



AKTU COLLEGE CODE - 047



Dr. Virendra Swarup Institute of Computer Studies

(An Institute of Computer Application & Management)

Project Report

ON

Game Galaxy

**Master of Computer Application
(MCA 2022-2024)**



Submitted To -

Mr. Rahul Agnihotri
Head of Department
(M.C.A)

Submitted By-

Shivam Prajapati
2200470140046
Semester - IV

CONTENTS

SERIAL NO.	DESCRIPTION	PAGE NO.
1	Acknowledgement	3
2	Training Certificate	4
3	Plagiarism Report	5-8
4	Title & Introduction & Objective of Project	9-11
5	Project Category, Tools & Platform used & Technical Details	12
6	Hardware and Software Requirement	13
7	Feasibility Study	14
8	System Design	15-16
9	Module Description	17-18
10	Database Design	19-20
11	Data Flow Diagram	21-22
12	ER Diagram	23-24
13	Coding	25-104
14	Screenshots	105-113
15	Testing	114-115
16	Implementation and Maintenance	116
17	System Security Measure	117
18	Future Scope of the Project	118
19	Glossary	119
20	Bibliography	120

Acknowledgment

I would like to express my heartfelt gratitude to all those who have contributed to the completion of this project. Their support and encouragement have been invaluable throughout this journey.

First and foremost, I am deeply indebted to my HOD, Mr. Rahul Agnihotri sir for their unwavering guidance, patience, and encouragement. Their expertise and insights have played a pivotal role in shaping the direction of this project and refining its outcomes. I am truly grateful for their mentorship and support.

I extend my sincere thanks to the faculty members of the MCA department for their continuous support and encouragement. Their dedication to fostering academic excellence has been a constant source of inspiration.

I would like to acknowledge the assistance and cooperation of my peers and friends. Their valuable feedback, discussions, and moral support have been immensely helpful in overcoming challenges and staying motivated throughout this endeavour.

In conclusion, I am profoundly grateful to everyone who has contributed to this project, directly or indirectly. Your support has been instrumental in its success, and I am truly thankful for the opportunity to undertake this endeavour.

Mr. Rahul Agnihotri
[HOD-MCA]

Mr.Kamlesh C. Sharma
[Project Coordinator]

TRAINING CERTIFICATE

ID: E87B6FD46EDF



Certificate

This is to certify that

Shivam Prajapati

has successfully cleared the assessment for the skill

Java (Basic)

20 Jan, 2023

Date

J. Harishankaran
Harishankaran K

CTO, HackerRank



CERTIFICATE OF COMPETITION



Awarded To

Shivam Prajapati

For Successfully Completing Training on **JAVA WITH MYSQL**
from **01-04-2023** to **30-04-2023**

Conducted at **VSICS, Saket Nagar, Kanpur** with grade A.

K.M. Mishra
K.M. Mishra
Training and Placement Head

Rahul Agnihotri
RAHUL AGNIHOTRI
HOD (M.C.A.)

On behalf of KTCS INDIA
Vikash Dnivedi
Vikash Dnivedi
FOUNDER & DIRECTOR



Microsoft

Our Partners



PLAGIARISM REPORT



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism

SYSTEM DESIGN

System Design is the solution to the creation of a new system. **This is the important aspect made up of several steps.** The complete, efficient and successful system should provide the following in succession: -

- * From where should we start
- * Where we have to go
- * Where should we stop

The steps for the successful project are as follows -

1) We should define problem completely and the goals should be known before our destination In the next step, we should specify inputs and outputs of our interest, then the

structure of various database should be designed which will be used during the programming

2) Next, we should design our programs of user friendly nature and always provide a way to the **user to read back the origin if he/she find any complex problem at any stage**

We should know the function of each and every program which will leads us to or helps us to read at the specified goal.

3) Then we write these individual programs which later on joining solve our problem Next step involve then testing of these programs and correction – if necessary

4) At last, linking all the programs in a well-specified manner and combining in the form of a menu, sub menu etc. will be our defined problem.

Out of these defined steps, few of the major steps with respect to Project "Game Galaxy"

MODULE DESCRIPTION

This project includes the following modules for development of the project. These are as follows:

1) Game Engine:

Develop the core game engine responsible for managing game states, rendering graphics, handling input, and maintaining game logic.

2) Graphics and Animation:

Create modules for rendering graphics, including character sprites, backgrounds, and animations. Implement animation systems if needed.

3) User Interface (UI):

Design and implement user interfaces for menus, HUD (Heads-Up Display), and in-game UI components.

4) Game Logic:

Develop the logic for game rules, character behavior, enemy AI, and win/lose conditions.

5) Input Handling:

Create a module to manage user input, including keyboard, mouse, and game controller support.

6) Level Design:

Create tools or modules for designing game levels, including level layout, obstacles, collectibles, and objectives.

DATABASE DESIGN

Database Design is most important in any project. We are using the following table to store the Player Info Like Player Name and its Score Obtained in Respective Game in order to show High Score to player Winners of Previous Players and to keep track of number of times a Particular game is Played

DATABASE NAME :

- GameGalaxy

DATABASE TABLES :

- Snake_Game
- Pong_Game
- Java_Quiz_Game
- Tic_Tac

TABLE STRUCTURE

1) SNAKE_GAME : 2) JAVA QUIZ GAME :

2) PONG_GAME : 4) TIC_TAC :

DATA FLOW DIAGRAM

DFD represents a system at different level of abstraction DFD may be partitioned into levels that represent increasing information flow and functional details as increasing level

1) CONTEXT LEVEL DFD (LEVEL - 0 DFD) :

It represents entire system as single bubble and provides an overall picture of system

2) LEVEL 1 DFD :

It represents main Modules and how they interact with each other

ER DIAGRAM

We can express the overall logical structure of a database graphically with an E-R diagram. Its components are:

1) Rectangles representing entity sets.

2) Ellipse representing Attributes

3) Diamond representing Relationship Set

INTRODUCTION

Game Galaxy is an exciting and entertaining 2D platform game developed in Java, designed to showcase creative game play and engaging visuals.

There are many ways human can entertain themselves like Movie, Sports, Games etc. Many people love playing games there was a time when the games were physically played but these days after the introduction of Computers, Mobile, Play Station & Nintendo Switch etc...

Now you can have fun where/when ever you want on your PC, mobile etc.

This Project was our personal choice these days the 2D games are not much famous. We Created the game just because I wanted to explore the Graphical part of JAVA so we created the one on our own

This game include

Single Player Game :

- 1) Snake Game
- 2) Java Quiz Game

& Multi Player Game :

- 1) Tic Tac Toe
- 2) Pong Game

OBJECTIVES

The primary objective is to create a captivating 2D game using Java. Implement diverse game play elements, such as jumping, running, Showcase the importance of level design in enhancing player engagement and progression

These Games is Created to have FUN in your spare time

Playing video game fulfills a purpose ...

This could include gaming for:

- If you want to relax
- Entertainment
- Social interaction
- Stress Relaxation
- To Prevent boredom
- Challenges & Achievements

PROJECT CATEGORY

This project as title "Game Galaxy" is comes under the 2D Game Development Project. This application is developed with the help of Java Swing , Java FX and MYSQL DataBase. It can also categorized for Java Game development.

TOOLS/PLATFORM

This game is developed to run specifically on Windows

JavaFX (software platform for creating and delivering desktop applications as well as rich web applications that can run across a wide variety of devices)

Java Swing (API for providing a graphical user interface for Java programs)

MYSQL & JDBC (For Database Storage as Back end)

TECHNICAL DETAILS

Utilize Java for game development, leveraging Java's 2D graphics libraries.

Implement object-oriented programming principles for code organization and maintainability.

Employ Java's event-driven programming model for user input and game logic.

HARDWARE & SOFTWARE REQUIREMENT

SOFTWARE:

- Windows 7 or higher
- JRE Installed

HARDWARE :

- Laptop or Desktop holding Core 2 Duo or above processor

- 2 GB RAM or above

100MB Space

Matched Source

Similarity 25%

Title:TITLE OF THE PROJECT "School Management System - Academia.edu

Ø This is the important aspect made up of several steps ✓ . The complete, efficient and successful system should provide the following in succession: From where should we start Where we have to go Where should we stop If the project is to be successful, we will need answer these question.

https://www.academia.edu/7414819/TITLE_OF_THE_PROJECT_School_Management_System_INTRODUCTION

Similarity 13%

Title:408408639-School-Dbms - TITLE OF THE PROJECT " ...

Then the structure of various database should be designed which will be used during the programming. Next, we should design our programs of user ...

<https://www.studocu.com/in/document/lovely-professional-university/dbms/408408639-school-dbms/38924231>

Similarity 13%

Title:Synopsis Sample pdf - TITLE OF THE PROJECT "School ... - Studocu

We should define problem completely and the goals should be known before our destination In the next step, we should specify inputs and outputs of our interest Then the structure of various database should be designed which will be used during the programming Next, we should design our programs of user friendly nature and always provide a way to the user to read back the origin if he/she find ...

<https://www.studocu.com/row/document/takoradi-technical-university/information-technology/synopsis-sample-pdf/46764896>

Similarity 7%

Title:[www.studocu.com > row > documentSynopsis Sample pdf - TITLE OF THE PROJECT ... - Studocu](https://www.studocu.com/row/document/Synopsis Sample pdf - TITLE OF THE PROJECT ... - Studocu)

This project includes the following modules for development of the project. These are as follows: - SPLASH FORM This is a first form that displays the welcome screen for the user and also shows the information of developer or version etc.

<https://www.studocu.com/row/document/takoradi-technical-university/information-technology/synopsis-sample-pdf/46764896/>

Similarity 4%

Title:[The Entity-Relationship Model](https://www2.cs.sfu.ca/CourseCentral/354/zaiane/material/postscript/Chapter2.pdf)

<https://www2.cs.sfu.ca/CourseCentral/354/zaiane/material/postscript/Chapter2.pdf>

Similarity 4%

Title:

[www.slideshare.net > MuhammadYaseenSheikh1 > projectProject synopsis on java game development | PPT - SlideShare](https://www.slideshare.net/MuhammadYaseenSheikh1/projectProject synopsis on java game development | PPT - SlideShare)

Jun 24, 2021 • Now you can have fun where/when ever you want on your PC, Mobile etc. • This project was my personal choice these days the 2D games are not much famous. • I created the game just because I wanted to explore the Graphical part of JAVA and of course I played many games as a child.

TITLE OF THE PROJECT

“Game Galaxy”

INTRODUCTION

Game Galaxy is an exciting and entertaining 2D platform game developed in Java, designed to showcase creative game play and engaging visuals.

There are many ways human can entertain themselves like Movie, Sports, Games etc. Many people love playing games there was a time when the games were physically played but these days after the introduction of Computers, Mobile, Play Station & Nintendo Switch etc...

Now you can have fun where/when ever you want on your PC, mobile etc.

This Project was our personal choice these days the 2D games are not much famous. We Created the game just because I wanted to explore the Graphical part of JAVA so we created the one on our own

This game include

Single Player Game :

- 1) Snake Game
- 2) Java Quiz Game

& Multi Player Game :

- 1) Tic Tac Toe
- 2) Pong Game

OBJECTIVES

The primary objective is to create a captivating 2D game using Java. Implement diverse game play elements, such as jumping, running, Showcase the importance of level design in enhancing player engagement and progression

These Games is Created to have FUN in your spare time

Playing video game fulfills a purpose ...

This could include gaming for:

- If you want to relax
- Entertainment
- Social interaction
- Stress Relaxation
- To Prevent boredom
- Challenges & Achievements

PROJECT CATEGORY

This project as title “**Game Galaxy**” is comes under the **2D Game Development Project**. This application is developed with the help of Java Swing , Java FX and MYSQL DataBase. It can also categorized for Java Game development.

TOOLS/PLATFORM

This game is developed to run specifically on Windows

JavaFX (software platform for creating and delivering desktop applications as well as rich web applications that can run across a wide variety of devices)

Java Swing (API for providing a graphical user interface for Java programs)

MYSQL & JDBC (For Database Storage as Back end)

TECHNICAL DETAILS

Utilize Java for game development, leveraging Java's 2D graphics libraries.

Implement object-oriented programming principles for code organization and maintainability.

Employ Java's event-driven programming model for user input and game logic.

HARDWARE & SOFTWARE REQUIREMENT

SOFTWARE:

- Windows 7 or higher
- JRE Installed
- MySQL

HARDWARE :

- Laptop or Desktop holding Core 2 Duo or above processor
- 2 GB RAM or above
- 100MB Space

FEASIBILITY STUDY

The feasibility study for the "Game Galaxy" project involves evaluating the project from various perspectives to ensure that it is viable, achievable, and worth pursuing. The key areas considered in this feasibility study are technical feasibility, economic feasibility, operational feasibility, and legal feasibility.

Technical Feasibility

Objective: To assess whether the technology and tools required for developing "Game Galaxy" are available and adequate.

Economic Feasibility

Objective: To determine whether the financial resources required to develop and maintain "Game Galaxy" are within budget constraints.

Operational Feasibility

Objective: To evaluate whether the project can be successfully integrated and used in the intended environment.

Legal Feasibility

Objective: To ensure that the project complies with all relevant legal and regulatory requirements.

Overall Conclusion

Based on the evaluations conducted in the feasibility study, the "Game Galaxy" project is considered technically, economically, operationally, and legally feasible. The project is viable and worth pursuing, given the available resources, technologies, and the potential benefits it offers.

SYSTEM DESIGN

System Design is the solution to the creation of a new system. This is the important aspect made up of several steps. The complete, efficient and successful system should provide the following in succession: -

- * From where should we start
- * Where we have to go
- * Where should we stop

If the project is to be successful, we will need answer these question. The answer of these questions is schema manner and is known as system design.

A systematic manner will be followed so as to achieve beneficial result at the end. It involves starting with a vague idea and ultimately developing it up into a useful system. The design phase is transition from a user oriented to a document oriented to the programmers.

Software report can be broken into a series of steps starting with the basic ideas and ending with the finished project.

The steps for the successful project are as follows -

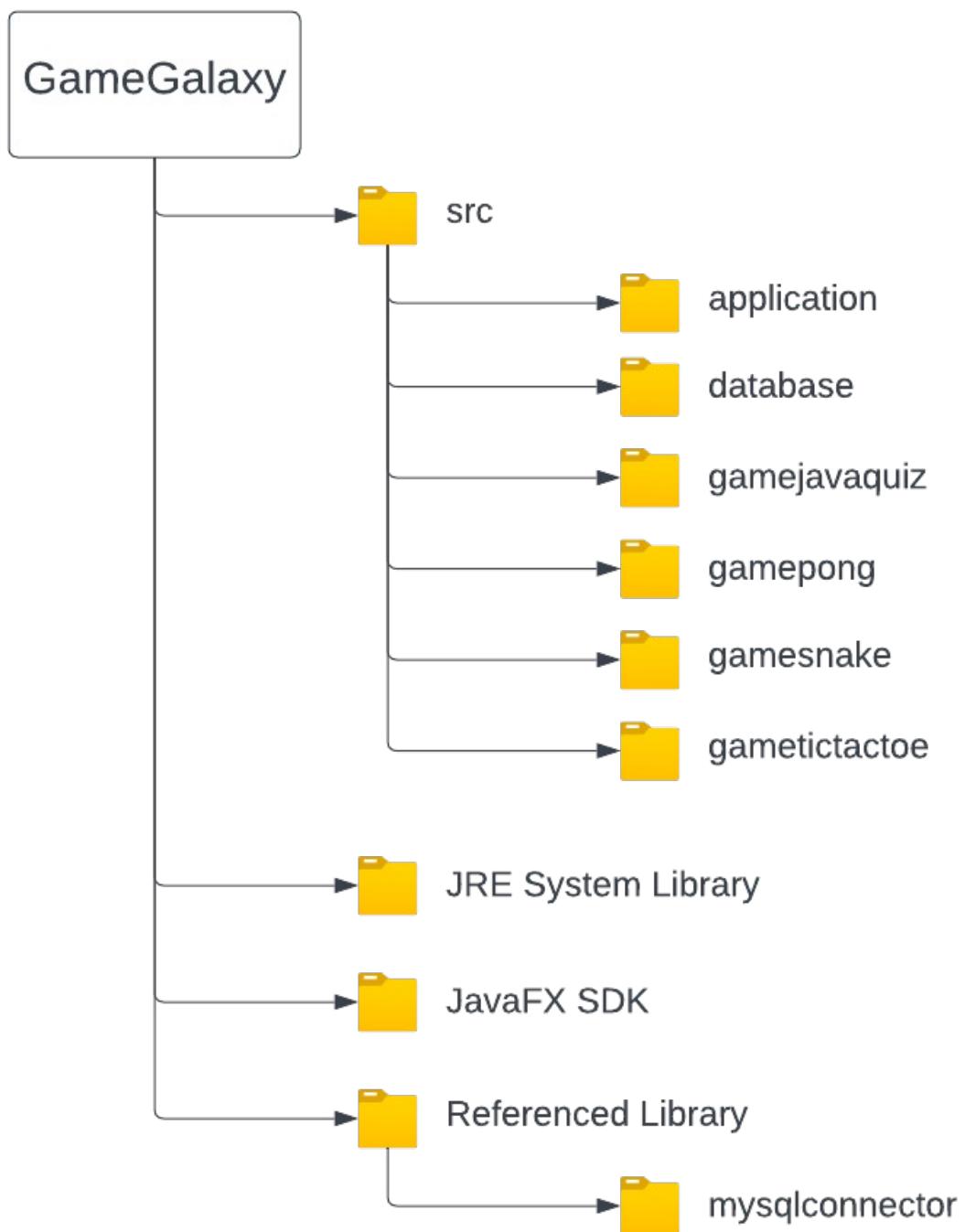
- 1) We should define problem completely and the goals should be known before our destination
In the next step, we should specify inputs and outputs of our interest, then the structure of various database should be designed which will be used during the programming
- 2) Next, we should design our programs of user friendly nature and always provide a way to the user to read back the origin if he/she find any complex problem at any stage
We should know the function of each and every program which will leads us to or helps us to read at the specified goal.
- 3) Then we write these individual programs which later on joining solve our problem Next step involve then testing of these programs and correction – if necessary

4) At last, linking all the programs in a well-specified manner and combining in the form of a menu, sub menu etc. will be our defined problem.

Out of these defined steps, few of the major steps with respect to Project

“Game Galaxy”

MODULE DESCRIPTION



MODULES USED IN THE PROJECT

This project includes the following modules for development of the project. These are as follows:

1) Game Engine:

Develop the core game engine responsible for managing game states, rendering graphics, handling input, and maintaining game logic.

2) Graphics and Animation:

Create modules for rendering graphics, including character sprites, backgrounds, and animations. Implement animation systems if needed.

3) User Interface (UI):

Design and implement user interfaces for menus, HUD (Heads-Up Display), and in-game UI components.

4) Game Logic:

Develop the logic for game rules, character behavior, enemy AI, and win/lose conditions.

5) Input Handling:

Create a module to manage user input, including keyboard, mouse, and game controller support.

6) Level Design:

Create tools or modules for designing game levels, including level layout, obstacles, collectibles, and objectives.

