

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



**NHÓM 9 – LỚP 20CTT3**

**BÁO CÁO**  
**NHẬP MÔN LẬP TRÌNH**  
**| Đề tài |**  
**ĐỒ ÁN HỌC KÌ 1**  
**2020-2021**  
**GAME CARO**

**| Giáo viên hướng dẫn |**

**Trương Toàn Thịnh**

**Môn: 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 9 – LỚP 20CTT3**

# **BÁO CÁO**

## **NHẬP MÔN LẬP TRÌNH**

**| Đề tài |**

**ĐỒ ÁN HỌC KÌ 1**

**2020-2021**

**GAME CARO**

**| Thành viên nhóm |**

**Nguyễn Cảnh Huy – 20120496**

**Thái Nguyễn Việt Hùng – 20120488**

**Vũ Hoàng – 20120484**

**Nguyễn Phước Hải – 20120467**

**Ngô Trọng Tín – 20120423**

**| Giáo viên hướng dẫn |**

**Trương Toàn Thịnh**

**Môn: Nhập môn lập trình**

**Thành phố Hồ Chí Minh – 2020**

# MỤC LỤC

<b>MỞ ĐẦU:</b>	<b>5</b>
<b>Nội dung cần báo cáo :</b>	<b>5</b>
<b>Giao diện Game :</b>	<b>5</b>
<b>GIỚI THIỆU CÁC CHỨC NĂNG GAME :</b>	<b>6</b>
<b>CÁC KỸ THUẬT XỬ LÝ:</b>	<b>7</b>
<b>GIẢI THÍCH CODE :</b>	<b>8</b>
<b>1/Khai báo thư viện,hàm và biến cần sử dụng:</b>	<b>8</b>
<b>2/Các hàm trong bài:</b>	<b>9</b>
<b>TÀI LIỆU THAM KHẢO :</b>	<b>28</b>

# DANH MỤC HÌNH

Hình 1: giao diện ngoài màn hình trò chơi .....	5
Hình 2: giao diện bàn cờ.....	5
Hình 3: các chức năng của game.....	6
Hình 4 các hàm ẩn/hiện trở chuột, cố định màn hình, di chuyển vị trí .....	9
Hình 5 vẽ caro .....	10
Hình 6 các hàm di chuyển .....	11
Hình 7 bảng điểm 1 .....	11
Hình 8 chia hàng và bảng điểm 2 .....	12
Hình 9 bảng hướng dẫn .....	12
Hình 10 resetdata.....	13
Hình 11 start/exit game .....	13
Hình 12 nước đi phòng thủ của máy .....	14
Hình 13 hàm haswiner .....	14
Hình 14 hàm hiệu ứng.....	15
Hình 15 hàm effectwin.....	15
Hình 16 hàm processfinish và askcontinue .....	16
Hình 17 hàm hasdraw/testboard/checkboard .....	16
Hình 18 save game .....	17
Hình 19 load game.....	18
Hình 20 các hàm máy tấn công.....	19
Hình 21 hàm thiết lập máy.....	20
Hình 22 hàm vẽ bàn cờ .....	21
Hình 23 hàm vẽ XO.....	22
Hình 24 hàm newgame .....	23
Hình 25 hàm about.....	24
Hình 26 hàm in và help .....	25
Hình 27 hàm menu.....	27
Hình 28 hàm main .....	27

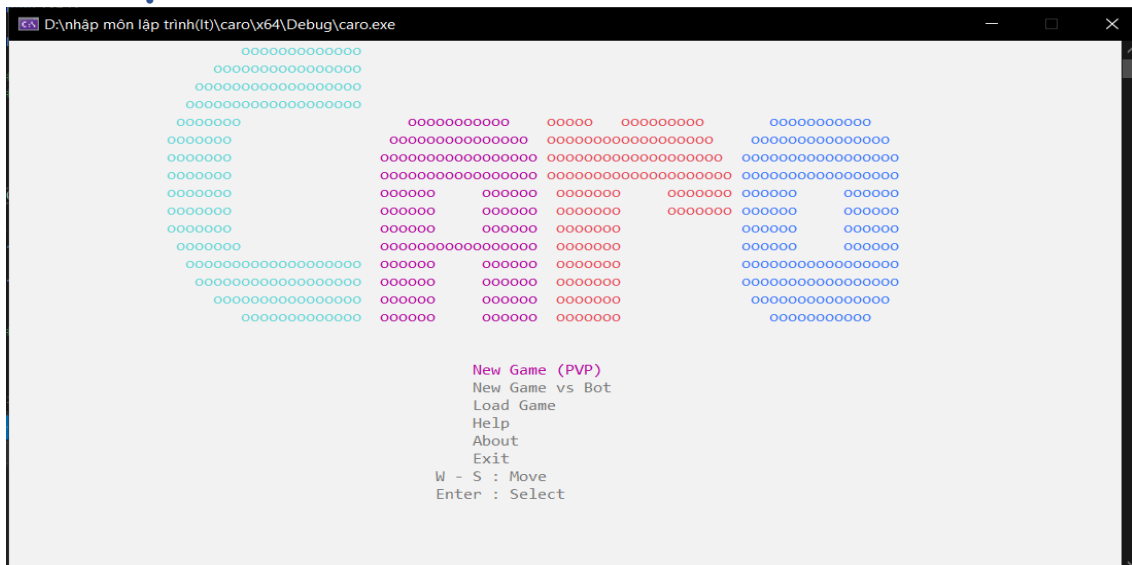
# MỞ ĐẦU

## MỞ ĐẦU:

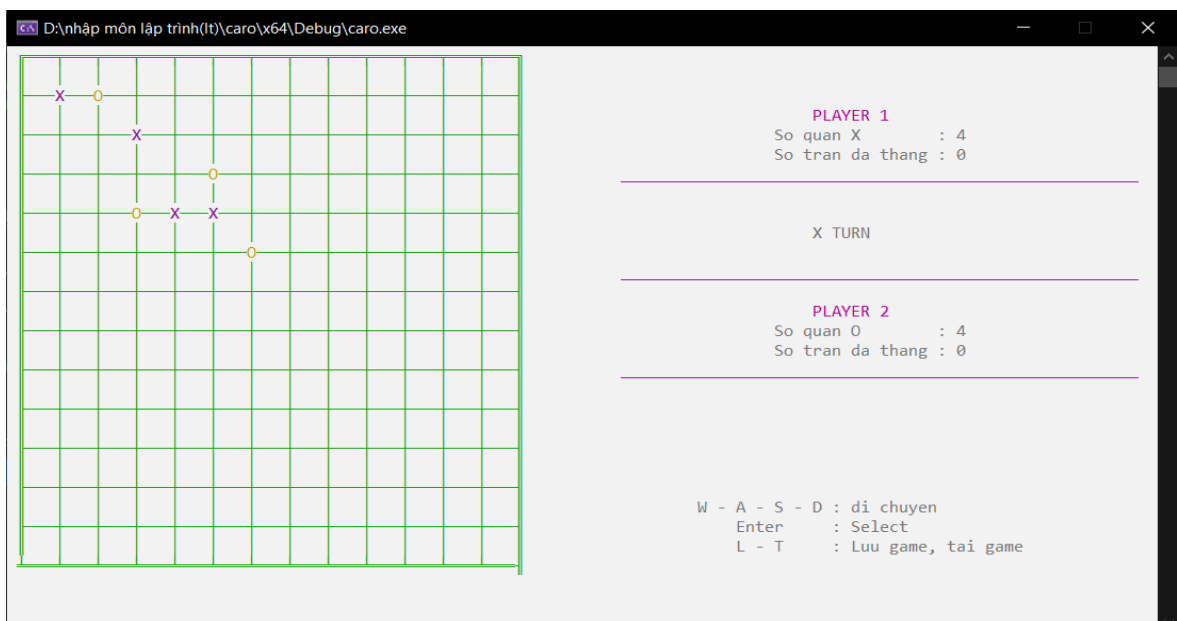
### Nội dung cần báo cáo :

- 1) Tất cả các kỹ thuật, thuật toán sinh viên đã tìm hiểu để thực hiện đồ án.
- 2) Mô tả các đoạn mã và các ý của từng chức năng trong đồ án.
- 3) Nêu rõ nguồn tham khảo .

