Project 1

<TIC-TAC-TOE>

Course

CSC-5 Fall 2013

Section

47982

Due Date

October 25th, 2013

Author

Oscar Martinez

Table of Contents

1 Introduction....3
2 Game Play and Rules....3
3 Development Summary....3
4 Flowchart....4
5 Sample Inputs/Outputs....5
    5.1 Tied Game....5
    5.2 Player Won....6
    5.3 Invalid Move....6
    5.4 Option to play again....7
6 Pseudo Code....7
7 Variables Used....8
8 Concepts Used....8
9 References....8
12 Program Listing....9

1 Introduction

Project title: TIC-TAC-TOE

For the programming project, I decided to go with TIC-TAC-TOE. I think that TIC-TAC-TOE is one of the first games that we learn as kids. This game is a great game to play when one has some free and all one needs is a computer or even just a simple piece of paper. I started with the idea of doing TIC-TAC-TOE for my project and got started writing the code thinking that it would be worthy of a programming game, without taking into account that this project has to be further expanded for the final project. When Dr. Lehr mentioned in class that TIC-TAC-TOE would be difficult to expand, it was already too late to start on something else. I should have probably consulted my idea with Dr. Lehr before getting too far into it. Hopefully there is a way to further expand this game to be worthy of a final project.

2 Game Play and Rules

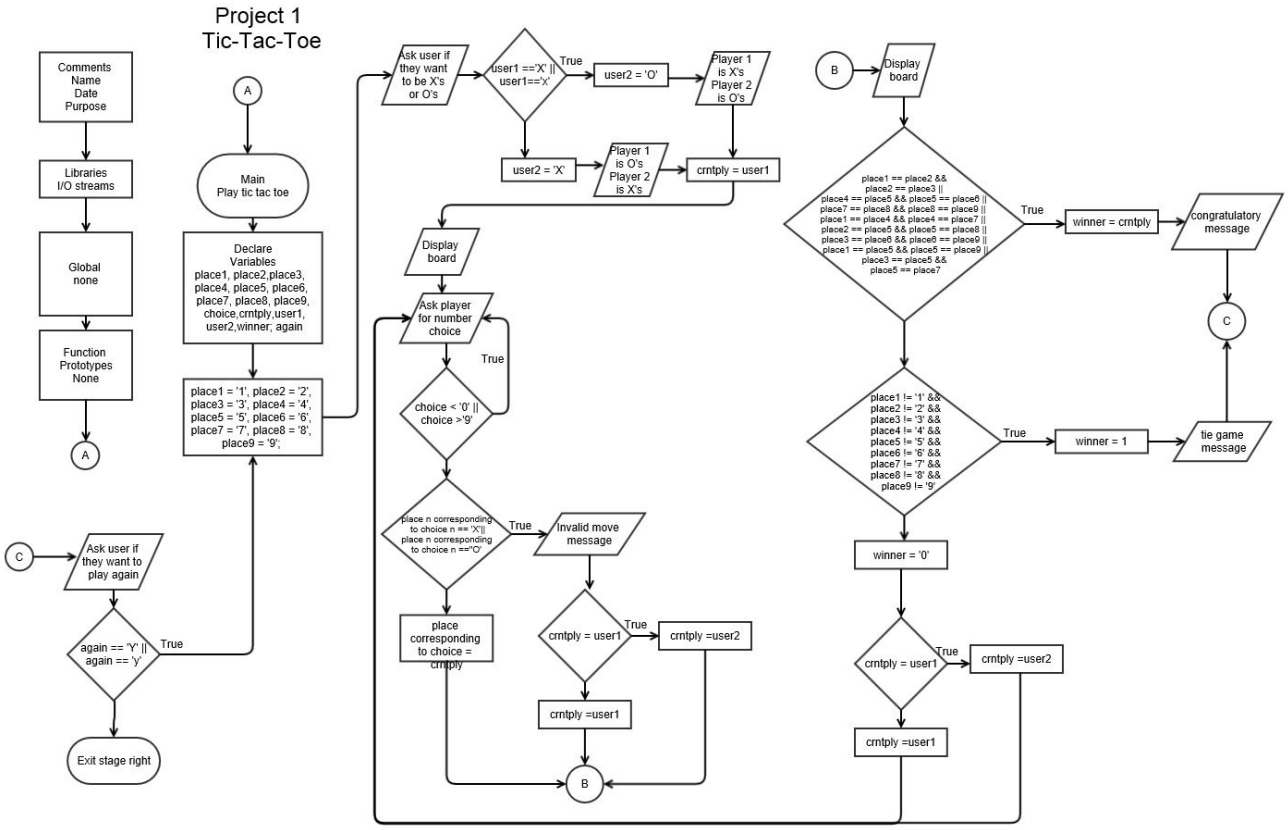
The game consists of a board with nine empty boxes where one of two players who are either an X or an O will take turns placing their letter on a box. Whichever player is able to get three of his/her letters in a row wins. The three letters have to be together either in a row column or diagonally. Should every box have a letter and there is no winner the game will be tied.

