

The logo for Oracle Academy. The word "ORACLE" is in a bold, orange, sans-serif font. Below it, the word "Academy" is in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

Academy

Java Fundamentals

3-1

Getting Started with Greenfoot

ORACLE
Academy



Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Objectives

- This lesson covers the following objectives:
 - Download and install Greenfoot
 - Describe the components of the Greenfoot interactive development environment
 - Describe classes and subclasses
 - Create an instance of a class
 - Recognize Java syntax used to correctly create a subclass



Download and Install Greenfoot

- For instructions on downloading and installing Greenfoot, please refer to Section 0: Java Curriculum Software Requirements of this course
- Select the Greenfoot version most suited to your needs and download it to your local machine



ORACLE
Academy

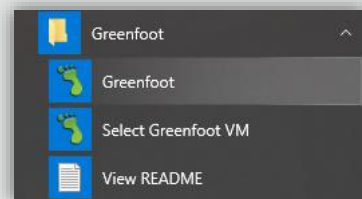
JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

4

Launch Greenfoot

- To launch Greenfoot:
 - Double-click the Greenfoot icon on your desktop
 - Select the Greenfoot program from your list of computer programs



<http://www.greenfoot.org/> is a great resource for Greenfoot. Here you will find lots of examples with most having the full source code. You will also find lots of discussions and you can post your own questions. You can also download Greenfoot here.

Greenfoot Textbook Scenarios

- To become familiar with Greenfoot, download and run the scenarios created by the authors of the Greenfoot textbook

A scenario is a game or simulation implemented in Greenfoot



- Instructions for working with existing scenarios are given on the following slides

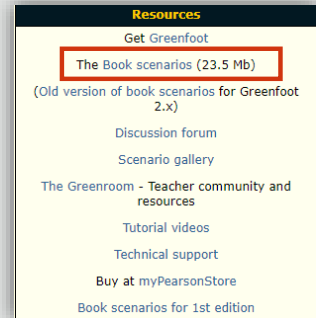
The scenarios start out slowly then begin to show some really good techniques to create great looking interactive games.

Steps to Download the Greenfoot Textbook Scenarios

- Go to the Greenfoot textbook webpage:

<http://www.greenfoot.org/book>

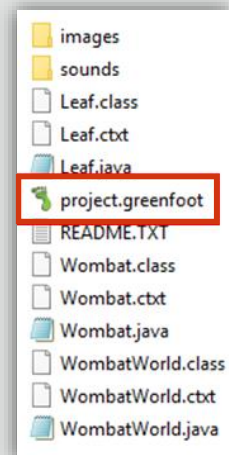
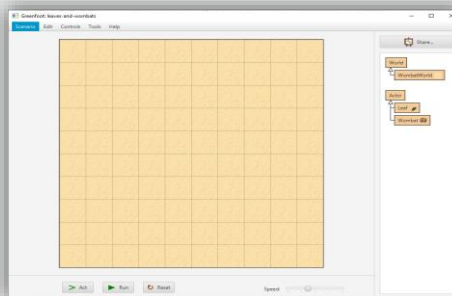
- Click the Book Scenarios link
- Save the zip file to a folder on your computer
- Extract the zip files to a folder on your computer
- Name this folder "Greenfoot Scenarios"
- If your computer does not have zip file extraction software, download free, open source software at 7zip.com



If you have not seen Greenfoot or Java before then even the basic scenarios may seem quite complicated. What you will soon find out is that you can develop interactive programs very quickly in Greenfoot, even if you have never done programming before.

Steps to Open a Scenario in Greenfoot

- From the Scenario menu, select Open
- From the Greenfoot scenarios folder you created on your computer, select the leaves-and-wombats scenario from the chapter01 folder (Answer “**Yes**”, if asked to update code)
- The scenario will open in a new window



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

8

The Greenfoot Development Environment does not have many controls or options. Rather than being restrictive, it helps the initial learning process.

Execution Controls

- Execution controls to run a scenario include:
 - Act:
 - Runs all actions in the scenario once
 - Run/Pause:
 - Runs all actions in the scenario repeatedly until Pause is clicked
 - Reset:
 - Resets the scenario back to its starting position
 - Speed:
 - Runs actions faster or slower



The Act method is useful when you are testing your program as it allows you to have all of your actors perform their code only once. The speed can also help slow down your world so that you can better see what is happening.

Defining Class Characteristics

A class contains the specifications that define the appearance and movement of an object. The class provides instructions to Greenfoot for how to create and display instances when they are added to your scenario.



- In nature, a bee has characteristics that are common to all bees: six legs and two wings
- The bee may also inherit the characteristics of its breed that gives it a specific color, shape and size
- In Greenfoot we would create a Bee class that defines these characteristics
- This defines how all bee objects included in the scenario look and act

In Java it is common practice when declaring a class to start it with a capital letter. This leads to more readable code where other users can deduce you are referring to a class when the word starts with a capital letter. So above we say Bee class rather than bee class.

