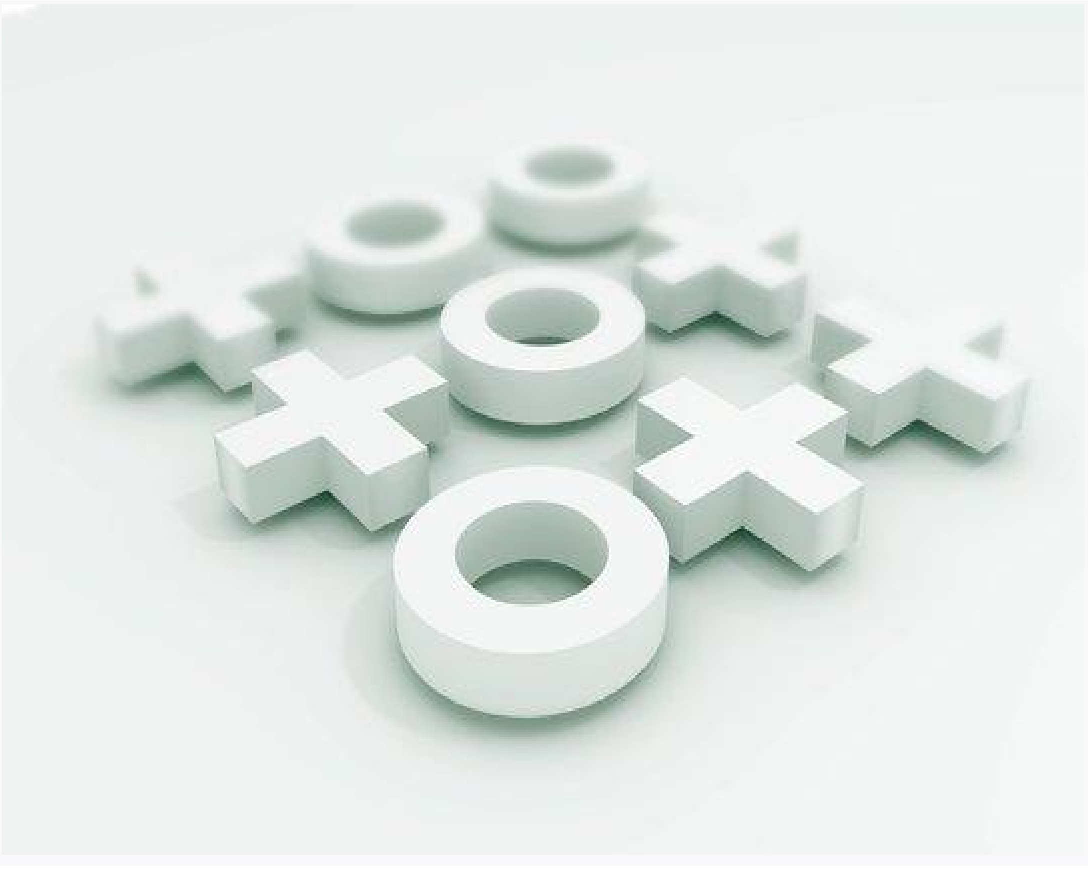
TIC TAC TOE GAME USING C



-Ramisetty Kumar
SBCS INDIA PVT LTD.

Contents:

1. Introduction

Tic Tac Toe, also known as Noughts and Crosses, is a timeless and beloved game enjoyed by people of all ages. It's a classic two-player game that requires strategy, critical thinking, and a dash of luck. The game is deceptively simple, yet it offers endless possibilities and excitement with each move.