### Giao diện Game :



Hình 1: giao diện ngoài màn hình trò chơi



Hình 2: giao diện bàn cờ

## GIỚI THIỆU CÁC CHỨC NĂNG GAME

### GIỚI THIỆU CÁC CHỨC NĂNG GAME :

- 1/ New Game ( PVP): Người đánh với người
- 2/ New Game vs Bot : Người đánh với máy
- 3/Load Game: Load lại trận đấu tạm dừng đã được lưu
- 4/Help: Hướng dẫn người chơi
- 5/About: Thông tin về game
- 6/Exit: Thoát game



Hình 3: các chức năng của game

## CÁC KỸ THUẬT XỬ LÝ

### CÁC KỸ THUẬT XỬ LÝ:

- 1/Xử lý thắng thua
- 2/Tìm nước đi cho máy
- 3/Xử lý âm thanh
- 4/Đổi màu kí tự
- 5/Đọc và xử lí file
- 6/Khung hình console
- 7/Ẩn/hiện con trỏ chuột

# GIẢI THÍCH CODE

## GIẢI THÍCH CODE :

### 1/Khai báo thư viện,hàm và biến cần sử dụng:

//Khai báo các thư viện cần dùng

```
#include "Windows.h"
#include "iostream"
#include "stdio.h"
#include "conio.h"
#include "ctype.h"
#include "mmsystem.h"
#include "stdlib.h"
#include <string>
#include <fstream>
#include <vector>
#include <deque>
#include <cstdlib>
```

```
using namespace std;
#pragma comment (lib, "winmm.lib")
```

//Khai báo các biến hằng

```
#define BOARD_SIZE 13
#define LEFT 3
#define TOP 1
#define Black 0
#define DarkBlue 1
#define DarkGreen 2
#define DarkCyan 3
#define DarkRed 4
#define DarkPink 5
#define DarkYellow 6
#define DarkWhite 7
#define Grey 8
#define Blue 9
#define Green 249
#define Cyan 11
#define Red 251
#define Pink 252
#define Yellow 253
#define White 15
#define defaultColor 7
```

//Khai báo mảng

```
struct _POINT { int x, y, c; }; // x: tọa độ dòng, y: tọa độ cột, c: đánh dấu
struct both { int val; int ok; }; //Ma trận bàn cờ
_POINT a[BOARD_SIZE][BOARD_SIZE];
bool turn; //true là người thứ nhất false là người thứ hai
int _COMMAND; // Biến nhận giá trị phím người dùng nhập
```



```

int nowX, nowY; //Tọa độ hiện hành trên màn hình bàn cờ
int point1 = 0, point2 = 0; //Điểm 2 người chơi
int dem1 = 0, dem2 = 0; //Số quân cờ 2 người chơi
bool isLoad = false; //true = loadGame, false = newGame
bool playWithBot = false; //true = newgame, false = playvsbot

```

```

deque<both> roadOfBot1, roadOfBot2; //đường đi của bot

```

## 2/Các hàm trong bài:

```

void AnTroChuot() //Ẩn trỏ chuột

```

```

void HienTroChuot() //Hiện trỏ chuột

```

```

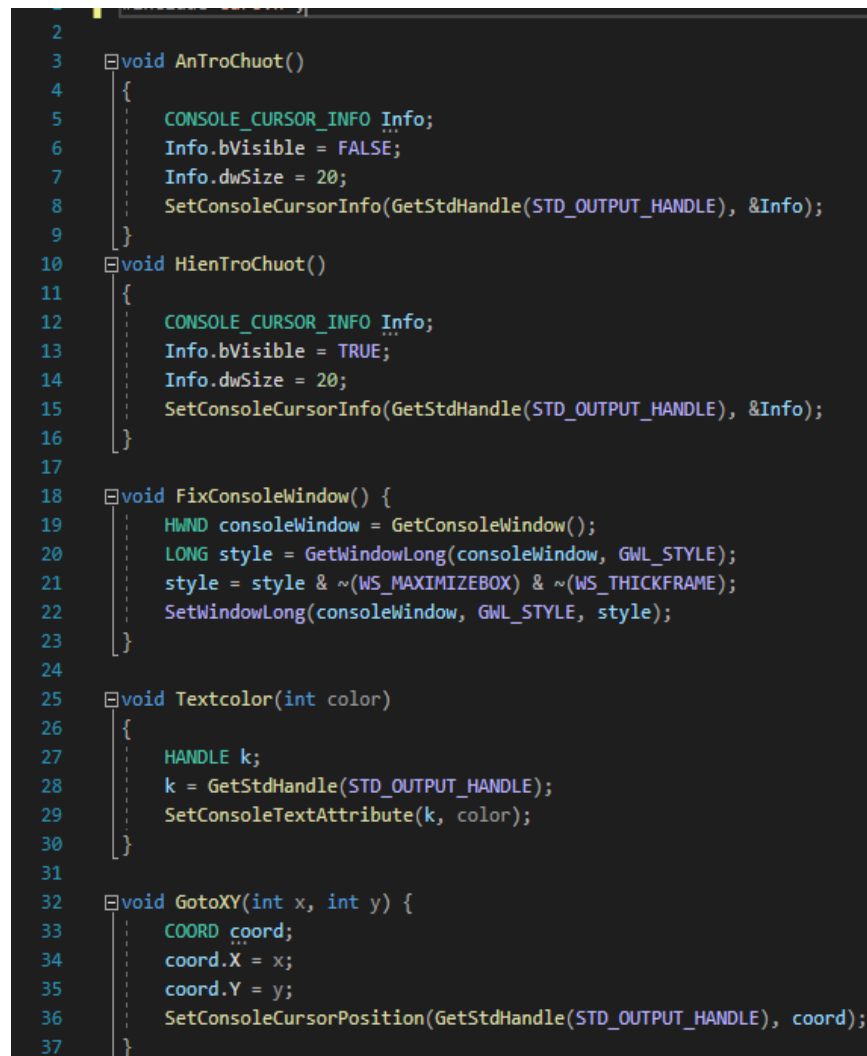
void FixConsoleWindow() //Cố định màn hình

```

```

void GotoXY(int x, int y) //Di chuyển

```



```

2
3 void AnTroChuot()
4 {
5     CONSOLE_CURSOR_INFO Info;
6     Info.bVisible = FALSE;
7     Info.dwSize = 20;
8     SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &Info);
9 }
10 void HienTroChuot()
11 {
12     CONSOLE_CURSOR_INFO Info;
13     Info.bVisible = TRUE;
14     Info.dwSize = 20;
15     SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &Info);
16 }
17
18 void FixConsoleWindow() {
19     HWND consoleWindow = GetConsoleWindow();
20     LONG style = GetWindowLong(consoleWindow, GWL_STYLE);
21     style = style & ~(WS_MAXIMIZEBOX) & ~(WS_THICKFRAME);
22     SetWindowLong(consoleWindow, GWL_STYLE, style);
23 }
24
25 void Textcolor(int color)
26 {
27     HANDLE k;
28     k = GetStdHandle(STD_OUTPUT_HANDLE);
29     SetConsoleTextAttribute(k, color);
30 }
31
32 void GotoXY(int x, int y) {
33     COORD coord;
34     coord.X = x;
35     coord.Y = y;
36     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
37 }

