Tic Tac Toe

Introduction

- This file helps to build your programming logic.
- This file represents building a game by C++ programming language i.e is Tic Tac Toe.
- Idea behind the project is not only for enjoying purpose; this documentation also clears your all the basic knowledge regarding how to use Functions, Arrays, Conditional statements, Looping statements etc. in C++.
- And the purpose of this project file is that too proving i.e how the programming is very interesting and helping to build our logical strength.
- Code written in Visual Studio (VS - CODE).
- Full Game code is provided at the end of the documentation.

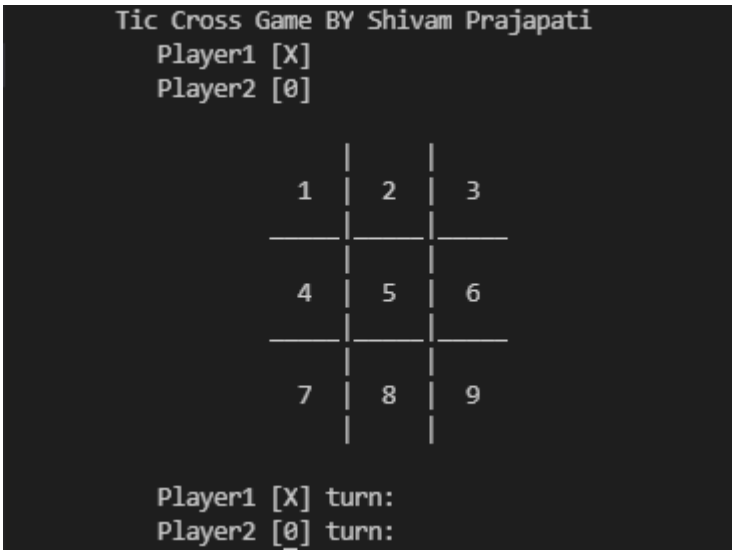
Steps for building the game

Step 1 : How to Make the Game interface.

Source code:

```
#include <iostream>
using namespace std;

int main()
{

    cout<<"     Tic Cross Game BY Shivam Prajapati";
    cout << "\n\tPlayer1 [X] \n\tPlayer2 [0]\n\n";




    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  1  |  2  |  3  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  4  |  5  |  6  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  7  |  8  |  9  \n";
    cout<<"\t\t     |     |     \n";

    cout<<"\n\tPlayer1 [X] turn: ";
    cout<<"\n\tPlayer2 [0] turn: ";


    return 0;
}
```

Output:


Tic Cross Game BY Shivam Prajapati
Player1 [X]
Player2 [0]
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
Player1 [X] turn:
Player2 [0] turn:

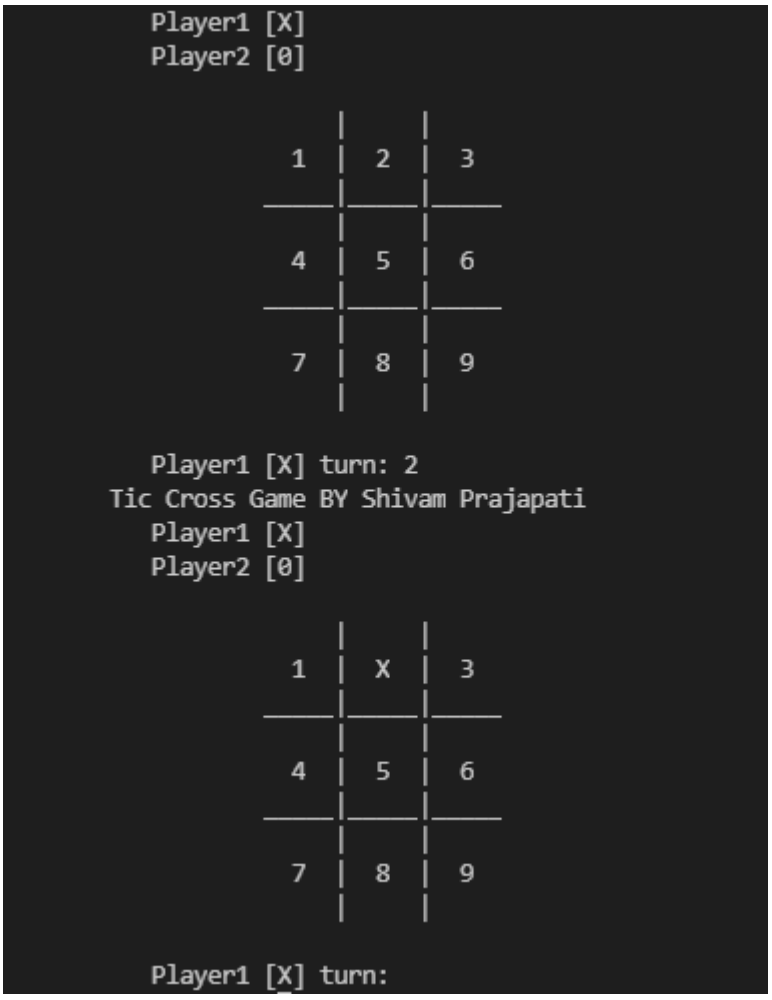
Step 2 : Using of 2 Dimensional Array in the program to use multiple variables of the same data type in Board.

