TIC-TAC-TOE

COURSE PROJECT REPORT

OF

CST 201 - DATA STRUCTURES Submitted by

Athul Chandran A K - ASI23CA021
Athul M B - ASI23CA022
Atul Biju - ASI23CA023
Avinash V Bhaskaran - ASI23CA024
Ayisha Shana Perumballi - ASI23CA025
Under the guidance of

Ms. MEENATCHI K V

To

APJ Abdul Kalam Technological University
In partial fulfillment of the requirements for the award of the
Degree of
Bachelor of Technology in Computer Science and Engineering
(Artificial Intelligence)

Department of Computer and Engineering Science (Artificial Intelligence)



ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY NOVEMBER 2024

TIC-TAC-TOE

COURSE PROJECT REPORT
OF
CST 201 - DATA STRUCTURES
Submitted by

Athul Chandran A K - ASI23CA021 Athul M B - ASI23CA022 Atul Biju - ASI23CA023 Avinash V Bhaskaran - ASI23CA024 Ayisha Shana Perumballi - ASI23CA025

Under the guidance of

Ms. MEENATCHI K V

To

APJ Abdul Kalam Technological University
In partial fulfillment of the requirements for the award of the
Degree of
Bachelor of Technology in Computer Science and Engineering
(Artificial Intelligence)



Department of Computer Science and Engineering (Artificial Intelligence)

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY

NOVEMBER 2024

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)



CERTIFICATE

Certified that this is a bonafide record of the course project entitled

TIC-TAC-TOE

Submitted by

Athul Chandran A K - ASI23CA021 Athul M B - ASI23CA022 Atul Biju - ASI23CA023 Avinash V Bhaskaran - ASI23CA024 Ayisha Shana Perumballi - ASI23CA025

during the year 2024-25 in partial fulfillment of the requirement for
the award of the degree of
Bachelor of Technology in Computer Science (Artificial
Intelligence) and Engineering

Faculty in Charge

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this course project. This endeavor was a collaborative effort, and we are sincerely thankful for the support, guidance, and assistance provided by various individuals and resources.

First and foremost, we extend our appreciation to **Prof. Meenatchi K. V.**, whose expertise and unwavering support played a pivotal role in shaping our project. Their valuable insights and constructive feedback were instrumental in refining our ideas and enhancing the overall quality of our work.

We would also like to thank our fellow classmates and team members for their dedication and hard work throughout the project. Each member brought a unique set of skills and perspectives, contributing to the synergy that defined our group.

Furthermore, we are grateful to Adi Shankara Institute Of Engineering And Technology, Kalady Department Of Computer Science And Engineering (Artificial Intelligence) for providing us with the necessary resources and a conducive learning environment. The facilities and access to research materials greatly facilitated our project development. In conclusion, this course project would not have been possible without the collective effort and support of everyone involved.

VISION AND MISSION OF THE DEPARTMENT

VISION

To be in the frontier of AI technology through quality of education, collaborative research and produce globally competitive, industry ready engineers with social commitment.

MISSION

- M1: Achieve excellence in the educational experience, fostering collaborative research through state-of-the-art infrastructure and innovative elements.
- **M2:** Establish industry collaboration to address interdisciplinary challenges across diverse applications.
- **M3:** Inspire students to develop into ethical, Innovative and entrepreneurial leaders through a socially-centered program

ABSTRACT

In the realm of interactive gaming, simplicity and strategic depth make classic games like Tic Tac Toe enduringly popular. The Tic Tac Toe project seeks to recreate this timeless game in a digital format using foundational programming techniques, designed to provide an engaging and intuitive experience for users. This program utilizes a 2D array structure to manage and display the game board, facilitating easy updates to reflect players' moves and ensuring smooth gameplay.

The game begins with an automatic setup of the board, displaying numbered positions that help players easily choose where to place their mark. Players are prompted to take turns selecting their desired position, with input validation in place to prevent invalid moves. The program dynamically updates the board after each move and evaluates the game state to check for a win, loss, or draw condition, allowing players to enjoy a fair and uninterrupted gaming experience.

To enhance usability, the system also offers an option to reset the board for a new game after each round, enabling repeated gameplay within a single session. This project not only showcases core programming concepts such as array manipulation and conditional logic but also emphasizes the value of user-friendly design in creating a seamless and enjoyable gaming interface. Ultimately, this Tic Tac Toe implementation serves as an entertaining tool for players, while also highlighting the elegance and efficiency of algorithmic approaches in classic game development.

