

TIC TAC TOE

21CSS101J- PROGRAMMING FOR PROBLEM SOLVING

Mini Project Report

Submitted by

P MANVITHA RAYAL [RA2211003010001]

B.Tech CSE CORE

NISHTHA DABAS [RA2211003010011]

B.Tech CSE CORE



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

**SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(under section 3 of UGC Act,1956)
S.R.M NAGAR, KATTANKULATHUR-603 203
KANCHEEPURAM DISTRICT
DECEMBER 2022**

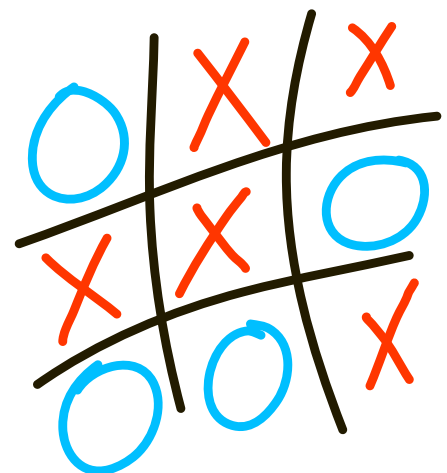
TABLE OF CONTENTS

chapter no.	title	page no
1	problem statement	3
2	methodology/procedure	4
3	coding	5
4	result	12
5	conclusion	13

PROBLEM STATEMENT

Basic Problem Statement :

You are supposed to design the game of tic-tac-toe using microcontroller(s). The circuit should be designed to play against a human player, through a suitable interface. The player can give inputs through simple switches and the display can be made on a matrix of LEDs. The game algorithm should be such that the circuit cannot be beaten (in the hardest level in case of multiple levels), and in the worst case the game ends in a draw.



PROCEDURE

The methods we used in our program are as follows:

private void strongLevel O:

To apply artificial intelligence in our program.

private int checkRow (int rownum, String rowname):

To check the combination whether any two symbols(X or O) are same for winning or blocking.

private int checkColumn(int columnnum, String columnname):

To check the combination whether any two symbols(X or O) are same for winning or blocking.

private Boolean checkCorner(String cornername, int countl):

To check the combination whether any two symbols(X or o) are same for winning or blocking.

private void start.NewGame:

To display the options for player.

private void set Value(int number, int position):

To set the symbols in the right place for winning and blocking.

private void setElse:

To set the symbols in an empty place randomly.

private void setCornero:

To create scope for computer to win in the third move

CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <time.h>
```

```
char board[3][3];
const char PLAYER = 'X';
const char COMPUTER = 'O';
```

```
void resetBoard();
void printBoard();
int checkFreeSpaces();
void playerMove();
void computerMove();
char checkWinner();
void printWinner(char);
```

```
int main()
{ printf("This is the game of Tic Tac Toe Created By NISHTHA.\n");
  char winner = '';
  char response = '';

  do
  {
    winner = '';
    response = '';
    resetBoard();

    while(winner == '' && checkFreeSpaces() != 0)
    {
```

CODE

```
printBoard();

    playerMove();
    winner = checkWinner();
    if(winner != ' ' || checkFreeSpaces() == 0)
    {
        break;
    }

    computerMove();
    winner = checkWinner();
    if(winner != ' ' || checkFreeSpaces() == 0)
    {
        break;
    }
}

printBoard();
printWinner(winner);

printf("\nWould you like to play again? (Y/N): ");
scanf("%c");
scanf("%c", &response);
response = toupper(response);
} while (response == 'Y');

printf("Thanks for playing!");

return 0;
}
```

CODE

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

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

int checkFreeSpaces()
{
    int freeSpaces = 9;

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            if(board[i][j] != ' ')
```

CODE

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

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

int checkFreeSpaces()
{
    int freeSpaces = 9;

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            if(board[i][j] != ' ')
```


CODE

```
{
    freeSpaces--;
}
}
}
return freeSpaces;
}
void playerMove()
{
    int x;
    int y;

    do
    {
        printf("Enter row #(1-3): ");
        scanf("%d", &x);
        x--;
        printf("Enter column #(1-3): ");
        scanf("%d", &y);
        y--;