DATABASE DESIGN

Database Design is most important in any project. We are using the following table to store the Player Info Like Player Name and its Score Obtained in Respective Game in order to show High Score to player Winners of Previous Players and to keep track of number of times a Particular game is Played

DATABASE NAME :

- **GameGalaxy**

DATABASE TABLES :

- **Snake_Game**
- **Pong_Game**
- **Java_Quiz_Game**
- **Tic_Tac**

TABLE STRUCTURE

1) SNAKE_GAME :

Snake_Game	
Field	Type
Player_ID	INT AUTO INC
Player_NAME	VARCHAR(30)
SCORE	INT
DATE	DATE

2) JAVA QUIZ GAME :

JAVA QUIZ GAME	
Field	Type
Player_ID	INT AUTO INC
Player_NAME	VARCHAR(30)
SCORE	INT
DATE	DATE

2) PONG_GAME :

PONG_GAME	
Field	Type
Player_ID	INT AUTO INC
Player_1	VARCHAR(30)
Player_2	VARCHAR(30)
Player1_Score	INT
Player2_Score	INT
Winner	VARCHAR(30)
Date	DATE

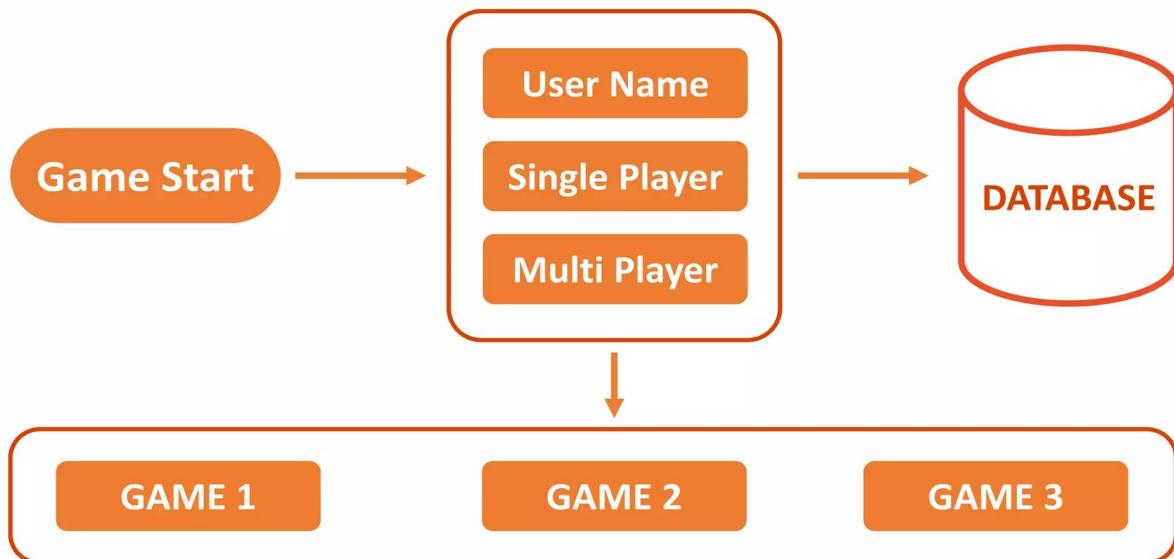
4) TIC_TAC :

TIC_TAC	
Field	Type
ID	INT AUTO INC
Player_1	VARCHAR(30)
Player_2	VARCHAR(30)
Winner	VARCHAR(30)
Date	DATE

DATA FLOW DIAGRAM

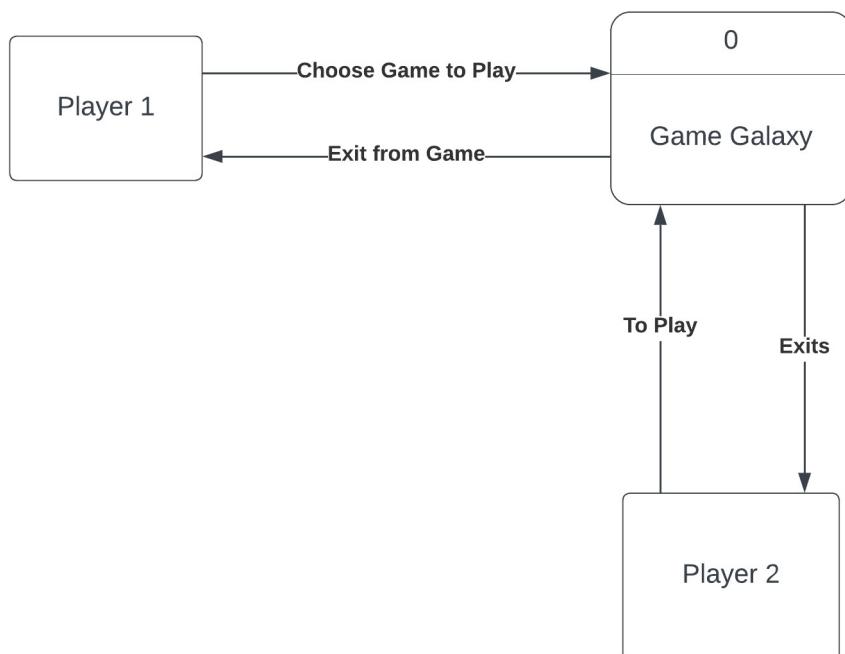
DFD represents a system at different level of abstraction DFD may be partitioned into levels that represent increasing information flow and functional details as increasing level.

FOR EXAMPLE :



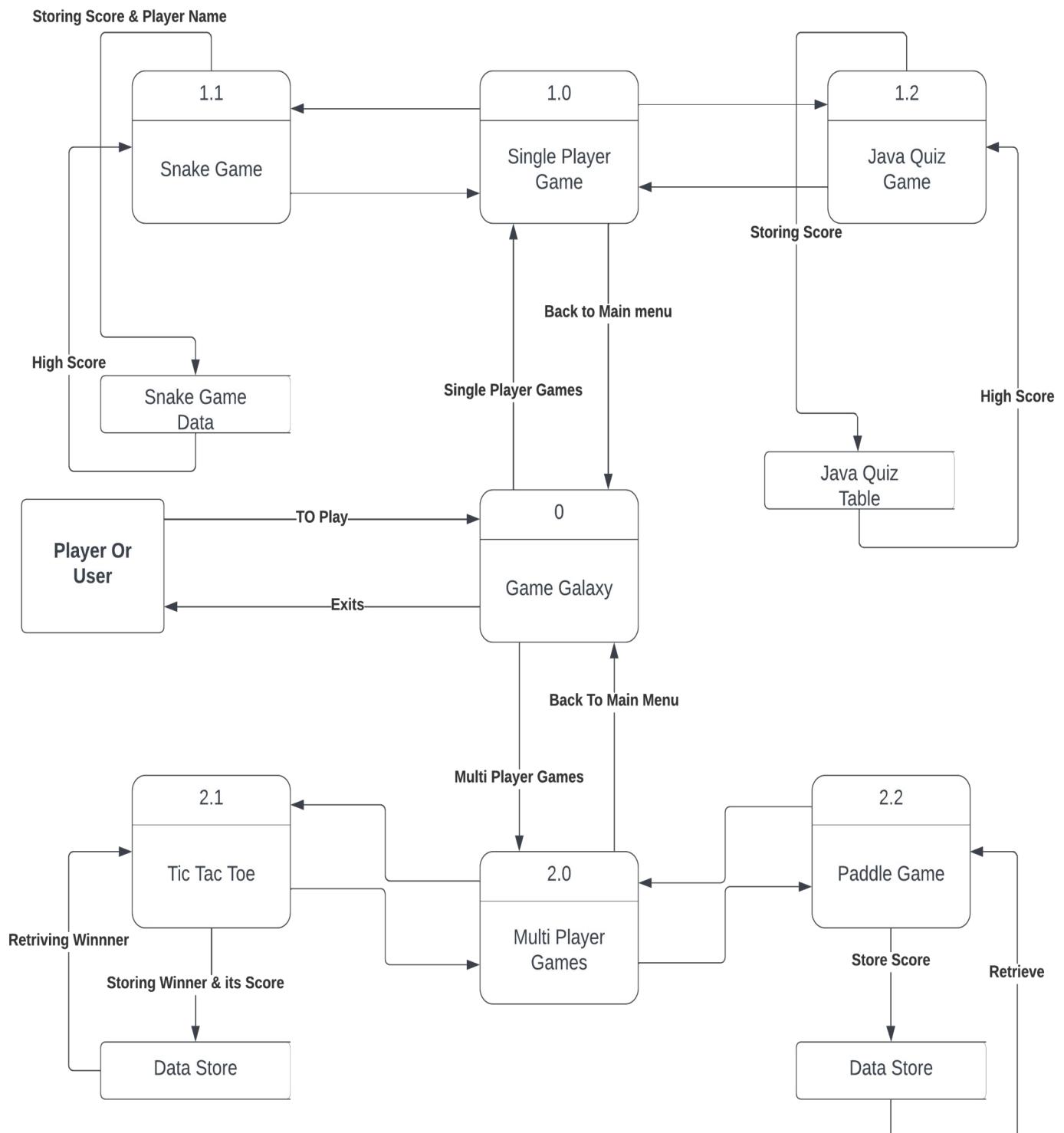
1) CONTEXT LEVEL DFD (LEVEL - 0 DFD) :

It represents entire system as single bubble and provides an overall picture of system



2) LEVEL 1 DFD :

It represent main Modules and how they interact with each other



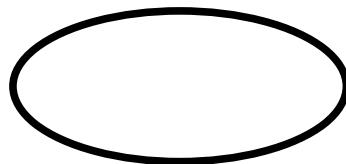
ER DIAGRAM

We can express the overall logical structure of a database graphically with an E-R diagram. Its components are:

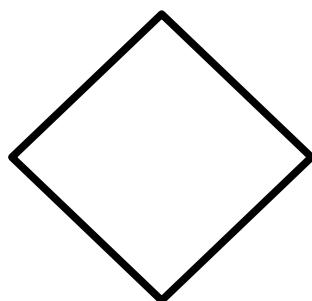
- 1) **Rectangles** representing entity sets.



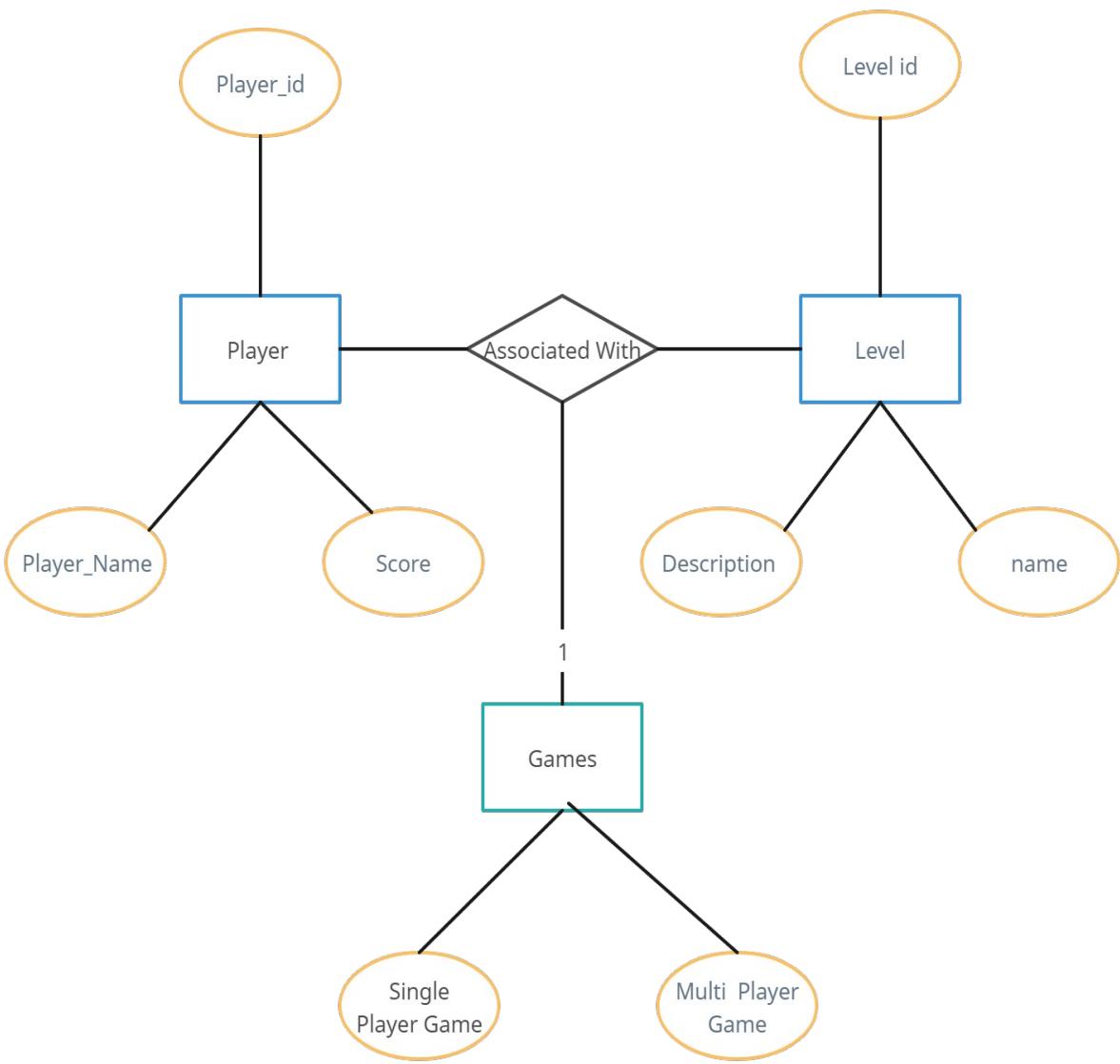
- 2) **Ellipse** representing Attributes



- 3) **Diamond** representing Relationship Set



ER DIAGRAM OF OUR PROJECT



CODING

PACKAGE 1 : application

Program 1 : application\Main.java;

```
package application;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.stage.Stage;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.image.Image;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            Parent root = FXMLLoader.load(getClass().getResource
                    ("GameGalaxyMainPage.fxml"));

            // Set application icon
            Image icon = new Image(getClass().getResourceAsStream
                    ("gameicon2.jpg"));
            primaryStage.getIcons().add(icon);

            primaryStage.setTitle("Game Galaxy");
            primaryStage.setResizable(false);
            primaryStage.setScene(new Scene(root));
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Program 2 : application\JavaQuizGamePageController.java

```
package application;

import java.io.IOException;

import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class JavaQuizGamePageController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToPlayJavaQuizGame(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
("PlayJavaQuizGame.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void displayJavaQuizGameScore(ActionEvent event) throws IOException {
    }

    public void switchToSinglePlayerPage(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
("SinglePlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }
}
```

Program 3 : application\SceneController.java

```
package application;

import java.io.IOException;

import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class SceneController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToSinglePlayerPage(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
("SinglePlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void switchToTwoPlayerPage(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
("TwoPlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void switchToSnakeGamePage(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
("SnakeGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void switchToTicTacToeGamePage(ActionEvent event) throws IOException {
```

```

root = FXMLLoader.load(getClass().getResource
        ("TicTacToePage.fxml"));
stage = (Stage) ((Node) event.getSource()).getScene() .
        getWindow();
scene = new Scene(root);
stage.setScene(scene);
stage.setResizable(false);
stage.show();
}

public void switchToPongGamePage(ActionEvent event) throws
IOException {
    root = FXMLLoader.load(getClass().getResource
        ("PongGamePage.fxml"));
    stage = (Stage) ((Node) event.getSource()).getScene() .
        getWindow();
    scene = new Scene(root);
    stage.setScene(scene);
    stage.setResizable(false);
    stage.show();
}

public void switchToJavaQuizGamePage(ActionEvent event) throws
IOException {
    root = FXMLLoader.load(getClass().getResource
        ("JavaQuizGamePage.fxml"));
    stage = (Stage) ((Node) event.getSource()).getScene() .
        getWindow();
    scene = new Scene(root);
    stage.setScene(scene);
    stage.setResizable(false);
    stage.show();
}

public void switchToGameGalaxy MainPage(ActionEvent event) throws
IOException {
    root = FXMLLoader.load(getClass().getResource
        ("GameGalaxyMainPage.fxml"));
    stage = (Stage) ((Node) event.getSource()).getScene() .
        getWindow();
    scene = new Scene(root);
    stage.setScene(scene);
    stage.setResizable(false);
    stage.show();
}

public void exitGame(ActionEvent event) throws IOException {
    Stage stage = (Stage) ((Node) event.getSource()).getScene() .
        getWindow();
    stage.close();
}
}

```

Program 4 : application\SnakeGamePageController.java

```
package application;

import java.io.IOException;

import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class SnakeGamePageController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToPlaySnakeGame(ActionEvent event) throws
        IOException {
        root = FXMLLoader.load(getClass().getResource
            ("PlaySnakeGame.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().get
            Window();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void displaySnakeGameScore(ActionEvent event) throws
        IOException {
    }

    public void switchToSinglePlayerPage(ActionEvent event) throws
        IOException {
        root = FXMLLoader.load(getClass().getResource
            ("SinglePlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().get
            Window();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }
}
```

Program 5 : application\TicTacToePageController.java

```
package application;

import java.io.IOException;

import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class TicTacToePageController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToPlayTicTacToeGame(ActionEvent event) throws
        IOException {
        root = FXMLLoader.load(getClass().getResource
            ("PlayTicTacToeGame.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().get
            Window();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void displayTicTacToeGameScore(ActionEvent event) throws
        IOException {
    }

    public void switchToTwoPlayerPage(ActionEvent event) throws
        IOException {
        root = FXMLLoader.load(getClass().getResource
            ("TwoPlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().get
            Window();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }
}
```

Program 6 : application\PlayJavaQuizGameController.java

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import database.InsertDataDB;
import gamejavaquiz.JavaQuizFrame;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class PlayJavaQuizGameController {

    @FXML
    TextField JavaQuizGamePlayerNameTextField;

    @FXML
    Label JavaQuizGameWarningLabel;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void javaQuizPlayButton(ActionEvent event) throws
IOException {
        String username = JavaQuizGamePlayerNameTextField.getText();
        int score;

        if (isValidName(username)) {
            JavaQuizFrame javaquizframe = new JavaQuizFrame();
            score = javaquizframe.getScore();
            System.out.println(score);

            try {
                InsertDataDB.insertToJavaQuizTable(username, score);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        root = FXMLLoader.load(getClass().getResource
("JavaQuizGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene(). .
        getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }
}
```

```
    } else {
        JavaQuizGameWarningLabel.setText("Please Enter Valid
Name!");
    }
}

private static boolean isValidName(String name) {
    // Regular expression to match only alphabetic characters and
    spaces
    String regex = "^[a-zA-Z ]+$/";
    // Create a Pattern object
    Pattern pattern = Pattern.compile(regex);

    // Create a Matcher object
    Matcher matcher = pattern.matcher(name);

    // Check if the string matches the pattern
    return matcher.matches();
}
}
```

Program 7 : application\PlayPongGameController.java

```
package application;

import gamepong.*;
import java.io.IOException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import database.InsertDataDB;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class PlayPongGameController {

    @FXML
    Label PongGameWarningLabel;

    @FXML
    TextField PongGamePlayerRedTextField;

    @FXML
    TextField PongGamePlayerBlueTextField;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void pongGamePlayButton(ActionEvent event) throws
IOException {
        String playerRed = PongGamePlayerRedTextField.getText();
        String playerBlue = PongGamePlayerBlueTextField.getText();

        // at index 0 has playerRed Score, at index 1 has playerBlue
        // Score
        // at index 3 has winner (1 -- means red has won, 0 -- means
        blue has won)
        ArrayList<Integer> data = new ArrayList<Integer>();

        if (isValidName(playerRed) && isValidName(playerBlue)) {
            PongGameFrame ponggameframe = new PongGameFrame(playerRed,
                playerBlue);
            data = ponggameframe.getWinner();
            System.out.println(data);
        }
    }
}
```

```

        try {
            InsertDataDB.insertToPaddleTable(playerRed,
playerBlue, data);
        } catch (SQLException e) {
            e.printStackTrace();
        }

        root = FXMLLoader.load(getClass().getResource
("PongGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    } else {
        PongGameWarningLabel.setText("Please Enter Valid Name!");
    }
}

private static boolean isValidName(String name) {
    // Regular expression to match only alphabetic characters and
    // spaces
    String regex = "^[a-zA-Z ]+$/";
    // Create a Pattern object
    Pattern pattern = Pattern.compile(regex);

    // Create a Matcher object
    Matcher matcher = pattern.matcher(name);

    // Check if the string matches the pattern
    return matcher.matches();
}
}

```

Program 8 : application\PlaySnakeGameController.java

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import database.InsertDataDB;
import gamesnake.SnakeGameFrame;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class PlaySnakeGameController {

    @FXML
    TextField SnakeGamePlayerNameTextField;

    @FXML
    Label SnakeGameWarningLabel;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void snakePlayButton(ActionEvent event) throws IOException
    {
        String username = SnakeGamePlayerNameTextField.getText();
        int score;

        if (isValidName(username)) {
            SnakeGameFrame snakegameframe = new SnakeGameFrame();
            score = snakegameframe.getScore();
            System.out.println(username);
            System.out.println(score);

            if (score != -1) {
                try {
                    InsertDataDB.insertToSnakeTable(username, score);
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }

        root = FXMLLoader.load(getClass().getResource
("SnakeGamePage.fxml"));
    }
}
```

```

        stage = (Stage) ((Node) event.getSource()).getScene() .
            getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    } else {
        SnakeGameWarningLabel.setText("Please Enter Valid Name!");
    }
}

private static boolean isValidName(String name) {
    // matches only alphabetic characters and spaces
    String regex = "^[a-zA-Z ]+$/";

