****

**Final Project: Flappy Bird**

**Course Title: Object-Oriented Programming(oop)**

**Course Instructor: Prof. Shahid Bhatti**

**Section: B**

**Date: 19/12/2024**

**Group Members:**

**Names Reg No.**

Fatima Farzad sp24-bse-025

Samra Alam sp24-bse-137

Samar Saqib sp24-bse-105

**Game: Flappy Bird**

Breakdown of Javafx Implementation on game program application**:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**1. Game Structure**

* **Main Class**: FlappyBird extends Application. It initializes the game and navigates between the menu, character selection, and gameplay screens.
* **Entry Point**: The main() method invokes launch(), starting the JavaFX application.

**2. Menu System**

* **showMenu()**: Displays the main menu with options to start the game, select a character, or exit.
  + A background image is loaded and styled.
  + Three buttons (Start Game, Select Character, Exit) are styled and added to the layout.
  + Clicking Start Game begins the game, while Select Character navigates to a character selection screen.

**3. Character Selection**

* **showCharacterSelection()**: Displays a screen to choose the bird character. Six bird images are provided, and clicking one sets the selectedCharacter variable to the chosen bird and returns to the menu.

**4. Game Screen**

* **showGame()**: Sets up the game environment:
  + A background image, bird (the selected character), and a score display are added.
  + The bird starts at a fixed position with velocity initialized to 0.
  + **User Interaction**: Pressing the space bar applies a jump velocity (birdVelocity = -5).

**A screenshot of a computer game

Description automatically generated**

**A screenshot of a computer game

Description automatically generated**

**5. Game Loop**

* **AnimationTimer**: Continuously updates the game by invoking updateGame():
  1. **applyGravity()**: Gravity is applied to the bird, increasing its downward velocity over time.
  2. **handleCollision()**: Detects if the bird hits the ground or flies off-screen. Ends the game if so.
  3. **generateAndMovePipes()**:
     + Pipes are generated with a random vertical gap.
     + Pipes move leftward, simulating motion.
     + Pipes are removed once off-screen, and the score is incremented.
  4. **checkCollisionsWithPipes()**: Checks if the bird intersects any pipes. Ends the game on collision.

**6. Gameplay Dynamics**

* **Pipes**: Created with a gap and random placement. Pipes move leftward at a constant speed.
* **Bird Movement**: The bird responds to gravity unless counteracted by a spacebar press.
* **Collision Handling**:
  + Collision with pipes or boundaries triggers gameOver(), ending the game and displaying the final score.

**7. Helper Methods**

* **createImageView()**: Creates an ImageView for given image resources, ensuring images are loaded and resized appropriately.
* **updateScore()**: Updates the score display after passing pipes.

**8. Game States**

* **isGameOver**: Controls whether the game continues or ends.
* When the game ends:
  + A "Game Over" message is displayed.
  + The score is displayed in red.

**9. Features**

* **Dynamic Character Selection**: Allows choosing different bird characters, which are then used in the game.
* **Collision Detection**: The intersects() method ensures accurate hit detection between the bird and pipes.
* **Scoring System**: The score increments each time the bird successfully passes through a pipe pair.

**10. Key Variables**

* **birdVelocity**: Controls the bird's upward/downward motion.
* **pipes**: Stores all active pipes on the screen.
* **score**: Tracks the player's score.

# **Source Code:**

package com.example.demo13;  
  
import javafx.animation.AnimationTimer;  
import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.input.KeyCode;  
import javafx.scene.layout.\*;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
import java.io.InputStream;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
  
public class FlappyBird extends Application {  
  
 private static final int *WIDTH* = 800;  
 private static final int *HEIGHT* = 600;  
  
 private Pane pane;  
 private Scene gameScene;  
  
 private ImageView bird;  
 private List<ImageView> pipes = new ArrayList<>();  
 private int score = 0;  
 private Text scoreText;  
  
 private double birdVelocity = 0;  
 private boolean isGameOver = false;  
  
  
 private String selectedCharacter = "/blueBird.png";  
  
 @Override  
 public void start(Stage primaryStage) {  
 showMenu(primaryStage);  
 }  
  
 private void showMenu(Stage primaryStage) {  
 Image backgroundImage = new Image("/homeScreenBG.png");  
 ImageView backgroundImageView = new ImageView(backgroundImage);  
 backgroundImageView.setFitWidth(*WIDTH*);  
 backgroundImageView.setFitHeight(*HEIGHT*);  
 backgroundImageView.setPreserveRatio(false);  
  