3 Development Summary

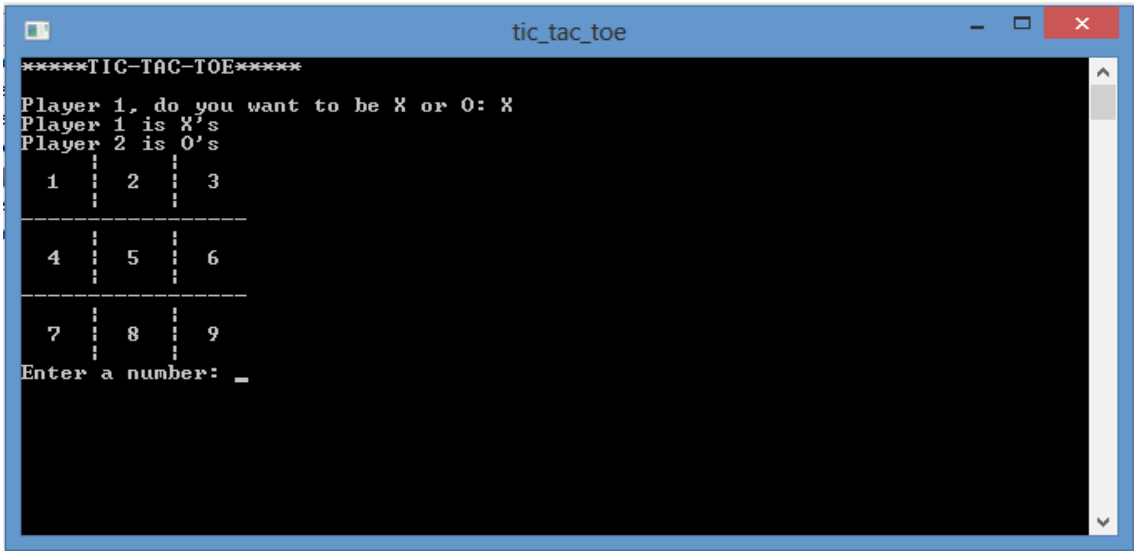
Lines of code	**226**
Comment Lines	**71**
Blank Lines(White Space)	**15**
Variables Used	**15**
Total Project Size	**312**

This project was created using the Netbeans IDE and the folwchart was created using Gliffy.