Classes in Greenfoot

- The class tells your scenario how its objects should look and act when the scenario is run
- When you add a class to your scenario, it appears in the class hierarchy (to the right of the world)
- You can add as many instances of the class as you wish to the scenario

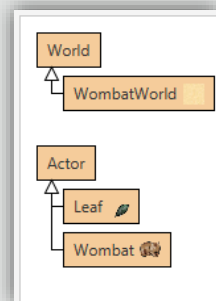


The naming convention for Classes in Java means that the first letter should be capitalized. This is a convention, not a rule and as such it will not be reported as an error.

Superclass Types

- Two types of superclasses appear in the Greenfoot class hierarchy:
- World:
 - Holds the subclasses that provide the background image for the scenario's world
 - Defines the size and resolution of the world
- Actor:
 - Holds the subclasses that produce the instances that act in the scenario

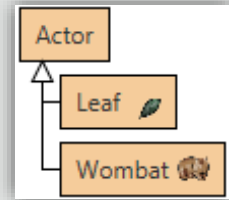
The overarching class of a group of classes is called a superclass. In this example - World and Actor



You will learn throughout this course that the idea of superclasses and subclasses is a very important concept in Object Orientated Programming (OOP).

Subclasses

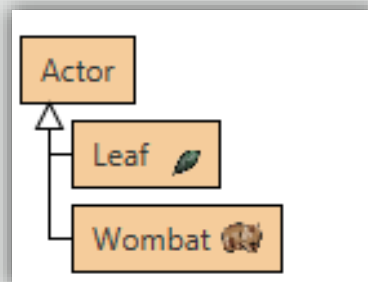
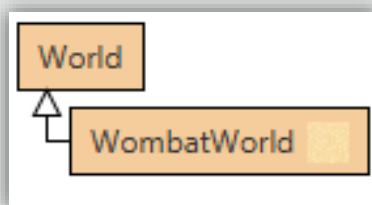
- Subclasses are a specialization of a class
- For example, the Wombat class is a subclass of the Actor superclass
- This subclass:
 - Inherits all of the properties of the Actor superclass, such as a pre-defined set of actions that Actor subclasses can perform
 - Has properties specific to its subclass, such as the image that gives Wombat objects their appearance
 - Can receive new properties that the programmer creates specifically for the subclass, such as images or actions



The idea of inheriting a superclass's behavior is a powerful advantage of programming in OOP. By creating your own subclass of the default Actor class you inherit all of the properties and actions that the Greenfoot developers have created. We will start to see what these are shortly.

Subclass Properties


- A subclass has an "is-a" relationship to a superclass (Wombat is a subclass of the Actor superclass)
- Properties can be modified (such as the class's name, image to display, or actions to perform)
- An arrow in the class hierarchy shows the subclass's relationship to the superclass

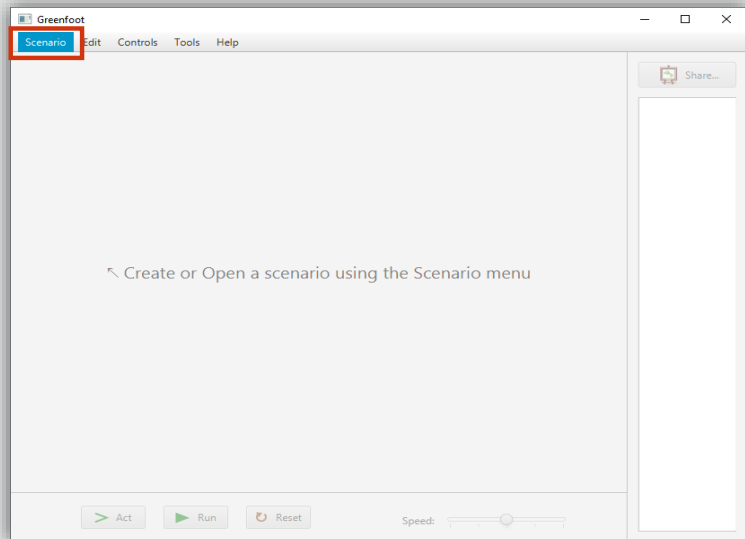


We can continue to add subclasses. We could add BabyWombat underneath Wombat. We would then add the differences between a Baby Wombat and a Wombat to that class.

Can you think of other class – subclass relationships?