    // Create a Pattern object
    Pattern pattern = Pattern.compile(regex);

    // Create a Matcher object
    Matcher matcher = pattern.matcher(name);

    // Check if the string matches the pattern
    return matcher.matches();
}
}

```

Program 9 : application\PlayTicToeGameController.java

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import database.InsertDataDB;
import gametictactoe.TicTacToeFrame;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class PlayTicToeGameController {

    @FXML
    Label TicTacToeGameWarningLabel;

    @FXML
    TextField TicTacToeGamePlayer1TextField;

    @FXML
    TextField TicTacToeGamePlayer2TextField;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void tictactoePlayButton(ActionEvent event) throws
IOException {
        String playerX = TicTacToeGamePlayer1TextField.getText();
        String player0 = TicTacToeGamePlayer2TextField.getText();
        int winner; // 1 means playerX won, 0 means player0 won

        if (isValidName(playerX) && isValidName(player0)) {
            TicTacToeFrame tictactoeframe = new
TicTacToeFrame(playerX, player0);
            winner = tictactoeframe.getWinner();
            System.out.println(winner);

            try {
                InsertDataDB.insertToTicTacTable(playerX, player0,
winner);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

root = FXMLLoader.load(getClass().getResource
        ("TicTacToePage.fxml"));
stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
scene = new Scene(root);
stage.setScene(scene);
stage.setResizable(false);
stage.show();
} else {
    TicTacToeGameWarningLabel.setText("Please Enter Valid
        Name!");
}
}

private static boolean isValidName(String name) {
    // Regular expression to match only alphabetic characters and
    // spaces
    String regex = "^[a-zA-Z ]+$/";
    // Create a Pattern object
    Pattern pattern = Pattern.compile(regex);

    // Create a Matcher object
    Matcher matcher = pattern.matcher(name);

    // Check if the string matches the pattern
    return matcher.matches();
}
}

```

Program 10 : application\PongGamePageController.java

```
package application;

import java.io.IOException;

import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class PongGamePageController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToPlayPongGame(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
                ("PlayPongGame.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }

    public void displayPongGameScore(ActionEvent event) throws IOException {
    }

    public void switchToTwoPlayerPage(ActionEvent event) throws IOException {
        root = FXMLLoader.load(getClass().getResource
                ("TwoPlayerGamePage.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.setResizable(false);
        stage.show();
    }
}
```

Program 11 : application\GameGalaxyMainPage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.SceneController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <ImageView fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@ MainPageImage.jpg" />
                    </image>
                </ImageView>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;" text="GAME
GALAXY"
                      textFill="#fafafa">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button fx:id="button_two_player_game" layoutX="38.0"
                      layoutY="271.0" mnemonicParsing="false"
                      onAction="#switchToTwoPlayerPage" prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Two Player Game">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                </cursor>
```

```

        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="button_single_player_game" layoutX="38.0"
    layoutY="200.0" mnemonicParsing="false"
    onAction="#switchToSinglePlayerPage"
prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Single Player Game">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="exit_button_gamegalaxymain_page"
layoutX="38.0"
    layoutY="334.0" mnemonicParsing="false"
onAction="#exitGame"
    prefHeight="35.0" prefWidth="210.0"
    style="-fx-background-color: #FED700;" text="Exit">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>

```

Program 12 : application\JavaQuizGamePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.JavaQuizGamePageController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <Imageview fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@MainPageImage.jpg" />
                    </image>
                </Imageview>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;" text="Java
                            Quiz Game"
                      textAlignment="CENTER" textFill="#fafafa"
                      wrapText="true">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button fx:id="button_java_quiz_scores" layoutX="38.0"
                      layoutY="271.0" mnemonicParsing="false"
                      onAction="#displayJavaQuizGameScore"
                      prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Scores">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                </Button>
            </children>
        </AnchorPane>
    </children>
</HBox>
```

```
    </cursor>
</Button>
<Button fx:id="button_java_quiz_play_game" layoutX="38.0"
    layoutY="200.0" mnemonicParsing="false"
    onAction="#switchToPlayJavaQuizGame"
prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;"'
    text="Play Game">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
<Button fx:id="button_java_quiz_exit" layoutX="38.0"
    layoutY="335.0" mnemonicParsing="false"
    onAction="#switchToSinglePlayerPage"
prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;"'
    text="Quit Game">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>
```

Program 13 : application\PlayJavaQuizGame.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.PlayJavaQuizGameController">
<children>
    <AnchorPane prefHeight="400.0" prefWidth="320.0">
        <children>
            <ImageView fitHeight="400.0" fitWidth="342.0"
pickOnBounds="true" preserveRatio="true">
                <image>
                    <Image url="@MainPageImage.jpg" />
                </image>
            </ImageView>
        </children>
    </AnchorPane>
    <AnchorPane prefHeight="402.0" prefWidth="295.0"
style="-fx-border-color: #FED700; -fx-border-width: 5px;">
        <children>
            <Label alignment="CENTER" contentDisplay="CENTER"
layoutX="53.0"
                layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                style="-fx-background-color: #FED700;" text="Java
Quiz Game"
                textAlignment="CENTER" textFill="#fafafa"
                wrapText="true">
                <font>
                    <Font name="Arial Rounded MT Bold" size="24.0" />
                </font>
            </Label>
            <Button layoutX="39.0" layoutY="334.0"
mnemonicParsing="false"
                onAction="#javaQuizPlayButton" prefHeight="35.0"
                prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play">
                <font>
                    <Font name="Arial Rounded MT Bold" size="18.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
        </children>
    </AnchorPane>
</children>

```

```
    </cursor>
</Button>
<TextField fx:id="JavaQuizGamePlayerNameTextField"
    layoutX="44.0" layoutY="258.0">
    <font>
        <Font name="Arial Rounded MT Bold" size="16.0" />
    </font>
</TextField>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="193.0" prefHeight="35.0" prefWidth="200.0"
    style="-fx-background-color: #FED700;" text="Enter
textFill="#fafafa">
    <font>
        <Font name="Arial Rounded MT Bold" size="24.0" />
    </font>
</Label>
<Label fx:id="JavaQuizGameWarningLabel" layoutX="43.0"
    layoutY="303.0" prefHeight="18.0"
    prefWidth="200.0" />
</children>
</AnchorPane>
</children>
</Hbox>
```

Name :"

Program 14 : application\PlayPongGame.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.PlayPongGameController">
  <children>
    <AnchorPane prefHeight="400.0" prefWidth="320.0">
      <children>
        <ImageView fitHeight="400.0" fitWidth="342.0"
                   pickOnBounds="true" preserveRatio="true">
          <image>
            <Image url="@MainPageImage.jpg" />
          </image>
        </ImageView>
      </children>
    </AnchorPane>
    <AnchorPane prefHeight="402.0" prefWidth="295.0"
                style="-fx-border-color: #FED700; -fx-border-width:
                        5px;">
      <children>
        <Label alignment="CENTER" contentDisplay="CENTER"
              layoutX="53.0"
              layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
              style="-fx-background-color: #FED700;" text="Pong
Game"
              textFill="#fafafa">
          <font>
            <Font name="Arial Rounded MT Bold" size="24.0" />
          </font>
        </Label>
        <Button layoutX="38.0" layoutY="340.0"
               mnemonicParsing="false"
               onAction="#pongGamePlayButton" prefHeight="35.0"
               prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play">
          <font>
            <Font name="Arial Rounded MT Bold" size="18.0" />
          </font>
        </cursor>
```

```

        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<TextField fx:id="PongGamePlayerRedTextField"
layoutX="44.0"
    layoutY="189.0">
<font>
    <Font name="Arial Rounded MT Bold" size="16.0" />
</font>
</TextField>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="154.0" prefHeight="35.0"
prefWidth="200.0"
    style="-fx-background-color: #FF0000;" text="Player RED :"
        textFill="#fafafa">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
</Label>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="229.0" prefHeight="35.0"
prefWidth="200.0"
    style="-fx-background-color: #0000FF;" text="Player BLUE :"
        textFill="#fafafa">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
</Label>
<TextField fx:id="PongGamePlayerBlueTextField"
layoutX="44.0"
    layoutY="264.0">
<font>
    <Font name="Arial Rounded MT Bold" size="16.0" />
</font>
</TextField>
<Label fx:id="PongGameWarningLabel" layoutX="44.0"
    layoutY="307.0" prefHeight="18.0"
prefWidth="200.0" />
</children>
</AnchorPane>
</children>
</Hbox>

```

Program 15 : application\PlaySnakeGame.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"           xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.PlaySnakeGameController">
  <children>
    <AnchorPane prefHeight="400.0" prefWidth="320.0">
      <children>
        <Imageview fitHeight="400.0" fitWidth="342.0"
                   pickOnBounds="true" preserveRatio="true">
          <image>
            <Image url="@MainPageImage.jpg" />
          </image>
        </Imageview>
      </children>
    </AnchorPane>
    <AnchorPane prefHeight="402.0" prefWidth="295.0"
                style="-fx-border-color: #FED700; -fx-border-width:
                        5px;">
      <children>
        <Label alignment="CENTER" contentDisplay="CENTER"
              layoutX="53.0"
              layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
              style="-fx-background-color: #FED700;" text="Snake
Game"
              textFill="#fafafa">
          <font>
            <Font name="Arial Rounded MT Bold" size="24.0" />
          </font>
        </Label>
        <Button layoutX="39.0" layoutY="334.0"
               mnemonicParsing="false"
               onAction="#snakePlayButton" prefHeight="35.0"
               prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play">
          <font>
            <Font name="Arial Rounded MT Bold" size="18.0" />
          </font>
        </Button>
      </children>
    </AnchorPane>
  </children>
</HBox>
```

```

<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
<TextField fx:id="SnakeGamePlayerNameTextField"
layoutX="44.0"
    layoutY="258.0">
<font>
    <Font name="Arial Rounded MT Bold" size="16.0" />
</font>
</TextField>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="193.0" prefHeight="35.0"
prefWidth="200.0"
    style="-fx-background-color: #FED700;" text="Enter
Name :"
    textAlignment="CENTER" textFill="#fafafa"
wrapText="true">
<font>
    <Font name="Arial Rounded MT Bold" size="24.0" />
</font>
</Label>
<Label fx:id="SnakeGameWarningLabel" layoutX="44.0"
    layoutY="301.0" prefHeight="18.0"
prefWidth="200.0"
    wrapText="true" />
</children>
</AnchorPane>
</children>
</Hbox>

```

Program 16 : application\PlayTicToeGame.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.PlayTicToeGameController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <Imageview fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@MainPageImage.jpg" />
                    </image>
                </Imageview>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;"
                      text="TicTacToe Game"
                      textAlignment="CENTER" textFill="#fafafa"
                      wrapText="true">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button layoutX="39.0" layoutY="334.0"
                      mnemonicParsing="false"
                      onAction="#tictactoePlayButton" prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;"'
                      text="Play">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                </Button>
            </children>
        </AnchorPane>
    </children>
</HBox>
```

```

<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
<TextField fx:id="TicTacToeGamePlayer1TextField"
layoutX="44.0"
    layoutY="190.0">
<font>
    <Font name="Arial Rounded MT Bold" size="16.0" />
</font>
</TextField>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="155.0" prefHeight="35.0" prefWidth="200.0"
style="-fx-background-color: #FED700;" text="Player X
textFill="#fafafa">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
</Label>
<Label alignment="CENTER" contentDisplay="CENTER"
layoutX="44.0"
    layoutY="228.0" prefHeight="35.0" prefWidth="200.0"
style="-fx-background-color: #FED700;" text="Player 0
textFill="#fafafa">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
</Label>
<TextField fx:id="TicTacToeGamePlayer2TextField"
layoutX="44.0"
    layoutY="263.0">
<font>
    <Font name="Arial Rounded MT Bold" size="16.0" />
</font>
</TextField>
<Label fx:id="TicTacToeGameWarningLabel" layoutX="45.0"
    layoutY="304.0" prefHeight="18.0"
prefWidth="200.0" />
</children>
</AnchorPane>
</children>
</Hbox>

```

Program 17 : application\PongGamePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.PongGamePageController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <ImageView fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@MainPageImage.jpg" />
                    </image>
                </ImageView>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;" text="Pong
Game"
                      textFill="#fafafa">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button fx:id="button_pong_game_scores" layoutX="38.0"
                      layoutY="271.0" mnemonicParsing="false"
                      onAction="#displayPongGameScore" prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Scores">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                </Button>
            </children>
        </AnchorPane>
    </children>
</HBox>
```

```
</Button>
<Button fx:id="button_pong_game_play_game" layoutX="38.0"
    layoutY="200.0" mnemonicParsing="false"
    onAction="#switchToPlayPongGame" prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play Game">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="button_pong_game_exit" layoutX="38.0"
    layoutY="335.0" mnemonicParsing="false"
    onAction="#switchToTwoPlayerPage" prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Quit Game">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>
```

Program 18 : application\SinglePlayerGamePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.SceneController">
<children>
    <AnchorPane prefHeight="400.0" prefWidth="320.0">
        <children>
            <Imageview fitHeight="400.0" fitWidth="342.0"
pickOnBounds="true" preserveRatio="true">
                <image>
                    <Image url="@MainPageImage.jpg" />
                </image>
            </Imageview>
        </children>
    </AnchorPane>
    <AnchorPane prefHeight="402.0" prefWidth="295.0"
style="-fx-border-color: #FED700; -fx-border-width: 5px;">
        <children>
            <Label alignment="CENTER" contentDisplay="CENTER"
layoutX="53.0"
                layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
style="-fx-background-color: #FED700;"
text="Single Player Game" textAlignment="CENTER"
textFill="#fafafa" wrapText="true">
                <font>
                    <Font name="Arial Rounded MT Bold" size="24.0" />
                </font>
            </Label>
            <Button fx:id="button_java_quiz_game" layoutX="38.0"
layoutY="271.0" mnemonicParsing="false"
onAction="#switchToJavaQuizGamePage"
prefHeight="35.0"
prefWidth="210.0" style="-fx-background-color:
#FED700;"'
text="Java Quiz Game">
                <font>
                    <Font name="Arial Rounded MT Bold" size="18.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
        </children>
    </AnchorPane>
</children>

```

```

        </cursor>
    </Button>
    <Button fx:id="button_snake_game" layoutX="38.0"
           mnemonicParsing="false"
           onAction="#switchToSnakeGamePage"
           prefHeight="35.0" prefWidth="210.0"
           style="-fx-background-color: #FED700;" text="Snake
           <font>
              <Font name="Arial Rounded MT Bold" size="18.0" />
           </font>
           <cursor>
              <Cursor fx:constant="HAND" />
           </cursor>
        </Button>
        <Button fx:id="go_back_button_singleplayerpage"
               layoutX="38.0"
               layoutY="336.0" mnemonicParsing="false"
               onAction="#switchToGameGalaxy MainPage"
               prefHeight="35.0"
               prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Go Back">
           <font>
              <Font name="Arial Rounded MT Bold" size="18.0" />
           </font>
           <cursor>
              <Cursor fx:constant="HAND" />
           </cursor>
        </Button>
    </children>
</AnchorPane>
</children>
</Hbox>

```

Program 19 : application\SnakeGamePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.SnakeGamePageController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <ImageView fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@MainPageImage.jpg" />
                    </image>
                </ImageView>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;" text="Snake
Game"
                      textFill="#fafafa">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button fx:id="button_snake_game_scores" layoutX="38.0"
                      layoutY="271.0" mnemonicParsing="false"
                      onAction="#displaySnakeGameScore" prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Scores">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                </Button>
            </children>
        </AnchorPane>
    </children>
</HBox>
```

```
</Button>
<Button fx:id="button_snake_game_play_game"
layoutX="38.0"
    layoutY="200.0" mnemonicParsing="false"
    onAction="#switchToPlaySnakeGame" prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play Game">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="button_snake_game_exit" layoutX="38.0"
    layoutY="335.0" mnemonicParsing="false"
    onAction="#switchToSinglePlayerPage"
prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Quit Game">
    <font>
        <Font name="Arial Rounded MT Bold" size="18.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>
```

Program 20 : application\TicTacToePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
       minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
       xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
       fx:controller="application.TicTacToePageController">
    <children>
        <AnchorPane prefHeight="400.0" prefWidth="320.0">
            <children>
                <Imageview fitHeight="400.0" fitWidth="342.0"
                           pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@MainPageImage.jpg" />
                    </image>
                </Imageview>
            </children>
        </AnchorPane>
        <AnchorPane prefHeight="402.0" prefWidth="295.0"
                   style="-fx-border-color: #FED700; -fx-border-width: 5px;">
            <children>
                <Label alignment="CENTER" contentDisplay="CENTER"
                      layoutX="53.0"
                      layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
                      style="-fx-background-color: #FED700;" text="Tic Tac Toe Game" textAlignment="CENTER"
                      textFill="#fafafa" wrapText="true">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="24.0" />
                    </font>
                </Label>
                <Button fx:id="button_tictactoe_game_scores"
                      layoutY="271.0" mnemonicParsing="false"
                      onAction="#displayTicToeGameScore"
                      prefHeight="35.0"
                      prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Scores">
                    <font>
                        <Font name="Arial Rounded MT Bold" size="18.0" />
                    </font>
                </cursor>
            </children>
        </AnchorPane>
    </children>
</HBox>
```

```

        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="button_tictactoe_game_play_game"
layoutX="38.0"
    layoutY="200.0" mnemonicParsing="false"
    onAction="#switchToPlayTicToeGame"
prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Play Game">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
<Button fx:id="button_tictactoe_game_exit" layoutX="38.0"
    layoutY="335.0" mnemonicParsing="false"
    onAction="#switchToTwoPlayerPage" prefHeight="35.0"
    prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Quit Game">
<font>
    <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>

```

Program 21 : application\TwoPlayerGamePage.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.SceneController">
<children>
    <AnchorPane prefHeight="400.0" prefWidth="320.0">
        <children>
            <ImageView fitHeight="400.0" fitWidth="342.0"
pickOnBounds="true" preserveRatio="true">
                <image>
                    <Image url="@MainPageImage.jpg" />
                </image>
            </ImageView>
        </children>
    </AnchorPane>
    <AnchorPane prefHeight="402.0" prefWidth="295.0"
style="-fx-border-color: #FED700; -fx-border-width: 5px;">
        <children>
            <Label alignment="CENTER" contentDisplay="CENTER"
layoutX="53.0"
                layoutY="51.0" prefHeight="81.0" prefWidth="187.0"
style="-fx-background-color: #FED700;"
text="Two Player Game" textAlignment="CENTER"
textFill="#fafafa" wrapText="true">
                <font>
                    <Font name="Arial Rounded MT Bold" size="24.0" />
                </font>
            </Label>
            <Button fx:id="button_tictactoe_game" layoutX="38.0"
layoutY="271.0" mnemonicParsing="false"
onAction="#switchToTicToeGamePage"
prefHeight="35.0"
prefWidth="210.0" style="-fx-background-color:
#FED700;"
text="Tic Tac Toe Game">
                <font>
                    <Font name="Arial Rounded MT Bold" size="18.0" />
                </font>
            </Button>
        </children>
    </AnchorPane>
</children>

```

```

</font>
<cursor>
  <Cursor fx:constant="HAND" />
</cursor>
</Button>
<Button fx:id="button_pong_game" layoutX="38.0"
layoutY="200.0"
  mnemonicParsing="false"
onAction="#switchToPongGamePage"
  prefHeight="35.0" prefWidth="210.0"
  style="-fx-background-color: #FED700;" text="Pong
<font>
  <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<opaqueInsets>
  <Insets />
</opaqueInsets>
<cursor>
  <Cursor fx:constant="HAND" />
</cursor>
</Button>
<Button fx:id="go_back_button_Twoplayerpage"
layoutX="38.0"
  layoutY="338.0" mnemonicParsing="false"
  onAction="#switchToGameGalaxy MainPage"
  prefHeight="35.0"
  prefWidth="210.0" style="-fx-background-color:
#FED700;" text="Go Back">
<font>
  <Font name="Arial Rounded MT Bold" size="18.0" />
</font>
<cursor>
  <Cursor fx:constant="HAND" />
</cursor>
</Button>
</children>
</AnchorPane>
</children>
</Hbox>

```

PACKAGE 2 : gamesnake

Program 1 : gamesnake\SnakeGameFrame.java