Source Code:

```
#include <iostream>
using namespace std;

int main()
{
    int choice;
    char board[3][3] = {{'1','2','3'},{'4','5','6'},{'7','8','9'}};

    cout<<"     Tic Cross Game BY Shivam Prajapati";
    cout << "\n\tPlayer1 [X] \n\tPlayer2 [0]\n\n";




    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[0][0]<<"  |  "<<board[0][1]<<"  |
"<<board[0][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
     cout<<"\t\t  "<<board[1][0]<<"  |  "<<board[1][1]<<"  |
"<<board[1][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[2][0]<<"  |  "<<board[2][1]<<"  |
"<<board[2][2]<<"  \n";
    cout<<"\t\t     |     |     \n";

    cout<<"\n\tPlayer1 [X] turn: ";


    cin>>choice;

    switch(choice){
        case 1:
             board[0][0] = 'X';
             break;
             case 2:
```

```
            board[0][1] = 'X';
            break;
            case 3:
            board[0][3] = 'X';
            break;

            default:
                cout<<"Invalid choice"<<endl;
                break;


    }
    cout<<"     Tic Cross Game BY Shivam Prajapati";
    cout << "\n\tPlayer1 [X] \n\tPlayer2 [0]\n\n";




    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[0][0]<<"  |  "<<board[0][1]<<"  |
"<<board[0][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
     cout<<"\t\t  "<<board[1][0]<<"  |  "<<board[1][1]<<"  |
"<<board[1][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[2][0]<<"  |  "<<board[2][1]<<"  |
"<<board[2][2]<<"  \n";
    cout<<"\t\t     |     |     \n";

    cout<<"\n\tPlayer1 [X] turn: ";




    return 0;
}
```

Output:


Player1 [X]
Player2 [0]
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
Player1 [X] turn: 2
Tic Cross Game BY Shivam Prajapati
Player1 [X]
Player2 [0]
1 | X | 3
4 | 5 | 6
7 | 8 | 9
Player1 [X] turn:

Step 3: Use of Function in the Program.

Source Code:

```
#include <iostream>
using namespace std;

     char board[3][3] = {{'1','2','3'},{'4','5','6'},{'7','8','9'}};

void display_board(){

    cout<<"\n\n     Tic Cross Game BY Shivam Prajapati";
    cout << "\n\tPlayer 1 [X] \n\tPlayer2 [0]\n\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[0][0]<<"  |  "<<board[0][1]<<"  |
"<<board[0][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
     cout<<"\t\t  "<<board[1][0]<<"  |  "<<board[1][1]<<"  |
"<<board[1][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[2][0]<<"  |  "<<board[2][1]<<"  |
"<<board[2][2]<<"  \n";
    cout<<"\t\t     |     |     \n";

}

int main(){
    int choice;

    display_board();

    cout<<"\n\tPlayer1 [X] turn: ";
     cin>>choice;

    switch(choice){
        case 1:
             board[0][0] = 'X';
             break;
             case 2:
             board[0][1] = 'X';
             break;
             case 3:
             board[0][3] = 'X';
```

```
            break;

            default:
                cout<<"Invalid choice"<<endl;
                break;


    }
    display_board();

    return 0;
}
```

In the above program we are making the Function i.e “display_board()”.
With the help of this Function we can display boards at different - different places inside the code.
Therefore we are not required to write the code of the Board again and again for displaying the board.

