Report On

Title of the Course Project:

TIC TAC TOE GAME

Submitted in partial fulfillment of the requirements of the Course project in Semester III of Second Year Artificial Intelligence and Data Science

by
Sakshi Patil (Roll No. 44)
Saloni Sutar (Roll No.58)
Rutuja Pednekar (Roll No.47)

Supervisor:
Prof. Sneha Yadav

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence & Data Science Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligience & Data Science Engineering

CERTIFICATE

This is to certify that the project entitled “Tic Tac Toe” is a bonafide work of Sakshi Patil (Roll No. 44)Saloni Sutar (Roll No.58)Rutuja Pednekar (Roll No.48)submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Computer Engineering.

Supervisor

Prof. Sneha Yadav		
Internal Examinar Dr. Tatwadarshi P. N. Head of Department		External Examiner Dr. H.V. Vankudre Principal

Abstract

The Java-based Tic-Tac-Toe Game Project is a classic implementation of the popular tic-tac-toe game developed using Java programming language. This project focuses on creating an interactive and user-friendly interface for players to engage in the timeless game of tic-tac-toe.The game employs object-oriented programming concepts, utilizing Java's graphical user interface (GUI) libraries to design an intuitive game board. Players can compete against each other or against a computer opponent, demonstrating strategic thinking and decision-making skills. The project incorporates various algorithms to enable the computer player to make intelligent moves, enhancing the gameplay experience.

Key features of the project include a visually appealing game board, responsive user interface, and an efficient algorithm for gameplay logic. The project aims to provide an entertaining and educational tool for individuals interested in learning Java programming and game development fundamentals.

Through this project, users can explore the implementation of game mechanics, user input handling, and graphical representation in Java, making it an ideal resource for beginners and enthusiasts alike. The Tic-Tac-Toe Game Project not only offers entertainment but also serves as a practical example of applying Java programming concepts in real-world applications.

Contents

1.Problem Statement

Create a digital Tic Tac Toe game that faithfully follows the traditional rules while offering an engaging experience. The project's challenge is to implement, graphical representation of the game board, and real-time event notifications.

- **Block Diagram**
- **Description**

Key Features:

1. Interactive Interface:
The program provides a 3x3 grid of buttons, each representing a cell on the Tic-Tac-Toe board. Players can click these buttons to make their moves.

2. Player Turns:
The game alternates between two players, denoted as 'X' and 'O'. Each player's moves are indicated by their respective symbols on the grid.

3. Winning Conditions:
The game checks for winning conditions after every move. A player wins if they have three of their symbols in a row, column, or diagonal. When a player wins, a pop-up message informs the players of the winner.

4. Draw Detection:
If the grid is filled and no player has won, the game declares a draw. A draw occurs when all the cells are filled, and no player has achieved a winning combination**.**

5. Reset Functionality:
After a win or draw, the board can be reset for a new game. All the buttons are cleared, and players can start a fresh round.

6. Visual Design:
The interface is designed with a black background and white buttons. The buttons change color to blue when clicked, providing visual feedback to the players.

Implementation Details:

The code initializes a Swing `JFrame` containing an array of `JButton` objects. Each button is associated with an index representing its position on the grid. When a button is clicked, it triggers the `onButtonClick` method, updating the button's text, disabling it to prevent further moves, and checking for win or draw conditions.

Conclusion:

This Java-based Tic-Tac-Toe implementation not only demonstrates the fundamental concepts of GUI programming in Java but also provides an interactive platform for players to enjoy the classic game. Its clear and concise design makes it an excellent example for beginners learning about graphical user interfaces and event-driven programming in Java.

Working

The provided Java program implements a graphical Tic-Tac-Toe game using Swing components. The game interface consists of a 3x3 grid of buttons, where players take turns clicking buttons to make their moves. The program ensures that the game logic is implemented correctly, handling player turns, win conditions, draws, and resetting the board for a new game. The graphical user interface is designed with a clear layout, utilizing appropriate fonts and colors. Overall, the program provides an interactive and functional implementation of the classic Tic-Tac-Toe game for two players.

2. Module Description

Graphical Interface: Utilizes Swing components for a user-friendly game interface**.**

Player Turns: Alternates between 'X' and 'O' for player moves**.**

Win Detection: Checks for winning conditions horizontally, vertically, and diagonally**.**

Draw Detection: Detects when the game ends in a draw.

Reset Functionality: Allows players to start a new game after a win or draw.

2.1 Brief description of software& hardware used and its programming

Software:

Java: The programming language used to write the Tic-Tac-Toe game code. Java is an object-oriented, class-based language widely used for building various applications, including desktop and web applications**.**

Swing Library: The code utilizes the Swing library, a part of Java's Abstract Window Toolkit (AWT), for creating graphical user interfaces (GUI) in Java applications. Swing provides a set of components for building desktop applications, and in this case, it's used to create buttons, frames, and dialogs for the game interface.

