PROJECT I

Title

TIC-TAC-TOE

Course

CIS-5

Section

40488

Date

January 30th, 2017

Author

Linh Vu

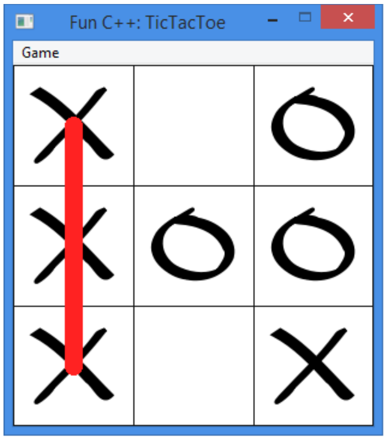
I. INTRODUCTION

Title: Tic-Tac-Toe

If you have not played Tic-Tac-Toe and are not familiar with the game, then here is a description from Wikipedia.

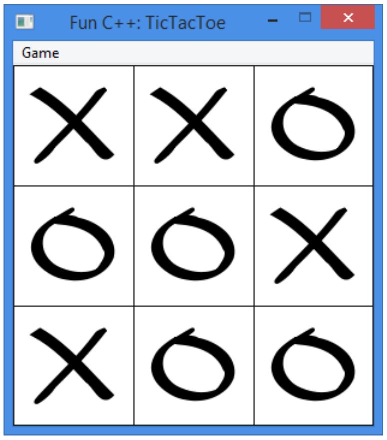
II. GAMES PLAY AND RULES

> Tic-Tac-Toe (or Noughts and crosses, Xs and Os) is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3x3 grid. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game.



X		O
X	O	O
X		X

X player wins



X	X	O
O	O	X
X	O	O

game ends in a draw

The game is to be played between two people opponents. Each of the two can start first.

There are two things to implement for creating the game: the game logic and the game user interface. In this article, I will show how can we use the same game logic implementation to build applications using technology. We will create this with C++ for the Windows Runtime.

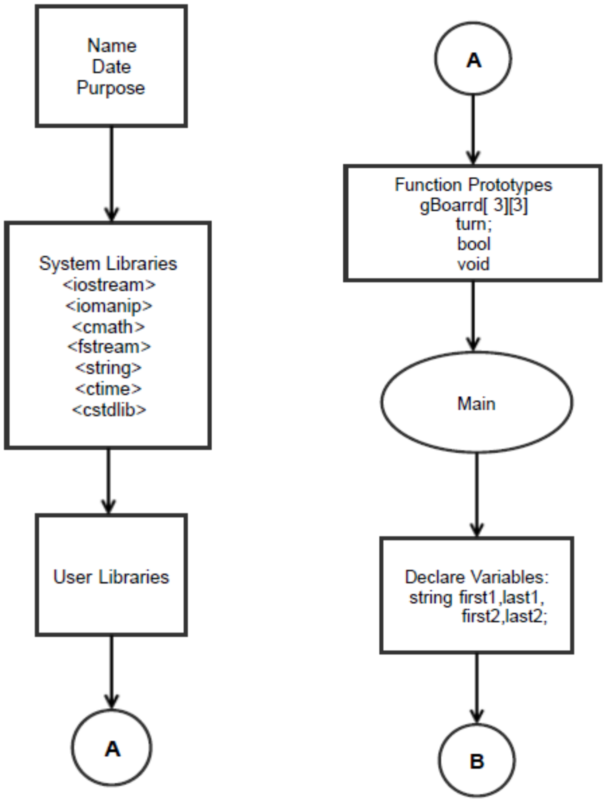
III. DEVELOPMENT SUMMARY:

