

Project 1

<Tic Tac Toe>

Course

CSC-5

Section

40651

Due Date

February 12, 2014

Author

Chanyap Ho

Introduction

Title: Tic Tac Toe

This is the classic game that everyone knows.

In this game, two player is required to line up their Xs or Os in a straight line in a 3x3 grid.

The player who succeed in placing three Xs or Os in a horizontal, vertical or diagonal row will win the game.

For example, Os' player won the game in the picture below.



Although simple in design and execution, it can provides lots of fun. In addition, I also added a win counter to the game for players to record their scores.

Summary

Version 1

Project size: 175 lines

The number of variable: 18

At first, I tried to code this program by using array. However, due to the lack of time and my lack of expertise, using array in this program proved too much for me at this moment. I have tried to implement the two dimensional array that was taught in class, but my effort was futile. It took me more time to debug it than write it.

Therefore, I have resorted to using the conventional way of writing C++ program without the use of any function prototypes and array. Hopefully I'll be able to rewrite the program using array and function prototype in the upcoming project while adding new features.

This project took me two days to write the code, and one day to finish the report.

The flowchart is the painful part, it took me 4 hours to complete it. I even lost my progress during one of the session due to internet acting up. Anyhow, I hope my project can live up to expectations.

Version 2

Project size: 292 lines

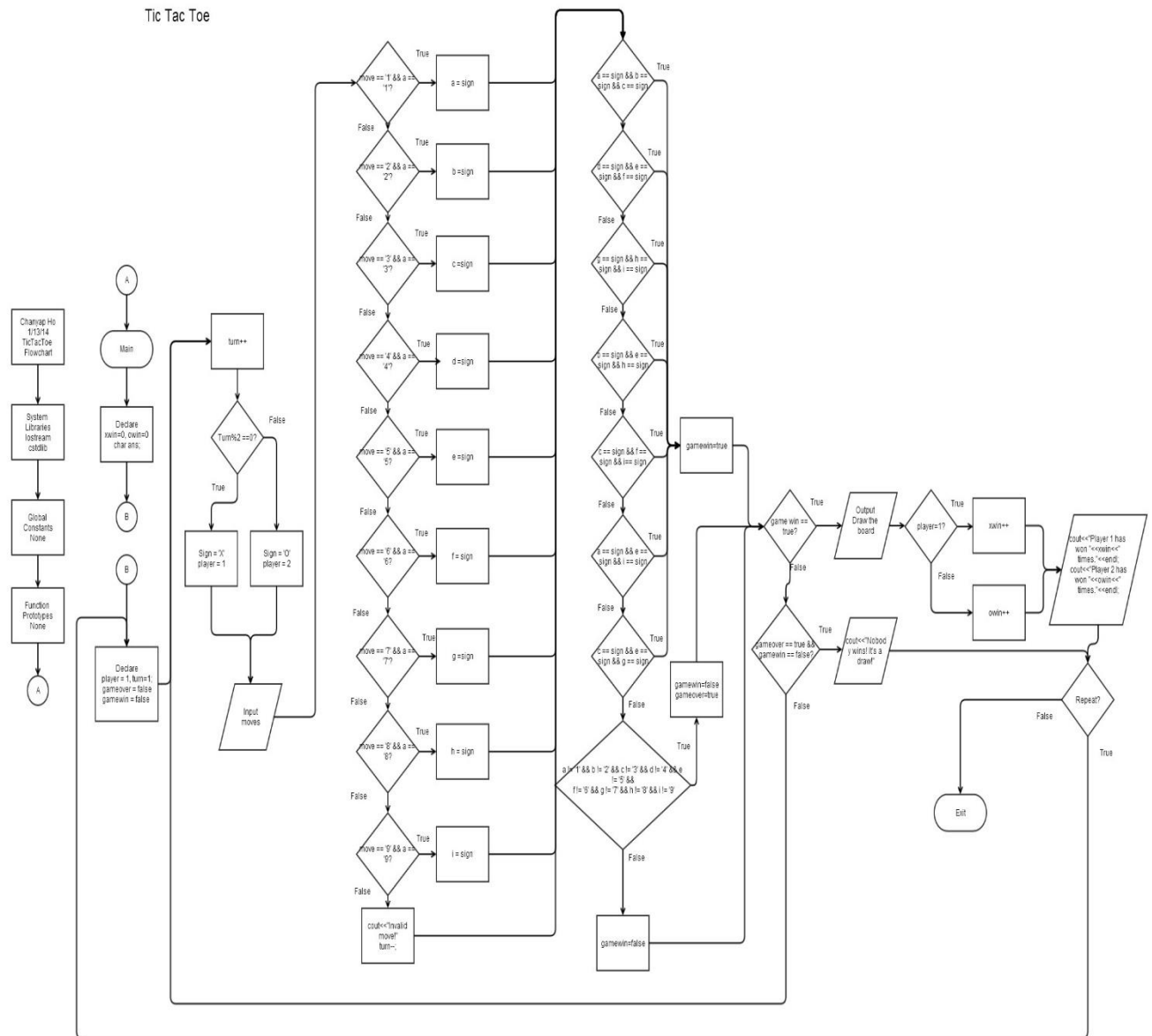
The number of variable: 20

I have successfully implemented a two dimensional array to my project. However, I left the way of a creating a 3x3 grid behind to display a better looking board when the game is over.

Also, I successfully utilized function prototypes and pass by references in my program.

In addition, I also added a function that allows user to save their win rate to a file in the program.

Flow Chart Version 1

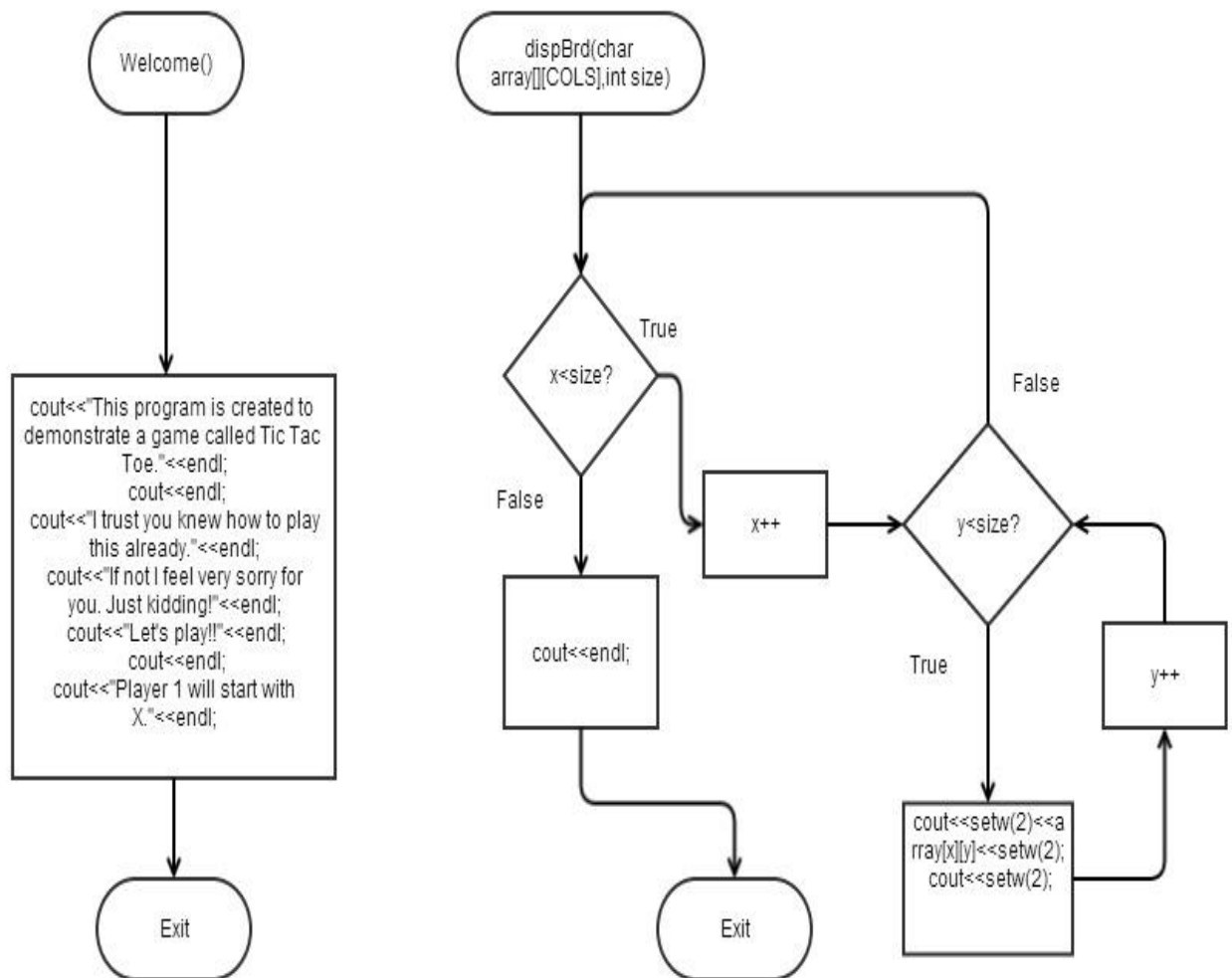


Since the flowchart is too small to be seen on the report, I have included an original copy in the folder.

