import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import javax.sound.sampled.\*;

import javax.swing.\*;

public class FlappyBird extends JPanel implements ActionListener, KeyListener {

private static final int WIDTH = 800;

private static final int HEIGHT = 600;

private static final int PIPE\_WIDTH = 80;

private static final int PIPE\_GAP = 200;

private static final int PLANE\_WIDTH = 100;

private static final int PLANE\_HEIGHT = 30;

private int planeY = HEIGHT / 2;

private float planeVelocity = 0;

private float gravity = 0.5f;

private float jumpStrength = -10f;

private int[] pipeX = new int[3];

private int[] pipeY = new int[3];

private Timer timer;

private boolean gameOver = false;

private boolean paused = false;

private int score = 0;

private int highScore = 0;

private BufferedImage planeImage;

private BufferedImage pipeTopImage;

private BufferedImage pipeBottomImage;

private BufferedImage gameOverImage;

private Clip backgroundMusicClip;

public FlappyBird() {

setPreferredSize(new Dimension(WIDTH, HEIGHT));

setBackground(new Color(135, 206, 235)); // Sky blue

timer = new Timer(20, this);

addKeyListener(this);

setFocusable(true);

resetGame();

loadBackgroundMusic();

// Load images

try {

planeImage = ImageIO.read(new File("plane.png"));

pipeTopImage = ImageIO.read(new File("pipe\_top.png"));

pipeBottomImage = ImageIO.read(new File("pipe\_bottom.png"));

gameOverImage = ImageIO.read(new File("game\_over.png"));

} catch (IOException e) {

e.printStackTrace();

}

}

private void loadBackgroundMusic() {

try {

File audioFile = new File("background\_music.wav");

AudioInputStream audioStream = AudioSystem.getAudioInputStream(audioFile);

AudioFormat format = audioStream.getFormat();

DataLine.Info info = new DataLine.Info(Clip.class, format);

backgroundMusicClip = (Clip) AudioSystem.getLine(info);

backgroundMusicClip.open(audioStream);

backgroundMusicClip.loop(Clip.LOOP\_CONTINUOUSLY);

} catch (UnsupportedAudioFileException | IOException | LineUnavailableException e) {

e.printStackTrace();

}

}

private void resetGame() {

planeY = HEIGHT / 2;

planeVelocity = 0;

score = 0;

gameOver = false;

paused = false;

for (int i = 0; i < 3; i++) {

pipeX[i] = WIDTH + i \* 300;

pipeY[i] = (int) (Math.random() \* (HEIGHT - PIPE\_GAP - 200)) + 100;

}

timer.start();

}

private void drawPipe(Graphics2D g2d, int x, int y) {

if (pipeTopImage != null) {

g2d.drawImage(pipeTopImage, x, y - pipeTopImage.getHeight(), PIPE\_WIDTH, pipeTopImage.getHeight(), null);

}

if (pipeBottomImage != null) {

g2d.drawImage(pipeBottomImage, x, y + PIPE\_GAP, PIPE\_WIDTH, pipeBottomImage.getHeight(), null);

}

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

// Draw background

g2d.setColor(new Color(135, 206, 235)); // Sky blue

g2d.fillRect(0, 0, WIDTH, HEIGHT);

// Draw ground

g2d.setColor(new Color(34, 139, 34)); // Forest green

g2d.fillRect(0, HEIGHT - 100, WIDTH, 100);

// Draw pipes

for (int i = 0; i < 3; i++) {

drawPipe(g2d, pipeX[i], pipeY[i]);

}

// Draw plane

if (planeImage != null) {

g2d.drawImage(planeImage, 100, planeY, PLANE\_WIDTH, PLANE\_HEIGHT, null);

} else {

g2d.setColor(Color.YELLOW);

g2d.fillOval(100, planeY, PLANE\_WIDTH, PLANE\_HEIGHT);

}

// Draw score

g2d.setColor(Color.WHITE);

g2d.setFont(new Font("Arial", Font.BOLD, 20));

g2d.drawString("Score: " + score, 10, 30);

g2d.drawString("High Score: " + highScore, 10, 60);

if (gameOver) {

if (gameOverImage != null) {

g2d.drawImage(gameOverImage, 0, 0, WIDTH, HEIGHT, null);

} else {

g2d.setColor(Color.RED);

g2d.setFont(new Font("Arial", Font.BOLD, 40));

g2d.drawString("GAME OVER", WIDTH / 2 - 100, HEIGHT / 2);

}

g2d.setFont(new Font("Arial", Font.BOLD, 20));

g2d.drawString("Press R to restart", WIDTH / 2 - 80, HEIGHT / 2 + 40);

}

if (paused) {

g2d.setColor(new Color(0, 0, 0, 128));

g2d.fillRect(0, 0, WIDTH, HEIGHT);

g2d.setColor(Color.WHITE);

g2d.setFont(new Font("Arial", Font.BOLD, 40));

g2d.drawString("PAUSED", WIDTH / 2 - 60, HEIGHT / 2);

}

}

@Override

public void actionPerformed(ActionEvent e) {

if (!gameOver && !paused) {

planeVelocity += gravity;

planeY += planeVelocity;

for (int i = 0; i < 3; i++) {

pipeX[i] -= 5;

if (pipeX[i] < -PIPE\_WIDTH) {

pipeX[i] = WIDTH;

pipeY[i] = (int) (Math.random() \* (HEIGHT - PIPE\_GAP - 200)) + 100;

}

if (pipeX[i] == 95) {

score++;

if (score > highScore) {

highScore = score;

}

}

if ((pipeX[i] < 100 + PLANE\_WIDTH && pipeX[i] + PIPE\_WIDTH > 100) &&

(planeY < pipeY[i] || planeY + PLANE\_HEIGHT > pipeY[i] + PIPE\_GAP)) {

gameOver = true;

}

}

if (planeY > HEIGHT - 100 - PLANE\_HEIGHT || planeY < 0) {

gameOver = true;

}

} else if (gameOver && backgroundMusicClip != null) {

backgroundMusicClip.stop();

}

repaint();

}

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_SPACE && !gameOver && !paused) {

planeVelocity = jumpStrength;

} else if (e.getKeyCode() == KeyEvent.VK\_R) {

resetGame();

} else if (e.getKeyCode() == KeyEvent.VK\_P) {

paused = !paused;

if (paused) {

timer.stop();

} else {

timer.start();

}

}

}

@Override

public void keyTyped(KeyEvent e) {}

@Override

public void keyReleased(KeyEvent e) {}

public static void main(String[] args) {

JFrame frame = new JFrame("Flappy Plane");

FlappyBird game = new FlappyBird();

frame.add(game);

frame.pack();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

    }

}