        if(board[x][y] != ' ')
        {
            printf("Invalid move!\n");
        }
        else
        {
            board[x][y] = PLAYER;
            break;
        }
    }
```

CODE

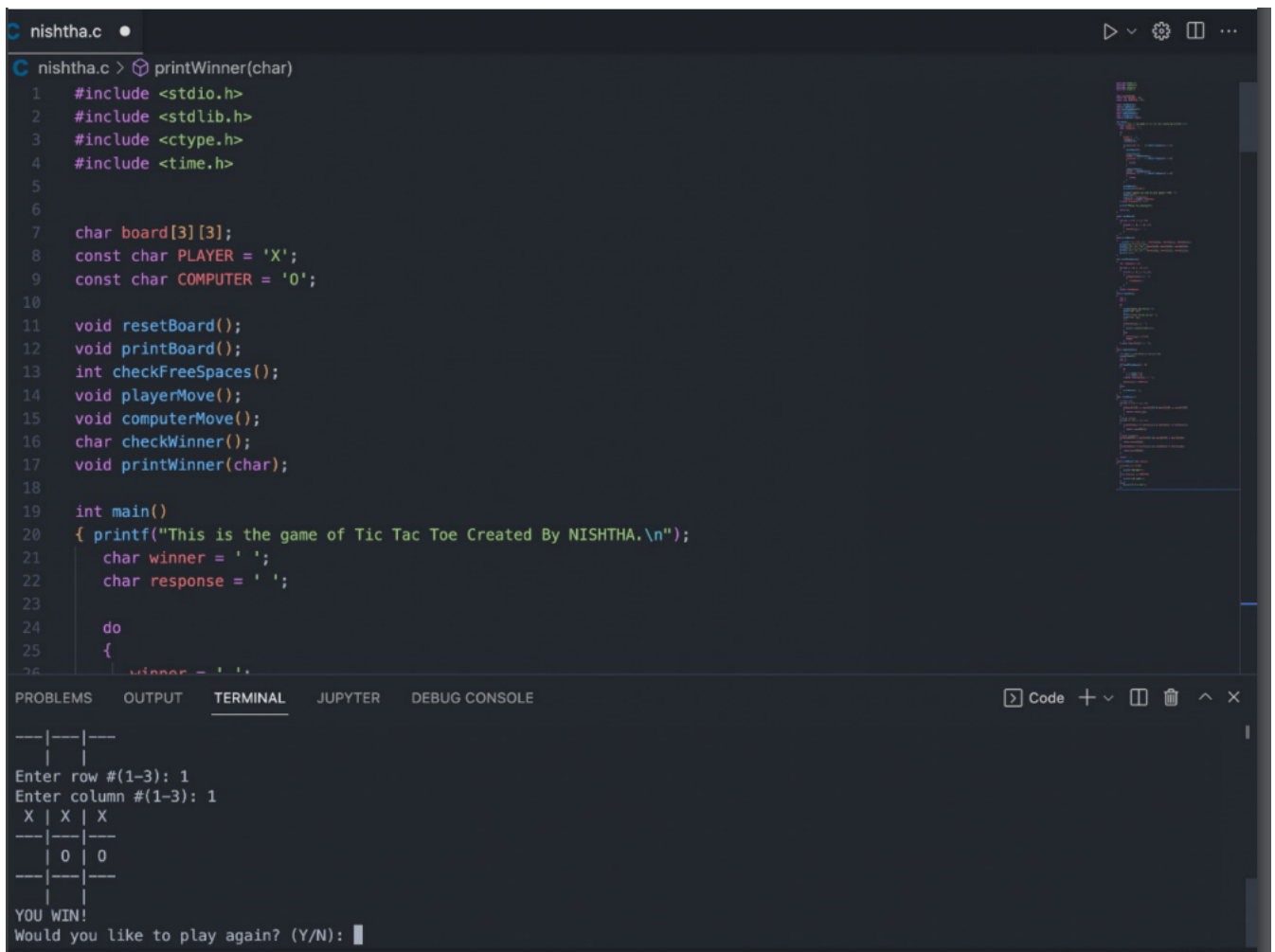
```
}  
} while (board[x][y] != ' ');  
  
}  
void computerMove()  
{  
    //creates a seed based on current time  
    srand(time(o));  
    int x;  
    int y;  
  
    if(checkFreeSpaces() > 0)  
    {  
        do  
        {  
            x = rand() % 3;  
            y = rand() % 3;  
            } while (board[x][y] != ' ');  
  
            board[x][y] = COMPUTER;  
        }  
        else  
        {  
            printWinner(' ');  
        }  
    }  
}  
char checkWinner()  
{  
    //check rows  
    for(int i = 0; i < 3; i++)  
    {  
        if(board[i][0] == board[i][1] && board[i][0] == board[i][2])  
        {  
            return board[i][0];  
        }  
    }  
}
```

CODE

```
}}
}
//check columns
for(int i = 0; i < 3; i++)
{
    if(board[0][i] == board[1][i] && board[0][i] == board[2][i])
    {
        return board[0][i];
    }
}
//check diagonals
if(board[0][0] == board[1][1] && board[0][0] == board[2][2])
{
    return board[0][0];
}
if(board[0][2] == board[1][1] && board[0][2] == board[2][0])
{
    return board[0][2];
}