Output:

Output is the same as the output of step 2.

Step 4: Again use of FUNCTION for Player turns.

Source Code:

```
#include <iostream>
using namespace std;

     char board[3][3] = {{'1','2','3'},{'4','5','6'},{'7','8','9'}};
     char turn = 'X';
     int row,column;

void display_board(){
    system("cls");
    cout<<"\n\n     Tic Cross Game BY Shivam Prajapati";
    cout << "\n\tPlayer1 [X] \n\tPlayer2 [O]\n\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[0][0]<<"  |  "<<board[0][1]<<"  |
"<<board[0][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
     cout<<"\t\t  "<<board[1][0]<<"  |  "<<board[1][1]<<"  |
"<<board[1][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[2][0]<<"  |  "<<board[2][1]<<"  |
"<<board[2][2]<<"  \n";
    cout<<"\t\t     |     |     \n";

}
void player_turn(){

    int choice;

    if(turn == 'X')
    cout<<"\n\tPlayer1 [X] turn: ";

    if(turn == 'O')
    cout<<"\n\tPlayer2 [O] turn: ";

    cin>>choice;

    switch(choice){
        case 1: row =0; column =0; break;
        case 2: row =0; column =1; break;
```

```
        case 3: row =0; column =2; break;
        case 4: row =1; column =0; break;
        case 5: row =1; column =1; break;
        case 6: row =1; column =2; break;
        case 7: row =2; column =0; break;
        case 8: row =2; column =1; break;
        case 9: row =2; column =2; break;

        default:
        cout<<"Invalid choice \n";
        break;
    }
    if(turn == 'X')
    {
        board[row][column] = 'X';
        turn = 'O';
    }
    else if(turn == 'O')
    {
        board[row][column] = 'O';
        turn = 'X';
    }
}
int main(){

    while(true)
{
    display_board();
    player_turn();
    display_board();
}




    return 0;
}
```

In the above program we are making again one function i.e “player_turn()” for player turns; Because this Game is 2 Players Game.
Then both players can play the game easily one after one and vice versa till the end of this game.

In the “display_board()” function we add one term i.e “ system(“cls”); ” for not repetition of the board.
Then all changes are modified into one board and the board updated continuously.

Step 5: Preventing the Overwriting of the data put by Players.

Condition Code:

```
if(turn == 'X' && board[row][column]!='X' && board[row][column]!='O' )
    {
        board[row][column] = 'X';
        turn = 'O';
    }
    else if(turn == 'O' && board[row][column]!='X' &&
board[row][column]!='O' )
    {
        board[row][column] = 'O';
        turn = 'X';
    }
    else{
        cout<<"Box already filled!\n Please try again!! \n\n";
        player_turn();
```

Output:

```
      Tic Cross Game BY Shivam Prajapati
        Player1 [X]
        Player2 [O]

                |     |
             1  |  X  |  3
           _____|_____|_____
                |     |
             4  |  5  |  6
           _____|_____|_____
                |     |
             7  |  8  |  9
                |     |

        Player2 [O] turn: 2
  Box already filled!
   Please try again!!


        Player2 [O] turn: 2
  Box already filled!
   Please try again!!


        Player2 [O] turn: 2
```

Step 6: Scenarios For Gameover.

