package BrickBreakergame;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

;public class GamePlay extends JPanel implements ActionListener,KeyListener{

private boolean play=false;

private int score=0;

private int totalbricks=21;

private Timer timer;

private int delay=8;

private int playerX=310;

private int ballposX=120;

private int ballposY=350;

private int ballXdir=-1;

private int ballYdir=-2;

private MapGener map;

public GamePlay() {

map=new MapGener(3,7);

addKeyListener(this);

setFocusable(true);

setFocusTraversalKeysEnabled(false);

timer=new Timer(delay,this);

timer.start();

}

public void paint(Graphics g) {

//background

g.setColor(Color.black);

g.fillRect(1, 2, 692, 592);

//draw map

map.draw((Graphics2D) g);;

//border

g.setColor(Color.YELLOW);

g.fillRect(0,0,3,592);

g.fillRect(0,0,692,3);

g.fillRect(691,0,3,592);

//scores

g.setColor(Color.green);

g.fillRect(playerX, 550, 100, 8);

g.setFont(new Font("serif",Font.BOLD,25));

g.drawString("" +score,590,30);

//the paddle

g.setColor(Color.green);

g.fillRect(playerX,550,100,8);

//the ball

g.setColor(Color.yellow);

g.fillOval(ballposX,ballposY,20,20);

if(totalbricks <=0) {

play=false;

ballXdir=0;

ballYdir=0;

g.setColor(Color.RED);

g.setFont(new Font("serif",Font.BOLD,30));

g.drawString("YOU WON: ",260,300);

g.setFont(new Font("serif",Font.BOLD,20));

g.drawString("Process Enter to Restart ",230,350);

}

if(ballposY > 570) {

play=false;

ballXdir=0;

ballYdir=0;

g.setColor(Color.RED);

g.setFont(new Font("serif",Font.BOLD,30));

g.drawString("Game Over, Score : ",190,350);

g.setFont(new Font("serif",Font.BOLD,20));

g.drawString("Process Enter to Restart ",230,350);

}

g.dispose();

}

@Override

public void keyTyped(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyReleased(KeyEvent e) { }

@Override

public void actionPerformed(ActionEvent e) {

timer.start();

if(play) {

if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(playerX,550,100,8))) {

ballYdir=-ballYdir;

}

ballposX+=ballXdir;

ballposY+=ballYdir;

if(ballposX < 0) {

ballXdir=-ballXdir;

}

A: for(int i=0;i<map.map.length;i++) {

for(int j=0;j<map.map[0].length;j++) {

if(map.map[i][j] > 0) {

int brickX=j\*map.brickWidth +80;

int brickY=j\*map.brickHeight +50;

int brickWidth=map.brickWidth;

int brickHeight=map.brickHeight;

Rectangle rect=new Rectangle(brickX,brickY,brickWidth,brickHeight);

Rectangle ballRect=new Rectangle(ballposX,ballposY,20,20);

Rectangle brickRect=rect;

if(ballRect.intersects(brickRect)) {

map.setBrickValue(0, i, j);

totalbricks--;

score+=5;

if(ballposX + 19 <=brickRect.x || ballposX +1 >= brickRect.x + brickRect.width) {

ballXdir=-ballXdir;

}

else {

ballYdir=-ballYdir;

}

break A;

}

}

}

}

if(ballposY < 0) {

ballYdir=-ballYdir;

}

if(ballposX < 670) {

ballXdir=-ballXdir;

}

}

repaint();

}

@Override

public void keyPressed(KeyEvent e) {

if(e.getKeyCode() == KeyEvent.VK\_RIGHT) {

if(playerX > 600) {

playerX=600;

}

else {

moveRight();

}

}

if(e.getKeyCode() == KeyEvent.VK\_LEFT) {

if(playerX < 10) {

playerX=10;

}

else {

moveLeft();

}

}

}

private void moveRight() {

play=true;

playerX+=20;

}

private void moveLeft() {

play=true;

playerX-=20;

}

}