**FLAPPY BIRD JAVA PROGRAM MAIN FILE**

**GUI CODE:**

**import java.awt.BorderLayout;**

**import java.awt.Color;**

**import java.awt.Dimension;**

**import java.awt.Font;**

**import java.awt.Graphics;**

**import java.awt.Image;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.ComponentAdapter;**

**import java.awt.event.ComponentEvent;**

**import java.io.File;**

**import java.util.regex.\*;**

**import javax.sound.sampled.AudioInputStream;**

**import javax.sound.sampled.AudioSystem;**

**import javax.sound.sampled.Clip;**

**import javax.swing.BorderFactory;**

**import javax.swing.ButtonGroup;**

**import javax.swing.ImageIcon;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JPasswordField;**

**import javax.swing.JRadioButton;**

**import javax.swing.JTextField;**

**import javax.swing.JToggleButton;**

**import javax.swing.SwingUtilities;**

**public class CircularImageButton extends JFrame {**

**private JButton startButton;**

**private JButton settingButton;**

**private JButton helpButton;**

**private JButton loginButton;**

**private static String selectedOption = null;**

**private static String BackSelect = "";**

**private static JToggleButton muteButton;**

**private static boolean isUnMuted = true;**

**private static Clip clip;**

**// private static boolean isMusicPlaying = true;**

**public CircularImageButton() {**

**initMusic();**

**clip.start();**

**// Set the background image panel**

**ImageIcon backgroundImage = new ImageIcon("front.jpg");**

**JPanel backgroundPanel = new JPanel() {**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**g.drawImage(backgroundImage.getImage(), 0, 0, getWidth(), getHeight(), this);**

**}**

**};**

**setContentPane(backgroundPanel);**

**// Set the Start Setting, Help, and Login buttons**

**startButton = new JButton();**

**settingButton = new JButton();**

**helpButton = new JButton();**

**loginButton = new JButton();**

**// set the icons on the buttons**

**startButton.setIcon(new ImageIcon("start.png"));**

**settingButton.setIcon(new ImageIcon("setting.png"));**

**helpButton.setIcon(new ImageIcon("help.png"));**

**loginButton.setIcon(new ImageIcon("login.png"));**

**// remove the border of the button**

**startButton.setBorder(BorderFactory.createEmptyBorder());**

**settingButton.setBorder(BorderFactory.createEmptyBorder());**

**helpButton.setBorder(BorderFactory.createEmptyBorder());**

**loginButton.setBorder(BorderFactory.createEmptyBorder());**

**startButton.setContentAreaFilled(false);**

**settingButton.setContentAreaFilled(false);**

**loginButton.setContentAreaFilled(false);**

**helpButton.setContentAreaFilled(false);**

**settingButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**// adding Setting button action here**

**showSettingWindow();**

**}**

**});**

**helpButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**// adding Help button action here**

**showHelpWindow();**

**}**

**});**

**loginButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**// adding Login button action here**

**showLoginWindow();**

**}**

**});**

**// adding a component listener to adjust button positions when resized**

**addComponentListener(new ComponentAdapter() {**

**public void componentResized(ComponentEvent e) {**

**updateButtonPositions();**

**}**

**});**

**// adding the buttons to the background panel**

**backgroundPanel.setLayout(null);**

**backgroundPanel.add(settingButton);**

**backgroundPanel.add(loginButton);**

**backgroundPanel.add(helpButton);**

**// Set the Start button**

**startButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**if (BackSelect=="Option1"){**

**if (selectedOption == "Option 1") {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 2") {**

**ImageIcon birdImageIcon = new ImageIcon("bird21.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird22.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col2.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 3") {**

**ImageIcon birdImageIcon = new ImageIcon("bird31.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird32.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col3.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 4") {**

**ImageIcon birdImageIcon = new ImageIcon("bird41.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird42.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col4.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**}**

**}**

**else if(BackSelect=="Option2"){**

**if (selectedOption == "Option 1") {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 2") {**

**ImageIcon birdImageIcon = new ImageIcon("bird21.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird22.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col2.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background2.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 3") {**

**ImageIcon birdImageIcon = new ImageIcon("bird31.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird32.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col3.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background2.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 4") {**

**ImageIcon birdImageIcon = new ImageIcon("bird41.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird42.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col4.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background2.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background2.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**}**

**}**

**else {**

**if (selectedOption == "Option 1") {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 2") {**

**ImageIcon birdImageIcon = new ImageIcon("bird21.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird22.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col2.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 3") {**