CONTENTS

Chapter	Title	Page No
1	Introduction	
2	Program and Design	
3	Flowchart and Program Logic	
4	Program Code	
5	Testing and Validation	
6	Conclusion	
7	References	

CHAPTER 1 INTRODUCTION

The Tic Tac Toe Game is an interactive console-based application developed in C, designed to bring the classic two-player game to life on the screen. This project utilizes a 2D array data structure to represent the game board, offering a straightforward and efficient setup for players to engage in competitive play. Each cell on the board can hold one of the following states:

- Empty: A position available for marking.
- X or O: Symbols representing each player's moves.

Key Features of the Program:

1. Dynamic Game Board:

♣ A 3x3 array is used to track player moves, making it easy to update and reset the board. Each player's move is validated to ensure fair play and prevent overwriting previous moves.

2. Turn-Based User Interaction:

4 The program prompts each player to select their move by entering a position on the board. Input validation and guidance messages enhance the user experience, helping players follow game rules while making moves.

3. Win and Draw Detection:

♣ After each move, the system checks the game board for a winning combination. The program evaluates all rows, columns, and diagonals to determine if a player has won or if the game ends in a draw, providing immediate feedback on game outcomes.

4. Replay Option:

4 At the end of each game, users are offered an option to reset the board and play again. This feature supports extended gameplay in a single session, giving users a seamless experience without needing to restart the program.

5. Memory Efficiency:

♣ By utilizing stack-based memory allocation, the program maintains simplicity and efficiency in resource usage, ideal for lightweight applications like this interactive game.

Through these features, the Tic Tac Toe Game provides an engaging and user-friendly interface that makes it easy for players to enjoy the classic game in a digital format. The project showcases fundamental programming concepts in C, including array manipulation, conditional logic, and interactive console design, making it both an entertaining tool and an educational demonstration of structured programming in C.

CHAPTER 2 PROGRAM DESIGN

1. Overview

The Tic Tac Toe Game provides a console-based, two-player game experience where players take turns marking spaces in a 3x3 grid. The program uses a 2D array to store and manage game moves, providing an interactive, user-friendly approach to gameplay.

2. Data Structure

Game Board Structure:

- ♣ A 3x3 character array (`char board[3][3]`) represents the game board.
- **♣** Each cell in the array holds either:
- ♣ A number from 1-9 (for unoccupied positions)
- **↓** 'X' or 'O' (to represent the players' moves)

char board[3][3];

3. Function Descriptions

resetBoard:

- o Purpose: Initializes or resets the game board with numbers 1-9.
- o Parameters: The 3x3 `board` array.
- o Returns: None (void).

printBoard:

- o Purpose: Displays the current state of the game board.
- o Parameters: The 3x3 `board` array.
- o Returns: None (void).

checkWin:

- Purpose: Evaluates the board for a win condition or draw.
- o Parameters: The 3x3 `board` array.
- o Returns: Integer (1 for win, 0 for ongoing game, -1 for draw).

takeTurn:

- o Purpose: Manages player input, validates choices, and places player's mark.
- o Parameters: Player number, choice of position, and 3x3 `board` array.
- Returns: None (void).

4. Program Flow

1. Initialization:

♣ Initialize game variables, including the game board and a `player` variable to keep track of turns.

2. Reset Board:

♣ Call `resetBoard` to set up the game board with numbers from 1-9.

3. Game Loop:

- **4** Turn Management:
 - Alternate turns between Player 1 (X) and Player 2 (O).
- **♣** Input Handling:
 - Prompt players to select a position by entering a number from 1 to 9.
 - Validate input to ensure the chosen position is available.
- **♣** Update and Display Board:
 - Update the board based on player input and call `printBoard` to display the current game state.

