# **PPS- MINI PROJECT**

# Tic-tac-toe

#### PREPARED BY:

SRISTI KUMARI(RA2111003011025)

**ANANMAY ANAND(RA2111003011011)** 

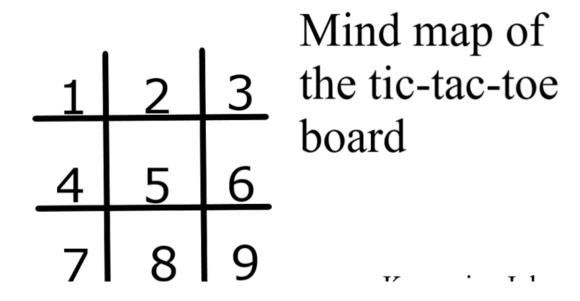
#### **Problem definition:**

Tic-tac-toe is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. Here, I have developed a mini project in C Tic Tac Toe game – a simple console application without graphics. It is the same noughts and crosses or the Xs and Os, the other names for Tic-Tac-Toe, we have played with paper and pencil during our school days! This Tic Tac Toe game in C is compiled in Code::Blocks with gcc compiler.

### **Abstract:**

While making a Tic Tac Toe game using C language, we paid special attention to the use of arrays. The Xs and Os are kept in different arrays, and they are passed between several functions in the code to keep track of how the game goes. With the code here one can play the game choosing either X or O **against the computer**.

This Tic Tac Toe C game is such that one will have to input a numeric character, from 1 to 9, to select a position for X or O into the space/block one wants. For example: if one is playing with O and they input 2, the O will go to the first row – second column. If they want to place O in the third row – first column, you have to enter 7. And, it is similar for the other positions.



### **List of functions used:**

#### void menu()

In this mini project, this function displays the menu or welcome screen of this project. With this function, you can select whether you wish to play the game with X or with O or can even exit the game.

#### void go(int n)

For deciding whose turn it is

#### void start\_game()

For the scenario in which the computer wins

#### void check\_draw()

For the scenario in which the game comes to a draw

void draw\_board()

void player\_first()

For the scenario in which the player wins

**void put\_X\_O(char ch, int pos)** – This function puts one of the numerical characters one inputs into the respective position in Tic-Tac-Toe. For example: if one is playing with X and they input 2, the X will go to the first row – second column. If they want to place X in the third row – first column, they have to enter 7. And, it is similar for the other positions.

void gotoxy (int x, int y) - This function allows the user to print text in any
place on the screen.

#### Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include <windows.h>
int board[10] = \{2,2,2,2,2,2,2,2,2,2,2\};
int turn = 1, flag = 0;
int player, comp;
void menu();
void go(int n);
void start game();
void check draw();
void draw board();
void player first();
void put X O(char ch, int pos);
COORD coord= {0,0}; // this is global variable
//center of axis is set to the top left cornor of the screen
void gotoxy(int x, int y)
    coord.X=x;
    coord.Y=y;
    SetConsoleCursorPosition(GetStdHandle(STD OUTPUT HANDLE),coord);
}
void main()
    system("cls");
    menu();
    getch();
void menu()
    int choice;
    system("cls");//to clear the previous screen
    printf("\n----");
    printf("\n1 : Play with X");
    printf("\n2 : Play with 0");
    printf("\n3 : Exit");
    printf("\nEnter your choice:>");
    scanf("%d", &choice);
    turn = 1;
    switch (choice)
    case 1:
       player = 1;
        comp = 0;
        player first();
        break;
    case 2:
       player = 0;
        comp = 1;
        start game();
        break;
```

