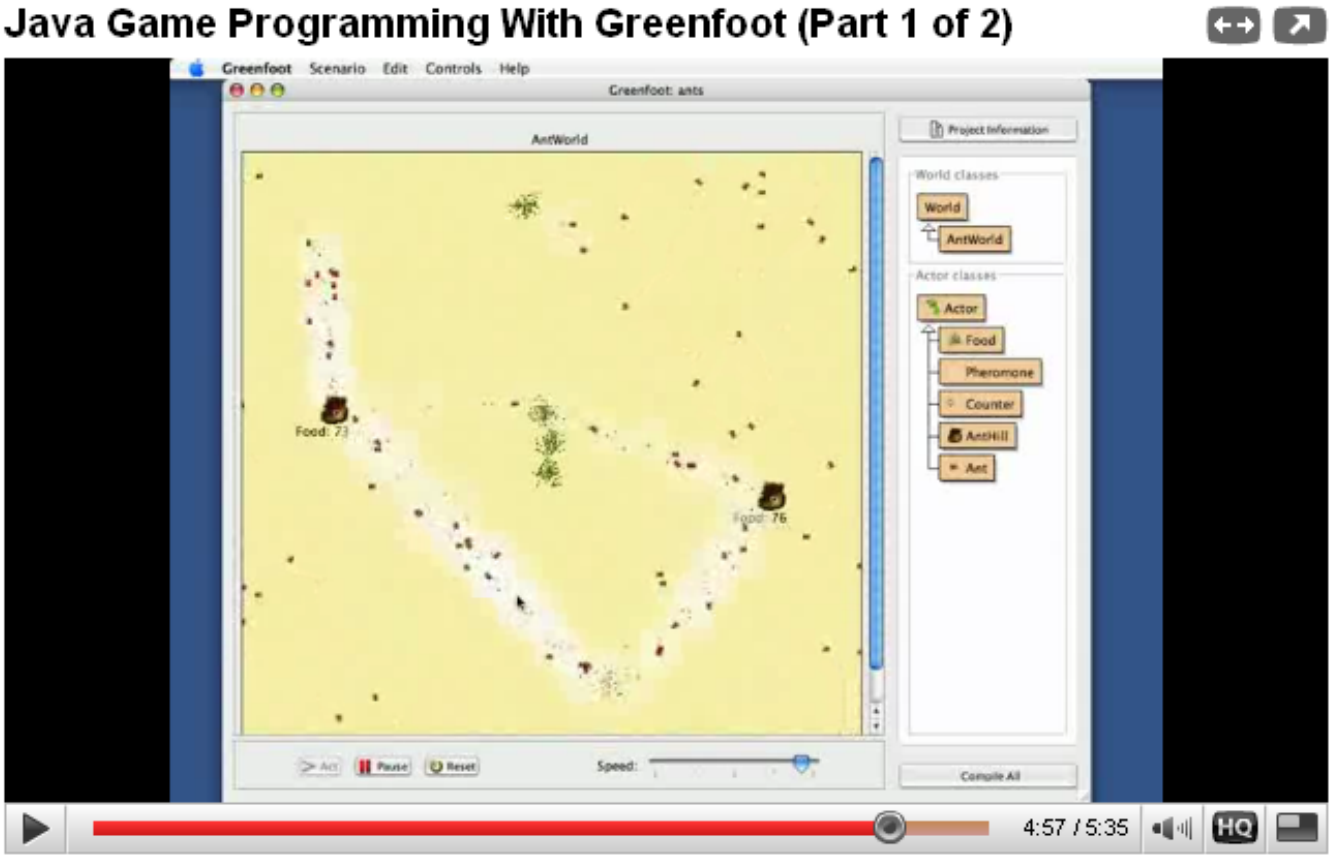
What is Greenfoot?



- A free environment that makes it easy to create 2D animations, simulations, and games
  - While teaching object-oriented concepts in Java
  - Built on top of BlueJ
  - Created at the Un. of Kent in England and Deakin University

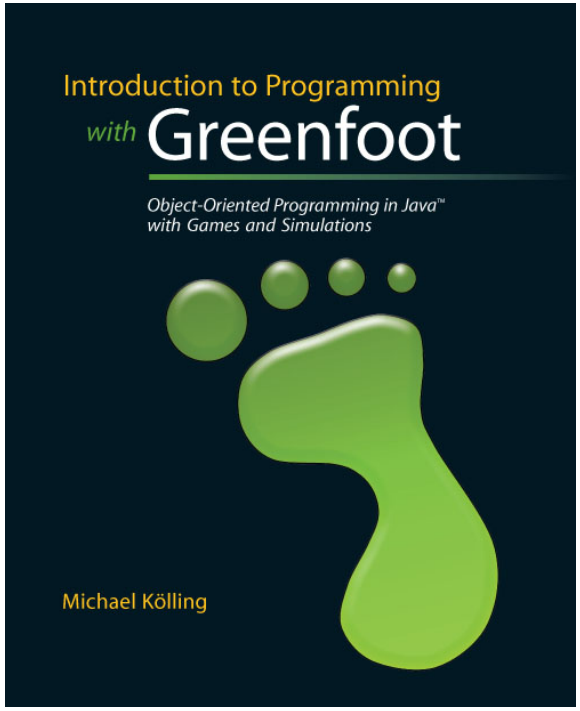
Watch – Java Programming with Greenfoot (Part I)

