******

***School of Mechanical & Manufacturing Engineering (SMME),***

***National University of Science and Technology (NUST),***

***Sector H-12, Islamabad***

Program: BE-Aerospace Section: AE-01

Session: Fall 2023 Semester: 1st

Course Title: Fundamentals of Programming (CS-109)

Project

**“TIC TAC TOE”**

***Name: Muhammad Arriyaan Ali***

***CMS: 456787***

Description:

This program first initializes the board with ‘-’ to represent empty spaces. It then randomly places ‘X’ or ‘O’ in an empty space until the board is full. The *PrintBoard* function is used to display the current state of the board. Please note that this program uses *rand()* function to generate random numbers, so don’t forget to include *<cstdlib>* and *<ctime>* headers.

The program does not take user input but uses random choice to initiate a placement of cross or zero. Morever, *srand* is primarily used so that every time a different random sequence is generated.

Some features of the program includes it s ability to stop placing new X/O values when a win condition is met. Not only that but a draw, being a high probability in TIC TAC TOE is also stated if occurred.

The isFull function is used to check if the Tic Tac Toe board is full or not. In the context of the provided C++ program, the game continues until all the cells of the board are filled with either ‘X’ or ‘O’. Here’s how it works:

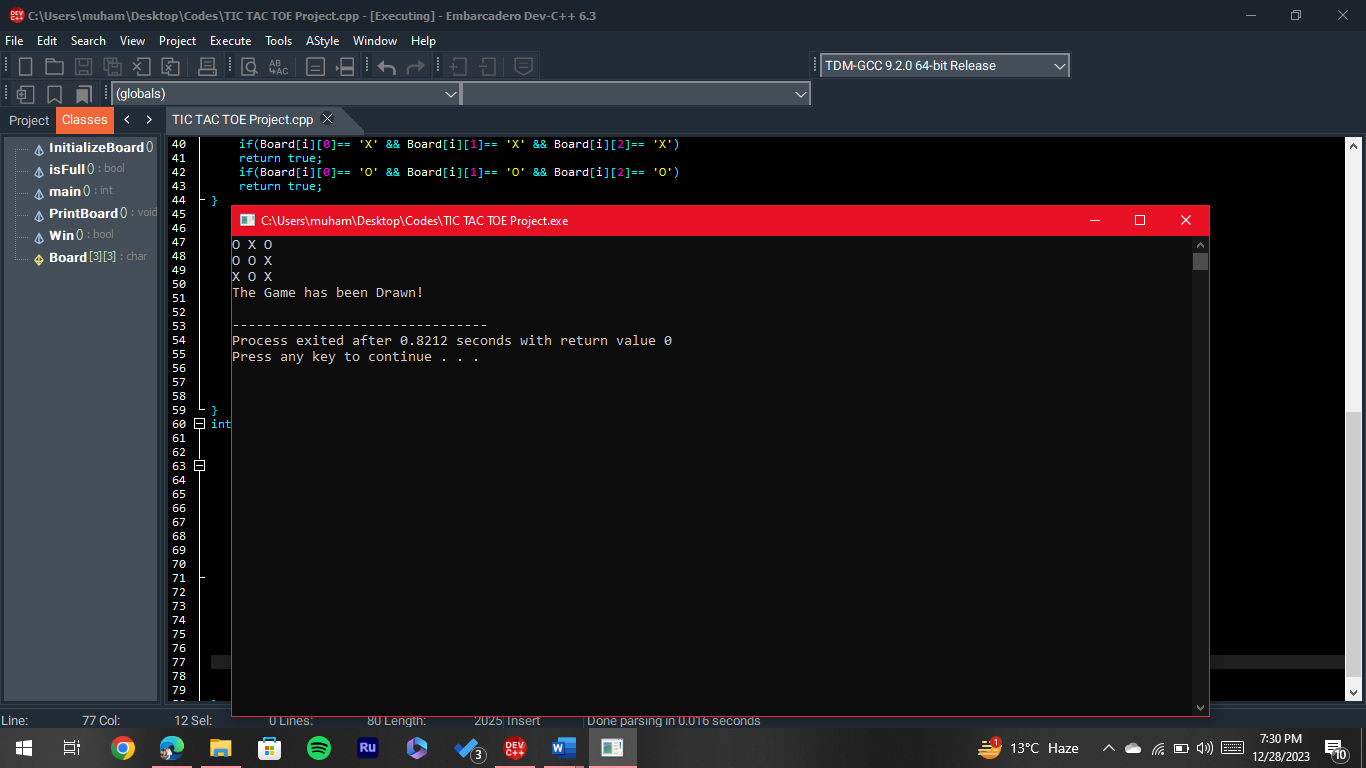
* The function iterates over each cell in the 2D array (the board).
* If it finds a cell that is still marked with ‘-’, which represents an empty space, it returns false, indicating that the board is not yet full.
* If no empty cells are found after checking the entire board, it returns true, indicating that the board is full.