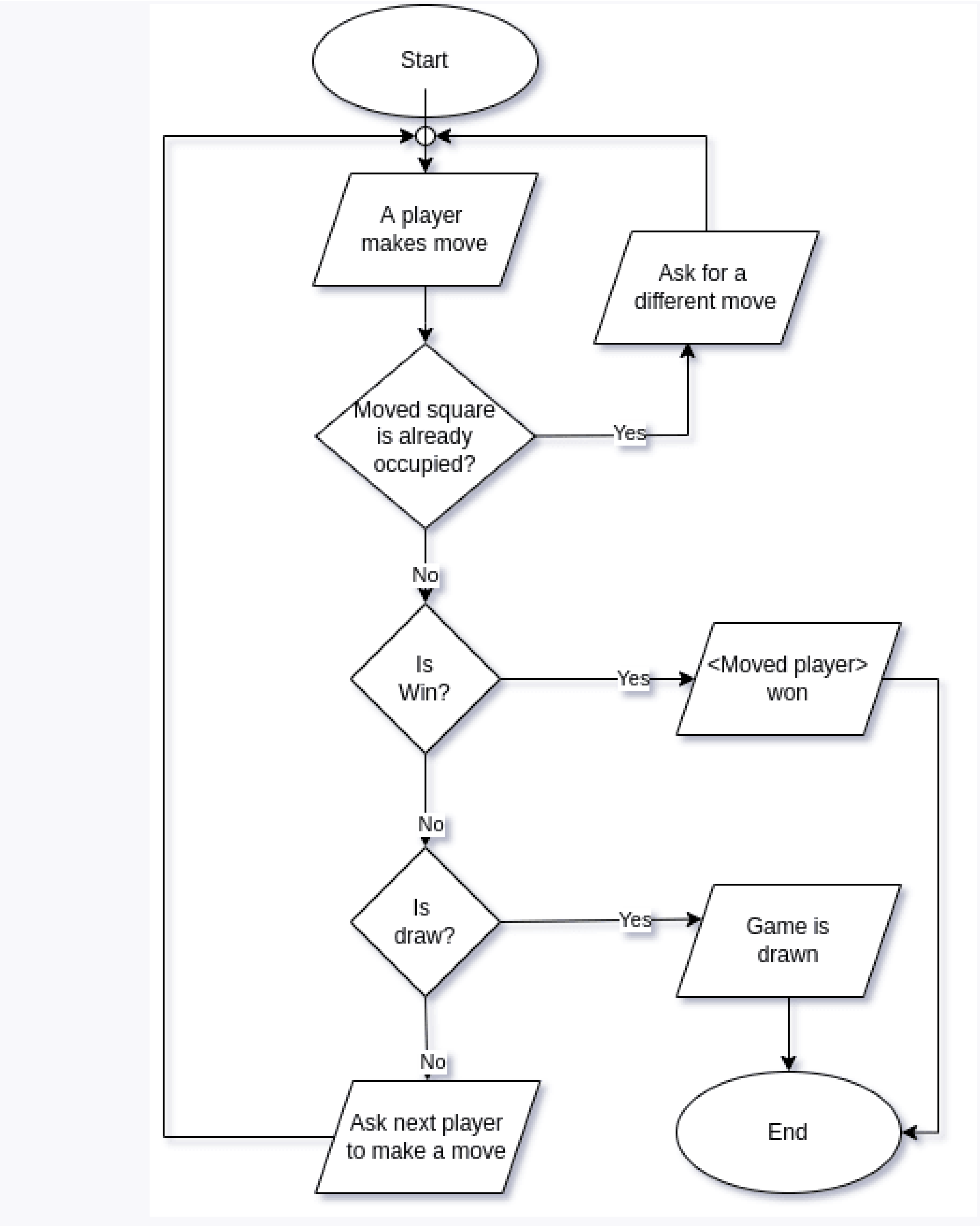
Purpose of the Document:

This document serves as a comprehensive guide and documentation for the Tic Tac Toe game implemented in the C programming language. It includes detailed explanations, a flowchart, and images to facilitate understanding.

Project Overview:

The Tic Tac Toe game is a classic two-player board game. The project aims to create a console-based version of the game, allowing two players to play on the same computer.