http://www.youtube.com/watch?v=NcGe141R2yA



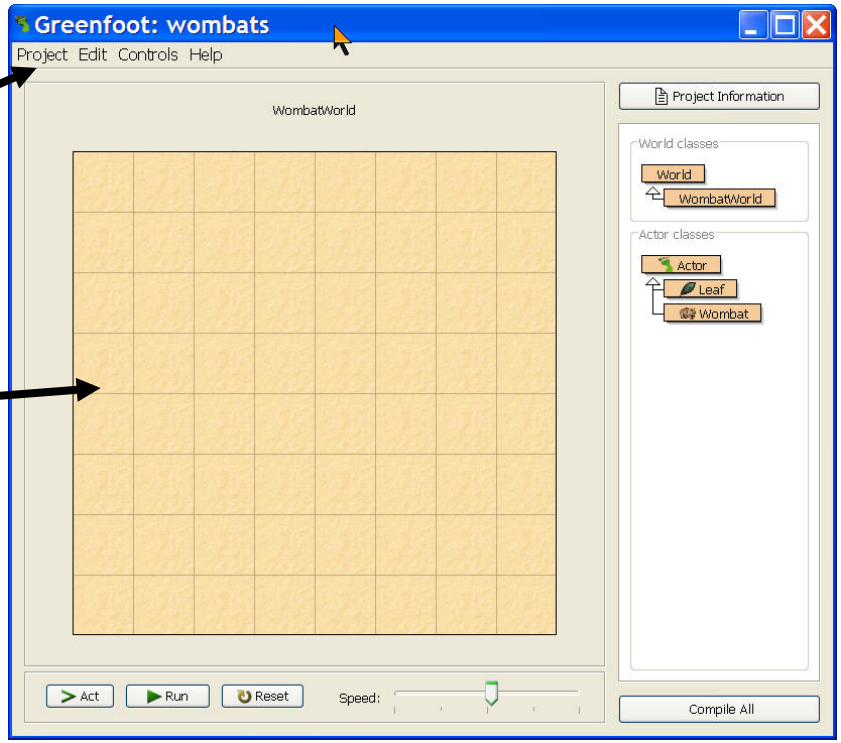
Greenfoot Resources

- Web site
  - http://www.greenfoot.org
- Scenarios
  - http://www.greenfoot.org/scenarios/index.html
- Tutorial
  - http://www.greenfoot.org/doc/tutorial.html
- Book: Introduction to Programming with Greenfoot by Michael Kölling



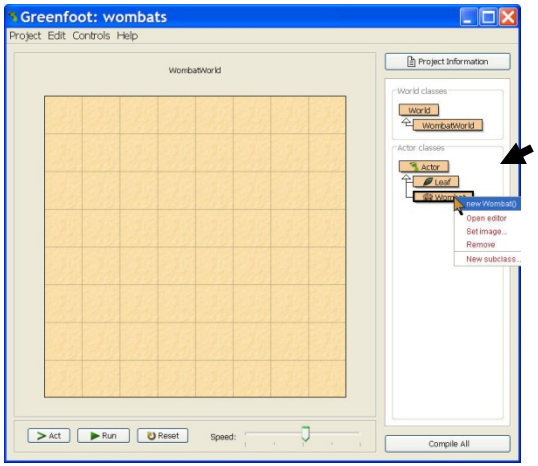
Getting Started

- To start Greenfoot click on greenfoot.exe
- Click on Project in the menu
  - Choose Open
  - Pick the wombats scenario
  - It starts by showing an empty WombatWorld
    - Which is a 2D grid of cells

