TIC-TAC-TOEGAME

Project submitted to the SRM University – AP, Andhra Pradesh for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Y.TEJA SWAROOP(AP23110010282)

B.BHARADWAJ(AP23110010254)

G.MANJEESH(AP23110010309)

M.UMA MAHESH(AP23110010264)

Under the Guidance of
Dr. Kavitha rani

SRM University–AP

Neerukonda, Mangalagiri, Guntur Andhra Pradesh – 522 240 [November, 2024]

Certificate

Date: November-2024

This is to certify that the work present in this Project entitled “**TIC-TAC-TOE GAME**” has been carried out by group-10 under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in School of Engineering and Sciences.

Supervisor

Prof. / Dr.Kavitha rani.

Assistant Professor.

Department of Computer Science.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Acknowledgements

We would like to express our gratitude **Dr. Kavitha rani** for assisting us with the research at every step. Without the advice and assistance of our adviser, this study effort would not be feasible. The suggestions, feedbacks, and encouragement from our advisors were extremely helpful in assisting us in finishing this research.

Abstract

This project is a C++ implementation of the classic Tic-Tac-Toe game, designed for two players. The game features a simple console interface where players alternately input their moves to mark spaces on a 3x3 grid with X or O. The objective is to align three identical symbols either horizontally, vertically, or diagonally to achieve a win. If the grid is filled without a winner, the game ends in a draw. This project highlights fundamental C++ programming concepts, including control structures, functions, and basic input/output handling, offering an engaging way to learn and apply C++ logic.

1. Introduction

Tic-Tac-Toe is a classic two-player game that has been enjoyed by people of all ages for generations. Its simplicity and strategic depth make it an ideal project for exploring fundamental programming concepts. This report documents the process of creating a digital version of Tic-Tac-Toe, highlighting the methodologies employed, the challenges encountered, and the solutions implemented.

The primary objective of this project was to develop a functional and

user-friendly Tic-Tac-Toe game using C++. By leveraging basic programming constructs such as loops, conditionals, and functions, we aimed to create an engaging experience that mirrors the traditional pen-and-paper version.

Additionally, this project serves as a practical exercise in software development, allowing us to apply theoretical knowledge in a real-world context.

Throughout the report, we will discuss the planning and design phases, the implementation process, and the testing and evaluation of the game. We will also explore potential future enhancements that could expand the game's functionality and improve user experience. Ultimately, this project not only reinforces our programming skills but also fosters creativity and problem-solving abilities in game development.

2. Methodology

The development of the Tic-Tac-Toe game followed a structured methodology that encompassed several key phases: planning, design, implementation, testing, and evaluation.

1. Planning: The project began with defining the scope and objectives of the game. We outlined the core features, such as player turns, move validation, win detection, and the overall game flow.

3. Design : Next, we designed the game architecture. This involved creating a class structure to encapsulate the game logic, including methods for displaying

the board, making moves, checking for wins, and determining if the board is

full. We also sketched a simple text-based user interface to facilitate player

interaction.

3. Implementation: The coding phase involved translating the design into C++

code. We utilized functions to modularize the code, ensuring that each

component (e.g., move validation, win checking) was clearly defined and

easily maintainable. The game loop was implemented to manage player turns

and game state.

4. Testing :After implementation ,we conducted thorough testing to identify and fix

any bugs. This included testing various scenarios, such as winning

combinations, invalid moves, and full boards, to ensure the game behaved as

expected.