```
package gamesnake;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import javax.swing.border.Border;

public class SnakeGameFrame extends JFrame {
    private SnakeGamePanel snakegamepanel;

    public SnakeGameFrame() {
        snakegamepanel = createPanel();

        SwingUtilities.invokeLater(() -> {
            Border customBorder = new CustomBorder(5); // adds border to
                // the game
            this.getRootPane().setBorder(customBorder);
            // this.setLayout(new BorderLayout());
            this.add(snakegamepanel);
            this.setTitle("Snake Game");
            // this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            this.setUndecorated(true); // to remove windows default title
                bar
            this.setResizable(false);
            this.pack();
            this.setVisible(true);
            this.setLocationRelativeTo(null);
        });
    }

    snakegamepanel.playAgain.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playAgain();
        }
    });

    public int getScore() {

        while (!snakegamepanel.isButtonClicked()) {
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
        }
        return snakegamepanel.getScore();
    }

public void playAgain() {

    this.getContentPane().removeAll();
    snakegamepanel = createPanel();
    this.getContentPane().add(snakegamepanel);
    snakegamepanel.requestFocusInWindow();
    this.revalidate();
    this.repaint();

    snakegamepanel.playAgain.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playAgain();
        }
    });
}

private SnakeGamePanel createPanel() {
    return new SnakeGamePanel();
}
}
```

Program 2 : gamesnake\SnakeGamePanel.java

```
package gamesnake;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.FontMetrics;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.Random;

import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.Timer;

public class SnakeGamePanel extends JPanel implements ActionListener {

    static final int SCREEN_WIDTH = 600;
    static final int SCREEN_HEIGHT = 600;
    static final int UNIT_SIZE = 25;
    static final int GAME_UNIT = (SCREEN_HEIGHT * SCREEN_WIDTH) /  UNIT_SIZE;
    static final int DELAY = 65;
    final int x[] = new int[GAME_UNIT];
    final int y[] = new int[GAME_UNIT];
    int bodyParts = 6;
    int applesEaten = 0;
    int appleX;
    int appleY;
    private int score;
    public JButton playAgain = new JButton("PLAY AGAIN");
    private JButton button = new JButton("Save Score & EXIT");
    private JButton button2 = new JButton("EXIT GAME");
    private boolean buttonClicked = false;
    char direction = 'R';
    boolean running = false;
    Timer timer;
    Random random;

    SnakeGamePanel() {

        playAgain.setBounds(200, 315, 200, 75);
        playAgain.setFont(new Font("Arial", Font.BOLD, 16));
        playAgain.setBackground(new Color(254, 215, 0));
        playAgain.setForeground(Color.white);
        playAgain.setFocusable(false);

        random = new Random();
        button.setBounds(200, 400, 200, 75);
        button.setFont(new Font("Arial", Font.BOLD, 16));
```

```

button.setBackground(new Color(254, 215, 0));
button.setForeground(Color.white);
button.setFocusable(false);

button2.setBounds(200, 485, 200, 75);
button2.setFont(new Font("Arial", Font.BOLD, 16));
button2.setBackground(new Color(254, 215, 0));
button2.setForeground(Color.white);
button2.setFocusable(false);

this.setPreferredSize(new Dimension(SCREEN_WIDTH,
    SCREEN_HEIGHT));
this.setBackground(new Color(51, 204, 255));
this.setFocusable(true);
this.addKeyListener(new MyKeyAdapter());
start_game();
}

public void start_game() {
    New_Apple();
    running = true;
    timer = new Timer(DELAY, this);
    timer.start();
}

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    draw(g);
}

public void draw(Graphics g) {
    if (running) {
        g.setColor(new Color(255, 255, 0));
        g.fillOval(appleX, appleY, UNIT_SIZE, UNIT_SIZE);

        for (int i = 0; i < bodyParts; i++) {
            if (i == 0) {
                g.setColor(new Color(254, 215, 0));
                g.fillRect(x[i], y[i], UNIT_SIZE, UNIT_SIZE);
            } else {
                g.setColor(new Color(203, 172, 0));
                g.fillRect(x[i], y[i], UNIT_SIZE, UNIT_SIZE);
            }
        }

        g.setColor(new Color(254, 215, 0));
        g.setFont(new Font("SansSerif Bold", Font.ITALIC, 30));
        FontMetrics metrics = getFontMetrics(g.getFont());
        g.drawString("Score : " + applesEaten,
            (SCREEN_WIDTH - metrics.stringWidth("Score : " +
            applesEaten)) / 2, g.getFont().getSize());
    } else {
        gameover(g);
    }
}

```

```

        }

    }

public void New_Apple() {
    appleX = random.nextInt((int) (SCREEN_WIDTH / UNIT_SIZE)) *
        UNIT_SIZE;
    appleY = random.nextInt((int) (SCREEN_HEIGHT / UNIT_SIZE)) *
        UNIT_SIZE;
}

public void move() {
    for (int i = bodyParts; i > 0; i--) {
        x[i] = x[i - 1];
        y[i] = y[i - 1];
    }
    switch (direction) {
        case 'U':
            y[0] = y[0] - UNIT_SIZE;
            break;
        case 'D':
            y[0] = y[0] + UNIT_SIZE;
            break;
        case 'R':
            x[0] = x[0] + UNIT_SIZE;
            break;
        case 'L':
            x[0] = x[0] - UNIT_SIZE;
            break;
    }
}

public void checkApple() {
    if ((x[0] == appleX) && (y[0] == appleY)) {
        bodyParts++;
        applesEaten++;
        New_Apple();
    }
}

public void CheckCollisions() {
    for (int i = bodyParts; i > 0; i--) {
        if ((x[0] == x[i]) && (y[0] == y[i])) {
            running = false;
        }
    }

    if (x[0] < 0) {
        running = false;
    }

    if (x[0] > SCREEN_WIDTH) {
        running = false;
    }
}

```

```

if (y[0] < 0) {
    running = false;
}

if (y[0] > SCREEN_HEIGHT) {
    running = false;
}

if (!running) {
    timer.stop();
}
}

public void gameover(Graphics g) {
    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            score = applesEaten;
            buttonClicked = true;
            SwingUtilities.getWindowAncestor(button).dispose();
        }
    });
}

button2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        score = -1;
        buttonClicked = true;
        SwingUtilities.getWindowAncestor(button2).dispose();
    }
});
this.add(button);
this.add(button2);
this.add(playAgain);

g.setColor(new Color(254, 215, 0));
g.setFont(new Font("SansSerif Bold", Font.BOLD, 85));
FontMetrics metrics1 = getFontMetrics(g.getFont());
g.drawString("GAME OVER", (SCREEN_WIDTH -
metrics1.stringWidth("GAME OVER")) / 2, SCREEN_HEIGHT / 2);

g.setColor(new Color(254, 215, 0));
g.setFont(new Font("SansSerif Bold", Font.BOLD, 40));
FontMetrics metrics2 = getFontMetrics(g.getFont());
g.drawString("Score : " + applesEaten,
(SCCREEN_WIDTH - metrics2.stringWidth("Score : " +
applesEaten)) / 2,
g.getFont().getSize());
}

public int getScore() {
    return score;
}

```

```

public boolean isButtonClicked() {
    return buttonClicked;
}

@Override
public void actionPerformed(ActionEvent e) {
    if (running) {
        move();
        checkApple();
        CheckCollisions();
    }
    repaint();
}

public class MyKeyAdapter extends KeyAdapter {
    @Override
    public void keyPressed(KeyEvent e) {
        switch (e.getKeyCode()) {
            case KeyEvent.VK_LEFT:
                if (direction != 'R') {
                    direction = 'L';
                }
                break;
            case KeyEvent.VK_RIGHT:
                if (direction != 'L') {
                    direction = 'R';
                }
                break;
            case KeyEvent.VK_UP:
                if (direction != 'D') {
                    direction = 'U';
                }
                break;
            case KeyEvent.VK_DOWN:
                if (direction != 'U') {
                    direction = 'D';
                }
                break;
        }
    }
}
}

```

PACKAGE 3 : gamejavaquiz

Program 1 : gamejavaquiz\JavaQuizFrame.java

```
package gamejavaquiz;

import java.awt.BorderLayout;

import javax.swing.JFrame;
import javax.swing.SwingUtilities;
import javax.swing.border.Border;

public class JavaQuizFrame extends JFrame {

    private JavaQuizPanel javaquizpanel;

    public JavaQuizFrame() {
        javaquizpanel = new JavaQuizPanel();

        SwingUtilities.invokeLater(() -> {
            Border customBorder = new CustomBorder(5);
            this.getRootPane().setBorder(customBorder);
            this.add(javaquizpanel);
            this.setTitle("Java Quiz Game");
            this.setUndecorated(true);
            this.setResizable(false);
            this.pack();
            this.setVisible(true);
            this.setLocationRelativeTo(null);
        });
    }

    public int getScore() {
        while (!javaquizpanel.isButtonClicked()) {
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        return javaquizpanel.getScore();
    }
}
```

Program 2 : gamejavaquiz\JavaQuizPanel.java

```
package gamejavaquiz;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashSet;
import java.util.Random;
import java.util.Set;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.Timer;

public class JavaQuizPanel extends JPanel implements ActionListener {

    String[] questions = {
        "which company created java ?",
        "In which year java was created ?",
        "what was java originally called ?",
        "What is Java?",
        "Explain the main features of Java.",
        "What are the primitive data types in Java?",
        "What is the default value of the boolean variable in Java?",
        "Which keyword is used to declare a constant in Java?",
        "What is the purpose of the 'static' keyword in Java?",
        "Which class is the superclass for all classes in Java?",
        "What is the main purpose of the 'StringBuilder' class in Java?",
        "What is the difference between '==' and '.equals()' when comparing strings in Java?",

        "What is the output of the following code? int x = 5;
System.out.println(x++);",
        "Which Java keyword is used to implement multiple inheritance?",

        "What is the purpose of the 'super' keyword in Java?",

        "How can you handle exceptions in Java?",

        "What is the significance of the 'final' keyword in Java?",

        "What is the difference between 'ArrayList' and 'LinkedList' in Java?"};

    String[][] options = {
        {"A. Sun MicroSystems", "B. Oracle", "C. Microsoft", "D. Alphabet"}},
```

{"A. 1989", "B. 1995", "C. 1972", "D. 1980"},
 {"A. Apple", "B. Oak", "C. COBOL", "D. Kotlin"},
 {"A. Java is a programming language.", "B. Java is a type of coffee.", "C. Java
is an island in Indonesia.", "D. Java is a database management system."},
 {"A. Object-oriented programming", "B. Platform independence", "C. Memory management", "D. All of the above"},
 {"A. int", "B. float", "C. boolean", "D. All of the above"},
 {"A. true", "B. false", "C. 0", "D. 1"},
 {"A. const", "B. final", "C. static", "D. constant"},
 {"A. To create an instance of a class", "B. To indicate that a method or variable
belongs to the class rather than an instance", "C. To make a class non-instantiable",
"D. To indicate a constant variable"},
 {"A. Object", "B. Main", "C. JavaLang", "D. None of the above"},
 {"A. To create mutable sequences of characters", "B. To represent a fixed-size,
immutable sequence of characters", "C. To format strings", "D. To parse XML documents"},
 {"A. '==' compares object references, while '.equals()' compares the content
of the objects", "B. '==' compares the content of the objects, while '.equals()' compares
object references", "C. Both '==' and '.equals()' are used interchangeably", "D.
'==' and '.equals()' are only applicable to primitive data types"},
 {"A. 6", "B. 5", "C. 4", "D. 6 (with a space)"},
 {"A. extends", "B. implements", "C. inheritance", "D. super"},
 {"A. It refers to the superclass of the current class", "B. It is used to call the
constructor of the superclass", "C. It is used to access the superclass's fields and
methods", "D. It is used to create an instance of the
superclass"},
 {"A. By using try-catch blocks", "B. By declaring exceptions in the method
signature", "C. By using the 'finally' block", "D. All of the above"},
 {"A. It indicates that a class cannot be extended", "B. It indicates that a method
cannot be overridden", "C. It indicates that a variable cannot be modified after
initialization", "D. All of the above"},
 {"A. 'ArrayList' is based on an array, while 'LinkedList' is based on a linked list",
"B. 'ArrayList' allows faster random access, while 'LinkedList' is efficient for
insertions and deletions", "C. 'ArrayList' is thread-safe, while
'LinkedList' is not", "D. There is no significant difference between 'ArrayList' and
'LinkedList'" } ;

```

char[] answers = {
  'A','B','B','A','D','D','B','B',
  'B','A','A','A','B','B','B','D','D','B'
};
  
```

```

char guess;
char answer;
int index;
int correct_guesses = 0;
int total_questions = questions.length;
int result;
int seconds = 30;
  
```

```

JTextField textfield = new JTextField();
JTextArea textarea = new JTextArea(); //to hold the current
questions
JButton buttonA = new JButton();
  
```

```

 JButton buttonB = new JButton();
 JButton buttonC = new JButton();
 JButton buttonD = new JButton();

 JButton saveAndExitButton = new JButton("SAVE AND EXIT");
 JButton exitButton = new JButton("Exit");
 JButton quitquizButton = new JButton("Quit Quiz");

 JLabel answer_labelA = new JLabel();
 JLabel answer_labelB = new JLabel();
 JLabel answer_labelC = new JLabel();
 JLabel answer_labelD = new JLabel();
 JLabel time_label = new JLabel();
 JLabel seconds_left = new JLabel();
 JTextField number_right = new JTextField();
 JTextField percentage = new JTextField();

 private boolean button_clicked = false;

 Set<Integer> usedNumbers = new HashSet<>();
 Random random = new Random();
 static int qus_no = 0;

 Timer timer = new Timer(1000, new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        seconds--;
        seconds_left.setText(String.valueOf(seconds));
        if (seconds <= 0) {
            displayAnswer();
        }
    }

 });

 public JavaQuizPanel() {

    textfield.setBounds(0, 0, 1500, 50);
    textfield.setBackground(new Color(25, 25, 25));
    textfield.setForeground(new Color(25, 255, 0));
    textfield.setFont(new Font("Ink Free", Font.BOLD, 30));
    textfield.setBorder(BorderFactory.createBevelBorder(1));
    textfield.setHorizontalAlignment(JTextField.CENTER);
    textfield.setEditable(false);

    textarea.setBounds(0, 50, 1500, 50);
    textarea.setLineWrap(true);
    textarea.setWrapStyleWord(true);
    textarea.setBackground(new Color(25, 25, 25));
    textarea.setForeground(new Color(25, 255, 0));
    textarea.setFont(new Font("MV Boli", Font.BOLD, 30));
    textarea.setBorder(BorderFactory.createBevelBorder(1));
}

```

```

textarea.setEditable(false);

buttonA.setBounds(0, 100, 200, 100);
buttonA.setFont(new Font("MV Boli", Font.BOLD, 35));
buttonA.setFocusable(false);
buttonA.addActionListener(this);
buttonA.setText("A");

buttonB.setBounds(0, 200, 200, 100);
buttonB.setFont(new Font("MV Boli", Font.BOLD, 35));
buttonB.setFocusable(false);
buttonB.addActionListener(this);
buttonB.setText("B");

buttonC.setBounds(0, 300, 200, 100);
buttonC.setFont(new Font("MV Boli", Font.BOLD, 35));
buttonC.setFocusable(false);
buttonC.addActionListener(this);
buttonC.setText("C");

buttonD.setBounds(0, 400, 200, 100);
buttonD.setFont(new Font("MV Boli", Font.BOLD, 35));
buttonD.setFocusable(false);
buttonD.addActionListener(this);
buttonD.setText("D");

answer_labelA.setBounds(225, 100, 1500, 100);
answer_labelA.setBackground(new Color(50, 50, 50));
answer_labelA.setForeground(new Color(25, 255, 0));
answer_labelA.setFont(new Font("MV Boli", Font.PLAIN, 30));

answer_labelB.setBounds(225, 200, 1500, 100);
answer_labelB.setBackground(new Color(50, 50, 50));
answer_labelB.setForeground(new Color(25, 255, 0));
answer_labelB.setFont(new Font("MV Boli", Font.PLAIN, 30));

answer_labelC.setBounds(225, 300, 1500, 100);
answer_labelC.setBackground(new Color(50, 50, 50));
answer_labelC.setForeground(new Color(25, 255, 0));
answer_labelC.setFont(new Font("MV Boli", Font.PLAIN, 30));

answer_labelD.setBounds(225, 400, 1500, 100);
answer_labelD.setBackground(new Color(50, 50, 50));
answer_labelD.setForeground(new Color(25, 255, 0));
answer_labelD.setFont(new Font("MV Boli", Font.PLAIN, 30));

seconds_left.setBounds(1400, 600, 100, 100);
seconds_left.setBackground(new Color(25, 25, 25));
seconds_left.setForeground(new Color(255, 0, 0));
seconds_left.setFont(new Font("Ink Free", Font.BOLD, 60));
seconds_left.setBorderStyle(BorderFactory.createBevelBorder(1));
seconds_left.setOpaque(true);
seconds_left.setHorizontalAlignment(JTextField.CENTER);

```

```

seconds_left.setText(String.valueOf(seconds));

time_label.setBounds(1400, 575, 100, 20);
time_label.setBackground(new Color(50, 50, 50));
time_label.setForeground(new Color(255, 0, 0));
time_label.setFont(new Font("MV Boli", Font.PLAIN, 20));
time_label.setHorizontalAlignment(JTextField.CENTER);
time_label.setText("timer >:D");

number_right.setBounds(695, 125, 110, 75);
number_right.setBackground(new Color(25, 25, 25));
number_right.setForeground(new Color(25, 255, 0));
number_right.setFont(new Font("Ink Free", Font.BOLD, 30));
number_right.setHorizontalAlignment(JTextField.CENTER);
number_right.setEditable(false);

percentage.setBounds(695, 200, 110, 75);
percentage.setBackground(new Color(25, 25, 25));
percentage.setForeground(new Color(25, 255, 0));
percentage.setFont(new Font("Ink Free", Font.BOLD, 30));
percentage.setBorder(BorderFactory.createBevelBorder(1));
percentage.setHorizontalAlignment(JTextField.CENTER);
percentage.setEditable(false);

saveAndExitButton.setBounds(625, 325, 250, 100);
saveAndExitButton.setFont(new Font("Arial", Font.BOLD, 25));
saveAndExitButton.setBackground(new Color(25, 255, 0));
saveAndExitButton.setForeground(Color.white);
saveAndExitButton.setFocusable(false);

exitButton.setBounds(625, 435, 250, 100);
exitButton.setFont(new Font("Arial", Font.BOLD, 25));
exitButton.setBackground(new Color(25, 255, 0));
exitButton.setForeground(Color.white);
exitButton.setFocusable(false);

quitquizButton.setBounds(625, 600, 250, 100);
quitquizButton.setFont(new Font("Arial", Font.BOLD, 25));
quitquizButton.setBackground(new Color(25, 255, 0));
quitquizButton.setForeground(Color.white);
quitquizButton.setFocusable(false);

quitquizButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        result = -1;
        button_clicked = true;

        SwingUtilities.getWindowAncestor(quitquizButton).dispose();
    }
});

```

```

this.add(quitquizButton);
this.add(time_label);
this.add(seconds_left);
this.add(answer_labelA);
this.add(answer_labelB);
this.add(answer_labelC);
this.add(answer_labelD);
this.add(buttonA);
this.add(buttonB);
this.add(buttonC);
this.add(buttonD);
this.add(textarea);
this.add(textfield);

this.setPreferredSize(new Dimension(1500, 700)); //note border applies to the frame
when we use setPreferredsize
this.setBackground(new Color(50, 50, 50));
this.setFocusable(true);

this.setLayout(null);

index = getIndex();
nextQuestion();
}

public int getIndex() {

int randomIndex;
while (usedNumbers.size() < total_questions) {
    randomIndex = random.nextInt((total_questions - 1) - 0 + 1) +
    if (!usedNumbers.contains(randomIndex)) {
        usedNumbers.add(randomIndex);
        return randomIndex;
    }
}
return total_questions;
}

public void nextQuestion() { //method for displaying the next
question

if (index >= total_questions) {
    results();
} else {
   textfield.setText("Question " + (qus_no += 1));
    textarea.setText(questions[index]);
    answer_labelA.setText(options[index][0]);
    answer_labelB.setText(options[index][1]);
    answer_labelC.setText(options[index][2]);
    answer_labelD.setText(options[index][3]);
    timer.start();
}
}
}

```