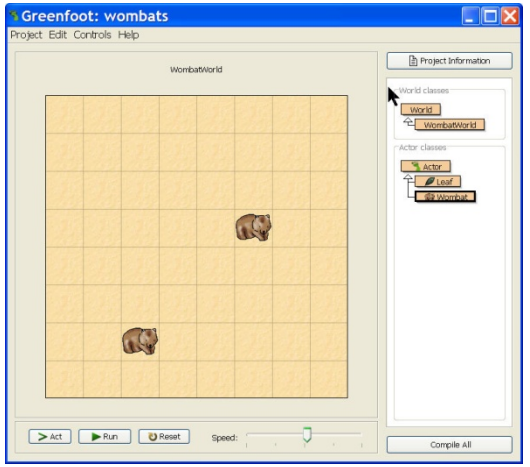


Adding Objects to the World

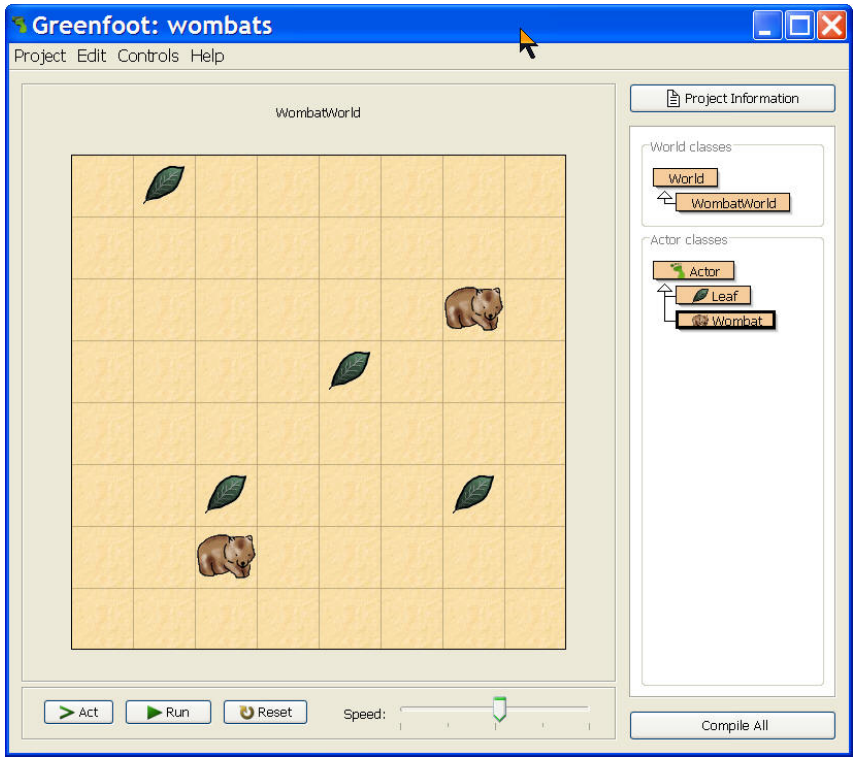
- Right click on Wombat in the class display to see a pop-up menu
  - Create a Wombat by selecting
    - new Wombat()
- A picture of a wombat will appear attached to the cursor
  - Click in the world to place the wombat
  - Shift-click to place more than one object



Class Display



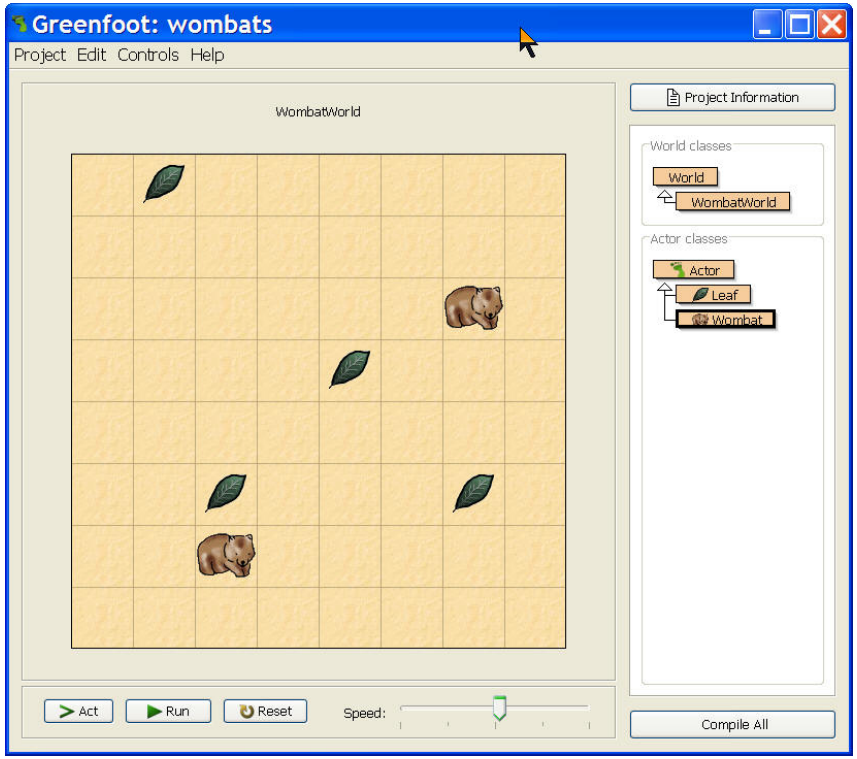
Add Leaves



- Add several leaves to the world
  - Right click on Leaf in the class display
    - Hold down the shift key
      - Click in the grid to place each leaf

