A Project Report

On

Tic Tac Toe Game

For The Course

“Software Development Project-I”

By

Md Kabir Hosen

ID:it-23041

Professor

Dr. Ziaur Rahman

Department of ICT, MBSTU.

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902, Dhaka, Bangladesh

Declaration

This is to certify that the work presented in this project is carried out by the candidate under the supervision of **Dr. Ziaur Rahman** in the department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature of Supervisor
Dr. Ziaur Rahman
Professor
Dept. of ICT, MBSTU

Acknowledgements

I must sense grateful to the Almighty Allah to complete the dissertation. At the outset, I would like to express gratitude to my supervisor Dr. Ziaur Rahman, Professor, Dept. of Information and Communication Technology, MBSTU who has supported our plan to continue Software Development Project-I. I also like to express gratitude to our supervisor, for his valuable guidance and insight, encouragement, support and reliance throughout the project. However, it is not possible acknowledge properly the effort of our honorable teacher in writing words. We are, as always, indebted to our family. The love and support of our parents remain bedrock of our life.

Contents

Page No.

Chapter – 1

Introduction of Project

Tic Tac Toe is a popular game played on a 3x3 grid. The objective of the game is to get three of your own marks (either X or O) in a row, either horizontally, vertically, or diagonally. In this project, we will be implementing a Tic Tac Toe game using the C programming language.

The game will be played between two players, where each player takes turns placing their mark on the grid. The players can choose whether they want to be X or O. The game will continue until one player wins or the grid is completely filled without a winner, resulting in a draw.

The source code is not that long, over 122 lines. It is compiled in Code::Blocks IDE with GCC compiler.

To help you understand the calendar project better, there are lots of comments within the source code. This project is aimed to show you

Chapter 2

2.1 - Programming Language

Programming Language : As we know, to communicate with a person, we need a specific language, similarly to communicate with computers, programmers also need a language is called Programming language.

Before learning the programming language, let's understand what is language?

What is Language?

Language is a mode of communication that is used to share ideas, opinions with each other. For example, if we want to teach someone, we need a language that is understandable by both communicators.

What is a Programming Language?

A programming language is a computer language that is used by programmers (developers) to communicate with computers. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.

A programming language is mainly used to develop desktop applications, websites, and mobile applications.

Types of programming language:

1. Low-level programming language:

Low-level language is machine-dependent (0s and 1s) programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast.

Low-level language is further divided into two parts -

i. Machine Language:

Machine language is a type of low-level programming language. It is also called as machine code or object code. Machine language is easier to read because it is normally displayed in binary or hexadecimal form (base 16) form. It does not require a translator to convert the programs because computers directly understand the machine language programs.

The advantage of machine language is that it helps the programmer to execute the programs faster than the high-level programming language.

ii. Assembly Language:

Assembly language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a symbolic

and human-understandable form. It uses an assembler to convert the assembly language to machine language.
The advantage of assembly language is that it requires less memory and less execution time to execute a program.

2. High-level programming language:

High-level programming language (HLL) is designed for developing user-friendly software programs and websites. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program).
The main advantage of a high-level language is that it is easy to read, write, and maintain.
High-level programming language includes Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift programming language.

A high-level language is further divided into three parts -

i. Procedural Oriented programming language:

Procedural Oriented Programming (POP) language is derived from structured programming and based upon the procedure call concept. It divides a program into small procedures called routines or functions.

Procedural Oriented programming language is used by a software programmer to create a program that can be accomplished by using a programming editor like IDE, Adobe Dreamweaver, or Microsoft Visual Studio.

The advantage of POP language is that it helps programmers to easily track the program flow and code can be reused in different parts of the program.

Example: C, FORTRAN, Basic, Pascal, etc.

03.Middle-level programming language:

Middle-level programming language lies between the low-level programming language and high-level programming language. It is also known as the intermediate programming language and pseudo-language.
A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.
Example: C, C++, language

2.2 - C Programming Language

C is an imperative procedural language, supporting structured programming, lexical variable scope, and recursion, with a static type system. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support.

Most Important Features of C Language:

- Simple and Efficient.
- Fast.
- Portability.
- Extensibility.
- Function-rich libraries.
- Dynamic Memory Management.
- Modularity with Structured Language.
- Mid-level Programming Language

Advantages:

- It is easy to understand. One of the main reasons why people choose C over other programming languages is its simplicity. ...
- Presence of many Libraries. ...
- Easy to write. ...
- Low cost. ...
- Fast execution speed. ...
- Portable. ...
- Easy debugging. ...
- Procedure Oriented Language.

