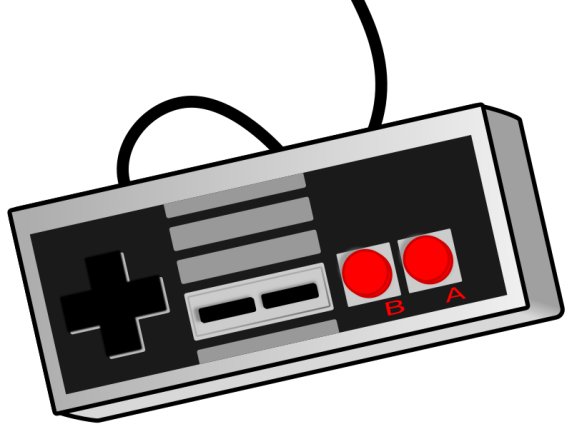


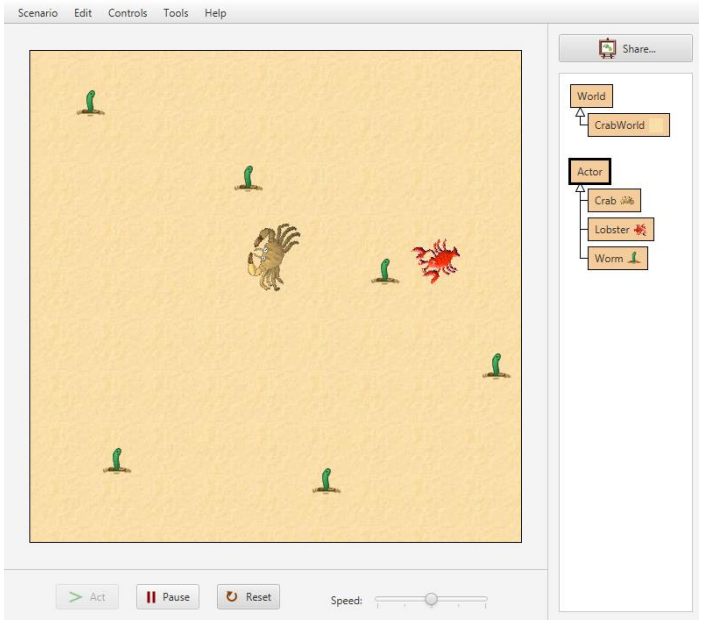
Lab_4

Little Crab (Part 3)

Outline

Adaptation of the Crab Tutorial from Greenfoot.org by using Stride instead of Java (Part 3)

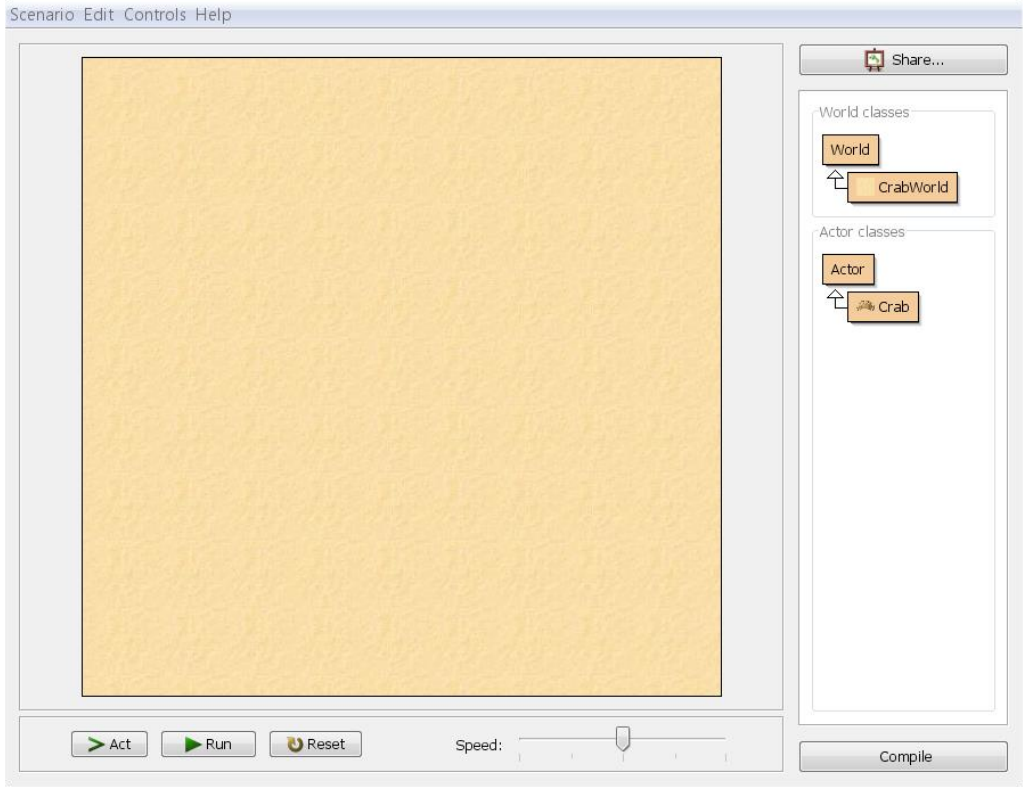
- Creating new World
- Drawing Text on the Background
- Detecting Win Condition
- Changing active World
- Adding Mechanics for Lobster
- Adding Game Over World
- Winning the Game