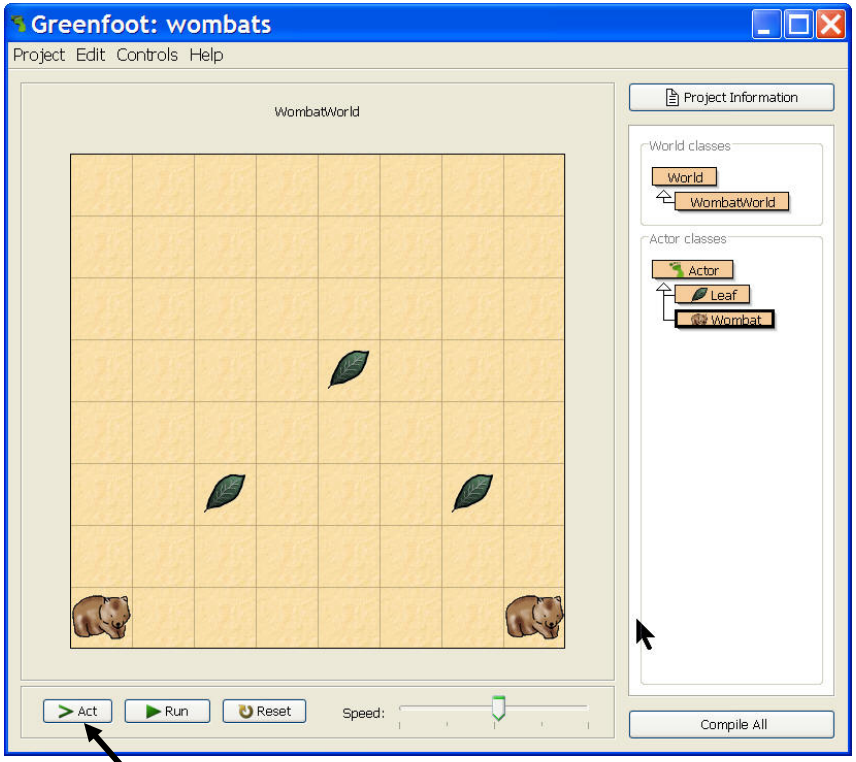
Greenfoot Objects

- There are two basic types of things in Greenfoot
  - Worlds and Actors
    - Other objects are children (subclassses) of these
      - Inherit properties and behaviors from parent (superclass)
  - Worlds hold actors
    - a stage for actors
  - Actors can act
    - They may move
    - They may stay where they are



Run the Simulation

- Click on act to execute one simulation time step
  - Act one time
- Click on run to have it continuously execute time steps
  - Keep acting till yo
    - Click on pause to stop
    - Click on reset to start over
      - With a new WombatWorld



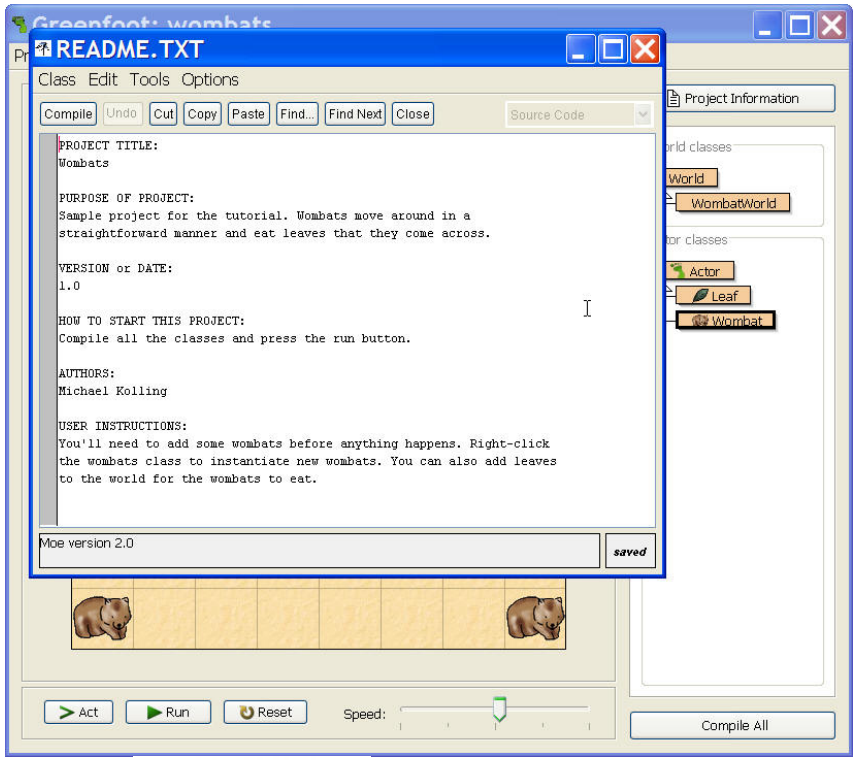
Execution Controls

What Happened?

