

# **Tic tac toe game**

## **A Mini Project**

**Academic Year: 2021 - 22 ODDSEMESTER**

**Department with Specialization : B. Tech – Computer Science and  
Engineering with Specialization  
in Artificial Intelligence and  
machine Learning**

**Semester : I**  
**Course Code : 18CSS101J**  
**Course Title : Programming and Problem  
Solving**

*Submitted by*

**Shaik Mohammed Fahad [RA2111026010331]**

**Krsna Asthana [RA2111026010351]**

**Akshath kumar [RA2111026010349]**

**Shubham Gour [RA2111026010328]**

**Aniket Sonar [RA2111026010357]**

*Under the Guidance of*

**Prabakaran J**

**(Professor,SRM)**



**DEPARTMENT OF NETWORKING AND COMMUNICATIONS  
COLLEGE OF ENGINEERING AND TECHNOLOGY**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

FEBRUARY 2022

## **Abstract for Tic Tac Toe Game project:**

**Tic-Tac-Toe game can be played by two players where the square block (3 x 3) can be filled with a cross (X) or a circle (O). The game will toggle between the players by giving the chance for each player to mark their move. When one of the players make a combination of 3 same markers in a horizontal, vertical or diagonal line the program will display which player has won, whether X or O. In this program, we implement a 3x3 tic-tac-toe game. The game is designed so that two players can play tic-tac-toe using C-language. The program will contain a display function and a select function to place the symbol as well as toggle between the symbols allowing each player a turn to play the game. The program will update after each player makes their move and check for the conditions of game as it goes on. Overall program works without any bugs and is able to use**

## Algorithm

Tic-Tac-Toe is a two player game where each player uses a signature. The player who successfully places three respective signatures in a vertical, horizontal or diagonal row is the winner.

**Step 1:** Start

**Step 2:** Use 3 X 3 matrix to get 9 cell tic tac toe board.

**Step 3:** Signature 'X' is used for player 1 and 'O' for player 2. And the players play alternatively.

**Step 4:** Get index as input from the players. And the index range is from cell 1 to cell 9.

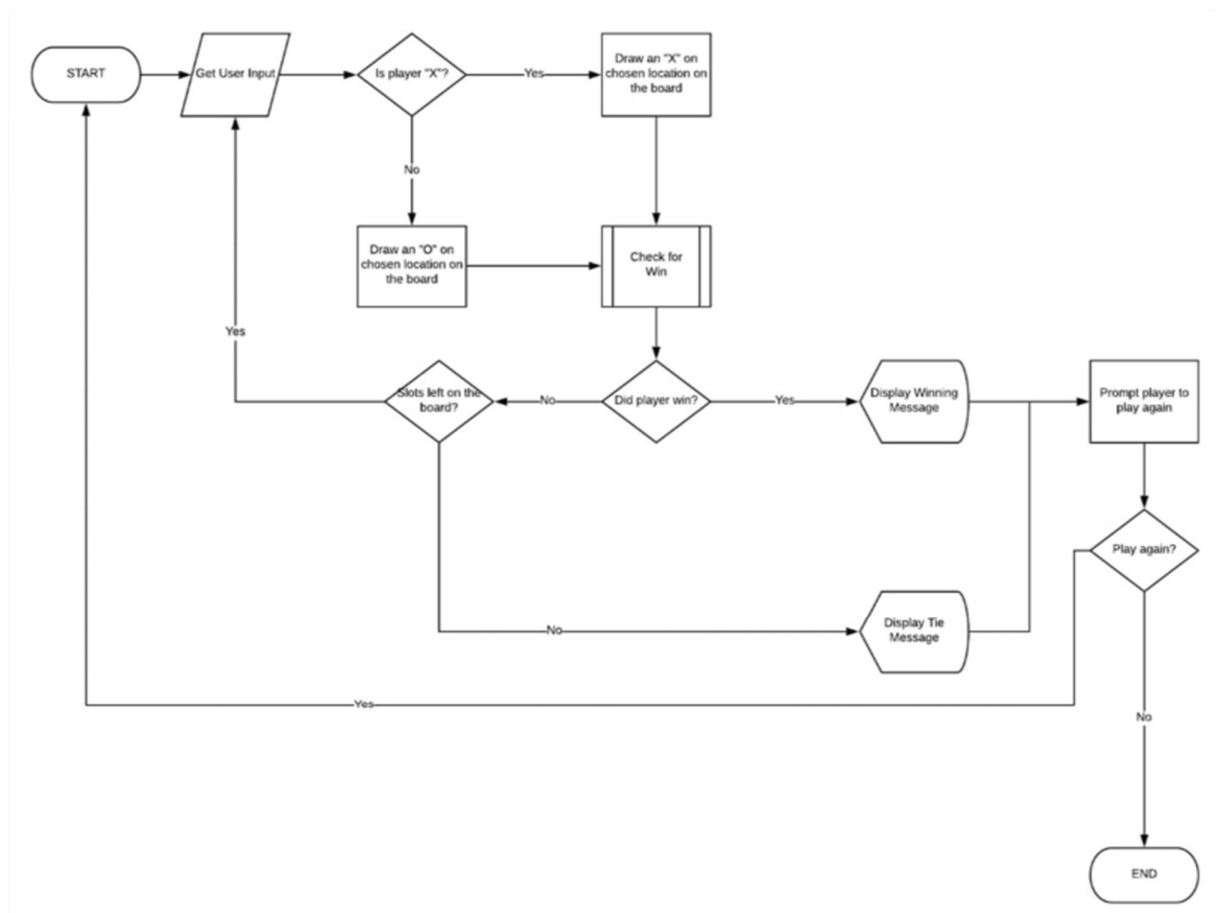
**Step 5:** After each move, check whether that player has placed three respective signatures in vertical, horizontal or diagonal row. If yes, declare that particular player as winner.

