import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.awt.image.BufferedImage;

import java.io.\*;

import java.util.\*;

import java.awt.geom.Point2D;

import javax.swing.JColorChooser;

import javax.swing.JOptionPane;

import java.applet.\*;

public class paint2 extends JApplet{

static int mode1=0;

static int flag=0;

static Color fgc,bgc;

public void init(){

//JFrame frame = new JFrame("Paint It");

this.setSize(900,900);

//Container content = frame.getContentPane();

JPanel content=new JPanel();

content.setLayout(new BorderLayout());

final PadDraw drawPad = new PadDraw();

content.add(drawPad, BorderLayout.CENTER);

JPanel panel = new JPanel();

JPanel p1=new JPanel();

//creates a JPanel

panel.setPreferredSize(new Dimension(32,68));

panel.setMinimumSize(new Dimension(32,68));

panel.setMaximumSize(new Dimension(32,68));

//This sets the size of the panel

JButton clearButton = new JButton("Clear");

//creates the clear button and sets the text as "Clear"

clearButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.clear();

}

});

JButton textbut =new JButton("text");

textbut.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

flag=2;

drawPad.textfunc(flag);

}

});

JButton redButton = new JButton("Red");

redButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.red();

}

});

JButton blackButton = new JButton("Black");

blackButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.black();

}

});

JButton magentaButton = new JButton("Pink");

magentaButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.magenta();

}

});

JButton blueButton = new JButton("Blue");

blueButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.blue();

}

});

JButton eButton = new JButton("Eraser");

eButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.erase();

}

});

JButton greenButton = new JButton("Green");

//green button

greenButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

drawPad.green();

}

});

JButton ovalbutton = new JButton("Rectangle");

//green button

ovalbutton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

mode1=1;

drawPad.oval(mode1);

}

});

JButton linebut = new JButton("Line");

//green button

linebut.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

mode1=0;

drawPad.oval(mode1);

}

});

JButton circlebut = new JButton("Circle");

//green button

circlebut.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

mode1=2;

drawPad.oval(mode1);

}

});

JRadioButton fg=new JRadioButton("Foreground"); final JLabel fglabel=new JLabel(" "); fglabel.setOpaque(true); fglabel.setBackground(Color.BLACK);

JRadioButton bg=new JRadioButton("Background"); final JLabel bglabel=new JLabel(" "); bglabel.setOpaque(true); bglabel.setBackground(Color.BLACK);

//fg.setSelected(true);

ButtonGroup bgrp=new ButtonGroup(); bgrp.add(fg); bgrp.add(bg);

bg.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

//JColorChooser cc = new JColorChooser();

//int showConfirmDialog = JOptionPane.showConfirmDialog(this,cc);

//if(showConfirmDialog==JOptionPane.YES\_OPTION){

bgc=Color.black;

bgc=JColorChooser.showDialog(null,"Select a colour",bgc);

//}

flag=1;

drawPad.getcolorb(bgc,flag);

drawPad.setBackground(bgc);

}

});

fg.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

//JColorChooser cc = new JColorChooser();

//int showConfirmDialog = JOptionPane.showConfirmDialog(this,cc);

//if(showConfirmDialog==JOptionPane.YES\_OPTION){

fgc=Color.black;

fgc=JColorChooser.showDialog(null,"Select a colour",fgc);

flag=0;

drawPad.getcolorf(fgc,flag);

//}

}

});

/\*blackButton.setPreferredSize(new Dimension(16, 16));

magentaButton.setPreferredSize(new Dimension(16, 16));

//redButton.setPreferredSize(new Dimension(16, 16));

blueButton.setPreferredSize(new Dimension(16, 16));

greenButton.setPreferredSize(new Dimension(16,16));

//sets the sizes of the buttons\*/

panel.add(greenButton);

panel.add(blueButton);

panel.add(eButton);

panel.add(magentaButton);

panel.add(blackButton);

panel.add(redButton);

panel.add(clearButton);

panel.add(linebut);

panel.add(circlebut);

panel.add(textbut);

p1.add(fg);

p1.add(bg);

panel.add(ovalbutton);

//adds the buttons to the panel

content.add(panel, BorderLayout.NORTH);

content.add(p1,BorderLayout.WEST);

add(content);

content.setVisible(true);

setVisible(true);

//sets the panel to the left

//frame.setSize(700, 700);

//sets the size of the frame

//frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//makes it so you can close

//frame.setVisible(true);

//makes it so you can see it

}

}

class PadDraw extends JComponent

{

Image image;

Point dot[]=new Point[1000];

int nod=0;

int mode=0;

int flag=0;

int x,y,rad;

Color fgp,bgp;

//this is gonna be your image that you draw on

Graphics2D graphics2D;

//this is what we'll be using to draw on

int currentX, currentY, oldX, oldY;

private Color bgcolor,color;

//these are gonna hold our mouse coordinates

private BufferedImage \_bufImage = null;

