Daniel Gonos

CIS 17A

11/05/2017

48969

**Project 1**

**Introduction:**

For my project I decided to make a tic tac toe game. Originally I thought of doing a card or casino type game but I love tic tac toe and thought it would be a really fun project to work on. It is a classic tic tac toe game. It has two user enter in their name age and whether they know how to play or not. If they do not know then the game will display the instructions on how to play. When a person wins it outputs the winner’s name along with congratulations. If there is no winner then it will display a draw!

**Summary:**

The program is around 200 lines.

The program uses 14 variables.

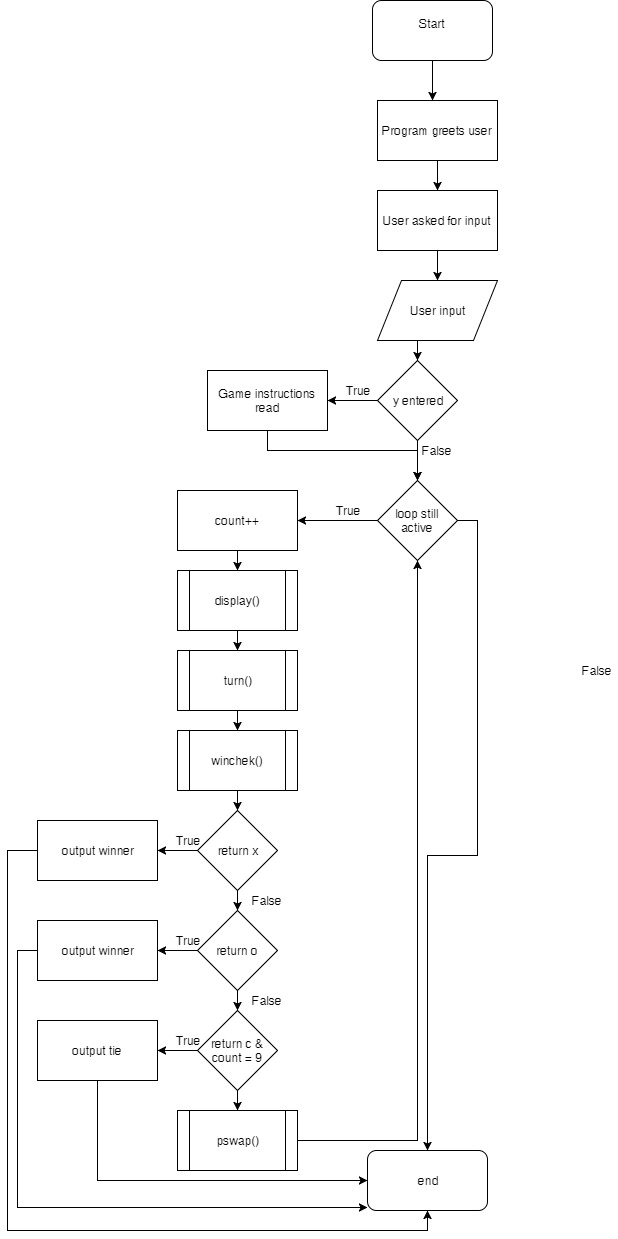
There are 4 different functions that are used in the program.

I did my best to implement as many things that we learned in class up until this point. I wanted to include even more but did not have enough time to add additional features. I would like to add different modes (2 of 3), (3 of 5) and the ability to play another game when one game ends. I also would like to have it keep track of player one and player twos total wins. And have this data output to an external file. It would be fun to be able to add some sort of modified version of tic tac toe along with even adapting a way to have each person have a wallet with wages that can be bet on each game.

I had a week to work on it since the midterm was just turned in this past week which did not give me as much time to think of additional features and implement/test them. I think that I can make this an awesome project with more time put into it I really want to up the overall display of it either with just using characters or even by trying to teach myself as much as possible from a gui library and trying to apply them to the project to give it some life.

I used pointers, c strings, 2d array, structures, allocated memory and deallocated memory.