4 Flowchart

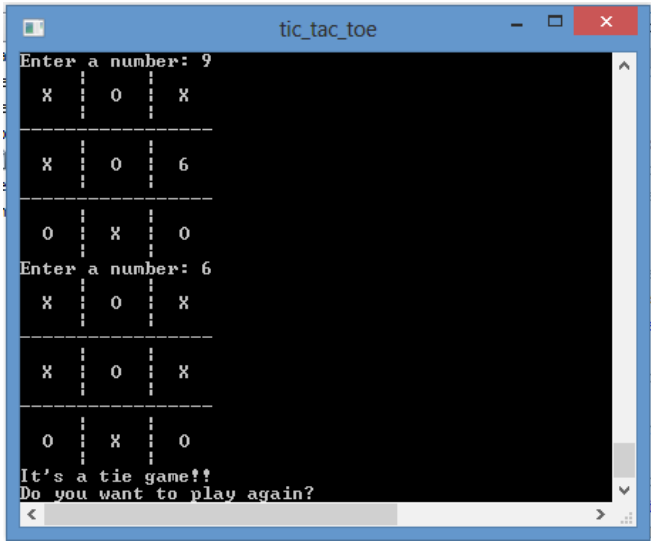

Project 1
Tic-Tac-Toe
Comments
Name
Date
Purpose
Libraries
I/O streams
Global
none
Function
Prototypes
None
A
Main
Play tic tac toe
Declare
Variables
place1, place2,place3,
place4, place5, place6,
place7, place8, place9,
choice,crntply,user1,
user2,winner; again
place1 = '1', place2 = '2',
place3 = '3', place4 = '4',
place5 = '5', place6 = '6',
place7 = '7', place8 = '8',
place9 = '9';
Ask user if
they want
to be X's
or O's
user1 =='X' ||
user1=='x'
True
user2 = 'O'
Player 1
is X's
Player 2
is O's
user2 = 'X'
Player 1
is O's
Player 2
is X's
crntply = user1
Display
board
Ask player
for number
choice
True
choice < '0' ||
choice >'9'
place n corresponding
to choice n == 'X'||
place n corresponding
to choice n =='O'
True
Invalid move
message
place
corresponding
to choice =
crntply
crntply = user1
True
crntply =user2
crntply =user1
B
B
Display
board
place1 == place2 &&
place2 == place3 ||
place4 == place5 && place5 == place6 ||
place7 == place8 && place8 == place9 ||
place1 == place4 && place4 == place7 ||
place2 == place5 && place5 == place8 ||
place3 == place6 && place6 == place9 ||
place1 == place5 && place5 == place9 ||
place3 == place5 &&
place5 == place7
True
winner = crntply
congratulatory
message
C
place1 != '1' &&
place2 != '2' &&
place3 != '3' &&
place4 != '4' &&
place5 != '5' &&
place6 != '6' &&
place7 != '7' &&
place8 != '8' &&
place9 != '9'
True
winner = 1
tie game
message
winner = '0'
crntply = user1
True
crntply =user2
crntply =user1
C
Ask user if
they want to
play again
again == 'Y' ||
again == 'y'
True
Exit stage right
A

