Project 1

<Connect 4>

CSC – 5, 42644

Celestino, Xyril

04/18/2017

**Introduction**

Title: Connect 4

Connect 4 is played on a 6x7 sized board. Each player must match 4 units of their respective symbol in a row to win the game. The way in which the row of four can be either placed vertically, horizontally, diagonally going down, and diagonally going up. Each player takes turns dropping their pieces into the board, with the board filling from the bottom to the top. Each player must try to get a row of 4 of their pieces, while blocking and preventing their opponent from getting their own pieces to line up into rows of 4.

If neither player is able to score and the board has no more room to input more pieces, the game is ended in a draw.

**Program Size**

Project size = ~140 lines

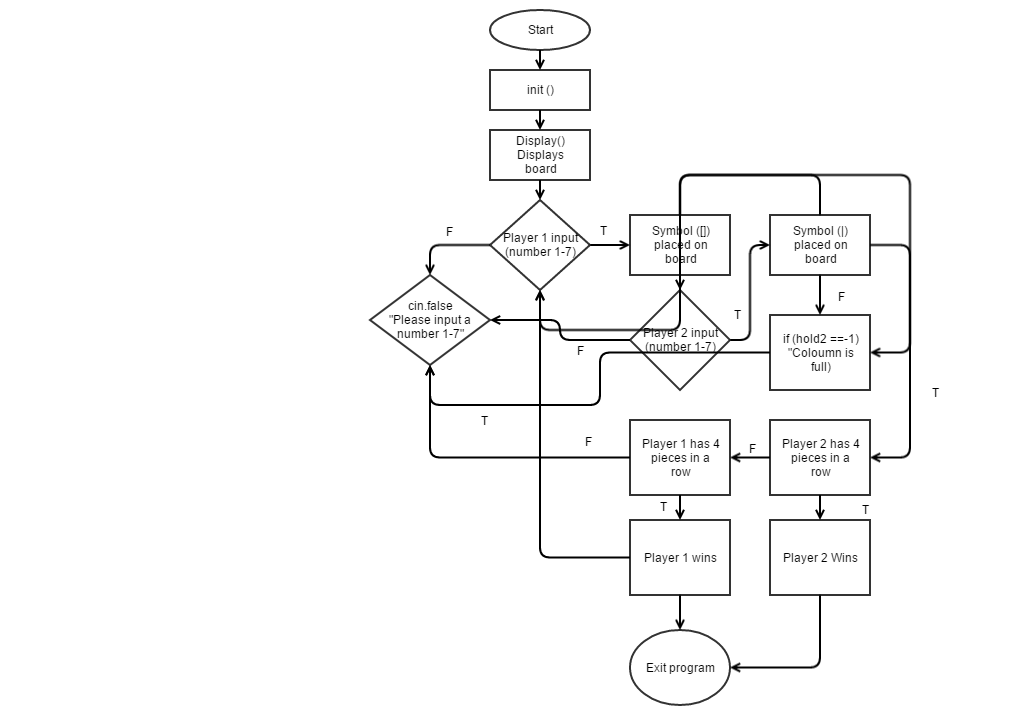
Number of Variables: aprox 12

Number of Functions: aprox 20

**Summary**

Uses multiple if, else, do, while statements throughout. Also includes bool statements used as practice for bool statements.

**Flow Chart**



**Psuedo Code**

*Initialize*

*Display Gameboard*

*If Player 1 insert game piece (1-7)*

*Show player 1 game piece on the board*

*Else If Player 2 insert game piece (1-7)*

*Show player 2 game piece on the board*

*If Number inputted is less than 1 and greater than 7*

*Else Ask user to input number between 1 and 7*

*Start again until player wins*

*If no row equals 4*

*Cout “Draw”*

*If Player 2 wins game with 4 in row*

*Else Player 1 wins game*

*Void display*

*Create horizontal line*

*Create vertical line*

*Bool Check for wins*

*Check for vertical wins*

*Check for horizontal wins*

*Check for diagonal wins (\)*

*Check for diagonal wins (/)*

*Int drop pieces*

*If piece is placed, inputs player character*

**Full Code (Minus Function Definitions)**

/\*

\* File: Project1\_Connect4

\* Author: Xyril Celestino

\* Created on April 16, 2017, 11:01 PM

\* Purpose: Connect 4 game.