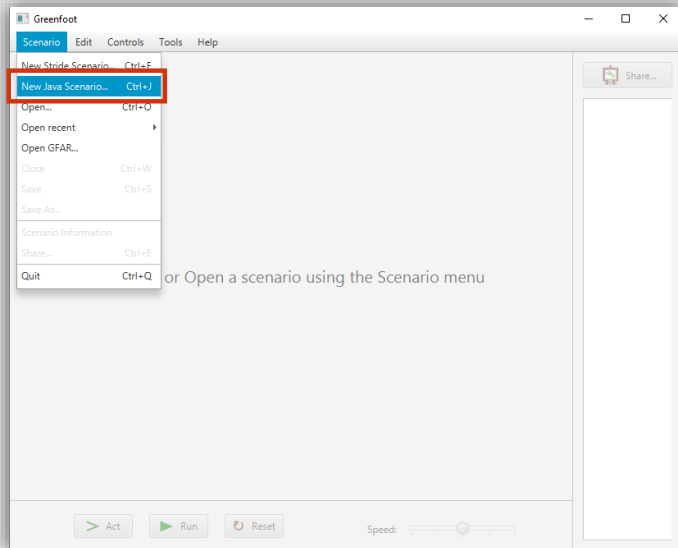
Try It – Create a new scenario

- Open Greenfoot 
- Select the Scenario Menu



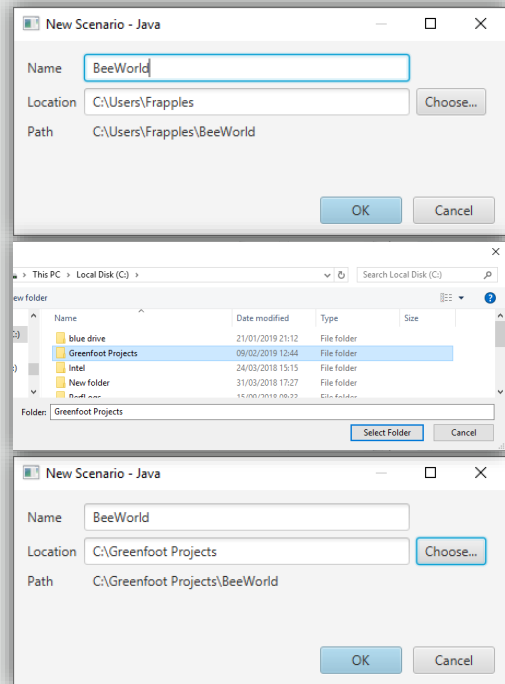
Try It – Create a new scenario

- Select New Java Scenario...



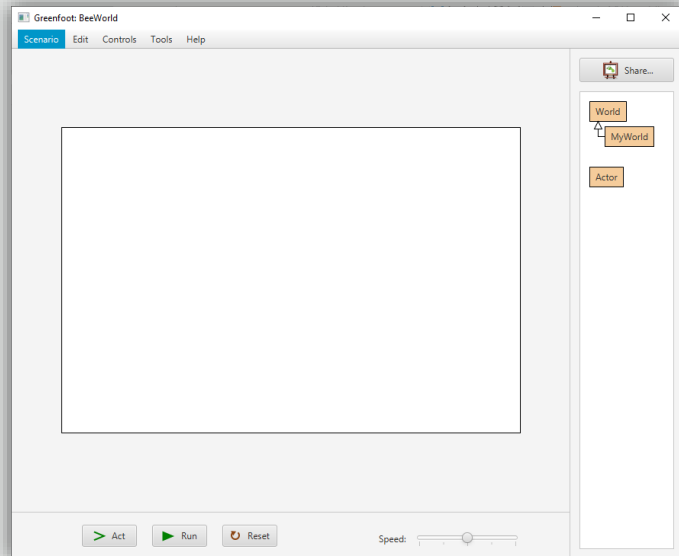
Try It – Create a new scenario

- Name the scenario
“BeeWorld”
- Click Choose... to navigate to where you plan to save your Greenfoot files
- Click “OK”



Try it – Create a new scenario

- Your scenario should look like this:



ORACLE
Academy

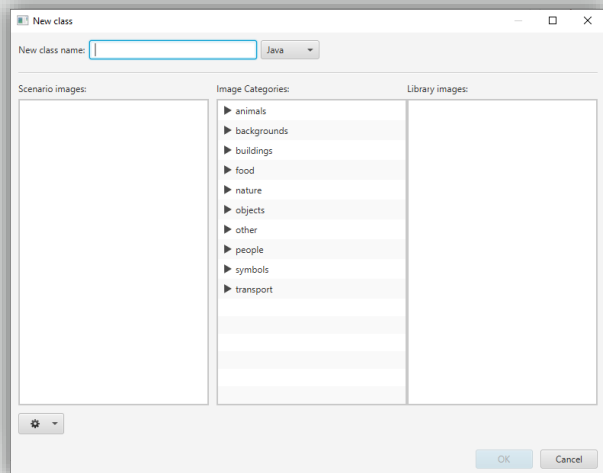
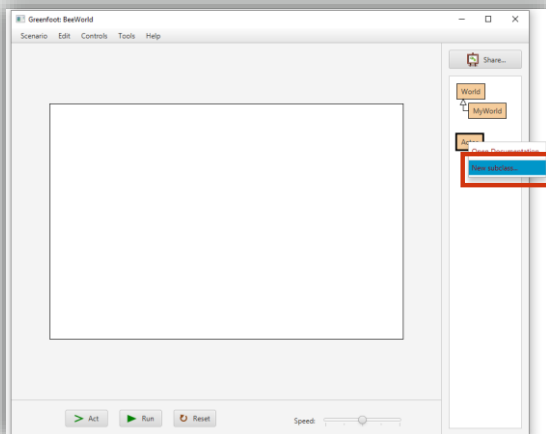
JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