```

Hình 4 các hàm ẩn/hiện trỏ chuột, cố định màn hình, di chuyển vị trí

```
void PrintCaro2() //Vẽ chữ caro
```

```
//Di chuyển đến từng vị trí thích hợp để vẽ các chữ
```

```

38
39 void PrintCaro2()
40 {
41     Sleep(200);
42     PlaySound(TEXT("nhac123.wav"), NULL, SND_FILENAME | SND_ASYNC); //hàm phát nhạc
43
44     AnTroChuat();
45     int x = 25, y = 0;
46     Sleep(200);
47     Textcolor(Red); //đổi màu kí tự
48     GotoXY(x, y); cout << "oooooooooooooooo";
49     GotoXY(x - 3, y + 1); cout << "oooooooooooooooo";
50     GotoXY(x - 5, y + 2); cout << "oooooooooooooooo";
51     GotoXY(x - 6, y + 3); cout << "oooooooooooooooo";
52     GotoXY(x - 7, y + 4); cout << "ooooooo";
53     GotoXY(x - 8, y + 5); cout << "ooooooo";
54     GotoXY(x - 8, y + 6); cout << "ooooooo";
55     GotoXY(x - 8, y + 7); cout << "ooooooo";
56     GotoXY(x - 8, y + 8); cout << "ooooooo";
57     GotoXY(x - 8, y + 9); cout << "ooooooo";
58     GotoXY(x - 8, y + 10); cout << "ooooooo";
59     GotoXY(x - 7, y + 11); cout << "ooooooo";
60     GotoXY(x - 6, y + 12); cout << "oooooooooooooooo";
61     GotoXY(x - 5, y + 13); cout << "oooooooooooooooo";
62     GotoXY(x - 3, y + 14); cout << "oooooooooooooooo";
63     GotoXY(x, y + 15); cout << "oooooooooooooooo";
64     Sleep(200);
65     Textcolor(Yellow);
66     GotoXY(x - 4 + 22, y + 4); cout << "oooooooooooo";
67     GotoXY(x - 4 + 22 - 2, y + 5); cout << "oooooooooooooooo";
68     GotoXY(x - 4 + 22 - 3, y + 6); cout << "oooooooooooooooo";
69     GotoXY(x - 4 + 22 - 3, y + 7); cout << "oooooooooooooooo";
70     GotoXY(x - 4 + 22 - 3, y + 8); cout << "ooooooo oooooo";
71     GotoXY(x - 4 + 22 - 3, y + 9); cout << "ooooooo oooooo";
72     GotoXY(x - 4 + 22 - 3, y + 10); cout << "ooooooo oooooo";
73     GotoXY(x - 4 + 22 - 3, y + 11); cout << "oooooooooooooooo";
74     GotoXY(x - 4 + 22 - 3, y + 12); cout << "ooooooo oooooo";
75     GotoXY(x - 4 + 22 - 3, y + 13); cout << "ooooooo oooooo";
76     GotoXY(x - 4 + 22 - 3, y + 14); cout << "ooooooo oooooo";
77     GotoXY(x - 4 + 22 - 3, y + 15); cout << "ooooooo oooooo";
78     Sleep(200);
79     Textcolor(Pink);
80
81     GotoXY(x + 33, y + 4); cout << "ooooo oooooooooo";
82     GotoXY(x + 33, y + 5); cout << "oooooooooooooooo";
83     GotoXY(x + 33, y + 6); cout << "oooooooooooooooo";
84     GotoXY(x + 33, y + 7); cout << "oooooooooooooooo";
85     GotoXY(x + 1 + 33, y + 8); cout << "ooooooo oooooo";
86     GotoXY(x + 1 + 33, y + 9); cout << "ooooooo oooooo";
87     GotoXY(x + 1 + 33, y + 10); cout << "ooooooo";
88     GotoXY(x + 1 + 33, y + 11); cout << "ooooooo";
89     GotoXY(x + 1 + 33, y + 12); cout << "ooooooo";
90     GotoXY(x + 1 + 33, y + 13); cout << "ooooooo";
91     GotoXY(x + 1 + 33, y + 14); cout << "ooooooo";
92     GotoXY(x + 1 + 33, y + 15); cout << "ooooooo";
93     Sleep(200);
94     Textcolor(Green);
95     GotoXY(x + 57, y + 4); cout << "oooooooooooo";
96     GotoXY(x + 57 - 2, y + 5); cout << "oooooooooooooooo";
97     GotoXY(x + 57 - 3, y + 6); cout << "oooooooooooooooo";
98     GotoXY(x + 57 - 3, y + 7); cout << "oooooooooooooooo";
99     GotoXY(x + 57 - 3, y + 8); cout << "ooooooo oooooo";
100    GotoXY(x + 57 - 3, y + 9); cout << "ooooooo oooooo";
101    GotoXY(x + 57 - 3, y + 10); cout << "ooooooo oooooo";
102    GotoXY(x + 57 - 3, y + 11); cout << "ooooooo oooooo";
103    GotoXY(x + 57 - 3, y + 12); cout << "oooooooooooooooo";
104    GotoXY(x + 57 - 3, y + 13); cout << "oooooooooooooooo";
105    GotoXY(x + 57 - 2, y + 14); cout << "oooooooooooooooo";
106    GotoXY(x + 57, y + 15); cout << "oooooooooooo";
107    Textcolor(defaultColor);
108    Sleep(5);
109

```

Hình 5 vẽ caro

//Các hàm di chuyển con trỏ

//Thay đổi tọa độ x,y để di chuyển con trỏ

void MoveRight()

void MoveLeft()

void MoveDown()

void MoveUp()

```

258
259 void MoveRight() {
260     if (nowX < a[BOARD_SIZE - 1][BOARD_SIZE - 1].x) { //con trỏ không vượt qua phạm vi bàn cờ
261         nowX += 4;
262         GotoXY(nowX, nowY);
263     }
264 }
265
266 void MoveLeft() {
267     if (nowX > a[0][0].x) {
268         nowX -= 4;
269         GotoXY(nowX, nowY);
270     }
271 }
272
273 void MoveDown() {
274     if (nowY < a[BOARD_SIZE - 1][BOARD_SIZE - 1].y) {
275         nowY += 2;
276         GotoXY(nowX, nowY);
277     }
278 }
279
280 void MoveUp() {
281     if (nowY > a[0][0].y) {
282         nowY -= 2;
283         GotoXY(nowX, nowY);
284     }
285 }

```

Hình 6 các hàm di chuyển

void bangdiem1(int colorWin = 7) //bảng điểm người chơi 1

void bangdiem1phu(int colorWin = 7) //bảng thông báo khi người chơi 1 thắng

```

286 void bangdiem1(int colorWin = 7) {
287     GotoXY((BOARD_SIZE + 8) * 4, 3);
288     Textcolor(5 + 241 + colorWin);
289     if (playWithBot == false) //nếu không phải là playvsbot sẽ hiển thị player1 else player
290         cout << "PLAYER 1";
291     else cout << "PLAYER";
292     GotoXY((BOARD_SIZE + 7) * 4, 4);
293     Textcolor(colorWin + 241);
294     cout << "Số quân X : "; cout << dem1; //dem1 sẽ đếm số quân cờ của x và hiển thị ra
295     GotoXY((BOARD_SIZE + 7) * 4, 5);
296     cout << "Số trận đã thắng : "; cout << point1; //point1 hiển thị số điểm người chơi
297     if (dem1 <= dem2)
298     {
299         GotoXY((BOARD_SIZE + 8) * 4, 9); //dùng đk số quân cờ mỗi bên để hiển thị lượt kế tiếp là của ai
300         cout << "X's TURN" << endl;
301     }
302     else
303     {
304         GotoXY((BOARD_SIZE + 8) * 4, 9);
305         cout << "O's TURN" << endl;
306     }
307 }
308
309 void bangdiem1phu(int colorWin = 7) {
310     GotoXY((BOARD_SIZE + 8) * 4, 3);
311     Textcolor(5 + 241 + colorWin);
312     if (playWithBot == false)
313         cout << "PLAYER 1";
314     else cout << "PLAYER";
315     GotoXY((BOARD_SIZE + 7) * 4, 4);
316     Textcolor(colorWin + 241);
317     cout << "Số quân X : "; cout << dem1;
318     GotoXY((BOARD_SIZE + 7) * 4, 5);
319     cout << "Số trận đã thắng : "; cout << point1;
320     GotoXY((BOARD_SIZE + 8) * 4, 9);
321     cout << " X WIN " << endl; //bảng phụ dùng để in ra các hiệu ứng khi thắng
322 }

```

Hình 7 bảng điểm 1

```
void chiahang(int colorWin = 7) //chia hàng phân cách các bảng điểm
```

```
void bangdiem2(int colorWin = 7) //bảng điểm người chơi 2
```

```
void bangdiem2phu(int colorWin = 7) //bảng thông báo khi người chơi 2 thắng
```

// bảng điểm 1 và 2 để hiển thị các quân cờ đánh, số trận thắng tên và lượt đánh. Ta dùng các biến để hiển thị số quân và trận thắng. Ta còn so sánh số quân đã đánh hay các biến dem1 dem2 để hiển thị lượt đánh của mỗi bên

```

323
324
325 void chiahang(int colorWin = 7) //chia rõ các hàng
326 {
327     GotoXY((BOARD_SIZE + 3) * 4, 6);
328     cout << "_____";
329     GotoXY((BOARD_SIZE + 3) * 4, 11);
330     cout << "_____";
331     GotoXY((BOARD_SIZE + 3) * 4, 16);
332     cout << "_____";
333 }
334
335 void bangdiem2(int colorWin = 7) {
336     GotoXY((BOARD_SIZE + 8) * 4, 13);
337     Textcolor(5 + 241 + colorWin);
338     if (playWithBot == false)
339         cout << "PLAYER 2";
340     else cout << "BOT ";
341     GotoXY((BOARD_SIZE + 7) * 4, 14);
342     Textcolor(colorWin + 241);
343     cout << "Số quân 0      : "; cout << dem2;
344     GotoXY((BOARD_SIZE + 7) * 4, 15);
345     cout << "Số trận đã thắng : "; cout << point2;
346     if (dem1 <= dem2)
347     {
348         GotoXY((BOARD_SIZE + 8) * 4, 9);
349         cout << "X's TURN" << endl;
350     }
351     else
352     {
353         GotoXY((BOARD_SIZE + 8) * 4, 9);
354         cout << "O's TURN" << endl;
355     }
356 }
357
358

```

Hình 8 chia hàng và bảng điểm 2

```
void banghuongdan() //bảng hướng dẫn sử dụng các phím
```

// di chuyển đến vị trí thích hợp và in các câu hướng dẫn cho người chơi

```

374
375 void banghuongdan() { //in ra các hướng dẫn chơi ngay trong màn hình bàn cờ
376     Textcolor(248);
377     GotoXY((BOARD_SIZE + 5) * 4, 23);
378     cout << "W - A - S - D : di chuyen";
379     GotoXY((BOARD_SIZE + 5) * 4, 24);
380     cout << "    Enter    : Select";
381     GotoXY((BOARD_SIZE + 5) * 4, 25);
382     cout << "    L - T    : Luu game, tai game";
383 }
384

