Project of PF lab

#include <iostream>

#include <stdlib.h>

#include <windows.h>

#define HEIGHT 6

#define WIDTH 7

using namespace std;

void draw\_board();

void player\_movement(int player);

bool check\_for\_winner(int x, int y, int player);

bool check\_diagonal\_combo\_SW\_NE(int x, int y, int player);

bool check\_diagonal\_combo\_NW\_SE(int x, int y, int player);

bool check\_vertical\_combo(int x, int y, int player);

bool check\_horizontal\_combo(int x, int y, int player);

int board\_info[HEIGHT][WIDTH] = { {0,0,0,0,0,0,0},

{0,0,0,0,0,0,0},

{0,0,0,0,0,0,0},

{0,0,0,0,0,0,0},

{0,0,0,0,0,0,0},

{0,0,0,0,0,0,0} };

int LastMoveX, LastMoveY;

int main()

{

int counter = 0;

bool winner = false;

srand(GetTickCount());

cout << "Please select a number from 1-7" << endl;

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

cout << "---------------------";

draw\_board();

for (int i = 0; i < 21; i++)

{

player\_movement(1);

draw\_board();

winner = check\_for\_winner(LastMoveX, LastMoveY, 1);

if (winner)

{

cout << "\nYou Win" << endl;

break;

}

player\_movement(2);

draw\_board();

winner = check\_for\_winner(LastMoveX, LastMoveY, 2);

if (winner)

{

cout << "\nYou Win" << endl;

break;

}

}

system("PAUSE");

return 0;

}

void draw\_board()

{

cout << endl;

for (int y = 0; y < HEIGHT; y++)

{

for (int x = 0; x < WIDTH; x++)

{

cout << "| ";

if (board\_info[y][x] == 0) cout << " ";

else if (board\_info[y][x] == 1) cout << "X";

else if (board\_info[y][x] == 2) cout << "O";

}

cout << "\n---------------------" << endl;

}

}

void player\_movement(int player)

{

int choice;

cout << "\nPlayer" << player << ", please select a number from 1 - 7: ";

cin >> choice;

if (cin.fail())

{

cout << "Error!";

exit(1);

}

while (choice > WIDTH || choice <= 0)

{

cout << "\nPlease select again: ";

cin >> choice;

}

int number = 0;

while (board\_info[(HEIGHT - 1) - number][(choice - 1)] != 0)

{

number++;

if (number > (HEIGHT - 1))

{

cout << "\nPlease select again: ";

cin >> choice;

number = 0;

}

};

board\_info[(HEIGHT - 1) - number][choice - 1] = player;

LastMoveY = (HEIGHT - 1) - number;

LastMoveX = choice - 1;

}

bool check\_for\_winner(int x, int y, int player)

{

bool winner;

if (check\_diagonal\_combo\_SW\_NE(x, y, player)) return true;

else if (check\_diagonal\_combo\_NW\_SE(x, y, player)) return true;

else if (check\_vertical\_combo(x, y, player)) return true;

else if (check\_horizontal\_combo(x, y, player)) return true;

else return false;

}

bool check\_diagonal\_combo\_SW\_NE(int x, int y, int player)

{

int score = 1;

int count = 1;

while ((y - count >= 0) && (x + count < WIDTH))

{

if (board\_info[y - count][x + count] == player)

{

score++;

count++;

}

else break;

}

count = 1;

while ((y + count < HEIGHT) && (x - count >= 0))

{

if (board\_info[y + count][x - count] == player)

{

score++;

count++;

}

else break;

}

if (score == 4) return true;

else return false;

}

bool check\_diagonal\_combo\_NW\_SE(int x, int y, int player)

{

int score = 1;

int count = 1;

while ((y + count >= 0) && (x + count < WIDTH))

{

if (board\_info[y + count][x + count] == player)

{

score++;

count++;

}

else break;

}

count = 1;

while ((y - count < HEIGHT) && (x - count >= 0))

{

if (board\_info[y - count][x - count] == player)

{

score++;

count++;

}

else break;

}

if (score == 4) return true;

else return false;

}

bool check\_vertical\_combo(int x, int y, int player)

{

int score = 1;

int count = 1;

while (y + count >= 0 && y + count < HEIGHT)

{

if (board\_info[y + count][x] == player)

{

score++;

count++;

}

else break;

}

if (score == 4) return true;

else return false;

}

bool check\_horizontal\_combo(int x, int y, int player)

{

int score = 1;

int count = 1;

while ((x + count >= 0) && (x + count < WIDTH))

{

if (board\_info[y][x + count] == player)

{

score++;

count++;

}

else break;

}

count = 1;

while ((x - count < WIDTH) && (x - count >= 0))

{

if (board\_info[y][x - count] == player)

{

score++;

count++;

}

else break;

}

if (score == 4) return true;

else return false;

}

Result

