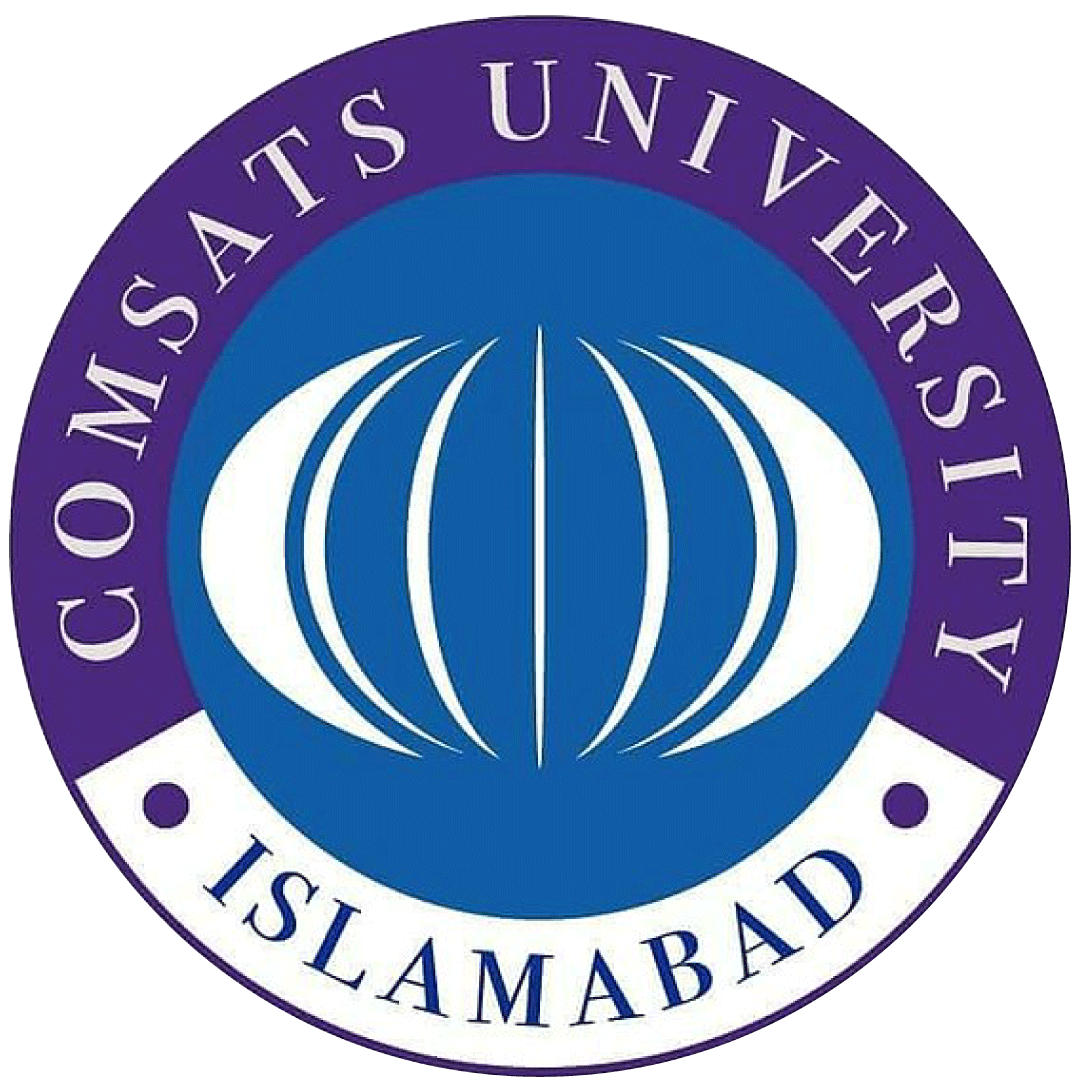
****

**Assignment 3**

**Group Member:**

Fatima Farzad (sp24-bse-025)

Samra Alam (sp24-bse-137)

Samar Saqib(sp24-bse-105)

