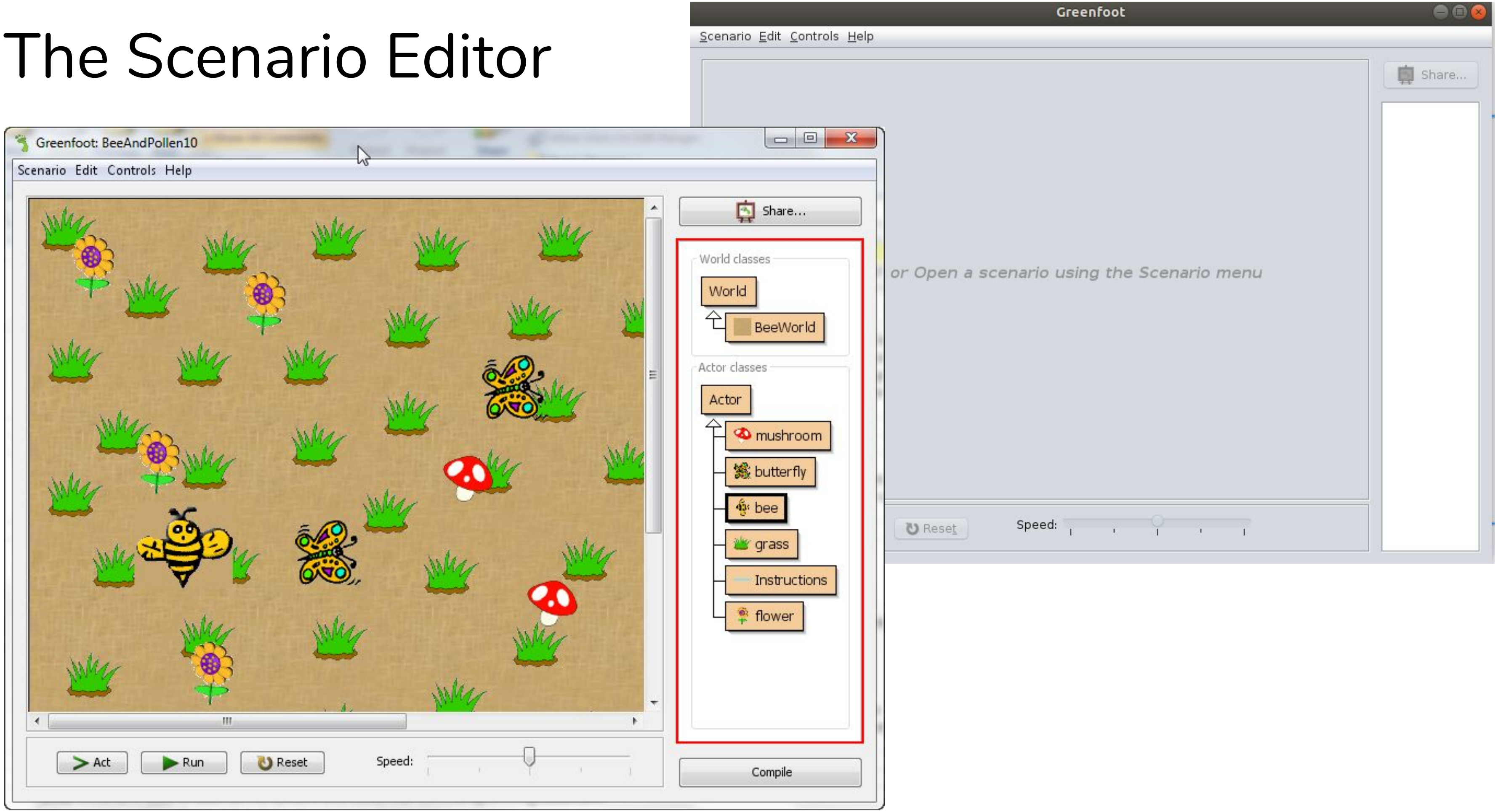
Building games in Greenfoot

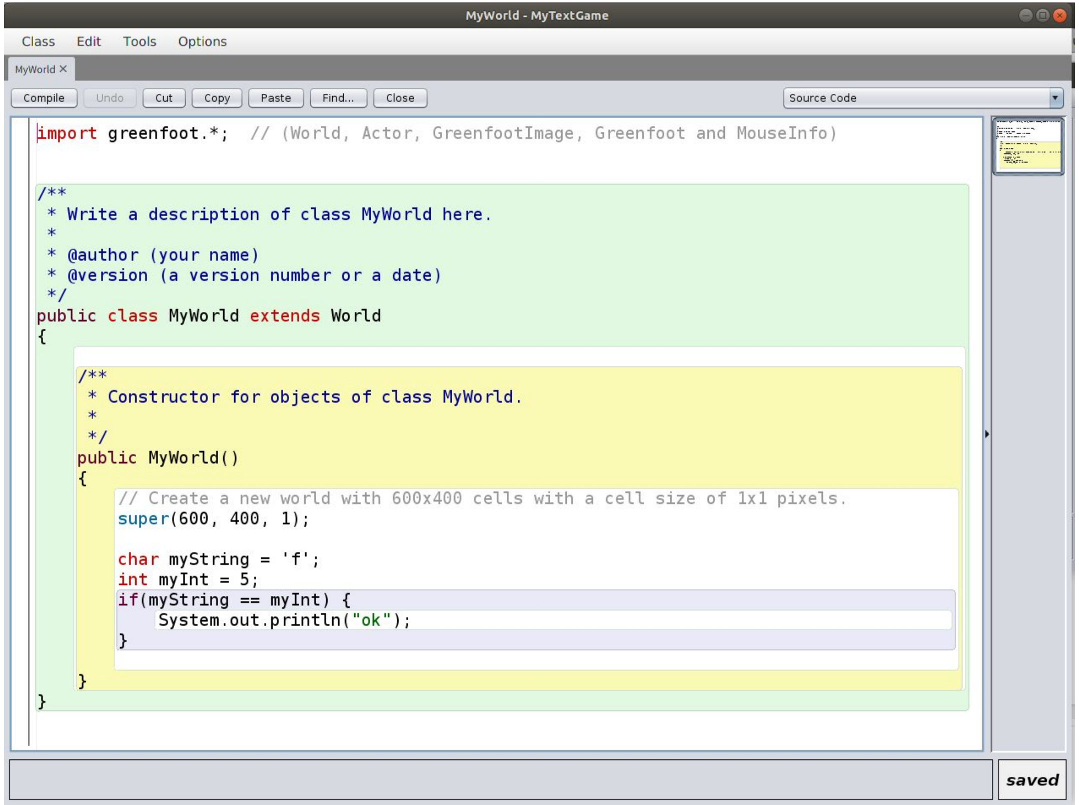
– introduction

The Scenario Editor



or Open a scenario using the Scenario menu

The Code editor



```
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);

        char myString = 'f';
        int myInt = 5;
        if(myString == myInt) {
            System.out.println("ok");
        }

    }
}
```

Why a Scenario Editor?

- Easier way to get the results of the code on screen

- A level editor is a good thing to have.

A typical first greenfoot game is an action game

- but not necessarily!

The ingredients of a 2D action game

At least:

- Player
- Enemies
- Interactivity
- Game map / setup
- Victory conditions
- Clock / timer