18

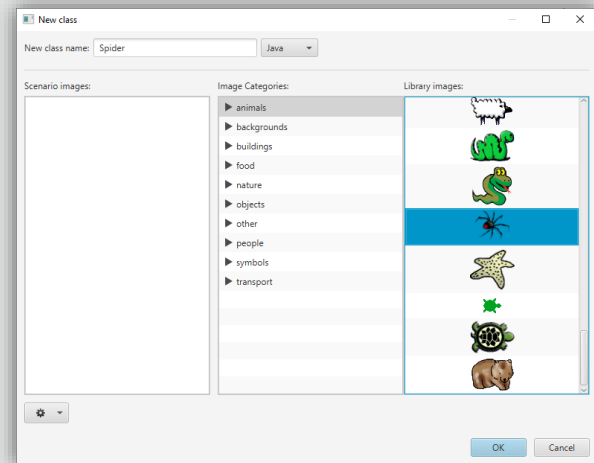
Try it – Create a new scenario

- Next, you will add a Spider, a Bee, and a Fly
- Right+click on the Actor class, and select New subclass



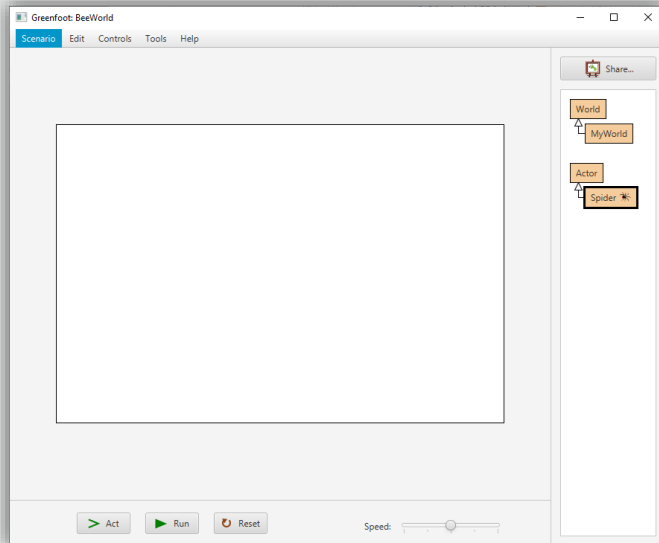
Try it – Create a new scenario

- Type “Spider” as the New class name:
 - Select the “animals” Image Category
 - Select “spider.png” from Library Images list
 - Click “Ok”



Try it – Create a new scenario

- Your results should look like this:



ORACLE
Academy

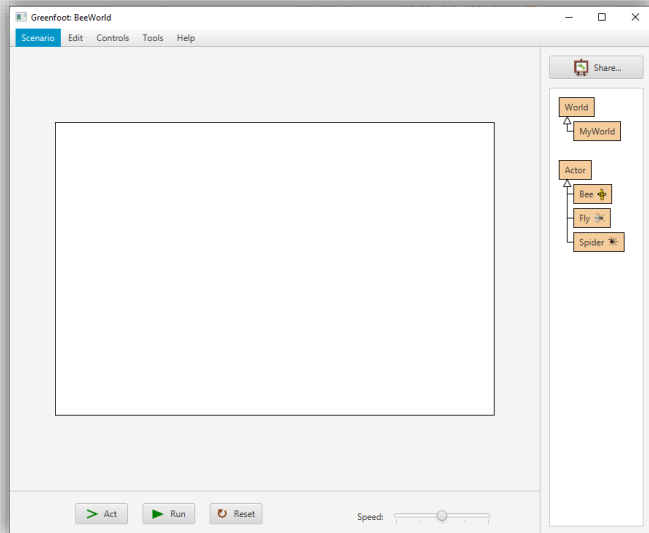
JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

21

Try it – Create a new scenario

- Now add a Bee and a Fly
- Your results should look like this:



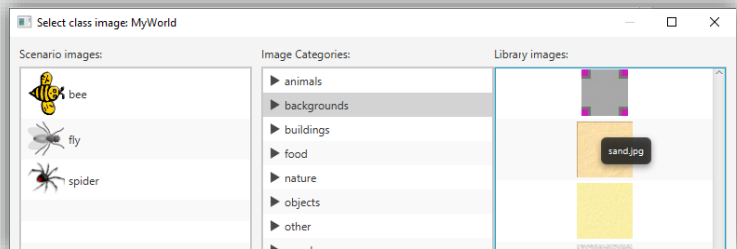
Try it – Create a new scenario

- Final step – Add a background to your scenario
- Right+click on MyWorld, and choose “Set Image”



- From the “backgrounds” Image Categories, select sand.jpg

- Click “Ok”

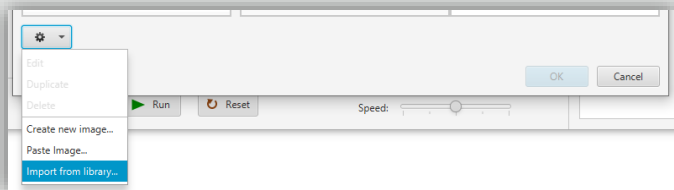


Steps to Create a New Subclass with an Image Imported From Your Computer

- Right click on the World or Actor superclass
- Select New Subclass...



