

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



NHÓM 22- MÔN NHẬP LẬP TRÌNH
LỚP 20CTT3B

BÁO CÁO ĐỒ ÁN CUỐI KÌ
GAME CARO TRÊN CONSOLE WINDOW

Giáo viên hướng dẫn:
TRƯỜNG TOÀN THỊNH

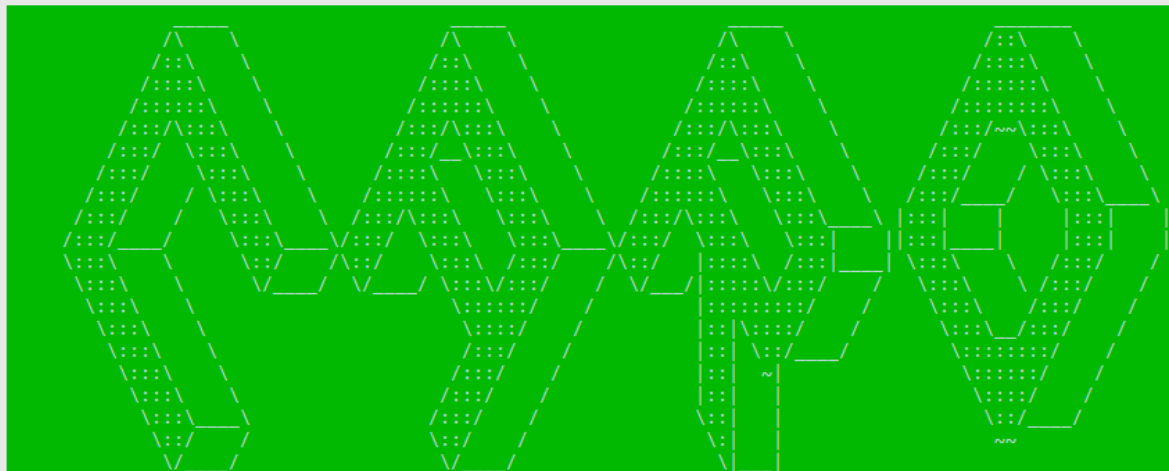
Môn học: Nhập môn Lập Trình

Thành Phố Hồ Chí Minh -2020

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



NHÓM 22- MÔN NHẬP MÔN LẬP TRÌNH
LỚP 20CTT3B



Thành viên thực hiện:

20120442 – Nguyễn Hữu Chính

20120452 – Đinh Viết Danh

18120445 – Hoàng Nguyễn Hải Long

20120478 – Nguyễn Vũ Hiếu

20120489 – Võ Phi Hùng

MỤC LỤC:

MỤC LỤC:	3
1 GIỚI THIỆU ĐỀ TÀI	5
1.1 Lí do làm đề tài	5
1.2 Yêu cầu:	5
1.2.1 Xử lý lưu/tải trò chơi (save/load)	5
1.2.2 Nhận biết thắng/thua/hòa.	5
1.2.3 Xử lý hiệu ứng thắng/thua/hòa.	5
1.2.4 Xử lý giao diện màn hình khi chơi.	5
1.2.5 Xử lý màn hình chính.	5
1.3 Hướng giải quyết vấn đề:	6
2 NỘI DUNG CHI TIẾT	6
2.1 Giới thiệu trò chơi:	6
2.2 Thêm một số hàm phụ trợ:	6
2.2.1 Hàm tô màu:	6
2.2.2 Hàm in theo tọa độ:	6
2.3 Xử lý lưu/tải trò chơi (save/load):	6
2.3.1 Hàm lưu game:	7
2.3.2 Hàm Load board:	8
2.3.3 Hàm load game:	9
2.4 Nhận biết thắng/thua/hòa:	9
2.4.1 Hàm kiểm tra hòa:	9
2.4.2 Nhóm hàm kiểm tra hàng, cột, chéo:	10
2.4.3 Hàm kiểm tra thắng:	13
2.5 Xử lý hiệu ứng thắng/thua/hòa:	14
2.5.1 Hàm X thắng:	14
2.5.2 Hàm O thắng:	14
2.5.3 Hàm hòa:	15
2.6 Xử lý giao diện màn hình khi chơi:	16
2.6.1 Hàm cập nhật thông tin ván đấu:	16
2.7 Xử lý màn hình chính:	17
2.7.1 Nhóm hàm hiển thị:	17

2.7.2	Hàm di chuyển và lựa chọn ngoài game.....	22
2.7.3	Hàm lựa chọn luật chơi:.....	24
2.7.4	Hàm Menu chính	24
2.8	Hàm bắt đầu game mới:	26
2.9	Hàm Main:	28
3	Lời kết:.....	28

1 GIỚI THIỆU ĐỀ TÀI

1.1 Lí do làm đề tài

Ngày nay công nghệ thông tin trở thành một lĩnh vực mũi nhọn trong công cuộc phát triển kinh tế xã hội. Cùng với công nghệ sinh học và năng lượng mới, công nghệ thông tin (CNTT) vừa là công cụ, vừa là động lực thúc đẩy quá trình công nghiệp hóa, hiện đại hóa đất nước.

Do đó, nhu cầu giải trí của con người cũng phải tăng cao. Có thể nói Game đã và đang trở thành một nhu cầu giải trí thiết yếu trong cuộc sống của con người hiện nay. Hoạt động về Game đã xuất hiện ở Việt Nam rất lâu và nó bắt nguồn từ những Game đơn giản và thực tế, thậm chí có nguồn gốc trước khi chưa có máy tính.

- Càng ngày nó càng trở nên thu hút người chơi và trở thành một môn thể thao điện tử ở nước ta. Và CARO cũng không phải là ngoại lệ, đã và đang là một môn thể thao, là một game đình đám thể hiện trí não của con người. Với luật chơi dễ hiểu và dễ tiếp cận, Caro là game đơn giản dễ chơi và dễ lập trình cho những sinh viên mới tiếp cận lập trình.

1.2 Yêu cầu:

Bổ sung một số chức năng vào mã nguồn giáo viên hướng dẫn đã cung cấp, bao gồm:

1.2.1 Xử lý lưu/tải trò chơi (save/load)

Trong hướng dẫn chưa xử lý việc lưu/tải trò chơi. Ta cần cài đặt thêm tính năng này. Khi người dùng nhấn phím 'L' thì sẽ hiện dòng chữ yêu cầu người dùng nhập tên tập tin muốn lưu trạng thái hiện hành của trò chơi. Khi người dùng nhấn phím 'T' thì sẽ hiện dòng chữ yêu cầu người dùng nhập tên tập tin muốn tải lại.

1.2.2 Nhận biết thắng/thua/hòa.

Ta cần bổ sung tính năng kiểm tra quy luật thắng/thua /hòa trong caro. Từ đó sẽ kiểm tra sau mỗi bước đi của người chơi.

1.2.3 Xử lý hiệu ứng thắng/thua/hòa.

Trong hướng dẫn, khi thắng/thua/hòa chỉ hiển thị dòng chữ báo hiệu đơn giản. Ta hãy cài đặt hiệu ứng giúp sinh động hơn.

1.2.4 Xử lý giao diện màn hình khi chơi.

Trong quá trình chơi, cho hiển thị các thông số của hai người chơi, ví dụ người chơi thứ nhất đã đánh bao nhiêu bước, người chơi thứ hai đã thua mấy ván... Sinh viên tự tổ chức giao diện màn hình sao cho rõ ràng, sinh động.

1.2.5 Xử lý màn hình chính.

Trước khi vào trò chơi, hiển thị danh sách menu, ví dụ như "New Game", "Load Game", "Settings", ... Như vậy sẽ giúp chương trình caro hoàn thiện và giống thực tế một trò chơi hơn.

1.3 Hướng giải quyết vấn đề:

Các thành viên xem lại các kiến thức đã học trên lớp

Tham khảo các tài liệu, nguồn Internet, giáo trình,.. chính thống.