- Questions
  - How did the wombats move?
  - Did they eat any leaves?
  - Did the leaves move?
  - What happened when a wombat reached the edge of the world?
- If you don't know the answer to any of these questions place more wombats and/or leaves in the world and run again
  - Or click on act to see things in slow motion

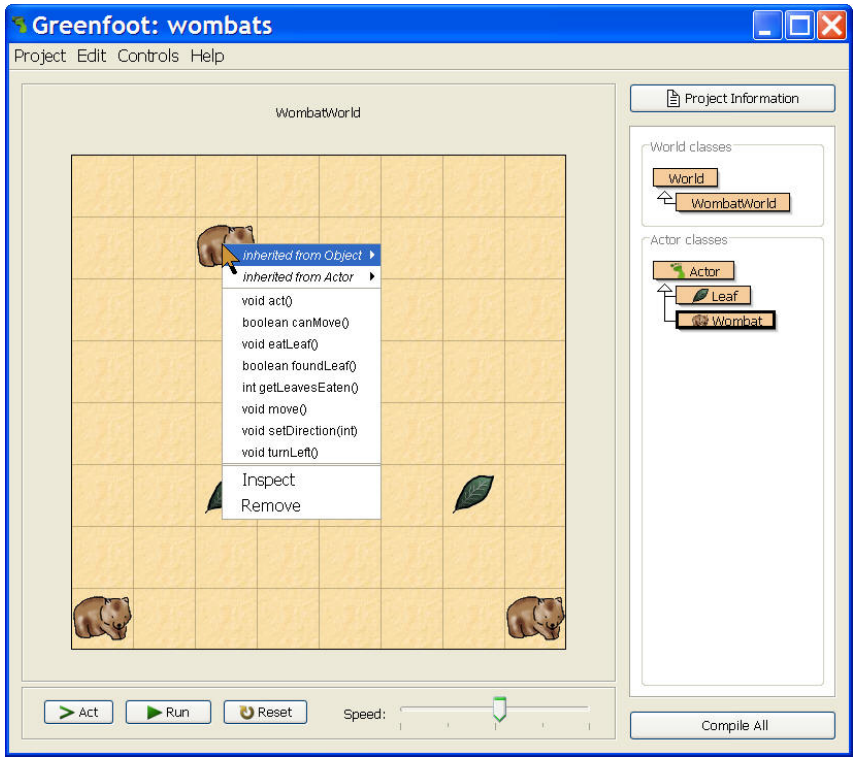
The Wombat Scenario

- To get information on any scenario
  - Click on the Project Information button
    - In top right corner above the class display
  - Read the displayed documentation
    - Click close when done