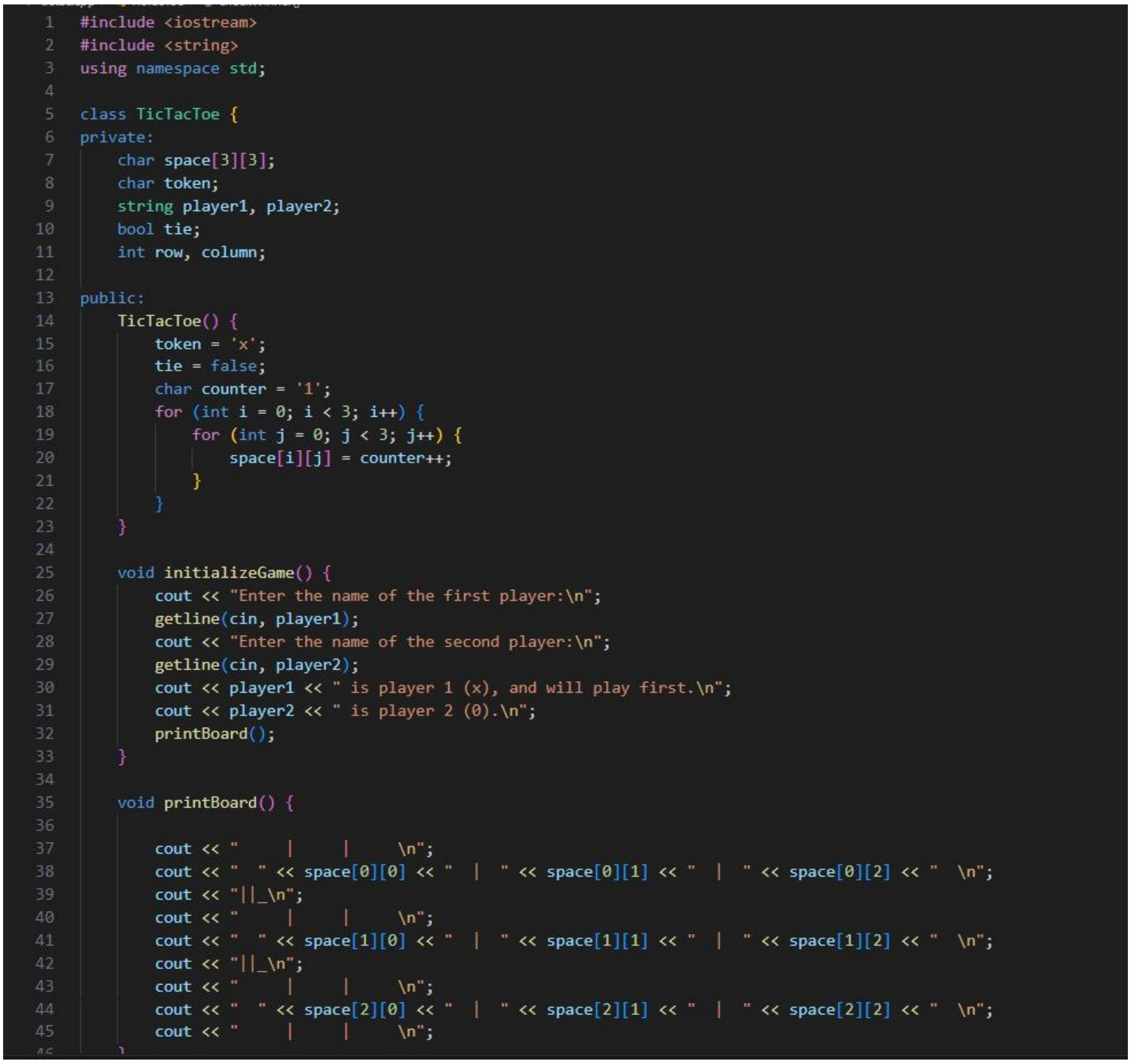
5. Evaluation: Finally, we evaluated the game based on user feedback and

