**Lab 1 - Intro to Mercury**

In order to get started with our game development, we need to get JGrasp set up for it to work with Mercury. Go through this lab on your own and alert a teacher if you run into any issues.

**Part 1 – Setting up JGrasp**

* Create a folder called “Project 3”
* Look for “Game Development - Day 1” on Edmodo, download the zip file, and extract its contents into the “Project 3” folder
* Open up JGrasp and navigate to the to the “Project 3” folder
* Go to Settings -> PATH/CLASSPATH -> Workspace
* On the window that pops up, look for the “New” button on the middle right
  + Click on “Browse” and navigate to the “Project 3” folder
  + Inside the “Project 3” folder, open the “native” folder, then click “Choose”, “OK”
* In the same popup, look at the tabs on the top for “CLASSPATHS” and click that tab.
* Click the “New” button, then the first “Browse” button at the top
  + Go to the “Project 3” folder
  + Select the “LWJGL.jar” file
  + Click “OK”
* Click on “New” again, then the first “Browse” button at the top
  + Go to the “Project 3” folder
  + Select the “Mercury.jar” file
  + Click “OK”
* Click the “OK” button to close the main pop up

**Part 2 – MyFirstGame**

Let’s make sure we have everything set up correctly by creating a basic “game”

* Create a folder called “Game Tutorials”
* Create a class in the “Game Tutorials” folder called **MyFirstGame**
* Copy the code on the next page in your class file (The rest of the instructions for this part are on the next page):

import com.radirius.mercury.framework.\*;  
import com.radirius.mercury.graphics.\*;  
  
public class MyFirstGame extends Core {

public MyFirstGame(CoreSetup coreSetup) {  
 super(coreSetup);  
 }  
  
 public static void main(String[] args) {  
 CoreSetup coreSetup = new CoreSetup("My First Game");  
 coreSetup.width = 800;  
 coreSetup.height = 600;  
  
 new MyFirstGame(coreSetup).start();  
 }  
  
 public void init() {  
   
 }  
  
 public void update() {  
   
 }  
  
 public void render(Graphics g) {  
   
 }  
  
 public void cleanup() {  
   
 }  
}

**Part 2 Continued**

* Compile and run your program, you should see a window pop up with a black background.
* You did it, you made your first game! (Yay?...)

**Part 3 – Adding Shapes**

Let’s try adding some actual stuff to make our game look more like an ACTUAL game. Let’s start by adding a green rectangle appear in our window:

* + Add the following import statement to the top of your file:

import com.radirius.mercury.math.geometry.\*;

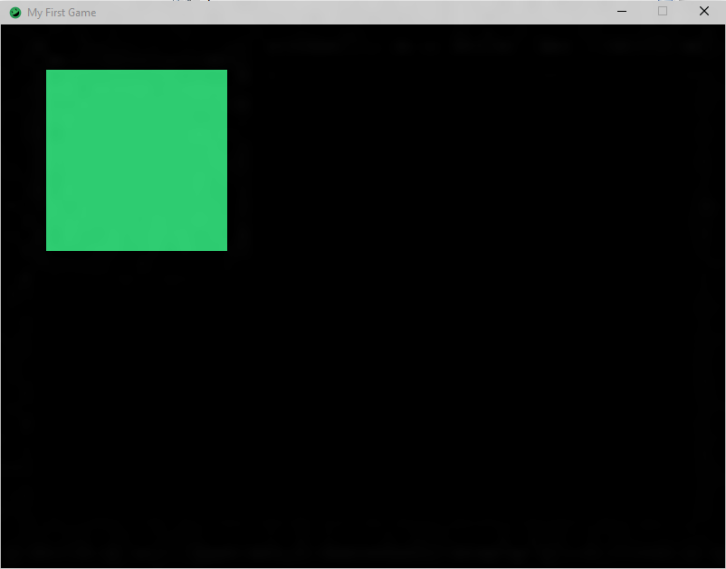
* + Look for the **render( Graphics g )** method and add the following code inside it:

Rectangle rect = new Rectangle( 50, 50, 200, 200 );

g.setColor( Color.GREEN );

g.drawShape( rect );

* + Compile and run your code and you should see the following:



