Ahmet Yurdal

186035

Software Engineering

Java Brick Game

Gui

Ball–I create the ball as white it move well and intersection is good . I couldn't add as image

Paddle – I create it and it move well, intersection is good with ball and i made it blue I could add image

I creates bricks ass yellow with red borders

Action

- Paddle moving
- Ball automatic moving when it start
- When the bricks get hit by ball they get lost from the screen

I create 3 class

- Mainclass =main

- Play =where defines ball, paddle, background ,borders ,game logic ,and other functions ball move paddle move
- ObjAction= I create 2d graphics here (bricks)

- The ball cant move outside from left ,right or bottom
- The paddle cant move outside from left and right
- When ball is fall from buttom the game will end and game over
- Program calculate the score and it show the user at the right top and the score will shown to user at the end of the game
- I used keylistener to get right click left click and enter from the user and move paddle to touch ball

Code

Mainclass.java

```
import javax.swing.*;


public class mainclass {

    public static void main(String[] args) {


        // define JFrame object
         JFrame obje=new JFrame();
         //setting bounds
         obje.setBounds(15,15,710,600);
        // to give name to game
         obje.setTitle("JAVA Brick Game ");
         play gameply=new play();
         //
        //connecting each other
```

```
        obje.add(gameply);
         obje.setResizable(false);
         //
        // to make panel visit
         obje.setVisible(true);
        //program will close when user click close button
         obje.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

}
```