5 Sample Inputs/Outputs



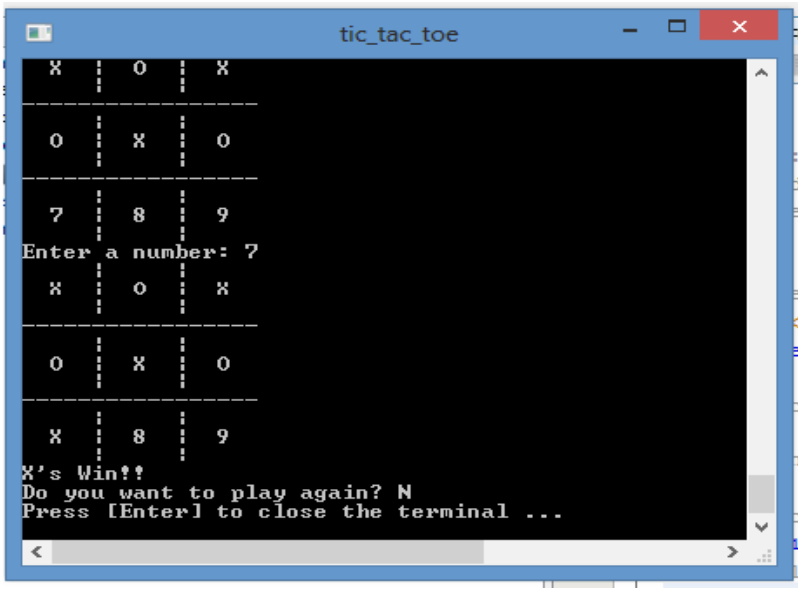
This screen shows the start of the game

5.1 Tied Game



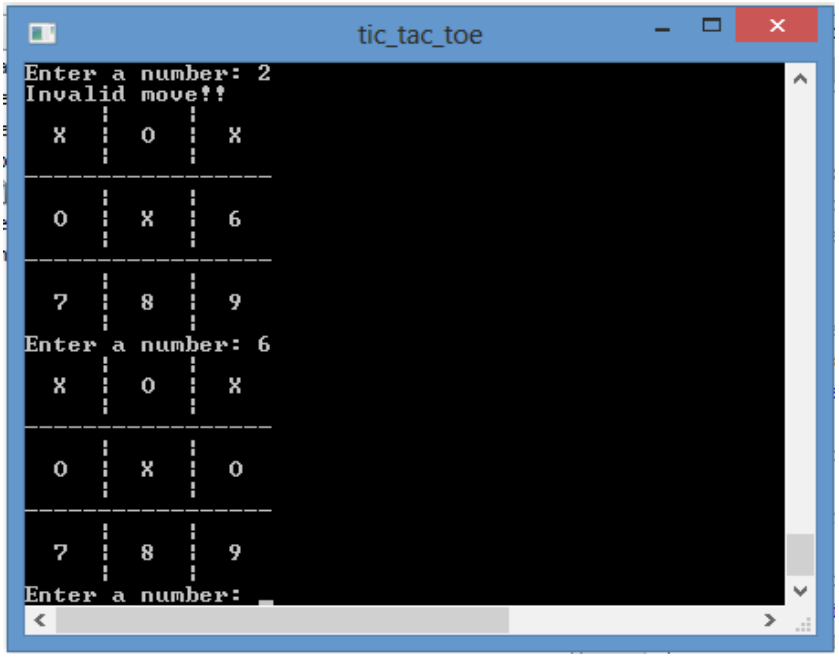
This screen demonstrates a tied game and asks the user if they would like to play again.

5.2 Player Won



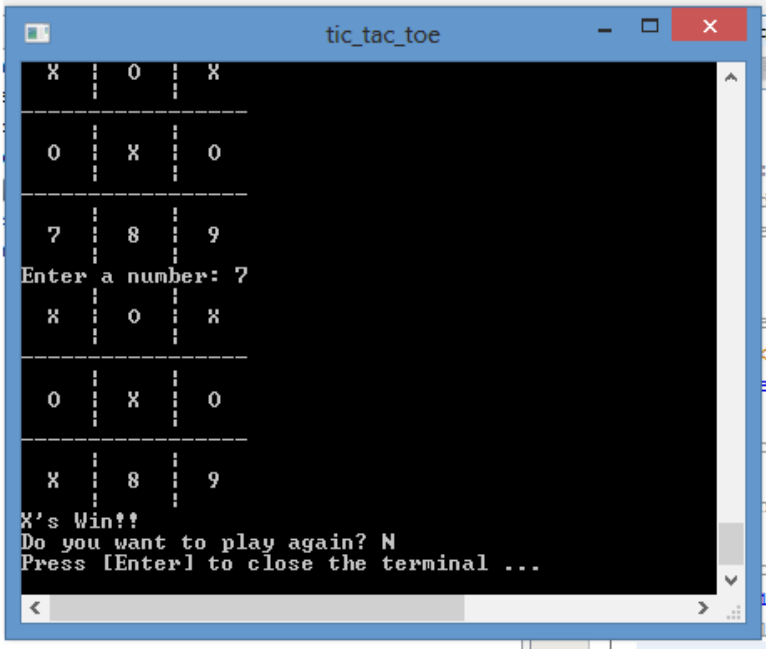
In this case the player that is X's wins and he/she is asked if they would like to play again.

5.3 Invalid Move



Should the player input a number that is already taken he/she will be asked to input another number.

5.4 Option to play again



Once the game is over, the user is asked if they would like to play again. If so, the game is restarted and if not, then the program exits.

6 Pseudo Code

//Declare variables

//Assign numbers to each box

//Heading

//Ask the user if they want to be X's or O's

//Sets user1 to current player

//Display board

//Prompts user for a move and validates

//Keeps switching between players ans asking for a play while there is no winner

//Checks that the move made by the user is one of the choices on the grid

//Changes the number on the grid to the current player's letter

** The following is repeated for choices 1-9

//Checks to see if the move made by the current user is valid

//meaning there is no X or O

//If the move is invalid, the current player is changed

//so that the same player can go again

//If the move is valid, the number chosen by the player changes to X or O

**

//Displays board

//Checks if current player won (row, column, diagonal)

// Checks for tied game

//If no winner and not a tried game, switches player and asks for another move

//Asks player if they want to play again if there's a winner or it's tied

7 Variables Used

Variable Name	**Purpose**
place1,place2,place3,place4,place5,place6,place7, place8,place9	Boxes on the board
choice	Player's move
crntply	Current player (X or O)
user1 & user2	Keeps track of player's game piece (X or O)
winner	Holds the status of winner
again	Choice of whether a player wants to play again