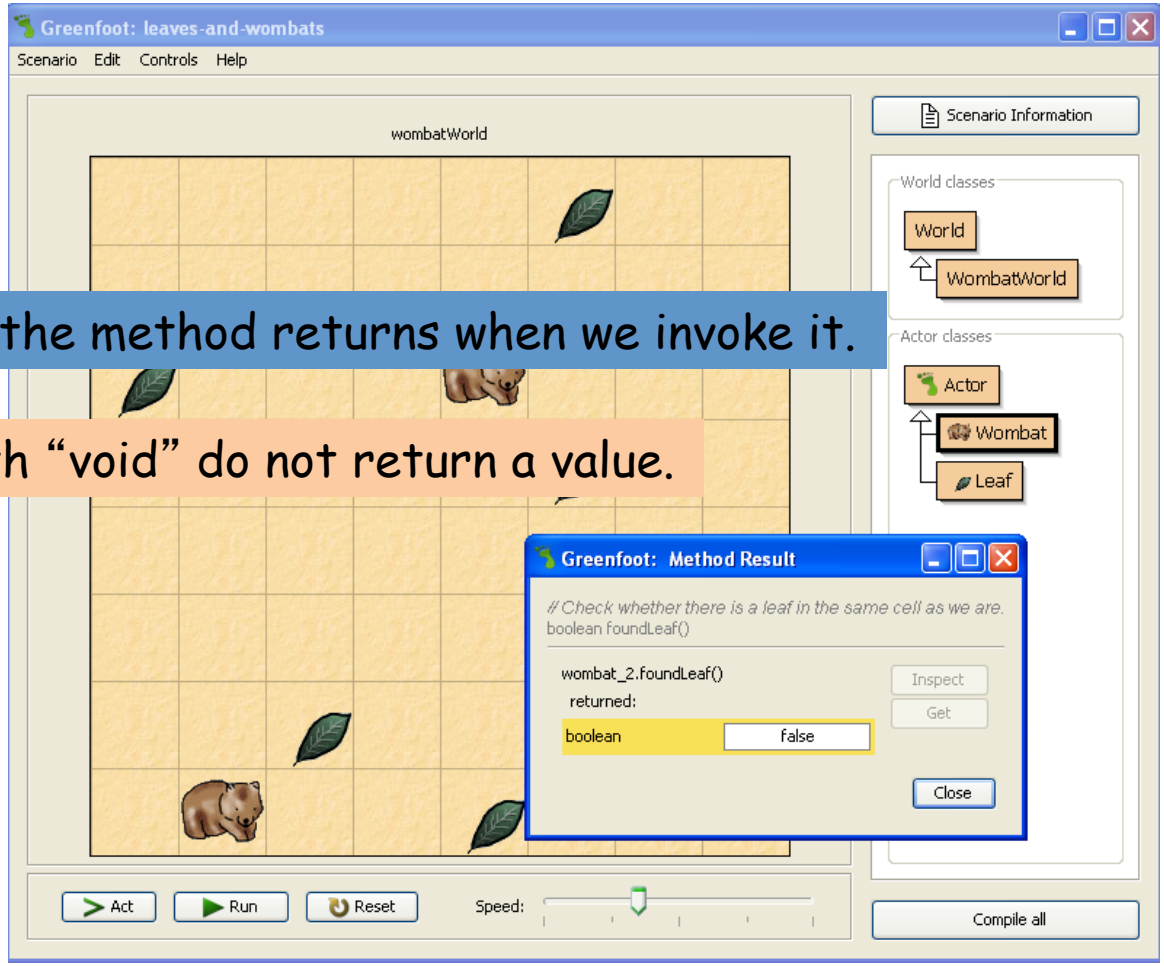


Actor Methods

- You can see and execute all methods that an actor knows how to do
  - By right clicking on an actor in the grid
  - Select a method to execute it
    - You can invoke inherited methods as well



Return Types

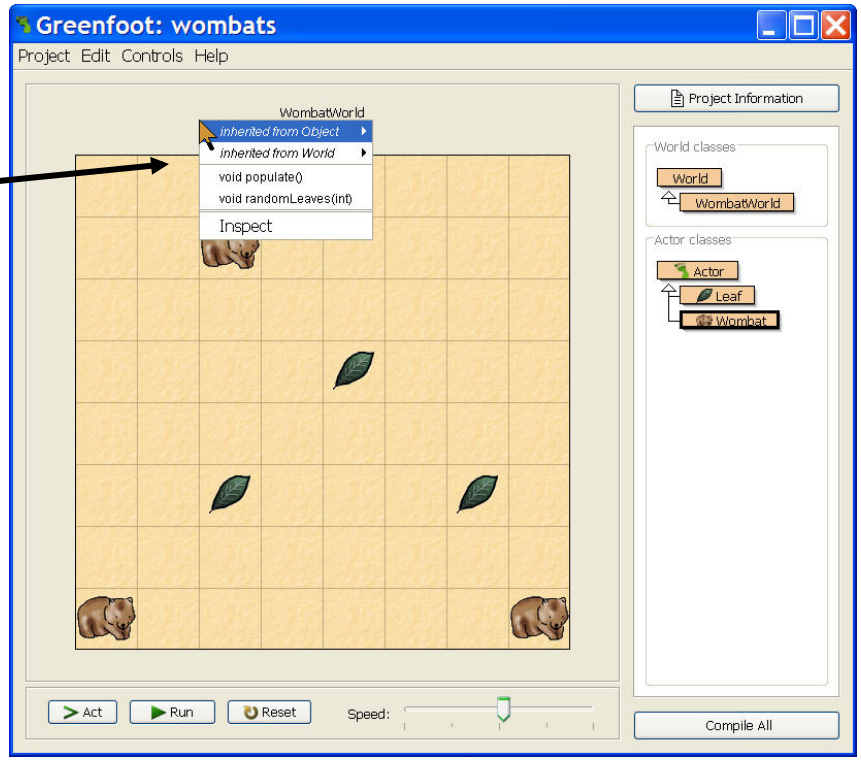


Tells us what the method returns when we invoke it.

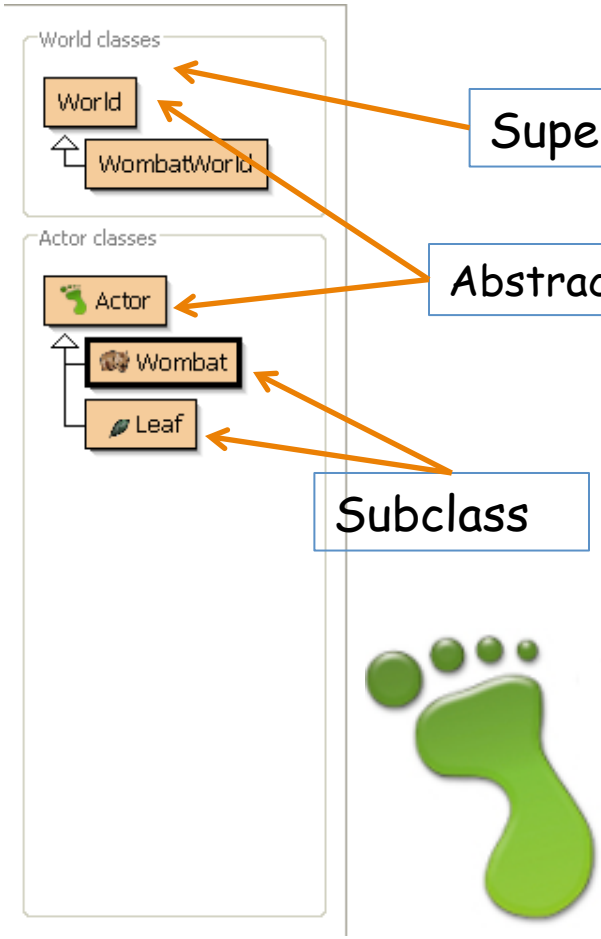
Methods with "void" do not return a value.

