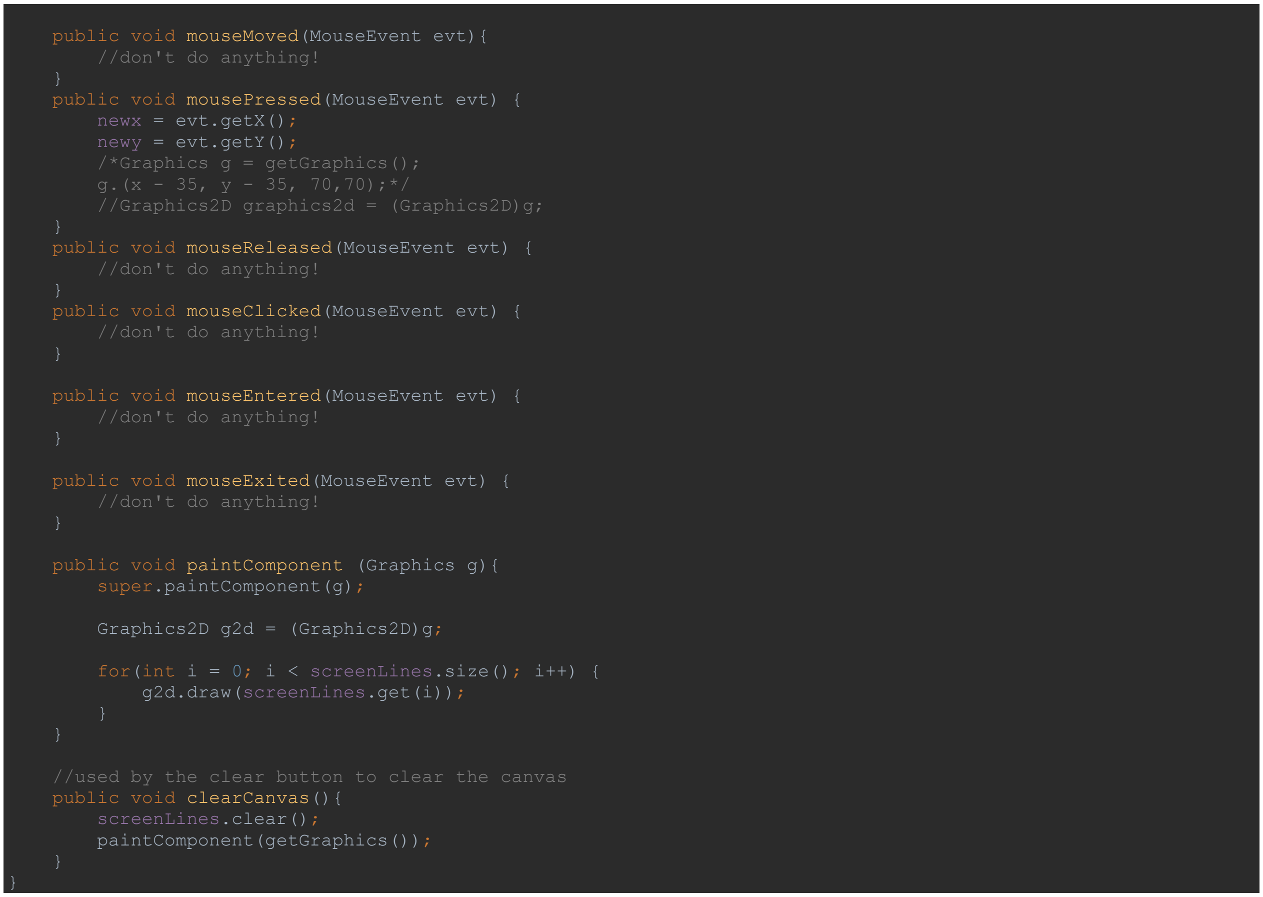
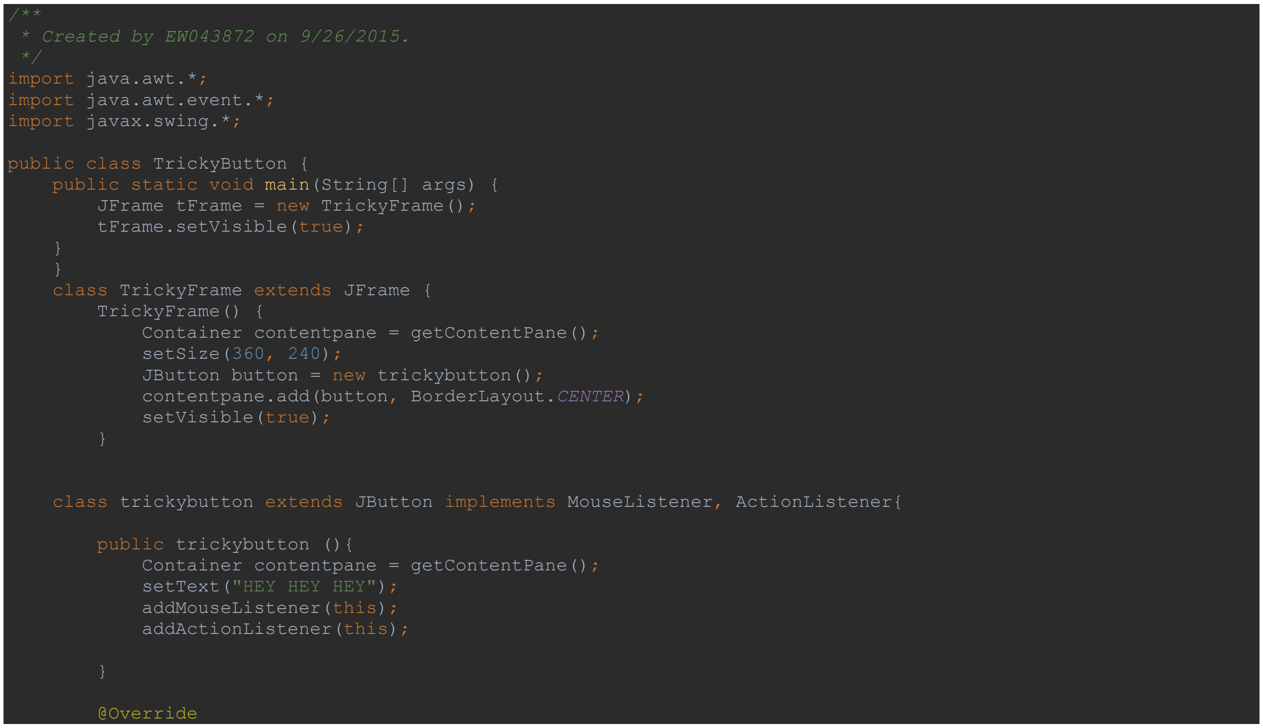
Part A

```java
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.Line2D;
import java.util.ArrayList;
import javax.swing.*;
import javax.swing.event.*;

/**
 * Created by EW043872 on 9/25/2015.
 */

public class EtchaSketch {
    public static void main(String[] args) {
        JFrame frame = new DrawFrame();
        frame.setVisible(true);
    }


}

class DrawFrame extends JFrame {
    public DrawFrame(){
        setTitle("Etch-a-Sketch");
        setSize(600, 400);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e){
                System.exit(0);
            }
        });
        //Make the ClearButton
        JButton button = new JButton("Shake it. Shake, shake, shake it.");
        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){

            }
        });

        Container contentPane = getContentPane();
        ECanvas EtchySketchy = new ECanvas();
        contentPane.add(EtchySketchy, BorderLayout.CENTER);
        contentPane.add(button, BorderLayout.SOUTH);

        //Clearing the canvas
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                EtchySketchy.clearCanvas();
            }
        });
    }

}



class ECanvas extends JPanel implements MouseMotionListener, MouseListener {

    private int oldx,oldy, newx, newy;
    ArrayList<Line2D> screenLines = new ArrayList<>();

    public ECanvas() {
        addMouseMotionListener(this);
        addMouseListener(this);
    }