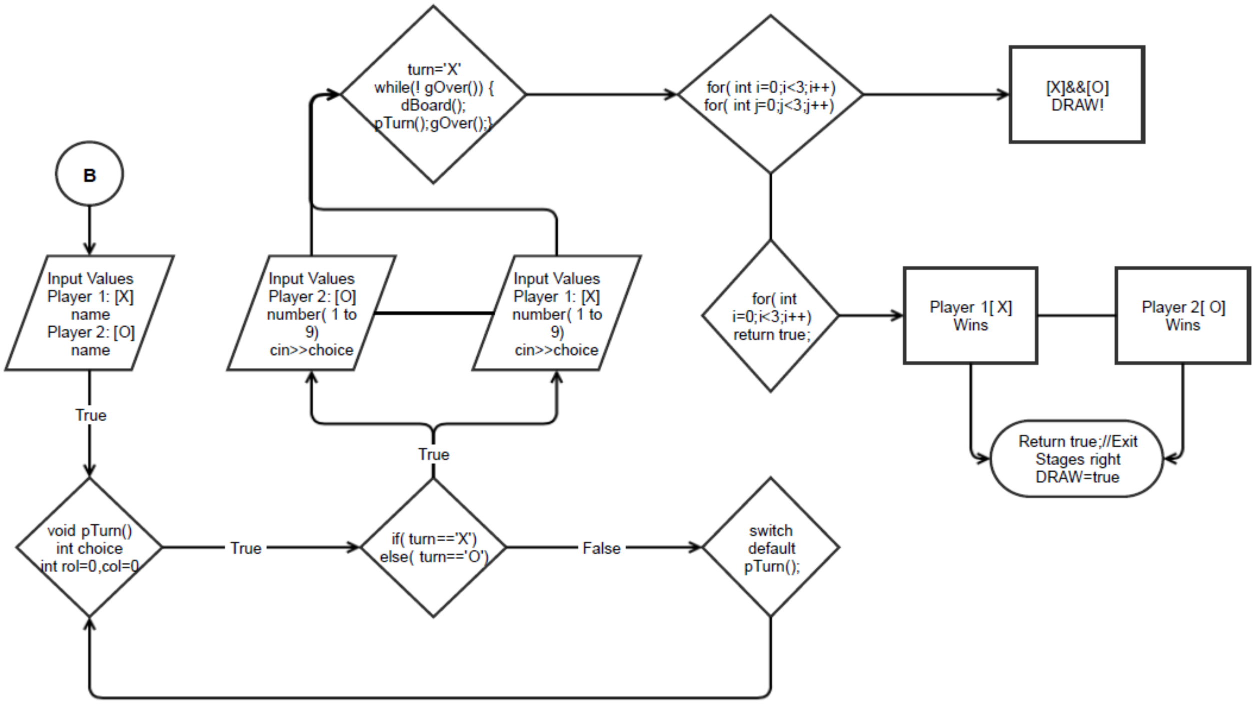
Lines of Code: 121
Comment Lines: 36
Blank Lines (White Spaces): 13
Total Lines of Source File: 157

This game is running on C++ Program. The hardest part of this program is to output the board for players to input X and O. Instead of clicking into the box and type X or O, I created a switch for players to input numbers from 1 to 9. Each box will have a number, Player 1 is assigned as “X” and Player 2 is assigned as “O”. The board will display after Player 1 and Player 2 typed in their first and last names. The program will asks Player 1 to input a number from 1 to 9 then hit enter, the board then will display again with “X” replaces the number that Player 1 chose. After that will be Player 2 turns to play, the program will ask the same thing and “O” will be replace the number that Player 2 chose.

The game will continue until it finds a winner, if one of the two Players type a wrong number, the program will ask the player to input a correct number again. This continue until all 9 boxes are fill up with Xs and Os. From this point the program will find a winner, if a winner cannot be found [X] and [O] are DRAW.

IV. FLOWCHART:





V. PSEUDO CODE:

```
//System Libraries
//Input Output Library
//Formatting
//Math Library
//File I/O
//String Object
//Time for Random Seed
//Setting Random Seed
//Namespace std of system libraries

//User libraries

//Global Constants
//Such as PI, Vc, -> Math/Science Values
//As well as conversions from system of unites to another

//Function Prototypes
//Game board output      (gBoard[3][3])
//Player turns           (turn)
//Game won               (DRAW=false)
//Game over              (gOver())
//Display Board          (dBoard())
//Player switching turns (pTurn())
```

```
//Main -> //Executable code begins here!
    //Declare Variables (2 Players)

    //Main Menu Introducing the Game
            //Enter (first1) and (last1) name – Player1[X]
            //Enter (first2) and (last2) name – Player2[O]

    //Mapping out the game
            //Player1[X] turn
            //Player2[O] turn

//Game Board Output
            //TIC-TAC-TOE
            //Player 1 (X):    -    Player 2 (O):
            //Choose a number from 1 to 9
            //Choose right (O) turns
            //Choose wrong (input a correct number again)
            //Player2[0] turns

//Player Switching Turns
            //Declare Variables (int choice)
                                  (int row=0,col=0)

//Plyer Turns
//Mapping out
//Player Turns on the Game
//Game over
            //Check for a winner
            //Check for a draw

//Output all the game to a file

//Close files and Exit Stage right!
```

