Please note, I believe there was a typo in the practical description, it stated that Y was the distance from the left and X was the distance from the top. I think it should be the other way around so I’ve coded it that way.

Lee

package sketch;

/\*\*

\*

\* @author 09092543

\*/

import java.awt.Container;

import java.awt.Dimension;

import java.awt.Point;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.util.ArrayList;

import javax.swing.BoxLayout;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JButton;

import javax.swing.JTextField;

import javax.swing.JLabel;

public class Sketch extends JFrame implements MouseListener,MouseMotionListener,ActionListener {

private static final Dimension SKETCH\_SIZE = new Dimension(360, 360);

private JPanel controlPanel = new JPanel();

private JButton deleteButton = new JButton("Delete");

private JTextField xTextField = new JTextField("x");

private JTextField yTextField = new JTextField("y");

ArrayList<Point> points = new ArrayList<Point>();

SketchPanel sketchPanel = new SketchPanel(points);

public Sketch() {

super("Sketch");

Container contentPane = getContentPane();

contentPane.setLayout(new BoxLayout(contentPane, BoxLayout.Y\_AXIS));

sketchPanel.addMouseListener(this);

sketchPanel.addMouseMotionListener(this);

sketchPanel.setPreferredSize(SKETCH\_SIZE);

contentPane.add(sketchPanel);

controlPanel.setLayout(new BoxLayout(controlPanel, BoxLayout.X\_AXIS));

controlPanel.add(deleteButton);

controlPanel.add(xTextField);

controlPanel.add(yTextField);

xTextField.setEditable(false);

yTextField.setEditable(false);

contentPane.add(controlPanel);

deleteButton.addActionListener(this);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

pack();

setResizable(false);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == deleteButton) {

int i = points.size();

if(i > 0){

points.remove(i-1);

repaint();

}

}

}

public void mouseClicked(MouseEvent e) {

points.add(e.getPoint());

repaint();

}

public void mousePressed(MouseEvent e) {

}

public void mouseReleased(MouseEvent e) {

}

public void mouseEntered(MouseEvent e) {

}

public void mouseExited(MouseEvent e) {

}

public void mouseMoved(MouseEvent e) {

int x = e.getX();

int y = e.getY();

xTextField.setText("X: "+x);

yTextField.setText("Y: "+y);

repaint();

}

public void mouseDragged(MouseEvent e) {

}

public static void main(String[] args) {

new Sketch();

}

}

package sketch;

/\*\*

\*

\* @author 09092543

\*/

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Point;

import java.util.List;

import javax.swing.JPanel;

public class SketchPanel extends JPanel {

private static final int DIAMETER = 16;

private List<Point> points;

public SketchPanel(List<Point> points) {

this.points = points;

}

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(Color.RED);

int size = points.size();

int[] x = new int[size];

int[] y = new int[size];

int i = 0;

for (Point p : points) {

x[i] = p.x;

y[i] = p.y;

i++;

}

g.drawPolyline(x, y, i);

if (size > 0) {

g.fillOval(x[size - 1] - DIAMETER / 2, y[size - 1] - DIAMETER / 2,

DIAMETER, DIAMETER);

}

}

}