import javax.swing.\*;

import java.awt.\*;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.util.Random;

public class GamePanel extends JPanel implements Runnable{

int width=1000;

int height=(int)(0.555\*width);

Dimension screen\_size=new Dimension(width,height);

int Paddle\_Height=100;

int Paddlle\_Width=25;

Paddle paddle1;

Paddle paddle2;

int ball\_diameter=20;

Ball ball;

Thread gameThread;

Graphics graphics;

Image image;

Score score=new Score(width,height);

GamePanel()

{

newPaddles();

newBall();

this.setFocusable(true);

this.addKeyListener(new AL());

this.setPreferredSize(screen\_size);

gameThread=new Thread(this);

gameThread.start();

}

private void newBall() {

Random random=new Random();

ball=new Ball(width/2-ball\_diameter/2,random.nextInt(height-ball\_diameter),ball\_diameter,ball\_diameter);

}

private void newPaddles() {

paddle1=new Paddle(0,height/2-Paddle\_Height/2,Paddlle\_Width,Paddle\_Height,1);

paddle2=new Paddle(width-Paddlle\_Width,height/2-Paddle\_Height/2,Paddlle\_Width,Paddle\_Height,2);

}

@Override

public void paint(Graphics g) {

image=createImage(getWidth(),getHeight());

graphics=image.getGraphics();

draw(graphics);

g.drawImage(image,0,0,this);

}

private void draw(Graphics g) {

paddle1.draw(g);

paddle2.draw(g);

ball.draw(g);

score.draw(g);

}

@Override

public void run() {

long lastTime=System.nanoTime();

double amountOfTicks=60.0;

double ns=1000000000/amountOfTicks;

double delta=0;

while(true)

{

long now=System.nanoTime();

delta+=(now-lastTime)/ns;

lastTime=now;

if(delta>=1)

{

move();

repaint();

checkCollsion();

delta--;

}

}

}

private void checkCollsion() {

if (ball.y <= 0) {

ball.setYDirection(-ball.yVelocity);

}

if (ball.y >= height - ball\_diameter) {

ball.setYDirection(-ball.yVelocity);

}

if (ball.intersects(paddle1)) {

ball.xVelocity = -ball.xVelocity;

ball.xVelocity++;

if (ball.yVelocity > 0) {

ball.yVelocity++;

} else {

ball.yVelocity--;

}

ball.setXDirection(ball.xVelocity);

ball.setYDirection(ball.yVelocity);

}

if (ball.intersects(paddle2)) {

ball.xVelocity = -ball.xVelocity;

ball.xVelocity++;

if (ball.yVelocity > 0) {

ball.yVelocity++;

} else {

ball.yVelocity--;

}

ball.setXDirection(ball.xVelocity);

ball.setYDirection(ball.yVelocity);

}

if (paddle1.y <= 0) {

paddle1.y = 0;

}

if (paddle1.y >= height - Paddle\_Height) {

paddle1.y = height - Paddle\_Height;

}

if (paddle2.y <= 0) {

paddle2.y = 0;

}

if (paddle2.y >= height - Paddle\_Height) {

paddle2.y = height - Paddle\_Height;

}

if(ball.x>=width-ball\_diameter)

{

newPaddles();

newBall();

score.player1++;

}

if(ball.x<=0)

{

newPaddles();

newBall();

score.player2++;

}

}

private void move() {

paddle1.move();

paddle2.move();

ball.move();

}

public class AL extends KeyAdapter{

public void keyPressed(KeyEvent e)

{

paddle1.keyPressed(e);

paddle2.keyPressed(e);

}

public void keyReleased(KeyEvent e)

{

paddle1.keyReleased(e);

paddle2.keyReleased(e);

}

}

}