**Course Title:**

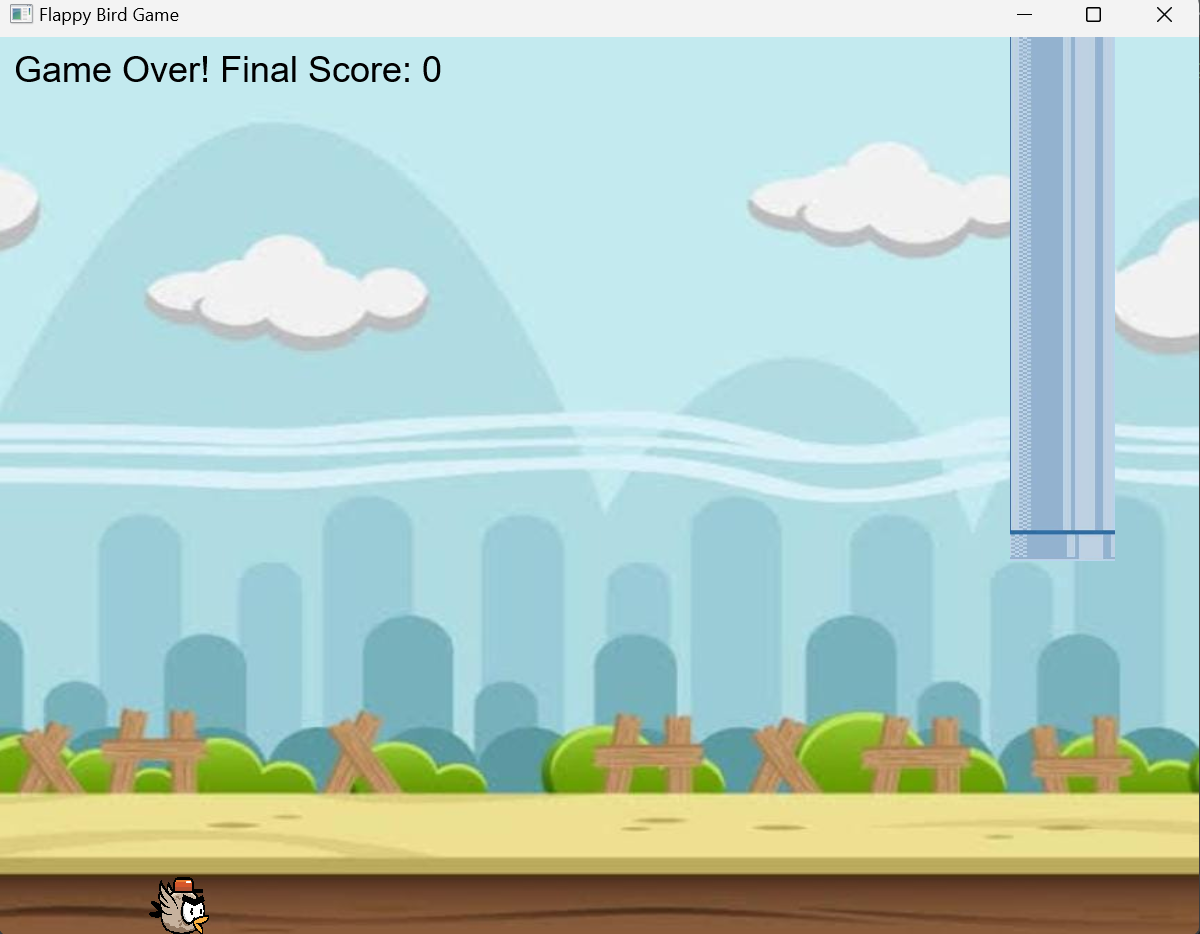
Object Oriented Programming\

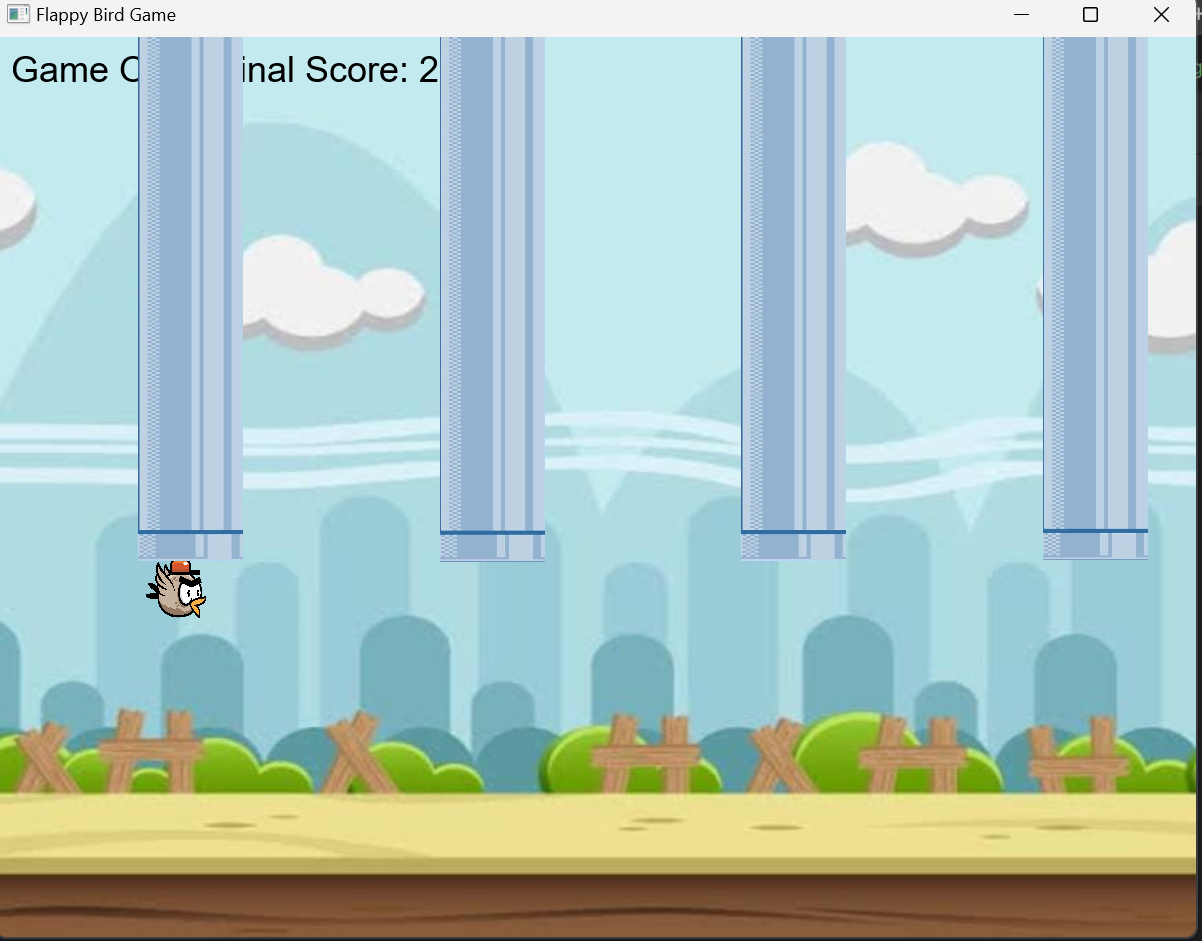
**Assignment Name:**

Flappy Bird

**OUTPUT:**







**Explanation of code**

**1. Class Overview**

* **Class Name**: FlappyBird/MenuGame
* **Extends**: Application, enabling the use of JavaFX framework.
* **Purpose**: Combines a menu screen with gameplay, allowing users to navigate between the two.

**2. Main Components**

**Constants**

* **WIDTH** and **HEIGHT**: Define the dimensions of the game window (800x600).

**Game Elements**

* **Pane gameRoot**: The root layout for the game screen.
* **Scene gameScene**: The primary scene for the game.
* **ImageView bird**: Represents the bird character.
* **List<ImageView> pipes**: Holds the pipe obstacles.
* **Text scoreText**: Displays the player's score.
* **birdVelocity**: Manages the bird's vertical motion.
* **isGameOver**: Tracks the game's status.

**3. start Method**

* **Entry Point**: The start method initializes the application.
* **Initial Action**: Calls showMenu(primaryStage) to display the main menu.

**4. Menu Screen (showMenu)**

* **Background**: An image fills the menu screen.
* **Title**: Displays the text "FLAPPY BIRD" with custom styling.
* **Buttons**:
  + **Start Game**: Calls showGame to start gameplay.
  + **Select Character**: Currently prints a placeholder message.
  + **Exit**: Closes the application.
* **Layout**: Uses VBox for vertically aligned buttons and StackPane to combine the title and background.