4. Check Game State:

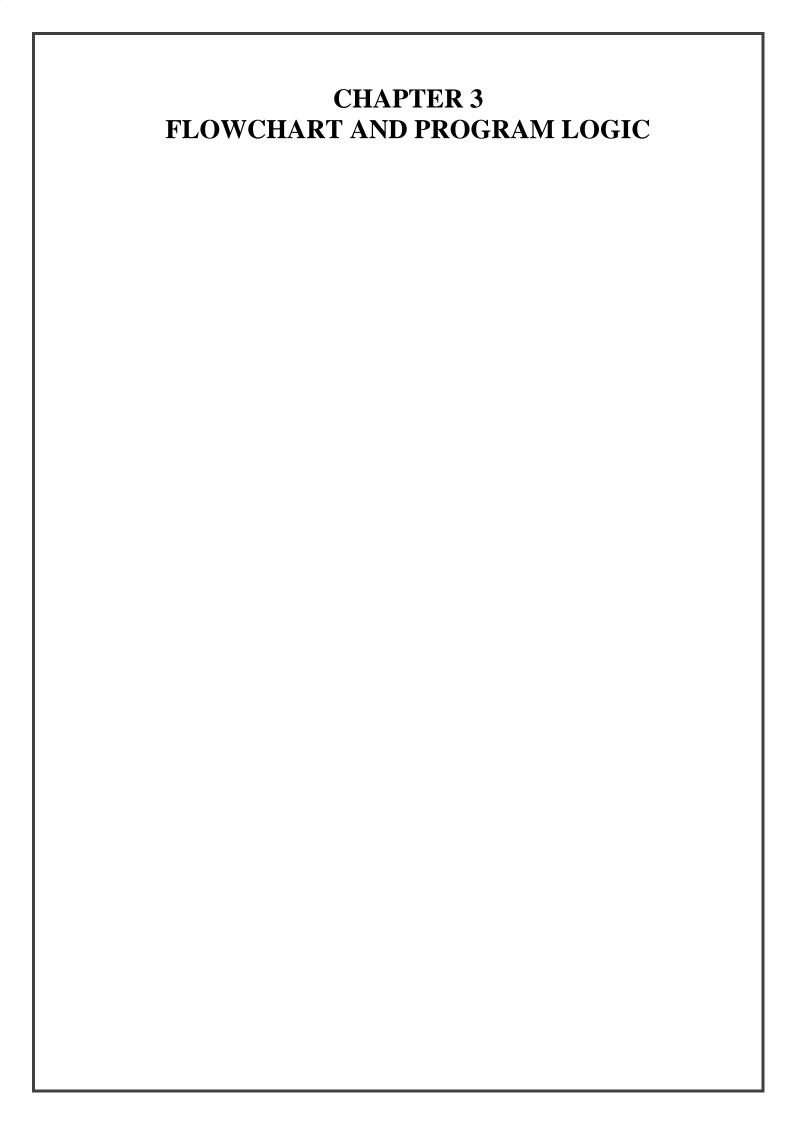
♣ After each move, call `checkWin` to evaluate if a player has won, if the game is a draw, or if it should continue.

5. Replay Option:

- ♣ Prompt players to start a new game or exit.
- ♣ If players choose to replay, reset the game board with `resetBoard` and reinitialize necessary variables.

6. End Program:

♣ Terminate the program gracefully, displaying a closing message for the players.