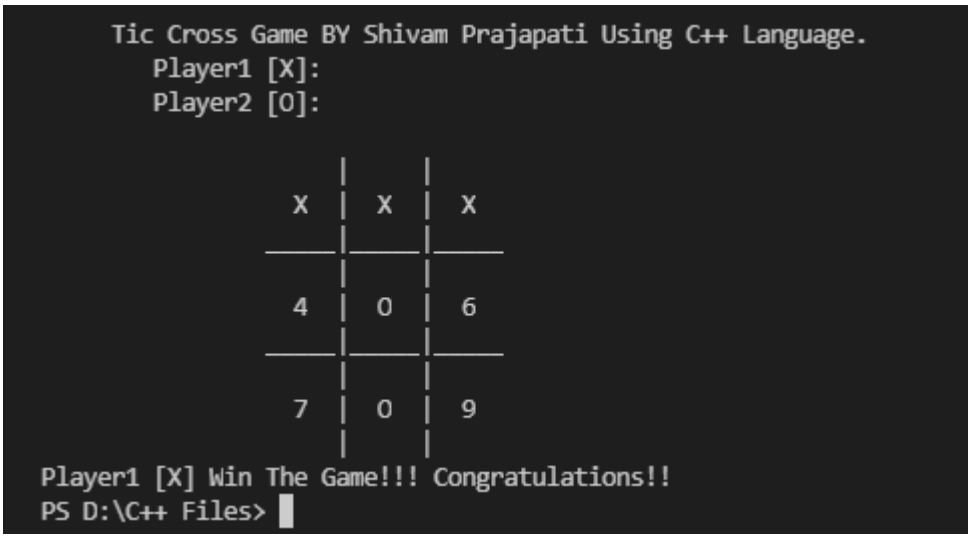
1st Scenario: Player [1] or Player [2] Wins.
2nd Scenario: Game is continued.
3rd Scenario: Game Draw.

For Game Scenarios we are again using the FUNCTION.
We make One Function i.e “bool gameover()”
In this Function with the help of Conditional Statement we make the condition for our Game Overing Scenario.

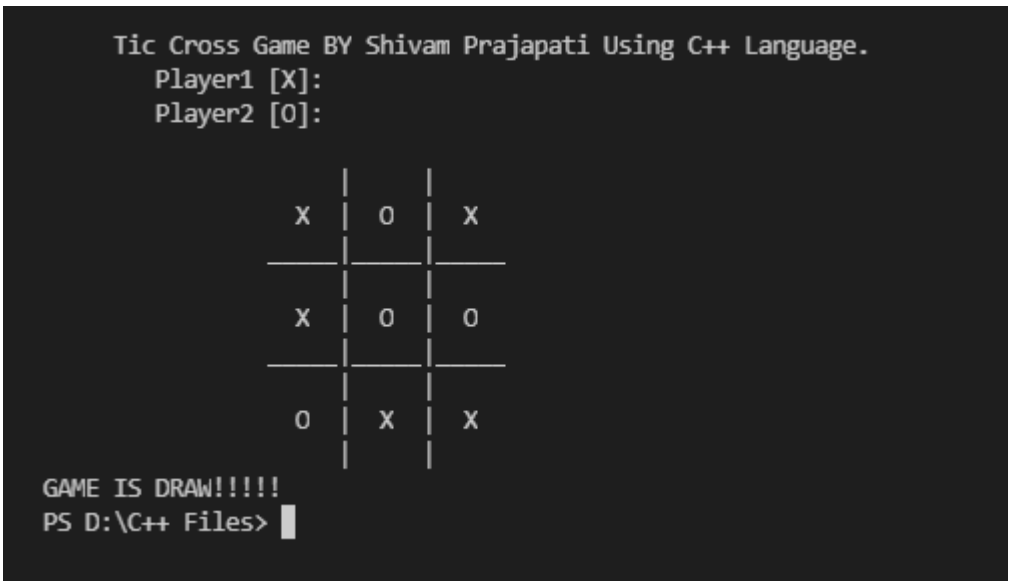
Source Code:

```
bool gameover(){
    //Check For Wining The Game.
    for(int i=0; i<3; i++)
    if (board[i][0] == board[i][1] && board[i][0] == board[i][2] ||
board[0][i] == board[1][i] && board[0][i] == board[2][i])
    return false;

    if (board[0][0] == board[1][1] && board[0][0] == board[2][2] ||
board[0][2] == board[1][1] && board[0][0] == board[2][0])
    return false;


    //If there is any box not filled then Game is now in playing mode.
    for(int i=0; i<3; i++)
    for(int j=0; j<3; j++)
    if (board[i][j] != 'X' && board[i][j] != 'O' )
    return true;

    //When the Game is Draw.
    draw = true;
    return false;

}
```

