|  |
| --- |
| import javax.swing.event.\*;  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  import javax.swing.border.\*;  import java.util.\*;  public class GUI implements ActionListener, KeyListener  {  JFrame f;    //Rect Coordinates  public static int x, y;    //diameter of the circle  public static final int diameter = 8;    //Width and height of the screen  public static int width = 750, height = 600;    //Color of the current dot  Color dotColor = Color.RED;  //Color of all the previous dot circle  Color lineColor = Color.GREEN;    JPanel mainPanel, paintPanel, btnPanel;  JButton btnLine, btnDot, btnClear;    PaintClass paint;  Coord coord;    ArrayList<Coord> coordArray;    public GUI()  {  f = new JFrame("PAINT");    //initial coordinates  x = width / 2;  y = height / 2;    paint = new PaintClass();    btnPanel = new JPanel(new FlowLayout());    btnLine = new JButton("Set Line Color");  btnDot = new JButton("Set Dot Color");  btnClear = new JButton("CLEAR");    //Add the  coordArray = new ArrayList<Coord>();    mainPanel = new JPanel(new BorderLayout());  mainPanel.setPreferredSize(new Dimension(width, height));  mainPanel.setBorder(new EtchedBorder(EtchedBorder.LOWERED));    paint.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));    btnLine.addActionListener(this);  btnDot.addActionListener(this);  btnClear.addActionListener(this);    //paint.addKeyListener(this);  f.addKeyListener(this);    btnPanel.add(btnDot);  btnPanel.add(btnLine);  btnPanel.add(btnClear);    mainPanel.add(paint, BorderLayout.CENTER);  mainPanel.add(btnPanel, BorderLayout.SOUTH);    //Disable KeyListener on buttons  btnDot.setFocusable(false);  btnLine.setFocusable(false);  btnClear.setFocusable(false);    //Request to Focus KeyListener to the paint Panel and the JFrame  //paint.requestFocusInWindow();  f.requestFocusInWindow();    f.add(mainPanel);    f.pack();  f.setLocationRelativeTo(null);  f.setVisible(true);    }    public void keyTyped(KeyEvent e){}  public void keyReleased(KeyEvent e){}    /\*\*  \* Move the rectangle when the arrow keys are pressed  \*/  public void keyPressed(KeyEvent e)  {  int keyCode = e.getKeyCode();    if(keyCode == 37) //LEFT  x -= 3;    else if(keyCode == 38) //UP  y -= 3;    else if(keyCode == 39) //RIGHT  x += 3;    else if(keyCode == 40) //DOWN  y += 3;    //repaint the screen  if(keyCode >=37 && keyCode <=40)  paint.repaint();  }//end keyPressed(KeyEvent)    public void actionPerformed(ActionEvent e)  {    //Draw a line on the screen  if(e.getSource() == btnDot)  {  //Choose a color the current circle  dotColor = JColorChooser.showDialog(f, "Choose Dot Color", dotColor);    //Default dot color  if(dotColor == null)  dotColor = Color.BLACK;    }    //Draw a rectangle on the screen  else if(e.getSource() == btnLine)  {  //Choose a color for the preious circles line  lineColor = JColorChooser.showDialog(f, "Choose Line Color", lineColor);    //Default line color  if(lineColor == null)  lineColor = Color.GREEN;    }    //Reset all the previous and current shapes on the paint screen  else if(e.getSource() == btnClear)  {  coordArray.clear();    }//end if    paint.repaint();    //paint.requestFocusInWindow();  f.requestFocusInWindow();  }    //Main  public static void main(String[] args)  {  new GUI();  }    class PaintClass extends JPanel  {    public void paintComponent(Graphics g)  {  super.paintComponent(g);    //Set the background to white color  g.setColor(Color.BLACK);  g.fillRect(0, 0, getWidth(), getHeight());    //Draw the previous circles  for(Coord oldCircle : coordArray)  circle(g, oldCircle.getX(), oldCircle.getY(), lineColor);    //Draw the current circle  circle(g, (int)x, (int)y, dotColor);    //Add the coordinates to the ArrayList of Class Coord  coord = new Coord((int)x, (int)y);  coordArray.add(coord);    }//end paintComponent(Graphics)    //Draw Circle  public void circle(Graphics g, int x, int y, Color color)  {  g.setColor(color);  g.fillOval(x - diameter/2, y - diameter/2, diameter, diameter);  g.drawOval(x - diameter/2, y - diameter/2, diameter, diameter);  }//end circle(Graphics, int, int, int, int)    }    }  public class Coord {    private int x, y;    public Coord(int x, int y)  {  this.x = x;  this.y = y;  }    //Get the x coordinate of the previous circle  public int getX()  {  return this.x;  }    //Get the y coordinate of the previous circle  public int getY()  {  return this.y;  }    public String toString()  {  return null;  }  } |
| Screenshots: |