As you can see, the code above created a green rectangle. The constructor for the **Rectangle** class takes the following parameters:

* + The X position of the **UPPER-LEFT** corner of the rectangle
  + The Y position of the **UPPER-LEFT** corner of the rectangle
  + The width (horizontal) of the rectangle
  + The height (vertical) of the rectangle

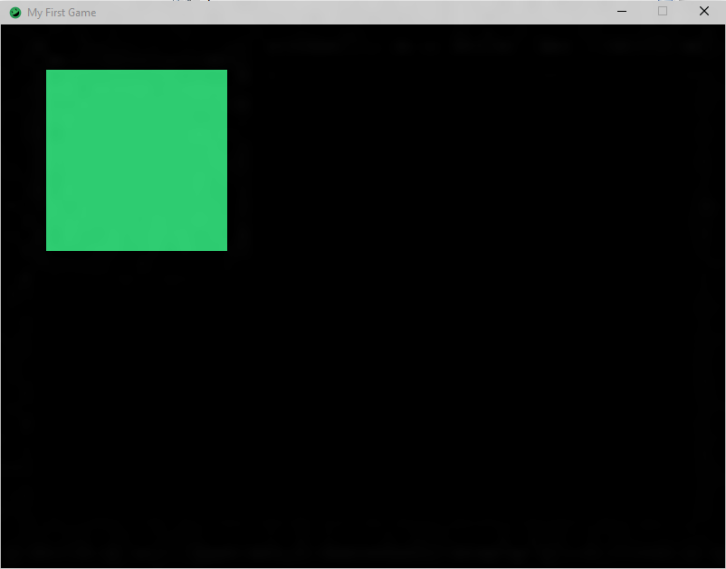
All of the above values are in **pixels** (dots on the screen). So, we created a rectangle that starts at position ( 50, 50 ) and is 200 x 200

If you look back at our original code, you’ll see the following:

CoreSetup coreSetup = new CoreSetup("My First Game");  
 coreSetup.width = 800;  
 coreSetup.height = 600;

This says we are creating a window that is **800 pixels wide** by **600 pixels tall.** The upper-left corner of the window is always ( 0, 0 ). X values run left-to-right, while Y values run up-to-down. Let’s look at an image of the screen to get a better idea of values:

0 X 800

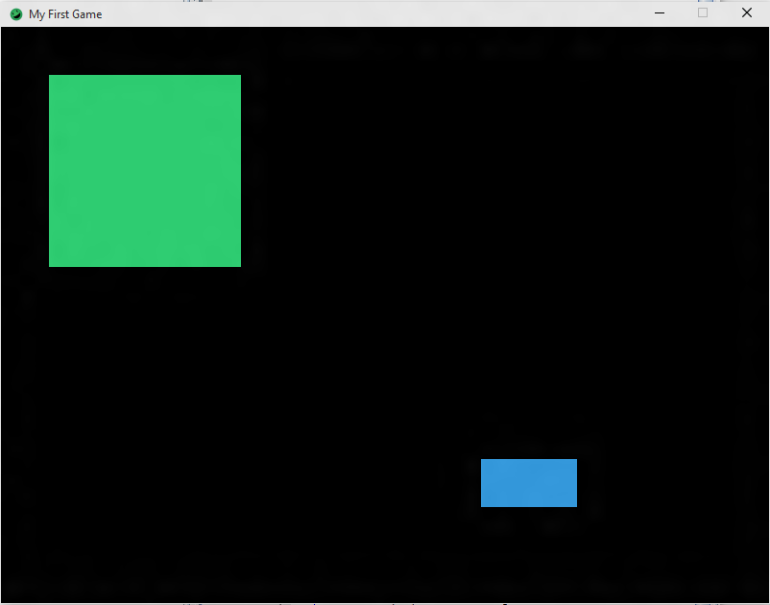


0

Y

600

So our green square is 50 pixels to the right and 50 pixels down from the upper-left corner of the window. Let’s try creating another rectangle. Your rectangle should be at position ( 500, 450 ), 100 pixels wide by 50 pixels tall, and the color **BLUE**. Compile and run your code and you should see the following:



**Part 4 - Final**: For the rest of class, trying making more rectangles of different colors, positions, and sizes.