Java Graphics (Swing)

- Java comes with a graphics library called Swing as part of the JDK.
- Swing is based on the Model View Controller (MVC) Architecture
    - Model: Represents the data and logic of the application
    - View: The "visible" components of the application
    - Controller: The "glue" that ties the Model and View together.
- Swing allows tools and functionality to allow programmers to create a custom Graphical User Interface (GUI).

Frame

- A top level window with a title and a border.
- Organized in a Cartesian x,y coordinate organization
    - Top left of the window is (0,0)
    - x increases as it moves right on the window
    - y increases as it moves down the window

Example

```
// Construct a JFrame object
JFrame frame = new JFrame();

// Sets the size of a frame
frame.setSize(500,500);

// Sets the title
frame.setTitle("ICS 45J Frame");

// Sets the default close operation
// When user closes frame, application terminates
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Makes the Frame visible
frame.setVisible(true);
```

Components

- You can't draw on a Frame directly.
- Frame contains certain components that contain the graphic pieces you want to add.
- Customized components must extend JComponent.

Example (Drawing a Rectangle)

```
public class DrawComponent extends JComponent {

      public void paintComponent(Graphics g) {
            // Graphics2D - a better implementation of 2D Graphics
            Graphics2D g2 = (Graphics2D) g;

            // params (x, y of upper left, width, height)
            Rectangle box = new Rectangle(5,10,20,30);
            g2.draw(box);
      }
}
```

// then add the component to the frame

```
DrawComponent x = new DrawComponent();
frame.add(x);
frame.setVisible(true);
```

Example (Drawing other Shapes)

```
Rectangle box = new Rectangle(5,10,20,30);
g2.draw(box);

//  x, y upper left bounding box, width, height
Ellipse2D.Double circle = new Ellipse2D.Double(30,10,100,50);
g2.draw(circle);

// x1, y1, x2, y2
Line2D.Double line = new Line2D.Double(50,50,80,80);
g2.draw(line);
```

Colors

- Can set your own custom colors using RGB values.
- Can set the Graphics2D object with a specific color and then draw in that color.

Example

```
Color white = new Color(0,0,0) // R,G,B values in parameters
Color black = new Color(255,255,255)

g2.setColor(Color.BLUE);
// params (x, y of upper left, width, height)
Rectangle box = new Rectangle(5,10,20,30);
g2.draw(box);

g2.setColor(Color.GREEN);
Ellipse2D.Double circle = new Ellipse2D.Double(30,10,100,50);
g2.draw(circle);

g2.setColor(Color.ORANGE);
Line2D.Double line = new Line2D.Double(50,50,80,80);
g2.draw(line);
```

Draw vs. Fill

- draw() draws the outline of shapes in whatever color was set.
- fill() draws the shape and fills it in with whatever color was set.

Other Components

- Buttons
  - Provide a component that performs some action when "clicked."
- Text
  - Can specify number of characters or an area for text.
- Radio Buttons
  - Only one selection can be made at a time.
- Combo Boxes
  - Pull down menu with option to enter selection.
- … and many more!

Layout Management

- Programmers can define certain layouts when components are added to the Frame.
- **Flow Layout** (default): Arranges components from left to right, top to bottom.
- For a list of layouts, see https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html