```

Hình 9 bảng hướng dẫn

`void ResetData()` //tạo dữ liệu lại như ban đầu

// nếu load game ta giữ nguyên dữ liệu

// nếu game mới ta chạy vòng for để đưa các ô cờ đã đánh trở về ô trống (0). Gán các giá biến biểu thị lượt đánh, số quân về ban đầu

```

384
385 void ResetData() {
386     PlaySound(TEXT("nhac.wav"), NULL, SND_FILENAME | SND_ASYNC);
387     for (int i = 0; i < BOARD_SIZE; i++) {
388         for (int j = 0; j < BOARD_SIZE; j++) {
389             a[i][j].x = 4 * j + LEFT + 2; // Trùng với hoành độ màn hình bàn cờ
390             a[i][j].y = 2 * i + TOP + 1; // Trùng với tung độ màn hình bàn cờ
391
392             if (isLoad) continue; //nếu load game thì không reset
393             a[i][j].c = 0; // 0 nghĩa là chưa ai đánh dấu, nếu đánh dấu phải theo quy
394                 //định như sau: -1 là lượt true đánh, 1 là lượt false đánh
395         }
396     }
397     HienTroChuot();
398
399     if (!isLoad) { //nếu loadgame thì không gán lại biến đếm và turn
400         dem1 = dem2 = 0;
401         turn = true;
402     }
403
404     _COMMAND = -1; // Gán lượt và phím mặc định
405     nowX = a[0][0].x; nowY = a[0][0].y; // Thiết lập lại tọa độ hiện hành ban đầu
406 }
407

```

Hình 10 resetdata

`void StartGame()` //bắt đầu game. Nếu load game thì lấy lại giữ liệu đã lưu và vẽ bàn cờ, không thì reset lại và vẽ bàn cờ

`void ExitGame()` //thoát game

`bool inBoard(int x, int y)` //giới hạn phạm vi di chuyển và đánh quân cờ ở trong bảng. Hay nói cách khác kiểm tra vị trí x,y còn thuộc bàn cờ hay không

`both check(int x, int y, int n, int m, bool ok = false)` // đếm số quân trùng với ô x,y theo hướng m,n và kiểm tra cuối hàng có quân khác với ô hiện tại hay không. Kiểm tra xem có quân khác chặn 2 đầu hay không. Trả về struct both

```

518
519 void StartGame() {
520     Textcolor(241);
521     system("cls");
522     ResetData(); //lấy lại dữ liệu ban đầu
523     DrawBoard(BOARD_SIZE + 1); //vẽ bàn cờ
524     if (isLoad) DrawXO(); //nếu là loadgame thì tiếp tục chơi
525     isLoad = false;
526     GotoXY(a[0][0].x, a[0][0].y); //nếu newgame con trỏ sẽ ở vị trí a00
527 }
528
529 void ExitGame() {
530     point1 = 0, point2 = 0;
531     Textcolor(247);
532     playWithBot = false;
533     system("cls");
534 }
535
536 bool inBoard(int x, int y) {
537     if (0 <= x && x < BOARD_SIZE && 0 <= y && y < BOARD_SIZE) return true; //tọa độ con trỏ trong phạm vi bàn cờ
538     return false;
539 }
540
541 both check(int x, int y, int n, int m, bool ok = false) {
542     int dem = 0;
543     int newX = x, newY = y;
544     while (1) {
545         newX = newX + n; //dịch tọa độ theo hướng n
546         newY = newY + m; //dịch tọa độ theo hướng m
547         if (!inBoard(newX, newY)) break; //nếu tọa độ mới không trong bàn cờ thì break
548         if (a[newX][newY].c == a[x][y].c) dem++;
549         else break;
550     }
551     if (ok == true) return{ newX, newY };
552     return{ dem, bool(a[newX][newY].c != 0) };
553 }
554
555

```

Hình 11 start/exit game

```

void kiemtrahang(int a, int b, int c, int d) //kiểm tra số quân cờ trên 1 hàng
//máy phòng thủ
bool uutien() // ưu tiên nước đi cho máy
void themNuocDi(int a, int b, int c, int d) //xác định nước đi của máy

```

```

caro (Global Scope) about()
560 void kiemtrahang(int a, int b, int c, int d) {
561     ans1 = check(x, y, a, b), ans2 = check(x, y, c, d);
562     Count = ans2.val + ans1.val + 1; //số quân bên trái+số quân bên phải+1 quân ở giữa
563     rightRule = (ans1.ok && ans2.ok);
564 }
565
566 bool uutien() {
567     if (Count >= 3 && !ans1.ok && !ans2.ok) return true; //nếu đếm số quân người chơi từ 3 đến 4 quân
568     if (Count >= 4 && (!ans1.ok && ans2.ok) || (ans1.ok && !ans2.ok)) return true; //thì máy sẽ ưu tiên chặn lại
569     return false;
570 }
571
572 void themNuocDi(int a, int b, int c, int d) { //máy sẽ thêm nước đi cho nó
573     both res1 = check(x, y, a, b, true), res2 = check(x, y, c, d, true);
574     if (uutien()) {
575         if (!ans1.ok) {
576             if (inBoard(res1.val, res1.ok)) {
577                 roadOfBot1.push_back(res1);
578             }
579         }
580         if (!ans2.ok) {
581             if (inBoard(res2.val, res2.ok)) {
582                 roadOfBot1.push_back(res2);
583             }
584         }
585     }
586     else {
587         if (!ans1.ok) {
588             if (inBoard(res1.val, res1.ok)) {
589                 roadOfBot2.push_back(res1);
590             }
591         }
592         if (!ans2.ok) {
593             if (inBoard(res2.val, res2.ok)) {
594                 roadOfBot2.push_back(res2);
595             }
596         }
597     }
598 }
599

```

Hình 12 nước đi phòng thủ của máy

```

int hasWinner() { //kiểm tra người thắng.

```

```

601 int hasWinner() {
602     y = (nowX - LEFT - 2) / 4;
603     x = (nowY - TOP - 1) / 2;
604     // ngang
605     kiemtrahang(0, 1, 0, -1);
606
607     if (turn && playWithBot) { //nếu đến lượt của 0 và đang ở chế độ playvsbot
608         themNuocDi(0, 1, 0, -1); //thì máy sẽ thêm nước đi
609     }
610
611     if (Count == 5 && !rightRule) return 1; //nếu đếm số quân theo hàng đứng = 5 và k bị chặn sẽ return 1 báo chiến thắng
612     // dọc
613     kiemtrahang(1, 0, -1, 0);
614     if (turn && playWithBot) { //tương tự như kiểm tra hàng ngang
615         themNuocDi(1, 0, -1, 0);
616     }
617     if (Count == 5 && !rightRule) return 2;
618     // chéo
619     kiemtrahang(1, 1, -1, -1);
620     if (turn && playWithBot) { //tương tự như kiểm tra hàng ngang
621         themNuocDi(1, 1, -1, -1);
622     }
623     if (Count == 5 && !rightRule) return 3;
624     // chéo
625     kiemtrahang(1, -1, -1, 1);
626     if (turn && playWithBot) { //tương tự như kiểm tra hàng ngang
627         themNuocDi(1, -1, -1, 1);
628     }
629     if (Count == 5 && !rightRule) return 4;
630
631     return 0;
632 }
633

```

Hình 13 hàm haswiner

`void hieuUng(int x, int y, int n, int m, int color) //hiệu ứng khi chiến thắng`

```

633
634 void hieuUng(int x, int y, int n, int m, int color) {
635     int dem = 0;
636
637     int newX = x, newY = y;
638
639     color = color % 200; //hiệu ứng nhấp nháy đổi màu
640     GotoXY(a[newX][newY].x, a[newX][newY].y);
641     Textcolor(color++);
642     if (a[newX][newY].c == -1) cout << "X";
643     else cout << "O";
644     while (1) {
645         Sleep(25);
646
647         newX = newX + n;
648         newY = newY + m;
649         if (linBoard(newX, newY)) break;
650         if (a[newX][newY].c == a[x][y].c) dem++;
651         else break;
652
653         Textcolor(color = (color + 20) % 200);
654         if (a[newX][newY].c == -1) {
655             AnTroChuat();
656
657             bangdiem1phu(color);
658             GotoXY(a[newX][newY].x, a[newX][newY].y);
659             cout << "X";
660         }
661         else {
662             AnTroChuat();
663             bangdiem2phu(color);
664             GotoXY(a[newX][newY].x, a[newX][newY].y);
665             cout << "O";
666         }
667     }
668 }