- In the New Class window, select the “Gear” icon, and choose Import From library...

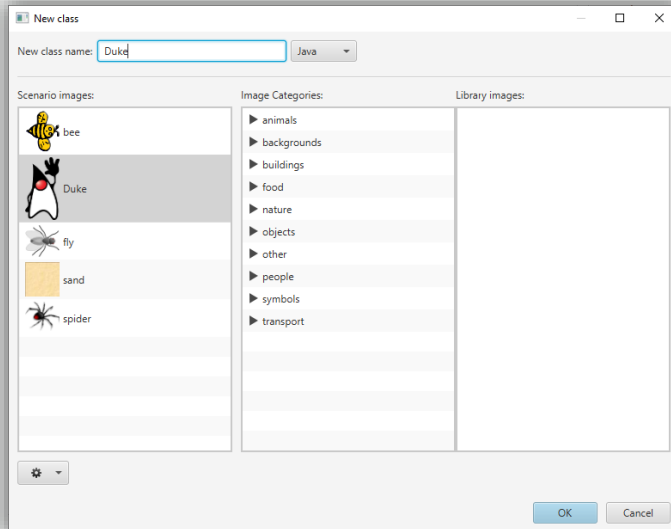
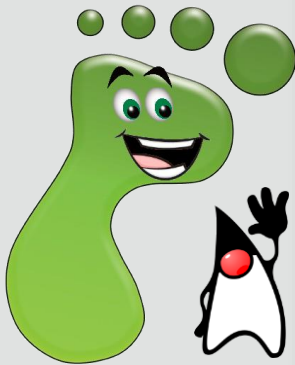


- Navigate to an image on your computer, name it, then click “Ok”

This can be an image that is created by you in a “Paint” program, a photo, or an icon or image that you have saved on your computer.

Steps to Create a New Subclass with an Image Imported From Your Computer

- In this example, we have imported an image of the Java Icon, called “Duke”



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

25

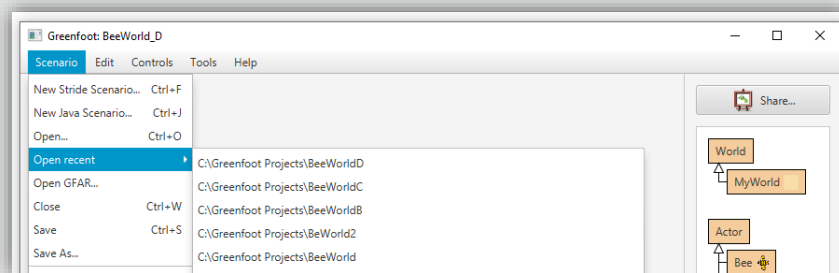
When you import a file it will be copied to the Scenario's Image folder. You can edit it within this folder at any point, although you may have to right click on the class and reselect the image.

Images help bring games to life. Simply by changing your graphics a game can take on a whole new life. It is worth learning about how to make and work with transparent images. This allows parts of the image to be transparent (normally the part between the image and the edge of the image). If we look at the bee image then everything surrounding it is transparent. If we look at the screen shot on the slide in the top bee image we see the grey background, and in the middle left image we see the white background. This is because there is a transparent edge to the bee image.

View the “How to Create Transparent Images” document, located in Section 3, Lesson 1 Student Resources.

Save Multiple Versions of Scenarios

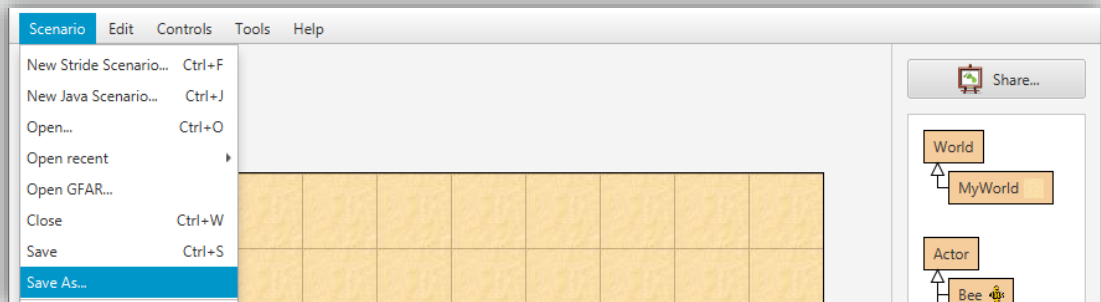
- Save the scenario frequently as you work
- Each time you close Greenfoot, it saves your current work
- Save multiple versions of scenarios:
 - To return to an earlier version of a scenario
 - To have multiple scenarios to work from



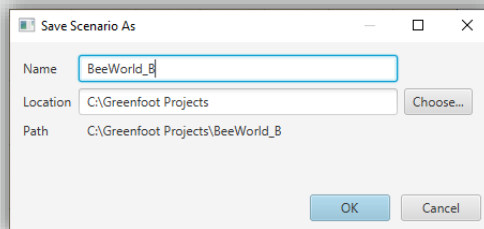
Greenfoot will save your program as you go along, but each time it overwrites the previous version. It is worth having more than one copy of your code at different stages so you can always go back to a previous working version.

