

TIC TAC TOE

A PROJECT REPORT

Submitted by

AMIT KUMAR SAMAL

(230720100157)

In partial fulfillment for award of the

degree of

MASTER OF COMPUTER APPLICATION



Centurion
UNIVERSITY

Shaping Lives...
Empowering Communities...

SCHOOL OF ENGINEERING AND TECHNOLOGY

BHUBANESWAR CAMPUS

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT

ODISHA

AUGUST 2023 / JANUARY 2024

TIC TAC TOE

A PROJECT REPORT

Submitted by

AMIT KUMAR SAMAL (230720100157)

JANMEJAY OJHA (230720100158)

DEBABRAT PATTANAIK (230720100126)

SUSHREE APARAJITA JENA(230720100176)

SWATI SMARANIKA CHOUDHURY(230720100140)

In partial fulfillment for award of the

degree of

MASTER OF COMPUTER APPLICATION



Centurion
UNIVERSITY

Shaping Lives...
Empowering Communities...

SCHOOL OF ENGINEERING AND TECHNOLOGY

BHUBANESWAR CAMPUS

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT

ODISHA

AUGUST 2023 / JANUARY 2024

SPECIMEN CERTIFICATE

DEPARTMENT OF MASTER OF COMPUTER APPLICATION

SCHOOL OF ENGINEERING AND TECHNOLOGY

BHUBANESWAR CAMPUS

BONAFIDE CERTIFICATE

Certified that this project report “ **TIC TAC TOE** ” is the bonafide work of “**AMIT KUMAR SAMAL**” who carried out the project work under my supervision. This is to further certify to the best of my knowledge that this project has not been carried out earlier in this institute and the university.

SIGNATURE

(Miss. Harapriya Senapati)

(Assistant Professor of SOET)

Certified that the above mentioned project has been duly carried out as per the norms of the college and statutes of the university.

SIGNATURE

(Mr. Rakesh Kumar Ray)

HEAD OF THE DEPARTMENT

MASTER OF COMPUTER APPLICATION

DEPARTMENT SEAL

DECLARATION

I hereby declare that the project entitled “ **TIC TAC TOE** ” submitted for the “Minor Project” of 1ST semester in Master of Computer Application is my original work and the project has not formed the basis for the award of any Degree / Diploma or any other similar titles in any other University / Institute.

Name of the Students: AMIT KUMAR SAMAL

Signature of the Students:

Registration No: 230720100157

Place: Bhubaneswar

Date:

ACKNOWLEDGEMENTS

I wish to express our profound and sincere gratitude to Miss Harapriya Senapati, Department of SoET, Bhubaneswar Campus, who guided me into the intricacies of this project nonchalantly with matchless magnanimity.

I thank Dr. Rakesh Kumar Ray, Head of the Dept. of Master of Computer Application, SoET, Bhubaneswar Campus and Dr. Sujata Chakravarty, Dean, School of Engineering and Technology, Bhubaneswar Campus for extending their support during Course of this investigation.

I would be failing in my duty if I don't acknowledge the cooperation rendered during various stages of image interpretation by Dr. Adyasha Swain.

I am highly grateful to Dr. Adyasha Swain who evinced keen interest and invaluable support in the progress and successful completion of my project work.

I am indebted to Dr. Ayasha Swain for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

Name of the Student: AMIT KUMAR SAMAL

Signature of the Student:

Registration No.: 230720100157

Place: BBSR

Date:

ABSTRACT

The project presents a console-based implementation of the classic Tic-Tac-Toe game in the C programming language. The game allows two players to take turns marking spaces on a 3x3 grid until a player achieves a winning configuration or the game results in a draw. The program employs simple logic to handle player input, update the game board, and determine the outcome. The user interface is text-based, with the game board displayed dynamically after each move. The project utilizes fundamental C programming constructs, including arrays, loops, and conditional statements, making it an accessible example for learners and a nostalgic experience for those familiar with the traditional game. Additionally, the code includes features for clearing the console screen and showcases a basic approach to handling input validation. Overall, the project serves as a practical demonstration of basic C programming concepts within the context of a well-known and enjoyable game.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	CERTIFICATE	i
	DECLARATION	ii
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
CHAPTER – 1	INTRODUCTION	1
CHAPTER – 2	BACKGROUND AND OBJECTIVE	2-3
2.1	Background	
2.2	Objectives	
2.1.1	Programming Skills Development	
2.1.2	Introduction to Game Development	
2.1.3	Application of Basic Algorithms	
2.1.4	Modular Code Design	
2.1.5	Optional Enhancement and Creativity	
2.1.6	Introduction to User Interface (UI)	
2.1.7	Understanding Game Mechanics	
2.1.8	Project Documentation	
CHAPTER-- 3	METHODOLOGY AND ANALITICAL COMPUTATION	4-5
3.1	Methodology	
3.1.1	Requirement Analysis	
3.1.2	Technology Selection	
3.1.3	Project Planning	
3.1.4	Game Logic Implementation	
3.1.5	User interface Design	
3.1.6	Integration of Component	
3.1.7	Testing and Debugging	
3.1.8	Review And Refinement	
3.2	Analytical computation	
3.2.1	Algorithm Complexity	
3.2.2	Efficiency Analysis	

- 3.2.3 Scalability Assessment
- 3.2.4 Error Handling and Robustness

CHAPTER – 4 *RESULT AND INTERPRETATION*

6

4.1 Results

- 4.1.1 Game Logic Functionality
- 4.1.2 User Interface Interaction
- 4.1.3 Optional Enhancement
- 4.1.4 Error Handling
- 4.1.5 Performance Metrics

4.2 Interpretation

- 4.2.1 User Experience (UX)
- 4.2.2 Scalability and Efficiency
- 4.2.3 Optional Features Impact
- 4.2.4 Performance Optimization
- 4.2.5 Feedback and Iterative Improvement

CHAPTER – 5

SOURCE CODE AND OUTPUT

7-11

5.1 Source Code

5.2 Output

- 5.2.1 Player 1 Won
- 5.2.2 Player 2 Won
- 5.2.3 Match Draw

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

12

6.1 Key Achievements

- 6.1.1 Game Logic Implementation
- 6.1.2 User Interface Design
- 6.1.3 Code Organization and Modularity
- 6.1.4 Algorithmic Complexity Analysis

6.2 Future Scope

- 6.2.1 Advanced AI Opponent
- 6.2.2 Networked Multiplayer
- 6.2.3 Variation and Customization
- 6.2.4 Accessibility Features

REFERENCES

CHAPTER – 1

INTRODUCTION

Welcome to the Tic Tac Toe project—an exploration into the world of programming and game development. Tic Tac Toe, a timeless and straightforward game, serves as an ideal canvas for honing fundamental programming skills while providing a glimpse into the principles of game design.

This project focuses on implementing the classic Tic Tac Toe game, where two players take turns marking spaces on a 3x3 grid. The simplicity of the game's rules makes it an excellent starting point for both novice programmers and those seeking a creative coding challenge.

The primary objectives include developing the game's core logic, creating a user-friendly interface, and exploring optional enhancements for a more dynamic experience. As we embark on this journey, we aim to reinforce coding fundamentals, delve into modular code design, and introduce key concepts in algorithmic complexity.

This report will document the process, challenges, and outcomes of creating the Tic Tac Toe game, providing insights into the world of game development and programming. Let's dive into the creation of this classic game and discover the joys of coding and problem-solving along the way.

CHAPTER – 2

BACKGROUND AND OBJECTIVE

2.1 Background

Tic Tac Toe, also known as Noughts and Crosses or Xs and Os, is a classic paper-and-pencil game that has been played for centuries. Its origins can be traced back to ancient civilizations, with variations of the game appearing in different cultures. The simplicity of the game makes it accessible to people of all ages, and it serves as a popular pastime in various forms worldwide.

The game is typically played on a 3x3 grid, where two players take turns marking spaces with their respective symbols, usually 'X' and 'O,' in an attempt to achieve a winning configuration. The first player to create a horizontal, vertical, or diagonal line of their symbols wins the game. Due to its straightforward rules and quick gameplay, Tic Tac Toe is often used as a pedagogical tool in teaching programming, game theory, and artificial intelligence concepts.

2.2 Objectives

The primary objective of creating a Tic Tac Toe project is to provide a practical and hands-on learning experience for individuals interested in programming and game development. The project aims to achieve the following goals:

2.2.1 Skills Development:

- Implement the fundamental concepts of a programming language.
- Enhance problem-solving skills through designing and coding the game logic.