World Methods

- Right click near the title of the world
  - To show the world menu
  - Select populate()
    - This will add wombats and leaves to the world
  - Then click on run
    - What happens?
      - Do all the leaves get eaten?



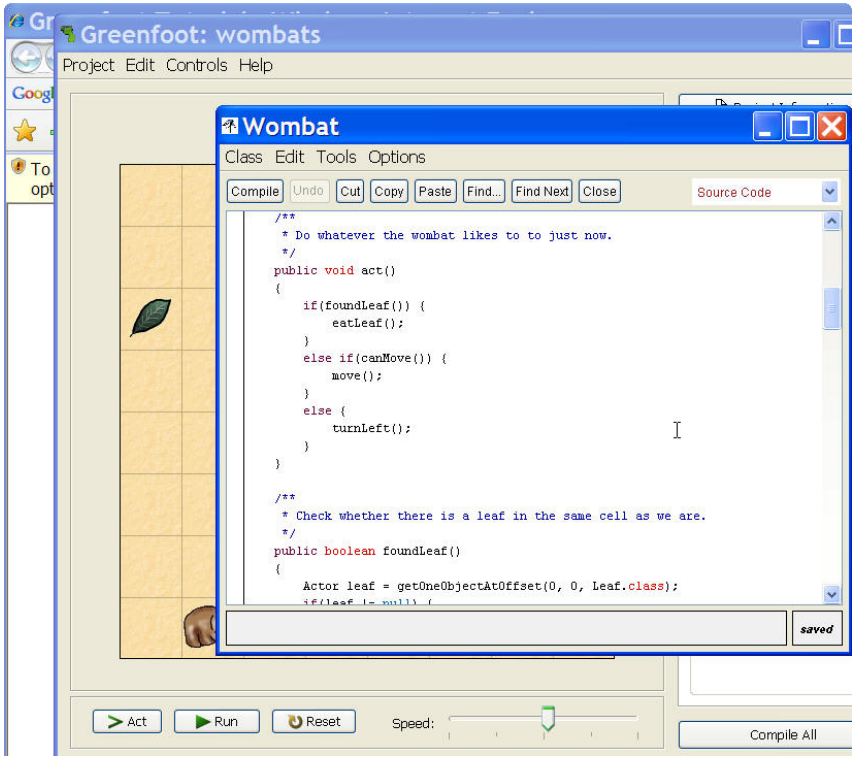
Class Diagrams



- Superclasses contain methods that give functionality to subclasses.
- Abstract Classes are needed but don't create actors.
- Subclasses inherit the methods of its superclass.

Changing Wombat Behavior

- Left click on the Wombat class
  - To show the code for the Wombat class
  - Read the act method
    - Modify it to turn left a random number of times (0-3) if it can't move and isn't eating a leaf
      - Create a turnRandom method and call it instead
      - Use Greenfoot.getRandomNumber(4)



Attack the Problem

- Establish a turnRandom() method
- Establish a variable "turns" that will hold the number of turns.
- Establish a loop to count the number of turns
- Establish a counter, set it to zero, count up by 1

Possible Solution Part I

```
    /**
     * Turn in a random direction.
     */
    public void turnRandom()
    {
        // get a random number between 0 and 3...
        int turns = Greenfoot.getRandomNumber(4);

        // ...an turn left that many times.
        for(int i=0; i<turns; i++) {
            turnLeft();
        }
    }
```