performance. This involved assessing the user experience and identifying

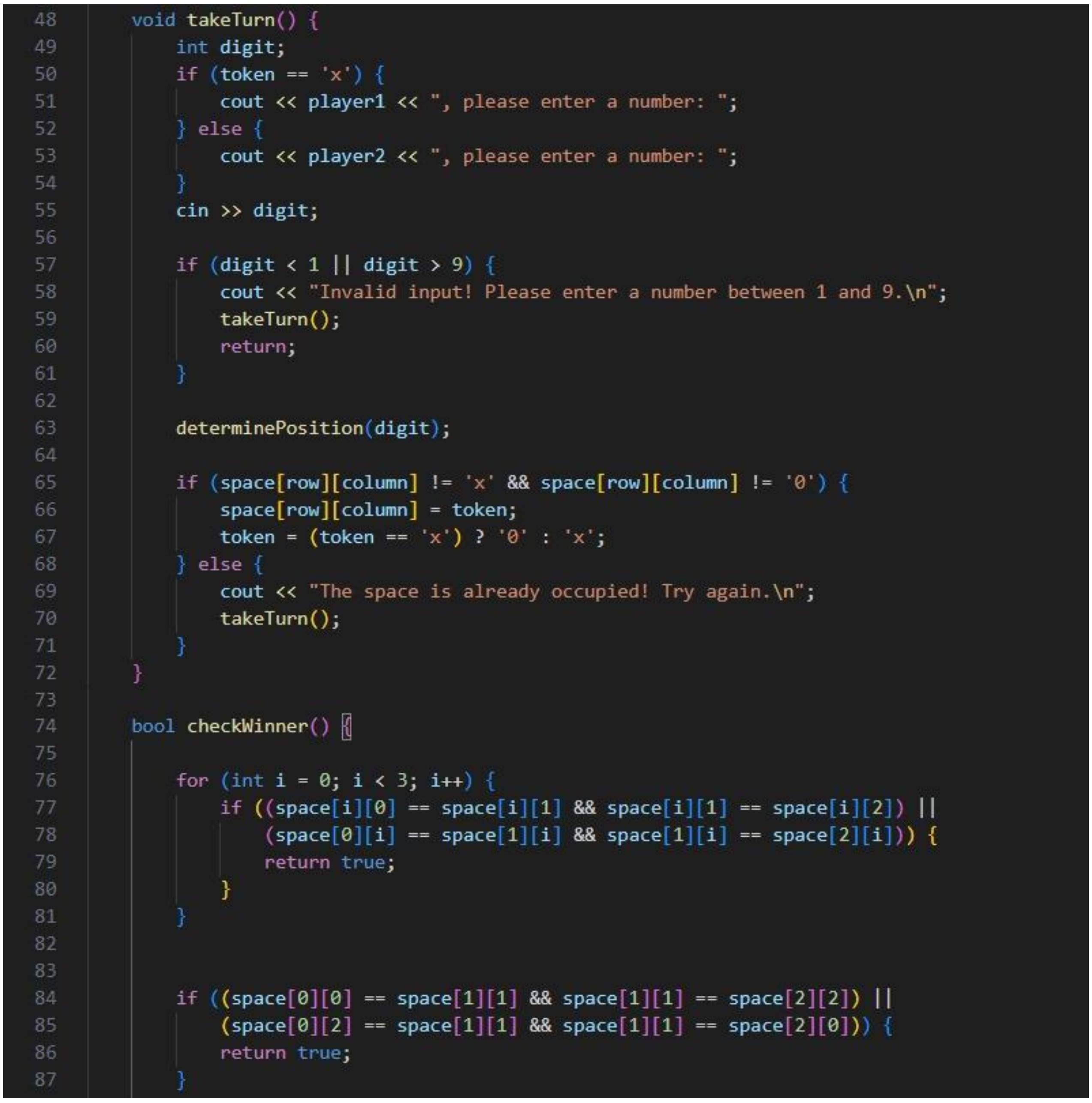
areas for improvement, which will inform future enhancements.