Thảo luận và đưa ra giải pháp.

2 NỘI DUNG CHI TIẾT

2.1 Giới thiệu trò chơi:

Các chức năng chính của trò chơi bao gồm:

- Bắt đầu trò chơi
- Save/Load game
- Lựa chọn chế độ game
- Bảng thành tích
- Hướng dẫn
- Thoát

Về phần trò chơi, người chơi sẽ sử dụng các phím W/A/D/S hoặc các phím mũi tên để di chuyển trên bàn cờ, phím Enter/Space để đánh dấu X/O vào bàn cờ Caro, phím Esc để sử dụng các chức năng phụ khi cần thiết. Sau đây là giới thiệu về trò chơi cũng như các nhóm hàm mà trò chơi cần.

2.2 Thêm một số hàm phụ trợ:

2.2.1 Hàm tô màu:

1	<code>void textcolor(int x)</code>
2	{
3	HANDLE mau;
4	mau = GetStdHandle(STD_OUTPUT_HANDLE);
5	SetConsoleTextAttribute(mau, x);
6	}

Hàm này giúp ta tô màu các ký tự được in ra màn hình sau khi gọi hàm. X là số nguyên, mỗi số nguyên tương ứng với mỗi màu khác nhau, ta có thể tùy theo nhu cầu mà lựa chọn.

2.2.2 Hàm in theo tọa độ:

1	<code>void PrintText(string text, int x, int y)</code>
2	{
3	GotoXY(x, y);
4	cout << text;
5	}

Di chuyển tới vị trí có tọa độ x, y và in ra chuỗi text.

2.3 Xử lý lưu/tải trò chơi (save/load):

Trong mục này ta sẽ sử dụng các hàm trong một thư viện chuẩn trong C++ là **fstream**:

Kiểu dữ liệu	Miêu tả
ofstream	Kiểu dữ liệu này biểu diễn Output File Stream và được sử dụng để tạo các file và để ghi thông tin tới các file đó
ifstream	Kiểu dữ liệu này biểu diễn Input File Stream và được sử dụng để đọc thông tin từ các file
fstream	Kiểu dữ liệu này nói chung biểu diễn File Stream, và có các khả năng của cả ofstream và ifstream, nghĩa là nó có thể tạo file, ghi thông tin tới file và đọc thông tin từ file

2.3.1 Hàm lưu game:

1	<code>int SaveGame()</code>
2	<code>{</code>
3	<code> string filename, files[100];</code>
4	<code> PrintText("NHAP TEN FILE: ", 70, 10);</code>
5	<code> cin >> filename;</code>
6	<code> filename += ".txt";</code>
7	<code> ofstream file(filename);</code>
8	<code> for (int i = 0; i < BOARD_SIZE; i++)</code>
9	<code> {</code>
10	<code> for (int j = 0; j < BOARD_SIZE; j++)</code>
11	<code> file << _A[j][i].c << " ";</code>
12	<code> file << endl;</code>
13	<code> }</code>
14	<code> file << _TURN;</code>
15	<code> file.close();</code>
16	<code> ifstream listfiles("listFiles.txt");</code>
17	<code> int line = 0;</code>
18	<code> listfiles >> line;</code>
19	<code> for (int i = 0; i < line; i++)</code>
20	<code> {</code>
21	<code> listfiles >> files[i];</code>
22	<code> if (filename == files[i])</code>
23	<code> {</code>
24	<code> listfiles.close();</code>

25	<code>return 0;</code>
26	<code>}</code>
27	<code>}</code>
28	<code>listfiles.close();</code>
29	<code>ofstream _listfiles("listFiles.txt"); //mở file để ghi</code>
30	<code>_listfiles << line + 1 << endl; //tăng số dòng</code>
31	<code>for (int i = 0; i < line; i++)</code>
32	<code>{</code>
33	<code>_listfiles << files[i] << endl;</code>
34	<code>}</code>
35	<code>_listfiles << filename; //thêm file</code>
36	<code>_listfiles.close();</code>
37	<code>return 0;</code>

Từ dòng 2 đến dòng 13 là các lệnh lưu thông tin các đánh dấu của bàn cờ, đây chính là các phần tử của mảng (A[i][j].c), dòng 14 lưu lượt hiện hành. Từ dòng 15 đến dòng 36 lưu tên file vào list file để sau vào loadgame có thể chọn lựa

2.3.2 Hàm Load board:

1	<code>void LoadBoard()</code>
2	<code>{</code>
3	<code>Xmoves = 0; Omoves = 0;</code>
4	<code>for (int i = 0; i < BOARD_SIZE; i++)</code>
5	<code>for (int j = 0; j < BOARD_SIZE; j++)</code>
6	<code>{</code>
7	<code>_A[i][j].x = 4 * j + LEFT + 2;</code> <code>// Trùng với hoành độ màn hình bàn cờ</code>
8	<code>_A[i][j].y = 2 * i + TOP + 1;</code> <code>// Trùng với tung độ màn hình bàn cờ</code>
9	<code>if (_A[i][j].c == -1)</code>
10	<code>PrintText("X", _A[i][j].x, _A[i][j].y); Xmoves++;</code>
11	<code>else if (_A[i][j].c == 1</code>
12	<code>PrintText("O", _A[i][j].x, _A[i][j].y); Omoves++;</code>
13	<code>}</code>
14	<code>UpDate(Xmoves, Omoves, Xwins, Owins);</code>
15	<code>_X = _A[0][0].x; _Y = _A[0][0].y;</code>
16	<code>GotoXY(_X, _Y);</code>

17	}
----	---

Dựa vào mảng các phần tử đánh dấu ma trận ta sẽ suy ra được tọa độ x, y theo và sử dụng hàm PrintText đi đến các điểm đó và in các phần tử theo đúng thứ tự được lưu. (từ dòng 4 đến dòng 13)

2.3.3 Hàm load game:

1	<code>int LoadGame(string filename)</code>
2	{
3	<code>ifstream file(filename);</code>
4	<code>for (int i = 0; i < BOARD_SIZE; i++)</code>
5	<code>for (int j = 0; j < BOARD_SIZE; j++)</code>
6	<code>file >> _A[j][i].c;</code>
7	<code>file >> _TURN;</code>
8	<code>file.close();</code>
9	<code>LoadBoard();</code>
10	<code>return 1;</code>

Từ dòng 3 đến dòng 6 tải lên từ file đã lưu các phần tử đánh dấu bàn cờ. Dòng 6 truyền vào lượt đi tiếp theo, dòng 9 sử dụng hàm LoadBoard() đã xây dựng để vẽ các điểm X, O theo đúng đánh dấu.

2.4 Nhận biết thắng/thua/hòa:

2.4.1 Hàm kiểm tra hòa:

Thuật toán của hàm này là kiểm tra đánh dấu của tất cả các ô trên bàn cờ, nếu tất cả đều đã đánh thì xem như hòa.

1	<code>int CheckDraw()</code>
2	{
3	<code>int count = 0;</code>
4	<code>for (int i = 0; i < BOARD_SIZE; i++)</code>
5	{
6	<code>for (int j = 0; j < BOARD_SIZE; j++)</code>
7	{
8	<code>if (_A[i][j].c != 0)</code>
9	<code>{ count++; }</code>
10	}
11	<code>if (count == BOARD_SIZE * BOARD_SIZE)</code>
12	<code>{ return 1;}</code>

13	<code>return 0;</code>
14	<code>}</code>

2.4.2 Nhóm hàm kiểm tra hàng, cột, chéo:

Thuật toán hàm này là từ điểm vừa đánh, ta kiểm tra xung quanh nó. Ví dụ theo hàng, ta sẽ kiểm tra bên trái của điểm vừa đánh, khởi tạo 1 biến đếm và nếu gặp đánh dấu giống nó, cộng biến đếm lên 1, dừng khi gặp đánh dấu khác nó, tiếp tục kiểm tra bên phải. Sau khi kiểm tra trái phải, nếu biến đếm lớn hơn bằng 4 thì xác nhận người chơi lượt vừa đánh win, các hàm kiểm tra cột, đường chéo là tương tự.