Hardware:

- The Tic-Tac-Toe game implemented in this code does not have specific hardware requirements. It is a simple desktop application that can run on standard computers, laptops, or any device capable of executing Java applications.
- Since the game's interface is basic and the logic is not computationally intensive, it can run on most hardware configurations without any issues.
-
- **Code**

```
import java.util.*;

public class GFG {

    static String[] board;
    static String turn;


    // CheckWinner method will
    // decide the combination
    // of three box given below.
    static String checkWinner()
    {
        for (int a = 0; a < 8; a++) {
            String line = null;

            switch (a) {
            case 0:
                line = board[0] + board[1] + board[2];
                break;
            case 1:
                line = board[3] + board[4] + board[5];
                break;
            case 2:
                line = board[6] + board[7] + board[8];
                break;
            case 3:
                line = board[0] + board[3] + board[6];
                break;
            case 4:
                line = board[1] + board[4] + board[7];
                break;
            case 5:
                line = board[2] + board[5] + board[8];
                break;
            case 6:
                line = board[0] + board[4] + board[8];
                break;
            case 7:
                line = board[2] + board[4] + board[6];
                break;
            }
            //For X winner
            if (line.equals("XXX")) {
                return "X";
            }

            // For O winner
            else if (line.equals("OOO")) {
                return "O";
            }
        }

        for (int a = 0; a < 9; a++) {
            if (Arrays.asList(board).contains(
                    String.valueOf(a + 1))) {
                break;
            }
            else if (a == 8) {
                return "draw";
            }
        }

       // To enter the X Or O at the exact place on board.
        System.out.println(
            turn + "'s turn; enter a slot number to place "
            + turn + " in:");
        return null;
    }

    // To print out the board.
    /* |---|---|---|
       | 1 | 2 | 3 |
       |-----------|
       | 4 | 5 | 6 |
       |-----------|
       | 7 | 8 | 9 |
       |---|---|---|*/

    static void printBoard()
    {
        System.out.println("|---|---|---|");
        System.out.println("| " + board[0] + " | "
                           + board[1] + " | " + board[2]
                           + " |");
        System.out.println("|-----------|");
        System.out.println("| " + board[3] + " | "
                           + board[4] + " | " + board[5]
                           + " |");
        System.out.println("|-----------|");
        System.out.println("| " + board[6] + " | "
                           + board[7] + " | " + board[8]
                           + " |");
        System.out.println("|---|---|---|");
    }

    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        board = new String[9];
        turn = "X";
        String winner = null;

        for (int a = 0; a < 9; a++) {
            board[a] = String.valueOf(a + 1);
        }

        System.out.println("Welcome to 3x3 Tic Tac Toe.");
        printBoard();

        System.out.println(
            "X will play first. Enter a slot number to place X in:");

        while (winner == null) {
            int numInput;

            // Exception handling.
            // numInput will take input from user like from 1 to 9.
            // If it is not in range from 1 to 9.
            // then it will show you an error "Invalid input."
        try {
            numInput = in.nextInt();
            if (!(numInput > 0 && numInput <= 9)) {
                System.out.println(
                    "Invalid input; re-enter slot number:");
                continue;
            }
        }
        catch (InputMismatchException e) {
            System.out.println(
                "Invalid input; re-enter slot number:");
            continue;
        }

        // This game has two player x and O.
        // Here is the logic to decide the turn.
        if (board[numInput - 1].equals(
                String.valueOf(numInput))) {
            board[numInput - 1] = turn;

            if (turn.equals("X")) {
                turn = "O";
            }
            else {
                turn = "X";
            }

            printBoard();
            winner = checkWinner();
        }
        else {
            System.out.println(
                "Slot already taken; re-enter slot number:");
        }
    }

    // If no one win or lose from both player x and O.
    // then here is the logic to print "draw".
    if (winner.equalsIgnoreCase("draw")) {
        System.out.println(
            "It's a draw! Thanks for playing.");
    }

    // For winner -to display Congratulations! message.
```

```
    else {
      System.out.println(
        "Congratulations! " + winner
        + "'s have won! Thanks for playing.");
    }
   in.close();
  }
}
```

Output :

Below is the output of the above program :

```
Welcome to 3x3 Tic Tac Toe.
|---|---|---|
| 1 | 2 | 3 |
|-----------|
| 4 | 5 | 6 |
|-----------|
| 7 | 8 | 9 |
|---|---|---|
X will play first. Enter a slot number to place X in:
3
|---|---|---|
| 1 | 2 | X |
|-----------|
| 4 | 5 | 6 |
|-----------|
| 7 | 8 | 9 |
|---|---|---|
O's turn; enter a slot number to place O in:
5
|---|---|---|
| 1 | 2 | X |
|-----------|
| 4 | O | 6 |
|-----------|
| 7 | 8 | 9 |
|---|---|---|
X's turn; enter a slot number to place X in:
6
|---|---|---|
| 1 | 2 | X |
|-----------|
| 4 | O | X |
|-----------|
| 7 | 8 | 9 |
|---|---|---|
O's turn; enter a slot number to place O in:
1
|---|---|---|
| O | 2 | X |
|-----------|
| 4 | O | X |
|-----------|
| 7 | 8 | 9 |
|---|---|---|
X's turn; enter a slot number to place X in:
9
|---|---|---|
| O | 2 | X |
|-----------|
| 4 | O | X |
|-----------|
| 7 | 8 | X |
|---|---|---|
Congratulations! X's have won! Thanks for playing.
```

4. Result and Conclusion

Result:

The Tic-Tac-Toe game project was successfully implemented using Java. The graphical user interface (GUI) displayed a 3x3 grid with buttons where players could make their moves. The game mechanics allowed players to take turns, indicating their moves with 'X' or 'O'. The program accurately detected winning combinations and declared the corresponding player as the winner. In the case of a draw, the game correctly identified the situation and displayed the appropriate message.

Conclusion:

In conclusion, the project demonstrated a strong understanding of Java's Swing framework and event-driven programming. The game's user interface was intuitive and responsive, providing an enjoyable gaming experience. The implementation showcased effective handling of user input, game logic, and GUI components. Moving forward, potential enhancements could include additional features such as a score tracker, sound effects, or an

AI opponent for single-player mode. Overall, the project successfully achieved its goals, offering a functional and visually appealing rendition of the classic Tic-Tac-Toe game.

5. References

https://www.javatpoint.com/