Project name :Brick Breaker Game

Code :

```
package Project;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
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 javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class BrickBreaker extends JPanel implements
KeyListener,ActionListener, Runnable {
  static boolean right = false;
  static boolean left = false;
  int ballx = 160;
  int bally = 218;
  int batx = 160;
  int baty = 245;
  int brickx = 70;
  int bricky = 50;

  int brickBreadth = 30;
  int brickHeight = 20;
  Rectangle Ball = new Rectangle(ballx, bally, 5, 5);
  Rectangle Bat = new Rectangle(batx, baty, 40, 5);
 // Rectangle Brick;// = new Rectangle(brickx, bricky, 30, 10);
  Rectangle[] Brick = new Rectangle[12];

  int movex = -1;
  int movey = -1;
  boolean ballFallDown = false;
  boolean bricksOver = false;
  int count = 0;
  String status;
```

```
 BrickBreaker() {

}

 public static void main(String[] args) {
 JFrame frame = new JFrame();
 BrickBreaker game = new BrickBreaker();
 JButton button = new JButton("restart");
 frame.setSize(350, 450);
 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

 frame.add(game);
 frame.add(button, BorderLayout.SOUTH);
 frame.setLocationRelativeTo(null);
 frame.setResizable(false);
 frame.setVisible(true);
 frame.setTitle("Brick Breaker");
 button.addActionListener(game);

 game.addKeyListener(game);
 game.setFocusable(true);
 Thread t = new Thread(game);
 t.start();
}

public void paint(Graphics g) {
 g.setColor(Color.LIGHT_GRAY);
 g.fillRect(0, 0, 350, 450);
 g.setColor(Color.blue);
 g.fillOval(Ball.x, Ball.y, Ball.width, Ball.height);
 g.setColor(Color.green);
 g.fill3DRect(Bat.x, Bat.y, Bat.width, Bat.height, true);
 g.setColor(Color.GRAY);
 g.fillRect(0, 251, 450, 200);
 g.setColor(Color.red);
 g.drawRect(0, 0, 343, 250);
 for (int i = 0; i < Brick.length; i++) {
  if (Brick[i] != null) {
   g.fill3DRect(Brick[i].x, Brick[i].y, Brick[i].width,
     Brick[i].height, true);
  }
 }
```

```
 if (ballFallDown == true || bricksOver == true) {
  Font f = new Font("Arial", Font.BOLD, 20);
  g.setFont(f);
  g.drawString(status, 70, 120);
  ballFallDown = false;
  bricksOver = false;
 }
}

public void run() {
    createBricks();
//ballFallDown == false && bricksOver == false
 while (true) {
//   if(gameOver == true){return;}
  for (int i = 0; i < Brick.length; i++) {
   if (Brick[i] != null) {
    if (Brick[i].intersects(Ball)) {
     Brick[i] = null;
     // movex = -movex;
     movey = -movey;
     count++;
    }// end of 2nd if..
   }// end of 1st if..
  }// end of for loop..

  if (count == Brick.length) {// check if ball hits all bricks
   bricksOver = true;
   status = "YOU WON THE GAME";
   repaint();
  }

  repaint();
  Ball.x += movex;
  Ball.y += movey;

  if (left == true) {

   Bat.x -= 3;
   right = false;
  }
  if (right == true) {
   Bat.x += 3;
```

```
    left = false;
   }
   if (Bat.x <= 4) {
    Bat.x = 4;
   } else if (Bat.x >= 298) {
    Bat.x = 298;
   }
   // /===== Ball reverses when strikes the bat
   if (Ball.intersects(Bat)) {
    movey = -movey;
    // if(Ball.y + Ball.width >=Bat.y)
   }

   // ....ball reverses when touches left and right boundary
   if (Ball.x <= 0 || Ball.x + Ball.height >= 343) {
    movex = -movex;
   }// if ends here
   if (Ball.y <= 0) {// ////////////////|| bally + Ball.height >= 250
    movey = -movey;
   }// if ends here.....
   if (Ball.y >= 250) {
    ballFallDown = true;
    status = "YOU LOST THE GAME";
    repaint();
//    System.out.print("game");
   }
   try {
    Thread.sleep(10);
   } catch (Exception ex) {
   }// try catch ends here

  }// while loop ends here
 }
public void keyPressed(KeyEvent e) {
  int keyCode = e.getKeyCode();
  if (keyCode == KeyEvent.VK_LEFT) {
   left = true;
   // System.out.print("left");
  }
  if (keyCode == KeyEvent.VK_RIGHT) {
   right = true;
   // System.out.print("right");
  }
```

```
 }
public void keyReleased(KeyEvent e) {
  int keyCode = e.getKeyCode();
  if (keyCode == KeyEvent.VK_LEFT) {
   left = false;
  }
  if (keyCode == KeyEvent.VK_RIGHT) {
   right = false;
  }
 }
public void keyTyped(KeyEvent arg0) {
 }
public void actionPerformed(ActionEvent e) {
  String str = e.getActionCommand();
  if (str.equals("restart")) {
   this.restart();
  }
 }
 public void restart() {
  requestFocus(true);
  initializeVariables();
  createBricks();
  repaint();
 }
 public void initializeVariables(){
      ballx = 160;
      bally = 218;
      batx = 160;
      baty = 245;
      brickx = 70;
      bricky = 50;
      Ball = new Rectangle(ballx, bally, 5, 5);
      Bat = new Rectangle(batx, baty, 40, 5);
      // Rectangle Brick;// = new Rectangle(brickx, bricky, 30, 10);
      Brick = new Rectangle[12];
      movex = -1;
      movey = -1;
      ballFallDown = false;
      bricksOver = false;
      count = 0;
      status = null;
 }
 public void createBricks(){
```

```
    for (int i = 0; i < Brick.length; i++) {
     Brick[i] = new Rectangle(brickx, bricky, brickBreadth, brickHeight);
     if (i == 5) {
      brickx = 70;
      bricky = (bricky + brickHeight + 2);
     }
     if (i == 9) {
      brickx = 100;
      bricky = (bricky + brickHeight + 2);

     }
     brickx += (brickBreadth+1);
    }
 }
}
```

Result :