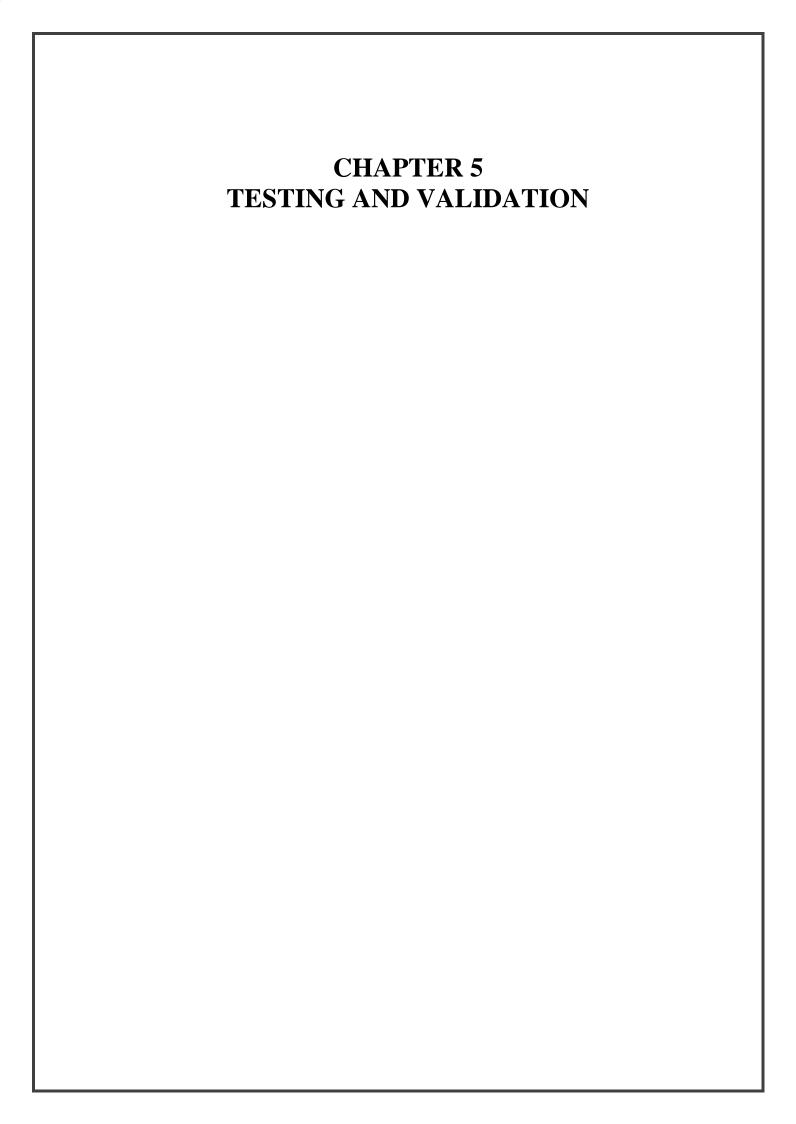


CHAPTER 4 PROGRAM CODE

```
#include <stdio.h>
void printBoard(char board[3][3]);
int checkWin(char board[3][3]);
void resetBoard(char board[3][3]);
int main() {
    char board[3][3];
    int playAgain = 1;
    while (playAgain) {
        resetBoard (board);
        printf("The board has been reset for a new game!\n"); // New game message
        int player = 1, choice;
        char mark;
        int result = 0; // 0: game continues, 1: win, -1: draw
        int moves = 0;
        while (result == 0) {
            printBoard(board);
            player = (player % 2) ? 1 : 2;
            mark = (player == 1) ? 'X' : 'O';
            printf("Player %d, enter the number of the square to place your %c: ",
player, mark);
            scanf("%d", &choice);
            if (choice < 1 || choice > 9) {
                printf("Invalid choice! Please choose a number between 1 and
9.\n'');
                continue;
            }
            int row = (choice - 1) / 3;
            int col = (choice - 1) % 3;
            if (board[row][col] != 'X' && board[row][col] != 'O') {
                board[row][col] = mark;
                moves++;
            } else {
                printf("That position is already taken! Try again.\n");
                continue;
            result = checkWin(board);
            if (result == 0 && moves == 9) {
                result = -1; // Draw
            }
            player++;
```

```
printBoard(board);
        if (result == 1) {
            printf("==> Player %d wins! Congratulations!\n", --player);
        } else {
            printf("==> It's a draw!\n");
        // Ask if players want to play again
        do {
            printf("Do you want to play again? (1 for Yes, 0 for No): ");
            scanf("%d", &playAgain);
            if (playAgain != 1 && playAgain != 0) {
                printf("Invalid input! Please enter 1 for Yes or 0 for No.\n");
        } while (playAgain != 1 && playAgain != 0);
        if (playAgain) {
            printf("\nStarting a new game...\n");
        } else {
            printf("Thank you for playing! Goodbye!\n"); // Revised exit message
    }
    return 0;
}
void printBoard(char board[3][3]) {
   printf("Current Board:\n");
   printf("\t\t %c | %c | %c \n", board[0][0], board[0][1], board[0][2]);
   printf("\t\t---|---\n");
    printf("\t \c | \c \n", board[1][0], board[1][1], board[1][2]);
    printf("\t\t---|---\n");
   printf("\t\t %c | %c | %c \n\n", board[2][0], board[2][1], board[2][2]);
}
int checkWin(char board[3][3]) {
    // Check rows and columns
    for (int i = 0; i < 3; i++) {
        if (board[i][0] == board[i][1] && board[i][1] == board[i][2]) {
            return 1; // Win
        if (board[0][i] == board[1][i] && board[1][i] == board[2][i]) {
            return 1; // Win
        }
    }
    // Check diagonals
    if (board[0][0] == board[1][1] && board[1][1] == board[2][2]) {
        return 1; // Win
    if (board[0][2] == board[1][1] && board[1][1] == board[2][0]) {
        return 1; // Win
    return 0; // No win yet
}
void resetBoard(char board[3][3]) {
```

```
for (int i = 0; i < 3; i++) {
   for (int j = 0; j < 3; j++) {
      board[i][j] = '1' + (i * 3 + j); // Reset to numbers 1-9</pre>
      }
}
```



CHAPTER 6 CONCLUSION

In conclusion, the **Tic Tac Toe Game** project showcases the effective use of a 2D array to manage game state and facilitate player interaction in a classic two-player game. The program has undergone extensive testing, ensuring functionality, input validation, and memory management are robust and reliable. Through this project, key programming concepts such as arrays, control structures, and user input handling have been explored and implemented effectively. Overall, this project serves as a practical demonstration of foundational programming skills in C, providing a fun and engaging experience for users while enhancing their understanding of software development principles.

CHAPTER 7 REFERENCES

1. Knuth Donald. "The Art of Computer Programming."

2. Skiena Steven. "The Algorithm Design Manual."

3. Rosen Kenneth H. "Discrete Mathematics and Its Applications."

4. Sedgewick Robert. "Algorithms", 4th Edition.

5. King K. N. "C Programming: A Modern Approach."

6. Plauger P. J. "The Standard C Library."

7. Kelley Al and Ira Pohl. "A Book on C."