**Step 6:** If no, give chance for other player.

**Step 7:** If all the cells in the Tic-Tac-Toe board got filled, then the game is drawn.

## Step 8: End

## Flowchart



## Source Code

```
#include <stdio.h>

char square[10] = { 'o', '1', '2', '3', '4', '5', '6', '7', '8', '9' }; // POSITION OF SQAURE

int checkwin();    // to check winner
void board();

int main()
{

printf("*****\n");

    printf("\t\t\t\t\tWELCOME\n");
printf("\t\t\t\t\tTIC TAC TOE GAME\n");
printf("*****\n");

    int player = 1, i, choice; // i= to check game is in progress or not

    char mark; // X,O
    do
    {
        board();
        player = (player % 2) ? 1 : 2;

        printf("Player %d, enter a number: ", 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 move ");

    player--;
    getch();
}
i = checkwin();
```

```
        player++;
    }while (i == - 1);

    board();

    if (i == 1)
        printf("==>\aPlayer %d win ", --player);
    else
        printf("==>\aGame draw");

    getch();

    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 board()
{

    printf("\n\n\tTic Tac Toe\n\n");

    printf("Player 1 (X) - Player 2 (O)\n\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");

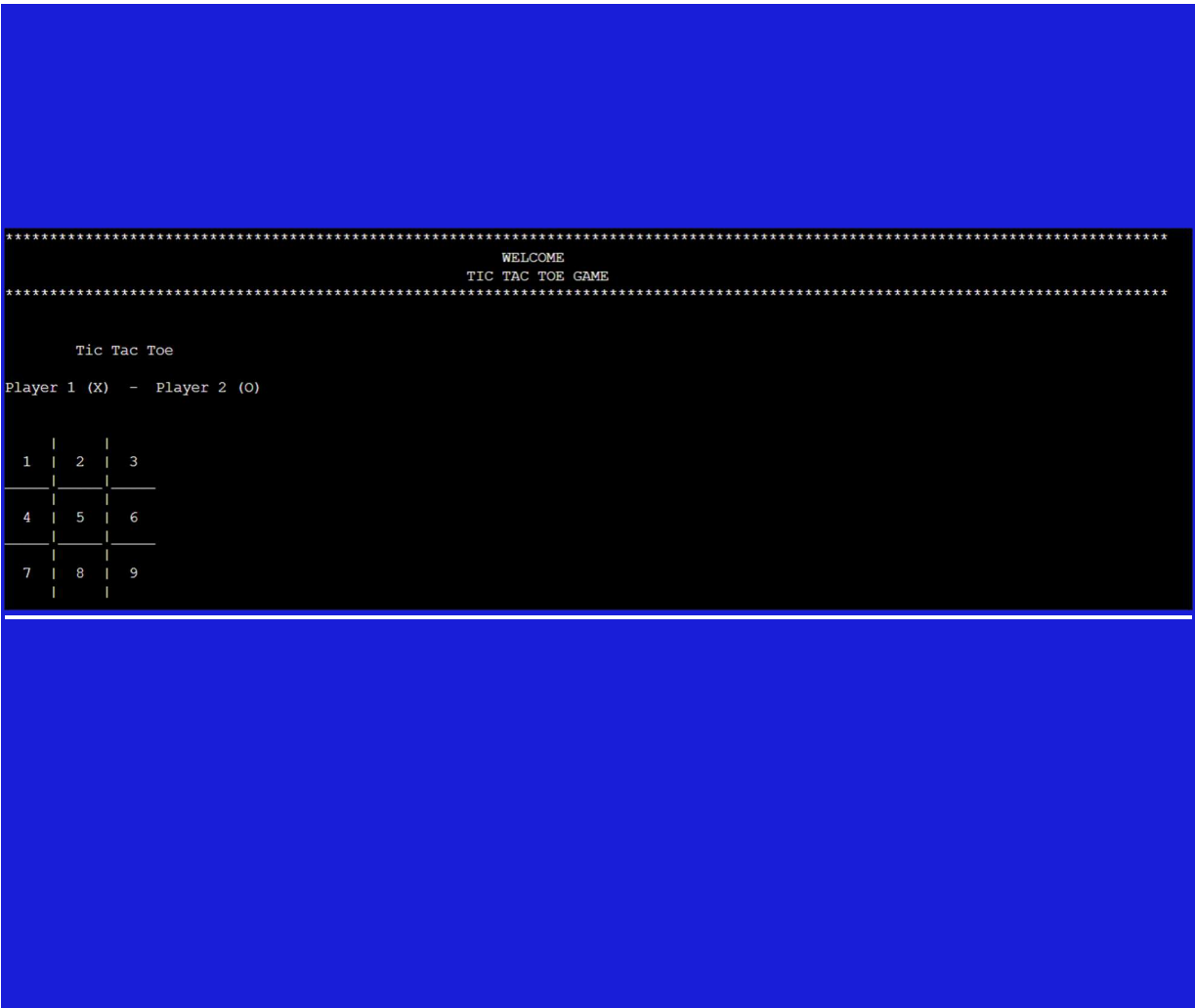