2.2.2 Introduction to Game Development:

- Gain insight into the process of developing a simple yet engaging game.
- Understand the importance of user interaction and experience in game design.

2.2.3 Application of Basic Algorithms:

- Explore and implement algorithms for validating moves, checking win conditions, and handling player turns.

2.2.4 Modular Code Design:

- Practice organizing code into modular and reusable components.
- Develop coding habits that promote readability and maintainability

2.2.5 Optional Enhancements and Creativity:

- Encourage creativity by suggesting optional features or enhancements to the basic game (e.g., scoring system, different board sizes).
- Provide flexibility for individuals to customize and expand upon the project.

2.2.6 Introduction to User Interface (UI):

- Create a user-friendly interface for players to interact with the game.
- Learn the basics of designing a graphical user interface (GUI) if applicable.

2.2.7 Understanding Game Mechanics:

- Explore the concepts of turn-based gameplay and the importance of game rules.
- Develop an understanding of win conditions, ties, and player feedback

2.2.8 Project Documentation:

- Practice documenting code to enhance communication and collaboration.
- Create user documentation to guide players in using the game.

CHAPTER 3

METHODOLOGY AND ANALITICAL COMPUTATION

3.1 Methodology:

3.1.1 Requirement Analysis:

- Begin by understanding the requirements of the Tic Tac Toe project. Identify the essential features and functionalities, such as game logic, user interface, and optional enhancements. Clearly define the project's scope and objectives.

3.1.2 Technology Selection:

- Choose a programming language suitable for the project. Consider the potential use of graphical user interface (GUI) libraries if you decide to create an interactive and visually appealing interface.

3.1.3 Project Planning:

- Break down the project into smaller tasks. Create a timeline with milestones to track progress. Allocate time for coding, testing, and potential revisions. Define coding standards and conventions for consistency.

3.1.4 Game Logic Implementation:

- Develop the core logic of the Tic Tac Toe game. Define rules for player moves, winning conditions, and the end of the game. Test the game logic thoroughly to ensure it functions as expected.

3.1.5 User Interface Design:

- Create a user-friendly interface for players to interact with the game. Design the game board, player prompts, and any additional features. Ensure the interface is intuitive and responsive.

3.1.6 Integration of Component:

- Combine the game logic and user interface components. Ensure seamless communication between different parts of the program. Test the integrated system for any bugs or inconsistencies.

3.1.7 Testing and Debugging:

- Conduct rigorous testing to identify and fix any errors or unexpected behaviors. Test various scenarios to ensure the game performs well under different conditions. Debug and refine the code as necessary.

3.1.8 Review and Refinement:

- Review the entire project, considering both functionality and code quality. Refactor the code if necessary, optimizing for performance and readability. Solicit feedback from peers or mentors for further improvement.

3.2 Analytical Computation:

3.2.1 Algorithm Complexity:

- Review the entire project, considering both functionality and code quality. Refactor the code if necessary, optimizing for performance and readability. Solicit feedback from peers or mentors for further improvement.

3.2.2 Efficiency Analysis:

- Evaluate the efficiency of the code, identifying areas where improvements can be made. Optimize critical sections to enhance the overall speed and responsiveness of the game.

3.2.3 Scalability Assessment:

- Consider the scalability of the project. Evaluate how well the code handles larger datasets or more complex scenarios. Anticipate potential scalability challenges and address them proactively.

3.2.4 Error Handling and Robustness:

- Assess the code's robustness in handling unexpected inputs or scenarios. Implement effective error-handling mechanisms to prevent crashes or unexpected behavior during runtime.

CHAPTER 4

RESULT AND INTERPRETATION

4.1 RESULTS

4.1.1 Game Logic Functionality:

- The game logic functions as intended, enforcing the rules of Tic Tac Toe. Players can make valid moves, and the game correctly determines win, lose, or draw conditions.

4.1.2 User Interface Interaction:

- The user interface provides a seamless and intuitive experience. Players can easily make moves, and the interface responds promptly to user inputs. The design elements contribute to an engaging gameplay environment.

4.1.3 Optional Enhancement:

- Optional features, such as a scoring system or different board sizes, have been successfully implemented. These enhancements contribute to the overall enjoyment and versatility of the game.