3. Discussion

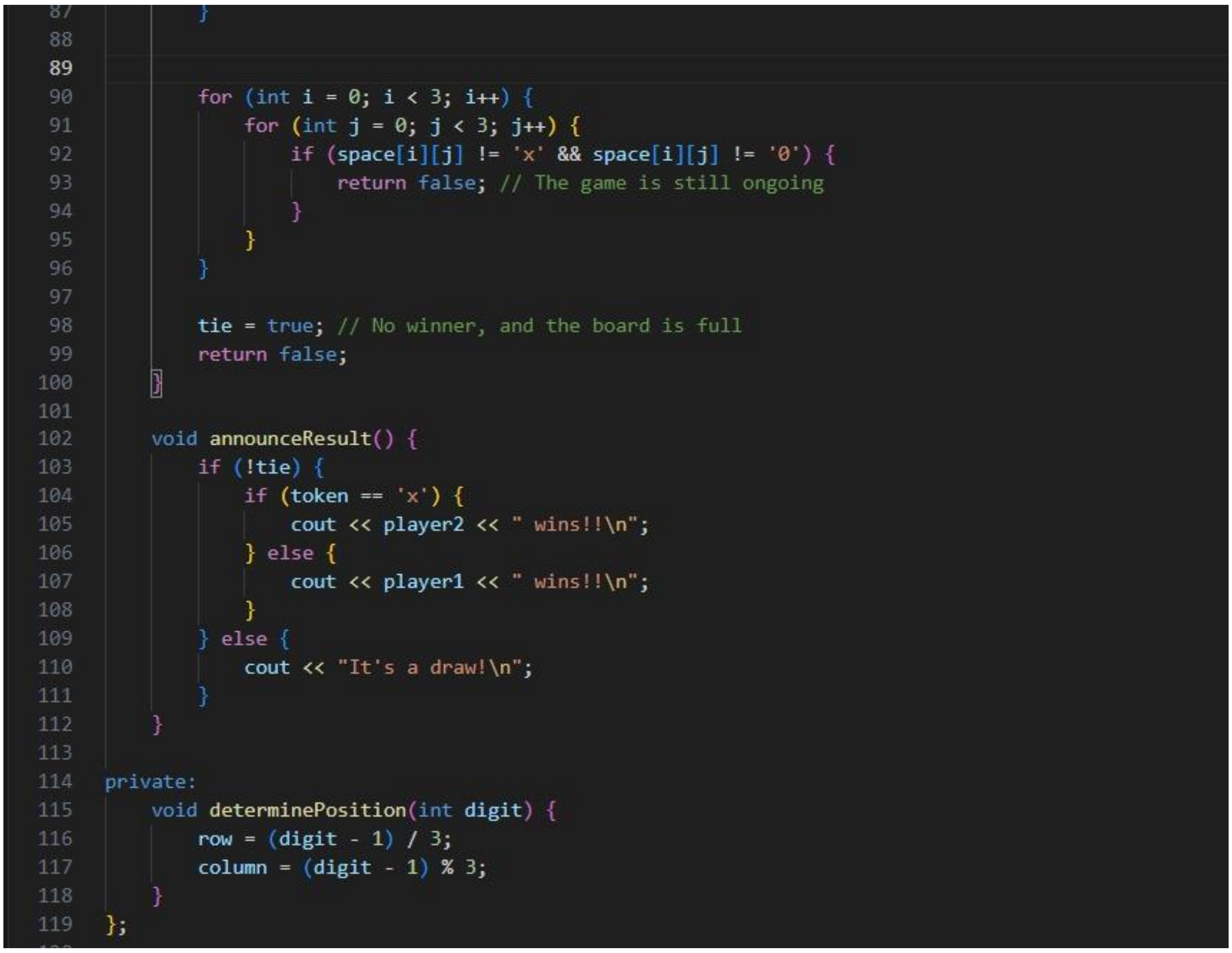
INPUT OF THE CODE:



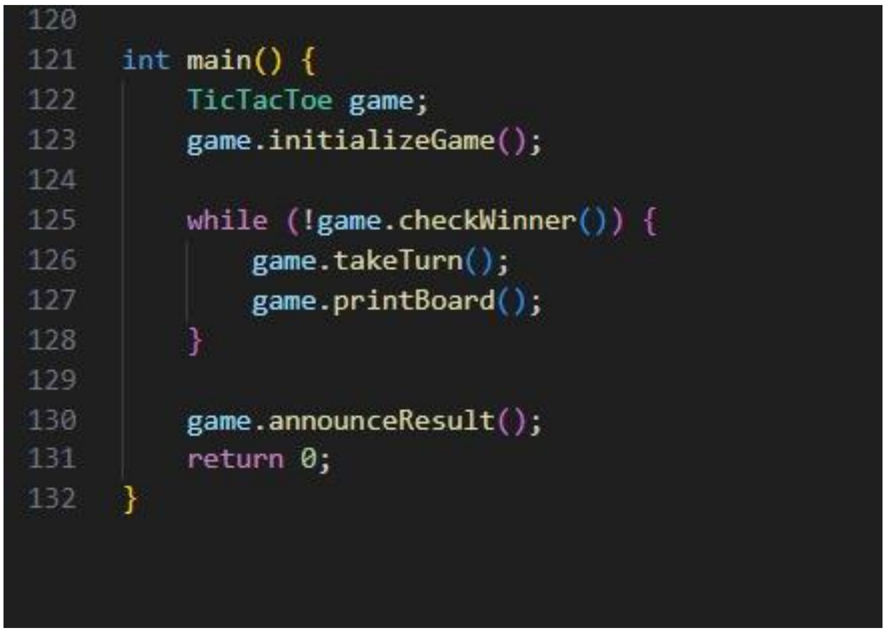
```cpp
#include <iostream>
#include <string>
using namespace std;

class TicTacToe {
private:
    char space[3][3];
    char token;
    string player1, player2;
    bool tie;
    int row, column;

public:
    TicTacToe() {
        token = 'x';
        tie = false;
        char counter = '1';
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                space[i][j] = counter++;
            }
        }
    }

    void initializeGame() {
        cout << "Enter the name of the first player:\n";
        getline(cin, player1);
        cout << "Enter the name of the second player:\n";
        getline(cin, player2);
        cout << player1 << " is player 1 (x), and will play first.\n";
        cout << player2 << " is player 2 (0).\n";
        printBoard();
    }

    void printBoard() {

        cout << "     |     |     \n";
        cout << "  " << space[0][0] << "  |  " << space[0][1] << "  |  " << space[0][2] << "  \n";
        cout << "||_\n";
        cout << "     |     |     \n";
        cout << "  " << space[1][0] << "  |  " << space[1][1] << "  |  " << space[1][2] << "  \n";
        cout << "||_\n";
        cout << "     |     |     \n";
        cout << "  " << space[2][0] << "  |  " << space[2][1] << "  |  " << space[2][2] << "  \n";
        cout << "     |     |     \n";
```



```cpp
    void takeTurn() {
        int digit;
        if (token == 'x') {
            cout << player1 << ", please enter a number: ";
        } else {
            cout << player2 << ", please enter a number: ";
        }
        cin >> digit;

        if (digit < 1 || digit > 9) {
            cout << "Invalid input! Please enter a number between 1 and 9.\n";
            takeTurn();
            return;
        }

        determinePosition(digit);

        if (space[row][column] != 'x' && space[row][column] != '0') {
            space[row][column] = token;
            token = (token == 'x') ? '0' : 'x';
        } else {
            cout << "The space is already occupied! Try again.\n";
            takeTurn();
        }
    }

    bool checkWinner() {

        for (int i = 0; i < 3; i++) {
            if ((space[i][0] == space[i][1] && space[i][1] == space[i][2]) ||
                (space[0][i] == space[1][i] && space[1][i] == space[2][i])) {
                return true;
            }
        }


        if ((space[0][0] == space[1][1] && space[1][1] == space[2][2]) ||
            (space[0][2] == space[1][1] && space[1][1] == space[2][0])) {
            return true;
        }
```