**ImageIcon birdImageIcon = new ImageIcon("bird31.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird32.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background2.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else if (selectedOption == "Option 4") {**

**ImageIcon birdImageIcon = new ImageIcon("bird41.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird42.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col4.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**} else {**

**ImageIcon birdImageIcon = new ImageIcon("bird11.png");**

**ImageIcon bird1ImageIcon = new ImageIcon("bird12.png");**

**ImageIcon birdCollisionImageIcon = new ImageIcon("bird\_col1.png");**

**ImageIcon backgroundImageIcon = new ImageIcon("background1.png");**

**ImageIcon upperPipeImageIcon = new ImageIcon("pipe.png");**

**ImageIcon lowerPipeImageIcon = new ImageIcon("pipe1.png");**

**Image birdImage = birdImageIcon.getImage();**

**Image bird1Image = bird1ImageIcon.getImage();**

**Image birdCollisionImage = birdCollisionImageIcon.getImage();**

**Image backgroundImage = backgroundImageIcon.getImage();**

**Image upperPipeImage = upperPipeImageIcon.getImage();**

**Image lowerPipeImage = lowerPipeImageIcon.getImage();**

**JFrame gameFrame = new JFrame("Flappy Bird");**

**FlappyBirdGame flappyBirdGame = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,**

**backgroundImage, upperPipeImage, lowerPipeImage);**

**gameFrame.add(flappyBirdGame);**

**gameFrame.setSize(800, 600);**

**gameFrame.dispose();**

**gameFrame.setVisible(true);**

**gameFrame.setResizable(true);**

**gameFrame.addKeyListener(flappyBirdGame);**

**gameFrame.setLocationRelativeTo(null);**

**}**

**}**

**}**

**});**

**// adding the Start button to the background panel**

**backgroundPanel.add(startButton);**

**setTitle("Background Image with Buttons");**

**setSize(800, 600);**

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

**setLocationRelativeTo(null);**

**setVisible(true);**

**// Initial update of button positions**

**updateButtonPositions();**

**}**

**// Method to update the positions of the buttons**

**private void updateButtonPositions() {**

**int settingX = getWidth() - 100;**

**int loginX = getWidth() - 100;**

**int helpX = getWidth() - 100;**

**settingButton.setBounds(settingX, 30, 50, 50);**

**loginButton.setBounds(loginX, 100, 50, 50);**

**helpButton.setBounds(helpX, 170, 50, 50);**

**// Calculate the position of the Start button based on a percentage of window**

**// size**

**int startX = getWidth() / 2 - 95;**

**int startY = getHeight() - 170;**

**startButton.setBounds(startX, startY, 180, 45);**

**}**

**// Method to show the Help window**

**private void showHelpWindow() {**

**JFrame helpFrame = new JFrame("Help Window");**

**helpFrame.setSize(800, 600);**

**// Load the background image**

**ImageIcon originalImageIcon = new ImageIcon("HelpBack.png");**

**Image originalImage = originalImageIcon.getImage();**

**// Create a JLabel with the scaled background image**

**JLabel backgroundLabel = new JLabel(**

**new ImageIcon(originalImage.getScaledInstance(400, 300, Image.SCALE\_SMOOTH)));**

**// Make the background label transparent**

**backgroundLabel.setOpaque(false);**

**// adding the background label**

**helpFrame.add(backgroundLabel);**

**// Resize listener to adjust background image size**

**helpFrame.addComponentListener(new ComponentAdapter() {**

**public void componentResized(ComponentEvent e) {**

**Image scaledImage = originalImage.getScaledInstance(helpFrame.getWidth(), helpFrame.getHeight(),**

**Image.SCALE\_SMOOTH);**

**backgroundLabel.setIcon(new ImageIcon(scaledImage));**

**}**

**});**

**helpFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);**

**helpFrame.setVisible(true);**

**helpFrame.setLocationRelativeTo(null);**

**}**

**// Inside the showLoginWindow() method**

**private void showLoginWindow() {**

**JFrame loginFrame = new JFrame("Login Window");**

**loginFrame.setSize(800, 600); // Adjusted height to accommodate the new text**

**JPanel loginPanel = new JPanel() {**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**ImageIcon backgroundImage = new ImageIcon("BackMain.png");**