Disadvantages:

- Lack of Object Orientation. ...
- Inefficient Memory Management. ...
- No Garbage Collection. ...
- Run-time checking. ...
- Concept of namespace is not present in C. ...
- Absence of Exception Handling. ...
- Lacks Constructor and Destructor.

2.3 IDE details

Code::Blocks:

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Features

Compilers

Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including Fortran and D. A plug-in system is included to support other programming languages.

Code editor

The IDE features syntax highlighting and code folding (through its Scintilla editor component), C++ code completion, class browser, a hex editor and many other utilities. Opened files are organized into tabs. The code editor supports font and font size selection and personalized syntax highlighting colors.

Debugger

The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

Chapter – 3

3.1 – Code Header File:

- #include<stdio.h>
- #include<conio.h>

#include<stdio.h>:

In C, stdio. h header file is a component of the C Standard Library. It provides functions and macros for input and output operations, such as displaying output on the screen and reading input from the user or files.

#include<conio.h>:

#include<conio. h> It is a header file used in c and cpp and it includes inbuilt functions like getch() and clrscr(). It stand for console input ouput i.e. it takes input from keyboard and displays it on screen. Common in built function like getch() clrscr() stored in <conio.

4.1 - Function Name & Details

checkWin(): (This function is used to check which player is the winner**)**

printBoard(): (This function is used to draw the game board)

Chapter – 5

5.1 – Source Code

Source Code:

```
#include <stdio.h>
#include <conio.h>

void printBoard();
int checkWin();
void system();

char board[]={'0','1','2','3','4','5','6','7','8','9'};

void main(){
   int player=1,input,status=-1;
   printBoard();


   while (status==-1)
   {
      player=(player%2==0) ? 2 : 1;
```

```
    char mark=(player==1) ? 'X' :'O';
    printf("Please enter Number For Player %d\n",player);
    scanf("%d",&input);
if(input<1 || input>9){
    printf("invalid input");
}

board[input]=mark;
printBoard();

int result=checkWin();

if(result==1){
    printf("Player %d is the Winner",player);
    return;
}else if(result==0){
    printf("draw");
    return;
}
```

```
    player++;
    }


}

void printBoard(){
    system("cls");
    printf("\n\n");
    printf("=== TIC TAC TOE ===\n\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c  \n",board[1],board[2],board[3]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c  \n",board[4],board[5],board[6]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c  \n",board[7],board[8],board[9]);
    printf("     |     |     \n");
    printf("\n\n");
```

```
}


int checkWin(){

    if(board[1]==board[2] && board[2]==board[3]){
        return 1;
    }
    if(board[1]==board[4] && board[4]==board[7]){
        return 1;
    }
    if(board[7]==board[8] && board[8]==board[9]){
        return 1;
    }
    if(board[3]==board[6] && board[6]==board[9]){
        return 1;
    }
    if(board[1]==board[5] && board[5]==board[9]){
        return 1;
    }
```

```
if(board[3]==board[5] && board[5]==board[7]){
   return 1;
}
if(board[2]==board[5] && board[5]==board[8]){
   return 1;
}
if(board[4]==board[5] && board[5]==board[6]){
   return 1;
}

int count=0;
for (int i = 1; i <=9; i++)
{
   if(board[i]=='X' ||  board[i]=='O'){
      count++;
   }
}

if(count==9){
   return 0;
```

