Noah Streveler

Assignment 3

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

public class Homework {

    public static void main(String[] args) {
        JFrame f = new AppFrame("");
    }
}

class AppFrame extends JFrame{
    public AppFrame(String title) {
        super(title);

        //add panels
        Info i = new Info();
        this.setLayout(new GridLayout(1,2));
        this.add(new TopPanel(i));

        this.setSize(1000, 500);
        this.setLocation(0, 0);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
                this.setVisible(true);
        }
}

class TopPanel extends JPanel{
        Info in;
        int num = 0;
        double size = 30, x, y;
        JTextArea shapeSize, color, location;
        java.util.List<Shape> shapes;
        Color c = new Color(0,0,0);

        public TopPanel(Info i) {
                super();
                in = i;
                shapes = new ArrayList<Shape>();
                this.setBorder(BorderFactory.createTitledBorder("Panel"));
                this.setLayout(new BorderLayout());
                this.addMouseListener(new MsListener());

                shapeSize = new JTextArea();//Initializing text box
                shapeSize.setBounds(305, 340, 125, 70);//Setting box size
                this.add(shapeSize, "Center");//Setting location of the box

                color = new JTextArea();//Initializing text box
                color.setBounds(435, 340, 125, 70);//Setting box size
                this.add(color, "Center");//Setting location of the box
```

```
location = new JTextArea();//Initializing text box
location.setBounds(565, 340, 125, 70);//Setting box size
this.add(location, "Center");//Setting location of the box

JPanel q = new JPanel();
this.add(q, "Center");

JButton btnTri = new JButton("Triangle");
JButton btnRect = new JButton("Rectangle");
JButton btnCirc = new JButton("Circle");
JButton btnUndo = new JButton("Undo");

JPanel p = new JPanel();
p.add(btnTri);
p.add(btnRect);
p.add(btnCirc);
p.add(btnUndo);
this.add(p, "North");

JButton btnSize = new JButton("Shape Size");
JButton btnColor = new JButton("Color");
JButton btnLocate = new JButton("Location");
JButton btnClear = new JButton("Clear Text Boxes");

JPanel b = new JPanel();
b.add(btnSize);
```

```
b.add(btnColor);
b.add(btnLocate);
b.add(btnClear);
this.add(b, "South");

btnTri.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
                num=1;
        }
});

btnRect.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
                num=2;
        }
});

btnCirc.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
                num=3;
        }
});

btnUndo.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        if(shapes.size()>0) {
            shapes.remove(0);
            repaint();
        }
    }
});

btnSize.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        String value = shapeSize.getText();
        size = Integer.valueOf(value);
    }
});

btnColor.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e) {
        String full = color.getText();
        int r = full.indexOf(',');
        int g = full.indexOf(',', r+1);
        int b = full.indexOf(',', g+1);

        b = Integer.valueOf(full.substring(g + 1, full.length()));
        g = Integer.valueOf(full.substring(r + 1, g));
        r = Integer.valueOf(full.substring(0, r));
        c = new Color(r, g, b);
    }

});

btnLocate.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e) {
        location.append("(" + (int)(x) + ", " + (int)(y) + ")");
    }

});

btnClear.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e) {
        shapeSize.setText("");
        color.setText("");
        location.setText("");
    }

});
```

```
}

class MsListener extends MouseAdapter{

    @Override
    public void mouseClicked(MouseEvent e) {
        shape(e);//Drawing the actual shapes
    }

}

public void shape(MouseEvent e) {
    x = e.getX();
    y = e.getY();

    switch (num){//Easily able to add new shapes
        case 1:
            drawTri(x,y);
            break;
        case 2:
            drawRec(x,y);
            break;
        case 3:
            drawCirc(x,y);
            break;
    }
}
```

```
    public void drawTri(double x, double y) {
        int[] xpoints = {(int)x, (int)x+(int)size, (int)x+((int)size/2)};
        int[] ypoints = {(int)y, (int)y, (int)y-20+(int)size};//constants for base look
        Graphics2D g2 = (Graphics2D) getGraphics().create();
        Shape tri = new Polygon(xpoints, ypoints, 3);
        shapes.add(tri);
        g2.draw(tri);
    }

    public void drawRec(double x, double y) {
        Graphics2D g2 = (Graphics2D) getGraphics().create();
        Shape rect = new Rectangle2D.Double(x, y, size + 15,size);
        g2.setColor(c);
        shapes.add(rect);
        g2.draw(rect);
    }

    public void drawCirc(double x, double y) {
        Graphics2D g2 = (Graphics2D) getGraphics().create();
        Shape oval = new Ellipse2D.Double(x, y, size,size);
        g2.setColor(c);
        shapes.add(oval);
        g2.draw(oval);
    }
}
```

```
class Info{
        String info;

        public String getInfo() {
                return info;
        }

        public void setInfo(String info) {
                this.info = info;
        }

}
```