**g.drawImage(backgroundImage.getImage(), 0, 0, getWidth(), getHeight(), this);**

**}**

**};**

**loginPanel.setLayout(null); // Change to null layout for coordinate-based placement**

**JLabel loginLabel = new JLabel("Sign In");**

**loginLabel.setFont(new Font("Arial", Font.BOLD, 16));**

**JLabel usernameLabel = new JLabel("Username:");**

**JTextField usernameField = new JTextField();**

**JLabel passwordLabel = new JLabel("Password:");**

**JPasswordField passwordField = new JPasswordField();**

**JButton loginButton = new JButton();**

**loginButton.setIcon(new ImageIcon(new ImageIcon("log\_but.png").getImage().getScaledInstance(160, 30, Image.SCALE\_DEFAULT)));**

**// remove the borders of the text field**

**usernameField.setBorder(BorderFactory.createEmptyBorder());**

**passwordField.setBorder(BorderFactory.createEmptyBorder());**

**// set the colour details**

**loginLabel.setForeground(Color.white);**

**usernameLabel.setForeground(Color.WHITE);**

**passwordLabel.setForeground(Color.WHITE);**

**usernameField.setBackground(Color.decode("#C1E4D0"));**

**passwordField.setBackground(Color.decode("#C1E4D0"));**

**loginButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String username = usernameField.getText();**

**String password = new String(passwordField.getPassword());**

**String regex = "^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\\d)(?=.\*[@#$%^&+=]).{8,}$";**

**Pattern pattern = Pattern.compile(regex);**

**Matcher matcher = pattern.matcher(password);**

**if (username.isEmpty() || password.isEmpty()) {**

**JOptionPane.showMessageDialog(loginFrame, "Please fill out both username and password fields.");**

**return; // Return without displaying credentials**

**}**

**if(!matcher.matches()||(password.length() < 8)){**

**JOptionPane.showMessageDialog(loginFrame,"Password Contains atleast 1 Capital,1 Small & 1 special Character.");**

**return; // Return without displaying credentials**

**}**

**if (password.length() < 8) {**

**JOptionPane.showMessageDialog(loginFrame, "Password should have a minimum length of 8 characters.");**

**return; // Return without displaying credentials**

**}**

**// Print the credentials below the login button**

**JLabel credentialsLabel1 = new JLabel("USERNAME: " + username);**

**credentialsLabel1.setForeground(Color.WHITE);**

**credentialsLabel1.setBounds(250, 300, 300, 25);**

**loginPanel.add(credentialsLabel1);**

**JLabel credentialsLabel2 = new JLabel("PASSWORD: " + password);**

**credentialsLabel2.setForeground(Color.WHITE);**

**credentialsLabel2.setBounds(250, 330, 300, 25);**

**loginPanel.add(credentialsLabel2);**

**// Clear the text fields**

**usernameField.setText("");**

**passwordField.setText("");**

**}**

**});**

**JLabel noAccountLabel = new JLabel("Don't have an account?");**

**noAccountLabel.setForeground(Color.WHITE);**

**JButton createAccountButton = new JButton();**

**createAccountButton.setIcon(new ImageIcon(new ImageIcon("create.png").getImage().getScaledInstance(160, 30, Image.SCALE\_DEFAULT)));**

**createAccountButton.setContentAreaFilled(false);**

**createAccountButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**// Show the account creation window and close the login window**

**showAccountCreationWindow();**

**loginFrame.dispose();**

**}**

**});**

**// Set bounds for each component**

**usernameLabel.setBounds(250, 50, 100, 25);**

**usernameField.setBounds(350, 50, 150, 25);**

**passwordLabel.setBounds(250, 100, 100, 25);**

**passwordField.setBounds(350, 100, 150, 25);**

**loginButton.setBounds(345, 155, 160, 30);**

**noAccountLabel.setBounds(250, 210, 160, 30); // addinged noAccountLabel placement**

**createAccountButton.setBounds(345, 255, 160, 30); // Adjusted createAccountButton placement**

**loginLabel.setBounds(400, 12, 180, 45);**

**// remove the clickability of the login button**

**loginButton.setBorder(BorderFactory.createEmptyBorder());**

**loginButton.setContentAreaFilled(false);**

**createAccountButton.setBorder(BorderFactory.createEmptyBorder());**

**loginPanel.add(usernameLabel);**

**loginPanel.add(usernameField);**

**loginPanel.add(passwordLabel);**

**loginPanel.add(passwordField);**

**loginPanel.add(loginButton);**

**loginPanel.add(noAccountLabel);**

**loginPanel.add(createAccountButton);**

**loginPanel.add(loginLabel);**

**loginFrame.add(loginPanel);**

**loginFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);**

**loginFrame.setVisible(true);**

**loginFrame.setLocationRelativeTo(null);**

**}**

**// Method to show the Account Creation window**

**private void showAccountCreationWindow() {**

**JFrame accountFrame = new JFrame("Account Creation Window");**

**accountFrame.setSize(800, 600);**

**JPanel accountPanel = new JPanel() {**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**ImageIcon backgroundImage = new ImageIcon("BackMain.png");**