VI. PROGRAM C++:

```
//System Libraries
#include <iostream> //Input and Output Library
#include <iomanip>  //Formatting
#include <cmath>    //Math Library
#include <fstream>  //File I/O
#include <string>    //String Object
#include <ctime>    //Time for random seed
#include <cstdlib>  //Setting  random seed
```

```
using namespace std;

//User Libraries

//Global Constants
//Such as PI, Vc, -> Math/Science values
//as well as conversions from system of units to another

//Function Prototypes
char gBoard[3][3]= { {'1','2','3'},
                     {'4','5','6'},
                     {'7','8','9'} }, //Game board output
     turn;           //Player turns
bool DRAW=false, //Game Won
          gOver(); //Game over
void dBoard();     //Display Board
void pTurn();      //Player switching turns

//Executable code begins here!!!
int main(int argc, char** argv) {
   //Declare Variables
   string first1,last1, //First and last name (Player 1)
          first2,last2; //First and last name (Player 2)

  //Main Menu (Introducing The Game)
  cout<<"          Welcome Players."<<endl;
  cout<<endl;
  cout<<"We're about to play a game called Tic Tac Toe.\nPlease enter your first and last name"
        " as Player 1 and Player 2."<<endl;
  cout<<endl;
  cout<<"Player 1: [X]"<<endl;
  cin>>first1>>last1;
  cout<<"Player 2: [O]"<<endl;
  cin>>first2>>last2;

  //Mapping out the game
  turn='X';
  while (!gOver()) {
        dBoard();
        pTurn();
        gOver();
  }
  if (turn=='O'&&!DRAW) {
     dBoard();
     cout<<endl<<endl<<"Player 1 [X] "<<first1<<" "<<last1<<" Wins!\n";
  }
  else if (turn=='X'&&!DRAW) {
     dBoard();
     cout<<endl<<endl<<"Player 2 [O] "<<first2<<" "<<last2<<" Wins!\n";
  }
  else {
     dBoard();
     cout<<endl<<endl<<"[X] and [O] ==> DRAW!\n";
  }
  return 0;
}

 void dBoard () {
    //Game Board Output
    cout<<"\n\t\tTIC-TAC-TOE\n   ";
    cout<<"Player 1 (X):  -  Player 2 (O):  "<<endl;
    cout<<" -------------------------------"<<endl;
    cout<<"\t        |        |    "<<endl;
    cout<<"\t  "<<gBoard[0][0]<<"  |   "<<gBoard[0][1]<<"  |  "<<gBoard[0][2]<<endl;
    cout<<"\t_____|_____|_____"<<endl;
    cout<<"\t        |        |    "<<endl;
    cout<<"\t  "<<gBoard[1][0]<<"  |  "<<gBoard[1][1]<<"   |  "<<gBoard[1][2]<<endl;
    cout<<"\t_____|_____|_____"<<endl;
    cout<<"\t        |        |    "<<endl;
    cout<<"\t  "<<gBoard[2][0]<<"  |  "<<gBoard[2][1]<<"   |  "<<gBoard[2][2]<<endl;
    cout<<"\t        |        |        "<<endl<<endl;
 }

 void pTurn() {
    //Declare Variables
    int choice;
    int row=0, col=0;

    //Players Turns
    if (turn=='X') {
       cout<<"Player 1 turn [X]: "; //Player 1 get to play
    }
    else if (turn=='0') {
       cout<<"Player 2 turn [O]: "; //Player 2 get to play
    }
    cin>>choice;

   //Mapping out the game
   switch (choice) {
      case 1: row=0;col=0;break;
      case 2: row=0;col=1;break;
      case 3: row=0;col=2;break;
```

```
       case 4: row=1;col=0;break;
       case 5: row=1;col=1;break;
       case 6: row=1;col=2;break;
       case 7: row=2;col=0;break;
       case 8: row=2;col=1;break;
       case 9: row=2;col=2;break;
       default:
           cout<<"You did not enter a correct number! Try again!\n";
           pTurn();
  }

  //Player turns on the game
  if (turn=='X'&&gBoard[row][col]!='X'&&gBoard[row][col]!='O') {
      gBoard[row][col]='X';
      turn='O';
  }
  else if (turn=='O'&&gBoard[row][col]!='X'&&gBoard[row][col]!='O') {
      gBoard[row][col]='O';
      turn='X';
  }
  else {
      cout<<"The cell you chose is used! Try again\n";
      pTurn;
  }
}

//Game Over
bool gOver() {
    for (int i=0;i<3;i++) { //Check for a winner
        if ((gBoard[i][0]==gBoard[i][1]&&gBoard[i][1]==gBoard[i][2])||
             (gBoard[0][i]==gBoard[1][i]&&gBoard[1][i]==gBoard[2][i])||
             (gBoard[0][0]==gBoard[1][1]&&gBoard[1][1]==gBoard[2][2])||
             (gBoard[0][2]==gBoard[1][1]&&gBoard[1][1]==gBoard[2][0])) {
            return true;
        }
    }
    for (int i=0;i<3;i++) { //Check for draw
        for (int j=0;j<3;j++) {
            if (gBoard[i][j]!='X'&&gBoard[i][j]!='O') {
                return false;
            }
        }
    }
    DRAW=true;
    return true;
}
```

VII. REFERENCE:

1. Gaddis, Tony. “starting out with >>> C++ From Control Structures through Objects,.” 8th Edition
2. C++ Assignments 1, 2, 3, 4.
3. www.cplusplus.com