 Text title = new Text("FLAPPY BIRD");  
 title.setFont(Font.*font*("Roman", 50));  
 title.setStyle("-fx-fill: Yellow; -fx-stroke: black; -fx-stroke-width: 2px;");  
  
 Button startButton = new Button("Start Game");  
 Button characterSelectButton = new Button("Select Character");  
 Button exitButton = new Button("Exit");  
  
 String buttonStyle = "-fx-background-color: linear-gradient(to bottom, #ff7f50, #ff4500); " +  
 "-fx-text-fill: white; " +  
 "-fx-font-size: 18px; " +  
 "-fx-font-weight: bold; " +  
 "-fx-border-color: black; " +  
 "-fx-border-width: 2px; " +  
 "-fx-border-radius: 10px; " +  
 "-fx-background-radius: 10px;";  
 startButton.setStyle(buttonStyle);  
 characterSelectButton.setStyle(buttonStyle);  
 exitButton.setStyle(buttonStyle);  
  
 startButton.setOnAction(e -> showGame(primaryStage));  
 //characterSelectButton.setOnAction(e -> System.out.println("Open Character Selection!"));  
 characterSelectButton.setOnAction(e -> showCharacterSelection(primaryStage));  
 exitButton.setOnAction(e -> primaryStage.close());  
  
 VBox buttonBox = new VBox(20, startButton, characterSelectButton, exitButton);  
 buttonBox.setAlignment(Pos.*CENTER*);  
  
 StackPane root = new StackPane();  
 root.getChildren().addAll(backgroundImageView, buttonBox, title);  
 StackPane.*setAlignment*(title, Pos.*TOP\_CENTER*);  
 StackPane.*setAlignment*(buttonBox, Pos.*CENTER*);  
 root.setStyle("-fx-background-color: lightblue; " +  
 "-fx-border-color: black; " +  
 "-fx-border-width: 5px; " +  
 "-fx-border-radius: 10px; " +  
 "-fx-background-radius: 10px;");  
  
 Scene menuScene = new Scene(root, *WIDTH*, *HEIGHT*);  
  
 primaryStage.setTitle("Flappy Bird Menu");  
 primaryStage.setScene(menuScene);  
 primaryStage.show();  
 }  
  
 private void showCharacterSelection(Stage primaryStage) {  
 Text title = new Text("Select Your Character");  
 title.setFont(Font.*font*("Roman", 40));  
 title.setStyle("-fx-fill: Yellow; -fx-stroke: black; -fx-stroke-width: 2px;");  
  
 ImageView char1 = createImageView("/brownBird.png", 80, 80);  
 ImageView char2 = createImageView("/whiteBird.png", 80, 80);  
 ImageView char3 = createImageView("/blueBird.png", 80, 80);  
 ImageView char4 = createImageView("/pinkBird.png", 80, 80);  
 ImageView char5 = createImageView("/greenBird.png", 80, 80);  
 ImageView char6 = createImageView("/yellowBird.png", 80, 80);  
  
  
 char1.setOnMouseClicked(e -> {  
 selectedCharacter = "/brownBird.png";  
 showMenu(primaryStage);  
 });  
 char2.setOnMouseClicked(e -> {  
 selectedCharacter = "/whiteBird.png";  
 showMenu(primaryStage);  
 });  
 char3.setOnMouseClicked(e -> {  
 selectedCharacter = "/blueBird.png";  
 showMenu(primaryStage);  
 });  
 char4.setOnMouseClicked(e -> {  
 selectedCharacter = "/pinkBird.png";  
 showMenu(primaryStage);  
 });  
 char5.setOnMouseClicked(e -> {  
 selectedCharacter = "/greenBird.png";  
 showMenu(primaryStage);  
 });  
 char6.setOnMouseClicked(e -> {  
 selectedCharacter = "/yellowBird.png";  
 showMenu(primaryStage);  
 });  
  
 HBox characterBox = new HBox(20, char1, char2, char3, char4, char5, char6);  
 characterBox.setAlignment(Pos.*CENTER*);  
  
 VBox vBox = new VBox(30, title, characterBox);  
 vBox.setAlignment(Pos.*CENTER*);  
 vBox.setStyle("-fx-background-color: lightblue;");  
  
 Scene charScene = new Scene(vBox, *WIDTH*, *HEIGHT*);  
 primaryStage.setScene(charScene);  
 }  
  
  
 private void showGame(Stage primaryStage) {  
 pane = new Pane();  
 gameScene = new Scene(pane, *WIDTH*, *HEIGHT*);  
  