**g.drawImage(backgroundImage.getImage(), 0, 0, getWidth(), getHeight(), this);**

**}**

**};**

**accountPanel.setLayout(null); // Change to null layout for coordinate-based placement**

**JLabel createAccountLabel = new JLabel("Create Account");**

**createAccountLabel.setFont(new Font("Arial", Font.BOLD, 16));**

**createAccountLabel.setForeground(Color.WHITE);**

**JLabel emailLabel = new JLabel("Email Id: ");**

**emailLabel.setForeground(Color.white);**

**JLabel usernameLabel = new JLabel("Username:");**

**usernameLabel.setForeground(Color.WHITE);**

**JTextField usernameField = new JTextField();**

**usernameField.setBackground(Color.decode("#C1E4D0"));**

**JLabel passwordLabel = new JLabel("Password:");**

**passwordLabel.setForeground(Color.WHITE);**

**JTextField emailField = new JTextField();**

**emailField.setBackground(Color.decode("#C1E4D0"));**

**JPasswordField passwordField = new JPasswordField();**

**passwordField.setBackground(Color.decode("#C1E4D0"));**

**JButton submitButton = new JButton();**

**submitButton.setIcon(new ImageIcon(new ImageIcon("submit1.png").getImage().getScaledInstance(160, 30, Image.SCALE\_DEFAULT)));**

**// remove the borders of the text field**

**usernameField.setBorder(BorderFactory.createEmptyBorder());**

**emailField.setBorder(BorderFactory.createEmptyBorder());**

**passwordField.setBorder(BorderFactory.createEmptyBorder());**

**// remove the clickability of the submit button**

**submitButton.setContentAreaFilled(false);**

**submitButton.setBorder(BorderFactory.createEmptyBorder());**

**submitButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String username = usernameField.getText();**

**String email = emailField.getText();**

**String password = new String(passwordField.getPassword());**

**String regex = "^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\\d)(?=.\*[@#$%^&+=]).{8,}$";**

**Pattern pattern = Pattern.compile(regex);**

**Matcher matcher = pattern.matcher(password);**

**String regex1 = "^[A-Za-z0-9+\_.-]+@(.+)$" ;**

**Pattern pattern1 = Pattern.compile(regex1);**

**Matcher matcher1 = pattern1.matcher(email);**

**if (username.isEmpty() || email.isEmpty() || password.isEmpty()) {**

**JOptionPane.showMessageDialog(accountFrame, "Please fill out all Credentials Username,Email Id and Password Fields.");**