Mục này được bổ sung thêm so với yêu cầu đề án: người chơi sẽ được chọn luật chơi có hoặc không chặn hai đầu ở menu chính. Biến LuatCHD là 1 khi áp dụng chặn 2 đầu không thắng và là 0 khi không chơi chặn hai đầu. cuối mỗi hàm check ta thấy có 1 câu lệnh if, câu lệnh này trả về kết quả theo đúng luật mà người chơi chọn.

Hàm kiểm tra hàng

1	<code>int CheckRow(int pi, int pj, int LuatCHD)</code>
2	<code>{</code>
3	<code>int i = pi, j = pj, count = 0, Dem = 0;</code>
4	<code>while ((j - 1 >= 0) && ((_A[i][j - 1].c == 1 && _TURN == false)</code> <code> (_A[i][j - 1].c == -1 && _TURN == true))))</code>
5	<code>{</code>
6	<code>count++;</code>
7	<code>j--;</code>
8	<code>}</code>
9	<code>if ((j - 1 >= 0) && ((_A[i][j - 1].c == -1 && _TURN == false)</code> <code> (_A[i][j - 1].c == 1 && _TURN == true))))</code>
10	<code>Dem++;</code>
11	<code>i = pi;</code>
12	<code>j = pj;</code>
13	<code>return 0;</code>
14	<code>while ((j + 1 <= (BOARD_SIZE - 1))</code> <code>&& ((_A[i][j + 1].c == 1 && _TURN == false)</code> <code> (_A[i][j + 1].c == -1 && _TURN == true))))</code>
15	<code>{</code>
16	<code>count++;</code>
17	<code>j++;</code>
18	<code>}</code>
19	<code>if ((j + 1 <= (BOARD_SIZE - 1))</code> <code>&& ((_A[i][j + 1].c == -1 && _TURN == false)</code> <code> (_A[i][j + 1].c == 1 && _TURN == true))))</code>
20	<code>Dem++;</code>
21	<code>if ((count == 4 && (!LuatCHD (LuatCHD && Dem != 2)))</code> <code> count > 4)</code>

22	<code>return 1;</code>
23	<code>else return 0;</code>
24	<code>}</code>

Hàm kiểm tra cột

1	<code>int CheckColumn(int pi, int pj, int LuatCHD)</code>
2	<code>{</code>
3	<code>int i = pi, j = pj, count = 0, Dem = 0;</code>
4	<code>while ((i - 1 >= 0) && ((_A[i - 1][j].c == 1 && _TURN == false)</code> <code> (_A[i - 1][j].c == -1 && _TURN == true)))</code>
5	<code>{</code>
6	<code>count++;</code>
7	<code>i--;</code>
8	<code>}</code>
9	<code>if ((i - 1 >= 0) && ((_A[i - 1][j].c == -1 && _TURN == false)</code> <code> (_A[i - 1][j].c == 1 && _TURN == true)))</code>
10	<code>Dem++;</code>
11	<code>i = pi;</code>
12	<code>j = pj;</code>
13	<code>return 0;</code>
14	<code>while ((i + 1 <= (BOARD_SIZE - 1))</code> <code>&& ((_A[i + 1][j].c == 1 && _TURN == false)</code> <code> (_A[i + 1][j].c == -1 && _TURN == true)))</code>
15	<code>{</code>
16	<code>count++;</code>
17	<code>i++;</code>
18	<code>}</code>
19	<code>if ((i + 1 <= (BOARD_SIZE - 1))</code> <code>&& ((_A[i + 1][j].c == -1 && _TURN == false)</code> <code> (_A[i + 1][j].c == 1 && _TURN == true)))</code>
20	<code>Dem++;</code>
21	<code>if ((count == 4 && (!LuatCHD (LuatCHD && Dem != 2)))</code> <code> count > 4)</code>
22	<code>return 1;</code>
23	<code>else return 0;</code>
24	<code>}</code>

Hàm kiểm tra đường chéo

1	<code>int Check1Cross(int pi, int pj, int LuatCHD)</code>
2	<code>{</code>
3	<code>int i = pi, j = pj, count = 0, Dem = 0;</code>
4	<code>while (((i - 1 >= 0) && (j - 1 >= 0))</code> <code>&& ((_A[i-1][j - 1].c == 1 && _TURN == false)</code> <code> (_A[i-1][j - 1].c == -1 && _TURN==true)))</code>
5	<code>{</code>
6	<code>count++;</code>
7	<code>i--;</code>
	<code>j--;</code>
8	<code>}</code>
9	<code>if (((i - 1 >= 0) && (j - 1 >= 0))</code> <code>&& ((_A[i - 1][j - 1].c== -1 && _TURN == false)</code> <code> (_A[i-1][j - 1].c == 1 && _TURN==true)))</code>
10	<code>Dem++;</code>
11	<code>i = pi;</code>
12	<code>j = pj;</code>
13	<code>return 0;</code>
14	<code>while (((i + 1 <= (BOARD_SIZE - 1))</code> <code>&& (j + 1 <= (BOARD_SIZE - 1)))</code> <code>&& ((_A[i + 1][j + 1].c == 1 && _TURN == false)</code> <code> (_A[i+1][j + 1].c == -1 && _TURN == true)))</code>
15	<code>{</code>
16	<code>count++;</code>
17	<code>i++;</code>
	<code>j--;</code>
18	<code>}</code>
19	<code>if (((i + 1 <= (BOARD_SIZE - 1))</code> <code>&& (j + 1 <= (BOARD_SIZE - 1)))</code> <code>&& ((_A[i + 1][j + 1].c == -1 && _TURN == false)</code> <code> (_A[i+1][j + 1].c == 1 && _TURN == true)))</code>
20	<code>Dem++;</code>
21	<code>if ((count == 4 && (!LuatCHD (LuatCHD && Dem != 2)))</code> <code> count > 4)</code>
22	<code>return 1;</code>
23	<code>else return 0;</code>
24	<code>}</code>

Hàm kiểm tra đường chéo phụ

1	<code>int Check1Cross(int pi, int pj, int LuatCHD)</code>
2	<code>{</code>
3	<code>int i = pi, j = pj, count = 0, Dem = 0;</code>
4	<code>while (((i - 1 >= 0) && (j + 1 <= (BOARD_SIZE - 1)))</code> <code>&& ((_A[i-1][j + 1].c == 1 && _TURN == false)</code> <code> (_A[i-1][j+1].c == -1 && _TURN == true)))</code>
5	<code>{</code>
6	<code>count++;</code>
7	<code>i--;</code>
	<code>j--;</code>
8	<code>}</code>
9	<code>if (((i-1 >= 0) && (j + 1 <= (BOARD_SIZE - 1)))</code> <code>&& ((_A[i-1][j + 1].c == -1 && _TURN == false)</code> <code> (_A[i - 1][j + 1].c == 1 && _TURN == true)))</code>
10	<code>Dem++;</code>
11	<code>i = pi;</code>
12	<code>j = pj;</code>
13	<code>return 0;</code>
14	<code>while (((i + 1 <= (BOARD_SIZE - 1)) && (j - 1 >= 0))</code> <code>&& ((_A[i + 1][j - 1].c == 1 && _TURN == false)</code> <code> (_A[i + 1][j - 1].c == -1 && _TURN == true)))</code>
15	<code>{</code>
16	<code>count++;</code>
17	<code>i++;</code>
	<code>j--;</code>
18	<code>}</code>
19	<code>if (((i + 1 <= (BOARD_SIZE - 1)) && (j - 1 >= 0))</code> <code>&& ((_A[i + 1][j - 1].c == -1 && _TURN == false)</code> <code> (_A[i + 1][j - 1].c == 1 && _TURN == true)))</code>
20	<code>Dem++;</code>
21	<code>if ((count == 4 && (!LuatCHD (LuatCHD && Dem != 2)))</code> <code> count > 4)</code>
22	<code>return 1;</code>
23	<code>else return 0;</code>
24	<code>}</code>

2.4.3 Hàm kiểm tra thắng:

Trả về các giá trị kiểm tra của 4 trường hợp trên.