2.Flowchart


Start
A player makes move
Moved square is already occupied?
Yes
Ask for a different move
No
Is Win?
Yes
<Moved player> won
No
Is draw?
Yes
Game is drawn
No
Ask next player to make a move
End

3. Game Rules and Objectives

Objective of the Game:

The objective is to be the first player to get three of their marks (X or Y) in a row, either horizontally, vertically, or diagonally, on a 3x3 grid.

Game Components:

Game Board: A 3x3 grid where players make their moves.

How to Win:

A player wins the game if they have three of their marks in a row, either horizontally, vertically, or diagonally.

How to Play:

Players take turns placing their marks on the board. The game continues until one player wins or the board is full, resulting in a draw.

4. Implementation Overview

Programming Language:

The game is implemented in the ‘C’ programming language.

Data Structures:

The game uses a 2D array to represent the game board and simple variables to manage player turns and input.

Game Logic:

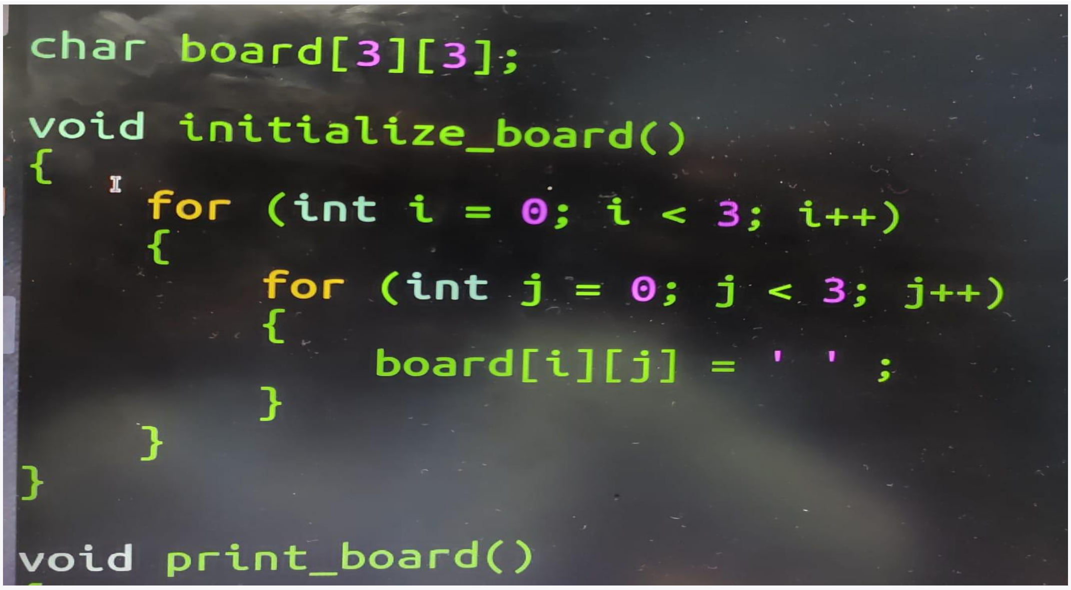
The core game logic involves checking for a win after each move and determining if the game ends in a draw.

5. Code Explanation

In this section, you will find detailed explanations of key functions used in the game:

initialize_board():