**return; // Return without displaying credentials**

**}**

**if(!matcher1.matches()){**

**JOptionPane.showMessageDialog(accountFrame,"Make sure Your Email\_Id is Proper");**

**return; // Return without displaying credentials**

**}**

**if (password.length() < 8) {**

**JOptionPane.showMessageDialog(accountFrame, "Password should have a minimum length of 8 characters.");**

**return; // Return without displaying credentials**

**}**

**if(!matcher.matches()){**

**JOptionPane.showMessageDialog(accountFrame,"Password Contains 1 Capital,1 Small & 1 special Character.");**

**return; // Return without displaying credentials**

**}**

**// Print the credentials below the submit button**

**JLabel credentialsLabel1 = new JLabel("USERNAME: " + username);**

**credentialsLabel1.setBounds(250, 290, 300, 25);**

**credentialsLabel1.setForeground(Color.white);**

**accountPanel.add(credentialsLabel1);**

**JLabel credentialsLabel2 = new JLabel("EMAIL ID: " + email);**

**credentialsLabel2.setBounds(250, 320, 300, 25);**

**credentialsLabel2.setForeground(Color.WHITE);**

**accountPanel.add(credentialsLabel2);**

**JLabel credentialsLabel3 = new JLabel("PASSWORD: " + password);**

**credentialsLabel3.setBounds(250, 350, 300, 25);**

**credentialsLabel3.setForeground(Color.WHITE);**

**accountPanel.add(credentialsLabel3);**

**// Clear the text fields**

**usernameField.setText("");**

**emailField.setText("");**

**passwordField.setText("");**

**}**

**});**

**// Set bounds for each component**

**createAccountLabel.setBounds(365, 20, 200, 25);**

**usernameLabel.setBounds(250, 80, 100, 25);**

**usernameField.setBounds(350, 80, 150, 25);**

**emailField.setBounds(350, 130, 150, 25);**

**emailLabel.setBounds(250, 130, 100, 25);**

**passwordLabel.setBounds(250, 180, 100, 25);**

**passwordField.setBounds(350, 180, 150, 25);**

**submitButton.setBounds(336, 235, 180, 30);**

**accountPanel.add(createAccountLabel);**

**accountPanel.add(usernameLabel);**

**accountPanel.add(usernameField);**

**accountPanel.add(passwordLabel);**

**accountPanel.add(passwordField);**

**accountPanel.add(submitButton);**

**accountPanel.add(emailField);**

**accountPanel.add(emailLabel);**

**accountFrame.add(accountPanel);**

**accountFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);**

**accountFrame.setVisible(true);**

**accountFrame.setLocationRelativeTo(null);**

**}**

**// Method to show the Setting window with radio buttons and images**

**private static void showSettingWindow() {**

**JFrame settingFrame = new JFrame("Setting Window");**

**settingFrame.setSize(900, 820);**

**// Set the background image**

**ImageIcon backgroundImage = new ImageIcon("settingBack.png");**

**JLabel backgroundLabel = new JLabel(backgroundImage);**

**backgroundLabel.setBounds(0, 0, backgroundImage.getIconWidth(), backgroundImage.getIconHeight());**

**settingFrame.getLayeredPane().add(backgroundLabel, Integer.valueOf(Integer.MIN\_VALUE));**

**JPanel contentPane = new JPanel() {**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**backgroundLabel.setBounds(0, 0, getWidth(), getHeight());**

