Scenario: You will need to kill all of the zombies before they eat your brains!!

1. Create the following Actors classes (all as a subclass of Actor) and set the images you want to use from the supplied scenario images:
   - **Hero**
   - **Blob** (please use cwlogo.gif)
   - **Zombie**

   (**Note:** Make sure you use a capital as your Actor name!)

2. Set the image for the MyWorld class by right clicking on the class and selecting "Set image" (for now use one of the background assets supplied)

3. Compile your scenario and place your Hero in the middle of the stage by right clicking on the class and selecting newHero(). Once you are happy with the placement, right click on the stage and choose "Save the World". This will now compile with your hero added to the stage.

4. Open the editor of the Hero class, we are going to set key presses for left, right and space bar with the following code (add this within the public void act() function because is an action that the Actor does):

```
//Set left key press function
if (Greenfoot.isKeyDown("left"))
{
  setRotation (getRotation()  - 2);
}

 // Set right key press function
 if (Greenfoot.isKeyDown("right"))
{
setRotation (getRotation() + 2);
}
```

(**Note**: The number is the speed at which your hero rotates.)

5. Now we need to call the function fire() when we press the spacebar. Below that code (still within the act() function add the following:

```
//Call fire
if (Greenfoot.isKeyDown("space"))
{
 fire();
}
```

6. It won't compile yet because we need to create the fire() method, to do this use the code below (outside of the act() this time because it's a new function):

```
private void fire()
{
Blob blob = new Blob(); //Name of actor, notice the capital
getWorld().addObject(blob, getX(), getY());  //sets blog at same X Y coordinates as cannon
}
```

Compile your code run the game and see what you get!
(**Hint:** CTRL + Space is a quick way to see hints for code completion)

7. You should see a blob appear on your character - we need to make the blob move using a custom method with the **Mover** class.

   To do this, create a new subclass of Actor with the name **Mover** (no image). Open this up in the editor and delete all the code it comes with...we're replacing it with a bit more functionality from Greenfoot's website that we need. The code you need is in the **mover.txt file** (this is in zombie folder). Paste the code into your class Mover and press compile.

8. Open the **Blob** class editor and add the move function for your ammo by using the following code inside the act () function (again its part of the class's actions):

   ```
   move(10.0); //The distance should be a double meaning it requires a decimal place
   ```

   This won't compile yet because the **Blob** class won't be able to see the **Mover** class, we need to change the class to extend from Mover instead, so change `public class Blob extends Actor` to `public class Blob extends Mover`, now compile, run and see what happens...

9. There are lots of blobs being created because the programme is working too quickly on the isKeyDown space method. We need to change it by going into the **Hero** class editor again and overwriting our space bar method with this:

   ```
   //Fire
   if ("space".equals(Greenfoot.getKey()))
   {
   fire();
   }
   ```

   When you run your game you should now just have the one blob every time you hit space bar.

10. We need to now make our blobs move with the direction of our character. To do this paste the following code into the fire() function in the **Hero** class editor.

    ```
    blob.setRotation(getRotation()); //gets the rotation of the hero
    ```

11. Let's create our zombies! Open up the editor for the zombie class and paste the following in the act() function:

    ```
    move(1);//This is the zombie speed

    if (Greenfoot.getRandomNumber(100) < 10)
    {
     turn(Greenfoot.getRandomNumber(90) - 45 );
    }
    ```

    Now compile this code and add a few zombie actors to your World (same as before right click and choose newZombie())– add as many as you want then **Save the world** again. Run and see what happens!
    (**Note:** When adding zombies to your stage you can add multiple by holding down shift.)

12. To stop them hitting the walls we need to add the following code in the Zombie class below the last lot of code but within the act() still:

```
if (getX() <=5 || getX() >= getWorld().getWidth() -5)
{
 turn(180);
}

if (getY() <=5 || getY() >= getWorld().getHeight() -5)
 {
  turn(180);
}
```

Compile and run! They shouldn't be turning as soon as they hit the walls now.

13. Now we need to set up the zombie eating function. Open up the editor on your Zombie class again and add the following code outside of the act() because it's a new function.

```
public void eatHero()
{
 Actor hero;
hero = getOneObjectAtOffset(0, 0, Hero.class);
 if (hero != null)
   {
  World world;
  world = getWorld();
  world.removeObject(hero);
   Greenfoot.playSound("eating.wav");//This is a soundclip that plays when the zombies get you
  }

}
```

We also need to call this function in the act () code so do this by adding the following line at the end but inside of the void act() function:

```
eatHero();
```

Your zombies should be able to "eat" your Hero!

14. Right, let's set up the hero's defence... make the zombie killing function on the **Blob** class by using the following code (this is a new function so remember not to put this inside the act())

```
public void killZombie()
{
 Actor zombie;
 zombie = getOneObjectAtOffset(0, 0, Zombie.class);
 if (zombie != null)
  {
  World world;
  world = getWorld();
  world.removeObject(zombie);
  Greenfoot.playSound("eating.wav"); //This is a soundclip that plays when we kill our zombie
```

```
  }
}
```

Don't forget we also now need to call this function from within the act() function too so use the following code to call the new zombie killing function:

```
killZombie();
```

Make sure your game compiles then move onto the next step.

15. Now we're nearly done – we need to end the game when all the zombies are gone or if your hero gets eaten. Open up the MyWorld class editor and paste the following code:

```
private boolean isGameOver;
private boolean isGameRunning;

public void act()
{
  if ( ! isGameOver ) {
    int numEnemies = getObjects( Zombie.class ).size();
    int numHero = getObjects( Hero.class ).size();

    if ( !isGameRunning && numEnemies > 0 ) {
      isGameRunning = true;
    } else if (( isGameRunning && numEnemies == 0 ) || ( isGameRunning && numHero == 0 )) {
      isGameOver = true;
      onEndGame();
    }
  }
  }

public void onEndGame()
{
System.out.println( "Game over!" ); Greenfoot.stop();  //Displays Game over message and
stops the game
}
```

16. Compile your game and get shooting. You can make it harder / easier by adding more zombies and affecting the speed of the Blob or the Zombies etc. using the slider on the game or changing the numbers in the game. Once you've made your changes to speed / your world make sure you save your world and your scenario so that it looks the same next time you open it.

If you've got time...

1. If you've got time you could try adding your own sound effects – to do this record a short soundclip using Windows sound recorder and save it as a **.wav** file and save it in the folder **zombie\sounds**. Then all you have to do is go into the code of the Zombie or Hero and replace the file eating.wav with yourfilename.wav.
   (**Note**: It must only be a .wav to work!)

2. Try making your own backgrounds – just go into Paint and open up the blank.gif file (this is stored under **zombie\images.** When you've finished save the background and update your MyWorld class with the new image (do this by right clicking and choosing Set image).