CECS 277 – Lecture 15 – GUI Applications

GUI – A GUI is a Graphical User Interface. Creating a graphical interface for your program is a great way to make your program more user friendly. To make a GUI application you will need to import two packages. javax.swing creates the application window and components, and java.awt contains graphics and events.

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

Your class also needs to extend the JFrame class in order to create the main window:

```
 public class GuiApp extends JFrame {
```

Your window needs a panel to draw components on, this is a class inherited from JPanel.

```
     DrawingPanel panel = new DrawingPanel();
```

Now the window needs to be constructed:

```
     GuiApp() {
          super("Window"); //title of the app
          setSize(300, 300); //window size
          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
          this.getContentPane().add(panel);
          setVisible(true); //display window
     }
```

Drawing Shapes – The Graphics class allows you to draw different shapes and colors to the application window, but you must include a special method called paintComponent(). The upper left hand corner of the window is the location (0,0) (including the title bar). The horizontal axis is the x direction (ranges from 0 to getWidth()) and the vertical axis is the y direction (ranges from 0 to getHeight()).

```
public class DrawingPanel extends JPanel {
     public DrawingPanel(){
          this.setSize(300,300);
          this.setBackground(Color.BLUE);
     }
     public void paintComponent(Graphics g){
          super.paintComponent(g);
```

<u>Line</u> – drawLine draws a line from the first position x1,y1 to the second position x2,y2.

```
          g.drawLine( 20, 42, 75, 75);
```

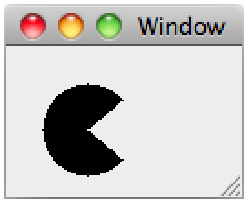
<u>Rectangle</u> – drawRect draws the outline of a rectangle and fillRect draws a filled in rectangle. They both take in parameters of the x and y location of the upper left hand corner of the rectangle, and the width and height of the rectangle. If the width and height are the same, then it will draw a square.

```
          g.drawRect( 20, 42, 50, 50);
```

Oval – drawOval and fillOval create ovals that take in the x and y location of the upper left hand corner of a box bounding the oval, and then the width and height of the oval.

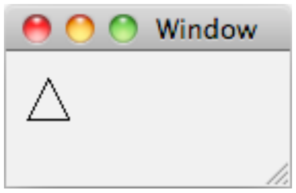
```
        g.fillOval( 20, 42, 50, 50);
```

Arc – drawArc draws the outline of a portion of an oval and fillArc draws a filled in portion of an oval. They both take in the x and y location of the upper left hand corner of a bounding box containing the oval, the width and height of the oval, the angle measurement to the beginning of the arc, and the angle of the arc.



```
        g.fillArc( 20, 42, 50, 50, 40, 280);
```

Polygon – drawPolygon draws the outline of a polygon, fillPolygon draws a filled in polygon, and drawPolyline draws a segmented line. They take in parameters of an array of x locations, an array of y locations and the number of points.



```
        int [] xPts = {10, 20, 30};
        int [] yPts = {55, 35, 55};

        g.drawPolygon( xPts, yPts, 3);
```

String – drawString draws a string at the given x and y location starting from the lower left hand corner of the string.

```
        g.drawString( "Hello", 15, 25);
```

Image – Draws an image to the window. You can only use JPG, GIF, or PNG image files, and they need to be stored in the project folder. The statement requires a try/catch block.

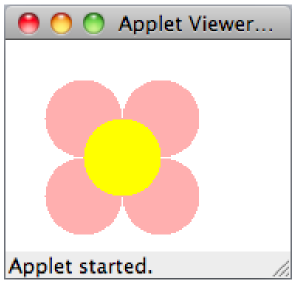
```
    BufferedImage img = ImageIO.read(new File("img.gif"));
    g.drawImage(pic, 10, 10, this); //x, y coords
```

Changing Colors – You can change colors of your shapes by either using default colors (Black, Blue, Green, Gray, Yellow, Red, etc) or by creating your own using RGB values (which each range from 0 to 255).

```
        g.setColor(Color.GREEN);
        Color purple = new Color(125, 35, 200);
        g.setColor(purple);
```

Making a Pattern Method – You can create a new shape by drawing a series of objects.

```
        drawFlower(g, 75, 75, 50);
    }
    public void drawFlower(Graphics g,int x,int y,int s){
        g.setColor(Color.PINK);
        g.fillOval(x, y, s, s);
        g.fillOval(x-s, y-s, s, s);
        g.fillOval(x, y-s, s, s);
        g.fillOval(x-s, y, s, s);
        g.setColor(Color.YELLOW);
        g.fillOval(x-s/2, y-s/2, s, s);
    }
}
```


Applet Viewer...
Applet started.