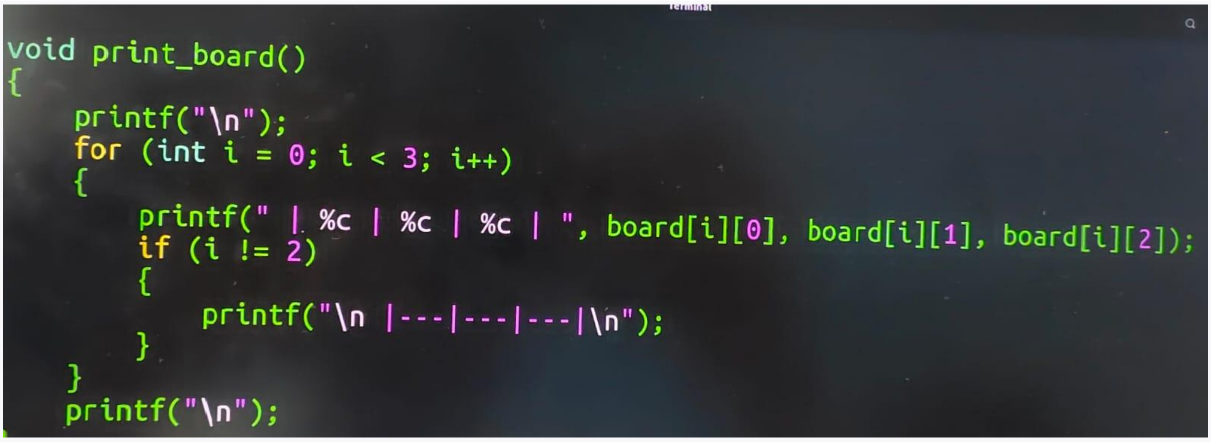
This function initializes the game board by filling all cells with empty spaces (' ').



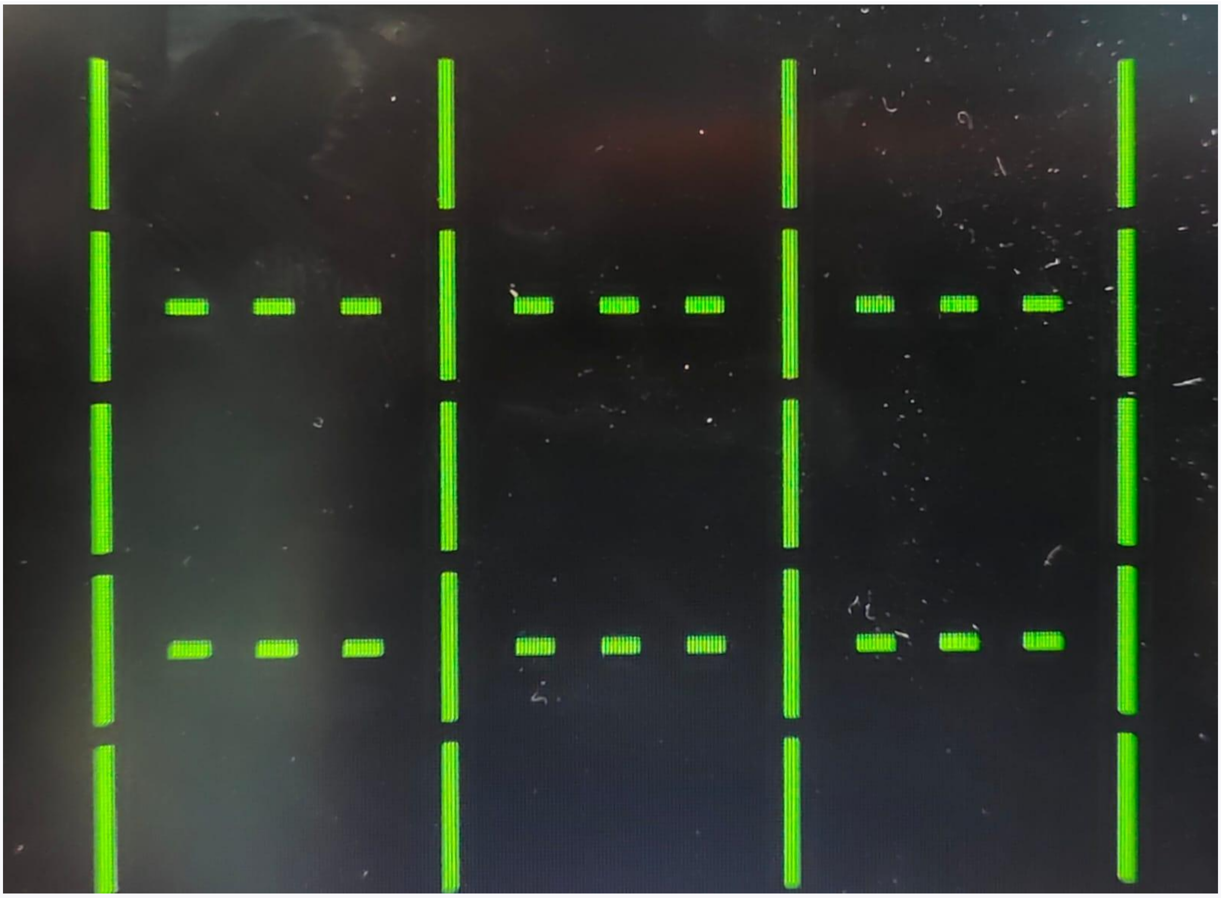
```
char board[3][3];

void initialize_board()
{
     for (int i = 0; i < 3; i++)
     {
          for (int j = 0; j < 3; j++)
          {
               board[i][j] = ' ' ;
          }
     }
}

void print_board()
```