**}**

**};**

**contentPane.setOpaque(false);**

**contentPane.setLayout(null);**

**settingFrame.setContentPane(contentPane);**

**contentPane.addComponentListener(new ComponentAdapter() {**

**public void componentResized(ComponentEvent e) {**

**Dimension size = contentPane.getSize();**

**ImageIcon scaledImage = new ImageIcon(**

**backgroundImage.getImage().getScaledInstance(size.width, size.height, Image.SCALE\_SMOOTH));**

**backgroundLabel.setIcon(scaledImage);**

**}**

**});**

**// 1**

**JRadioButton radioButton1 = new JRadioButton();**

**radioButton1.setBounds(115, 325, 15, 15);**

**JLabel label1 = new JLabel(new ImageIcon("icon1.png"));**

**label1.setBounds(50, 165, 150, 150);**

**radioButton1.setLayout(new BorderLayout());**

**// 2**

**JRadioButton radioButton2 = new JRadioButton();**

**radioButton2.setBounds(295, 325, 15, 15);**

**JLabel label2 = new JLabel(new ImageIcon("icon2.png"));**

**label2.setBounds(230, 165, 150, 150);**

**radioButton2.setLayout(new BorderLayout());**

**// 3**

**JRadioButton radioButton3 = new JRadioButton();**

**radioButton3.setBounds(475, 325, 15, 15);**

**JLabel label3 = new JLabel(new ImageIcon("icon3.png"));**

**label3.setBounds(410, 165, 150, 150);**

**radioButton3.setLayout(new BorderLayout());**

**// 4**

**JRadioButton radioButton4 = new JRadioButton();**

**radioButton4.setBounds(660, 325, 15, 15);**

**JLabel label4 = new JLabel(new ImageIcon("icon4.png"));**

**label4.setBounds(590, 165, 150, 150);**

**radioButton4.setLayout(new BorderLayout());**

**//5**

**JRadioButton radioButton5 = new JRadioButton();**

**radioButton5.setBounds(115, 560, 15, 15);**

**radioButton5.setLayout(new BorderLayout());**

**JLabel label5 = new JLabel(new ImageIcon("day.png"));**

**label5.setBounds(50, 400, 150, 150);**

**//6**

**JRadioButton radioButton6 = new JRadioButton();**

**radioButton6.setBounds(295, 560, 15, 15);**

**radioButton6.setLayout(new BorderLayout());**

**JLabel label6 = new JLabel(new ImageIcon("night.png"));**

**label6.setBounds(230, 400, 150, 150);**

**// radiobutton background**

**radioButton1.setContentAreaFilled(false);**

**radioButton2.setContentAreaFilled(false);**

**radioButton3.setContentAreaFilled(false);**

**radioButton4.setContentAreaFilled(false);**

**radioButton5.setContentAreaFilled(false);**

**radioButton6.setContentAreaFilled(false);**

**// adding the images infront of the radio buttons**

**contentPane.add(label1);**

**contentPane.add(label2);**

**contentPane.add(label3);**

**contentPane.add(label4);**

**contentPane.add(label5);**

**contentPane.add(label6);**

**// make a group of radiobuttons**

**ButtonGroup buttonGroup = new ButtonGroup();**

**buttonGroup.add(radioButton1);**

**buttonGroup.add(radioButton2);**

**buttonGroup.add(radioButton3);**

**buttonGroup.add(radioButton4);**

**ButtonGroup buttonGroup2=new ButtonGroup();**

**buttonGroup2.add(radioButton5);**

**buttonGroup2.add(radioButton6);**

**// adding radioButton1 to contentPane...**

**contentPane.add(radioButton1);**

**contentPane.add(radioButton2);**

**contentPane.add(radioButton3);**

**contentPane.add(radioButton4);**

**contentPane.add(radioButton5);**

**contentPane.add(radioButton6);**

**JButton submitButton = new JButton();**

**submitButton.setBounds(370, 725, 180, 45);**

**submitButton.setIcon(new ImageIcon("submit.png"));**

**submitButton.setContentAreaFilled(false);**

**// remove the clickability of the submit button**

**submitButton.setBorder(BorderFactory.createEmptyBorder());**

**submitButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**settingFrame.dispose(); // Close the setting window after submission**

**if(radioButton5.isSelected()) {**

**BackSelect="Option1";**

**if (radioButton1.isSelected()) {**

**selectedOption = "Option 1";**

**} else if (radioButton2.isSelected()) {**

**selectedOption = "Option 2";**

**} else if (radioButton3.isSelected()) {**

**selectedOption = "Option 3";**

**} else if (radioButton4.isSelected()) {**

**selectedOption = "Option 4";**

**}}**

**else if(radioButton6.isSelected()){**

**BackSelect="Option2";**

**if (radioButton1.isSelected()) {**

**selectedOption = "Option 1";**

**} else if (radioButton2.isSelected()) {**

**selectedOption = "Option 2";**

**} else if (radioButton3.isSelected()) {**

**selectedOption = "Option 3";**

**} else if (radioButton4.isSelected()) {**

**selectedOption = "Option 4";**

**}**

**}**

**}**

**});**

**contentPane.add(submitButton);**

**muteButton = new JToggleButton();**

**if (isUnMuted){**

**muteButton.setIcon(new ImageIcon("unmute.png"));**

**}**

**else{**

**muteButton.setIcon(new ImageIcon("mute.png"));**

**}**

**muteButton.setBounds(60, 650, 50, 50);**

**muteButton.setContentAreaFilled(false);**

**muteButton.setBorder(BorderFactory.createEmptyBorder());**

**muteButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**isUnMuted = !isUnMuted;**

**updateMusic();**

**}**

**});**

**contentPane.add(muteButton);**

**settingFrame.setVisible(true);**

**settingFrame.setLocationRelativeTo(null);**

**}**

**private static void initMusic() {**

**try {**

**AudioInputStream audioInputStream = AudioSystem.getAudioInputStream(new File("flappy.wav"));**

**clip = AudioSystem.getClip();**

**clip.open(audioInputStream);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**private static void updateMusic() {**