    public void mouseDragged(MouseEvent evt){ //this is where the drawing happens
        oldx = newx;
        oldy = newy;
        newx = evt.getX();
        newy = evt.getY();
        Line2D newLine = new Line2D.Double(oldx, oldy, newx, newy);
        screenLines.add(newLine);
        Graphics g = getGraphics();
        Graphics2D g2d = (Graphics2D)g;
        g2d.draw(newLine);
    }
```

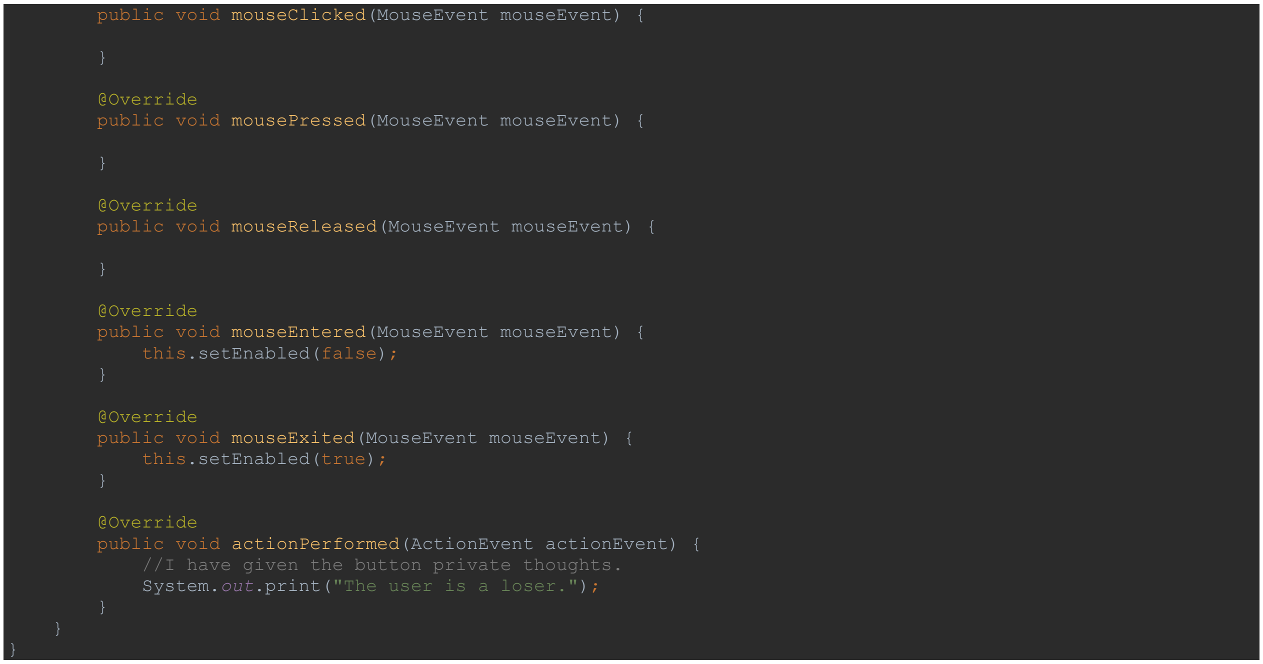


```
    public void mouseMoved(MouseEvent evt){
        //don't do anything!
    }
    public void mousePressed(MouseEvent evt) {
        newx = evt.getX();
        newy = evt.getY();
        /*Graphics g = getGraphics();
        g.(x - 35, y - 35, 70,70);*/
        //Graphics2D graphics2d = (Graphics2D)g;
    }
    public void mouseReleased(MouseEvent evt) {
        //don't do anything!
    }
    public void mouseClicked(MouseEvent evt) {
        //don't do anything!
    }

    public void mouseEntered(MouseEvent evt) {
        //don't do anything!
    }

    public void mouseExited(MouseEvent evt) {
        //don't do anything!
    }

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

        Graphics2D g2d = (Graphics2D)g;

        for(int i = 0; i < screenLines.size(); i++) {
            g2d.draw(screenLines.get(i));
        }
    }

    //used by the clear button to clear the canvas
    public void clearCanvas(){
        screenLines.clear();
        paintComponent(getGraphics());
    }
}
```

Part B



```
/**
 * Created by EW043872 on 9/26/2015.
 */