Steps to Save a Scenario

- In the Scenario menu, select Save As...



- Save a copy to a folder on your computer



The Save option overwrites the previous version. The Save As option creates a new copy.

Instances of a Class

- The Bee class defines the characteristics of a Bee
- Such as: movement, color, size etc
- A Bee that flies in a field, or rests on a flower, is a physical object that is a unique instance of the Bee class
- An instance object holds its own unique set of characteristics as defined in the class, but can be manipulated and changed



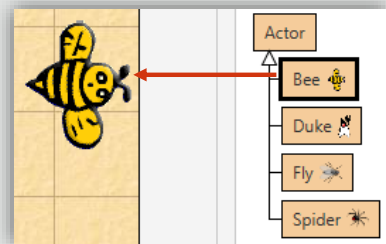
Think of a class as a blueprint or plan. So a Class that represents a house is not an actual house, just the plans for that house. If we build a house to this plan then this becomes an instance of the House Class. i.e. it becomes a House. We can have many instances of a class.

Greenfoot Instances

- One or many instances of a class can be added to the scenario
 - Actor instances move and act in your scenario
 - World instances provide the background for your scenario
- Instances can perform the behaviors written by the programmer in the class's source code



Instances are the objects from a class that act in your scenario



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

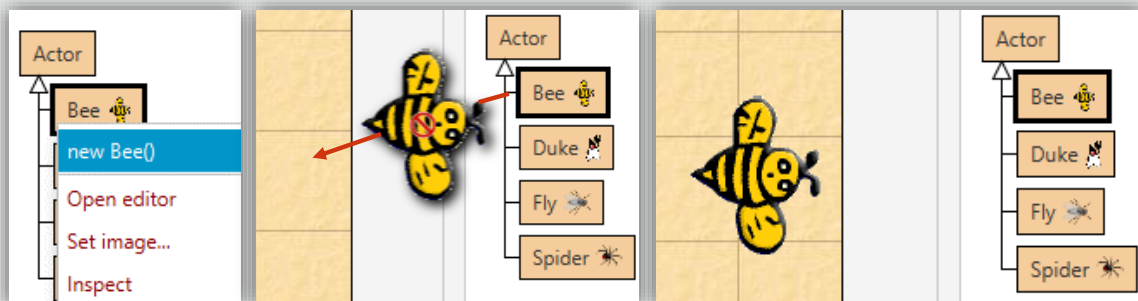
Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

29

Instances of the same class are autonomous. Once created they have their own values which can make each instance of the class's behavior a bit different.

Steps to Add an Instance to a Scenario

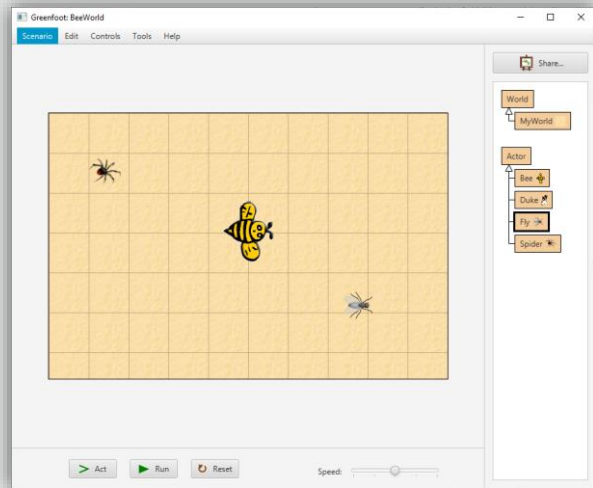
- Right click on the class
- Click the new [class name] option
- Drag the instance into the scenario with your cursor
- Later, you will program the instance to act by writing source code in the class's Code editor



Multiple instances of a class can be added to a scenario by holding down the SHIFT key and clicking the world multiple times.

Steps to Add an Instance to a Scenario

- Right+click and add one instance of the Bee, Fly, and Spider
- Place them in your scenario so that it looks similar to this:



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

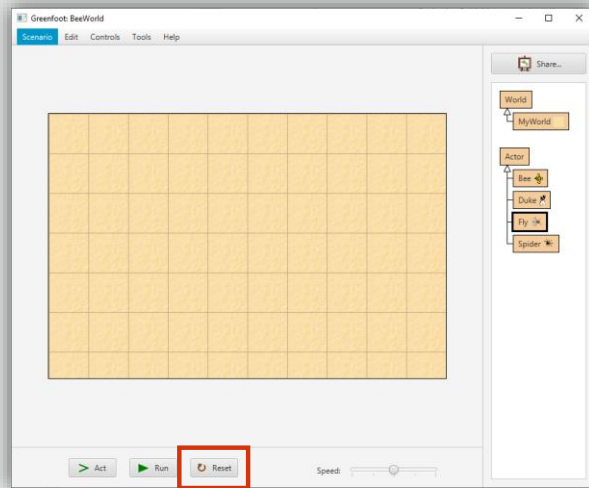
Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

31

Multiple instances of a class can be added to a scenario by holding down the SHIFT key and clicking the world multiple times.