Step 1: Setting up the Crab Scenario

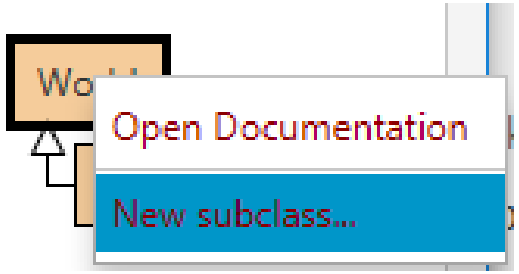
The Crabs Scenario (Part 3)

- Download the **Lab_4.zip** file from Omnivox, which contains the **modern-crab** Scenario
- Unzip the contents to somewhere on your USB key or hard disk.
- Open the scenario in that location with **Greenfoot**
- You should see the standard Greenfoot interface, with an empty sandy world

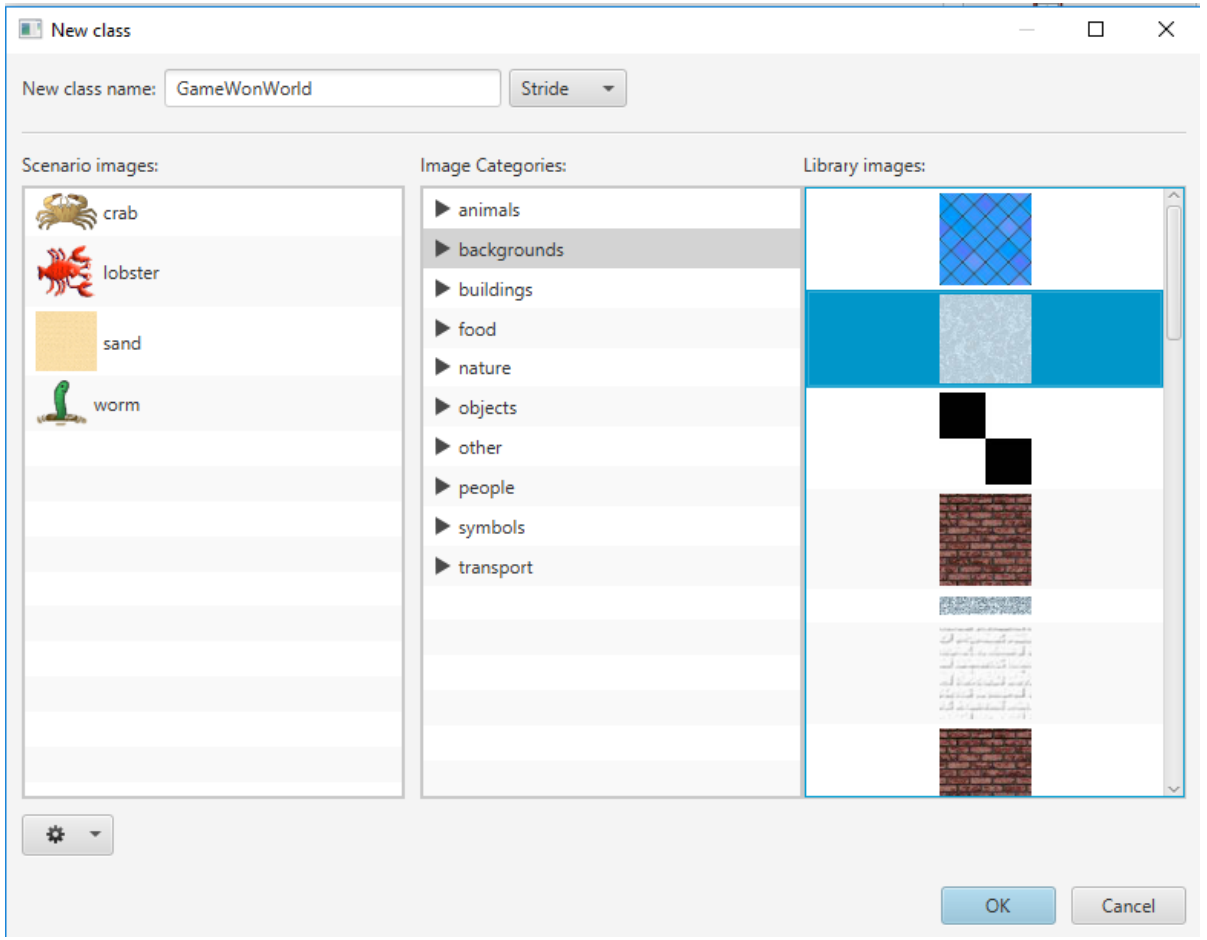


Step 2: Create the GameWonWorld class

Right-click on the **World** class and select new subclass.



Enter ***GameWonWorld*** for the name of the class and select a background image



Step 3: Implementing GameWonWorld

Double-click on the **GameWonWorld** class

Adjust the size of the World to match the CrabWorld size by Changing the constructor parameters from ***super(600, 400, 1)*** to ***super(560, 560, 1)***

- The 3 numbers contain the number of cells (columns, rows) and the third number is the size of cells in pixels.

Draw the text "**You Won the Game!**"

```
Constructor for objects of class GameWonWorld.

public GameWonWorld()
    super(560, 560, 1)
    showText("You won the game!", 260, 260)
```

Step 4: Improving the Text Rendering in GameWonWorld

The **showText()** method in the World class is limited, and "ugly"

We can write our own method to draw text making use of the **Greenfoot Font** class

Write method on the write and replace the method to draw the **You Win** line on the screen.

Call it such as below:

```
public void showTextWithBigBlackFont(String message, int x, int y)
    var GreenfootImage bg = getBackground()
        Font font = new Font(50)
    bg.setFont(font)
    bg.setColor(Color.BLACK)
    bg.drawString(message, x, y)
```

```
showTextWithBigBlackFont("You Win", 180, 200)