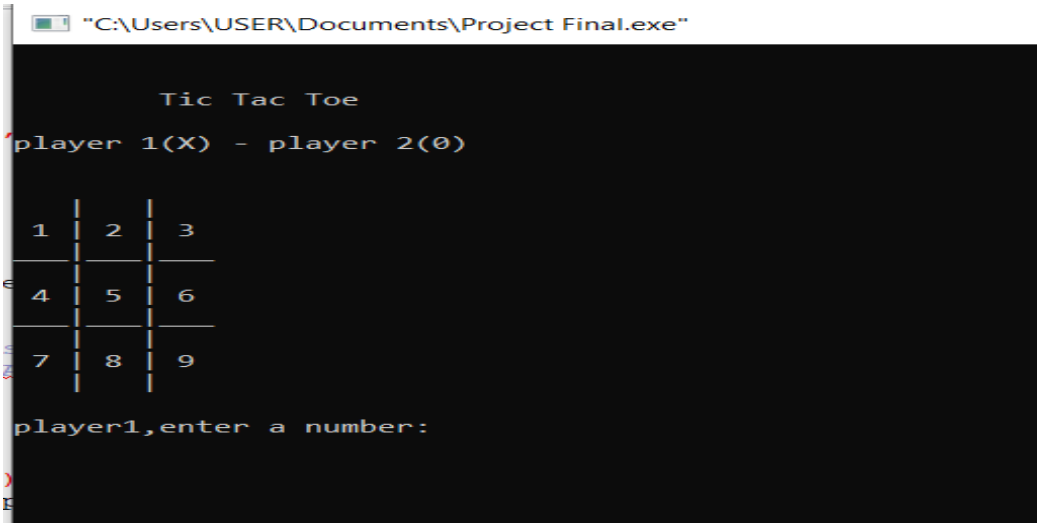
```
    }
    return -1;
}
```

Chapter – 6

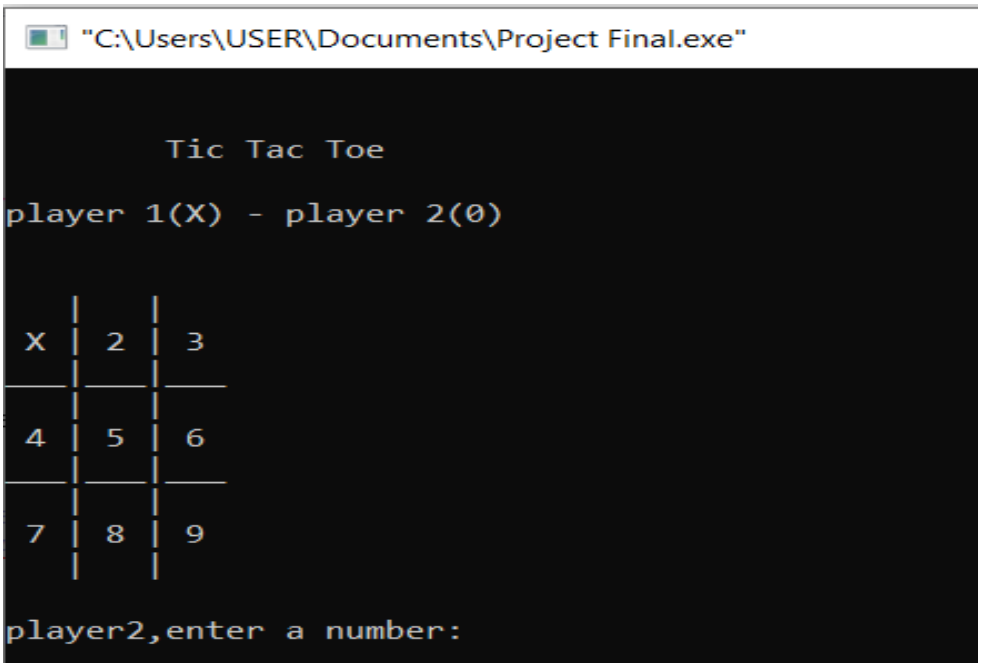
6.1 - Output

Output:

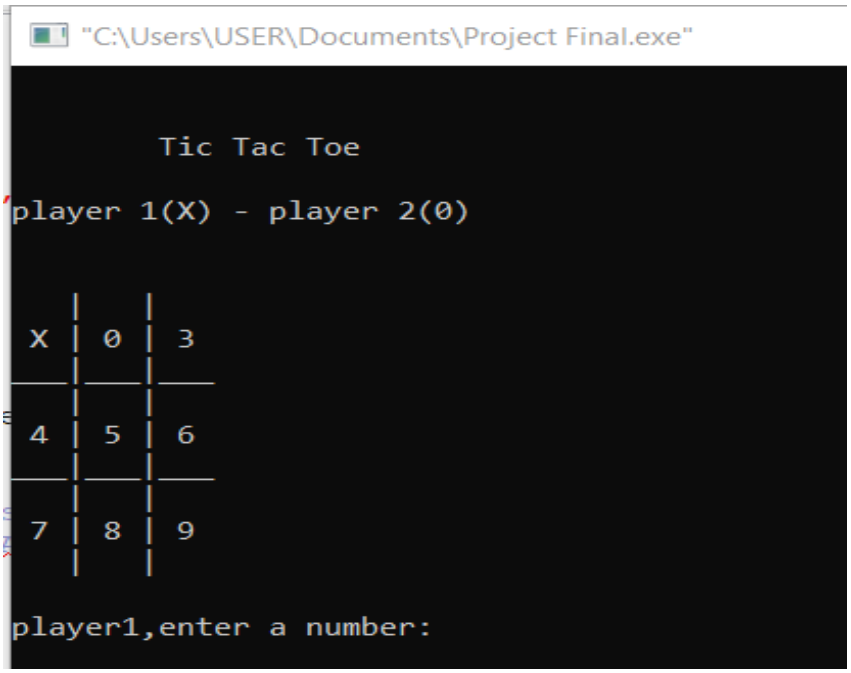
1.The main tic tac toe board:



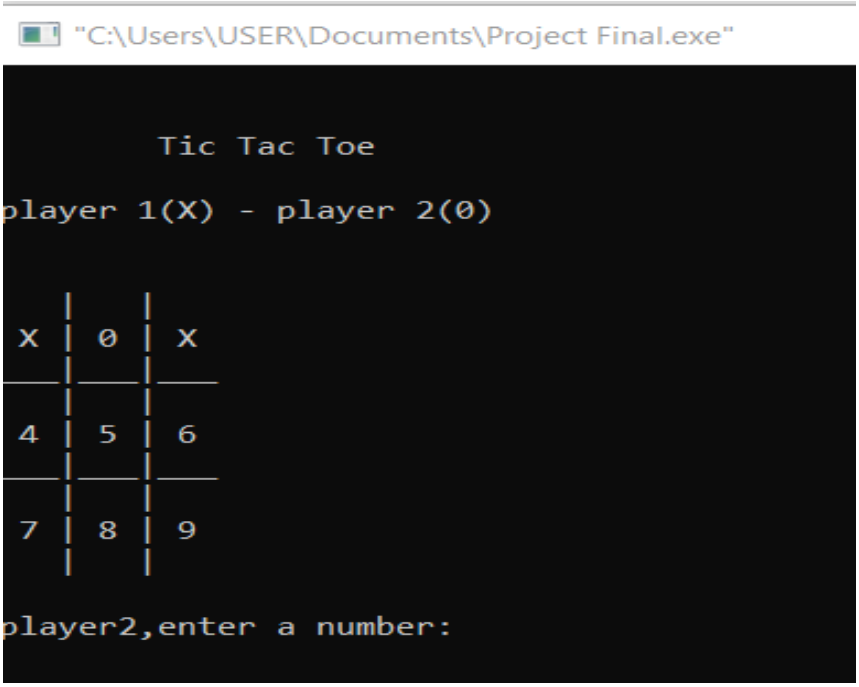
2.First player played for the first time:



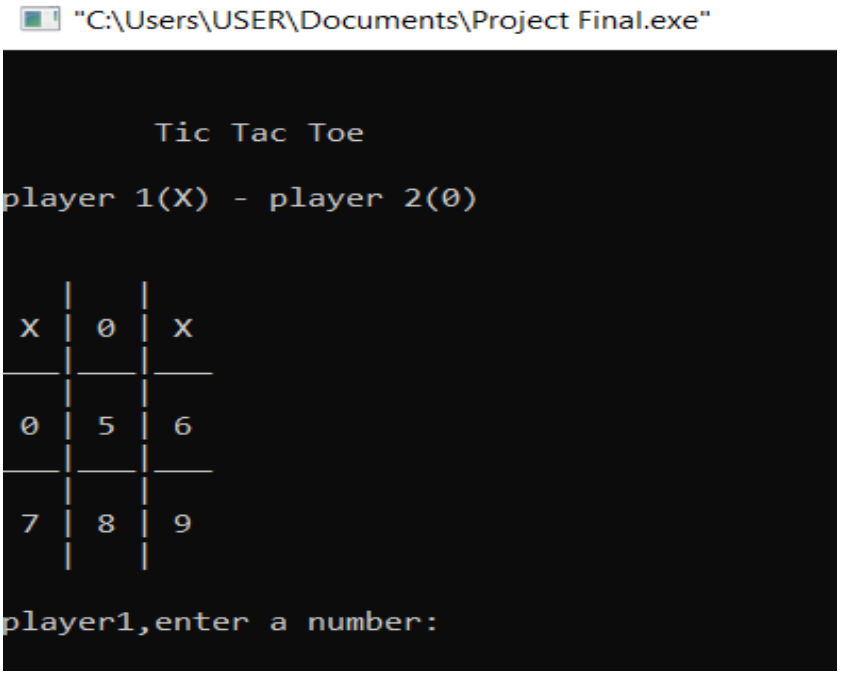
3. Second player played for the first time:



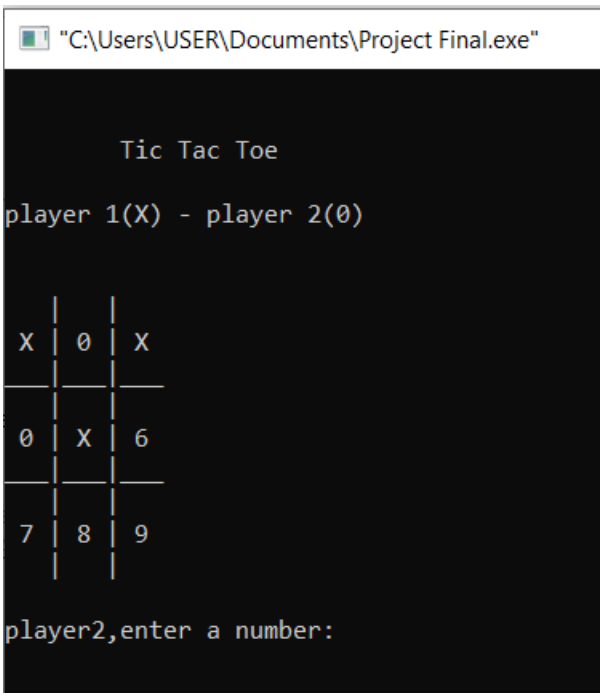
4. First player played for the second time:



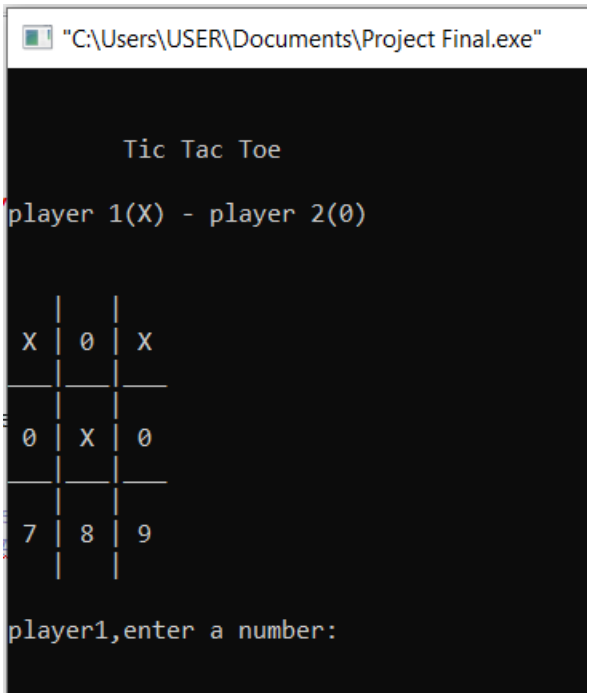
5. Second player played for the second time:



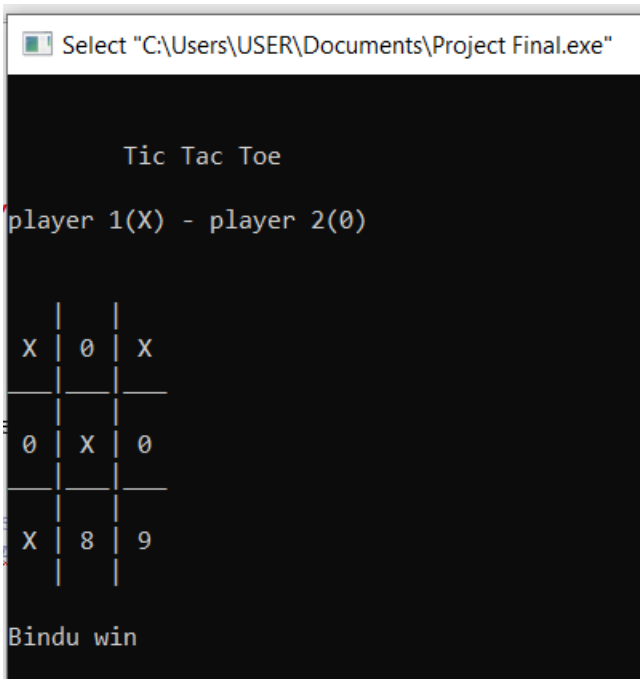
6. First player played for the third time:



7. second player played for the third time:



8.player 1 wins the game.



Chapter – 7

7.1 – Conclusion

For developing this project we faced some difficulties which are by the directions of our honorable supervisor sir. We are still working it for adding some additional features to make this project more user friendly.

1. Tic Tac Toe game in C is a fun and challenging project.

2. It helps improve coding skills and understanding of algorithms.
3. The project can be extended to implement AI-based players.

7.2 – Limitation

One limitation of a tic tac toe game project is that it can become repetitive and predictable over time. Since there are a limited number of possible moves and strategies in tic tac toe, experienced players may quickly learn optimal moves and the game can become less challenging.

7.3– Future work

I know the limitation of our project. So I can improve it in future. I will upload out code at github and open to edit. So that others can contribute to our code and make is more efficient.