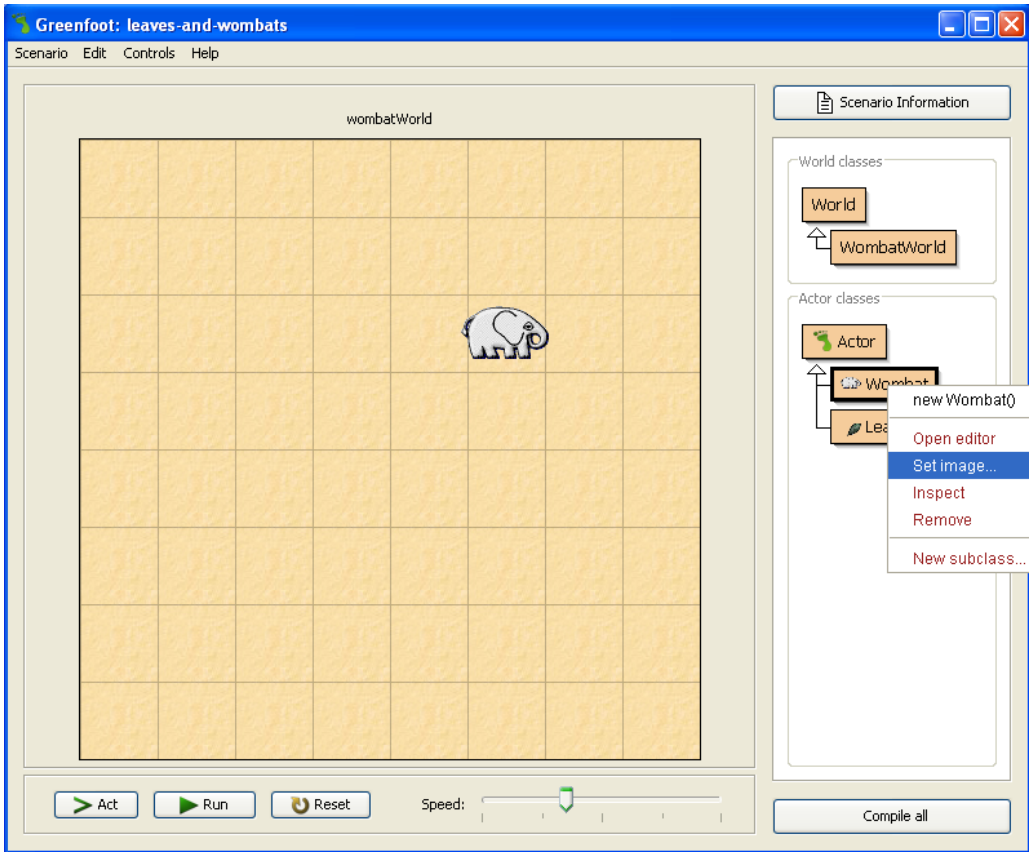
Attack the Problem Part II

- Use a conditional statement to say that if the wombat is pointed in a specific direction, then change that direction randomly

Possible Solution Part II

```
/**
 * Do whatever the wombat likes to to just now.
 */
public void act()
{
    if(foundLeaf()) {
        eatLeaf();
    }
    else if(canMove()) {
       if (Greenfoot.getRandomNumber(2) == 1)
       turnRandom();
       else
       move();
    }
    else {
        turnRandom();
    }
}
```

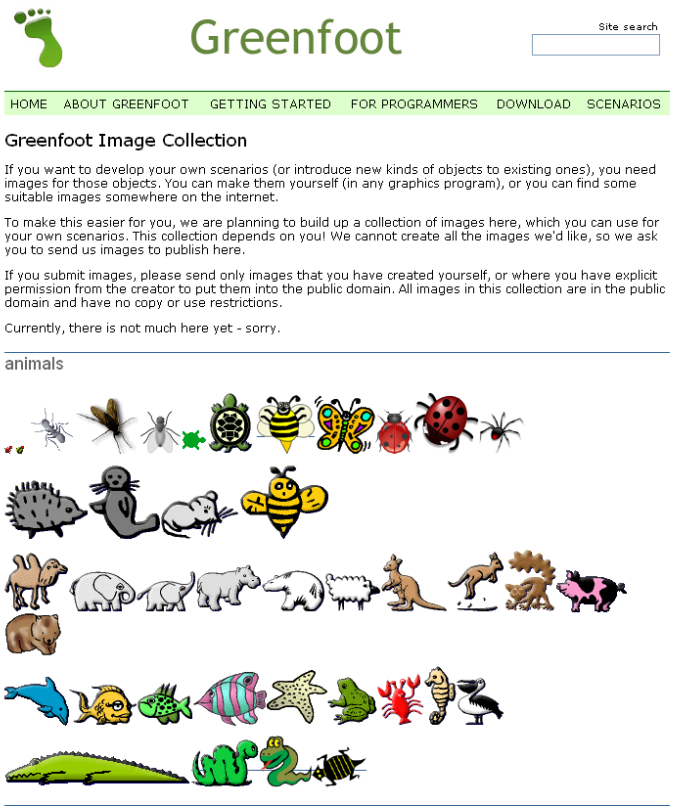
Changing Actors



Changing Actors

1. Right click on the actor's class
2. Choose "Set Image"
3. Choose the image from the library
4. Compile the program

Adding Actors from the Greenfoot Image Library



Adding Actors

1. Navigate to the website.
2. Right click and download the image to your folder on the school's network drive.
3. Save it to your images folder for the project.
4. Open the project.
5. Right click on the actor's class.
6. Choose "Set Image".
7. The image should appear in the left pane because you added it to the project image folder.
8. Choose the image.
9. Compile the program.

http://www.greenfoot.org/download/images.html

Open Challenge
(Coding Movement)

- Study the habits of how a particular character moves in real life.
- How could you code these different types of movements?
  Look at movement in other games or check out how to create movement in video tutorials.

http://www.greenfoot.org/doc/videos.html

Videos are also on YouTube.com