**5. Game Screen (showGame)**

* **Root Layout**: Pane contains all game elements.
* **Background**: Adds a background image to the game.
* **Bird**: Places the bird at an initial position (100px, vertically centered).
* **Score Display**: A Text element in the top-left corner shows the score.
* **Back to Menu Button**: Allows navigation back to the menu.
* **Key Input**: Listens for SPACE key presses to make the bird jump.
* **Game Loop**: Uses an AnimationTimer to continuously update the game state.

**6. Gameplay Logic**

**updateGame**

* Handles core game updates:
  + Applies gravity to the bird.
  + Manages pipe generation and movement.
  + Checks for collisions.

**Gravity (applyGravity)**

* Simulates downward motion by increasing birdVelocity.
* Ends the game if the bird touches the ground.

**Pipe Management (generateAndMovePipes)**

* Dynamically generates top and bottom pipes.
* Moves pipes leftward.
* Removes off-screen pipes and increments the score.

**Collision Detection**

* Checks if the bird collides with a pipe or the ground, ending the game.

**Game Over**

* Stops gameplay by setting isGameOver to true.
* Displays the final score in the score text.

**7. Helper Methods**

**createImageView**

* Loads an image resource and returns an ImageView with specified dimensions.
* Logs a message if the resource is not found.

**updateScore**

* Updates the score text on the screen.

**8. User Interface Features**

* **Menu Styling**:
  + Buttons have a gradient background and a bold border.
  + The menu layout is visually distinct with a border and rounded corners.
* **Game Styling**:
  + Uses a plain game background and simple score display.
* **Navigation**:
  + Users can seamlessly switch between the menu and the game.

**9. Future Improvements**

* Implement the **Select Character** functionality.
* Add sound effects and animations for a more engaging experience.
* Include a restart option after the game ends.

**Flow of the Application**

1. **Start Menu**:
   * Displayed when the application launches.
   * Users can start the game, select a character, or exit.
2. **Gameplay**:
   * Bird flies through pipes; the player earns points by avoiding collisions.
   * Gravity and pipe movement create the challenge.
3. **Game Over**:
   * Displays the final score.
   * Players can return to the main menu.

**Resource code**

**FlappyBirdMenu**

package com.example.oopproject2;

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.layout.\*;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class **FlappyBirdMen** extends Application {

@Override

public void start(Stage primaryStage) {

Image backgroundImage = new Image("Screenshot 2024-12-12 144512.png");

ImageView backgroundImageView = new ImageView(backgroundImage);

backgroundImageView.setFitWidth(800);

backgroundImageView.setFitHeight(600);

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 -> openGameScreen(primaryStage));

characterSelectButton.setOnAction(e -> System.out.println("Open Character Selection!"));

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 scene = new Scene(root, 800, 600);

primaryStage.setTitle("Flappy Bird Menu");

primaryStage.setScene(scene);

primaryStage.show();

}

private void openGameScreen(Stage primaryStage) {

Text gameTitle = new Text("Game Screen");

gameTitle.setFont(Font.font("Roman", 30));

gameTitle.setStyle("-fx-fill: white; -fx-stroke: black; -fx-stroke-width: 1px;");

Button exitButton = new Button("Exit");

exitButton.setStyle("-fx-background-color: red; -fx-text-fill: white; -fx-font-weight: bold;");

exitButton.setOnAction(e -> primaryStage.close());

Button pauseButton = new Button("Pause");

pauseButton.setStyle("-fx-background-color: orange; -fx-text-fill: white; -fx-font-weight: bold;");

pauseButton.setOnAction(e -> System.out.println("Game Paused"));

HBox topControls = new HBox(10, pauseButton, exitButton);

topControls.setAlignment(Pos.TOP\_RIGHT);

topControls.setStyle("-fx-padding: 10px;");

StackPane gameRoot = new StackPane();

gameRoot.getChildren().addAll(gameTitle, topControls);

StackPane.setAlignment(gameTitle, Pos.TOP\_CENTER);

StackPane.setAlignment(topControls, Pos.TOP\_RIGHT);

gameRoot.setStyle("-fx-background-color: darkgreen; " + "-fx-border-color: black; " + "-fx-border-width: 5px; " +