Flow Chart Version 2



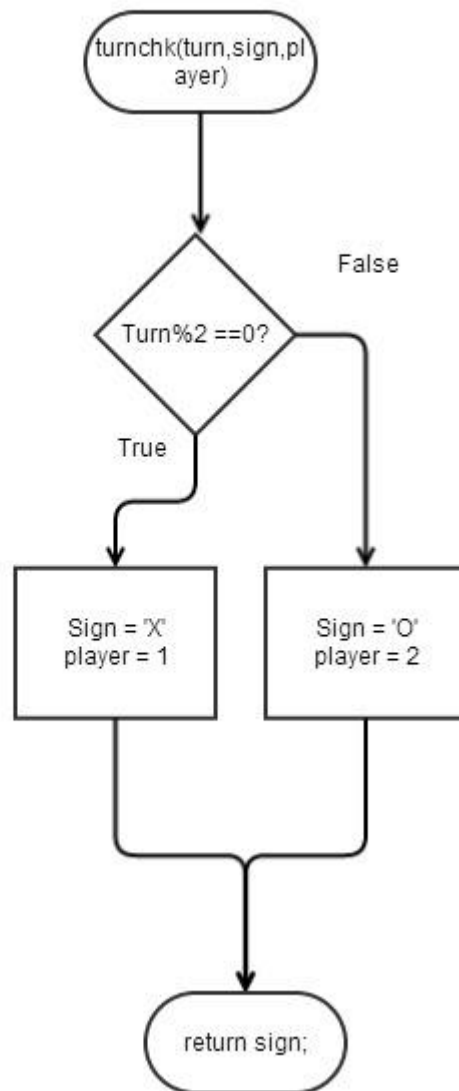
Welcome and displayBoard function



Welcome function is being used to display the welcome message.

The dispBrd function is being used to draw a 3x3 grid using a 2d array.

Turncheck function



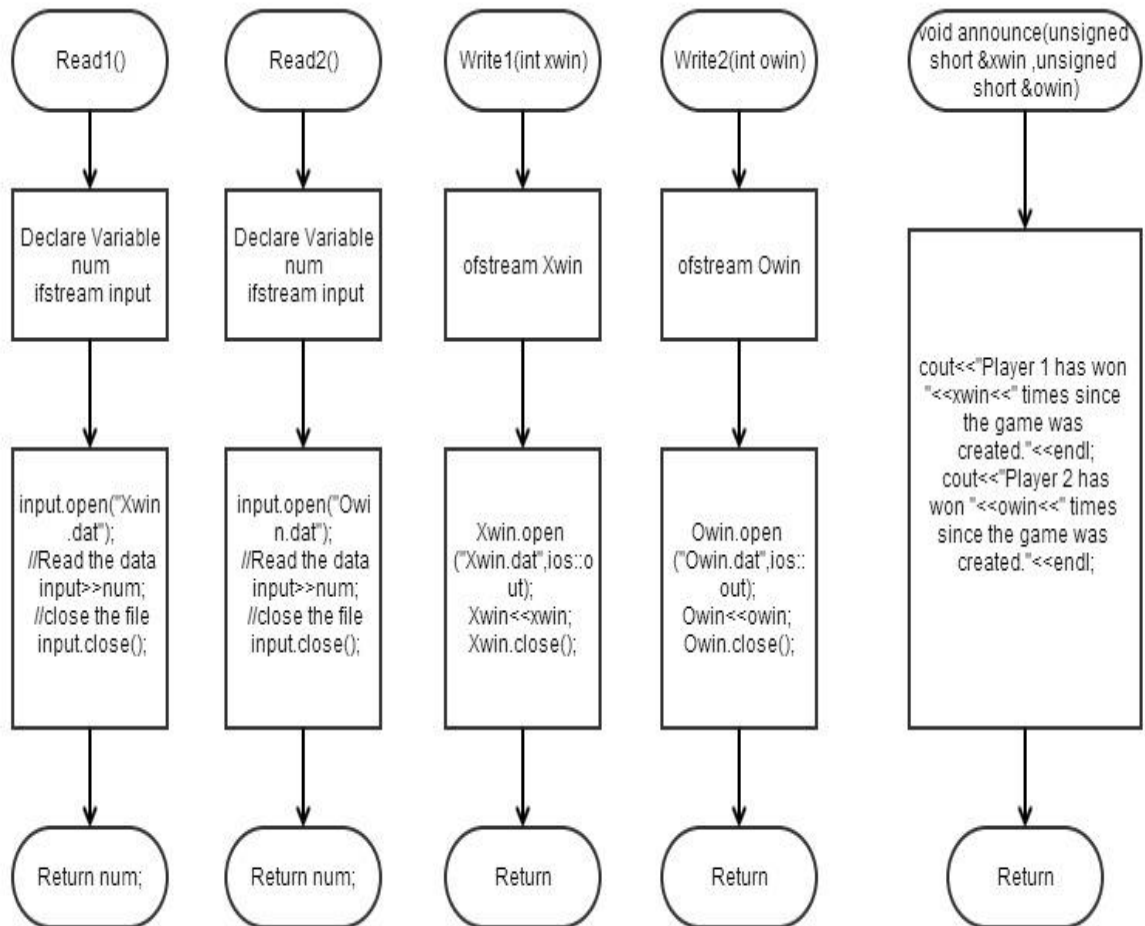
In this turncheck function, turns are being decided by a “if-else” loop.