```

Hình 14 hàm hiệu ứng

`void effectWin() //hiển thị hiệu ứng chiến thắng`

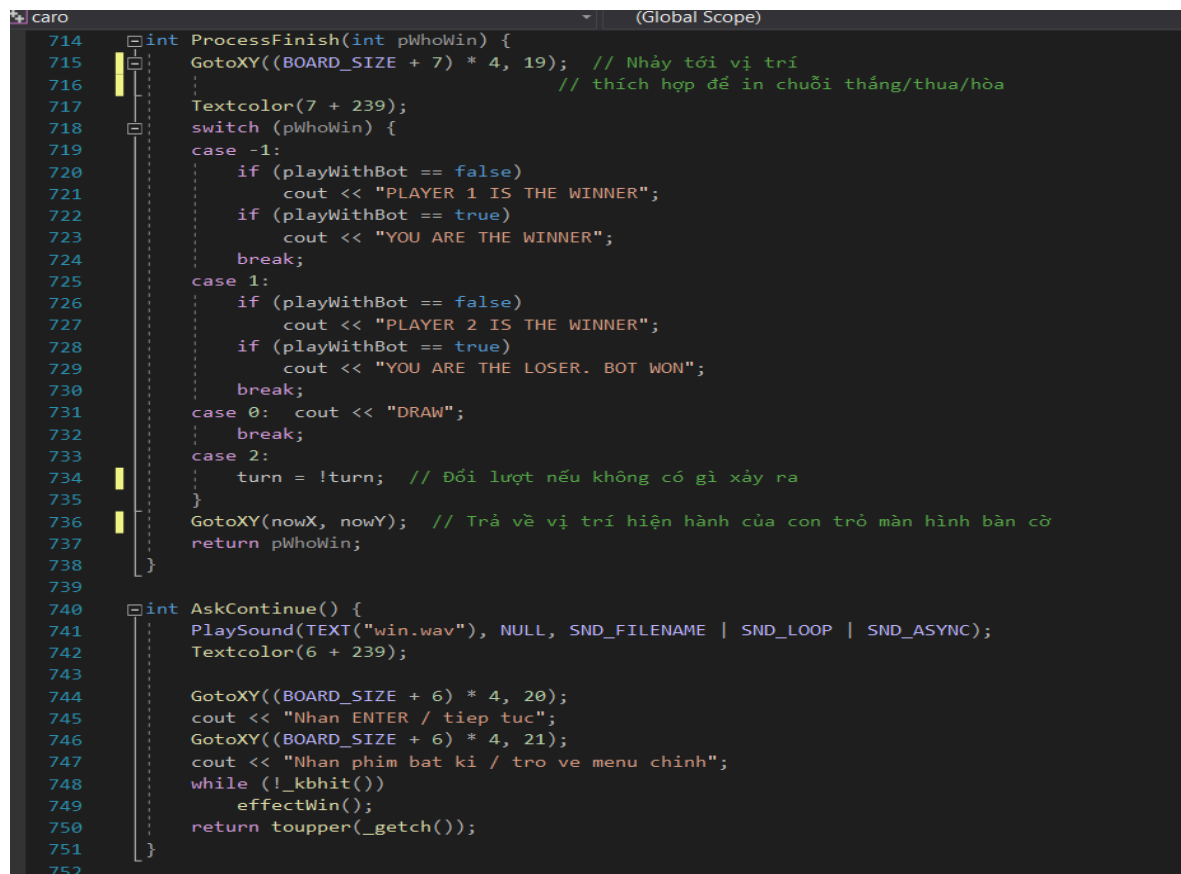
```

672 void effectWin() {
673     int dem = 0;
674     int y = (nowX - LEFT - 2) / 4;
675     int x = (nowY - TOP - 1) / 2;
676     if (hasWinner() == 1) { //nếu ==1 nghĩa là thắng theo hàng ngang
677
678         while (dem < 2)
679         {
680             hieuUng(x, y, 0, 1, color++);
681             hieuUng(x, y, 0, -1, color++);
682             dem++;
683         }
684     }
685     if (hasWinner() == 2) { //nếu ==2 nghĩa là thắng theo hàng dọc
686
687         while (dem < 2) // dùng đk biến đếm để chờ hiệu ứng
688         {
689             hieuUng(x, y, 1, 0, color++);
690             hieuUng(x, y, -1, 0, color++);
691             dem++;
692         }
693     }
694     if (hasWinner() == 3) { //nếu ==3 nghĩa là thắng theo chéo
695
696         while (dem < 2)
697         {
698             hieuUng(x, y, 1, 1, color++);
699             hieuUng(x, y, -1, -1, color++);
700             dem++;
701         }
702     }
703     if (hasWinner() == 4) //nếu ==4 nghĩa là thắng theo chéo
704     {
705         while (dem < 2)
706         {
707             hieuUng(x, y, -1, 1, color++);
708             hieuUng(x, y, 1, -1, color++);
709             dem++;
710         }
711     }

```

Hình 15 hàm effectwin

```
int ProcessFinish(int pWhoWin) //Hàm xử lý khi người chơi thua
int AskContinue() { //hỏi người chơi có muốn tiếp tục chơi không
```



```

714 int ProcessFinish(int pWhoWin) {
715     GotoXY((BOARD_SIZE + 7) * 4, 19); // Nhảy tới vị trí
716                                     // thích hợp để in chuỗi thắng/thua/hòa
717     Textcolor(7 + 239);
718     switch (pWhoWin) {
719     case -1:
720         if (playWithBot == false)
721             cout << "PLAYER 1 IS THE WINNER";
722         if (playWithBot == true)
723             cout << "YOU ARE THE WINNER";
724         break;
725     case 1:
726         if (playWithBot == false)
727             cout << "PLAYER 2 IS THE WINNER";
728         if (playWithBot == true)
729             cout << "YOU ARE THE LOSER. BOT WON";
730         break;
731     case 0: cout << "DRAW";
732             break;
733     case 2:
734         turn = !turn; // Đổi lượt nếu không có gì xảy ra
735     }
736     GotoXY(nowX, nowY); // Trả về vị trí hiện hành của con trỏ màn hình bàn cờ
737     return pWhoWin;
738 }
739
740 int AskContinue() {
741     PlaySound(TEXT("win.wav"), NULL, SND_FILENAME | SND_LOOP | SND_ASYNC);
742     Textcolor(6 + 239);
743
744     GotoXY((BOARD_SIZE + 6) * 4, 20);
745     cout << "Nhấn ENTER / tiếp tục";
746     GotoXY((BOARD_SIZE + 6) * 4, 21);
747     cout << "Nhấn phím bất kỳ / trở về menu chính";
748     while (!kbhit())
749         effectWin();
750     return toupper(_getch());
751 }
752

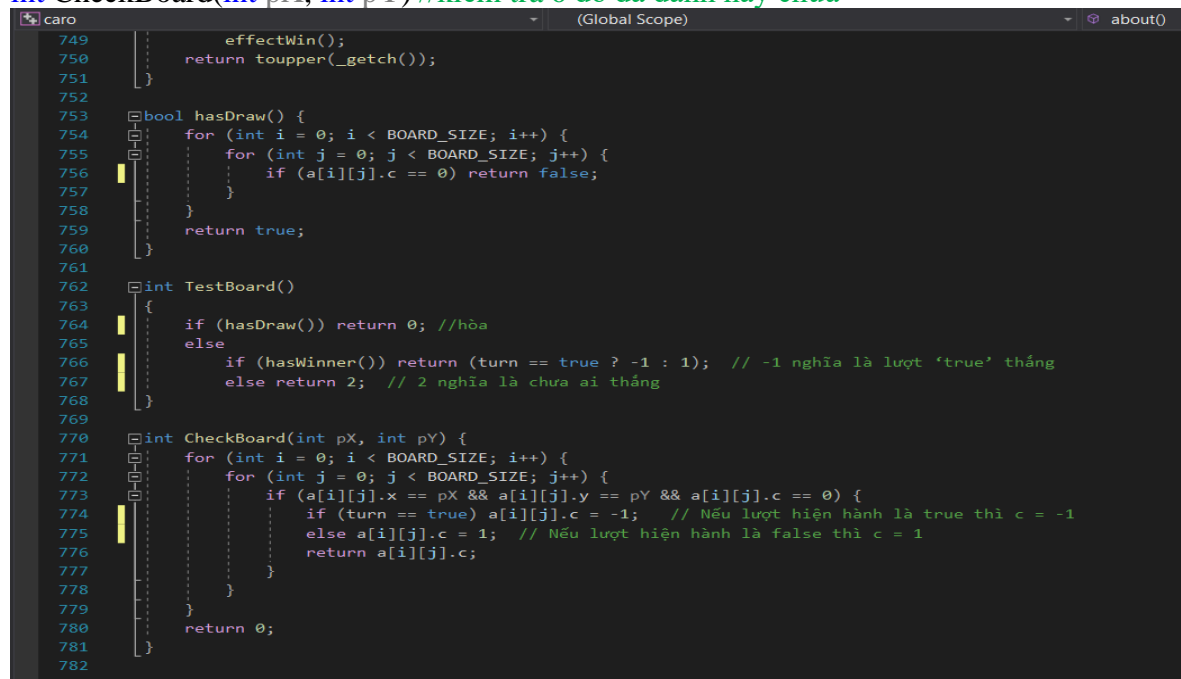
```