1	<code>int CheckWin(int pX, int pY, int LuatCHD)</code>
2	<code>{</code>
3	<code>int _i, _j; //Tìm i, j</code>
4	<code>_j = (_X - 2 - LEFT) / 4;</code>

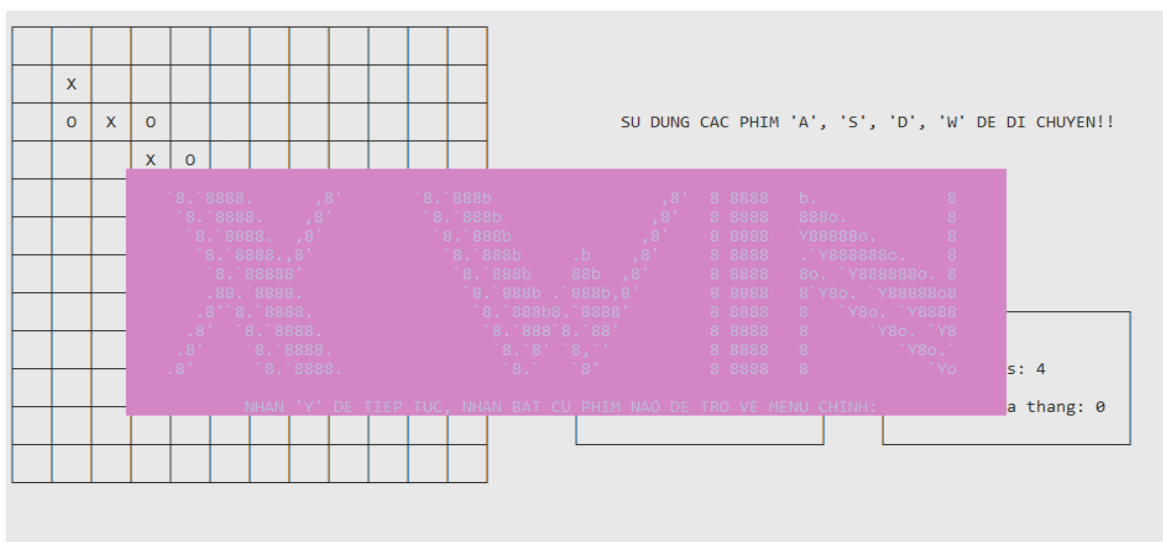
5	<code>_i = (_Y - 1 - TOP) / 2;</code>
6	<code>return CheckRow(_i, _j, LuatCHD)</code> <code> CheckColumn(_i, _j, LuatCHD)</code> <code> Check1Cross(_i, _j, LuatCHD)</code> <code> Check2Cross(_i, _j, LuatCHD);</code>
7	<code>}</code>

2.5 Xử lý hiệu ứng thắng/thua/hòa:

Hiệu ứng này được làm đơn giản là đi đến các điểm thích hợp để in ra màn hình các kết quả của ván đấu, sử dụng hàm Textcolor để màu sắc trở nên đẹp mắt hơn.

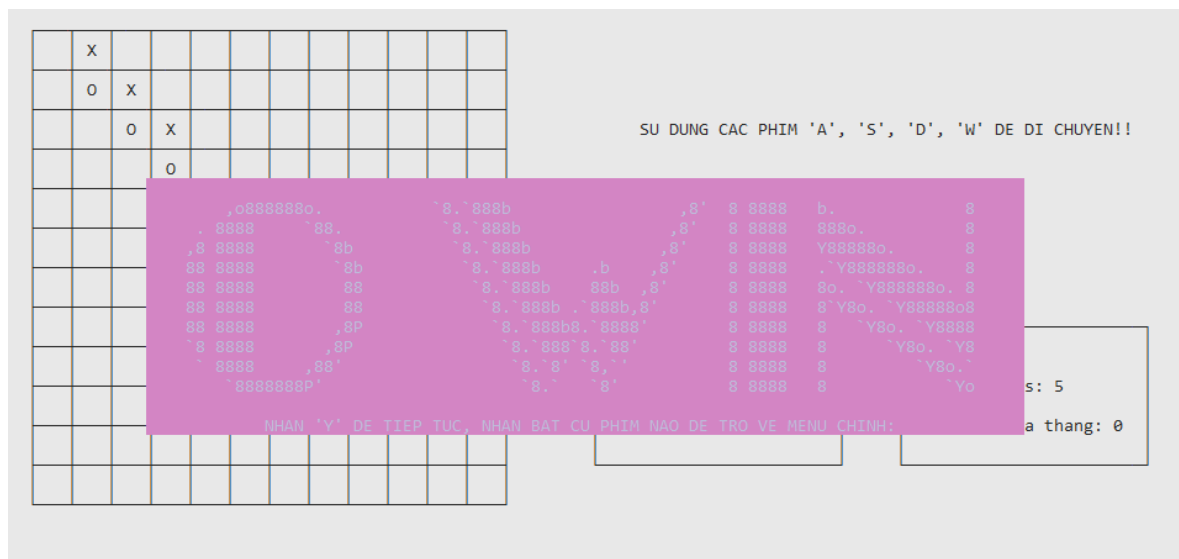
2.5.1 Hàm X thắng:

1	<code>void Xwin()</code>
2	<code>{</code>
3	<code>int x = 15;</code>
4	<code>int y = 10;</code>
5	<code>textcolor(89);</code>
6	<pre>PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 b. 8 ", x, y - 1); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 888o. 8 ", x, y + 0); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 888o. 8 ", x, y + 1); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 Y88888o. 8 ", x, y + 2); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 8o. Y888888o. 8 ", x, y + 3); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 8o. Y888888o. 8 ", x, y + 4); PrintText(" '8.'8888. ,8' '8.'888b ,8' 8 8888 8'Y8o. Y88888o8 ", x, y + 5); PrintText(" '8.'8.'8888. ,8' '8.'888b8.'8888' 8 8888 8 Y8o. Y8888 ", x, y + 6); PrintText(" '8.'8.'8888. ,8' '8.'888b8.'8888' 8 8888 8 Y8o. Y8 ", x, y + 7); PrintText(" '8.'8.'8888. ,8' '8.'8.'8.'8.'8' 8 8888 8 Y8o. Y8 ", x, y + 8); PrintText(" '8.'8.'8888. ,8' '8.'8.'8.'8.'8' 8 8888 8 Y8o. Y8 ", x, y + 9); PrintText(" '8.'8.'8888. ,8' '8.'8.'8.'8.'8' 8 8888 8 Y8o. Y8 ", x, y + 10); PrintText(" '8.'8.'8888. ,8' '8.'8.'8.'8.'8' 8 8888 8 Y8o. Y8 ", x, y + 11);</pre>
7	<code>textcolor(496);</code>
8	<code>}</code>