8 Concepts Used

Chapter	**Topics used**
2	Cout object, #include directive, char data type, variable assignments and initialization, comments, programming style.
3	cin object
4	Relational operators, the if statement, the if/else statement, the if/else if statement, validating user input, switch statement, logical operators, comparing characters and strings
5	Do-while loop, sentinels

9 References

Gaddis, Tony. *Starting out with C : From Control Structures through Objects*. 6th ed. Boston: Pearson Addison-Wesley, 2012. Print.

Savitch, Walter J., and Kenrick Mock. *Problem Solving with C*. 8th ed. Boston: Pearson Addison Wesley, 2012. Print.

Write ups of previous projects

12 Program Listing

```
/*
 * File:   main.cpp
 * Author: Oscar Martinez
 * Created on October 10, 2013, 7:37 PM
 * Purpose: Tic-Tac-Toe Project
 */

//Libraries
#include <iostream>
using namespace std;

//Global constants -> none

//Function prototypes -> none

//Execution begins here
int main(int argc, char** argv) {
    //Declare variables here
    char place1, place2, place3, place4, place5, place6,
         place7, place8, place9,   //Tic-Tac-Toe squares
         choice,                   //Player's move
         crntply,                  //Current Player
         user1, user2,             //Player one and player two
         winner,                   //Winner of game
         again;                    //Choice to play again


    do{
       //Assign numbers to each box
       place1 = '1', place2 = '2', place3 = '3', place4 = '4', place5 = '5',
       place6 = '6', place7 = '7', place8 = '8', place9 = '9';

       //Heading
```

```
cout << "*****TIC-TAC-TOE*****" <<endl <<endl;

//Ask the user if they want to be X's or O's
cout << "Player 1, do you want to be X or O: ";
cin >> user1;

//Displays each player with their corresponding letter (X or O)
if(user1 == 'X' || user1 == 'x'){
   user2 = 'O';
   cout << "Player 1 is X's" <<endl
        << "Player 2 is O's" <<endl;
}
else{
   user2 = 'X';
   cout << "Player 1 is O's" <<endl
        << "Player 2 is X's" <<endl;
}
//Sets user1 to the current player
crntply = user1;

//Display board
cout << "     |     |     "  << endl
    << "  " <<place1 <<"  |  " <<place2 <<"  |  " <<place3 <<endl
    << "     |     |     "  << endl
    << "-----------------"  << endl
    << "     |     |     "  << endl
    << "  " <<place4 <<"  |  " <<place5 <<"  |  " <<place6 <<endl
    << "     |     |     "  << endl
    << "-----------------"  << endl
    << "     |     |     "  << endl
    << "  " <<place7 <<"  |  " <<place8 <<"  |  " <<place9 <<endl
    << "     |     |     "  << endl;

//Prompts user for a move and validates
```

```
//Keeps switching between players and asking for a play while there is no winner
do{
   //Checks that the move made by user is one of the choices on the grid
   do{
      cout << "Enter a number: ";
      cin >> choice;
   }while(choice < '0' || choice > '9');
   //Changes the number on the grid to the current player's letter
   switch(choice){
      case '1':{
         //Checks to see if the move made by current user is valid
         //meaning there is no X or O
         if(place1 == 'X' || place1 == 'O'){
            cout << "Invalid move!!" <<endl;
            //If the move is invalid, the current player is changed
            //so that the same player can go again
            if(crntply == user1){
               crntply = user2;
            }
            else{
               crntply = user1;
            }
         }
         //If the move is valid, the number chosen by player changes
         //to X or O
         else{
            place1 = crntply;
         }
         break;
      }
      case '2':{
         //Checks to see if the move made by current user is valid
         //meaning there is no X or O
         if(place2 == 'X' || place2 == 'O'){
```

```
            cout << "Invalid move!!" <<endl;
            //If the move is invalid, the current player is changed
            //so that the same player can go again
            if(crntply == user1){
               crntply = user2;
            }
            else{
               crntply = user1;
            }
         }
         else{
            place2 = crntply;
         }
         break;
      }
      case '3':{
         //Checks to see if the move made by current user is valid
         //meaning there is no X or O
         if(place3 == 'X' || place3 == 'O'){
            cout << "Invalid move!!" <<endl;