If turn is even number, then it shall be player 1’s turn.

If turn is odd number, then it shall be player 2’s turn.

The sign are being returned in order to mark the grid.

Read, write, and announce function



In this flowchart, I have 5 functions dedicated to reading, saving and writing data.

Two read function are used to retrieve data from “Xwin.dat” and “Owin.dat.”

Two write function are used to save data to respective data file.

The last announce function is used to report the winnings since the creation of the game.

Pseudocode V1

Initialize

Set X's win counter to 0

Set O's win counter to 0

Do

Set Player to 1

Set Turn to 1

Set gameover to false

Set gamewin to false

Draw the 3 x 3 grid

+ 1 to Turn

If the remainder of Turn is 0

Sign = X

Player = 1

Remind player's turn to play

Else

Sign = O

Player = 2

Remind player's turn to play

Get input

//check move's validity

If move is valid and a is available

Mark 'a' as sign

else If move is valid and b is available

Mark 'b' as sign

Else If move is valid and c is available

Mark 'c' as sign

Else If move is valid and a is available

Mark 'd' as sign

Else If move is valid and e is available

Mark 'e' as sign

Else If move is valid and f is available

Mark 'f' as sign

Else If move is valid and g is available

Mark 'g' as sign

Else If move is valid and h is available

Mark 'h' as sign

Else If move is valid and i is available

Mark 'i' as sign

Else

Prompt for valid move

-1 to turn in order to compensate for move taken

//check win condition

If a is equivalent to sign and b is equivalent to sign and c is equivalent to sign

Set gamewin to true

Else If d is equivalent to sign and e is equivalent to sign and f is equivalent to sign

Set gamewin to true

Else If g is equivalent to sign and h is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If a is equivalent to sign and d is equivalent to sign and g is equivalent to sign

Set gamewin to true

Else If b is equivalent to sign and e is equivalent to sign and h is equivalent to sign

Set gamewin to true

Else If c is equivalent to sign and f is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If a is equivalent to sign and e is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If c is equivalent to sign and e is equivalent to sign and g is equivalent to sign

Set gamewin to true

Else If a is not equal to '1' and b is not equal to '2' and c is not equal to '3' and d is not equal to '4' and e is not equal to '5' and f is not equal

to '6' and g is not equal to '7' and h is not equal to '8' and i is not equal to '9'

Set gamewin to false

Set gameover to true

Else

Set gamewin to false

//display result

If gamewin is true

Print the placement on the board

Print the player who won

If player 1 won

Add 1 to X's win counter

If player 2 won

Add 1 to O's win counter

Else if gameover is true and gamewin is false

Print the placement on the board

Print "Nobody wins! It's a draw!"

Print Player 1 win count

Print Player 2 win count

While gameover is false

Ask for repeat

Get answer

While answer is yes

Exit Program

Pseudocode V2

Declare xwin and owin as unsigned short

Declare ans as char data type

Do

Set Player to 1

Set Turn to 1

Set gameover to false

Set gamewin to false

Run the welcome function to show welcome message

Read from xwin.dat

Read from owin.dat

Announce the win count so far

Initialize the game

Set up the 2d array

Do

Draw the 3 x 3 grid using a 2d array function.