//Now for the constructors

public PadDraw(){

//setDoubleBuffered(false);

repaint();

addMouseListener(new MouseAdapter(){

public void mousePressed(MouseEvent e){

oldX = e.getX();

oldY = e.getY();

}

});

//if the mouse is pressed it sets the oldX & oldY

//coordinates as the mouses x & y coordinates

addMouseMotionListener(new MouseMotionAdapter(){

public void mouseDragged(MouseEvent e){

System.out.println("mode in mouse:" +mode);

//currentX = e.getX();

//currentY = e.getY();

dot[nod++]=new Point(e.getX(),e.getY());

if(mode==1)

{

System.out.println("hello");

if(graphics2D != null)

if(flag==0)

{

graphics2D.drawRect(oldX, oldY,currentX-oldX, currentY-oldY);

}

if(flag==1)

{

graphics2D.fillRect(oldX, oldY,currentX-oldX, currentY-oldY);

}

repaint();

}

else if(mode==0)

{

if(graphics2D != null)

graphics2D.drawLine(oldX, oldY, currentX, currentY);

oldX = currentX;

oldY = currentY;

repaint();

}

else if(mode==2)

{

if(graphics2D != null)

x=(oldX+currentX)/2;

y=(oldY+currentY)/2;

rad=(int)Point2D.distance(oldX, oldY, currentX, currentY);

rad=rad/2;

System.out.println("in drawcirc");

if(flag==1)

{

graphics2D.fillOval(x-rad, y-rad,rad\*2,rad\*2);}

if(flag==0)

graphics2D.drawOval(x-rad, y-rad,rad\*2,rad\*2);

repaint();

}

}

});

//while the mouse is dragged it sets currentX & currentY as the mouses x and y

//then it draws a line at the coordinates

//it repaints it and sets oldX and oldY as currentX and currentY

addMouseListener(new MouseAdapter(){

public void mouseReleased(MouseEvent e) {

// This will save the shape that has been dragged by

// drawing it onto the bufferedImage where all shapes

// are written.

currentX = e.getX(); // save ending coordinates

currentY = e.getY();

//--- Draw the current shape onto the buffered image.

//Graphics2D grafarea = \_bufImage.createGraphics();

//drawCurrentShape(grafarea);

if(flag==2)

{

JTextArea ta=new JTextArea();

JPanel area = new JPanel();

area.add(ta);

ta.setBounds(oldX,oldY,currentX-oldX,currentY-oldY);

ta.setColumns(currentX-oldX);

ta.setLineWrap(true);

}

repaint();}

});

}

public void paintComponent(Graphics g){

if(image == null){

image = createImage(getSize().width, getSize().height);

graphics2D = (Graphics2D)image.getGraphics();

graphics2D.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

clear();

}

g.drawImage(image, 0, 0, null);

}

/\*public void setForeGroundColor(Color fg)

{

color=fg;

}\*/

//this is the painting bit

//if it has nothing on it then

//it creates an image the size of the window

//sets the value of Graphics as the image

//sets the rendering

//runs the clear() method

//then it draws the image

public void clear(){

graphics2D.setPaint(Color.white);

graphics2D.fillRect(0, 0, getSize().width, getSize().height);

graphics2D.setPaint(Color.black);

repaint();

}

//this is the clear

//it sets the colors as white

//then it fills the window with white

//thin it sets the color back to black

public void red(){

graphics2D.setPaint(Color.red);

repaint();

}

//this is the red paint

public void black(){

graphics2D.setPaint(Color.black);

repaint();

}

//black paint

public void magenta(){

graphics2D.setPaint(Color.magenta);

repaint();

}

//magenta paint

public void blue(){

graphics2D.setPaint(Color.blue);

repaint();

}

//blue paint

public void green(){

graphics2D.setPaint(Color.green);

repaint();

}

public void erase(){

//graphics2D.setPaint(Color.white);

//dot[nod++]=new Point(evt.getX(),evt.getY());

//Graphics2D g2=(Graphics2D)area.getGraphics();

graphics2D.setPaint(Color.white);

int i=0;

for(i=0;i<nod-1;i++)

graphics2D.fillOval(dot[i].x, dot[i].y, 10,10);

}

public void oval(int mode)

{

//x=(oldX+currentX)/2;

//y=(oldY+currentY)/2;

this.mode=mode;

System.out.println("in rectangle");

//Graphics2D g2=(Graphics2D)jPanel1.getGraphics();

//graphics2D.setColor(Color.BLACK);

System.out.println("oldx:" +oldX);

System.out.println("mode in rec:" +mode);

}

public void getcolorf(Color fgp,int flag)

{

this.fgp=fgp;

this.flag=flag;

graphics2D.setColor(fgp);

}

public void getcolorb(Color bgp,int flag)

{

this.bgp=bgp;

this.flag=flag;

graphics2D.setColor(bgp);

//graphics2D.setPaint(bgp);

//graphics2D.fillRect(0, 0, getSize().width, getSize().height);

//green paint

}

public void textfunc(int flag)

{

this.flag=flag;

}}