            if(crntply == user1){
               crntply = user2;
            }
            else{
               crntply = user1;
            }
         }
         else{
            place3 = crntply;
         }
         break;
      }
      case '4':{
         //Checks to see if the move made by current user is valid
```

```
    //meaning there is no X or O
    if(place4 == 'X' || place4 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
        }
        else{
            crntply = user1;
        }
    }
    else{
        place4 = crntply;
    }
    break;
}
case '5':{
    //Checks to see if the move made by current user is valid
    //meaning there is no X or O
    if(place5 == 'X' || place5 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
        }
        else{
            crntply = user1;
        }
    }
    else{
        place5 = crntply;
    }
```

```
    break;
}
case '6':{
    //Checks to see if the move made by current user is valid
    //meaning there is no X or O
    if(place6 == 'X' || place6 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
        }
        else{
            crntply = user1;
        }
    }
    else{
        place6 = crntply;
    }
    break;
}
case '7':{
    //Checks to see if the move made by current user is valid
    //meaning there is no X or O
    if(place7 == 'X' || place7 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
        }
        else{
            crntply = user1;
        }
```

```
    }
    else{
        place7 = crntply;
    }
    break;
}
case '8':{
    //Checks to see if the move made by current user is valid
    //meaning there is no X or O
    if(place8 == 'X' || place8 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
        }
        else{
            crntply = user1;
        }
    }
    else{
        place8 = crntply;
    }
    break;
}
case '9':{
    //Checks to see if the move made by current user is valid
    //meaning there is no X or O
    if(place9 == 'X' || place9 == 'O'){
        cout << "Invalid move!!" <<endl;
        //If the move is invalid, the current player is changed
        //so that the same player can go again
        if(crntply == user1){
            crntply = user2;
```

```
            }
            else{
                crntply = user1;
            }
        }
        else{
            place9 = crntply;
        }
        break;
    }
}

//Display board
cout << "     |     |     "  << endl
    << "  " <<place1 <<"  |  " <<place2 <<"  |  " <<place3 <<endl
    << "     |     |     "  << endl
    << "-----------------"  << endl
    << "     |     |     "  << endl
    << "  " <<place4 <<"  |  " <<place5 <<"  |  " <<place6 <<endl
    << "     |     |     "  << endl
    << "-----------------"  << endl
    << "     |     |     "  << endl
    << "  " <<place7 <<"  |  " <<place8 <<"  |  " <<place9 <<endl
    << "     |     |     "  << endl;

//Checks to see if the current player won
//Checks if the rows, columns or diagonals are the same
if(place1 == place2 && place2 == place3 ||
   place4 == place5 && place5 == place6 ||
   place7 == place8 && place8 == place9 ||
   place1 == place4 && place4 == place7 ||
   place2 == place5 && place5 == place8 ||
   place3 == place6 && place6 == place9 ||
   place1 == place5 && place5 == place9 ||
```

```
   place3 == place5 && place5 == place7){
    //If a row, diagonal, or column is the same the player that
    //just went is the winner
    winner = crntply;
    //Displays the winner
    cout << winner <<"'s Win!!" <<endl;
}
//If all the boxes have either an O or X, and no one won,
//the game is tied
else if( place1 != '1' &&
         place2 != '2' &&
         place3 != '3' &&
         place4 != '4' &&
         place5 != '5' &&
         place6 != '6' &&
         place7 != '7' &&
         place8 != '8' &&
         place9 != '9'){
    //Displays tie game message
    cout << "It's a tie game!!" <<endl;
    //Sets winner to 1 so that it can exit the loop and stop asking for moves
    winner = '1';
}
//There is no winner
else{
    //Sets winner to 0 so it can keep asking for moves
    winner = '0';
    //Switches players
    if(crntply == user1){
       crntply = user2;
    }
    else{
       crntply = user1;
    }
```

```
        }
    }while(winner == '0');
    //Prompts the user if they want to play again
    cout << "Do you want to play again? ";
    cin >> again;
  }while(again == 'Y' || again == 'y');
  return 0;
}
```