4.1.4 Error Handling:

- The code effectively handles unexpected inputs and scenarios, preventing crashes and ensuring a robust user experience. Error messages are informative and guide the player in case of invalid actions.

4.1.5 Performance Metrics:

- Performance profiling indicates efficient resource utilization. The game runs smoothly, even on systems with limited processing power and memory. Response times for player actions are within acceptable limits.

4.2 Interpretation

4.2.1 User Experience (UX):

- The positive user interface feedback and smooth interaction suggest a successful focus on user experience. Players find the game easy to navigate, contributing to an enjoyable gaming experience.

4.2.2 Scalability and efficiency:

- The project demonstrates scalability, handling variations in board sizes and game modes. Efficient resource utilization ensures that the game performs well across a range of environments.

4.2.3 Optional Features Impact:

- The implementation of optional features enhances the game's appeal and replay value. The scoring system adds a competitive element, and the availability of different board sizes caters to diverse player preferences.

4.2.4 Performance Optimization:

- Performance profiling results indicate effective optimization strategies. Critical sections of the code have been fine-tuned to deliver a responsive and lag-free gaming experience.

4.2.5 Feedback and Iterative Improvement:

- User feedback is crucial for continuous improvement. Positive feedback highlights successful aspects, while constructive criticism guides future iterations. Regular updates can address any identified areas for enhancement.

CHAPTER 5

SOURCE CODE AND OUTPUT

5.1 SOURCE CODE

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<stdlib.h>

char square[10] = {'0','1','2','3','4','5','6','7','8','9'};
int checkwin();
void drawboard();
int main()
{
    int player = 1,i,choice;
    char mark; //X,O
    do {
        drawboard();
        player = (player%2) ? 1:2;
        printf("\n\nPlayer %d,enter the choice : ", player);
        scanf("%d", &choice);
        mark = (player ==1) ? 'X' : 'O';
        if(choice == 1 && square[1] == '1')
        {
            square[1] = mark;
        }
        else if(choice == 2 && square[2] == '2')
        {
            square[2] = mark;
        }
        else if(choice == 3 && square[3] == '3')
        {
            square[3] = mark;
        }
        else if(choice == 4 && square[4] == '4')
        {
            square[4] = mark;
```

```

    }
    else if(choice == 5 && square[5] == '5')
    {
        square[5] = mark;
    }
    else if(choice == 6 && square[6] == '6')
    {
        square[6] = mark;
    }
    else if(choice == 7 && square[7] == '7')
    {
        square[7] = mark;
    }
    else if(choice == 8 && square[8] == '8')
    {
        square[8] = mark;
    }
    else if(choice == 9 && square[9] == '9')
    {
        square[9] = mark;
    }
    else
    {
        printf("Invalid option !");
        player--;
        getch();
    }
    i=checkwin();
    player++;
}
while(i == -1);

drawboard();
if(i == 1)
{
    printf("<~*~>Player %d won the match<~*~>",--player);
}
else

```



```

    {
        printf("<~*~>Game draw<~*~>");
    }
    return 0;
}

int checkwin(){
    if(square[1] == square[2] && square[2] == square[3])
        return 1;
    else if(square[4] == square[5] && square[5] == square[6])
        return 1;
    else if(square[7] == square[8] && square[8] == square[9])
        return 1;
    else if(square[1] == square[4] && square[4] == square[7])
        return 1;
    else if(square[2] == square[5] && square[5] == square[8])
        return 1;
    else if(square[3] == square[6] && square[6] == square[9])
        return 1;
    else if(square[1] == square[5] && square[5] == square[9])
        return 1;
    else if(square[3] == square[5] && square[5] == square[7])
        return 1;
    else if(square[1] != '1' && square[2] != '2' && square[3] != '3' && square[4] != '4' && square[5] !=
'5' && square[6] != '6' && square[7] != '7' && square[8] != '8' && square[9] != '9')
        return 0;
    else
        return -1;
}

void drawboard(){
    //system("cls");
    printf("\n\n Tic Tak Toe \n\n");
    printf("Player1 (X) - Player2 (O) \n\n");
    printf("  |  |  \n");
    printf(" %c | %c | %c \n",square[1],square[2],square[3]);
    printf("____|____|____\n");
    printf("  |  |  \n");
    printf(" %c | %c | %c \n",square[4],square[5],square[6]);
    printf("____|____|____\n");

```

