Objectives

1. Understand how to start developing a graphical application in Java using Graphics API

Graphics with Java

We can draw graphics in Java using Java swing and Java 2D API. Swing is a package that lets you create GUI applications. Java swing API contains many classes for creating various UI elements. The hierarchy of Java swing API is as the figure 1 below.

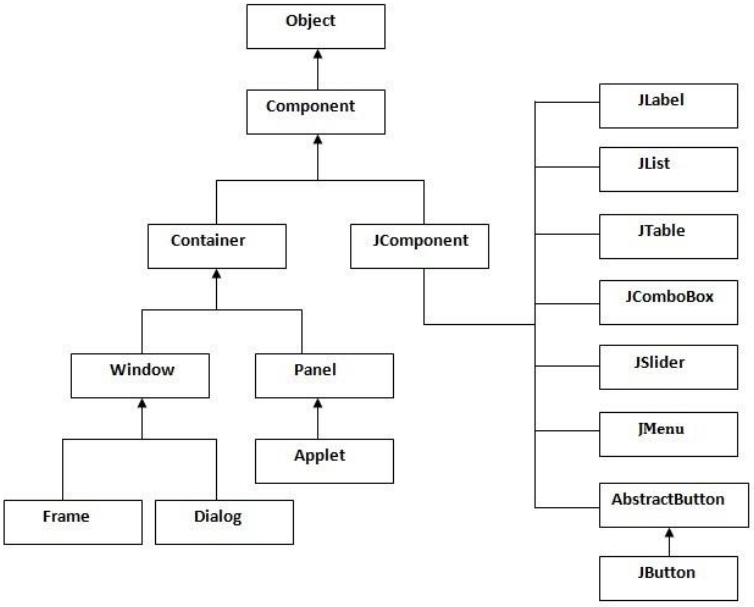

Figure 1 – The hierarchy of Java swing API [1]

We will not go in detail on how Java swing works. We will only use a benefit of Java swing to draw graphics. Therefore, what we will do here is to initiate a Java swing window which we can draw 2D graphics on it.

To create a paintable Java swing window, we will first create a class called ***GraphicsSwing*** which is extended from JPanel. Importantly, package java.awt and javax.swing are required to be imported.

GraphicsSwing.java

```
import java.awt.*;
import javax.swing.*;


class GraphicsSwing extends JPanel
{
   public static void main(String[] args)
   {
   }
}
```

In the main method, this JPanel is added to a JFrame. Also, set some needed attributes such as window's title, size, and behaviours.

GraphicsSwing.java

```
...
public static void main(String[] args)
{
   GraphicsSwing m = new GraphicsSwing();

   JFrame f=new JFrame();
   f.add(m);
   f.setTitle("First Swing");
   f.setSize(600,600);
   f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   f.setVisible(true);
}
...
```

Once the program is compiled and run, a GUI window will be shown in desired size and title.

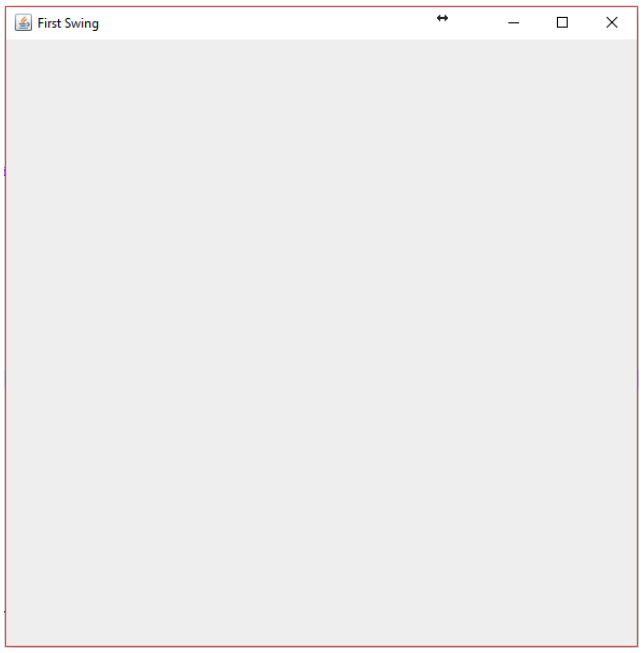

Figure 2 – First Java GUI window

With a GUI window is ready, we can now draw Graphics using Java 2D API. Add a method call ***paintComponent*** with ***Graphics g*** parameter. This will be the method where everything is drawn.

GraphicsSwing.java

```
...
public void paintComponent(Graphics g)
{
	g.drawString("Hello", 40, 40);
	g.setColor(Color.BLUE);
	g.fillRect(130, 30, 100, 80);
	g.drawOval(30, 130, 50, 60);
	g.setColor(Color.RED);
	g.drawLine(0, 0, 200, 30);
	g.fillOval(130, 130, 50, 60);
	g.drawArc(30, 200, 40, 50, 90, 60);
	g.fillArc(30, 130, 40, 50, 180, 40);
}
...
```

The example above will show the drawing as below.

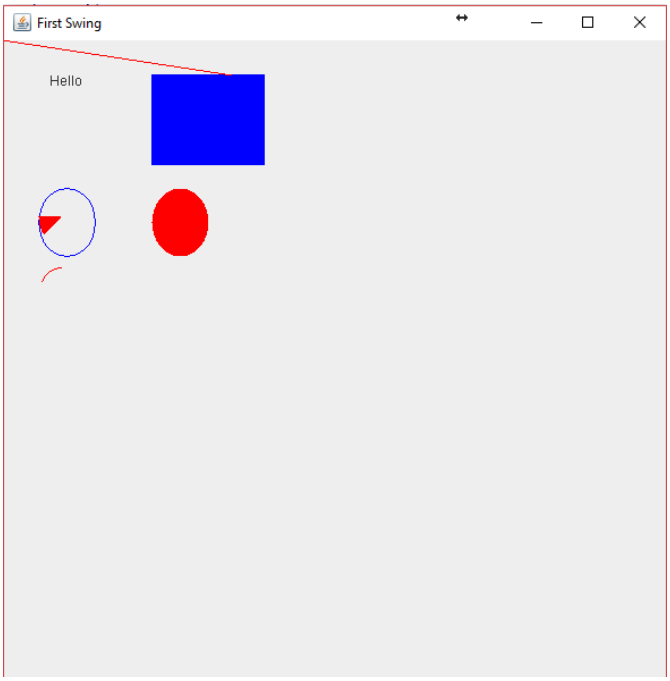

Figure 3 – First drawing

Our intention here is to imitate the pixel plots for geometry drawing algorithms in next couple of weeks. Ideally, each pixel plot should be done by drawing a 1x1 pixel dot. However, sadly, Java 2D API does not contain any dot drawing. Therefore, to draw a dot, we simply draw 1x1 pixel square or 1x1 pixel circle. We can write this in a separate method.

GraphicsSwing.java

```
...
private void plot(Graphics g, int x, int y)
{
    g.fillRect(x, y, 1, 1);
}
...
```

With this, everything is prepared for any geometry drawing algorithms.

Tasks (Submit through Google Classroom)

1. Reproduce the figure 3.
2. Draw another 10 different colour dots wherever you want on the canvas which can easily be visible.
3. Write up on IF IT WERE YOU, HOW WILL YOU DRAW A LINE FROM TOP-LEFT CORNER TO THE BOTTOM-RIGHT CORNER OF THE SCREEN. You do not need to go on a detail, just give me a general idea.
4. Submit through Google Classroom before Monday 23.59.

References

1. SSS IT Pvt Ltd: The hierarchy of java swing API. (2017).