Lab 01: Drawing to the Screen

This lab serves as a quick Java refresher and introduction to GUI applications.

Getting Java Working on your own Machine.

You are welcome to use Remus/Romulus for this class.
If you prefer to run things locally on your own machine, the following may be helpful.
You're free to use an IDE (e.g., IntelliJ) if you wish.

Do not use any IDE that is named after a bird, or any IDE that claims to have medical expertise. (If that doesn't make sense to you, don't worry about it. If you have any questions, ask.)

Download the Java Development Kit: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
(You're free to use Java 8, Java 11 or Java 17.)
Now may need to edit your `PATH` variable to include the directory where you installed Java.
Be careful in this process. You have the opportunity to do horrible to your computer when you're editing your `PATH` variable.

https://www.java.com/en/download/help/path.xml

Ask me for help if you feel unsure about anything.

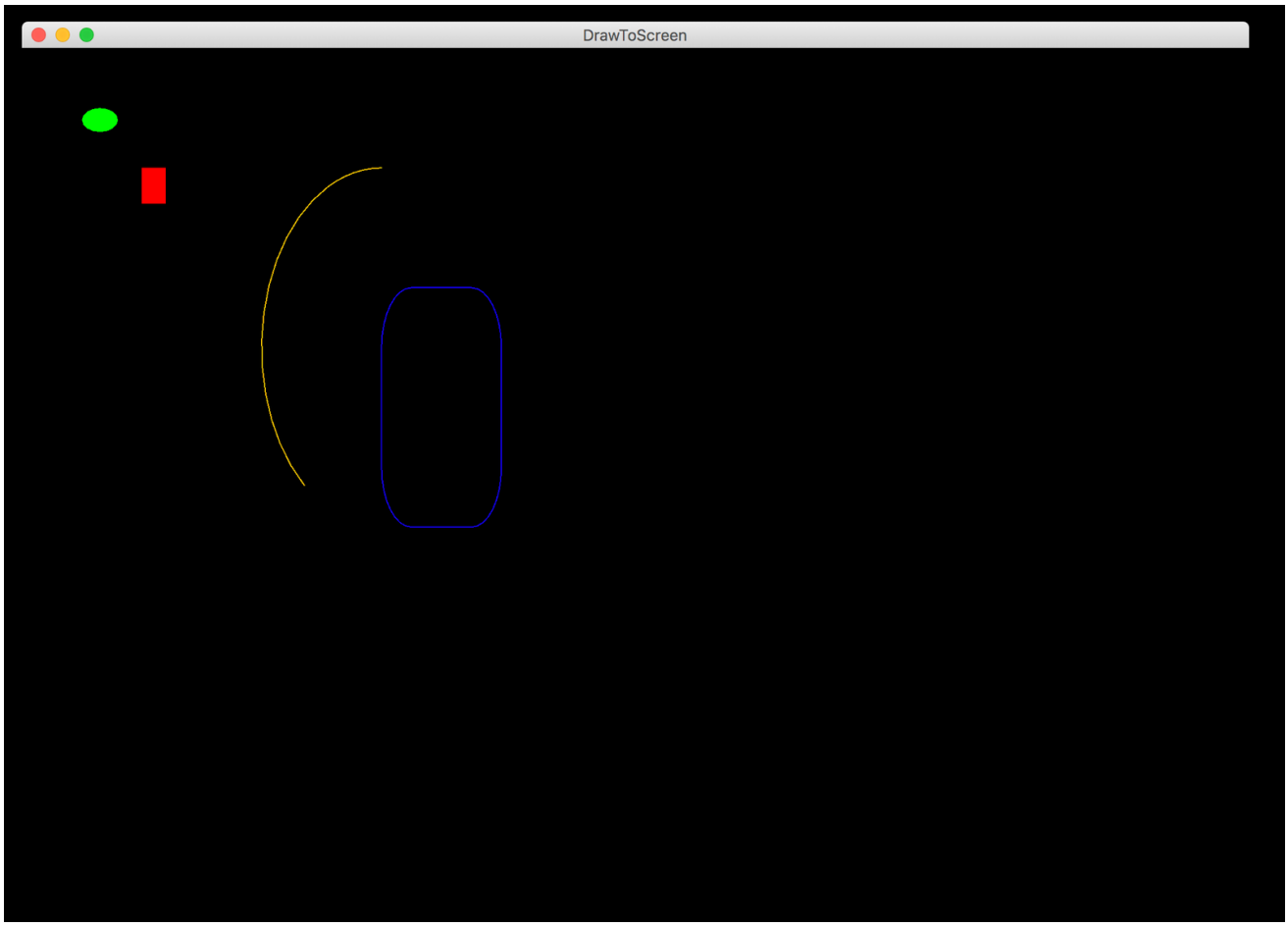
Drawing to the Screen

First, copy and paste the following code into a file called `DrawToScreen.java`.

```
1  import java.awt.Color;
2  import java.awt.Graphics;
3  import java.awt.Dimension;
4  import javax.swing.JPanel;
5  import javax.swing.JFrame;
6
7  public class DrawToScreen extends JPanel{
8      public static final int BOX_WIDTH = 1024;
9      public static final int BOX_HEIGHT = 768;
10
11     public DrawToScreen(){
12         this.setPreferredSize(new Dimension(BOX_WIDTH, BOX_HEIGHT));
13     }
14
15     //Your code here, if you want to define additional methods.
```

```
16
17      @Override
18      public void paintComponent(Graphics g) {
19          super.paintComponent(g);
20          //Your code here: feel free to remove what is below
21          g.setColor(Color.BLACK);
22          g.fillRect(0, 0, BOX_WIDTH, BOX_HEIGHT);
23
24          g.setColor(Color.GREEN);
25          g.fillOval(50, 50, 30, 20);
26
27          g.setColor(Color.RED);
28          g.fillRect(100, 100, 20, 30);
29
30          g.setColor(Color.BLUE);
31          g.drawRoundRect(300, 200, 100, 200, 50, 100);
32
33          g.setColor(Color.ORANGE);
34          g.drawArc(200, 100, 200, 300, 90, 140);
35      }
36
37      public static void main(String args[]){
38          JFrame frame = new JFrame("DrawToScreen");
39          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
40          frame.setContentPane(new DrawToScreen());
41          frame.pack();
42          frame.setVisible(true);
43      }
44  }
```