```

@Override
public void actionPerformed(ActionEvent e) { //method for action
    performed
    buttonA.setEnabled(false);
    buttonB.setEnabled(false);
    buttonC.setEnabled(false);
    buttonD.setEnabled(false);

    if (e.getSource() == buttonA) {
        answer = 'A';
        if (answer == answers[index]) {
            correct_guesses++;
        }
    }

    if (e.getSource() == buttonB) {
        answer = 'B';
        if (answer == answers[index]) {
            correct_guesses++;
        }
    }

    if (e.getSource() == buttonC) {
        answer = 'C';
        if (answer == answers[index]) {
            correct_guesses++;
        }
    }

    if (e.getSource() == buttonD) {
        answer = 'D';
        if (answer == answers[index]) {
            correct_guesses++;
        }
    }

    displayAnswer();
}

public void displayAnswer() //for displaying the answer

    timer.stop();
    buttonA.setEnabled(false);
    buttonB.setEnabled(false);
    buttonC.setEnabled(false);
    buttonD.setEnabled(false);

    if (answers[index] != 'A') {
        answer_labelA.setForeground(new Color(255, 0, 0));
    }
    if (answers[index] != 'B') {
        answer_labelB.setForeground(new Color(255, 0, 0));
    }

```

```

}

if (answers[index] != 'C') {
    answer_labelC.setForeground(new Color(255, 0, 0));
}
if (answers[index] != 'D') {
    answer_labelD.setForeground(new Color(255, 0, 0));
}

Timer pause = new Timer(2000, new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        answer_labelA.setForeground(new Color(25, 255, 0));
        answer_labelB.setForeground(new Color(25, 255, 0));
        answer_labelC.setForeground(new Color(25, 255, 0));
        answer_labelD.setForeground(new Color(25, 255, 0));

        answer = ' ';
        seconds = 30;
        seconds_left.setText(String.valueOf(seconds));
        buttonA.setEnabled(true);
        buttonB.setEnabled(true);
        buttonC.setEnabled(true);
        buttonD.setEnabled(true);
        index = getIndex();
        nextQuestion();

    }
}); //to add a pause after displaying the output

pause.setRepeats(false);
pause.start();
}

public void results() { //for showing the result

    this.remove(quitquizButton);

    this.revalidate();
    this.repaint();
    result = (int) (((double) correct_guesses / total_questions) * 100);

    saveAndExitButton.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            button_clicked = true;

            SwingUtilities.getWindowAncestor(saveAndExitButton).dispose();
        }
    });
}

```

```

exitButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        result = -1;
        button_clicked = true;
        SwingUtilities.getWindowAncestor(exitButton).dispose();
    }

});

buttonA.setEnabled(false);
buttonB.setEnabled(false);
buttonC.setEnabled(false);
buttonD.setEnabled(false);

textfield.setText("RESULTS!");
textarea.setText("");
answer_labelA.setText("");
answer_labelB.setText("");
answer_labelC.setText("");
answer_labelD.setText("");

number_right.setText("(" + correct_guesses + "/" + `total_questions +
")");
percentage.setText(result + "%");

this.add(percentage);
this.add(number_right);
this.add(saveAndExitButton);
this.add(exitButton);
}

public int getScore() {
    return result;
}

public boolean isButtonClicked() {
    return button_clicked;
}
}
}

```

PACKAGE 4 : gametictactoe

Program 1 : gametictactoe\TicTacToeFrame.java

```
package gametictactoe;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.border.Border;

public class TicTacToeFrame extends JFrame {

    TicTacToePanel tictactoepanel;
    private String playerX, player0;

    public TicTacToeFrame(String playerX, String player0) {

        this.playerX = playerX;
        this.player0 = player0;
        tictactoepanel = createPanel();

        SwingUtilities.invokeLater(() -> {
            Border customBorder = new CustomBorder(5);
            this.getRootPane().setBorder(customBorder);
            this.add(tictactoepanel);
            this.setTitle("Tic Tac Toe");
            this.setUndecorated(true);
            this.setResizable(false);
            this.pack();
            this.setVisible(true);
            this.setLocationRelativeTo(null);
        });
    }

    tictactoepanel.playAgain.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            playAgain();
        }
    });

}

private void playAgain() {
    this.getContentPane().removeAll();
    tictactoepanel = createPanel();
    this.getContentPane().add(tictactoepanel);
```

```
this.revalidate();
this.repaint();

tictactoepanel.playAgain.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        playAgain();
    }

});

}

private TicTacToePanel createPanel() {
    return new TicTacToePanel(playerX, playerO);
}

public int getWinner() {

    while (!tictactoepanel.isButtonClicked()) {
        try {
            Thread.sleep(100);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return tictactoepanel.winner;
}

}
```

Program 2 : gametictactoe\TicToePanel.java

```
package gametictactoe;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class TicToePanel extends JPanel implements ActionListener {

    Random random = new Random();
    JPanel button_panel = new JPanel();
    JLabel textfield = new JLabel();
    JButton[] buttons = new JButton[9];
    JButton playAgain = new JButton("PLAY AGAIN");
    JButton saveAndExit = new JButton("SAVE AND EXIT");
    JButton exit = new JButton("EXIT");
    boolean playerX_turn;
    private String playerX, player0;
    private boolean buttonClicked = false;
    int winner;

    TicToePanel(String playerX, String player0) {

        this.playerX = playerX;
        this.player0 = player0;

        playAgain.setBounds(0, 40, 266, 80);
        playAgain.setFont(new Font("Arial", Font.BOLD, 20));
        playAgain.setBackground(new Color(25, 255, 0));
        playAgain.setForeground(Color.white);
        playAgain.setFocusable(false);

        saveAndExit.setBounds(267, 40, 266, 80);
        saveAndExit.setFont(new Font("Arial", Font.BOLD, 20));
        saveAndExit.setBackground(new Color(25, 255, 0));
        saveAndExit.setForeground(Color.white);
        saveAndExit.setFocusable(false);

        exit.setBounds(534, 40, 266, 80);
        exit.setFont(new Font("Arial", Font.BOLD, 20));
        exit.setBackground(new Color(25, 255, 0));
        exit.setForeground(Color.white);
        exit.setFocusable(false);

        button_panel.setLayout(new GridLayout(3, 3));
        button_panel.add(textfield);
        for (int i = 0; i < 9; i++) {
            JButton button = buttons[i];
            button.addActionListener(this);
            button_panel.add(button);
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        JButton clickedButton = (JButton) e.getSource();
        if (buttonClicked) return;
        buttonClicked = true;
        if (playerX_turn) {
            clickedButton.setText(playerX);
            if (winner == 0) {
                if (checkWin(playerX)) {
                    clickedButton.setText("X");
                    winner = 1;
                } else if (checkWin(player0)) {
                    clickedButton.setText("O");
                    winner = 2;
                } else if (checkDraw()) {
                    clickedButton.setText("DRAW");
                    winner = 3;
                }
            }
        } else {
            clickedButton.setText(player0);
            if (winner == 0) {
                if (checkWin(player0)) {
                    clickedButton.setText("O");
                    winner = 2;
                } else if (checkWin(playerX)) {
                    clickedButton.setText("X");
                    winner = 1;
                } else if (checkDraw()) {
                    clickedButton.setText("DRAW");
                    winner = 3;
                }
            }
        }
        playerX_turn = !playerX_turn;
        buttonClicked = false;
    }

    private boolean checkWin(String player) {
        // Check rows, columns, and diagonals for three-in-a-row
        return false; // Implementation omitted
    }

    private boolean checkDraw() {
        // Check if all buttons are filled and no winner
        return false; // Implementation omitted
    }
}
```

```

textfield.setBounds(0, 0, 800, 120);
textfield.setBackground(new Color(25, 25, 25));
textfield.setForeground(new Color(25, 255, 0));
textfield.setFont(new Font("Ink Free", Font.BOLD, 65));
textfield.setHorizontalAlignment(JLabel.CENTER);
textfield.setVerticalAlignment(JLabel.CENTER);
textfield.setOpaque(true);

button_panel.setBounds(0, 120, 800, 680);
button_panel.setLayout(new GridLayout(3, 3));
button_panel.setBackground(new Color(150, 150, 150));

for (int i = 0; i < 9; i++) {
    buttons[i] = new JButton();
    button_panel.add(buttons[i]);
    buttons[i].setFont(new Font("MV Boli", Font.BOLD, 120));
    buttons[i].setFocusable(false);
    buttons[i].addActionListener(this);
}

this.setPreferredSize(new Dimension(800, 800));
this.setBackground(new Color(50, 50, 50));
this.setFocusable(true);
this.setLayout(null);

this.add(textfield);
this.add(button_panel);

firstTurn();
}

@Override
public void actionPerformed(ActionEvent e) {
    for (int i = 0; i < 9; i++) {
        if (e.getSource() == buttons[i]) {
            if (playerX_turn) {
                if (buttons[i].getText().equals("")) {
                    buttons[i].setForeground(Color.RED);
                    buttons[i].setText("X");
                    playerX_turn = false;
                    textfield.setText(player0 + " turn");
                    check();
                }
            } else {
                if (buttons[i].getText().equals("")) {
                    buttons[i].setForeground(Color.BLUE);
                    buttons[i].setText("O");
                    playerX_turn = true;
                    textfield.setText(playerX + " turn");
                    check();
                }
            }
        }
    }
}

```

```

        }
    }

public void check() {
    // (... existing code ...)
}

public void firstTurn() {
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    if (random.nextInt(2) == 0) {
        playerX_turn = true;
        textfield.setText(playerX + " turn");
    } else {
        playerX_turn = false;
        textfield.setText(player0 + " turn");
    }
}

public void oWins(int a, int b, int c) {
    buttons[a].setBackground(Color.GREEN);
    buttons[b].setBackground(Color.GREEN);
    buttons[c].setBackground(Color.GREEN);

    for (int i = 0; i < 9; i++) {
        buttons[i].setEnabled(false);
    }
    optionPanel(player0, 0);
}

public void xWins(int a, int b, int c) {
    buttons[a].setBackground(Color.GREEN);
    buttons[b].setBackground(Color.GREEN);
    buttons[c].setBackground(Color.GREEN);

    for (int i = 0; i < 9; i++) {
        buttons[i].setEnabled(false);
    }
    optionPanel(playerX, 1);
}

public void optionPanel(String name, int flag) {

    textfield.setBounds(0, 0, 800, 40);
    textfield.setFont(new Font("Ink Free", Font.BOLD, 25));
    textfield.setText(name + " Wins");

    saveAndExit.addActionListener(new ActionListener() {

```

```
@Override
public void actionPerformed(ActionEvent e) {
    winner = flag;
    buttonClicked = true;
    SwingUtilities.getWindowAncestor(saveAndExit).dispose();
}

});

exit.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        winner = -1;
        buttonClicked = true;
        SwingUtilities.getWindowAncestor(exit).dispose();
    }
});

this.add(playAgain);
this.add(exit);
this.add(saveAndExit);
}

public boolean isButtonClicked() {
    return buttonClicked;
}

}
```

PACKAGE 5 : gamepong

Program 1 : gamepong\PongGameFrame.java

```
package gamepong;

import javax.swing.JFrame;
import javax.swing.SwingUtilities;
import javax.swing.border.Border;
import java.awt.Color;
import java.util.ArrayList;

public class PongGameFrame extends JFrame {

    PongGamePanel ponggamepanel;
    ResultPanel resultpanel;
    private String playerRed, playerBlue;
    ArrayList<Integer> data = new ArrayList<>();

    public PongGameFrame(String playerRed, String playerBlue) {

        this.playerBlue = playerBlue;
        this.playerRed = playerRed;
        ponggamepanel = createPanel();

        SwingUtilities.invokeLater(() -> {

            Border customBorder = new CustomBorder(5);
            this.getRootPane().setBorder(customBorder);
            this.add(ponggamepanel);
            this.setUndecorated(true);
            this.setResizable(false);
            this.setBackground(Color.black);
            this.pack();
            this.setVisible(true);
            this.setLocationRelativeTo(null);

        });
    }

    public ArrayList<Integer> getWinner() {

        while (ponggamepanel.score.playerRed < 5 && ponggamepanel.score.playerBlue < 5) {
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        data.add(ponggamepanel.score.playerRed);
        data.add(ponggamepanel.score.playerBlue);
        if (ponggamepanel.score.playerRed == 5) {
            data.add(1);
        }
    }
}
```

```

    } else {
        data.add(0);
    }

    this.getContentPane().removeAll();
    ponggamepanel.stopGame();
    resultpanel = new ResultPanel(data, playerRed, playerBlue);
    this.getContentPane().add(resultpanel);
    resultpanel.requestFocusInWindow();
    this.revalidate();
    this.repaint();

    return data;
}

private PongGamePanel createPanel() {
    return new PongGamePanel(playerBlue, playerRed);
}
}
}

```

Program 2 : gamepong\PongGamePanel.java

```

package gamepong;

import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.Random;
import javax.swing.JPanel;

public class PongGamePanel extends JPanel implements Runnable {

    String playerRed, playerBlue;

    static final int GAME_WIDTH = 1300;
    static final int GAME_HEIGHT = (int) (GAME_WIDTH * (0.5555));
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH, GAME_HEIGHT);

    static final int BALL_DIAMETER = 20;
    static final int PADDLE_WIDTH = 25;
    static final int PADDLE_HEIGHT = 100;

    private volatile boolean running = true;

    Thread gameThread;
    Image image;

```

```

Graphics graphics;
Random random;
Paddle paddleRed;
Paddle paddleBlue;
Ball ball;
Score score;

PongGamePanel(String playerRed, String playerBlue) {

    newPaddles();
    newBall();

    this.playerBlue = playerBlue;
    this.playerRed = playerRed;

    score = new Score(GAME_WIDTH, GAME_HEIGHT);

    this.setPreferredSize(SCREEN_SIZE);

    this.setFocusable(true);
    this.addKeyListener(new AL());
    this.setLayout(null);

    gameThread = new Thread(this);
    gameThread.start();
}

public void stopGame() {
    running = false;
}

public void newBall() {

    random = new Random();
    ball = new Ball((GAME_WIDTH / 2) - (BALL_DIAMETER / 2), random.nextInt(GAME_HEIGHT
    - BALL_DIAMETER),
        BALL_DIAMETER, BALL_DIAMETER);

}

public void newPaddles() {

    paddleRed = new Paddle(0, (GAME_HEIGHT / 2) - (PADDLE_HEIGHT / 2), PADDLE_WIDTH,
    PADDLE_HEIGHT, 1);
    paddleBlue = new Paddle(GAME_WIDTH - PADDLE_WIDTH, (GAME_HEIGHT / 2) -
    (PADDLE_HEIGHT / 2), PADDLE_WIDTH,
        PADDLE_HEIGHT, 0);
}

public void paint(Graphics g) {

    image = createImage(getWidth(), getHeight());
    graphics = image.getGraphics();
}

```

```

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

}

public void draw(Graphics g) {

paddleRed.draw(g);
paddleBlue.draw(g);
ball.draw(g);
score.draw(g);

}

public void checkCollision() {

if (ball.y <= 0) {
    ball.setYDirection(-ball.yVelocity);
}
if (ball.y >= GAME_HEIGHT - BALL_DIAMETER) {
    ball.setYDirection(-ball.yVelocity);
}

if (ball.intersects(paddleRed)) {
    ball.xVelocity = Math.abs(ball.xVelocity);
    ball.xVelocity++;
    if (ball.yVelocity > 0) {
        ball.yVelocity++;
    } else {
        ball.yVelocity--;
    }
    ball.setxDirection(ball.xVelocity);
    ball.setyDirection(ball.yVelocity);
}

if (ball.intersects(paddleBlue)) {
    ball.xVelocity = Math.abs(ball.xVelocity);
    ball.xVelocity++;
    if (ball.yVelocity > 0) {
        ball.yVelocity++;
    } else {
        ball.yVelocity--;
    }
    ball.setxDirection(-ball.xVelocity);
    ball.setyDirection(ball.yVelocity);
}

if (paddleRed.y <= 0) {
    paddleRed.y = 0;
}
if (paddleRed.y >= GAME_HEIGHT - PADDLE_HEIGHT) {
    paddleRed.y = GAME_HEIGHT - PADDLE_HEIGHT;
}
}

```

```

if (paddleBlue.y <= 0) {
    paddleBlue.y = 0;
}
if (paddleBlue.y >= GAME_HEIGHT - PADDLE_HEIGHT) {
    paddleBlue.y = GAME_HEIGHT - PADDLE_HEIGHT;
}

if (ball.x <= 0) {
    score.playerBlue++;
    newPaddles();
    newBall();
    System.out.println("Player Blue: " + score.playerBlue);
}

if (ball.x >= GAME_WIDTH - BALL_DIAMETER) {
    score.playerRed++;
    newPaddles();
    newBall();
    System.out.println("Player Red: " + score.playerRed);
}

}

public void move() {

    paddleRed.move();
    paddleBlue.move();
    ball.move();

}

public void run() {

    long lastTime = System.nanoTime();
    double amountOfTicks = 60.0;
    double ns = 1000000000 / amountOfTicks;
    double delta = 0;
    while (running) {
        long now = System.nanoTime();
        delta += (now - lastTime) / ns;
        lastTime = now;
        if (delta >= 1) {
            move();
            checkCollision();
            repaint();
            delta--;
        }
    }
}

public class AL extends KeyAdapter {

```

```

@Override
public void keyPressed(KeyEvent e) {

    paddleRed.keyPressed(e);
    paddleBlue.keyPressed(e);

}

@Override
public void keyReleased(KeyEvent e) {

    paddleRed.keyReleased(e);
    paddleBlue.keyReleased(e);

}

}

```

Program 3 : gamepong\Ball.java

```

package gamepong;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Rectangle;
import java.util.Random;

public class Ball extends Rectangle {

```

```
    Random random;
```

```
    int xVelocity;
```

```
    int yVelocity;
```

```
    int initialSpeed = 2;
```

```
    Ball(int x, int y, int width, int height) {
```

```
        super(x, y, width, height);
```

```
        random = new Random();
```

```

int randomXDirection = random.nextInt(2);

if (randomXDirection == 0) {

    randomXDirection--;

}

setXDirection(randomXDirection * initialSpeed);

int randomYDirection = random.nextInt(2);

if (randomYDirection == 0) {

    randomYDirection--;

}

setYDirection(randomYDirection * initialSpeed);

}

public void setXDirection(int randomXDirection) {

    xVelocity = randomXDirection;

}

public void setYDirection(int randomYDirection) {

    yVelocity = randomYDirection;

}

public void move() {

    x += xVelocity;

    y += yVelocity;

}

public void draw(Graphics g) {

    g.setColor(Color.white);

    g.fillOval(x, y, width, height);

}

}

```

Program 4 : gamepong\Paddle.java

```
package gamepong;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Rectangle;
import java.awt.event.KeyEvent;

public class Paddle extends Rectangle {

    int id; // 1 for PlayerRed , 2 for PlayerBlue
    int yVelocity;
    int speed = 10;

    Paddle(int x, int y, int PADDLE_WIDTH, int PADDLE_HEIGHT, int id) {
        super(x, y, PADDLE_WIDTH, PADDLE_HEIGHT);
        this.id = id;
    }

    public void keyPressed(KeyEvent e) {
        switch (id) {

            case 1: // Red
                if (e.getKeyCode() == KeyEvent.VK_W) {
                    setYDirection(-speed);
                    move();
                }
                if (e.getKeyCode() == KeyEvent.VK_S) {
                    setYDirection(speed);
                    move();
                }
                break;

            case 0: // Blue
                if (e.getKeyCode() == KeyEvent.VK_UP) {
                    setYDirection(-speed);
                    move();
                }
                if (e.getKeyCode() == KeyEvent.VK_DOWN) {
                    setYDirection(speed);
                    move();
                }
                break;
        }
    }
}
```

```

public void keyReleased(KeyEvent e) {
    switch (id) {
        case 1:
            if (e.getKeyCode() == KeyEvent.VK_W) {
                setYDirection(0);
                move();
            }
            if (e.getKeyCode() == KeyEvent.VK_S) {
                setYDirection(0);
                move();
            }
            break;
        case 0:
            if (e.getKeyCode() == KeyEvent.VK_UP) {
                setYDirection(0);
                move();
            }
            if (e.getKeyCode() == KeyEvent.VK_DOWN) {
                setYDirection(0);
                move();
            }
            break;
    }
}

```