print_board():

The print_board() function displays the current state of the game board in a visually appealing format.

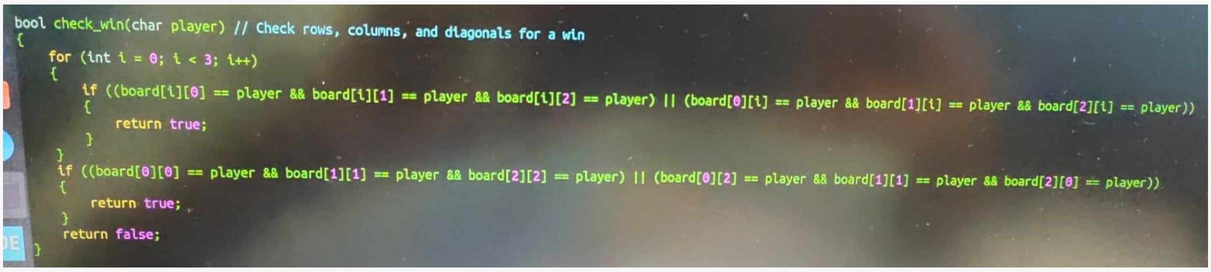


```
void print_board()
{
    printf("\n");
    for (int i = 0; i < 3; i++)
    {
        printf(" | %c | %c | %c | ", board[i][0], board[i][1], board[i][2]);
        if (i != 2)
        {
            printf("\n |---|---|---|\n");
        }
    }
    printf("\n");
```



check_win(char player):