**Flow Chart:**

****

**Source Code:**

#include <iostream>

#include <cstdlib>

#define nullptr nullptr

using namespace std;

struct Pinfo{ //structure that will hold a few basic details about the players

string name; //player name

int age; //player age

char exper; //check to see if player knows how to play

};

char board[3][3] = {'1','2','3','4','5','6','7','8','9'} ;//2d array that will created a 3x3 grid for out tic tac toe

char cplayer = 'X'; //default player that is set to X

char p1; // player 1

char p2; // player 2

//function prototypes

char winchek();

void pswap();

void turn();

void display();

int main() {

Pinfo \*ptr = nullptr;//ptr of struct

ptr = new Pinfo[2];//memory allocated for a struct array

int count = 0;//count used to keep track of loop number

cout << "Welcome To My TicTacToe Game" << endl;//welcome message

//user asked for input

cout << "Player 1 Enter Name:" << endl;

cin >> ptr[0].name;

cout << "Player 1 Enter Age:" << endl;

cin >> ptr[0].age;

cout << "Player One Needs Instructions?" << endl;

cout << "y for yes & n for no" << endl;

cin >> ptr[0].exper;

//second user asked for input

cout << "Welcome To My TicTacToe Game" << endl;

cout << "Player 2 Enter Name:" << endl;

cin >> ptr[1].name;

cout << "Player 1 Enter Age:" << endl;

cin >> ptr[1].age;

cout << "Player One Needs Instructions?" << endl;

cout << "y for yes & n for no" << endl;

cin >> ptr[1].exper;

if((ptr[1].exper=='y') || (ptr[0].exper=='y')){

cout <<"Tic Tac Toe is a fairly simple game. One player is X and one player is O. " << endl;

cout <<"You take turns by picking a spot on the 3x3 playing field and cannot pick " << endl;

cout <<"a spot that your opponent has already selected. The goal is to get three X's " << endl;

cout <<"or three O's in a row (depending on which one you are)." << endl;

cout <<"You can line up three in a row in any direction (vertical, horizontal & diagonal)" << endl;

cout <<"Example wins:"<< endl;

cout <<"X X X X 2 3 X 2 3"<< endl;

cout <<"4 5 6 X 5 6 4 X 6"<< endl;

cout <<"7 8 9 X 8 9 7 8 X"<< endl;

cout <<"Good Luck!"<< endl;

cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

};

bool active = true;//bool to keep loop running until it hits a break

while(active=1){

count++;//adds one to count as it loops

display();//runs display function that shows current board layout

turn();//turn function that has user input selection

if(winchek()=='X'){//checks win function to see that it has returned an 'X'

cout << "Player 1 Wins! (X)" << endl;

cout << "Congratulations " << ptr[0].name << "!" << endl;

break;//program ends on winner

}

else if(winchek()=='O'){//checks win function to see that it has returned an 'O'

cout << "Player 2 Wins! (O)" << endl;

cout << "Congratulations " << ptr[1].name << "!" << endl;

break;//program ends on winner

}

else if(winchek()=='C' && count==9){//checks win function to see that it has returned an 'C' and that it has looped 9 times

cout << "It Was A Draw!" << endl;

break;//program ends on winner

}

pswap();//at the end of each cycle the player char swaps

};

delete [] ptr; //free up allocated memory

return 0;

}

void display(){//function that outputs the 2d array after each loop

cout << board[0][0] << " "<<board[0][1]<< " " << board[0][2] << endl;

cout << board[1][0] << " "<<board[1][1]<< " " << board[1][2] << endl;

cout << board[2][0] << " "<<board[2][1]<< " " << board[2][2] << endl;

};

void turn(){//function that checks user input and modifies its appropriate element

int move; //used to check user input

cout << "Enter in the number you choose:" << endl;//user is asked to input a number

cin >> move;//user inputs number

if(move == 1) //checks input

board[0][0]=cplayer;//if equals the check it will change the number to an 'X' or 'O'

else if(move == 2)

board[0][1]=cplayer;

else if(move == 3)

board[0][2]=cplayer;

else if(move == 4)

board[1][0]=cplayer;

else if(move == 5)

board[1][1]=cplayer;

else if(move == 6)

board[1][2]=cplayer;

else if(move == 7)

board[2][0]=cplayer;

else if(move == 8)

board[2][1]=cplayer;

else if(move == 9)

board[2][2]=cplayer;

};

void pswap(){//function that changes player char after one loop it will check and turn 'X' to 'O' as well as 'O' to 'X'

if(cplayer=='X')

{ cplayer='O';

p2=cplayer;}

else if(cplayer=='O'){

cplayer='X';

p1=cplayer;

}

}

char winchek(){//checks for matching 'X' or 'O' vertically horizontally and diagonally

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

return 'X';

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

return 'X';

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

return 'X';

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

return 'X';

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

return 'X';

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

return 'X';

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

return 'X';

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

return 'X';//return winner

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

return 'O';

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

return 'O';

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

return 'O';

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

return 'O';

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

return 'O';

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

return 'O';

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

return 'O';

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

return 'O';//return winner

return 'C';//if there is no winner then 'C' is returned

};