2.5.2 Hàm O thắng:

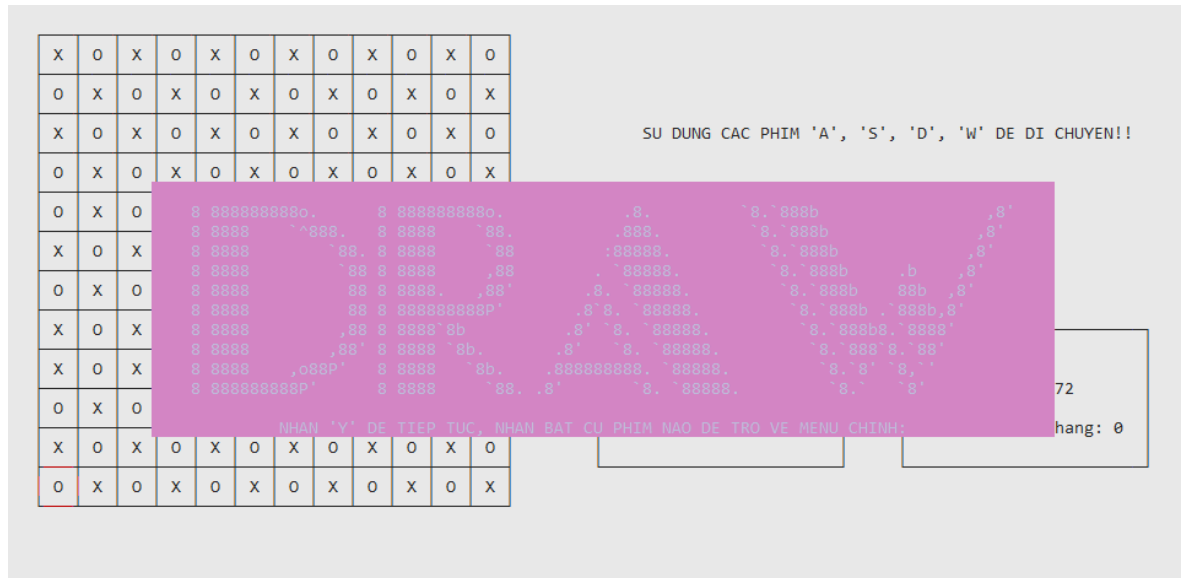
1	<code>void OWin()</code>
2	<code>{</code>
3	<code>int x = 15;</code>
4	<code>int y = 10;</code>
5	<code>textcolor(89);</code>
6	<pre>PrintText(" ,o888888o. '8.'888b ,8' 8 8888 b. 8 ", x, y - 1); PrintText(" 8888 88. '8.'888b ,8' 8 8888 888o. 8 ", x, y + 0); PrintText(" ,8 8888 '8b '8.'888b ,8' 8 8888 Y88888o. 8 ", x, y + 1); PrintText(" 88 8888 '8b '8.'888b ,b ,8' 8 8888 .Y888888o. 8 ", x, y + 2); PrintText(" 88 8888 88 '8.'888b 88b ,8' 8 8888 8o. Y888888o. 8 ", x, y + 3); PrintText(" 88 8888 88 '8.'888b '888b,8' 8 8888 8'Y8o. Y8888888 ", x, y + 4); PrintText(" 88 8888 ,8P '8.'888b8.'8888' 8 8888 8 Y8o. Y8888888 ", x, y + 5); PrintText(" 88 8888 ,8P '8.'888b8.'8888' 8 8888 8 Y8o. Y8888 ", x, y + 6); PrintText(" '8 8888 ,8P '8.'888'8.'88' 8 8888 8 Y8o. Y8 ", x, y + 7); PrintText(" '8888 ,88' '8.'8'8'8.'8' 8 8888 8 Y8o. Y8o. ", x, y + 8); PrintText(" '8888888P' '8.'8'8'8' 8 8888 8 Y8o. Y8o. ", x, y + 9); PrintText(" '8888888P' '8.'8'8'8' 8 8888 8 Y8o. Y8o. ", x, y + 10); PrintText(" NHAN 'Y' DE TIEP TUC, NHAN BAT CU PHIM NAO DE TRO VE MENU CHINH: ", x, y + 11);</pre>
7	<code>textcolor(496);</code>
8	<code>}</code>



2.5.3 Hàm hòa:

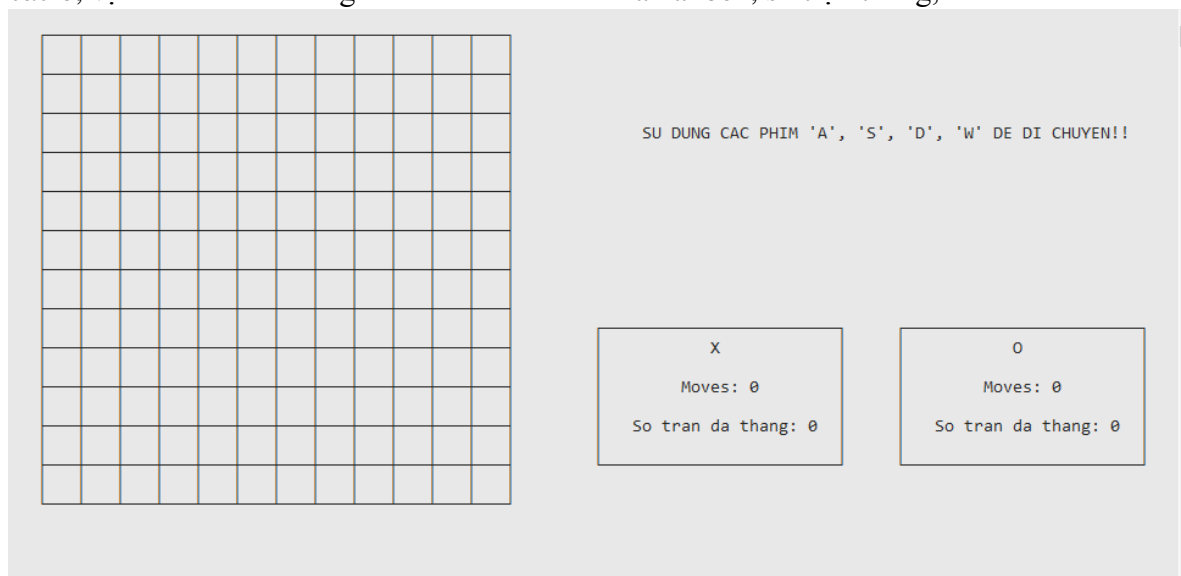
1	<code>void XODraw()</code>
2	<code>{</code>
3	<code>int x = 15;</code>
4	<code>int y = 10;</code>
5	<code>textcolor(89);</code>
6	<pre>PrintText(" 8 888888888o. 8 888888888o. .8. '8.'888b ,8' ", x, y - 1); PrintText(" 8 8888 ^888. 8 8888 88. ,88. '8.'888b ,8' ", x, y + 0); PrintText(" 8 8888 '88. 8 8888 '88 :88888. '8.'888b ,8' ", x, y + 1); PrintText(" 8 8888 '88 8 8888 ,88 .88888. '8.'888b ,b ,8' ", x, y + 2); PrintText(" 8 8888 88 8 8888 ,88' .8.'88888. '8.'888b 88b ,8' ", x, y + 3); PrintText(" 8 8888 88 8 888888888P' .8'8.'88888. '8.'888b8.'8888' ", x, y + 4); PrintText(" 8 8888 ,88 8 8888 8b .8'8.'88888. '8.'888b8.'8888' ", x, y + 5); PrintText(" 8 8888 ,88' 8 8888 '8b. .8'8.'88888. '8.'888b8.'8888' ", x, y + 6); PrintText(" 8 8888 ,o88P' 8 8888 8b. .8888888888. 88888. '8.'8'8'8.'88' ", x, y + 7); PrintText(" 8 888888888P' 8 8888 88. .88. '8.'88888. '8.'8'8'8.'88' ", x, y + 8); PrintText(" 8 888888888P' 8 8888 88. .8'8.'88888. '8.'8'8'8.'88' ", x, y + 9); PrintText(" NHAN 'Y' DE TIEP TUC, NHAN BAT CU PHIM NAO DE TRO VE MENU CHINH: ", x, y + 11);</pre>
7	<code>textcolor(496);</code>

8	}
---	---



2.6 Xử lý giao diện màn hình khi chơi:

Sửa hàm DrawBoard trong hướng dẫn để có giao diện bàn cờ giống thật hơn và có các ô, vị trí để in các thông tin như số nước đi của hai bên, số trận thắng,



2.6.1 Hàm cập nhật thông tin ván đấu:

1	<code>int UpDate(int Xmoves, int Omoves, int Xwins, int Owins)</code>
2	<code>{</code>
3	<code>GotoXY(72, 17); cout << "X";</code>
4	<code>GotoXY(69, 19); cout << "Moves: " << Xmoves;</code>
5	<code>GotoXY(103, 17); cout << "O";</code>
6	<code>GotoXY(100, 19); cout << "Moves: " << Omoves;</code>

7	GotoXY(64, 21); cout << "So tran da thang: " << Xwins;
8	GotoXY(95, 21); cout << "So tran da thang: " << Owins;
9	return 0;
10	}

Hàm Update() có chức năng hiện ra các thông tin của trận đấu như số lượt, số trận



thắng của X và O.