Hình 16 hàm processfinish và askcontinue

```
bool hasDraw() //xử lý hòa
```

```
int TestBoard() //kiểm tra đã thắng/thua/hòa chưa
```

```
int CheckBoard(int pX, int pY) //kiểm tra ô đó đã đánh hay chưa
```



```

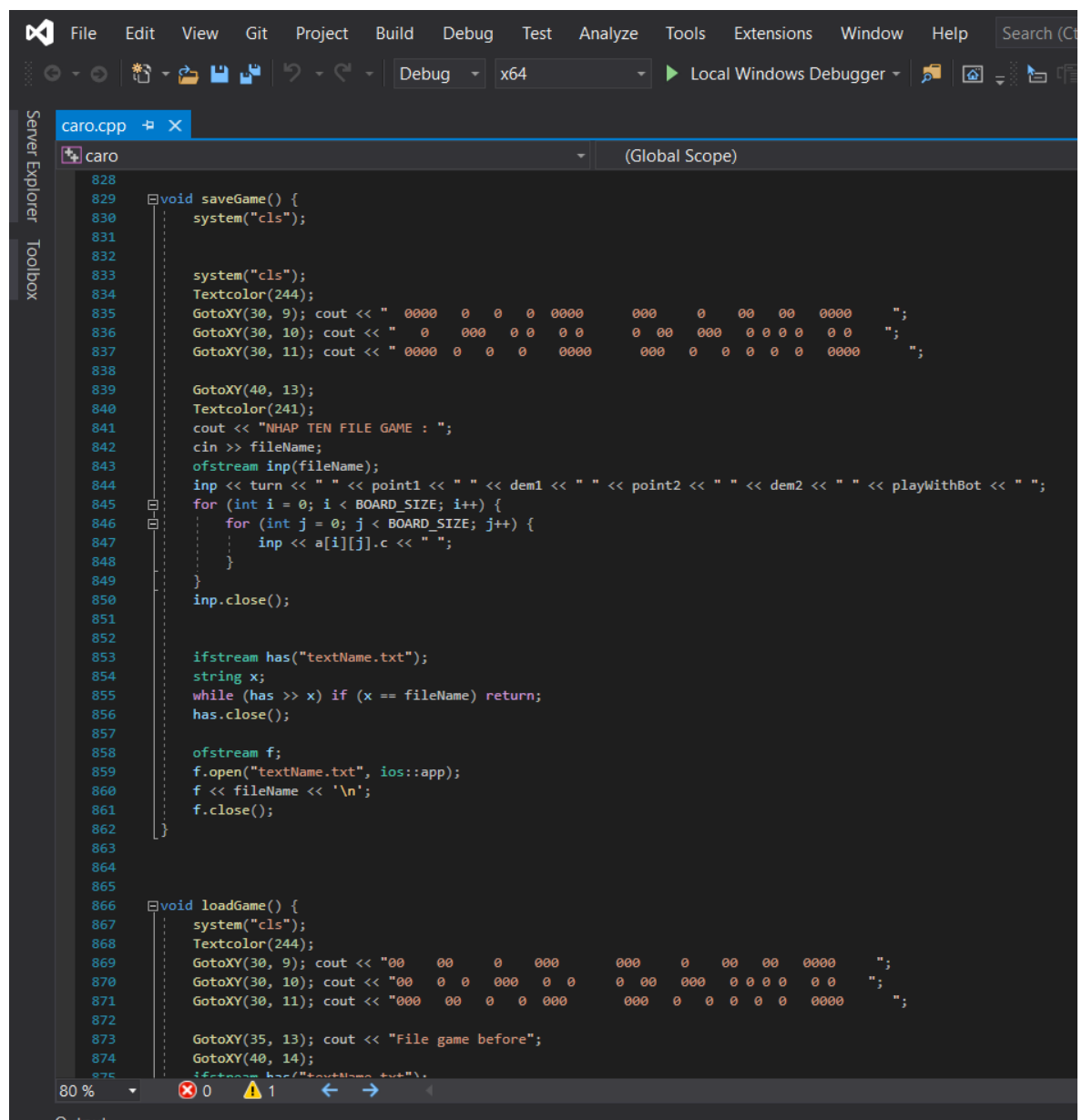
749     effectWin();
750     return toupper(_getch());
751 }
752
753 bool hasDraw() {
754     for (int i = 0; i < BOARD_SIZE; i++) {
755         for (int j = 0; j < BOARD_SIZE; j++) {
756             if (a[i][j].c == 0) return false;
757         }
758     }
759     return true;
760 }
761
762 int TestBoard()
763 {
764     if (hasDraw()) return 0; //hòa
765     else
766         if (hasWinner()) return (turn == true ? -1 : 1); // -1 nghĩa là lượt 'true' thắng
767         else return 2; // 2 nghĩa là chưa ai thắng
768 }
769
770 int CheckBoard(int pX, int pY) {
771     for (int i = 0; i < BOARD_SIZE; i++) {
772         for (int j = 0; j < BOARD_SIZE; j++) {
773             if (a[i][j].x == pX && a[i][j].y == pY && a[i][j].c == 0) {
774                 if (turn == true) a[i][j].c = -1; // Nếu lượt hiện hành là true thì c = -1
775                 else a[i][j].c = 1; // Nếu lượt hiện hành là false thì c = 1
776                 return a[i][j].c;
777             }
778         }
779     }
780     return 0;
781 }
782
783

```

Hình 17 hàm hasdraw/testboard/checkboard



`void saveGame() // lưu game`



```

828
829 void saveGame() {
830     system("cls");
831
832     system("cls");
833     Textcolor(244);
834     GotoXY(30, 9); cout << " 0000  0  0  0 0000    000  0  00  00 0000  ";
835     GotoXY(30, 10); cout << "   0  000  0 0  0 0    0 00  000  0 0 0 0  0 0  ";
836     GotoXY(30, 11); cout << " 0000  0  0  0 0000    000  0  0  0 0 0 0 0000  ";
837
838     GotoXY(40, 13);
839     Textcolor(241);
840     cout << "NHAP TEN FILE GAME : ";
841     cin >> fileName;
842     ofstream inp(fileName);
843     inp << turn << " " << point1 << " " << dem1 << " " << point2 << " " << dem2 << " " << playWithBot << " ";
844     for (int i = 0; i < BOARD_SIZE; i++) {
845         for (int j = 0; j < BOARD_SIZE; j++) {
846             inp << a[i][j].c << " ";
847         }
848     }
849     inp.close();
850
851     ifstream has("textName.txt");
852     string x;
853     while (has >> x) if (x == fileName) return;
854     has.close();
855
856     ofstream f;
857     f.open("textName.txt", ios::app);
858     f << fileName << '\n';
859     f.close();
860 }
861
862
863
864
865 void loadGame() {
866     system("cls");
867     Textcolor(244);
868     GotoXY(30, 9); cout << "00   00   0  000   000  0  00  00 0000  ";
869     GotoXY(30, 10); cout << "00   0  0  000  0 0   0 00  000  0 0 0 0  0 0  ";
870     GotoXY(30, 11); cout << "000  00  0  0 000   000  0  0  0 0 0 0 0000  ";
871
872     GotoXY(35, 13); cout << "File game before";
873     GotoXY(40, 14);
874     ifstream has("textName.txt");
875

```

Hình 18 save game

//lưu tên file người nhập vào file textName.txt

//tạo 1 file mới có tên là tên mà người dùng nhập và lưu tất cả dữ liệu của ván đấu vào file đó

`void loadGame()` // load game đã lưu

```

865
866 void loadGame() {
867     system("cls");
868     Textcolor(244);
869     GotoXY(30, 9); cout << "00  00  0  000  000  0  00  00  00  0000  ";
870     GotoXY(30, 10); cout << "00  0 0  000  0 0  0 00  000  0 0 0 0  0 0  ";
871     GotoXY(30, 11); cout << "000  00  0 0 000  000  0  0  0 0 0  0000  ";
872
873     GotoXY(35, 13); cout << "File game before";
874     GotoXY(40, 14);
875     ifstream has("textName.txt");
876     int dem = 0;
877     while (has >> str[++dem]) cout << str[dem], GotoXY(40, 14 + dem);
878     has.close();
879
880     while (1) {
881         GotoXY(40, 15 + dem);
882         Textcolor(7 + 241);
883         cout << "NHAP TEN FILE GAME :  ";
884         cin >> fileName;
885
886         bool ok = false;
887         for (int i = 1; i <= dem; i++) {
888             if (str[i] == fileName) ok = true;
889         }
890
891         if (!ok) {
892             GotoXY(40, 15 + dem);
893             Textcolor(248);
894             cout << "File game khong ton tai ";
895             int x = _getch();
896             if (x == 27) {
897                 isLoad = false;
898                 system("cls");
899                 return;
900             }
901         }
902         else break;
903     }
904
905     ifstream out(fileName); // load du lieu tu file
906     out >> turn >> point1 >> dem1 >> point2 >> dem2 >> playWithBot;
907     for (int i = 0; i < BOARD_SIZE; i++) {
908         for (int j = 0; j < BOARD_SIZE; j++) {
909             out >> a[i][j].c;
910         }
911     }
912 }
  