```
case 3:
       exit(1);
    default:
       menu();
}
int make2()
    if(board[5] == 2)
       return 5;
    if(board[2] == 2)
        return 2;
    if(board[4] == 2)
        return 4;
    if(board[6] == 2)
        return 6;
    if(board[8] == 2)
       return 8;
    return 0;
}
int make4()
{
    if(board[1] == 2)
        return 1;
    if(board[3] == 2)
        return 3;
    if(board[7] == 2)
        return 7;
    if(board[9] == 2)
       return 9;
    return 0;
}
int posswin(int p)
// p==1 then X p==0 then O
    int i;
    int check_val,pos;
    if(p == 1)
        check val = 18;
    else
        check val = 50;
    i = 1;
    while(i<=9)//row check
        if(board[i] * board[i+1] * board[i+2] == check_val)
            if(board[i] == 2)
                return i;
            if(board[i+1] == 2)
                return i+1;
            if(board[i+2] == 2)
                return i+2;
        i+=3;
    }
```

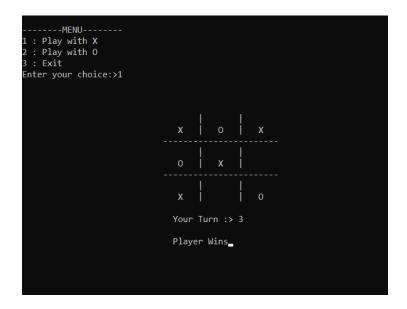
```
i = 1;
    while (i \le 3) / column \ check
        if(board[i] * board[i+3] * board[i+6] == check val)
            if(board[i] == 2)
                return i;
            if(board[i+3] == 2)
                return i+3;
            if(board[i+6] == 2)
                return i+6;
        }
        i++;
    }
    if(board[1] * board[5] * board[9] == check val)
        if(board[1] == 2)
            return 1;
        if(board[5] == 2)
            return 5;
        if(board[9] == 2)
            return 9;
    }
    if(board[3] * board[5] * board[7] == check val)
        if(board[3] == 2)
            return 3;
        if(board[5] == 2)
            return 5;
        if(board[7] == 2)
            return 7;
    return 0;
}
void go(int n)
    if(turn % 2)
        board[n] = 3;
        board[n] = 5;
    turn++;
}
void player first()
{
    int pos;
    check_draw();
    draw board();
    gotoxy(30,18);
    printf("Your Turn :> ");
    scanf("%d", &pos);
    if(board[pos] != 2)
        player_first();
    if(pos == posswin(player))
    {
```

```
go(pos);
        draw board();
        gotoxy(30,20);
        //textcolor(128+RED);
        printf("Player Wins");
        getch();
        exit(0);
    }
    go(pos);
    draw board();
    start_game();
void start game()
// p==1 then X p==0 then O
    if(posswin(comp))
        go(posswin(comp));
        flag = 1;
    else if(posswin(player))
        go(posswin(player));
    else if(make2())
        go(make2());
    else
        go(make4());
    draw board();
    if(flag)
        gotoxy(30,20);
        //textcolor(128+RED);
        printf("Computer wins");
        getch();
    else
        player_first();
}
void check draw()
    if(turn > 9)
        gotoxy(30,20);
        //textcolor(128+RED);
        printf("Game Draw");
        getch();
        exit(0);
}
void draw board()
    int j;
    for(j=9; j<17; j++)
        gotoxy(35,j);
                        |");
        printf("|
```

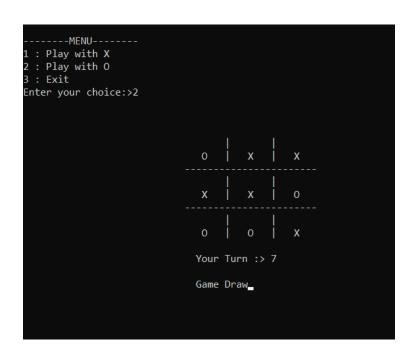
```
}
   gotoxy(28,11);
   printf("----");
   gotoxy(28,14);
   printf("----");
   for(j=1; j<10; j++)
       if(board[j] == 3)
          put_X_O('X',j);
       else if (board[j] == 5)
          put_X_O('O',j);
   }
void put X O(char ch,int pos)
   int m;
   int x = 31, y = 10;
   m = pos;
   if(m > 3)
       while (m > 3)
           y += 3;
          m = 3;
   if(pos % 3 == 0)
      x += 16;
   else
      pos %= 3;
      pos--;
       while(pos)
          x+=8;
          pos--;
   gotoxy(x,y);
   printf("%c",ch);
}
```

## **Game output Screenshots:**

## 1) PLAYER WINS



## 2) GAME DRAW



# 3) COMPUTER WINS