2.7 Xử lý màn hình chính:

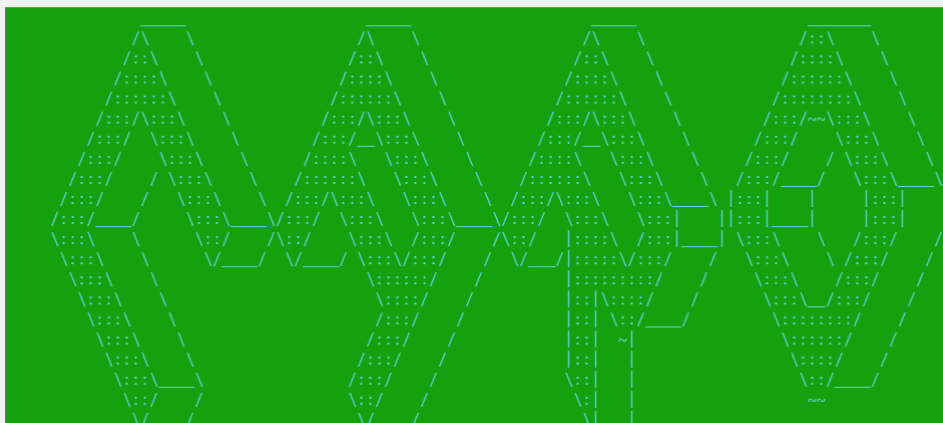
2.7.1 Nhóm hàm hiển thị:

Hàm hiển thị Logo:

1	<code>void DrawBigText()</code>
2	<code>{</code>
3	<code>textcolor(89);</code>
4	<pre>PrintText(" , 5, 1); PrintText(" * 5, 2); PrintText(" * 5, 3); PrintText(" * 5, 4); PrintText(" * 5, 5); PrintText(" * 5, 6); PrintText(" * 5, 7); PrintText(" * 5, 8); PrintText(" * 5, 9); PrintText(" * 5, 10); PrintText("5, 11); PrintText(" * 5, 12); PrintText(" * 5, 13); PrintText(" * 5, 14); PrintText(" * 5, 15); PrintText(" * 5, 16); PrintText(" * 5, 17); PrintText(" * 5, 18); PrintText(" * 5, 19); PrintText(" * 5, 20); PrintText(" * 5, 21);")</pre>
5	<code>textcolor(496);</code>
6	<code>}</code>

Hàm DrawBigText() để in lên mà hình Logo trò chơi ở trang chứa Menu chính

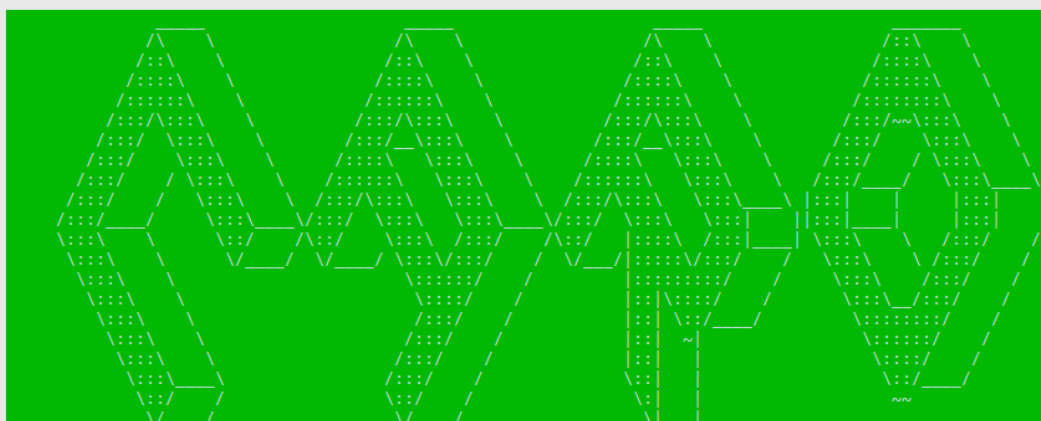
C:\Users\Pavilion\source\repos\Tét\Debug\Tét.exe



Hàm hiển thị Menu

Với menu là mảng `string` `menu[] = { "PLAY", "LOAD GAME", "OPTION", "CREDIT", "HELP", "EXIT" }`; in ra màn hình ở các vị trí thích hợp menu chính.

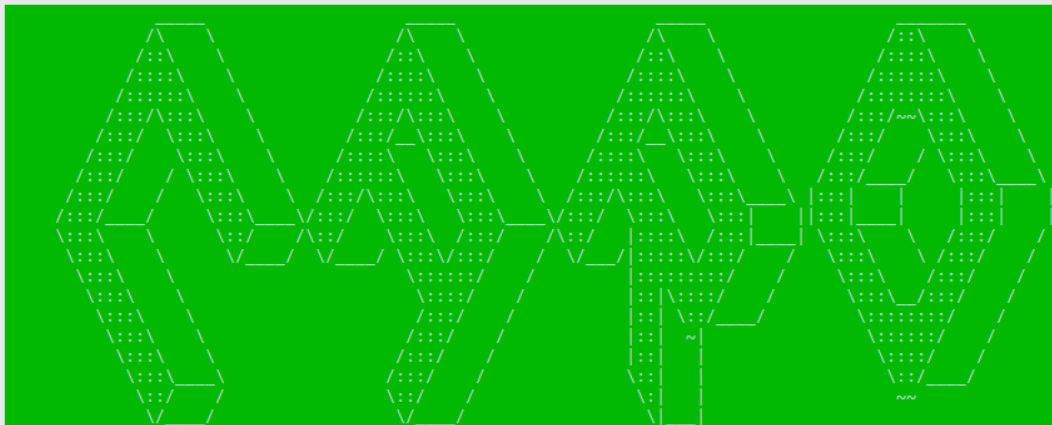
1	<code>int DrawMenu()</code>
2	<code>{</code>
3	<code> textcolor(15);</code>
4	<code> PrintText(menu[0], X, Y);</code>
5	<code> textcolor(496);</code>
6	<code> PrintText(menu[1], X, Y + 1);</code>
7	<code> PrintText(menu[2], X, Y + 2);</code>
8	<code> PrintText(menu[3], X, Y + 3);</code>
9	<code> PrintText(menu[4], X, Y + 4);</code>
10	<code> PrintText(menu[5], X, Y + 5);</code>
11	<code> return 6;</code>
12	<code>}</code>



```
PLAY
LOAD GAME
OPTION
CREDIT
HELP
EXIT
```

Hàm hiển thị Option

1	<code>int DrawOption()</code>
2	<code>{</code>
3	<code> textcolor(15);</code>
4	<code> PrintText(menu[0], X, Y);</code>
5	<code> textcolor(496);</code>
6	<code> PrintText(menu[1], X, Y + 1);</code>
7	<code> return 2;</code>
8	<code>}</code>



KHÔNG ÁP DỤNG LUẬT CHẠN HAI ĐẦU.
ÁP DỤNG LUẬT CHẠN HAI ĐẦU.