Output For 1st Scenario when one player win the game:



Output For 2nd Scenario when game is draw:



FULL GAME CODE:

```
#include <iostream>
using namespace std;

     char board[3][3] = {{'1','2','3'},{'4','5','6'},{'7','8','9'}};
     char turn = 'X';
     int row,column;
     bool draw = false;

void display_board(){
    system("cls");
    cout<<"\n\n     Tic Cross Game BY Shivam Prajapati Using C++
Language.";
    cout << "\n\tPlayer1 [X]: \n\tPlayer2 [O]:\n\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[0][0]<<"  |  "<<board[0][1]<<"  |
"<<board[0][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
     cout<<"\t\t  "<<board[1][0]<<"  |  "<<board[1][1]<<"  |
"<<board[1][2]<<"  \n";
    cout<<"\t\t_____|_____|_____\n";
    cout<<"\t\t     |     |     \n";
    cout<<"\t\t  "<<board[2][0]<<"  |  "<<board[2][1]<<"  |
"<<board[2][2]<<"  \n";
    cout<<"\t\t     |     |     \n";

}
void player_turn(){

    int choice;

    if(turn == 'X')
    cout<<"\n\tPlayer1 [X] turn: ";

    if(turn == 'O')
    cout<<"\n\tPlayer2 [O] turn: ";

    cin>>choice;

    switch(choice){
        case 1: row =0; column =0; break;
        case 2: row =0; column =1; break;
```

```
        case 3: row =0; column =2; break;
        case 4: row =1; column =0; break;
        case 5: row =1; column =1; break;
        case 6: row =1; column =2; break;
        case 7: row =2; column =0; break;
        case 8: row =2; column =1; break;
        case 9: row =2; column =2; break;

        default:
        cout<<"Invalid choice \n";
        break;
    }
    if(turn == 'X' && board[row][column]!='X' &&
board[row][column]!='O' )
    {
        board[row][column] = 'X';
        turn = 'O';
    }
    else if(turn == 'O' && board[row][column]!='X' &&
board[row][column]!='O' )
    {
        board[row][column] = 'O';
        turn = 'X';
    }
    else{
        cout<<"Box already filled!\n Please try again!! \n\n";
        player_turn();
    }
    display_board();
}
bool gameover(){
    //Check For Wining The Game.
    for(int i=0; i<3; i++)
    if (board[i][0] == board[i][1] && board[i][0] == board[i][2] ||
board[0][i] == board[1][i] && board[0][i] == board[2][i])
    return false;

    if (board[0][0] == board[1][1] && board[0][0] == board[2][2] ||
board[0][2] == board[1][1] && board[0][0] == board[2][0])
    return false;

    if (board[2][2] == board[1][1] && board[2][2] == board[0][0] ||
board[2][0] == board[1][1] && board[2][0] == board[0][2])
```

```
    return false;


    //If there is any box not filled then Game is now in playing mode.
    for(int i=0; i<3; i++)
    for(int j=0; j<3; j++)
    if (board[i][j] != 'X' && board[i][j] != 'O' )
    return true;

    //When the Game is Draw.
    draw = true;
    return false;

}
int main(){

    while(gameover())
{
    display_board();
    player_turn();
    gameover();
}

if(turn =='X'&& draw == false)
cout<<"Player2 [O] Win The Game!!! Congratulations!!\n";
else if(turn =='O' && draw == false)
cout<<"Player1 [X] Win The Game!!! Congratulations!!\n";
else
cout<<"GAME IS DRAW!!!!! \n";

    return 0;
}
```

“NOW ENJOY THE GAME”

THANK YOU.