    printf(" %c | %c | %c \n", square[7], square[8], square[9]);

    printf("  |  |  \n\n");
}
```

---



# Screenshots



Player 1, enter a number: 1

Tic Tac Toe

Player 1 (X) - Player 2 (O)

X	2	3
4	5	6
7	8	9

Player 2, enter a number: 2

Tic Tac Toe

Player 1 (X) - Player 2 (O)

X	O	3
4	5	6
7	8	9

Player 1, enter a number: 4

### Tic Tac Toe

Player 1 (X) - Player 2 (O)

X		O		3
<hr/>				
X		5		6
<hr/>				
7		8		9

Player 2, enter a number: 5

### Tic Tac Toe

Player 1 (X) - Player 2 (O)

X		O		3
<hr/>				
X		O		6
<hr/>				
7		8		9

```
Player 1, enter a number: 7

      Tic Tac Toe

Player 1 (X)  -  Player 2 (O)

  X |  O |  3
  --+---+--
  X |  O |  6
  --+---+--
  X |  8 |  9

==>Player 1 win

...Program finished with exit code 0
Press ENTER to exit console.
```

## Result:

The Tic Tac Toe game is most familiar among all the age groups. Intelligence can be a property of any purpose-driven decision maker. This basic idea has been suggested many times. An algorithm of playing Tic Tac Toe has been presented and tested that works in efficient way. Overall, the system works without any bugs.

## Conclusion:

We make Tic Tac Toe game successfully with the help of C language and it is very entertaining.

---