Hàm hiển thị Load File

Hàm này để load và in ra file giữ các filename game được lưu để chúng ta lựa chọn tiếp tục game.

1	<code>int DrawLoadFile()</code>
2	<code>{</code>
3	<code> ifstream listfiles("listFiles.txt"); //mở file để đọc</code>
4	<code> int line = 0;</code>
5	<code> listfiles >> line;</code>
6	<code> textcolor(15);</code>
7	<code> for (int i = 0; i < line; i++)</code>
8	<code> {</code>
9	<code> listfiles >> files[i];</code>
10	<code> PrintText(files[i], X, Y + i);</code>
11	<code> textcolor(496);</code>
12	<code> }</code>
13	<code> textcolor(496);</code>
14	<code> listfiles.close();</code>
15	<code> return line;</code>
16	<code>}</code>

```

Chon file game:
test1.txt
test2.txt
test3.txt
test4.txt

```

Hàm hiển thị Credit

1	<code>int Credit()</code>
2	<code>{</code>
3	<code>system("cls");</code>
4	<code>PrintText("CREDIT: ", 50, 14);</code>
5	<code>PrintText("18120445 - HOANG NGUYEN HAI LONG.", 45, 15);</code>
6	<code>PrintText("20120442 - NGUYEN HUU CHINH.", 45, 16);</code>
7	<code>PrintText("20120452 - DINH VIET DANH", 45, 17);</code>
8	<code>PrintText("20120478 - NGUYEN VU HIEU", 45, 18);</code>
9	<code>PrintText("20120489 -VO PHI HUNG", 45, 19);</code>
10	<code>return 0;</code>
11	<code>}</code>

Hàm Credit() có chức năng in ra thông tin về nhóm 22 khi người chơi chọn mục Credit trong Menu chính

Credit:

18120445 - HOANG NGUYEN HAI LONG.
 20120442 - NGUYEN HUU CHINH.
 20120452 - DINH VIET DANH
 20120478 - NGUYEN VU HIEU
 20120489 - VO PHI HUNG

Hàm hiển thị Luật

1	<code>int LawGame()</code>
2	<code>{</code>
3	<code>int x = 20;</code>
4	<code>int y = 5;</code>
5	<code>system("cls");</code>
6	<code>PrintText(" HUONG DAN: ", x, y);</code>
7	<code>PrintText(" Game su dung hai ki hieu 'X','O' de dai dien cho", x, ++y);</code>
8	<code>PrintText(" Su dung ban co kích thước 12x12. ", x, ++y);</code>
9	<code>PrintText(" Tren mot ban co chi cho phép hai người chơi tham gia. ", x, ++y);</code>
10	<code>PrintText(" Trong mục menu su dung các phím 'A', 'S', 'Enter' de di chuyển và chọn. ", x, ++y);</code>
11	<code>PrintText(" Chọn 'NEWGAME' de bắt đầu chơi ngay. ", x, ++y);</code>
12	<code>PrintText(" Người chơi su dung các phím 'A' de di chuyển sang trái, 'D' de di chuyển ", x, ++y);</code>
13	<code>PrintText(" sang phải, 'S' de di chuyển xuống dưới và 'W' de di chuyển lên trên một ", x, ++y);</code>
14	<code>PrintText(" Khi đang chơi người chơi có thể lưu game bằng cách nhấn phím 'L' và nhập ", x, ++y);</code>
15	<code>PrintText(" tên file cần lưu. Người chơi mô lại game đã lưu bằng 2 cách: 'LOAD GAME' ", x, ++y);</code>
16	<code>PrintText(" hoặc ấn phím 'T' và sử dụng nút 'A', 'S' de chọn file. ", x, ++y);</code>
17	<code>PrintText(" LUẬT CHƠI: ", x, ++y);</code>
18	<code>PrintText(" Người chơi được phép chọn luật cho trong mục 'OPTION' gồm luan chán hai ", x, ++y);</code>
19	<code>PrintText(" đấu và không chơi luật chán hai đấu. ", x, ++y);</code>
20	<code>PrintText(" Hai người chơi lần lượt di chuyển và chọn vào các ô chưa được đánh. Trò ", x, ++y);</code>
21	<code>PrintText(" chơi được xem là kết thúc khi một trong hai người chơi có 5 quân cờ liên ", x, ++y);</code>

22	<code>PrintText(" tiếp nhau doi voi luat khong chan hai dau hay co 5 quan co lien tiep nhau ", x, ++y);</code>
23	<code>PrintText(" va hai dau khong bi doi thu chan neu ap dung luat chan hai dau. Khi tat ", x, ++y);</code>
24	<code>PrintText(" ca cac o tren ban co deu kin ma chua ai chien thang thi game dau hoa nhau. ", x, ++y);</code>
25	<code>PrintText(" Chuc cac ban choi game vui ve. An bat cu phim nao de tro ve menu chinh... ", x, ++y);</code>
26	<code>return 0;</code>
27	<code>}</code>

Hàm Law() có chức năng in ra hướng dẫn và luật của trò chơi khi người chơi chọn mục Help trong Menu chính, sử dụng hàm PrintText để hướng dẫn nằm trong một khung trông đẹp mắt hơn.

HUONG DAN:

Game su dung hai ki hieu 'X','O' de dai dien cho nuoc co cua nguoi choi.
 Su dung ban co kich thuoc 12x12.
 Tren mot ban co chi cho phep hai nguoi choi tham gia.
 Trong muc menu su dung cac phim 'A', 'S', 'Enter' de di chuyen va chon.
 Chon 'NEWGAME' de bat dau choi ngay.

Nguoi choi su dung cac phim 'A' de di chuyen sang trai, 'D' de di chuyen sang phai, 'S' de di chuyen xuong duoi va 'W' de di chuyen len tren mot
 Khi dang choi nguoi choi co the luu game bang cach nhan phim 'L' va nhap ten file can luu. Nguoi choi mo lai game da luu bang 2 cach: 'LOAD GAME' hoac an phim 'T' va su dung nut 'A', 'S' de chon file.

LUAT CHOI:

Nguoi choi duoc phep chon luat cho trong muc 'OPTION' gom luan chan hai dau va khong choi luat chan hai dau.

Hai nguoi choi lan luot di chuyen va chon vao cac o chua duoc danh. Tro choi duoc xem la ket thuc khi mot trong hai nguoi choi co 5 quan co lien tiep nhau doi voi luat khong chan hai dau hay co 5 quan co lien tiep nhau va hai dau khong bi doi thu chan neu ap dung luat chan hai dau. Khi tat ca cac o tren ban co deu kin ma chua ai chien thang thi game dau hoa nhau. Chuc cac ban choi game vui ve. An bat cu phim nao de tro ve menu chinh... ■

2.7.2 Hàm di chuyển và lựa chọn ngoài game.

Hàm này chúng ta sử dụng để di chuyển lên xuống và chọn đối tượng trong menu chính, option, loadgame, ...