Steps to Add an Instance to a Scenario

- Now, click the “Reset” button
- What happened???



ORACLE
Academy

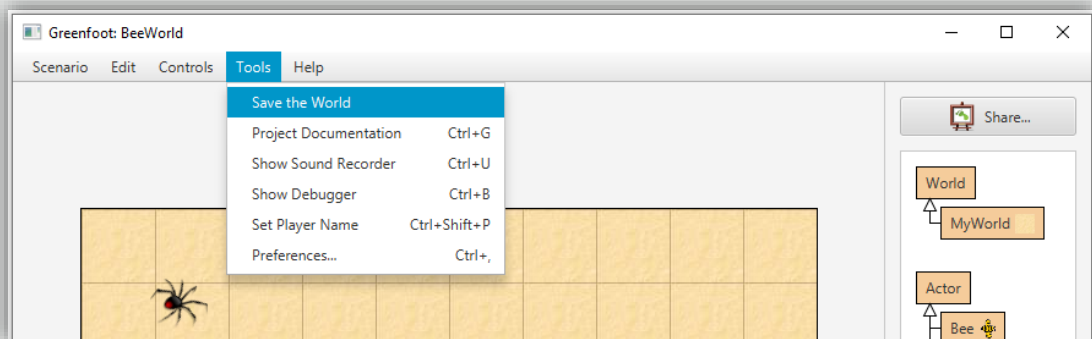
JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

32

Steps to Add an Instance to a Scenario

- Adding Instances from the Scenario Actor list is only temporary!
- To add them so that they remain in the Scenario:
 - Right+click and add one instance of the Bee, Fly, and Spider
 - Click the “Tools” menu, and select “Save the World”



Steps to Add an Instance to a Scenario

- A new screen will open, showing the Java code for the World:



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class MyWorld here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MyWorld extends World
{
    /**
     * Constructor for objects of class MyWorld.
     *
     */
    public MyWorld()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(600, 400, 1);
        prepare();
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
        Bee bee = new Bee();
        addObject(bee, 298, 184);
        Fly fly = new Fly();
        addObject(fly, 466, 289);
        Spider spider = new Spider();
        addObject(spider, 79, 79);
        fly.setLocation(462, 287);
    }
}
```

Class compiled - no syntax errors

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

34

Steps to Add an Instance to a Scenario

- Later in the course, you will learn more about this, but for now, Close the Code Editor



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class MyWorld here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MyWorld extends World
{
    /**
     * Constructor for objects of class MyWorld.
     *
     */
    public MyWorld()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(600, 400, 1);
        prepare();
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
        Bee bee = new Bee();
        addObject(bee, 298, 184);
        Fly fly = new Fly();
        addObject(fly, 466, 289);
        Spider spider = new Spider();
        addObject(spider, 79, 79);
        fly.setLocation(462, 287);
    }
}
```

Class compiled - no syntax errors

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

35

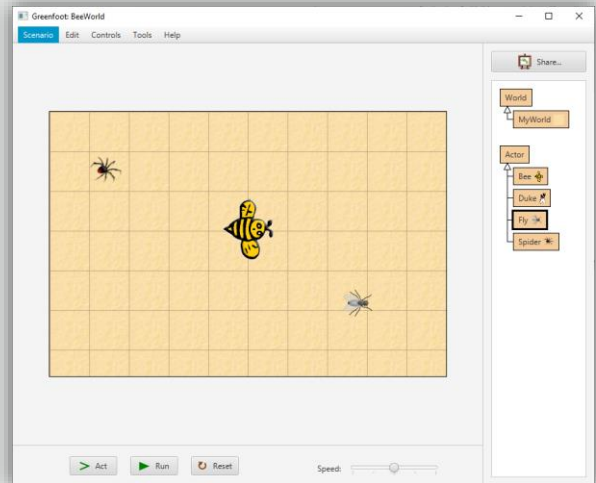
Steps to Add an Instance to a Scenario

- Now, click the “Reset” button again
- What happened???