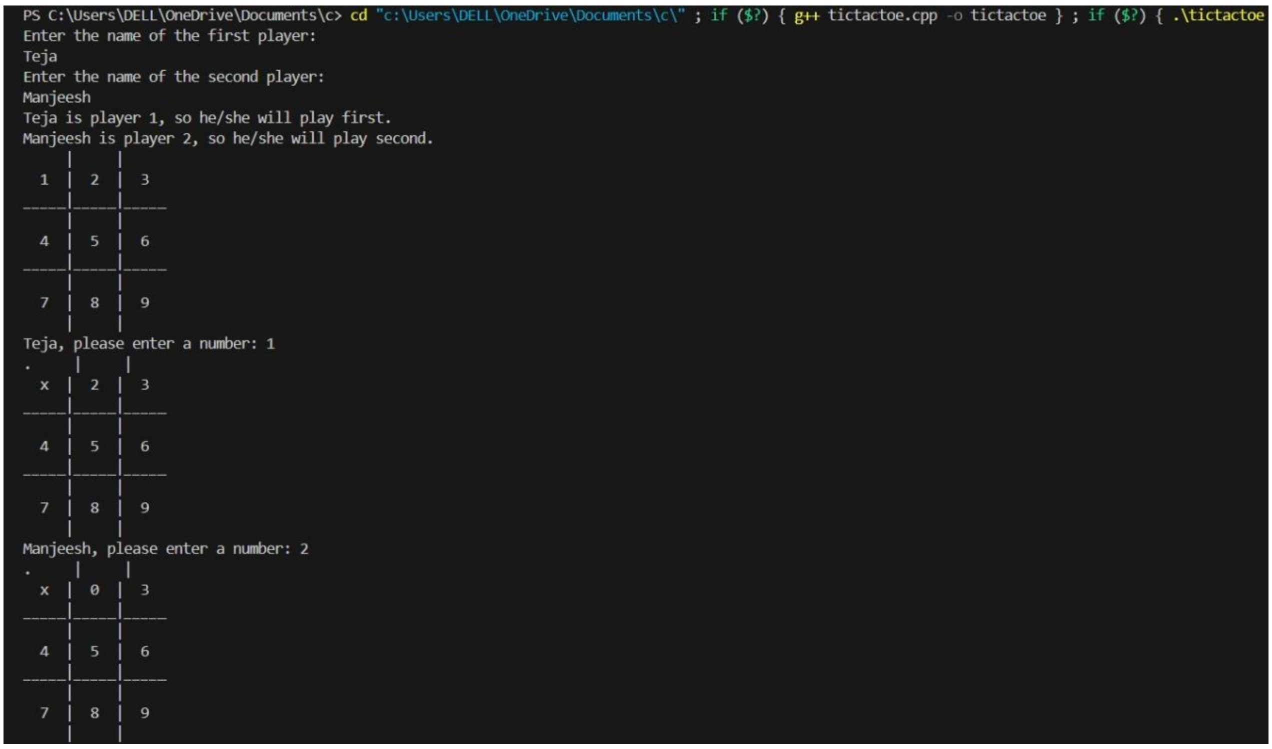


```cpp
        }


        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (space[i][j] != 'x' && space[i][j] != '0') {
                    return false; // The game is still ongoing
                }
            }
        }

        tie = true; // No winner, and the board is full
        return false;
    }

    void announceResult() {
        if (!tie) {
            if (token == 'x') {
                cout << player2 << " wins!!\n";
            } else {
                cout << player1 << " wins!!\n";
            }
        } else {
            cout << "It's a draw!\n";
        }
    }

private:
    void determinePosition(int digit) {
        row = (digit - 1) / 3;
        column = (digit - 1) % 3;
    }
};
```

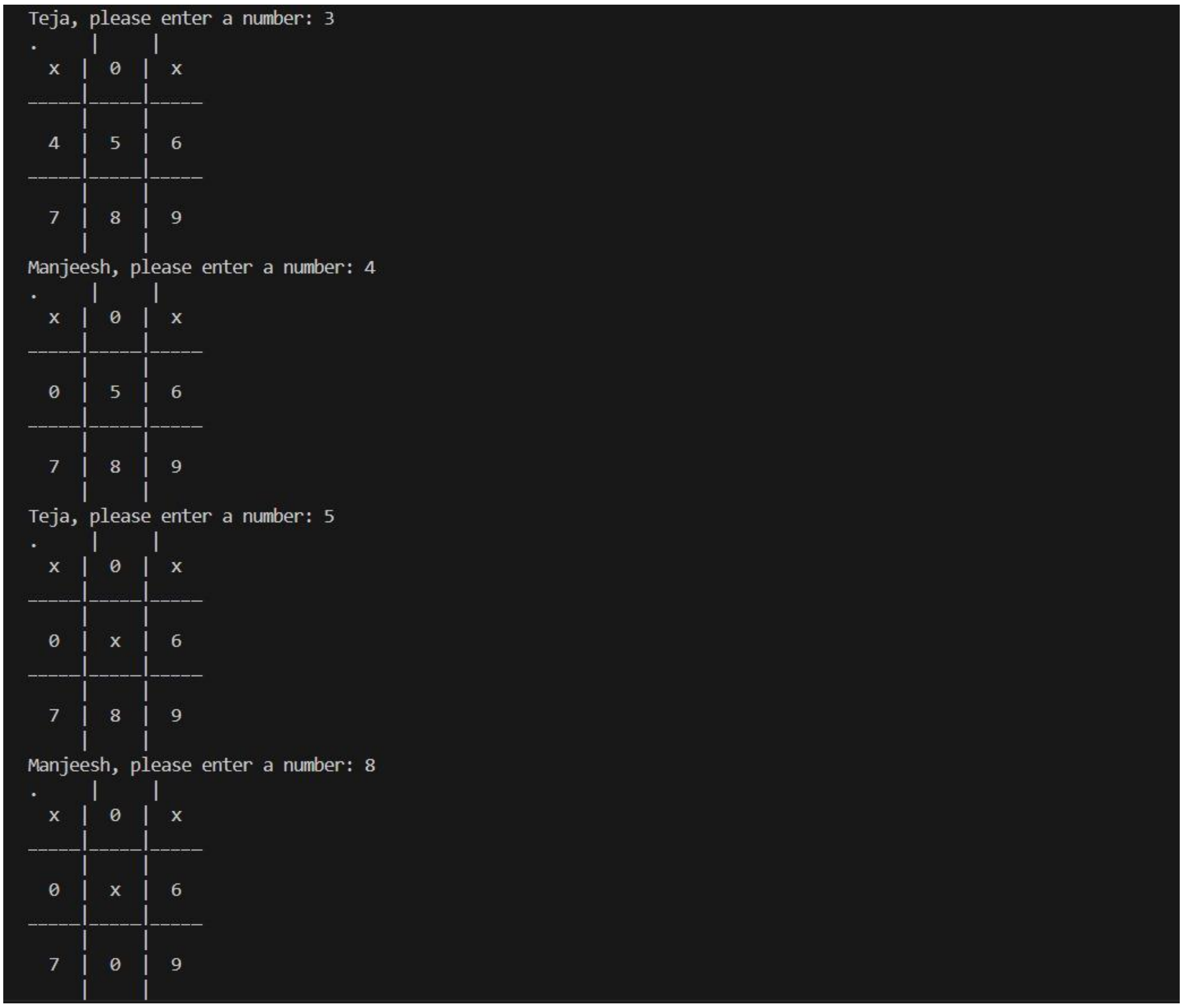


```cpp

int main() {
    TicTacToe game;
    game.initializeGame();

    while (!game.checkWinner()) {
        game.takeTurn();
        game.printBoard();
    }

    game.announceResult();
    return 0;
}
```