"-fx-border-radius: 10px; " +

"-fx-background-radius: 10px;");

Scene gameScene = new Scene(gameRoot, 800, 600);

primaryStage.setScene(gameScene);

}

public static void main(String[] args) {

launch(args);

}

}

import javafx.animation.AnimationTimer;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.Pane;

import javafx.scene.text.Text;

import javafx.stage.Stage;

import javafx.scene.input.KeyCode;

import javafx.scene.paint.Color;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class **FlappyBirdMenu** extends Application {

private static final int WIDTH = 800;

private static final int HEIGHT = 600;

private Pane pane;

private Scene scene;

private ImageView bird;

private List<ImageView> pipes = new ArrayList<>();

private int score = 0;

private Text scoreText;

private double birdVelocity = 0;

private boolean isGameOver = false;

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Flappy Bird Game");

pane = new Pane();

scene = new Scene(pane, WIDTH, HEIGHT);

ImageView background = createImageView("/img\_3.png", WIDTH, HEIGHT);

pane.getChildren().add(background);

bird = createImageView("/flappy-bird-character.png", 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);

scene.setOnKeyPressed(event -> {

if (event.getCode() == KeyCode.SPACE && !isGameOver) {

birdVelocity = -8;

}

});

primaryStage.setScene(scene);

primaryStage.show();

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() {

}

private void applyGravity() {

birdVelocity += 0.3;

bird.setLayoutY(bird.getLayoutY() + birdVelocity);

if (bird.getLayoutY() >= HEIGHT - bird.getFitHeight()) {

bird.setLayoutY(HEIGHT - bird.getFitHeight());

gameOver();

}

}

private void generateAndMovePipes() {

if (pipes.isEmpty() || pipes.get(pipes.size() - 1).getLayoutX() < WIDTH - 200) {

ImageView topPipe = createImageView("/img\_2.png", 70, Math.random() \* 10);

ImageView bottomPipe = createImageView("/img\_1.png", 70, HEIGHT - (Math.random() \* 2 + 250));

topPipe.setLayoutX(WIDTH);

bottomPipe.setLayoutX(WIDTH);

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 moveBird(int x, int y) {

bird.setLayoutX(x);

bird.setLayoutY(y);

}

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.gray(0.0));

}

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);

}

}

**Pipes Class**

package com.example.oopproject2;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

public class Pipe2 {

private ImageView topPipe;

private ImageView bottomPipe;

private double x;

private double speed;

private boolean passed;

public Pipe2(Image topImage, Image bottomImage, double x, double gapY, double gapHeight, double pipeWidth) {

this.x = x;

this.speed = 2; // Adjust speed if needed

this.passed = false;

// Top Pipe

topPipe = new ImageView(topImage);

topPipe.setFitWidth(pipeWidth);

topPipe.setPreserveRatio(false);

topPipe.setTranslateX(x);

topPipe.setTranslateY(gapY - topPipe.getImage().getHeight());

// Bottom Pipe

bottomPipe = new ImageView(bottomImage);

bottomPipe.setFitWidth(pipeWidth);

bottomPipe.setPreserveRatio(false);

bottomPipe.setTranslateX(x);

bottomPipe.setTranslateY(gapY + gapHeight);

}

public void update() {

x -= speed;

topPipe.setTranslateX(x);

bottomPipe.setTranslateX(x);

}

public boolean isOffScreen(double screenWidth) {

return x + topPipe.getFitWidth() < 0; }

public boolean isCollision(double birdX, double birdY, double birdWidth, double birdHeight) {

return (birdX + birdWidth > x && birdX < x + topPipe.getFitWidth()) &&

(birdY < topPipe.getTranslateY() + topPipe.getFitHeight() ||

birdY + birdHeight > bottomPipe.getTranslateY());

}

public ImageView getTopPipe() {

return topPipe;

}

public ImageView getBottomPipe() {

return bottomPipe;

}

public boolean isPassed() {

return passed;

}

public void setPassed(boolean passed) {

this.passed = passed;

}

public void setSpeed(double speed) {

this.speed = speed;}

}

package com.example.oopproject2;

public class **OnMouseClickedListener** {

public OnMouseClickedListener(FlappyBirdMenu flappyBirdMenu, Object p1) {

}

}