1	<code>void MoveUpMenu()</code>
2	<code>{</code>
3	<code>Y = Y - 1;</code>
4	<code>GotoXY(X, Y);</code>
5	<code>}</code>
6	<code>void MoveDownMenu()</code>
7	<code>{</code>
	<code>Y = Y + 1;</code>
8	<code>j--;</code>

9	}
10	int MoveAndChoose(int a, string A[], int _X, int _Y)
11	{
12	_COMMAND = toupper(_getch());
13	if (_COMMAND == 27)
14	while (1)
15	{
16	ExitGame();
17	return -1;
18	}
19	else
20	{
21	if (i >= 0 && i < a && Y <= 23 + a)
22	{
23	if (_COMMAND == 'W' && Y > _Y)
24	{
26	textcolor(496);
27	GotoXY(X, Y);
28	cout << A[i];
29	MoveUpMenu();
30	i--;
31	textcolor(15);
32	cout << A[i];
33	textcolor(496);
34	}
35	else if (_COMMAND == 'S' && Y < _Y + a - 1)
36	{
37	textcolor(496);
38	GotoXY(X, Y);
39	cout << A[i];
40	MoveDownMenu();
41	i--;
42	textcolor(15);

43	<code>cout << A[i];</code>
44	<code>textcolor(496);</code>
45	<code>}</code>
46	<code>else if (_COMMAND == 13)</code>
47	<code>return i;</code>
48	<code>}</code>
49	<code>}</code>
50	<code>}</code>
51	<code>}</code>

Các hàm MoveUpMenu() và MoveDownMenu() là các hàm thay đổi tọa độ con trỏ trong màn hình Menu chính. Hàm MoveAndChoose() thực hiện gọi hai hàm ở trên khi người chơi nhập “W” hay “S” và chọn khi người chơi nhấn Enter và xét hàm ExitGame() để xác nhận thoát nếu người chơi nhập Esc. Để có hiệu ứng tốt hơn ta sử dụng hàm textcolor. Ta truyền vào mảng kí tự mô tả menu, trước khi di chuyển xuống (hoặc lên) ta sẽ đổi màu về màu ban đầu, in ra phần tử mảng ở vị trí đó, và thay đổi tọa độ, in ra phần tử mảng tiếp theo, như vậy sẽ tạo được hiệu ứng như có 1 con trỏ màu đi lên xuống.

2.7.3 Hàm lựa chọn luật chơi:

1	<code>int ChooseOption(int a, string A[])</code>
2	<code>{</code>
3	<code>return MoveAndChoose(a, A, X, Y);</code>
4	<code>}</code>

Hàm ChooseOption() trả về giá trị biến LuatCHD ở trên. Ta gọi hàm MoveAndChoose().

2.7.4 Hàm Menu chính

Tạm bỏ qua biến isLoad(sẽ được giải thích ở hàm newgame), biến b là biến lựa chọn trong menu cho phép vào các lệnh tùy chọn để thực hiện các chức năng tương ứng. giá trị của biến b được xác định bởi hàm MoveAndChose. nếu như b bằng 0 chương trình sẽ vào hàm newgame và cho phép chơi game mới. bằng 1 sẽ vào chế độ load game, bằng 2 sẽ chọn được luật chơi,....

1	<code>int MainMenu(int isLoad)</code>
2	<code>{</code>
3	<code>int load = 0;</code>
4	<code>system("cls");</code>
5	<code>DrawBigText();</code>
6	<code>X = 55; Y = 23;</code>
7	<code>int b = isLoad ? 1 : MoveAndChoose(DrawMenu(), menu, 55, 23);</code>
8	<code>if (b == 0)</code>

9	{
10	int load = NewGame(Luat);
11	MainMenu(load);
12	{
13	else if (b == 1)
14	{
15	system("cls");
16	PrintText("CHON FILE GAME (ENTER DE CHON, ESC DE QUAY LAI): ", 40, 4);
17	X = 40; Y = 5;
18	findex = ChooseOption(DrawLoadFile(), files);
19	if (findex != -1)
20	load = NewGame(Luat, LoadGame(files[findex]));
21	}
22	else if (b == 2)
23	{
24	system("cls");
25	DrawBigText();
26	X = 40; Y = 23;
27	Luat = ChooseOption(DrawOption(), option);
28	
29	}
30	else if (b == 3)
31	{
32	Ccredit();
33	while (1)
34	{
35	if (toupper(_getch())) {break;}
36	}
37	}
38	else if (b == 4)
39	{
40	Lawgame();
41	while (1) {

42	<code>if (toupper(_getch())) {break;}</code>
43	<code>}</code>
44	<code>}</code>
45	<code>else if (b == 5) {</code>
46	<code>ExitGame();</code>
47	<code>exit(0);</code>
48	<code>return 0;</code>
49	<code>}</code>
50	<code>}</code>

2.8 Hàm bắt đầu game mới:

Giống như trong file hướng dẫn, và hàm NewGame này được bổ sung thêm chức năng lưu khi nhấn “L” và tải khi nhấn “T”. trở về biến isLoad ở hàm MainMenu, ta chú ý dòng 65, 67 của hàm NewGame khi nhấn T hàm trả về giá trị 1, dòng 10 ở hàm MainMenu nhận giá trị trả về đó vào biến load và đưa vào tham trị isload trong hàm đệ qui MainMenu ở dòng 11, MainMenu được thực hiện và tại dòng 7 không cần chọn biến chọn b đã được gán thẳng bằng 1 và vào ngay loadgame. Tóm lại biến isload để load game trong quá trình chơi.

1	<code>int NewGame(int LuatCHD, int a)</code>
2	<code>{</code>
3	<code>StartGame(a);</code>
4	<code>bool validEnter = true;</code>
5	<code>while (1</code>
6	<code>{</code>
7	<code>_COMMAND = toupper(_getch());</code>
8	<code>if (_COMMAND == 27)</code>
9	<code>{</code>
10	<code>ExitGame();</code>
11	<code>return 0;</code>
12	<code>}</code>
13	<code>else</code>
14	<code>{</code>
15	<code>else if (_COMMAND == 13)</code>
16	<code>else if (_COMMAND == 'W') MoveUp();</code>
17	<code>else if (_COMMAND == 'S') MoveDown();</code>
18	<code>else if (_COMMAND == 'D') MoveRight();</code>
19	<code>{</code>
20	<code>switch (CheckBoard(_X, _Y))</code>

21	{
22	case -1:
23	{
24	printf("X");
25	Xmoves++;
26	UpDate(Xmoves, Omoves, Xwins, Owins);
27	break;
28	}
29	case 1:
30	{
31	printf("O");
32	Omoves++;
33	UpDate(Xmoves, Omoves, Xwins, Owins);
34	break;
35	}
36	case 0: // Khi đánh vào ô đã đánh rồi
37	validEnter = false;
38	}
39	if (validEnter == true)
40	{
41	switch (ProcessFinish(TestBoard(_X,_Y,LuatCHD)))
42	{
43	case -1: case 1: case 0:
44	if (AskContinue() != 'Y')
45	{
46	system("cls");
47	MainMenu();
48	}
49	else
50	{
51	StartGame();
52	Xmoves = 0;
52	Omoves = 0;
53	GotoXY(_A[0][0].x, _A[0][0].y);

54	}
55	break;
56	}
57	}
58	validEnter = true;
59	}
60	else if (_COMMAND == 'L')
61	{
62	SaveGame();
63	return 0;
64	}
65	else if (_COMMAND == 'T')
66	{
67	Return 1;
68	}
69	}
70	}

2.9 Hàm Main:

1	int main() {
2	FixConsoleWindow();
3	int screen = MainMenu();
4	return 0;
5	}

3 Lời kết:

- Các thành viên trong nhóm suốt thời gian qua đã cùng nhau lập kế hoạch cho đồ án, cải tiến cũng như thêm bớt các chức năng.

- Đã có rất nhiều ý kiến cũng như các chức năng hay được đưa ra như nhập tên nhân vật hay là tạo chế độ đánh với máy, hay là chèn âm thanh,...). Nhưng vì để tránh gây rối cũng như phức tạp và đi xa khỏi mục tiêu Nhập môn lập trình, nhóm chúng em đã quyết định làm một game caro trên Console đơn giản nhưng cũng không kém phần sinh động, qua đó cũng giúp cho các thành viên không quá áp lực thời gian trong cái môn học khác cũng như giúp các thành viên tập trung thật vững cơ bản của Nhập môn lập trình.

- Tuy code của nhóm vẫn còn nhiều chỗ chưa được tối ưu, liên mạch cũng như dễ hiểu ở mức tối đa; nhưng nhóm đã làm rất tốt trong việc giao tiếp cũng như đạt được gần như mọi mục tiêu đã đề ra và hoàn thiện tốt sản phẩm cuối cùng của mình.