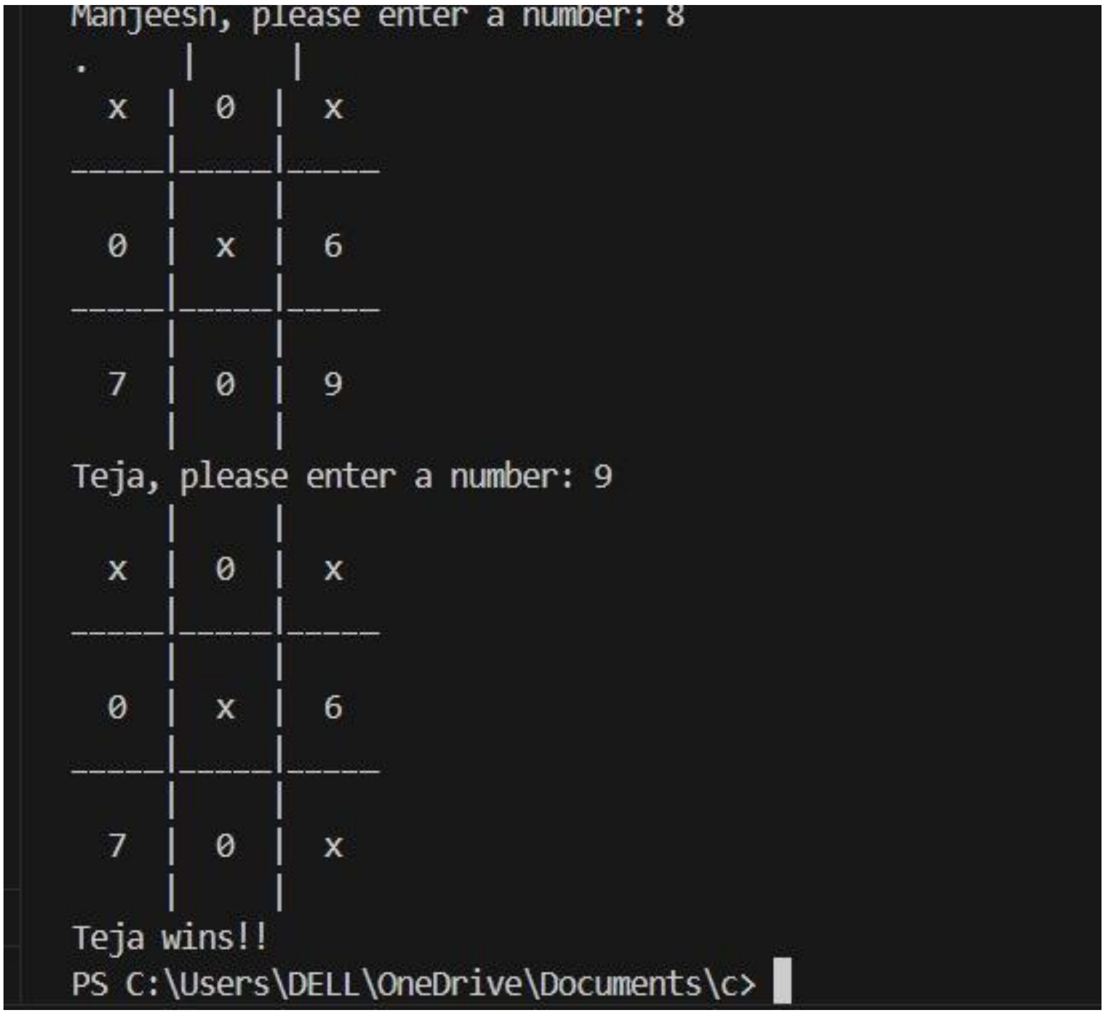
OUTPUT OF THE CODE:



```
PS C:\Users\DELL\OneDrive\Documents\c> cd "c:\Users\DELL\OneDrive\Documents\c\" ; if ($?) { g++ tictactoe.cpp -o tictactoe } ; if ($?) { .\tictactoe
Enter the name of the first player:
Teja
Enter the name of the second player:
Manjeesh
Teja is player 1, so he/she will play first.
Manjeesh is player 2, so he/she will play second.
     |     |
  1  |  2  |  3
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
Teja, please enter a number: 1
.     |     |
  x  |  2  |  3
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
Manjeesh, please enter a number: 2
.     |     |
  x  |  0  |  3
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```