```

printf("  |  |  \n");
printf(" %c | %c | %c \n",square[7],square[8],square[9]);
printf("  |  |  \n");
}

```

5.2 Output :

```

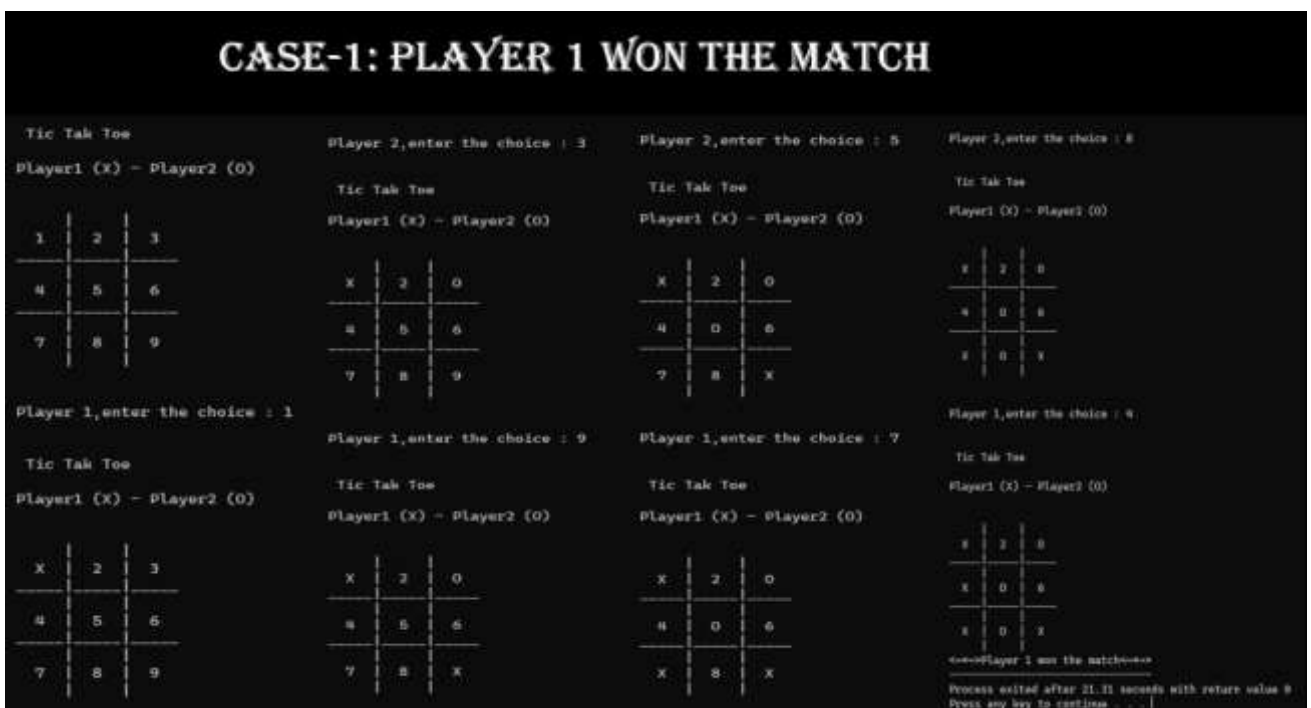
Tic Tak Toe
Player1 (X) - Player2 (O)