+ 1 to Turn

Run the turncheck function

Set sign = turncheck function output

If sign is equivalent to X

Set player to 1

Else set player to 2

Get input

check move's validity

If move is valid and a is available

 Mark 'a' as sign

 Set [2][0] position on gaming board to sign

else If move is valid and b is available

 Mark 'b' as sign

 Set [2][1] position on gaming board to sign

Else If move is valid and c is available

 Mark 'c' as sign

 Set [2][2] position on gaming board to sign

Else If move is valid and a is available

 Mark 'd' as sign

 Set [1][0] position on gaming board to sign

Else If move is valid and e is available

 Mark 'e' as sign

 Set [1][1] position on gaming board to sign

Else If move is valid and f is available

 Mark 'f' as sign

 Set [1][2] position on gaming board to sign

Else If move is valid and g is available

 Mark 'g' as sign

 Set [0][0] position on gaming board to sign

Else If move is valid and h is available

Mark 'h' as sign

Set [0][1] position on gaming board to sign

Else If move is valid and i is available

Mark 'i' as sign

Set [0][2] position on gaming board to sign

Else

Prompt for valid move

-1 to turn in order to compensate for move taken

Check win condition

If a is equivalent to sign and b is equivalent to sign and c is equivalent to sign

Set gamewin to true

Else If d is equivalent to sign and e is equivalent to sign and f is equivalent to sign

Set gamewin to true

Else If g is equivalent to sign and h is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If a is equivalent to sign and d is equivalent to sign and g is equivalent to sign

Set gamewin to true

Else If b is equivalent to sign and e is equivalent to sign and h is equivalent to sign

Set gamewin to true

Else If c is equivalent to sign and f is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If a is equivalent to sign and e is equivalent to sign and i is equivalent to sign

Set gamewin to true

Else If c is equivalent to sign and e is equivalent to sign and g is equivalent to sign

Set gamewin to true

Else If a is not equal to '1' and b is not equal to '2' and c is not equal to '3' and d is not equal to '4' and e is not equal to '5' and f is not equal to '6' and g is not equal to '7' and h is not equal to '8' and i is not equal to '9'

Set gamewin to false

Set gameover to true

Else

Set gamewin to false

//display result

If gamewin is true

Print the placement on the board

Print the player who won

If player 1 won

Add 1 to X's win counter

Write the new win count to xwin.dat

If player 2 won

Add 1 to O's win counter

Write the new win count to owin.dat

Else if gameover is true and gamewin is false

Print the placement on the board

Print "Nobody wins! It's a draw!"

Run the read1() function

Run the read2() function

Announce the new data

While gameover is false

Ask for repeat

Get answer

While answer is yes

Exit Program

Variables		
Type	Variable Name	Description
Unsigned short	Xwin	Win counter for X
	Owin	Win counter for o
	Player	Player
	Turn	Turn
Char	Ans	Answer when asked for repeat
	Sign	A placeholder for X or O
	A	Lower left portion of the board
	B	Lower mid portion of the board
	C	Lower right portion of the board
	D	Left portion of the board
	E	Mid portion of the board
	F	Right portion of the board
	G	Upper left portion of the board
	H	Upper mid portion of the board
	I	Upper right portion of the board
	Board[3][3]	An 3x3 array
	Move	Player input
Boolean	Gameover	Flag to indicate the game is over
	Gamewin	Flag to indicate the game has been won

Int	Size	Size to loop the dispBrd function
	num	The number in the data file

C++ Constructs

Syntax and Keywords	Examples
Type_Name Variables_Names_1, Variables_Names_2	Char ans; Char sign;
Variable = Expression	Player = 1, turn =1;
Modulo operator	Turn%2==0;
Type_Name Variables_Names_1 = Expression_for_Value_1, Variables_Names_2 = Expression_for_Value_2	Unsigned short owin=0;
Cin>>Variable_1>>Variable_2	Cin>>move;
Cout<<Variable_1<<Variable_2	Cout<< "Player "<<player<<"'s turn to play"<<endl;
Line Breaks in I/O	Cout<<endl;
(comparison_1) && (comparison_2)	move == '1' && a == '1'
If(Boolean_expression_1) Statement_1 Else if(Boolean_expression_2) Statement_2	if(gamewin == true){ }else if (gameover == true && gamewin == false){ }
Do-While Loop	Do{ }while(ans == 'Y' ans == 'y')
Increment/Decrement Operator	Turn++; Turn--;
Pass by Reference	Unsigned short &xwin;
Boolean Expression	Bool gameover=false;
Calling a member function	Input.open("Xwin.dat")

Manipulators	Cout<<endl;
Formatting	ios::out
Type_Name Array_Name[Declared_Size]	Char array[][COLS]
Type_Returned Function_Name(...,Base_Type Array_Name[],...)	void dispBrd(char[][COLS],int);