```
public void setYDirection(int yDirection) {
```

```
    yVelocity = yDirection;
```

```
}
```

```
public void move() {
```

```
    y = y + yVelocity;
```

```
}
```

```
public void draw(Graphics g) {
```

```
    if (id == 1) {
        g.setColor(Color.red);
    } else {
        g.setColor(Color.blue);
    }
}
```

```
    g.fillRect(x, y, width, height);
```

```
}
```

```
}
```

Program 5 : gamepong\ResultPanel.java

```
package gamepong;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;

public class ResultPanel extends JPanel {

    String red;
    String blue;
    int redscore;
    int bluescore;
    int id;

    JButton exitButton = new JButton("EXIT");
    JTextField nameTextField = new JTextField();
    JTextField scoreTextField = new JTextField();
    JLabel winner = new JLabel();

    ResultPanel(ArrayList<Integer> data, String playerRed, String playerBlue) {
        red = playerRed;
        blue = playerBlue;
        redscore = data.get(0);
        bluescore = data.get(1);
        id = data.get(2);

        nameTextField.setBounds(350, 50, 600, 150);
        nameTextField.setBackground(new Color(25, 25, 25));
        nameTextField.setForeground(new Color(25, 255, 0));
        nameTextField.setFont(new Font("Ink Free", Font.BOLD, 65));
        nameTextField.setBorder(BorderFactory.createBevelBorder(1));
        nameTextField.setHorizontalAlignment(JTextField.CENTER);
        nameTextField.setEditable(false);

        scoreTextField.setBounds(350, 200, 600, 125);
        scoreTextField.setBackground(new Color(25, 25, 25));
        scoreTextField.setForeground(new Color(25, 255, 0));
        scoreTextField.setFont(new Font("Ink Free", Font.BOLD, 45));
        scoreTextField.setBorder(BorderFactory.createBevelBorder(1));
        scoreTextField.setHorizontalAlignment(JTextField.CENTER);
        scoreTextField.setEditable(false);
    }
}
```

```

winner.setBounds(350, 375, 600, 125);
winner.setBackground(new Color(25, 25, 25));
winner.setForeground(new Color(25, 255, 0));
winner.setFont(new Font("Ink Free", Font.BOLD, 25));
winner.setHorizontalAlignment(JLabel.CENTER);

exitButton.setBounds(525, 550, 250, 100);
exitButton.setFont(new Font("Arial", Font.BOLD, 25));
exitButton.setBackground(new Color(25, 255, 0));
exitButton.setForeground(Color.white);
exitButton.setFocusable(false);

exitButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        SwingUtilities.getWindowAncestor(exitButton).dispose();
    }
});

this.add(exitButton);
this.add(winner);
this.add(nameTextField);
this.add(scoreTextField);

this.setBackground(new Color(50, 50, 50));
this.setLayout(null);
this.setFocusable(true);

displayResults();
}

public void displayResults() {

    nameTextField.setText(red + " vs " + blue);
    scoreTextField.setText(redscore + " : " + bluescore);

    if (id == 1) {
        winner.setText(red + " WINS");
    } else {
        winner.setText(blue + " WINS");
    }

}
}

```

Program 6 : gamepong\Score.java

```
package gamepong;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;

public class Score {

    static int GAME_WIDTH;
    static int GAME_HEIGHT;
    int playerRed;
    int playerBlue;

    Score(int GAME_WIDTH, int GAME_HEIGHT) {

        Score.GAME_WIDTH = GAME_WIDTH;
        Score.GAME_HEIGHT = GAME_HEIGHT;

    }

    public void draw(Graphics g) {

        g.setColor(Color.white);
        g.setFont(new Font("Consolas", Font.PLAIN, 60));

        g.drawLine(GAME_WIDTH / 2, 0, GAME_WIDTH / 2, GAME_HEIGHT);

        g.drawString(String.valueOf(playerRed / 10) +
            String.valueOf(playerRed % 10), (GAME_WIDTH / 2) - 85, 50);
        g.drawString(String.valueOf(playerBlue / 10) +
            String.valueOf(playerBlue % 10), (GAME_WIDTH / 2) + 20, 50);

    }

}
```

Program 7 : gamepong\CustomBorder.java &

Program 3 : gamesnake\CustomBorder.java &

Program 3 : gametictactoe\CustomBorder.java &

Program 3 : gamejavaquiz\CustomBorder.java

```
import java.awt.Color;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Insets;
```

```
import javax.swing.border.Border;

class CustomBorder implements Border {

    private final int borderWidth;

    public CustomBorder(int borderWidth) {
        this.borderWidth = borderWidth;
    }

    @Override
    public void paintBorder(Component c, Graphics g, int x, int y, int width, int height) {
        g.setColor(Color.BLACK);
        for (int i = 0; i < borderWidth; i++) {
            g.drawRect(x + i, y + i, width - 2 * i - 1, height - 2 * i - 1);
        }
    }

    @Override
    public Insets getBorderInsets(Component c) {
        return new Insets(borderWidth, borderWidth, borderWidth,
                           borderWidth);
    }

    @Override
    public boolean isBorderOpaque() {
        return true;
    }
}
```

PACKAGE 6 : database

Program 1 : database\DBConnector.java

```
package database;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.time.LocalDate;

public class DBConnector {

    private static Connection connection;

    private static final String URL = "jdbc:mysql://localhost:3306/gamegalaxy";
    private static final String USER = "root";
    private static final String PASSWORD = "password";

    static {
        try {
            // loading the jdbc driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // establishing connection
            connection = DriverManager.getConnection(URL, USER,
                                                     PASSWORD);

        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
            throw new RuntimeException("Failed to initialize the database
                                         connection.");
        }
    }

    // return connection
    public static Connection getConnection() {
        return connection;
    }

    // return date
    public static LocalDate getDate() {
        return LocalDate.now();
    }
}
```

Program 2 : database\InsertDataDB.java

```
package database;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

public class InsertDataDB {

    private static Connection connection;

    static {
        connection = DBConnector.getConnection();
    }

    public static void insertToSnakeTable(String player_name, int score) throws SQLException {
        String tablename = "snake_game";
        Date sqlDate = Date.valueOf(DBConnector.getDate()); // convert to
        database
        // sql query
        String sql = "INSERT INTO " + tablename + " (player_name, score,
        VALUES (?, ?, ?);"

        try (PreparedStatement stmt = connection.prepareStatement(sql)) {
            stmt.setString(1, player_name);
            stmt.setInt(2, score);
            stmt.setDate(3, sqlDate);

            // execute query
            stmt.executeUpdate();
            System.out.println("\n\n\t_____ data in snake game table
            inserted successfully_____");

        }
    }

    public static void insertToJavaQuizTable(String player_name, int
    throws SQLException {
        String tablename = "java_quiz_game";
        Date sqlDate = Date.valueOf(DBConnector.getDate()); // convert to
        database
        // sql query
        String sql = "INSERT INTO " + tablename + " (player_name, score,
        VALUES (?, ?, ?);"

        try (PreparedStatement stmt = connection.prepareStatement(sql)) {
```

```

stmt.setString(1, player_name);
stmt.setInt(2, score);
stmt.setDate(3, sqlDate);

// execute query
stmt.executeUpdate();
System.out.println("\n\n\t_____ data in java quiz table
inserted successfully_____");
}

}

public static void insertToPaddleTable(String player_red, String           player_blue,
ArrayList<Integer> data) throws SQLException {
    String tablename = "paddle_game";
    Date sqlDate = Date.valueOf(DBConnector.getDate()); // convert to           java sql
database

    // sql query
    String sql = "INSERT INTO " + tablename
        + " (player_red, player_blue, player_red_score,           player_blue_score,
winner, date) VALUES (?, ?, ?, ?, ?, ?)";

    try (PreparedStatement stmt = connection.prepareStatement(sql)) {
        stmt.setString(1, player_red);
        stmt.setString(2, player_blue);
        stmt.setInt(3, data.get(0));
        stmt.setInt(4, data.get(1));
        if (data.get(2) == 1) {
            stmt.setString(5, player_red);
        }
        if (data.get(2) == 0) {
            stmt.setString(5, player_blue);
        }
        stmt.setDate(6, sqlDate);

        // execute query
        stmt.executeUpdate();
        System.out.println("\n\n\t_____ data in paddle table
inserted successfully_____");
    }
}

public static void insertToTicTacTable(String player_x, String           player_o, int winner) throws
SQLException {
    String tablename = "tic_tac_game";
    Date sqlDate = Date.valueOf(DBConnector.getDate()); // convert to           java sql database

    // sql query
    String sql = "INSERT INTO " + tablename + " (player_x, player_o,           winner,
date) VALUES (?, ?, ?, ?)";

    try (PreparedStatement stmt = connection.prepareStatement(sql)) {

```

```
stmt.setString(1, player_x);
stmt.setString(2, player_o);

if (winner == 1) {
    stmt.setString(3, player_x);
}
if (winner == 0) {
    stmt.setString(3, player_o);
}

stmt.setDate(4, sqlDate);

// execute query
stmt.executeUpdate();
System.out.println("\n\n\t_____ data in tic tac table
inserted successfully_____");
}

}
```

Program 3 : database\ReadDataDB.java

```
package database;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ReadDataDB {

    private static Connection connection;

    static {
        connection = DBConnector.getConnection();
    }

    public static void readFromSnakeTable() throws SQLException {
        String tablename = "snake_game";

        // sql query
        String sql = "SELECT * FROM " + tablename;

        try (PreparedStatement stmt = connection.prepareStatement(sql)) {

            ResultSet rsdt = stmt.executeQuery();

            while (rsdt.next()) {
                int id = rsdt.getInt("idsnake_game");
                String player_name = rsdt.getString("player_name");
                int score = rsdt.getInt("score");
                Date date = rsdt.getDate("date");

                System.out.println("id: " + id + ", Player: " +
Score: " + score + ", Date: " + date);
            }
        }
    }

    public static void readFromJavaQuizTable() throws SQLException {
        String tablename = "java_quiz_game";

        // sql query
        String sql = "SELECT * FROM " + tablename;

        try (PreparedStatement stmt = connection.prepareStatement(sql)) {

            ResultSet rsdt = stmt.executeQuery();

            while (rsdt.next()) {
                int id = rsdt.getInt("idjava_quiz_game");
```

```

        String player_name = rslt.getString("player_name");
        int score = rslt.getInt("score");
        Date date = rslt.getDate("date");

        System.out.println("id: " + id + ", Player: " +
Score: " + score + ", Date: " + date);
    }
}

public static void readFromPaddleTable() throws SQLException {

    String tablename = "paddle_game";

    // sql query
    String sql = "SELECT * FROM " + tablename;

    try (PreparedStatement stmt = connection.prepareStatement(sql)) {

        ResultSet rslt = stmt.executeQuery();

        while (rslt.next()) {
            int id = rslt.getInt("idpaddle_game");
            String playerRed = rslt.getString("player_red");
            String playerBlue = rslt.getString("player_blue");
            int playerRedScore = rslt.getInt("player_red_score");
            int playerBlueScore = rslt.getInt("player_blue_score");
            String winner = rslt.getString("winner");
            Date date = rslt.getDate("date");

            System.out.println("id: " + id + ", Player Red: " +
playerRed +
                ", Player Blue: " + playerBlue +
                ", Red Score: " + playerRedScore +
                ", Blue Score: " + playerBlueScore +
                ", Winner: " + winner +
                ", Date: " + date);
        }
    }
}

public static void readFromTicTacTable() throws SQLException {

    String tablename = "tic_tac_game";

    // sql query
    String sql = "SELECT * FROM " + tablename;

    try (PreparedStatement stmt = connection.prepareStatement(sql)) {

        ResultSet rslt = stmt.executeQuery();

        while (rslt.next()) {

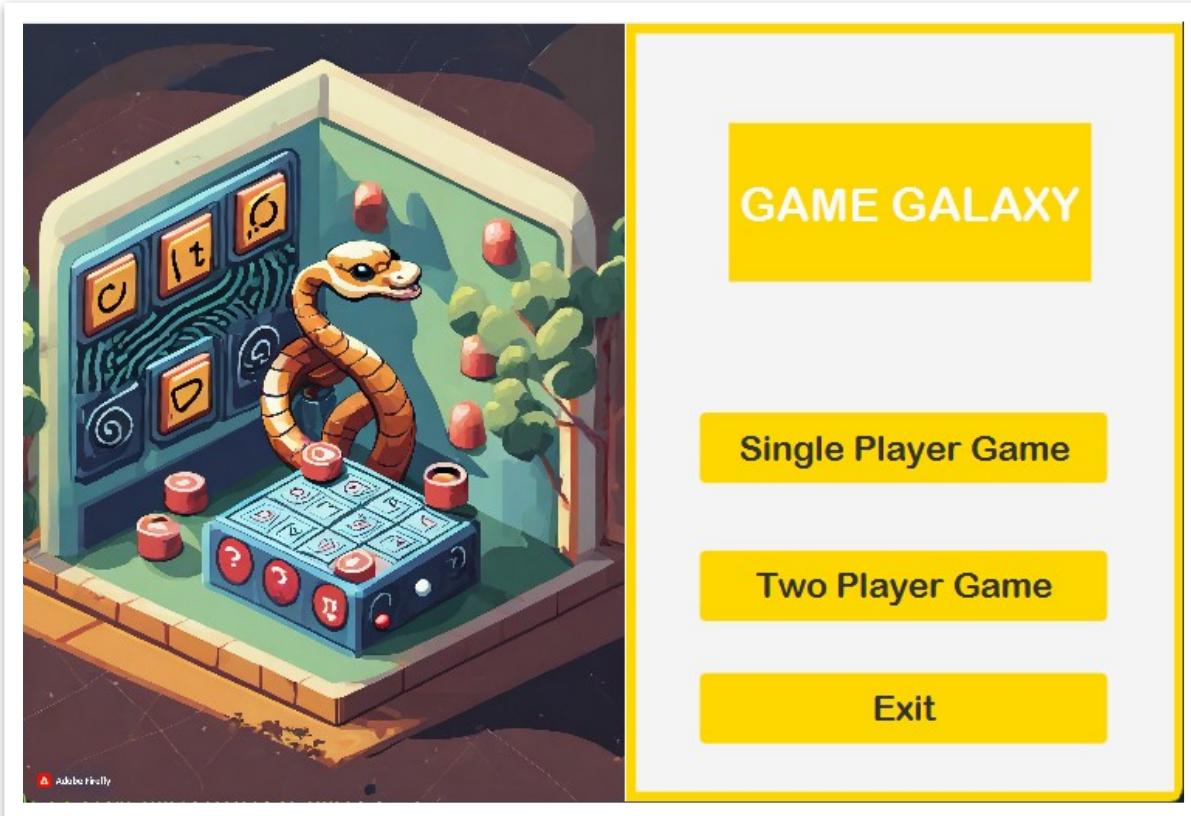
```

```
int id = rslt.getInt("idtic_tac");
String playerX = rslt.getString("player_x");
String playerO = rslt.getString("player_o");
String winner = rslt.getString("winner");
Date date = rslt.getDate("date");