## **Basic Operations:**

These operations were performed:

* **Defining/Calling Functions**
* **Global Variable Declaration**
* **Loops**
* **Random Function**
* **Boolean Expression**
* **Selection Statements (If/Else)**
* **Declaration of Multidimensional Array**

## **Outputs:**



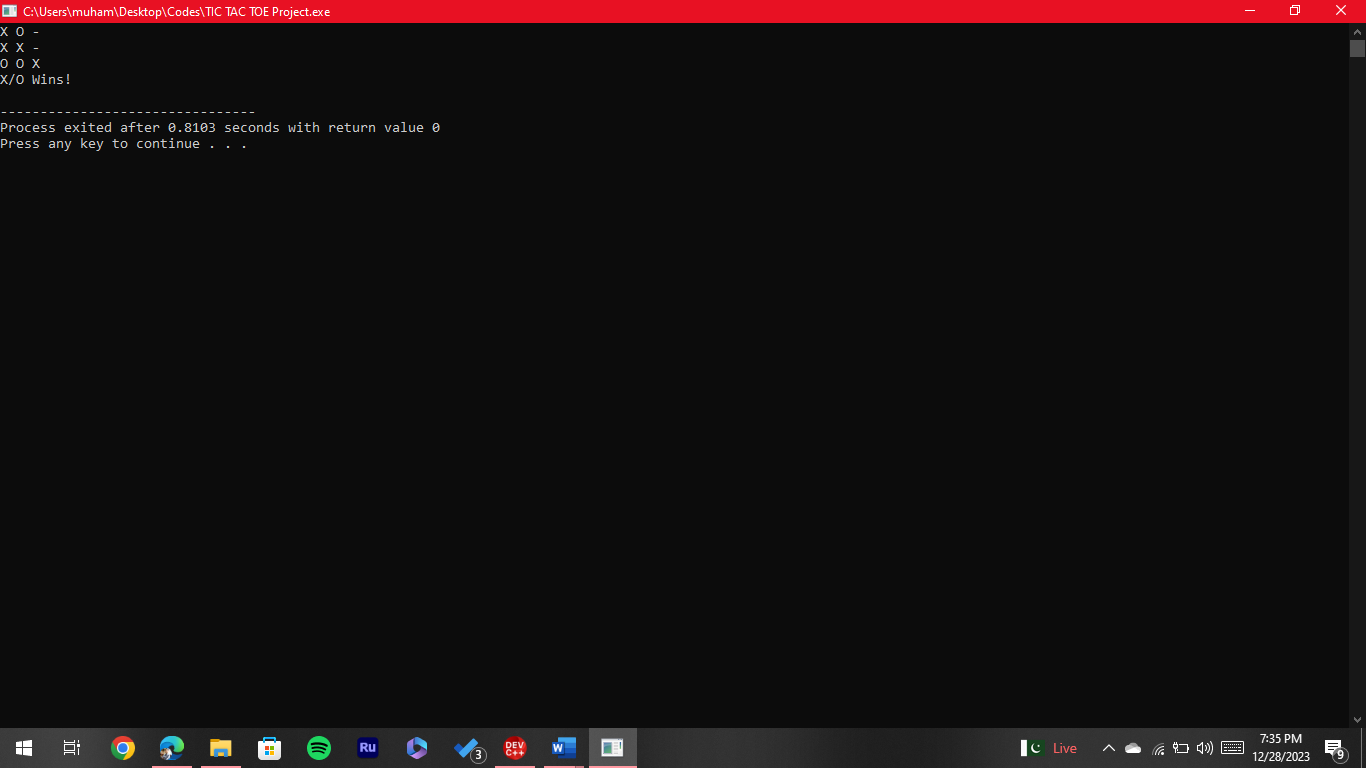


Figure 1: Game Ends Drawn

Figure 2: Game Ends in Win

## **Code:**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

char Board[3][3];

void InitializeBoard() {

for(int i=0; i<3; i++)

for(int j=0; j<3; j++)

Board[i][j] = '-';

}

void PrintBoard() { //Printing the board of Tic Tac Toe

for(int i=0; i<3; i++){

for(int j=0; j<3; j++){

cout<<Board[i][j]<< " ";

}

cout<<endl;

}

}

bool isFull() { //To check when the board is fully initialized

for(int i=0; i<3; i++)

for(int j=0; j<3; j++)

if(Board[i][j] == '-')

return false;

return true;

}

bool Win() {

//For Rows Win

for(int j=0; j<3; j++){

if(Board[0][j]== 'X' && Board[1][j]== 'X' && Board[2][j]== 'X')

return true;

if(Board[0][j]== 'O' && Board[1][j]== 'O' && Board[2][j]== 'O')

return true;

}

//For Coloumn Win

for(int i=0; i<3; i++){

if(Board[i][0]== 'X' && Board[i][1]== 'X' && Board[i][2]== 'X')

return true;

if(Board[i][0]== 'O' && Board[i][1]== 'O' && Board[i][2]== 'O')

return true;

}

//For Diagonal Win

int n=0; //Difference of Rows and Coloums

int m=2; //Number of Rows and Coloums (starting from 0-2)

if(Board[n][n]== 'X' && Board[n+1][n+1]== 'X' && Board[n+2][n+2]== 'X')

return true;

if(Board[n][n]== 'O' && Board[n+1][n+1]== 'O' && Board[n+2][n+2]== 'O')

return true;

if(Board[n][m]== 'X' && Board[n+1][m-1]== 'X' && Board[n+2][m-2]== 'X')

return true;

if(Board[n][m]== 'O' && Board[n+1][m-1]== 'O' && Board[n+2][m-2]== 'O')

return true;

return false;

}

int main() {

srand(time (0));

InitializeBoard();

while(!isFull()){

int i = rand() % 3;

int j = rand() % 3;

Win();

if(Board[i][j] == '-')

Board[i][j] = rand () % 2 == 0 ? 'X' : 'O';

if(!(!Win()))

break;

}

PrintBoard();

Win();

if(!Win())

cout<<endl<<"The Game has been Drawn!"<<endl;

if(!(!Win()))

cout<<endl<<"X/O Wins!"<<endl;

return 0;

}