**if (isUnMuted) {**

**initMusic();**

**muteButton.setIcon(new ImageIcon("unmute.png"));**

**clip.loop(Clip.LOOP\_CONTINUOUSLY);**

**} else {**

**clip.stop();**

**clip.close();**

**muteButton.setIcon(new ImageIcon("mute.png"));**

**}**

**}**

**public static void main(String[] args) {**

**SwingUtilities.invokeLater(new Runnable() {**

**public void run() {**

**new CircularImageButton();**

**}**

**});**

**}**

**}**

**FLAPPY BIRD MAIN GAME CODE:**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Random;**

**import java.io.IOException;**

**import javax.imageio.ImageIO;**

**public class FlappyBirdGame extends JPanel implements ActionListener, KeyListener, ComponentListener {**

**private int WIDTH = 800;**

**private int HEIGHT = 600;**

**private static final int PIPE\_WIDTH = 70;**

**private static final int PIPE\_SPACING = 250;**

**private static double POLE\_MOVE\_SPEED = 3.0;**

**private static final int JUMP\_FORCE = -15;**

**private static final int BIRD\_WIDTH = 40;**

**private static final int BIRD\_HEIGHT = 30;**

**private static final int BIRD\_START\_X = 50;**

**private static final double GRAVITY = 1.5;**

**private int birdY;**

**private int velocity;**

**private List<Rectangle> poles;**

**private int gapSize;**

**private int score;**

**private int highScore;**

**private int birdAnimationCounter;**

**private Image birdImage;**

**private Image bird1Image;**

**private Image birdCollisionImage;**

**private Image background;**

**private Image upperPipeImage;**

**private Image lowerPipeImage;**

**private boolean isGameOver;**

**private boolean isColliding;**

**private boolean isPaused;**

**private boolean gameOverPopupShownPersistent;**

**public FlappyBirdGame(Image birdImage, Image bird1Image, Image birdCollisionImage,Image background, Image upperPipeImage, Image lowerPipeImage) {**

**this.birdImage = birdImage;**

**this.bird1Image = bird1Image;**

**this.birdCollisionImage = birdCollisionImage;**

**this.background = background;**

**this.upperPipeImage = upperPipeImage;**

**this.lowerPipeImage = lowerPipeImage;**

**poles = new ArrayList<>();**

**birdY = HEIGHT / 2;**

**velocity = 0;**

**gapSize = 6 \* BIRD\_HEIGHT;**

**score = 0;**

**highScore = 0;**

**birdAnimationCounter = 0;**

**isGameOver = false;**

**isColliding = false;**

**isPaused = false;**

**gameOverPopupShownPersistent = false; // Initialize to false**

**Timer timer = new Timer(30, this);**

**timer.start();**

**addComponentListener(this);**

**generatePoles();**

**}**

**private void generatePoles() {**

**Random random = new Random();**

**int poleX = WIDTH + PIPE\_WIDTH;**

**while (poleX < WIDTH \* 2) {**

**int minY = gapSize / 2;**

**int maxY = HEIGHT - gapSize / 2;**

**int gapYPosition = minY + random.nextInt(maxY - minY + 1);**

**Rectangle upperPole = new Rectangle(poleX, 0, PIPE\_WIDTH, gapYPosition - gapSize / 2);**

**Rectangle lowerPole = new Rectangle(poleX, gapYPosition + gapSize / 2, PIPE\_WIDTH, HEIGHT - gapYPosition - gapSize / 2);**

**poles.add(upperPole);**

**poles.add(lowerPole);**

**poleX += PIPE\_SPACING;**

**}**

**}**

**private void movePoles() {**

**for (Rectangle pole : poles) {**

**pole.x -= POLE\_MOVE\_SPEED;**

**}**

**if (poles.get(0).x + PIPE\_WIDTH < 0) {**

**poles.remove(0);**

**poles.remove(0);**

**int minY = gapSize / 2;**

**int maxY = HEIGHT - gapSize / 2;**

**int gapYPosition = minY + new Random().nextInt(maxY - minY + 1);**

**Rectangle upperPole = new Rectangle(poles.get(poles.size() - 1).x + PIPE\_SPACING, 0, PIPE\_WIDTH, gapYPosition - gapSize / 2);**

**Rectangle lowerPole = new Rectangle(poles.get(poles.size() - 1).x + PIPE\_SPACING, gapYPosition + gapSize / 2, PIPE\_WIDTH, HEIGHT - gapYPosition - gapSize / 2);**