```
Teja, please enter a number: 3
.     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
Manjeesh, please enter a number: 4
.     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  0  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
Teja, please enter a number: 5
.     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  0  |  x  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
Manjeesh, please enter a number: 8
.     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  0  |  x  |  6
_____|_____|_____
     |     |
  7  |  0  |  9
     |     |
```



```
Manjeesh, please enter a number: 8
.     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  0  |  x  |  6
_____|_____|_____
     |     |
  7  |  0  |  9
     |     |
Teja, please enter a number: 9
     |     |
  x  |  0  |  x
_____|_____|_____
     |     |
  0  |  x  |  6
_____|_____|_____
     |     |
  7  |  0  |  x
     |     |
Teja wins!!
PS C:\Users\DELL\OneDrive\Documents\c>
```

4. Concluding Remarks

In conclusion, the development of the Tic-Tac-Toe game has provided valuable insights into fundamental programming concepts such as control structures, functions, and object-oriented design. Through this project, we successfully implemented a user-friendly interface that allows two players to engage in a classic game of strategy.

The key functionalities, including move validation, win detection, and player switching, were effectively integrated to ensure a smooth gaming experience. This project not only reinforced our understanding of C++ programming but also highlighted the importance of logical thinking and problem-solving skills in software development.

Moreover, the simplicity of Tic-Tac-Toe serves as an excellent foundation for exploring more complex game development concepts in the future. As we continue to enhance our programming skills, we can build upon this project by adding features such as an

AI opponent, a graphical user interface, or even online multiplayer capabilities.

Overall, the Tic-Tac-Toe game project has been a rewarding experience, showcasing the practical application of programming principles while fostering creativity and

Innovation in game design.

5. Future Work

Looking ahead, there are several exciting opportunities to enhance the Tic-Tac-Toe game further. One potential improvement is the implementation of an artificial intelligence (AI) opponent, allowing players to compete against the computer at varying difficulty levels. Additionally, developing a graphical user interface (GUI) would significantly improve user experience, making the game more visually appealing and accessible. Another avenue for expansion is to introduce online multiplayer functionality, enabling players to challenge friends or other users over the internet. Furthermore, we could explore variations of the game, such as larger grids or different winning conditions, to add complexity and replayability. These enhancements would not only enrich the gameplay experience but also provide valuable learning opportunities in advanced programming concepts and game development techniques.

References

1. The C++ programming language, BJARNE STROUSTRUP, fourth edition.