Where to find these ingredients in Greenfoot

Interactivity

Objects can move and are visible

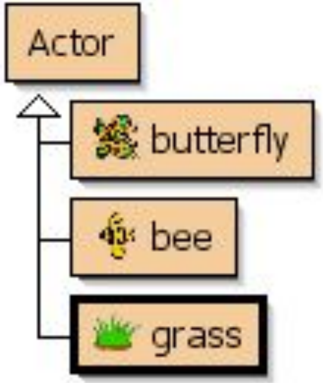
- Player
- Enemy

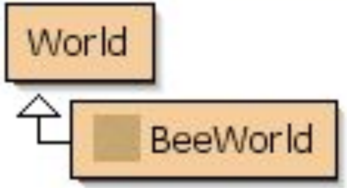
Actor

Objects that sits or are intangible

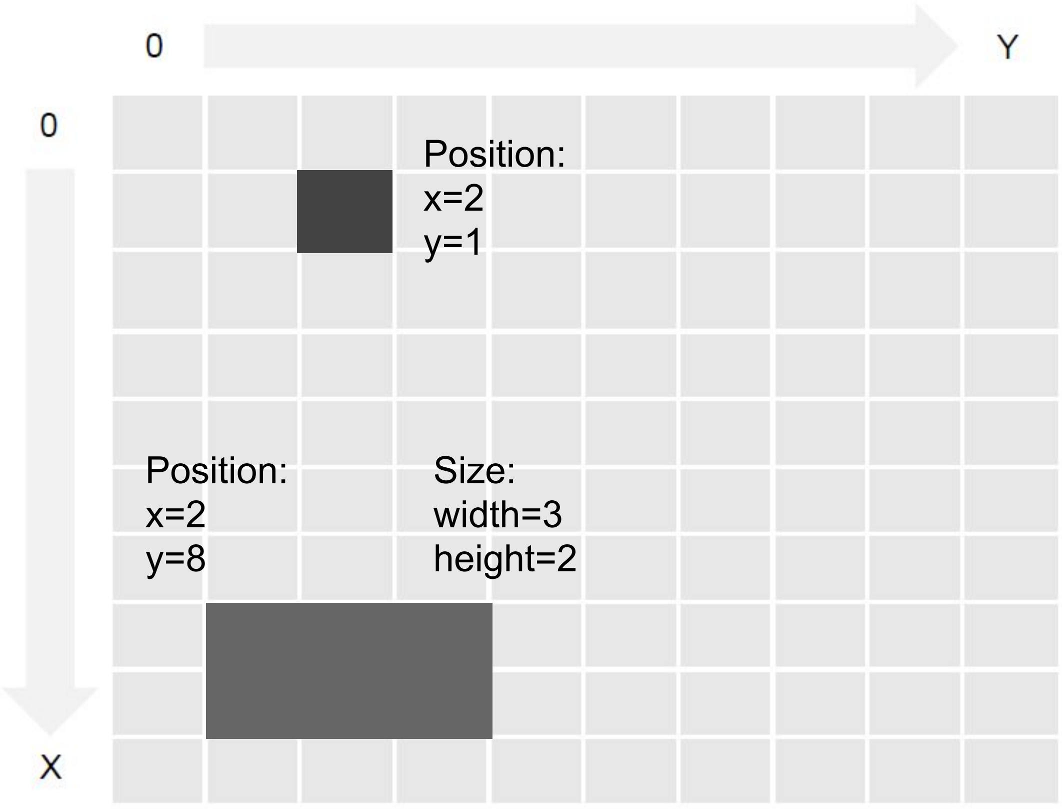
- Game map / setup
- Victory conditions
- Clock / timer

World





Position objects in using a coordinate system.



There are actually two coordinate systems

- World coordinates🗺️ =game world map
- Screen coordinates📺 =always pixels

Object orientation

Key concepts:

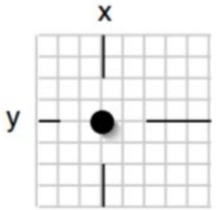
- class
- object
- state
- behaviour

Actors

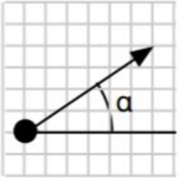
‘Actors’ have a predefined state

- image

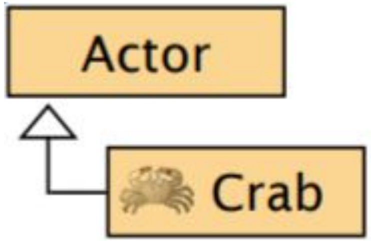
- location (in the world)



- rotation



The Act method



```
public void act {

        ...

}
```

code goes here

Method calls

Format:

```
methodName(parameter);
```

or `methodName();`

Examples:

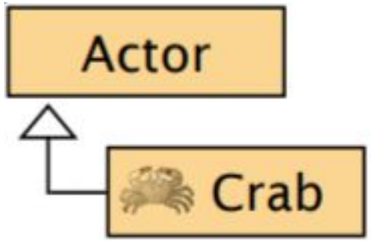
```
move(3);
```

```
turn(5);
```

Available methods

```
void move(int distance)

void turn(int amount)

int getX()

int getY()

…
```

Inherited from Actor



Method calls - examples

Specification

```
void move(int distance);

void turn(int amount);
```

Your code

```
move(12);

turn(45);
```

Return values

Specification	**Your code**
`int getX()`	397 ← **`getX();`**
`int getY()`	207 ← **`getY();`**

Let’s start coding!

Schedule

Tuesday

Focus on Chapters:

- ***Get started with Greenfoot***
- ***Add and position objects***
- ***Explore the Code Editor***

Wednesday: Remaining chapters, Thursday: General hints

Friday: LIVE-coding workshop using fresh new tutorial ✨

Instruction material

https://goo.gl/hKi2Lz