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot



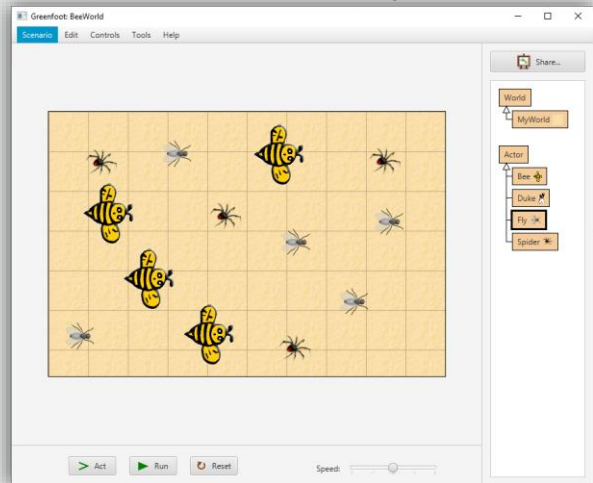
Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

36

The Java code added by Greenfoot has kept the instances in the World!

Steps to Add an Instance to a Scenario

- You can still move the Instances around or add new Instances
- Just click “Save the World” each time to keep the layout



ORACLE
Academy

JF 3-1
Getting Started with Greenfoot

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

37

The Java code added by Greenfoot has kept the instances in the World!

Source Code

- DNA gives humans certain characteristics, such as appearance, mobility, and communication
- Like DNA, source code is written to tell the class how its instances could act in the scenario

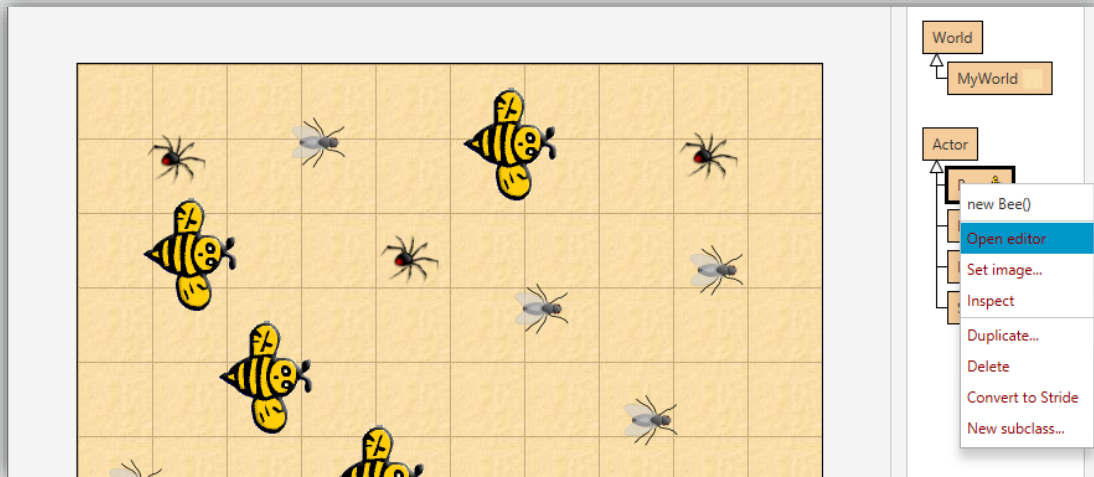
Source code defines what all instances of each class are capable of doing. The behavior of each instance is determined by the source code of its class.



When we create a subclass we will see that Greenfoot provides the outline of the class in code. This greatly speeds up building software and removes a very repetitive task.

Steps to View a Class's Source Code

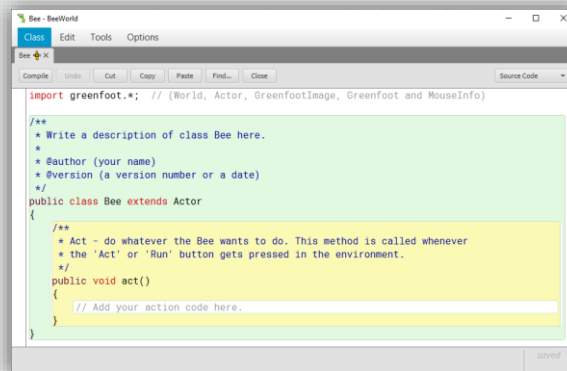
- Right click on a class in the class menu
- Select Open Editor



Note: Sometimes when people write programs they use multiple monitors or higher resolutions than you can support. Greenfoot remembers where a window was when it was last opened, so this can sometimes open up outside of your visible screen. You will have to bring this back into the visible screen using your operating system. This varies depending on what operating system you use. If you are going to have other people read your code it is worth remembering to open this up on the main window so as not to cause the problem above.

Code Editor

- The Code editor displays the class's source code
- This is where instructions are programmed for how instances of the class can act in the scenario
- In the following lessons, you will learn how to edit the Java code for your Greenfoot projects



Notice the line `public class Bee extends Actor`. This is saying that Bee is a subclass of Actor in Java.

Also notice that there are different background colors. This helps us identify different sections of the code.

Terminology

- Key terms used in this lesson included:
 - Class
 - Instance
 - Source code
 - Subclass
 - Superclass

Summary

- In this lesson, you should have learned how to:
 - Download and install Greenfoot
 - Describe the components of the Greenfoot interactive development environment
 - Describe classes and subclasses
 - Create an instance of a class
 - Recognize Java syntax used to correctly create a subclass