Play.java

```
//importing functions
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.image.renderable.*;
import java.awt.print.*;
import java.io.*;
import javax.swing.Timer;
import javax.swing.JPanel;


public class play extends JPanel implements KeyListener,ActionListener{ //
//defining positions,brickssize,balldirection,
    private boolean play=false;
    //setting score
    private int point=0;
    private Timer Ttimer; //
    private int TDelay=8; //
    private int XPlayer=310; //player position
    private int posballX=120;//position of ball (x)
    //position of ball (y)
```

```
    private int posballY=350;
    private int Xdirball=-1;//X direction of ball
    //direction ball Y
    private int Ydirball=-2;
    private int Bricks=21; //brick size
    //
    private ObjAction GMap;

    //paint function to graphics
    public void paint(Graphics graph) //in this class i create back ground-
drawmap-borders,ball and paddle
    {
        //background
        graph.setColor(Color.BLACK); //setting background color as black
        //position of background
        graph.fillRect(1, 1, 700, 600);

        //drawing map
        GMap.draw((Graphics2D)graph);

        //borders
        graph.setColor(Color.red); //setting border coolor
        //left-right and top
        graph.fillRect(0, 0, 3, 592);
        graph.fillRect(0, 0, 692, 3);
        graph.fillRect(691, 0, 3, 592);

        //scores
        graph.setColor(Color.RED); //score color
        //font name,font.type,and sizeoftext
        graph.setFont(new Font("serif",Font.BOLD,30));
        //to print point of user
        graph.drawString(""+point, 580, 45);

        //the paddle
        graph.setColor(Color.blue);//color of paddle
        //where it will shown
        graph.fillRect(XPlayer, 550, 100, 8);

        //the ball
        graph.setColor(Color.white); //ball color
        //positions
        graph.fillOval(posballX, posballY, 25, 25);


        if(posballY>570)
        {
            // in this statement if user missing the ball goes down and game
will stop
```

```
            play=false;
            Xdirball=0;
            Ydirball=0;
            // and will print the screen game over to user with score which
user get
            graph.setColor(Color.RED);
            graph.setFont(new Font("serif",Font.BOLD,35));
            graph.drawString("Game Over,Scores : "+point, 190, 300);


            //in there it will print buttom of game over (Press Enter to
restart)
            graph.setFont(new Font("serif",Font.BOLD,35));
            graph.drawString("Press Enter to restart  ", 230, 350);
        }
        if(Bricks<=0)
        {

            //when bricks finish game will finish and print to user you won
with press enter to start
            play=false;
            Xdirball=0;
            Ydirball=0;
            graph.setColor(Color.green);
            graph.setFont(new Font("serif",Font.BOLD,35));
            graph.drawString("You Won : "+point, 260, 300);

            graph.setFont(new Font("serif",Font.BOLD,35));
            graph.drawString("Press Enter to restart  ", 230, 350);
        }
        graph.dispose();
    }

    public play()
    {

        //play function (constructor) it will start the game
        GMap=new ObjAction(3,7);
        //keylistener
        addKeyListener(this);
        //
        setFocusable(true);
        //
        setFocusTraversalKeysEnabled(false);
        //
        Ttimer =new Timer(TDelay,this);
        //
        Ttimer.start();
```

```
    }
    @Override
    public void actionPerformed(ActionEvent e) { //here where action start to
work
        Ttimer.start();
        if(play==false) {
            if(new Rectangle(posballX,posballY,20,20).intersects(new
Rectangle(XPlayer,550,100,8)))
            { Ydirball=+Ydirball;   }
        }
        else if (play==true)
        {
            if(new Rectangle(posballX,posballY,20,20).intersects(new
Rectangle(XPlayer,550,100,8)))
            { Ydirball=-Ydirball;   }
            A:for(int i=0;i<GMap.map.length;i++)
            { //
                for(int j=0;j<GMap.map[0].length;j++)
                {
                    if(GMap.map[i][j]>0)
                    {
                        int brickY=i*GMap.HeightBrick+50;
                        int brickX=j*GMap.WidthBrick+80;

                        int brickHeight=GMap.HeightBrick;
                        int brickWidth=GMap.WidthBrick;


                        Rectangle rect=new
Rectangle(brickX,brickY,brickWidth,brickHeight);
                        Rectangle ballRect=new
Rectangle(posballX,posballY,20,20);
                        Rectangle brickRect=rect;

                        if(ballRect.intersects(brickRect))
                        {

                            GMap.setBrickValue(0, i, j);
                            point+=1;
                            Bricks--;


                            if(posballX+19<=brickRect.x||posballX+1>=brickRect
.x+brickRect.width)
                            {
                                Xdirball=-Xdirball;

                            }
                            else {
```

```
                            Ydirball=-Ydirball;

                        }
                        break A;
                    }

                }
            }
        }
        posballX+=Xdirball;
        posballY+=Ydirball;
        //to keep ball inside except buttom side
        if(posballX>670)
        {
            Xdirball=-Xdirball;
        }


        if(posballX<0 )
        {
            Xdirball=-Xdirball;
        }
        if(posballY<0)
        {
            Ydirball=-Ydirball;
        }
    }
    repaint();

    ///

}

@Override




public void keyPressed(KeyEvent e)
//key press function
{
    //to go left (paddle) with user left click input
    if(e.getKeyCode() ==KeyEvent.VK_LEFT) {
        if(XPlayer<10)
        {
            XPlayer=10;
        }
        else {
```

```
                moveLeft();
            }
        }
        //to go  restart game again when you click enter after game is
        //finish game will restart with this function
        if(e.getKeyCode()==KeyEvent.VK_ENTER)
        {
            if(!play)
            {
                //same positions and variable at the beginning
                play=true;
                //score

                point=0;
                //bricks
                Bricks=21;

                Xdirball=-1;
                Ydirball=-2;
                XPlayer=310;
                posballX=120;
                posballY=350;
                GMap=new ObjAction(3,7);

                repaint();
            }
        }
        ////to go right (paddle) with user right click input
        if(e.getKeyCode()==KeyEvent.VK_RIGHT)
        {
            if(XPlayer>=600)
            {
                XPlayer=600;
            }
            else {
                moveRight();
            }
        }



    }
    //it defines how much paddle goes right
    public void moveRight()
    {
        play=true;
        XPlayer+=50;

    }
```

```
    //it defines how much paddle goes left
    public void moveLeft()
    {
        play=true;
        XPlayer-=50;
    }
    @Override
    //no need to use in this program
    public void keyReleased(KeyEvent e) {
        // TODO Auto-generated method stub

    }
    //no need to use in this program
    @Override
    public void keyTyped(KeyEvent e) {
        // TODO Auto-generated method stub

    }

}
```

ObjAction.java

```
import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.Graphics2D;


public class ObjAction {

    public int WidthBrick;
    public int HeightBrick;
    public int map[][];

    public ObjAction(int row,int col)
    {
        map=new int[row][col];
        for(int i=0;i<map.length;i++)
        {
            for(int j=0;j<map[0].length;j++)
            {
```

```
                map[i][j]=1;
            }
        }
        WidthBrick=570/col;
        HeightBrick=170/row;

    }
    public void draw(Graphics2D g)
    {
        for(int i=0;i<map.length;i++)
        {
            for(int j=0;j<map[0].length;j++)
            {
                if(map[i][j]>0) {
                g.setColor(Color.yellow);
                g.fillRect(j*WidthBrick+80,i*HeightBrick+50 , WidthBrick,
HeightBrick);

                g.setStroke(new BasicStroke(3));
                g.setColor(Color.red);
                g.drawRect(j*WidthBrick+80, i*HeightBrick+50, WidthBrick,
HeightBrick);
                }}
}
    }
public void setBrickValue(int value,int row,int col)
{
    map[row][col]=value;

}
}
```