

Script started on 2020-12-17 17:47:44-0600

```
m_sadaf1@ares:~$ pwd
/home/students/m_sadaf1
m_sadaf1@ares:~$ cat tictactoe.info
Name: Madiha Sadaf
Class: CSC121 W01
```

Lab: Cat's, X's, or O's?

Level: 3.5

Description:

This is a Tic-Tac-Toe program!

```
m_sadaf1@ares:~$ cat tictactoe.tpq
Thought Provoking Questions:
```

1) Array.

2) By using 2D arrays. The [] always starts its position from 0. So for example, position 1 in a 2D array would be coded as [0][0] and position 9 as [2][2].

3) The winner is determined if the player manages to takeover the whole row of numbers either horizontally, vertically, or diagonally. Loops are not required if the programmer uses if else branch. Just set the bool to true and do the equality test in the program to look and mark the number that the player number chooses as either X or O.

```
m_sadaf1@ares:~$ cat tictactoe.cpp
```

```
#include <iostream>
#include <limits>
#include <array>
#include <cctype>
```

```
using namespace std;
```

```
const size_t side_length = 3;
```

```
short valid_play(array<array<char, side_length>,
                 side_length> & board, short choice, char pos);
short win(const array<array<char, side_length>,
          side_length> & board);
void print(const array<array<char, side_length>,
          side_length> & board);
```

```
int main()
{
    array<array<char, side_length>, side_length> board
        ={{{ '1', '2', '3'},
            {'4', '5', '6'},
            {'7', '8', '9'}}};

    short player = 1, state, choice;

    /*char pos;*/

    do
    {
        print(board);
        player = (player % 2 != 0) ? 1 : 2;
        cout<< "\nPlayer "<< player << ", enter a number: ";
        cin>> choice;
        choice--;

        /*pos = (player == 1) ? 'X' : 'O';*/
        player = static_cast<short>(player + valid_play
                                   (board, choice, player == 1 ? 'X' : 'O'));

        state = win(board);

        player++;
    }while(state == -1);
    print(board);
    if(state == 1)
        cout<< "Player "<<--player<< " win ";
    else
        cout<< "Draw\n";

    return 0;
}

short valid_play(array<array<char, side_length>,
                 side_length> & board, short choice, char pos)
{
    bool valid;
    short choice_x, choice_y;
    choice_x = static_cast<short>(choice / side_length);
    choice_y = static_cast<short>(choice % side_length);

    if (isdigit(board[choice_x][choice_y]))
    {
        board[choice_x][choice_y] = pos;
        valid = true;
    }
}
```

```

else
{
    cout<< "Invalid move, please try again .";

    /*player--;*/
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
    valid = false;
}
return valid ? 0 : -1;
}

short win(const array<array<char, side_length>,
side_length> & board)
{
    short ans;
    if (board[0][0] == board[0][1] && board[0][1] == board[0][2])
        ans = 1;

    else if (board[1][0] == board[1][1] && board[1][1] == board[1][2])
        ans = 1;

    else if (board[2][0] == board[2][1] && board[2][1] == board[2][2])
        ans = 1;

    else if (board[0][0] == board[1][0] && board[1][0] == board[2][0])
        ans = 1;

    else if (board[0][1] == board[1][1] && board[1][1] == board[2][1])
        ans = 1;

    else if (board[0][2] == board[1][2] && board[1][2] == board[2][2])
        ans = 1;

    else if (board[0][0] == board[1][1] && board[1][1] == board[2][2])
        ans = 1;

    else if (board[0][2] == board[1][1] && board[1][1] == board[2][0])
        ans = 1;

    else if (board[0][0] != '1' && board[0][1] != '2'
        && board[0][2] != '3' && board[1][0] != '4'
        && board[1][1] != '5' && board[1][2] != '6'
        && board[2][0] != '7' && board[2][1] != '8'
        && board[2][2] != '9')

        ans = 0;

    else

```

```

        ans = -1;

    return ans;
}

void print(const array<array<char, side_length>,
side_length> & board)
{
    cout<< "\n\n\tTic Tac Toe\n\n";

    cout<< "Player 1 (X) - Player 2 (O)" << endl << endl;
    cout<< endl;

    cout<< " " " "<<board[0][0] << " |" " "<< board[0][1] <<
        " |" " "<< board[0][2] << endl;

    cout<< "____+____+____" << endl;

    cout<< " " " "<< board[1][0] << " |" " "<< board[1][1] <<
        " |" " "<< board[1][2] << endl;

    cout<< "____+____+____" << endl;

    cout<< " " " "<< board[2][0] << " |" " "<<board[2][1] <<
        " |" " "<< board[2][2] << endl;

}

```

```

m_sadaf1@ares:~$ CPP tictactoe
tictactoe.cpp***

```

```

m_sadaf1@ares:~$ ./tictactoe.out

```

Tic Tac Toe

Player 1 (X) - Player 2 (O)

1		2		3
<hr/>				
4		5		6
<hr/>				
7		8		9

Player 1, enter a number: 1

Tic Tac Toe

Player 1 (X) - Player 2 (0)

X		2		3
<hr/>				
4		5		6
<hr/>				
7		8		9

Player 2, enter a number: 4

Tic Tac Toe

Player 1 (X) - Player 2 (0)

X		2		3
<hr/>				
0		5		6
<hr/>				
7		8		9

Player 1, enter a number: 2

Tic Tac Toe

Player 1 (X) - Player 2 (0)

X		X		3
<hr/>				
0		5		6
<hr/>				
7		8		9

Player 2, enter a number: 5

Tic Tac Toe

Player 1 (X) - Player 2 (0)

X		X		3
<hr/>				
0		0		6
<hr/>				
7		8		9

Player 1, enter a number: 3

Tic Tac Toe

Player 1 (X) - Player 2 (0)

X		X		X
<hr/>				
0		0		6
<hr/>				
7		8		9

Player 1 win m_sadaf1@ares:~\$./tictactoe.out

Tic Tac Toe

Player 1 (X) - Player 2 (0)

1		2		3
<hr/>				
4		5		6
<hr/>				
7		8		9

Player 1, enter a number: 6

Tic Tac Toe

Player 1 (X) - Player 2 (0)

1		2		3
<hr/>				
4		5		X
<hr/>				
7		8		9

Player 2, enter a number: 1

Tic Tac Toe

Player 1 (X) - Player 2 (0)

0		2		3
<hr/>				
4		5		X
<hr/>				
7		8		9

Player 1, enter a number: 2

Tic Tac Toe

Player 1 (X) - Player 2 (0)

0		X		3
<hr/>				
4		5		X
<hr/>				
7		8		9

Player 2, enter a number: 5

Tic Tac Toe

Player 1 (X) - Player 2 (0)

0		X		3
<hr/>				
4		0		X
<hr/>				
7		8		9

Player 1, enter a number: 3

Tic Tac Toe

Player 1 (X) - Player 2 (0)

0		X		X
<hr/>				
4		0		X
<hr/>				
7		8		9

Player 2, enter a number: 9

Tic Tac Toe

Player 1 (X) - Player 2 (0)

0		X		X
<hr/>				
4		0		X
<hr/>				
7		8		0

Player 2 win m_sadaf1@ares:~\$ exit
exit

Script done on 2020-12-17 17:49:26-0600