\*/

//System Libraries

#include <iostream>

using namespace std;

//User Libraries

//Global Constants

//Function Prototypes

void display();

bool check(int a, int b);

int drop(int b,char player);

char place[6][7];

//Execution begins here

int main(int argc, char\*\* argv) {

for (int a=0;a<=5;a++){

for (int b=0;b<=6;b++)

place[a][b] = ' ';

}

display();

int hold;

int hold2 = 0;

int PcPlace = 0;

bool gamewin = false;

char player = 15;

while(!gamewin){

if(hold2 != -1){

if(player == 15){

cout<<"Player 1 choose where to drop piece";

player = 254;

}

else{

cout<<"Player 2 choose where to drop piece";

player = 15;

}

}

while(true){

if(PcPlace == 42) break;

cin>>hold;

hold--;

if(hold <=6 && hold>= 0) break;

else cout<<"please enter a value between 1 and 7 = ";

if (cin.fail())

{

cin.clear();

char c;

cin>>c;

}

}

if (PcPlace == 42) break;

hold2 = drop (hold, player);

if (hold2 == -1) cout<<"Column is full,",

"please input number between 1 and 7 = ";

else{

gamewin = check(hold2,hold);

PcPlace ++;

cout<<endl<<endl<<endl;

display();

}

}

cout<<endl<<endl<<endl;

if (PcPlace == 42){

cout<<"Draw! No one wins!"<<endl;

return 0;

}

if(player == 15)

cout<<"The Winner is...Player 2!"<<endl;

else cout<<"The Winner is...Player 1!"<<endl;

return 0;

}

void display(){

cout<<" Connect 4! "<<endl;

cout<<" Player 1 = [] "<<endl;

cout<<" Player 2 = | "<<endl;

cout<<"1 2 3 4 5 6 7"<<endl;

for(int a = 0; a<= 5; a++)

{

for(int b=0; b <=6; b++){

cout<<"----";

}

cout<<endl;

for(int b=0; b <=6; b++){

cout<<"|"<<place[a][b]<<"|"<<" ";

//cout<<endl;

}

cout<<endl;

for(int b=0; b <=6; b++){

//cout<<"\_";

}

}

}

bool check(int a, int b){

int vertical = 1;

int horizontal = 1;

int diagonal1 = 1;

int diagonal2 = 1;

char player = place[a][b];

int i;

int ii;

//Check for Verticals

for(i=a -1;place[i][b] == player && i >=0;i--,vertical++);

for(i=a +1;place[i][b] == player && i <=6;i++,vertical++);

if (vertical >= 4) return true;

//Check for Horizontals

for(ii=b -1;place[a][ii] == player && ii >=0;ii--,horizontal++);

for(ii=b +1;place[a][ii] == player && ii <=6;ii++,horizontal++);

if (horizontal >= 4) return true;

//Check for Diagonal 1 (\)

for(i = a -1, ii= b -1;place[i][ii] == player && i>=0 && ii >=0; diagonal1 ++, i --, ii --);

for(i = a +1, ii = b+1;place[i][ii] == player && i<=5 && ii <=6;diagonal1 ++, i ++, ii ++);

if(diagonal1 >= 4) return true;

//Check for Diagonal 2 (/)

for(i = a -1, ii= b +1;place[i][ii] == player && i>=0 && ii <= 6; diagonal2 ++, i --, ii ++);

for(i = a +1, ii= b -1;place[i][ii] == player && i<=5 && ii >=0; diagonal2 ++, i ++, ii --);

if(diagonal2 >= 4) return true;

return false;

}

int drop(int b, char player){

if(b >= 0 && b <= 6)

{

if(place[0][b] == ' '){

int i;

for(i = 0; place[i][b] == ' '; i++)

if(i == 5){place[i][b] = player;

return i;}

i--;

place[i][b] = player;

return i;

}

else{

return -1;

}

}

else{

return -1;

}

}

**Acknowledgement**

* Dr. Mark Lehr
* Jeff Thomas
* Augie’s Coffee Roasters
* RCC Computer Lab

//Exit Program