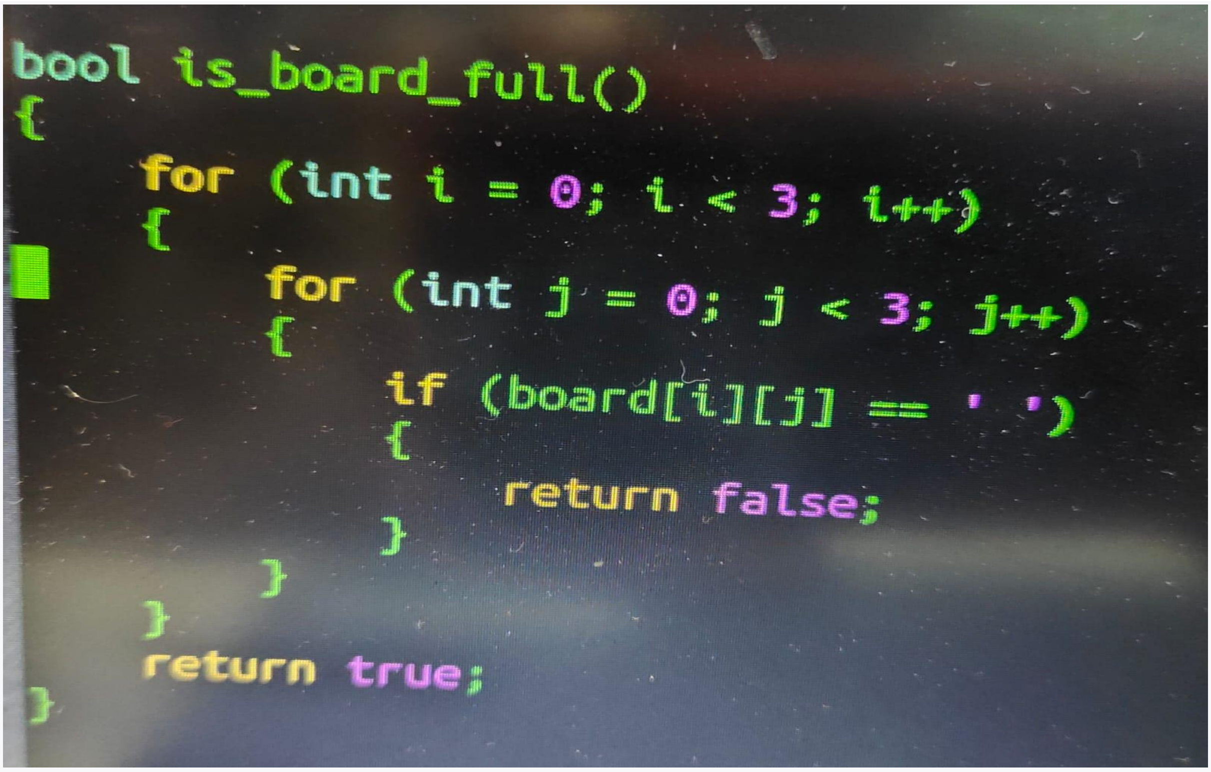
The check_win(char player) function checks if the specified player (X or Y) has won the game. It examines rows, columns, and diagonals to determine if the player has three marks in a row.



```
bool check_win(char player) // Check rows, columns, and diagonals for a win
{
    for (int i = 0; i < 3; i++)
    {
        if ((board[i][0] == player && board[i][1] == player && board[i][2] == player) || (board[0][i] == player && board[1][i] == player && board[2][i] == player))
        {
            return true;
        }
    }
    if ((board[0][0] == player && board[1][1] == player && board[2][2] == player) || (board[0][2] == player && board[1][1] == player && board[2][0] == player))
    {
        return true;
    }
    return false;
}
```

is_board_full():

The is_board_full() function checks if the game board is completely filled with marks. It iterates through the entire board and returns ‘true’ if there are no empty spaces left.



```
bool is_board_full()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (board[i][j] == ' ')
            {
                return false;
            }
        }
    }
    return true;
}
```

6. Conclusion

In this project,we successfully implemented a console-based Tic Tac Toe game in the C programming language.The primary objective was to recreate the classic board game experience in a digital format,allowing players to enjoy it conveniently on their computers

User experience:

Players can enjoy the timeless game of Tic Tac Toe on their computers with our user_friendly console interface.The game’s simple design and clear instructions make it accessible to players of all ages.

7. Future Improvements

1.Online Multiplayer Mode: Implement a server-client architecture that allows players to connect to a centralized game server over the internet.

2.Real-Time Gameplay: Enable real-time communication between players. Each player's moves are transmitted to the server and relayed to the opponent's computer, ensuring a synchronized gaming experience.

3.Enhanced Graphics: Transition from a console interface to a graphical user interface (GUI) with enhanced graphics and animations for a more visually appealing experience.

4.Voice Chat: Integrate voice chat capabilities to enable players to communicate during the game, adding a social dimension to the experience.

8. References

- GitHub: A widely-used platform for version control and collaboration on software projects.