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class TrickyButton {
    public static void main(String[] args) {
        JFrame tFrame = new TrickyFrame();
        tFrame.setVisible(true);
    }
    }
    class TrickyFrame extends JFrame {
        TrickyFrame() {
            Container contentpane = getContentPane();
            setSize(360, 240);
            JButton button = new trickybutton();
            contentpane.add(button, BorderLayout.CENTER);
            setVisible(true);
        }


    class trickybutton extends JButton implements MouseListener, ActionListener{

        public trickybutton (){
            Container contentpane = getContentPane();
            setText("HEY HEY HEY");
            addMouseListener(this);
            addActionListener(this);

        }

        @Override
```



```
        public void mouseClicked(MouseEvent mouseEvent) {

        }

        @Override
        public void mousePressed(MouseEvent mouseEvent) {

        }

        @Override
        public void mouseReleased(MouseEvent mouseEvent) {

        }

        @Override
        public void mouseEntered(MouseEvent mouseEvent) {
            this.setEnabled(false);
        }

        @Override
        public void mouseExited(MouseEvent mouseEvent) {
            this.setEnabled(true);
        }

        @Override
        public void actionPerformed(ActionEvent actionEvent) {
            //I have given the button private thoughts.
            System.out.print("The user is a loser.");
        }
    }
}
```