 ImageView background = createImageView("/background.png", *WIDTH*, *HEIGHT*);  
 pane.getChildren().add(background);  
  
 bird = createImageView(selectedCharacter, 40, 40);  
 bird.setLayoutX(100);  
 bird.setLayoutY(*HEIGHT* / 2);  
 pane.getChildren().add(bird);  
  
 scoreText = new Text();  
 scoreText.setFill(Color.*WHITE*);  
 scoreText.setStyle("-fx-font: 24 arial;");  
 scoreText.setLayoutX(10);  
 scoreText.setLayoutY(30);  
 pane.getChildren().add(scoreText);  
  
 gameScene.setOnKeyPressed(event -> {  
 if (event.getCode() == KeyCode.*SPACE* && !isGameOver) {  
 birdVelocity = -5;  
 }  
 });  
  
 primaryStage.setScene(gameScene);  
  
 AnimationTimer timer = new AnimationTimer() {  
 @Override  
 public void handle(long now) {  
 updateGame();  
 }  
 };  
 timer.start();  
 }  
  
 private void updateGame() {  
 if (!isGameOver) {  
 applyGravity();  
 handleCollision();  
 generateAndMovePipes();  
 checkCollisionsWithPipes();  
 }  
 }  
  
 private void handleCollision() {  
 if (bird.getLayoutY() >= *HEIGHT* - bird.getFitHeight() || bird.getLayoutY() <= 0) {  
 bird.setLayoutY(Math.*min*(*HEIGHT* - bird.getFitHeight(), bird.getLayoutY()));  
 gameOver();  
 }  
 }  
  
 private void applyGravity() {  
 birdVelocity += 0.4;  
 bird.setLayoutY(bird.getLayoutY() + birdVelocity);  
 }  
  
 private void generateAndMovePipes() {  
 double pipeGap = 150;  
 double pipeWidth = 70;  
  
 if (pipes.isEmpty() || pipes.get(pipes.size() - 1).getLayoutX() < *WIDTH* - 200) {  
 double gapStartY = 150 + Math.*random*() \* (*HEIGHT* - pipeGap - 300);  
  
 double topPipeHeight = gapStartY;  
 ImageView topPipe = createImageView("/topPipe.png", pipeWidth, topPipeHeight);  
 topPipe.setLayoutX(*WIDTH*);  
 topPipe.setLayoutY(0);  
  
 double bottomPipeHeight = *HEIGHT* - gapStartY - pipeGap;  
 ImageView bottomPipe = createImageView("/bottomPipe.png", pipeWidth, bottomPipeHeight);  
 bottomPipe.setLayoutX(*WIDTH*);  
 bottomPipe.setLayoutY(gapStartY + pipeGap);  
  
 pipes.add(topPipe);  
 pipes.add(bottomPipe);  
  
 pane.getChildren().addAll(topPipe, bottomPipe);  
 }  
  
 Iterator<ImageView> iter = pipes.iterator();  
 while (iter.hasNext()) {  
 ImageView pipe = iter.next();  
 pipe.setLayoutX(pipe.getLayoutX() - 3);  
  
 if (pipe.getLayoutX() + pipe.getFitWidth() < 0) {  
 pane.getChildren().remove(pipe);  
 iter.remove();  
 score++;  
 updateScore();  
 }  
 }  
 }  
  
 private void checkCollisionsWithPipes() {  
 for (ImageView pipe : pipes) {  
 if (bird.getBoundsInParent().intersects(pipe.getBoundsInParent())) {  
 gameOver();  
 }  
 }  
 }  
  
 private void gameOver() {  
 isGameOver = true;  
 scoreText.setText("Game Over! Final Score: " + score);  
 scoreText.setFill(Color.*RED*);  
  
 }  
  
 private void updateScore() {  
 scoreText.setText("Score: " + score);  
 }  
  
  
  
 private ImageView createImageView(String resourcePath, double width, double height) {  
 InputStream stream = getClass().getResourceAsStream(resourcePath);  
  
 if (stream != null) {  
 Image image = new Image(stream);  
 ImageView imageView = new ImageView(image);  
 imageView.setFitWidth(width);  
 imageView.setFitHeight(height);  
 return imageView;  
 } else {  
 System.*err*.println("Resource not found: " + resourcePath);  
 return new ImageView();  
 }  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}