

TIC TAC TOE GAME

Submitted by

Name of the Students: Aniket Mondal

Enrolment Number: 12022002019074

Section: D

Class Roll Number: 72

Stream: CSE-IoT

Subject: Programming for Problem Solving using C language

Subject Code: ESC103(Pr.)

Department: Basic Science and Humanities

Under the supervision of
Swarnendu Ghosh

Academic Year: 2022-26

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE SECOND SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITITES
INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA**



CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by **Aniket Mondal**, entitled **TIC TAC TOE Game** be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the second semester.

Head of the Department
Basic Sciences and Humanities
IEM, Kolkata

Project Supervisor

1 Introduction

This is about to design a program that allows two players to play the classic game of Tic-Tac-Toe.

1.1 Objective

To create a C program on Tic Tac Toe Game project.

1.2 Organization of the Project

I used the basic *stdio.h* and then I used the struct system to create a structure variable and then I added functions to call them.

2 Programs

```
#include <stdio.h>
```

```
char board[3][3]; // Tic Tac Toe board
```

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

```
void printBoard() {  
    printf("\n");  
    printf(" %c | %c | %c\n", board[0][0], board[0][1], board[0][2]);  
    printf("----+----+----\n");  
    printf(" %c | %c | %c\n", board[1][0], board[1][1], board[1][2]);  
    printf("----+----+----\n");  
    printf(" %c | %c | %c\n", board[2][0], board[2][1], board[2][2]);  
}
```

```

    printf("\n");
}

int checkWin() {
    // Check rows
    int i, j, k, l;
    for ( i = 0; i < 3; i++) {
        if (board[i][0] == board[i][1] && board[i][1] == board[i][2] && board[i][0] != '
    ') {
        return 1;
        }
    }

    // Check columns
    for ( j = 0; j < 3; j++) {
        if (board[0][j] == board[1][j] && board[1][j] == board[2][j] && board[0][j] != '
    ') {
        return 1;
        }
    }

    // Check diagonals
    if ((board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0] != '
    ') ||
        (board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2] != '
    ')) {
        return 1;
    }

    // Check for a tie
    int tie = 1;
    for ( k = 0; k < 3; k++) {
        for ( l = 0; l < 3; l++) {
            if (board[k][l] == ' ') {
                tie = 0;
                break;
            }
        }
    }
    if (tie) {
        return 2;
    }
}

```

```

    return 0;
}

int main() {
    int currentPlayer = 1; // Player 1 starts
    int row, col;
    int gameOver = 0;
    int winner;

    initializeBoard();

    printf("Tic Tac Toe Game\n");

    while (!gameOver) {
        printBoard();

        printf("Player %d's turn.\n", currentPlayer);
        printf("Enter the row (0-2): ");
        scanf("%d", &row);
        printf("Enter the column (0-2): ");
        scanf("%d", &col);

        if (row < 0 || row > 2 || col < 0 || col > 2 || board[row][col] != ' ') {
            printf("Invalid move. Please try again.\n");
            continue;
        }

        if (currentPlayer == 1) {
            board[row][col] = 'X';
            currentPlayer = 2; // switch to player 2
        }
        else {
            board[row][col] = 'O';
            currentPlayer = 1; // switch to player 1
        }
    }
}

```

3. Outputs:

```
Tic Tac Toe Game

  |  | 
--+--+
  |  | 
--+--+
  |  | 

Player 1's turn.
Enter the row (0-2): 2
Enter the column (0-2): 1
  |  | 
--+--+
  |  | 
--+--+
  | X | 

Player 2's turn.
Enter the row (0-2): 1
Enter the column (0-2): 1
  |  | 
--+--+
  | O | 
--+--+
  | X |
```