```

Hình 19 load game

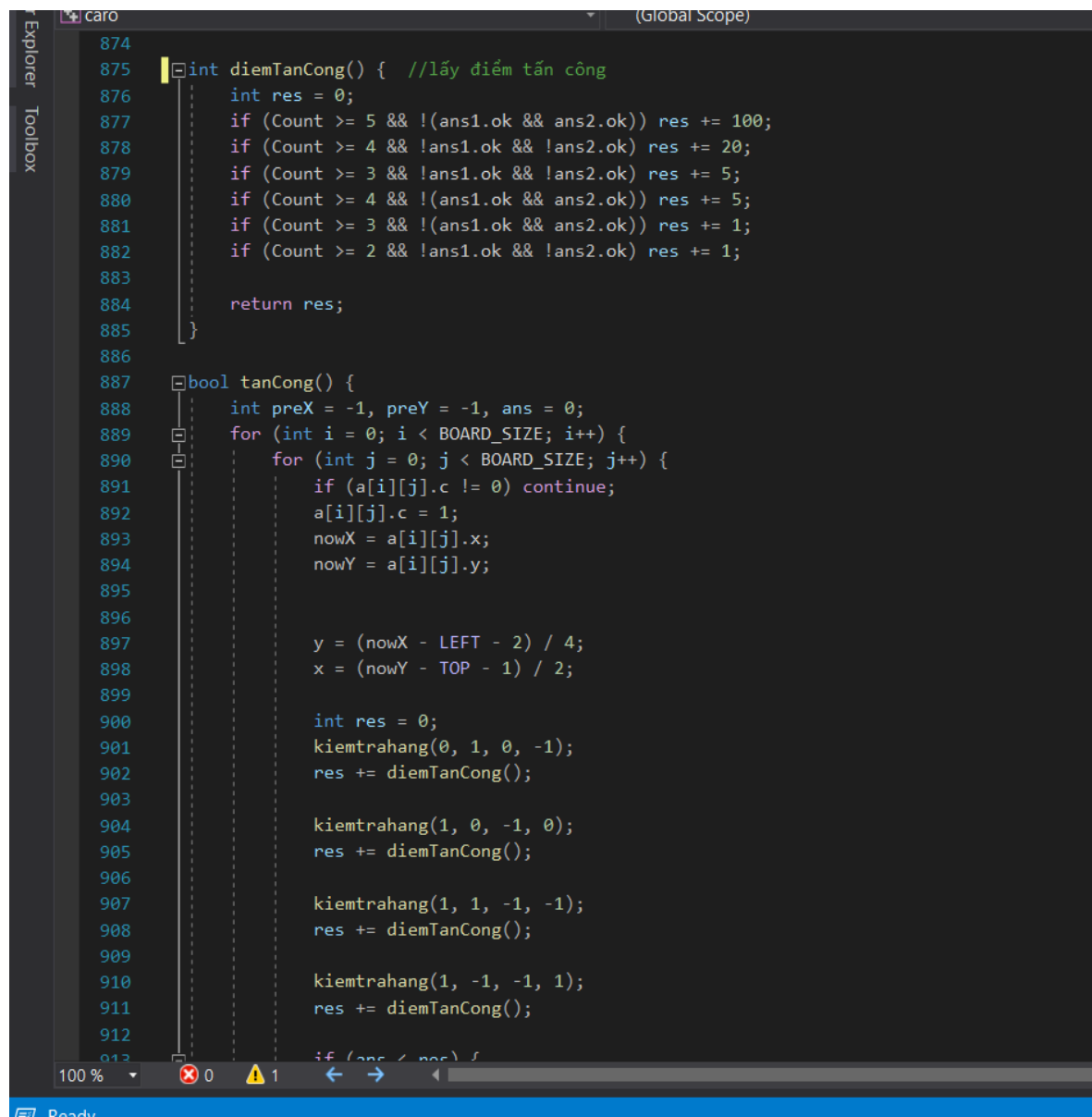
//nếu người dùng nhập tên file có tên trong file textName.txt thì sẽ mở dữ liệu ván đấu đã lưu với tên file người dùng nhập

//trong trường hợp người dùng nhập tên file không có trong textName.txt thì sẽ báo file không tồn tại

//máy tấn công

int diemTanCong() //tìm điểm tấn công

bool tanCong() //tấn công dựa vào điểm tấn công



```
874
875 int diemTanCong() { //lấy điểm tấn công
876     int res = 0;
877     if (Count >= 5 && !(ans1.ok && ans2.ok)) res += 100;
878     if (Count >= 4 && !ans1.ok && !ans2.ok) res += 20;
879     if (Count >= 3 && !ans1.ok && !ans2.ok) res += 5;
880     if (Count >= 4 && !(ans1.ok && ans2.ok)) res += 5;
881     if (Count >= 3 && !(ans1.ok && ans2.ok)) res += 1;
882     if (Count >= 2 && !ans1.ok && !ans2.ok) res += 1;
883
884     return res;
885 }
886
887 bool tanCong() {
888     int preX = -1, preY = -1, ans = 0;
889     for (int i = 0; i < BOARD_SIZE; i++) {
890         for (int j = 0; j < BOARD_SIZE; j++) {
891             if (a[i][j].c != 0) continue;
892             a[i][j].c = 1;
893             nowX = a[i][j].x;
894             nowY = a[i][j].y;
895
896
897             y = (nowX - LEFT - 2) / 4;
898             x = (nowY - TOP - 1) / 2;
899
900             int res = 0;
901             kiemtrahang(0, 1, 0, -1);
902             res += diemTanCong();
903
904             kiemtrahang(1, 0, -1, 0);
905             res += diemTanCong();
906
907             kiemtrahang(1, 1, -1, -1);
908             res += diemTanCong();
909
910             kiemtrahang(1, -1, -1, 1);
911             res += diemTanCong();
912
913             if (ans < res) {
```

Hình 20 các hàm máy tấn công

`void Bot()` //chọn vị trí thích hợp cho máy(easy) đánh (phòng thủ và tấn công).

```

931 void Bot() {
932     if (roadOfBot1.size()) {
933         int ran = rand() % roadOfBot1.size(); //random cách đi cho máy
934         _POINT& c = a[roadOfBot1[ran].val][roadOfBot1[ran].ok]; //máy đánh vào vị trí đã được random
935         c.c = 1;
936         GotoXY(c.x, c.y);
937
938         Textcolor(7 + 239);
939         cout << "0";
940         Textcolor(6 + 239);
941         chiahang();
942         dem2++;
943         bangdiem2();
944
945
946
947         nowX = c.x;
948         nowY = c.y;
949     }
950     else {
951         if (tanCong()) {
952
953             Textcolor(7 + 239);
954             cout << "0";
955             Textcolor(6 + 239);
956             chiahang();
957             dem2++;
958             bangdiem2();
959         }
960         else
961             if (roadOfBot2.size()) {
962                 int ran = rand() % roadOfBot2.size();
963                 _POINT& c = a[roadOfBot2[ran].val][roadOfBot2[ran].ok];
964                 c.c = 1;
965                 GotoXY(c.x, c.y);
966
967                 Textcolor(7 + 239);
968                 cout << "0";
969                 Textcolor(6 + 239);
970                 chiahang();
971                 dem2++;
972                 bangdiem2();
973
974                 nowX = c.x;
975                 nowY = c.y;
976             }
977         else {
978             _POINT c;
979             bool ok = false;
980             for (int i = 0; i < BOARD_SIZE; i++) {
981                 for (int j = 0; j < BOARD_SIZE; j++) {
982                     if (a[i][j].c == 0) c = a[i][j];
983                     if (a[i][j].c == -1) { ok = true; break; }
984                 }
985                 if (ok) break;
986             }
987         }

```

Hình 21 hàm thiết lập máy

`bool checkWinOrLose() ()` //kiểm tra thắng hay thua. Sau đó nếu người dung không thoát ra menu sẽ tạo lại game mới và cập nhật ván thắng thua cho đối tượng chơi

