```
// See, I've already imported all the things you'll need!
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.GridLayout;
import java.awt.Color;


import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Line2D;
import java.awt.geom.Point2D;
import javax.swing.JComponent;



public class SwingLab
{
    // frame properties
    private static final int FRAME_WIDTH = 400;
    private static final int FRAME_HEIGHT = 400;

    public static void main(String[] args)
    {

        // Instantiate a frame
        JFrame frame = new JFrame();

        // The buttons (one for each color)
        JButton bRed = new JButton("Red");
        JButton bYellow = new JButton("Yellow");
        JButton bBlue = new JButton("Blue");


        // Create a panel containing our buttons, and add the panel to
        // the frame.
        // The panel that holds the user interface components
        final JPanel container = new JPanel(new GridLayout(2,1));

        final JPanel panel = new JPanel(new GridLayout(1,1));
        final JPanel buttonPanel = new JPanel( new GridLayout(3,1) );

        Art artBox=new Art();
        panel.add(artBox);

        buttonPanel.add(bRed);
        buttonPanel.add(bYellow);
        buttonPanel.add(bBlue);

        container.add(panel);
        container.add(buttonPanel);

        frame.add(container);
```

```java
        // declare your listener classes and add them to the buttons
        // here.

        // you are going to call addActionListener and
        // addMouseListener for each button

        // you want to deal with the JPanel named 'panel' declared
        // above





        // Show the frame
        frame.setSize(FRAME_WIDTH, FRAME_HEIGHT);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }




    // The code below is to give advanced users something to play
    // with.  Clicking in the upper pane of the window will cause it
    // to redraw the lines, centering on the location you clicked.

    public static class Art extends JComponent
    {
        private int x;          // the center of the lines that are drawn
        private int y;

        public Art()
        {
            x=200;
            y=100;

            // this trivial MouseListener implementation handles
            // clicks inside the "art box"

            class ClickListener implements MouseListener
            {
                // we instantiate a ClickListener with an art object,
                // so we can update the center position when a mouse
                // event occurs

                public ClickListener(Art a)
                {
                    this.a = a;
                }

                public void mouseEntered(MouseEvent event) { }
                public void mouseExited(MouseEvent event) { }

                public void mousePressed(MouseEvent event)
                {
                    a.setPoint(event.getX(), event.getY());
                    a.repaint();
                }

                public void mouseReleased(MouseEvent event) { }

                public void mouseClicked(MouseEvent event) { }

                private Art a;
            }

            ClickListener cListener = new ClickListener(this);
            this.addMouseListener(cListener);
        }


        // this is the JComponent method that handles drawing the
        // lines
        public void paintComponent(Graphics g)
        {
            Graphics2D g2 = (Graphics2D) g;

            // Create the four lines
            Line2D.Double l1 = new Line2D.Double(0, 0, x, y);
            Line2D.Double l2 = new Line2D.Double(0, getHeight(), x, y);
            Line2D.Double l3 = new Line2D.Double(getWidth(), 0, x, y);
            Line2D.Double l4 = new Line2D.Double(getWidth(), getHeight(), x, y);

            // Draw the lines
            g2.draw(l1);
            g2.draw(l2);
            g2.draw(l3);
            g2.draw(l4);
        }

        public void setPoint(int x, int y)
        {
            this.x = x;
            this.y = y;
        }

    }
}
```