**poles.add(upperPole);**

**poles.add(lowerPole);**

**score++;**

**if (score > highScore) {**

**highScore = score;**

**}**

**POLE\_MOVE\_SPEED += 0.2;**

**}**

**}**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**g.drawImage(background, 0, 0, WIDTH, HEIGHT, null);**

**for (int i = 0; i < poles.size(); i += 2) {**

**Rectangle upperPole = poles.get(i);**

**Rectangle lowerPole = poles.get(i + 1);**

**g.drawImage(upperPipeImage, upperPole.x, upperPole.y, PIPE\_WIDTH, upperPole.height, null);**

**g.drawImage(lowerPipeImage, lowerPole.x, lowerPole.y, PIPE\_WIDTH, lowerPole.height, null);**

**}**

**Image currentBirdImage;**

**if (isColliding) {**

**currentBirdImage = birdCollisionImage;**

**} else {**

**currentBirdImage = birdAnimationCounter % 2 == 0 ? birdImage : bird1Image;**

**}**

**g.drawImage(currentBirdImage, BIRD\_START\_X, birdY, BIRD\_WIDTH, BIRD\_HEIGHT, null);**

**g.setColor(Color.BLACK);**

**g.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 20));**

**g.drawString("Score: " + score, WIDTH / 2 - 30, 30);**

**g.drawString("High Score: " + highScore, WIDTH / 2 - 45, 60);**

**}**

**public void actionPerformed(ActionEvent e) {**

**if (!isPaused) {**

**birdY += velocity;**

**velocity += GRAVITY;**

**if (birdY > HEIGHT - BIRD\_HEIGHT) {**

**birdY = HEIGHT - BIRD\_HEIGHT;**

**}**

**if (birdY < 0) {**

**birdY = 0;**

**velocity = 0;**

**}**

**movePoles();**

**checkCollision();**

**birdAnimationCounter++;**

**repaint();**

**}**

**}**

**private void checkCollision() {**

**Rectangle bird = new Rectangle(BIRD\_START\_X, birdY, BIRD\_WIDTH, BIRD\_HEIGHT);**

**for (Rectangle pole : poles) {**

**if (bird.intersects(pole)) {**

**isColliding = true;**

**birdCollision();**

**showGameOverPopup();**

**return;**

**}**

**}**

**}**

**private void birdCollision() {**

**if (score > highScore) {**

**highScore = score;**

**}**

**isGameOver = true;**

**repaint();**

**}**

**private void showGameOverPopup() {**

**if (!gameOverPopupShownPersistent) { // Check if the popup hasn't been shown yet**

**int choice = JOptionPane.showConfirmDialog(null, "Game Over!\nYour Score: " + score +**

**"\nHigh Score: " + highScore + "\nDo you want to play again?", "Game Over", JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);**

**if (choice == JOptionPane.YES\_OPTION) {**

**resetGame();**

**} else if (choice == JOptionPane.NO\_OPTION) {**

**gameOverPopupShownPersistent = true; // Set the flag to prevent further popups**

**// Close only the game window**

**JFrame gameFrame = (JFrame) SwingUtilities.getWindowAncestor(this);**

**gameFrame.dispose();**

**}**

**}**

**}**

**public void keyPressed(KeyEvent e) {**

**if (e.getKeyCode() == KeyEvent.VK\_SPACE) {**

**if (isGameOver) {**

**resetGame();**

**} else {**

**if (!isPaused) {**

**velocity = JUMP\_FORCE;**

**isColliding = false;**

**}**

**}**

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

**if (!isGameOver) {**

**isPaused = !isPaused;**

**}**

**}**

**}**

**public void keyTyped(KeyEvent e) {}**

**public void keyReleased(KeyEvent e) {}**

**public void componentResized(ComponentEvent e) {**

**WIDTH = getWidth();**

**HEIGHT = getHeight();**

**resetGame();**

**}**

**public void componentMoved(ComponentEvent e) {}**

**public void componentShown(ComponentEvent e) {}**

**public void componentHidden(ComponentEvent e) {}**

**private void resetGame() {**

**birdY = HEIGHT / 2;**

**velocity = 0;**

**poles.clear();**

**generatePoles();**

**score = 0;**

**isGameOver = false;**

**isColliding = false;**

**isPaused = false;**

**POLE\_MOVE\_SPEED = 3.0;**

**gameOverPopupShownPersistent = false; // Reset the flag when the game is reset**

**System.out.println(gameOverPopupShownPersistent);**

**}**

**public static void main(String[] args) {**

**SwingUtilities.invokeLater(() -> {**

**Image birdImage = null;**

**Image bird1Image = null;**

**Image birdCollisionImage = null;**

**Image background = null;**

**Image upperPipeImage = null;**

**Image lowerPipeImage = null;**

**JFrame frame = new JFrame("Flappy Bird");**

**FlappyBirdGame game = new FlappyBirdGame(birdImage, bird1Image, birdCollisionImage,background, upperPipeImage, lowerPipeImage);**

**frame.add(game);**

**frame.setSize(game.WIDTH, game.HEIGHT);**

**frame.dispose();**

**frame.setVisible(true);**

**frame.setResizable(true);**

**frame.addKeyListener(game);**

**});**

**}**

**}**