CECS 277 – Lecture 14 – GUI Applications

Graphical User Interface (GUI) – Creating a graphical interface for your program is a great way to make your program more user friendly and more visually appealing. To make a GUI application you will need to import two packages. javax.swing is used to create the application window and components, and java.awt contains the drawing tools.

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

Your class needs to extend the JFrame class in order to create the main window. In the constructor, functions of JFrame are called to do some basic setup:

```
public class Window extends JFrame {
     public static void main( String [] args ) {
          Window w = new Window();
     }
     public Window() {
          setBounds(100, 100, 500, 500);//x,y,w,h of window
          setTitle( "Drawing Demo" );
          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
          setContentPane( new Panel() );
          setVisible( true );
     }
```

In order to draw on the window, you will need to create a Panel to place on your Frame. This Panel is created as an inner class in the Window class, however, it can be created separately.

```
     public class Panel extends JPanel {
          public Panel() {
               setBackground( Color.BLUE );
          }
```

Drawing Shapes – The Graphics class allows you to draw different shapes and colors on the Panel. To do this, you must override the paintComponent() method. To draw shapes, you must specify a location on the window where it should be drawn. The upper left hand corner of the window is at (0,0) (not including the title bar). The horizontal axis is the x direction (0 - getWidth()) and the vertical axis is the y direction (0 - getHeight()). Shapes are drawn in order, with the newest shapes drawn on top of the oldest.

```
          public void paintComponent( Graphics g ) {
               super.paintComponent( g );
```

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

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

Rectangle – 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.

```
               g.drawRect( 20, 20, 50, 75 );
```

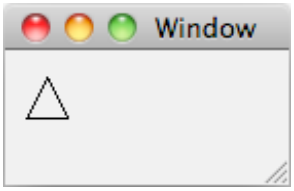
Oval – drawOval and fillOval create ovals that take in the x and y location of the upper left hand corner of a box bounding the oval, and then the width and height of the oval.

```
g.fillOval( 20, 20, 50, 75 );
```

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, 20, 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( img, 10, 10, this ); //x & y coordinates
```

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). All subsequent drawings made after changing the color will be in that color until the color is changed again. The default color is black.

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

Making a Stamp 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);
    }
}
```