

Java Programming with Greenfoot

Greenfoot and Java

- Greenfoot is middle-ware that sits between Java and you
- Full Java capability
- Has easy-to-use graphics/sound library
- Has a built-in framework for game loops

Greenfoot Demo

- Start a new project
- Add a custom world
 - Right-click World
 - Create a new subclass "MyWorld"
 - Choose a background image
- Compile the code and see what you get
 - Explanation of `super(600, 400, 1);`

Greenfoot Demo

- Add a custom actor
 - Right-click Actor
 - Create a new subclass `Ball`
 - Choose a ball image
- Add a Ball to the world
 - Open MyWorld. AFTER the super call, add:

```
Ball ball = new Ball();  
    addObject(ball, getWidth()/2, getHeight()/2);
```

- Compile and run the code
 - Explain what compiling does
 - You should see a ball in the center of the world

Greenfoot Demo

- Code explanation

`Ball ball = new Ball()` creates a new Ball object
`addObject(obj, x, y)` adds an object to the canvas
`getWidth()` gets the width of the canvas
`getHeight()` gets the height of the canvas

- Note the difference

Javascript

`var ball`

`var count`

vs

vs

Java

`Ball ball`

`int count`

Greenfoot Demo

- Animation in Greenfoot
 - Every subclass of Actor has an `act()` method
 - `act()` is called every tick
- Animate the ball
 - Add this code to the Ball's `act()`:
`setLocation(getX(), getY() + 2);`

Greenfoot Demo

- API Documentation
 - Double-click World or Actor
 - Help → Greenfoot Class Documentation
- Bounded vs Unbounded worlds
 - Unbounded is usually preferred, otherwise the center of an object cannot go past the screen edge
 - Change your custom World code:
`super(600, 400, 1, false);`

Global variables in Java

- Known as **attributes** or **instance variables**
- Declare inside class but before methods

- Add these instance variables to your Ball

```
int dx = 3;
```

```
int dy = 3;
```

- Change the Ball code so it moves by **dx, dy** each tick

Accessing World from Actors

- Each Actor has a method `getWorld()` that returns the World it's in.
- You can save this into a World object and then ask it questions

```
World w = getWorld();  
int height = w.getHeight();
```

- Or do it in one line

```
int height = getWorld().getHeight();
```

Bouncing Ball

- Finish the code so the ball bounces off all four walls
- Add more Balls
 - Select the Ball class and hold Shift
 - Click to drop more Balls
- Try adding Brick objects
 - When a Ball touches a Brick, the Brick is removed
 - Optional: Ball bounces off brick then Brick removed
- [Here is a video](#) that may help you if you are stuck.
- [Here is part 2](#), covering how to get width/height of an actor to make better ball bouncing code.