Now run `javac DrawToScreen.java` followed by `java DrawToScreen` (if you'd rather use IntelliJ, go for it), and you should see something like this:

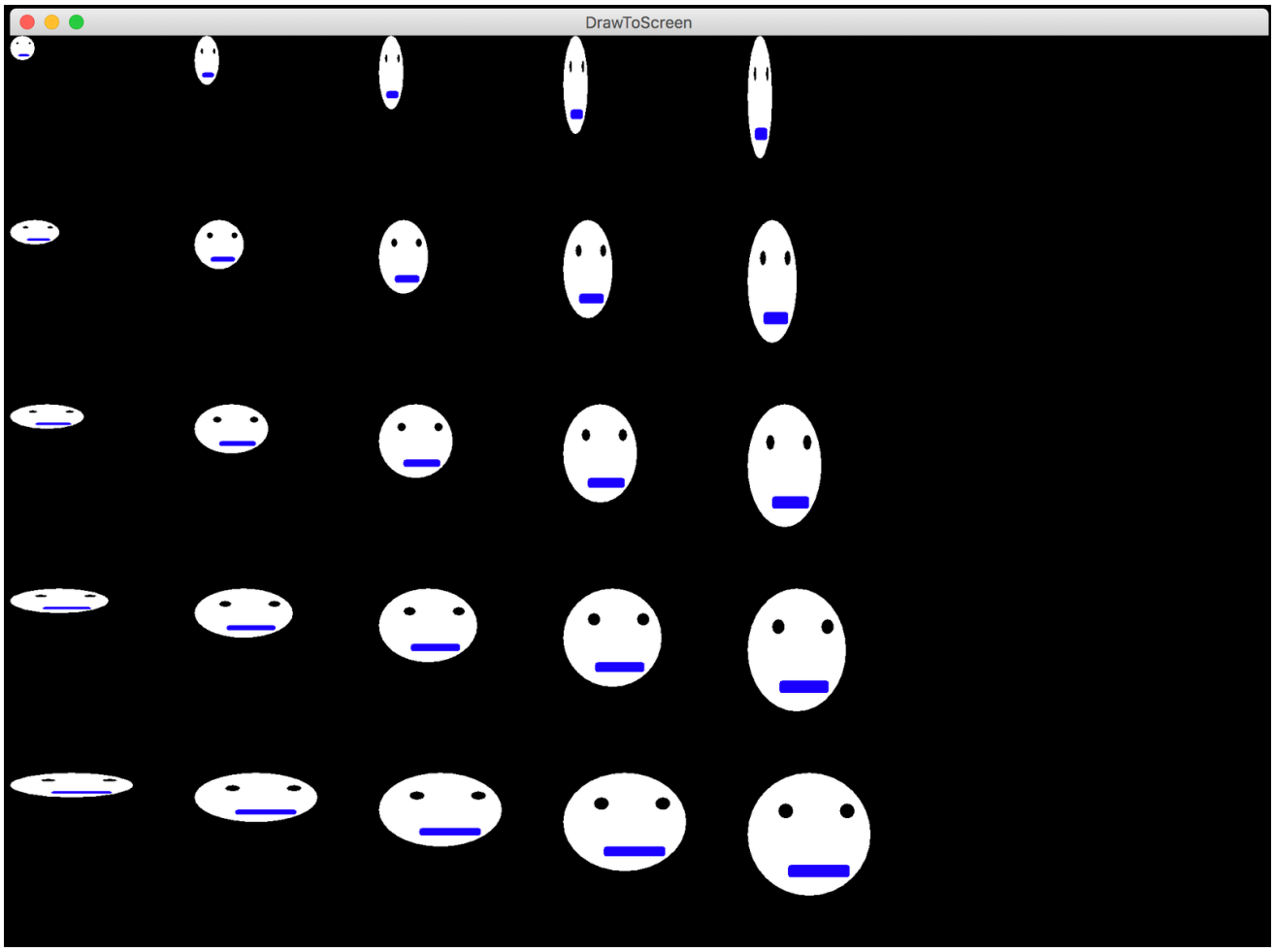


In this lab you will be editing the `paintComponent` method. For documentation on how methods like `fillRect` work, and other available methods, see:

http://docs.oracle.com/javase/8/docs/api/

To complete this lab you should:

1. Draw a object of your choosing. This can be a stick figure, a building, the face of a soulless clown from your childhood, or whatever you want, but it needs to contain at least 5 base shapes (ovals, rectangles, what have you).
2. Draw 25 copies of your object in a 5 x 5 grid. If you want to challenge yourself, make each copy distorted based on it's position in the grid, as in the following figure:



Note: Making 25 copies in a grid naively is a painful process of busywork. If you instead define a method which draws one copy of your object, though, and remember how nested loops work, then it's far easier.

What you should submit:

`DrawToScreen.java`

In addition, fill out the following survey:
https://cosc112s22.page.link/Lab01Survey