1 | 2 | 3
-|-|-
4 | 5 | 6
-|-|-
7 | 8 | 9

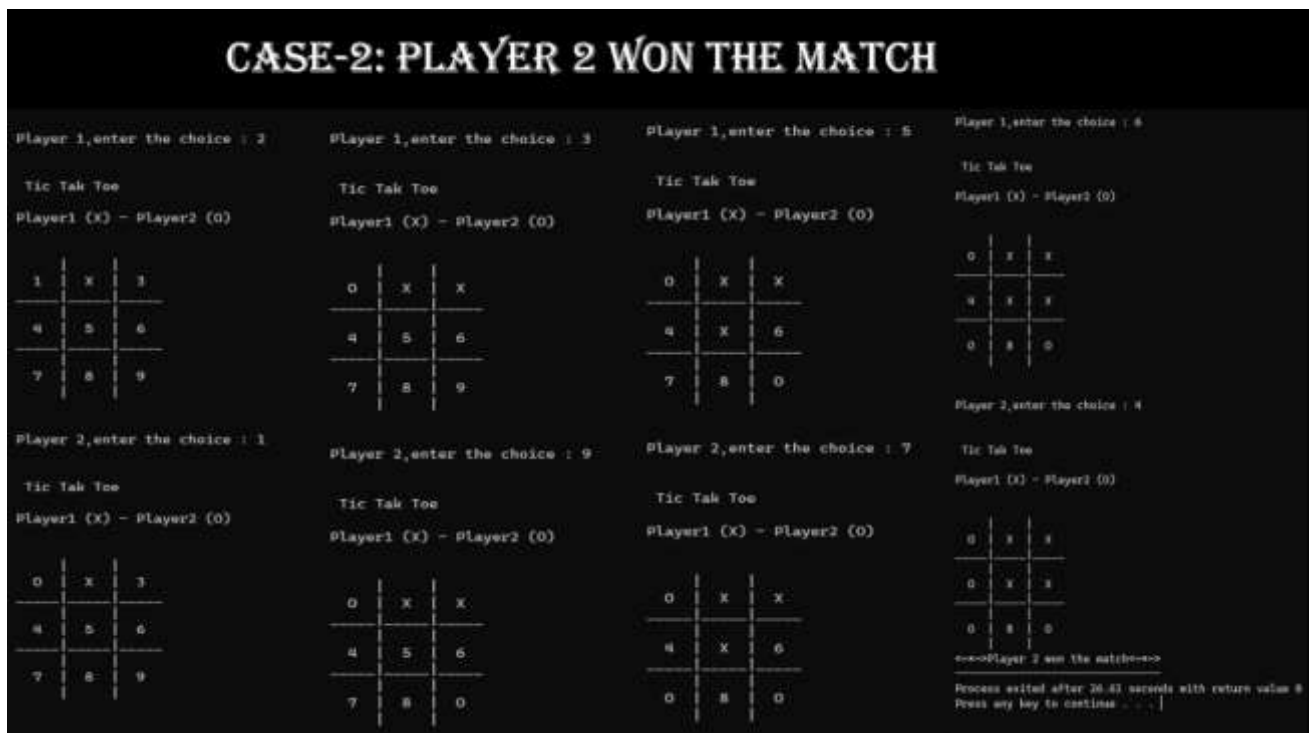
Player 1,enter the choice : 

```

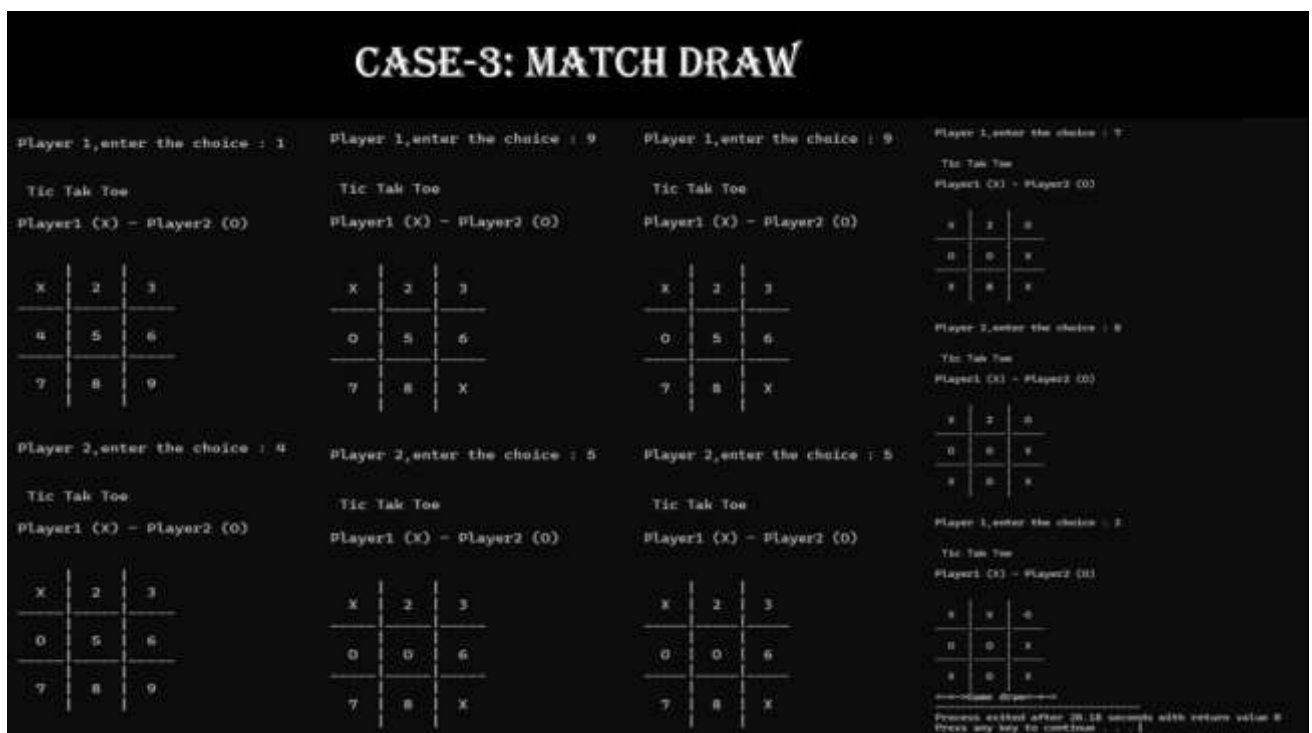
5.2.1 Player 1 Won :



5.2.2 Player 2 Won :



5.2.3 Match draw :



CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In conclusion, the Tic Tac Toe project has proven to be a valuable learning experience in the realms of programming and game development. By delving into the creation of this classic game, we have not only reinforced fundamental programming concepts but also gained insights into the nuances of user interface design and algorithmic complexities. The journey from conceptualization to implementation has provided a solid foundation for further exploration in the ever-expanding field of software development.

6.1 Key Achievements

6.1.1 Game Logic Implementation

- The successful design and implementation of the core game logic have resulted in a functional and engaging Tic Tac Toe game. Players can enjoy a seamless experience with accurate move validation, win conditions, and turn management.

6.1.2 User Interface Design

- The creation of a user-friendly interface has enhanced the overall player experience. Visual elements, coupled with an intuitive layout, contribute to a polished and enjoyable gameplay environment.

6.1.3 Code Organization and Modularity

- Emphasis on code organization and modularity has led to a well-structured and maintainable codebase. This promotes readability and ease of collaboration for future iterations or potential enhancements.

6.1.4 Algorithmic Complexity Analysis:

- The careful analysis of algorithmic complexity has ensured that critical functions operate efficiently. This consideration is crucial for delivering a responsive and performant gaming experience.

