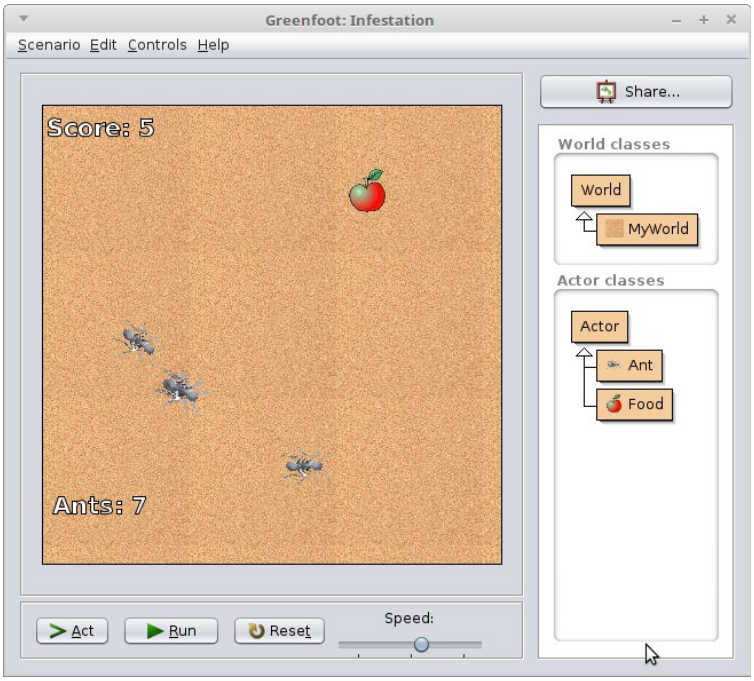
CS 134 In-class Challenge



Introduction
This is a challenge assignment that you will work on on your own, or with a teammate, to figure out as much as you can on your own.

Afterward, we will go through some of these challenges in class.

Testing
After each challenge, test the program and look for anything not working right.

Turn-In
You don't need to turn anything in; we will be working in class to write the final solution.
You might want to save your work to a flash drive or to your email for future reference.

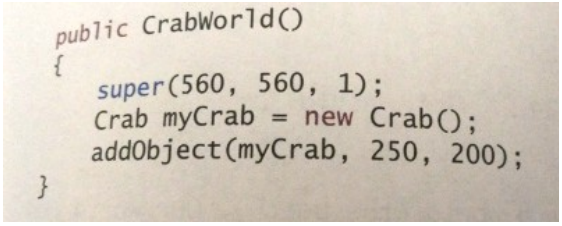
Challenge 1. (Chapter 3.2)
Create an "Ant" and a "Food" subclass of Actor.

Challenge 2. (Chapter 4.1, 4.2)
In the MyWorld class code, you're going to create two variables – an Ant variable and a Food variable.

```
public class MyWorld extends World
{
    Ant ant;
    Food food;
```

Make sure you create the Ant and Food variables as **class member variables**. This means that the declarations go inside the class, but outside of all functions.

Then, within the constructor method, initialize your ant and food variables as **new** objects (see page 54 of the textbook):



```
public CrabWorld()
{
    super(560, 560, 1);
    Crab myCrab = new Crab();
    addObject(myCrab, 250, 200);
}
```

Add the Food and one Ant to the game world at any x,y position. The function you will use is:

`void` **addObject**(Actor object, int x, int y)
Add an Actor to the world.

Challenge 3.
Open the Ant's code. In the act() method, add code so that the ant will always turn towards the mouse. Use this method:

`void` **turnTowards**(int x, int y)
Turn this actor to face towards a certain location.

You can get the mouse's coordinates with this block of code (do this before the turnTowards()):

```
MouseInfo mouse = Greenfoot.getMouseInfo();
int mx, my;
if(mouse!=null){
     mx = mouse.getX();
     my = mouse.getY();
}
```

Challenge 4.
Within the MyWorld class, create a new method.

- Return type: void
- Method name: moveFood
- Parameters: none

Within this method, move the food variable by using the method:

```
void  setLocation(int x, int y)
      Assign a new location for this actor.
```

Use the Greenfoot random number generator to choose a new X, Y coordinate, using the method:

```
static int  getRandomNumber(int limit)
            Return a random number between 0 (inclusive) and
            limit (exclusive).
```

Challenge 5. (Chapter 3.3)
Within the Ant's act() method, check to see whether the Ant is touching the Food with this method:

```
                  isTouching(java.lang.Class<?> cls)
protected boolean Checks whether this actor is touching any other objects of the given
                  class.
```

Use an if statement and check if it is touching. If it IS touching, then call MyWorld's moveFood function. First you will have to get the world:

```
MyWorld world = (MyWorld)getWorld();
```

Then you can use any of the custom functions you wrote:

```
World.moveFood();
```

Challenge 6. (Chapter 4.1)
Within the MyWorld class, create a new method.

- Return type: void
- Method name: addNewAnt
- Parameters: none

Within the method, you will add a new Ant object to the world. However, this time, you do not need to make an Ant variable like in challenge 2. Add a new ant to the world with the method:

```
void addObject(Actor object, int x, int y)
```
Add an Actor to the world.

And choose random numbers for the new X, Y coordinates with:

```
static int getRandomNumber(int limit)
```
Return a random number between 0 (inclusive) and limit (exclusive).

Like this:

```
addObject(new Ant(), x, y);
```

(except instead of x, y you will have your random numbers)

Challenge 7.
Extending the isTouching functionality from Challenge 5, make sure to also call the new addNewAnt method after the moveFood method:

```
World.addNewAnt();
```

Challenge 8.
Your program might crash if there are too many ants on the screen, causing a lot of ants to get the food, which causes a lot of ants to spawn. In your addNewAnt method, you will want to add:

```
if ( getObjects( Ant.class ).size() < 20 )
{
    // Original code here
}
```

With your original addObject code inside.