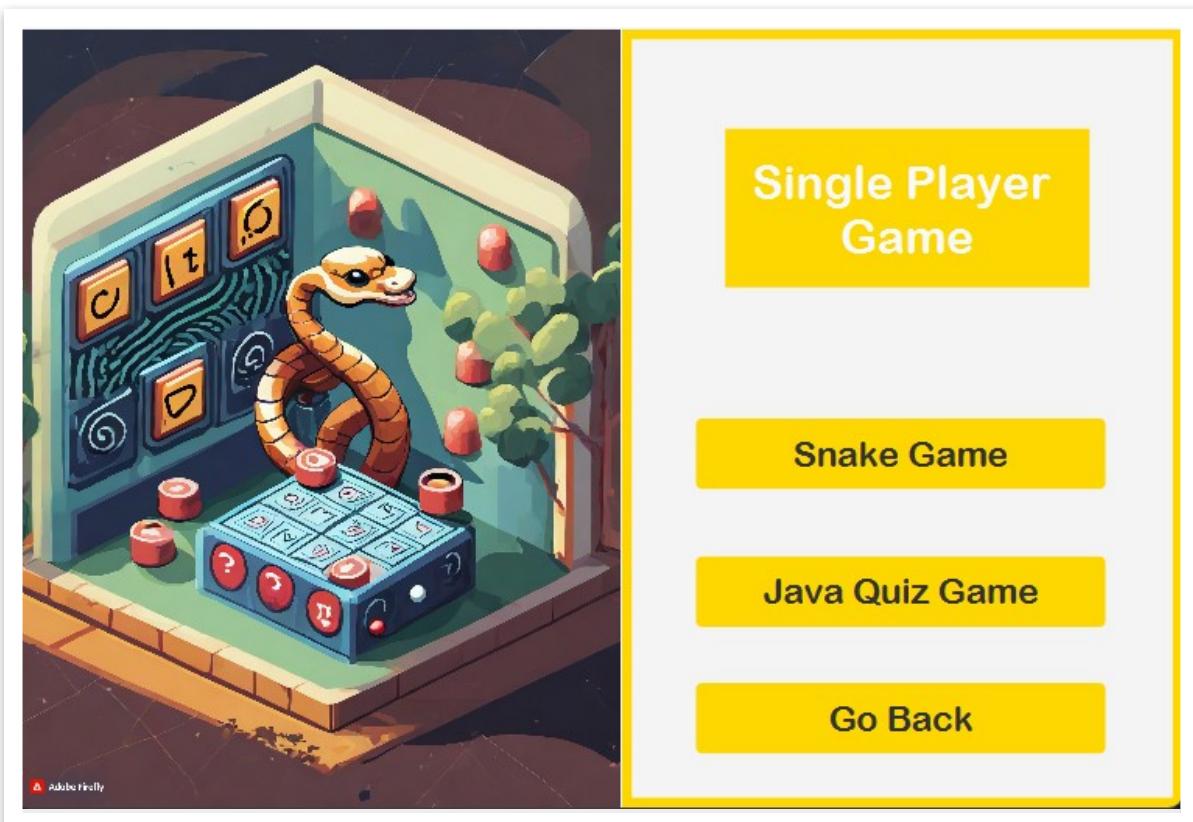
System.out.println("id: " + id + ", Player X: " + playerX +
    ", Player O: " + playerO +
    ", Winner: " + winner +
    ", Date: " + date);
}
}
}
```

SCREENSHOTS

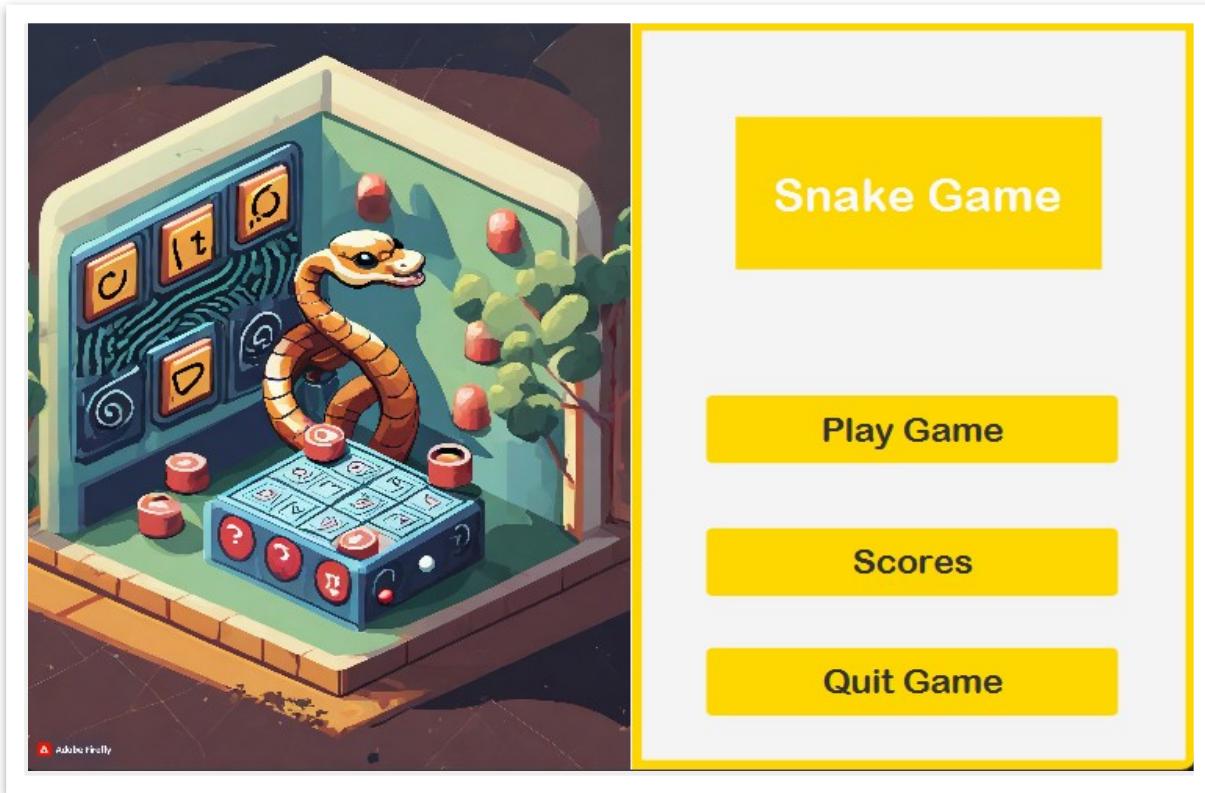
1) Front / Home Page :



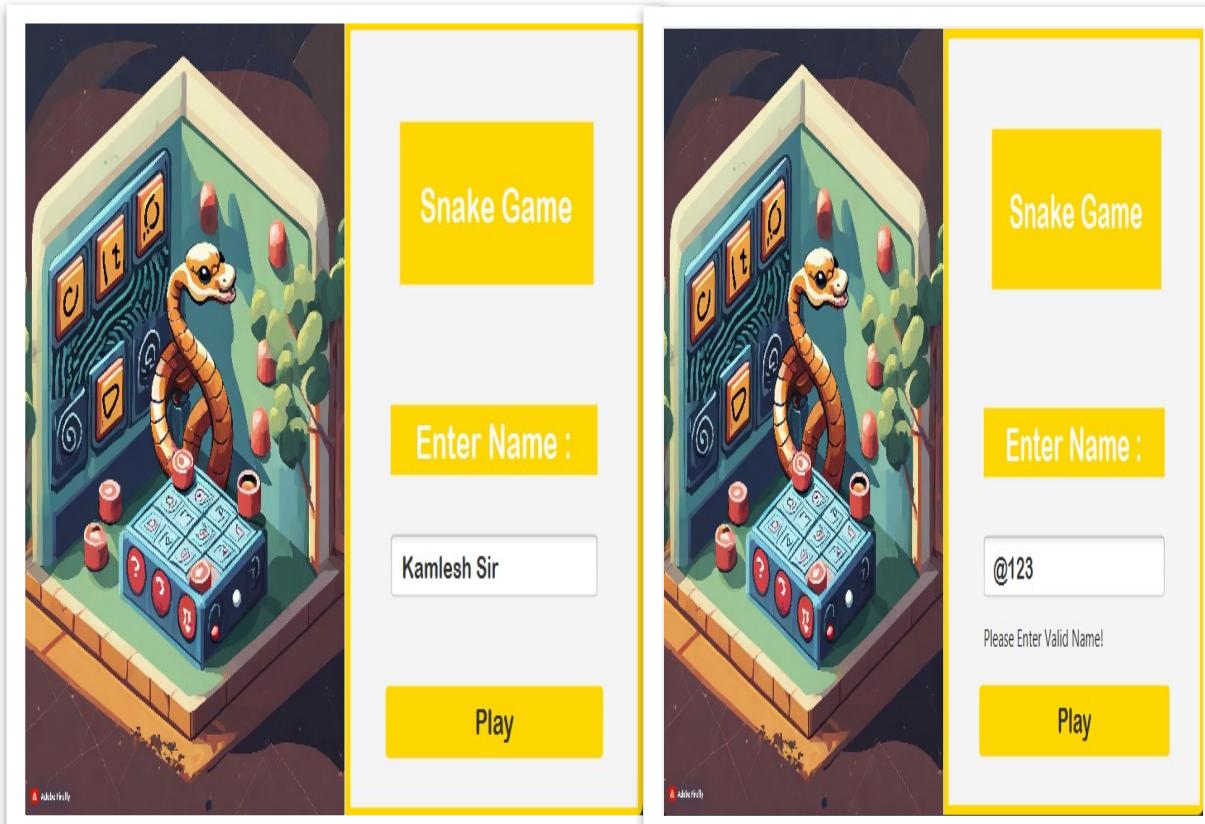
2) Single Player Page



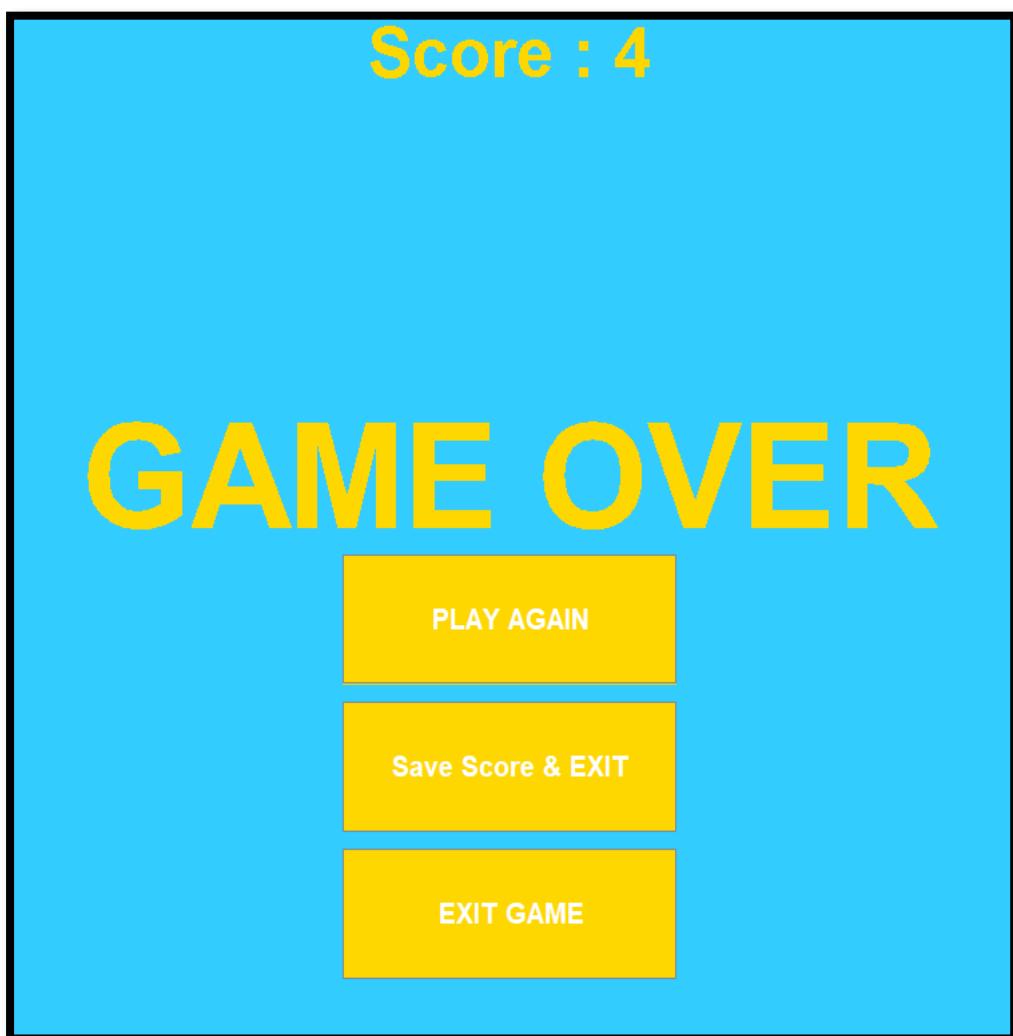
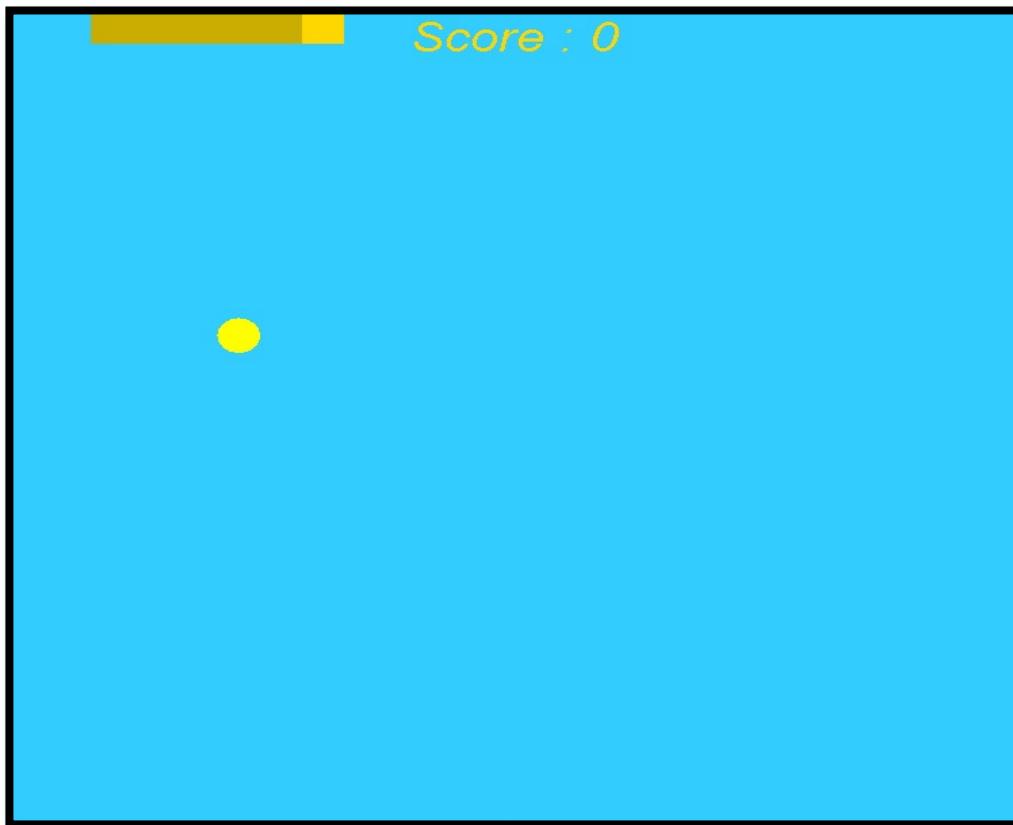
3) Snake Game Page



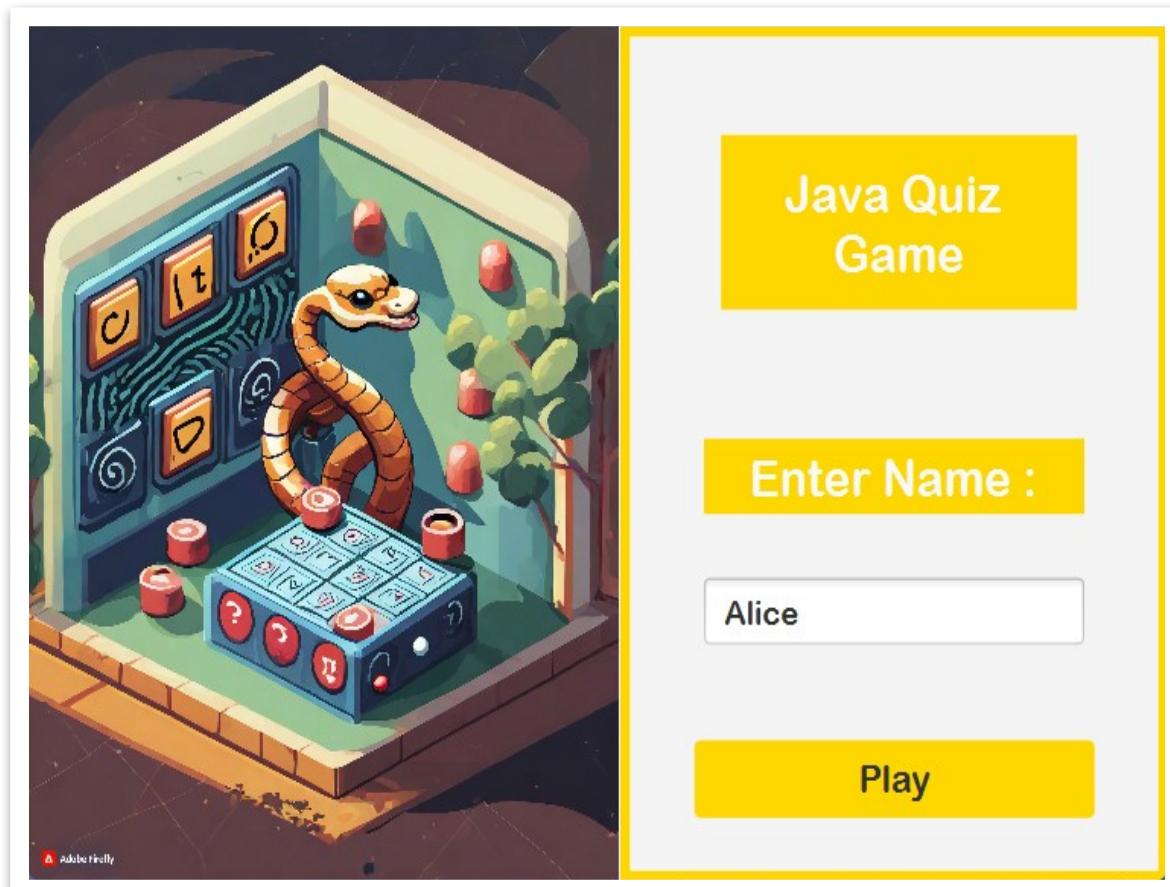
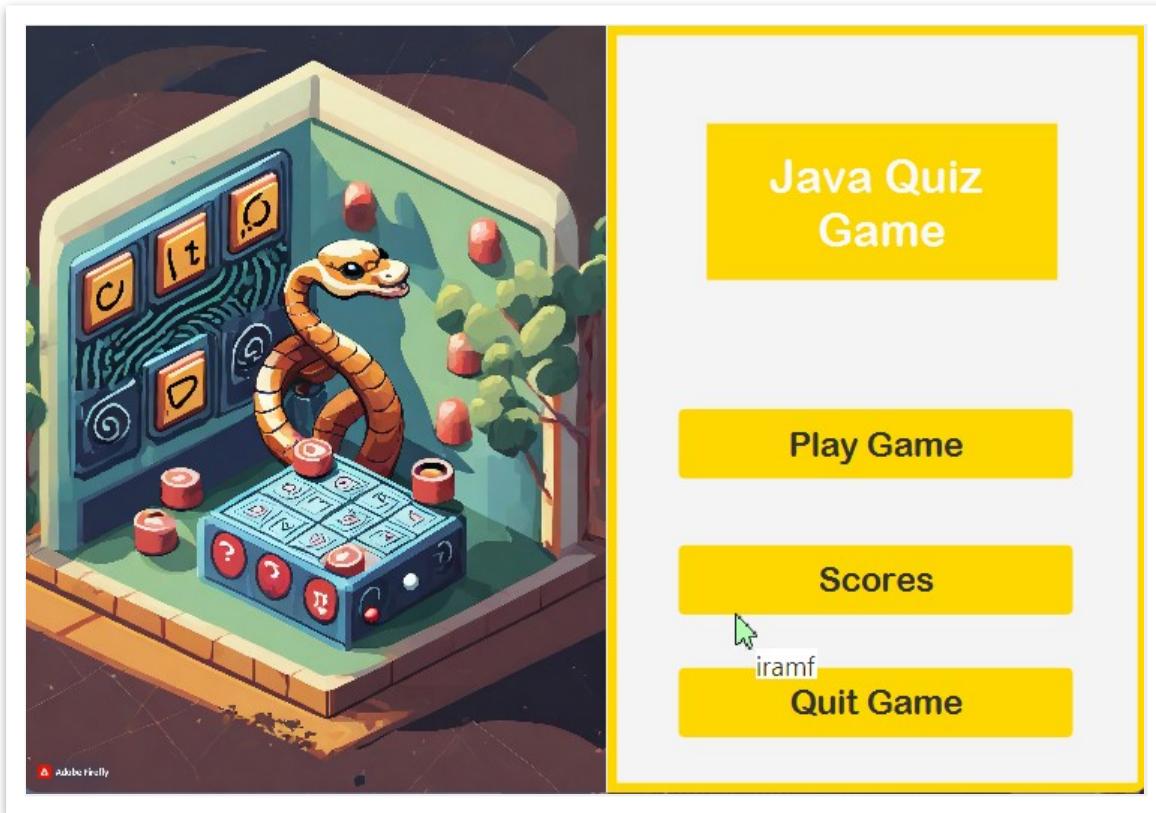
4) Input Page



4) Snake Game ScreenShot



5) Java Quiz Game Page



6) Java Quiz Game ScreenShot

Question 7

Which keyword is used to declare a constant in Java?

A	A: const
B	B: final
C	C: static
D	D: constant

timer >:D

[Quit Quiz](#)

19

RESULTS!

A	(12/18)
B	66%
C	
D	

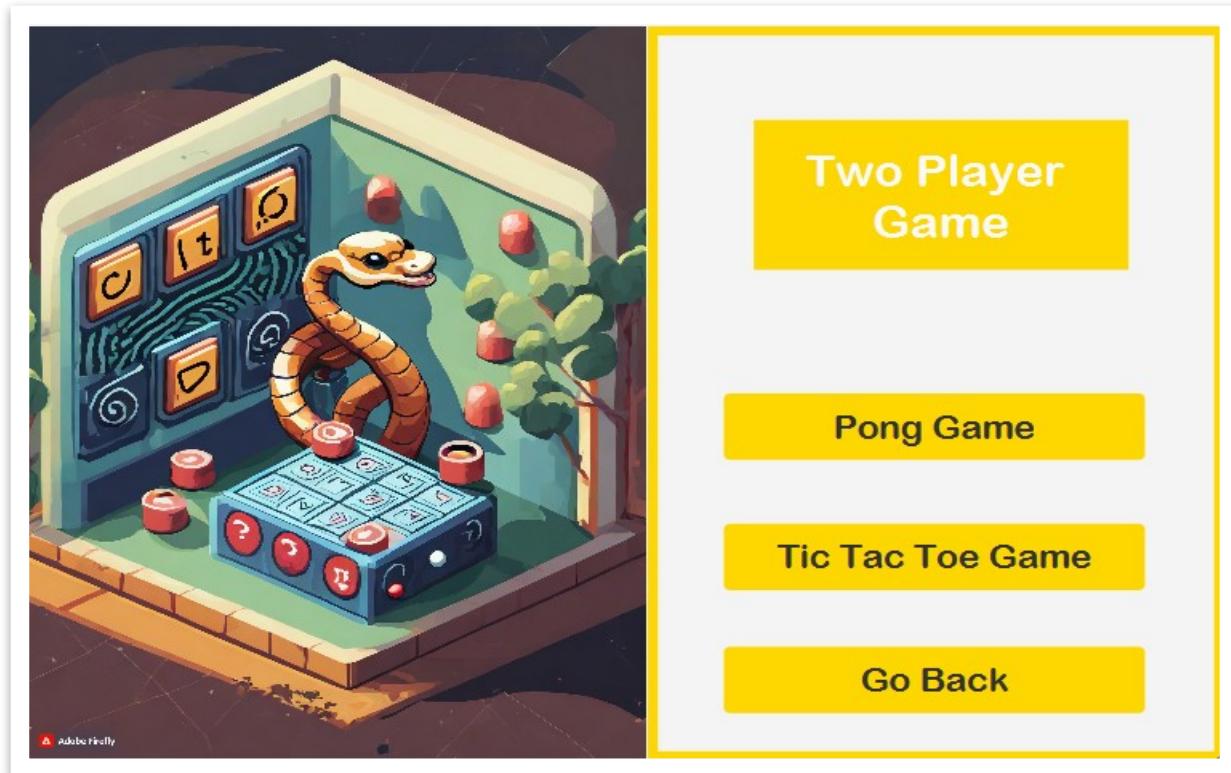
[SAVE AND EXIT](#)

[Exit](#)

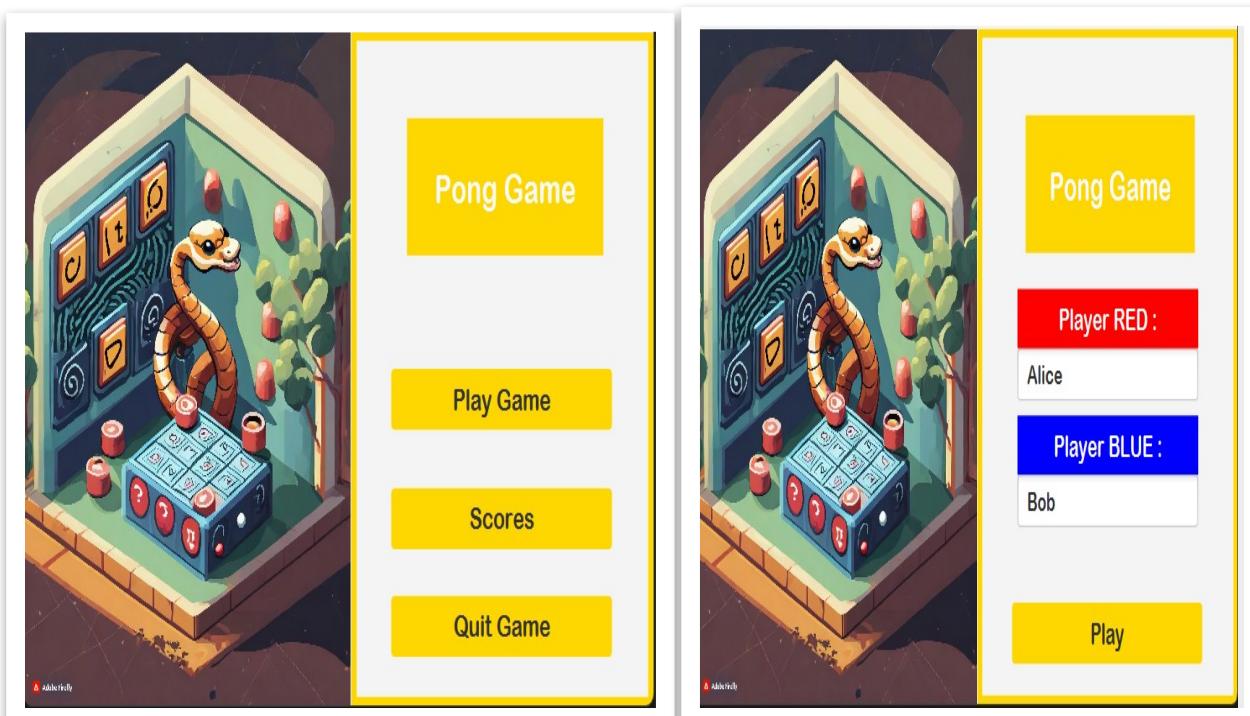
timer >:D

30

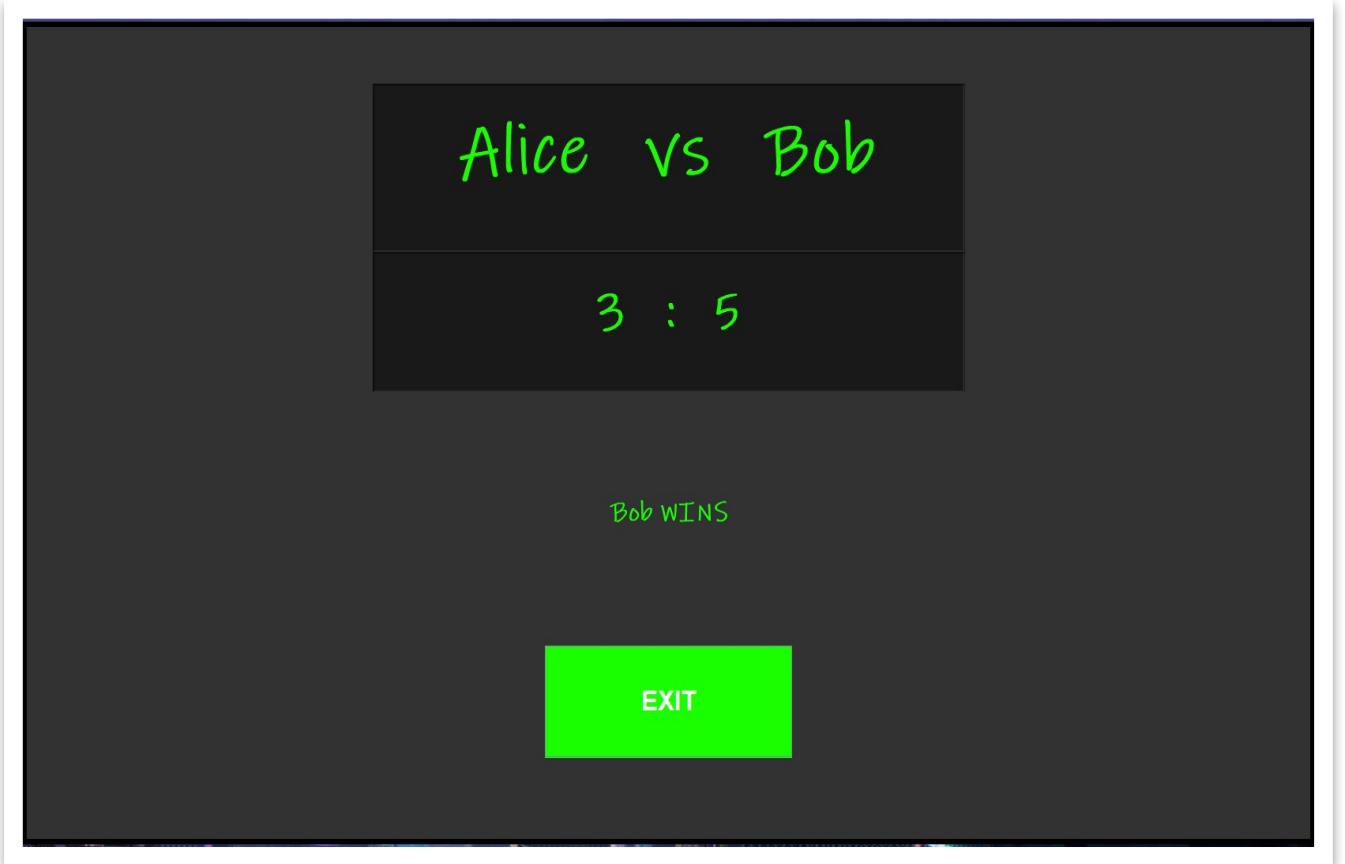
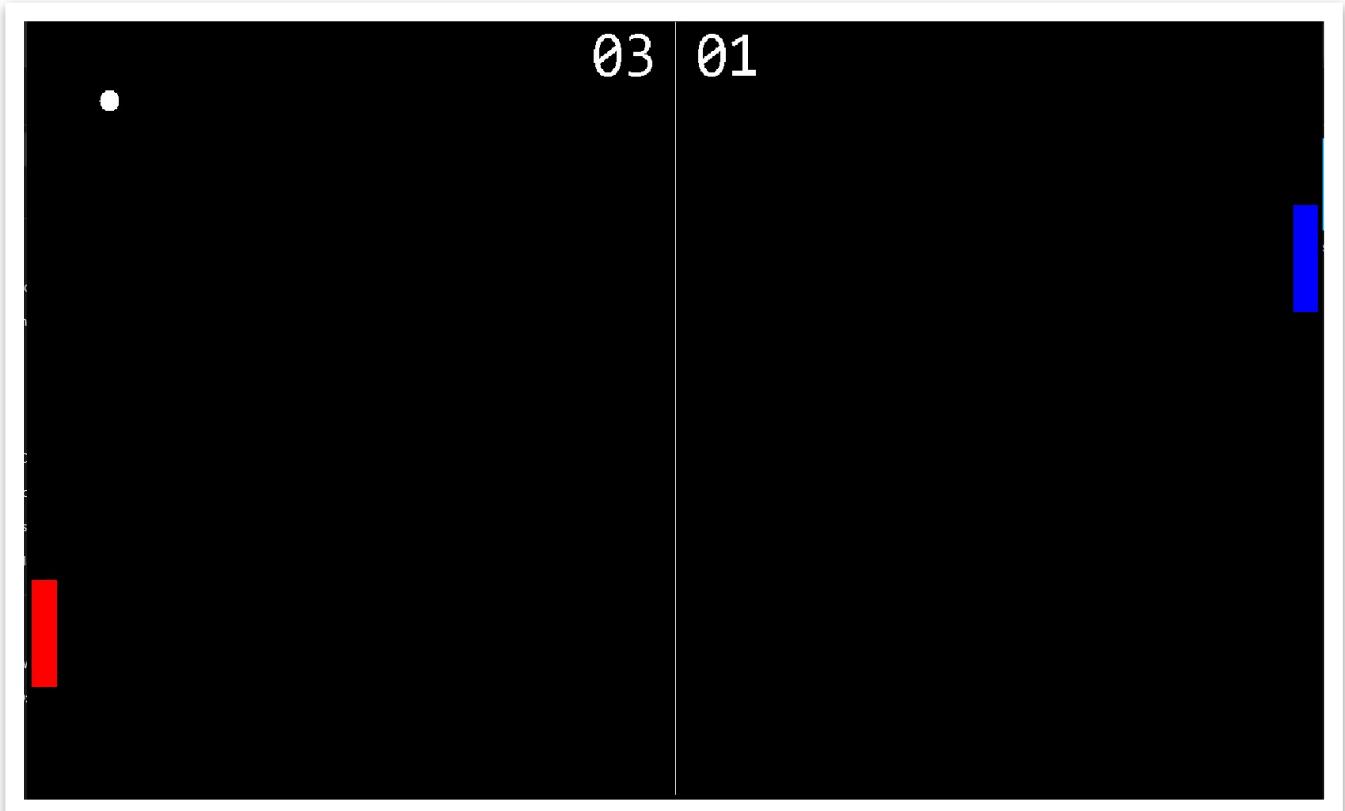
7) Two Player Game



8) Pong Game Page



9) Pong Game SnapShot



10) Tic Tac Toe Page

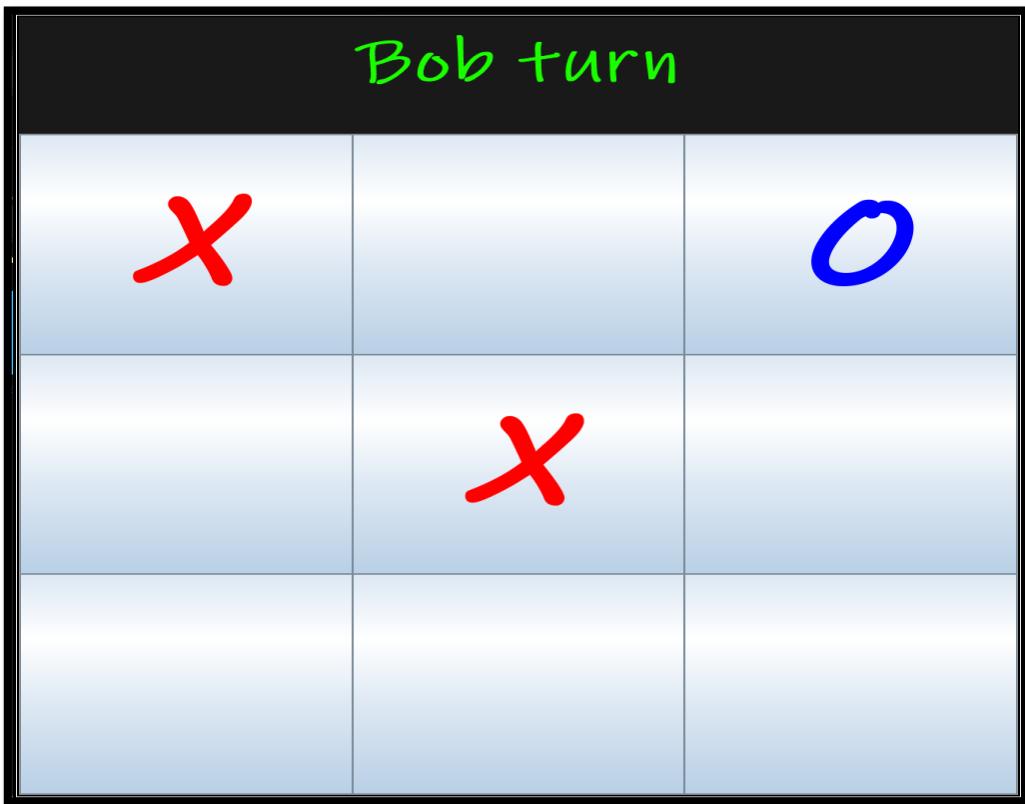


The image shows a mobile game interface for a Tic Tac Toe game. On the left is a decorative illustration of a golden-yellow snake coiled around a blue control panel with red buttons. The right side features a yellow header bar with the text "Tic Tac Toe Game". Below the header are three yellow rectangular buttons with black text: "Play Game", "Scores", and "Quit Game".



The image shows a mobile game interface for a Tic Tac Toe game. It features the same snake and control panel illustration on the left. The right side has a yellow header bar with the text "TicTacToe Game". Below the header are two sets of input fields: "Player X :" with the name "Alice" and "Player 0 :" with the name "Bob". At the bottom is a large yellow button labeled "Play".

10) Tic Tac Toe SnapShot



TESTING

Testing is a crucial part of any software development process, including game development. For our project, “Game Galaxy”, we have implemented a comprehensive testing strategy to ensure the quality and reliability of our game.

Unit Testing

In the initial stages, we performed unit testing on individual components of the game. This included testing the functionality of the game mechanics, user interface elements, and database interactions. We used JUnit, a widely-used testing tool for Java, to automate our unit tests and ensure that all components were functioning as expected.

Integration Testing

After unit testing, we moved on to integration testing. This involved testing the interaction between different components of the game. We checked if the game mechanics worked well with the user interface, and if the game state was correctly updated and reflected in the database.

System Testing

In system testing, we tested the game as a whole. We checked if all parts of the game worked together seamlessly to provide a smooth and enjoyable gaming experience. This included testing the game on different operating systems and devices to ensure compatibility.

Performance Testing

Performance testing was done to ensure that the game runs smoothly without any lags or crashes, even when played for extended periods. We also tested the game under different network conditions to ensure that multiplayer games run smoothly.

User Acceptance Testing

Finally, we conducted user acceptance testing. We had a group of users play the game and provide feedback. This helped us understand if the game met the users' expectations and if there were any areas we could improve.

Through rigorous testing, we have strived to ensure that “Game Galaxy” is not only fun to play but also reliable and free from bugs. We believe that this commitment to quality will greatly enhance the user experience.

Testing Process

1. Test Planning

- Define the scope, objectives, and approach for testing.
- Identify the test cases and scenarios for each type of testing.
- Determine the resources, schedule, and deliverables for the testing phase.

2. Test Case Development

- Develop detailed test cases for unit, integration, and system testing.
- Ensure test cases cover all possible scenarios, including edge cases and error conditions.

3. Test Execution

- Execute the test cases and document the results.
- Track any defects or issues discovered during testing and assign them for resolution.

4. Defect Management

- Use a defect tracking tool to record and manage defects.
- Prioritize defects based on their severity and impact on the game.
- Fix the defects and retest to ensure they have been resolved.

5. Reporting and Review

- Compile test reports summarizing the testing activities, results, and any outstanding issues.
- Review the test reports with the development team and stakeholders.

6. Final Validation

- Perform a final round of testing to validate that all major defects have been resolved and the game is ready for release.
- Ensure the game meets all acceptance criteria and is ready for deployment.

IMPLEMENTATION & MAINTENANCE

Implementation is the phase where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system.

For “Game Galaxy”, the implementation phase involved the following steps:

- 1. Environment Setup:** We set up the development environment with the necessary tools and technologies, including JavaFX, Java Swing, and MYSQL Database.
- 2. Coding:** We followed the design documents to start coding the game. We adhered to best practices for coding in Java, including proper use of data structures, efficient algorithms, and object-oriented principles.