```

Step 5: Transition to GameWonWorld when the game is finished

Detecting that the game is finished
- We should check every time step from the **Crab** object.
- This will happen when all worms are eaten.
- Transition to a ***GameWonWorld***

In the **Crab** class, you can
- add the two methods on the right
- call them from **act()**

```
When the list of Worm objects in the world is empty, we win the game!
public boolean isGameWon()
    var World world = getWorld()
    if (world.getObjects(Worm.class).isEmpty())
        return true
    else
        return false
```

```
Make the current world, the GameWonWorld!
public void transitionToGameWonWorld()
    var World gameWonWorld = new GameWonWorld()
    Greenfoot.setWorld(gameWonWorld)
```

```
public void act()                                    overrides method from Actor
    moveAndTurn()
    eat()

    if (isGameWon())
        transitionToGameWonWorld()
```

Step 6: Finish coding the Lobster

Create a **GameOverWorld**

Finish the lobster implementation

a) make your **crab** disappear when it intersects with the lobster
b) play a scary sound when it happens!
c) Transition to **GameOverWorld**

Randomize the initial direction of lobsters when they get instantiated

- Implement a **constructor** and call turn with a random value between 0 and 359

Step 7: More Interesting Things …

To make things interesting, transform **worms** into **lobsters** when they get touched by a lobster

a) Remove the worm on intersection with lobster
b) Add a new lobster at this position
   - Look at **Greenfoot** documentation to find out how to add an actor
   - Hint: World **AddObject()** method - Actor **getX()** and **getY()**

If you have time left, add your own twist to the game!!

Questions

?