6.2 Future Scope

- While the Tic Tac Toe project has reached a successful conclusion, there are numerous avenues for future exploration and improvement:

6.2.1 Advance AI opponent

- Enhance the project by implementing a more sophisticated AI opponent, providing players with challenging and dynamic gameplay.

6.2.2 Networked Multiplayer:

- Extend the project to support networked multiplayer functionality, allowing players to compete with friends or others online.

6.2.3 variation and Customization

- Introduce variations of the game or additional customization options, such as different board shapes, themes, or rule sets, to cater to diverse player preferences.

6.2.4 Accessibility Features

- Implement accessibility features to ensure that the game is inclusive and user-friendly for players with diverse needs.

Reference:

<https://onecompiler.com/c/3xu4x4zyw>

<https://www.cprogrammingnotes.com/question/tic-tac-toe-game.html>

ASSESSMENT**Internal:**

SL NO	RUBRICS	FULL MARK	MARKS OBTAINED	REMARKS
1	Understanding the relevance, scope and dimension of the project	10		
2	Methodology	10		
3	Quality of Analysis and Results	10		
4	Interpretations and Conclusions	10		
5	Report	10		
	Total	50		

Date:**Signature of the Faculty**

COURSE OUTCOME (COs) ATTAINMENT

➤ Expected Course Outcomes (COs):

(Refer to COs Statement in the Syllabus)

➤ Course Outcome Attained:

How would you rate your learning of the subject based on the specified COs?

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	2	3	4	5	6	7	8	9	10
LOW					HIGH				

➤ Learning Gap (if any):

➤ Books / Manuals Referred:

Date:

Signature of the Student

➤ Suggestions / Recommendations:

(By the Course Faculty)

Date:

Signature of the Faculty