- 3. Integration:** After coding individual modules, we integrated them to form the complete game. This included integrating the game mechanics with the user interface and ensuring that the game state was correctly updated in the database.
- 4. Testing:** We performed rigorous testing at each stage of implementation to ensure that the game was functioning as expected. This included unit testing, integration testing, system testing, performance testing, and user acceptance testing.

Maintenance

Maintenance is the final phase of the software development life cycle. It involves making modifications to the system or an individual component to alter attributes or improve performance.

For “Game Galaxy”, the maintenance phase will involve:

- 1. Monitoring:** Regularly monitoring the game to identify any issues or areas for improvement.
- 2. Updating:** Based on feedback from users and our own observations, we will make updates to the game. This could include adding new features, improving game mechanics, or enhancing the user interface.
- 3. Bug Fixing:** If any bugs are identified during the monitoring process, we will work on fixing them to ensure a smooth gaming experience for the users.
- 4. Optimization:** We will continuously work on optimizing the game’s performance. This could involve optimizing the code, improving the efficiency of the database interactions, or enhancing the graphics.

SYSTEM SECURITY MEASURES

Introduction

Ensuring the security of "Game Galaxy" is crucial to protect the integrity of the game and the data it handles. This section outlines the security measures implemented to safeguard the application, data, and user information from potential threats.

However we haven't implemented any security measures in this project

Security Measures

1. Authentication and Authorization

- **User Authentication:** Implement a secure login system requiring users to authenticate with a username and password. Use hashing algorithms like bcrypt to securely store passwords.
- **Role-Based Access Control (RBAC):** Define roles and permissions to restrict access to certain features and administrative functionalities based on the user's role.

2. Database Security

- **SQL Injection Prevention:** Use prepared statements and parameterized queries to prevent SQL injection attacks, ensuring that user input is properly sanitized.
- **Regular Backups:** Perform regular backups of the database to prevent data loss and facilitate recovery in case of a security breach or data corruption.

3. Input Validation and Sanitization

- **Client-Side Validation:** Implement input validation on the client side to provide immediate feedback to users.
- **Server-Side Validation:** Validate and sanitize all user inputs on the server side to ensure data integrity and prevent malicious data from being processed.

FUTURE SCOPE OF THE PROJECT

Nothing is perfect in this world. So, we are also no exception. Although, we have tried our best to present the information effectively, yet, there can be further enhancement in the Application. We have taken care of all the critical aspects, which need to take care of during the development of the Project.

Like the things this project also has some limitations and can further be enhanced by some one, because there are certain drawbacks that do not permit the system to be 100% accurate potential directions and opportunities for the game beyond its initial development. Here are some ideas to include

1. Expansion of Game Content: includes more light weighted 2D Games
2. Platform Compatibility:
3. Include Sound and Music for each game
4. Include Multiplayer over the Internet

GLOSSARY

- **2D Game:** A video game that operates in a two-dimensional world, allowing movement in two directions, up/down and left/right.
- **JavaFX:** A software platform for creating and delivering desktop applications, as well as rich web applications that can run across a wide variety of devices.
- **Java Swing:** A set of program components for Java programmers that provide the ability to create graphical user interface (GUI) components, such as buttons and scroll bars, that are independent of the windowing system for specific operating system.
- **MYSQL Database:** An open-source relational database management system based on SQL (Structured Query Language).
- **Snake Game:** A game where the player maneuvers a line which grows in length, with the line itself being a primary obstacle.
- **Java Quiz Game:** A game that tests the player's knowledge of Java programming language.
- **Tic Tac Toe:** A classic game where two players take turns marking spaces in a 3x3 grid with X or O, the aim being to get three marks in a horizontal, vertical, or diagonal row.
- **Pong Game:** A table tennis-themed arcade video game, featuring simple two-dimensional graphics, where players control virtual paddles to hit the ball back and forth.
- **Single Player Game:** A game where game play proceeds without the need for player interaction. Game play is designed for one player.
- **Multi Player Game:** A game which allows multiple players to play simultaneously in an interactive environment.
- **User Interface (UI):** The means by which the user and a computer system interact, in particular the use of input devices and software.

BIBLIOGRAPHY

- **Websites:**
 - ➔ GeeksForGeeks
 - ➔ StackOverFlow
 - ➔ ChatGPT
 - ➔ JavaFX
 - ➔ Javatpoint
 - ➔ lucidchart
- **Videos:**
 - ➔ youtube.com/BroCode
 - ➔ youtube.com/telusko
- **Oracle Java Tutorials**
 - ➔ Oracle. (n.d.). *The Java™ Tutorials*. Retrieved from <https://docs.oracle.com/javase/tutorial/>
- **JavaFX Documentation**
 - ➔ Oracle. (n.d.). *JavaFX Documentation*. Retrieved from <https://openjfx.io/>