الاسم : ود عابد فران الرقم الجامعي: 439016210 المشرف :مرتضى علي خان خان

Name : Wed Abed Farran ID: 439016210 constructor : Murtaza Ali Khan Khan

Brief description :

It's a simple game that have a moving polygon by the user and a randomly moving rectangls as the constructor require , for the interface I used apach netbeans IDE 12.0 , for the implementation Issues I actually had lots of issues trying to move the rectangules in random directions I tried to use Random but it didn't work and coudlnt implement the 6th riquier ,the other isssues are the lack of info in the internet about this course

```
package project;
import javax.swing.*;
import java.awt.*;
import java.awt.geom.*;
import java.awt.event.*;
import static java.lang.Math.*;
import java.util.*;
import java.util.Random;
/**
 *
 * @author wedfa
 */
public class Project extends JFrame{
   private static final int x=800; //x axis
   private static final int y=800; //y axis

   //constructor
   public Project () {
      setLayout(new BorderLayout()); //frame set
      setSize(x,y); //frame width and hight
```

```
        setTitle("my first game project");
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        PanelRectangleAnimation panel = new PanelRectangleAnimation();
        add(panel,BorderLayout.CENTER);
        pack();
        setVisible(true);
}



    public static void main(String[] args) {
     Project p=  new Project();
    }
}
class PanelRectangleAnimation extends JPanel implements Runnable{

            private static int x=800;  //panel_width
            private static int y=800; //panel_hight
        private static final int pW = 80;  //polygon width
        private static final int pH = 100; //polygon height

        // polygon coordinats
    private static int x0 = 180;
    private static int y0 = 600;
    private static int x1 = x0 + pW;
    private static int y1 = y0;
    private static int x2 = x0 + (x1 - x0) / 2;
```

```
private static int y2 = y0 - pH;

private static int xP[] = {x0, x1, x2}; //  array of polygon x axis point
private static int yP[] = {y0, y1, y2}; // array of polygon y axis point

private static int tx = 10;     // tranlsation distances along x-axis
private static int ty = tx;

private static int [] xaxis = new int[100]; // x arrays to store polygon coordinate
private static int [] yaxis = new int[100]; // y arrays to store polygon coordinate
private static int arraycounter=0;

private static final int RW = x / 15; // rectangle width
private static final int RH = RW;  // rectangle height

private static final int H = RW / 2; //spaces between rectangels
private static final int V = H;

//private static final int H =random.(0,750);
//private static final int V =random.(0,750);


private static final int NO_OF_RECT = 5; //number og rec
Rectangle rectAry[] = new Rectangle[NO_OF_RECT];

Thread mythread;
int vlex=5;    //to increament the H and V to move the rec
int xa=H,ya=V;
```

```
public PanelRectangleAnimation()      {
          setLayout(new BorderLayout());
                    setPreferredSize(new Dimension(x,y));
                    this.setBackground(Color.black);
          this.setFocusable(true);
          KeyPressListener KL = new KeyPressListener();
          addKeyListener(KL);
          mythread = new Thread(this);
         mythread.start();
        }

public void run() {
   boolean flag=true;
   while (flag)
   {
   try{
   Thread.sleep(500);
   }
   catch(InterruptedException e){}
        xa = xa + vlex; // to move the rec by 5
        ya = ya + vlex;
        if ( xa==746 || ya==746||xa<0||ya<0) {
            vlex=-vlex; // to bounced back in the opposite direction.it dosent work!!


        }


        repaint();
   }
}
```

```
        public void paintComponent(Graphics g) {

        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;
        g2.setStroke(new BasicStroke(5.0f));
        Polygon poly = new Polygon(xP, yP, xP.length);
        g2.setColor(Color.blue);
        g2.fill(poly);

////////////////////////////////
        int q=H;
        int p=V;
        int cc = 0;
        g2.setColor(Color.green);

        for (int j = 0; j < NO_OF_RECT; j++) {
           rectAry[cc] = new Rectangle(xa,ya, RW, RH);
           g2.fill(rectAry[cc]);
           cc= cc + 1;
           xa = xa  +RW+ H;
        }
        ////////////////////////
        for(int i=0;i<xaxis.length;i++)
        {
        g2.setColor(Color.white);
        g2.drawLine(xaxis[i], yaxis[i], xaxis[i], yaxis[i]);
        }
}
```

```
    private class KeyPressListener extends KeyAdapter {

    public void keyPressed(KeyEvent e) {

       int keyCode = e.getKeyCode();

       switch (keyCode) {
          case KeyEvent.VK_LEFT:
             moveLeft();
             break;
          case KeyEvent.VK_RIGHT:
             moveRight();
             break;
          case KeyEvent.VK_UP:
             moveUp();
             break;
          case KeyEvent.VK_DOWN:
             moveDown();
             break;
       }
       if(keyCode ==KeyEvent.VK_LEFT||keyCode ==KeyEvent.VK_RIGHT||keyCode
==KeyEvent.VK_UP||keyCode ==KeyEvent.VK_DOWN)
       {
       xaxis[arraycounter]=xP[2];
       yaxis[arraycounter]=yP[2];
       arraycounter++;
       if(arraycounter==99)
```

```
        {
        arraycounter=0;
        }
        }
        repaint();
    }

}
      public void moveUp() {
    for (int i = 0; i < yP.length; i++) {
        yP[i] = yP[i] - ty;
    }
}

public void moveDown() {
    for (int i = 0; i < yP.length; i++) {
        yP[i] = yP[i] + ty;
    }
}

public void moveRight() {
    for (int i = 0; i < xP.length; i++) {
        xP[i] = xP[i] + tx;
    }
}

public void moveLeft() {
    for (int i = 0; i < xP.length; i++) {
        xP[i] = xP[i] - tx;
```

```
        }
    }
}
```

Screenshoots :