return ' ';
}
void printWinner(char winner)
{
    if(winner == PLAYER)
    {
        printf("YOU WIN!");
    }
    else if(winner == COMPUTER)
    {
        printf("YOU LOSE!");
    }
    else{
        printf("IT'S A TIE!");
    }
}
```

OUTPUT



The screenshot shows a VS Code editor with a file named `nishtha.c`. The code is a C program for a Tic Tac Toe game. It includes headers for `stdio.h`, `stdlib.h`, `ctype.h`, and `time.h`. It defines a 3x3 board, player characters 'X' and 'O', and functions for resetting the board, printing the board, checking for free spaces, player and computer moves, checking for a winner, and printing the winner. The `main` function prints a message, initializes variables, and enters a loop for the game.

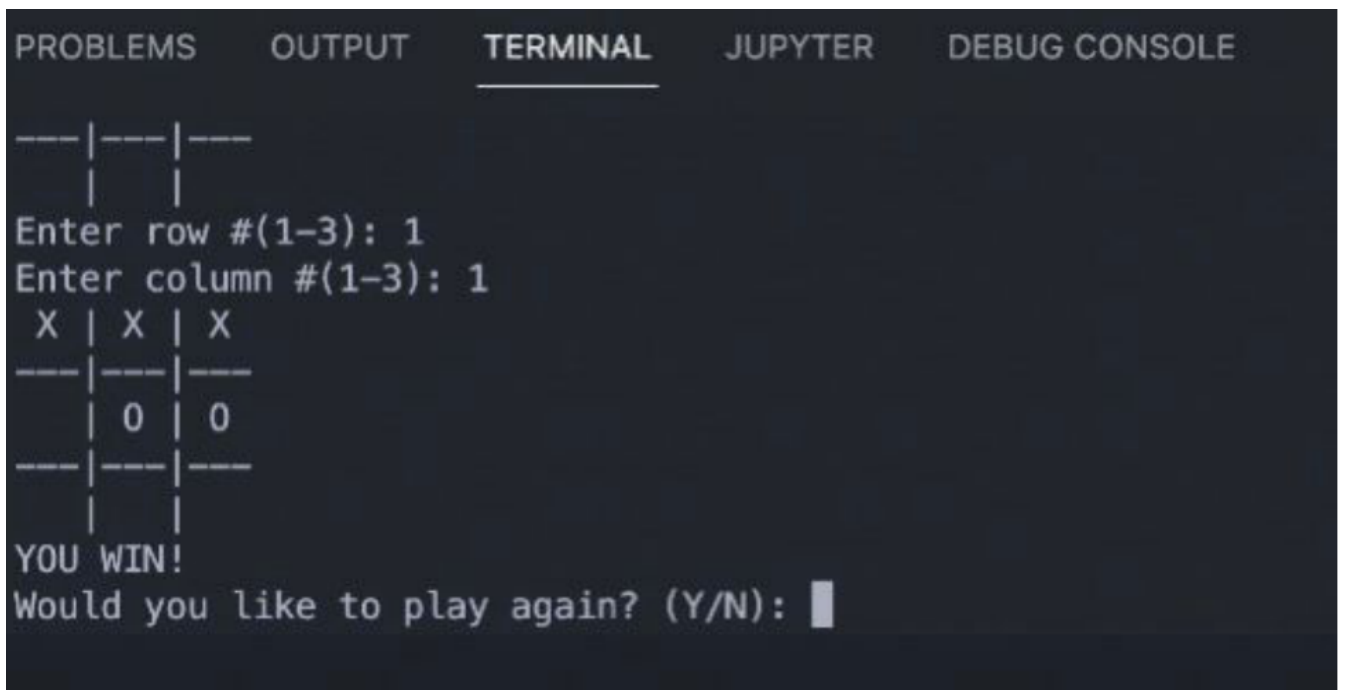
```
nishtha.c > printWinner(char)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4  #include <time.h>
5
6
7  char board[3][3];
8  const char PLAYER = 'X';
9  const char COMPUTER = 'O';
10
11 void resetBoard();
12 void printBoard();
13 int checkFreeSpaces();
14 void playerMove();
15 void computerMove();
16 char checkWinner();
17 void printWinner(char);
18
19 int main()
20 { printf("This is the game of Tic Tac Toe Created By NISHTHA.\n");
21   char winner = ' ';
22   char response = ' ';
23
24   do
25   {
26     winner = ' ';
```

The terminal output shows the game in progress. The board is displayed as:

```

| | |
| | |
| 0 | 0
| | |
```

The user has entered row 1 and column 1. The terminal shows "YOU WIN!" and asks "Would you like to play again? (Y/N):".



This block is a close-up of the terminal output from the VS Code editor. It shows the same Tic Tac Toe game state as the previous block, with the board and the prompt "Would you like to play again? (Y/N):".

```

| | |
| | |
| 0 | 0
| | |

YOU WIN!
Would you like to play again? (Y/N):
```

CONCLUSION

In this tic tac toe Program, we have used Visual Studio Code having MinGW 64-bit Compiler. Here, we have developed a mini project in C programming language called tic tac toe. The object of the game is line up 3 X markers or 3 O markers in a 3 by 3 grid. The markers can run horizontally, vertically or diagonally. Turns alternate between players. The version we build here allows two players to compete against each other or to have a single player compete against a pretty smart computer. This program can be optimized and is user friendly, the output is simple and can be understood easily and operated. The program can be used for the sake of entertainment purpose.



Thank You

SUBMITTED BY:
P MANVITHA RAYAL
(RA2211003010001)
NISHTHA DABAS
(RA2211003010011)