`void DrawBoard(int pSize)` //vẽ bàn cờ bằng các vòng for và bảng mã ascii

```

caro
409 void DrawBoard(int pSize) {
410     Textcolor(242);
411
412     for (int i = 0; i < pSize * 2; i++)
413     {
414         // Vẽ cột phải
415         Sleep(20);
416         GotoXY(pSize * 4 + 1, i + 1);
417         if ((i + 1) % 2 == 1)
418             cout << char(186);
419         else
420             cout << char(182);
421     }
422
423     for (int i = 0; i < pSize * 2; i = i + 2)
424     {
425         // Dòng thượng
426         Sleep(20);
427         GotoXY(1, i);
428         for (int j = 0; j < pSize * 4; j++)
429         {
430             if (j % 4 == 0)
431                 cout << char(197);
432             else
433                 cout << char(196);
434         }
435     }
436
437     for (int i = 0; i < pSize * 2; i++)
438     {
439         // Vẽ cột trái
440         Sleep(20);
441         GotoXY(1, i + 1);
442         if ((i + 1) % 2 == 1)
443             cout << char(186);
444         else
445             cout << char(199);
446     }
447
448     for (int i = 1; i < pSize * 2; i = i + 2)
449     {
450         for (int j = 0; j < pSize * 4; j = j + 4)
451         {
452             // Cột thượng
453             Sleep(20);
454             GotoXY(j + 1, i);
455             if (j == 0)
456                 continue;
457             cout << char(179);
458         }
459     }
460
461     GotoXY(1, 0);
462     for (int i = 0; i < pSize * 4; i++)
463     {
464         // Vẽ dòng đầu
465         Sleep(20);
466         if (i % 4 == 0)
467             cout << char(209);
468         else
469             cout << char(205);
470     }
471     GotoXY(1, 0); cout << char(281);
472     GotoXY(pSize * 4 + 1, 0); cout << char(187);
473
474     GotoXY(1, pSize * 2);
475     for (int i = 0; i < pSize * 4; i++)
476     {
477         // Vẽ dòng cuối
478         Sleep(20);
479         if (i % 4 == 0)
480             cout << char(207);
481         else
482             cout << char(205);
483     }
484     if (turn == false)
485     {
486         Textcolor(246);
487         chiahang();
488     }
489 }

```

Hình 22 hàm vẽ bàn cờ

`void DrawXO()` // vẽ các quân đã đánh khi người dùng load lại game đã lưu

```

500
501 void DrawXO() {
502     for (int i = 0; i < BOARD_SIZE; i++) {
503         for (int j = 0; j < BOARD_SIZE; j++) {
504             if (a[i][j].c != 0) {
505                 GotoXY(a[i][j].x, a[i][j].y);
506                 if (a[i][j].c == -1) {
507                     Textcolor(245);
508                     cout << "X";    //vẽ x vào tọa độ con trỏ
509                 }
510                 else {
511                     Textcolor(246);
512                     cout << "O";    //vẽ o vào tọa độ con trỏ
513                 }
514             }
515         }
516     }
517 }
518

```

Hình 23 hàm vẽ XO

`int newGame()` // game mới... sử dụng các hàm đã cài đặt để thiết lập một game mới

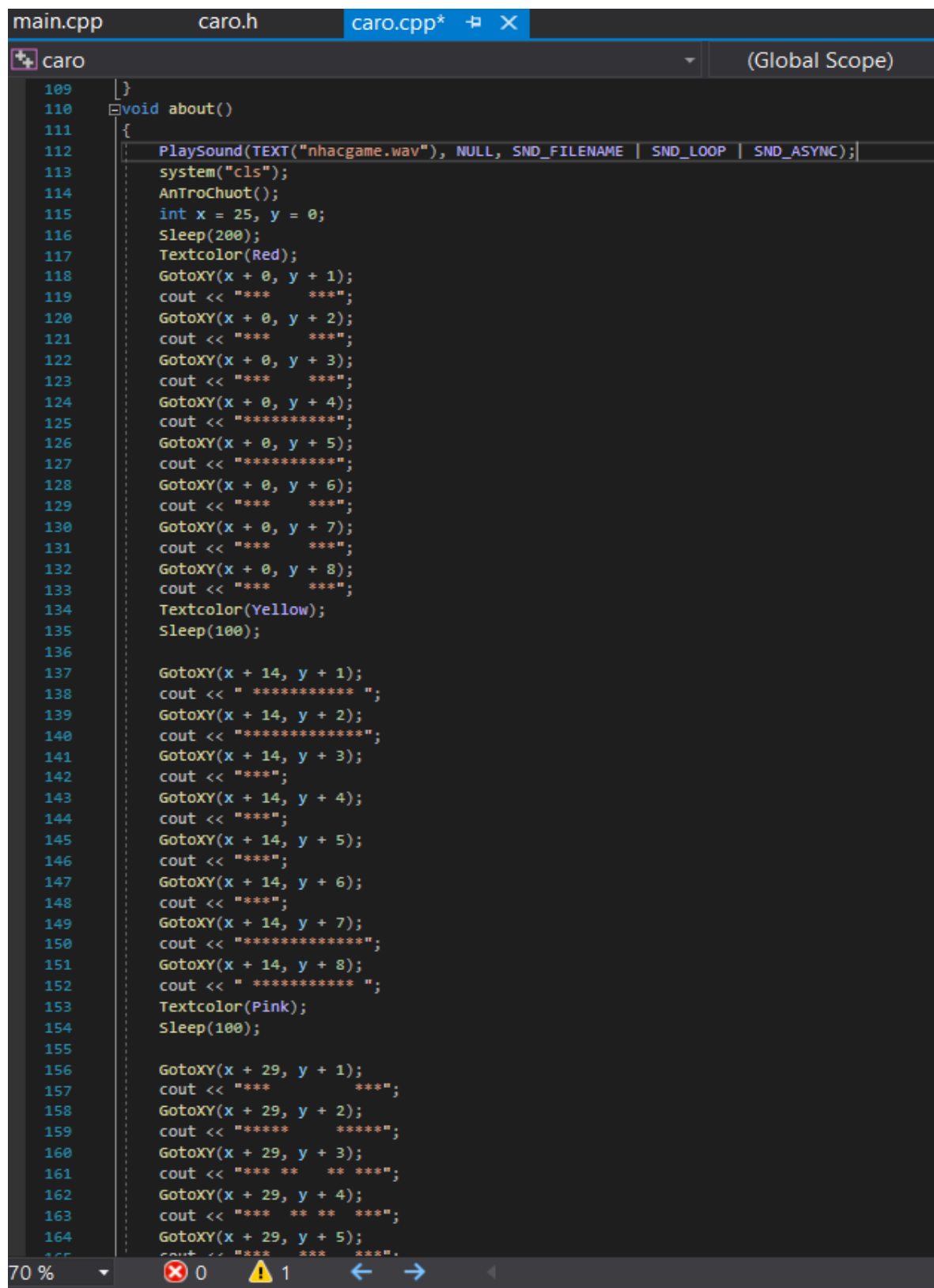
```

1027
1028 int newGame() {
1029     AnTroChuat();
1030     system("Color F3");
1031
1032
1033     StartGame();
1034     bool validEnter = true;
1035     while (1) {
1036         if (playWithBot && turn == false) {
1037             int x = nowX, y = nowY;
1038             Bot();
1039             if (checkWinOrLose()) return 0;
1040             nowX = x, nowY = y;
1041             GotoXY(nowX, nowY);
1042         }
1043
1044         _COMMAND = toupper(_getch()); //command: nhan ki tu ma nguoi dung nhap
1045         switch (_COMMAND) { //toupper: chuyen ki tu sang viet hoa
1046             case 27:
1047                 ExitGame();
1048                 return 0;
1049             case 'A':
1050             {
1051                 MoveLeft();
1052                 PlaySound(TEXT("dichuyen.wav"), NULL, SND_FILENAME | SND_ASYNC);
1053                 break;
1054             }
1055             case 'W': {
1056                 MoveUp();
1057                 PlaySound(TEXT("dichuyen.wav"), NULL, SND_FILENAME | SND_ASYNC);
1058                 ; break;
1059             }
1060             case 'S': {
1061                 MoveDown();
1062                 PlaySound(TEXT("dichuyen.wav"), NULL, SND_FILENAME | SND_ASYNC);
1063                 break; }
1064             case 'D': {
1065                 MoveRight();
1066                 PlaySound(TEXT("dichuyen.wav"), NULL, SND_FILENAME | SND_ASYNC);

```

Hình 24 hàm newgame

`void about() // thông tin về Nhóm`



The screenshot shows a C++ IDE with three tabs: 'main.cpp', 'caro.h', and 'caro.cpp\*'. The 'caro.cpp' tab is active, showing the 'about' function. The function starts with a comment '// thông tin về Nhóm' and a call to 'PlaySound'. It then uses 'system' to clear the screen and 'GotoXY' to move the cursor. It prints several lines of text, including '\*\*\*', '\*\*\*\*\*', and '\*\*\*\*\*', with different colors (Red, Yellow, Pink) and delays (Sleep). The function ends with a return statement.

```
109 }
110 void about()
111 {
112     PlaySound(TEXT("nhacgame.wav"), NULL, SND_FILENAME | SND_LOOP | SND_ASYNC);
113     system("cls");
114     AnTroChuat();
115     int x = 25, y = 0;
116     Sleep(200);
117     Textcolor(Red);
118     GotoXY(x + 0, y + 1);
119     cout << "***   ***";
120     GotoXY(x + 0, y + 2);
121     cout << "***   ***";
122     GotoXY(x + 0, y + 3);
123     cout << "***   ***";
124     GotoXY(x + 0, y + 4);
125     cout << "*****";
126     GotoXY(x + 0, y + 5);
127     cout << "*****";
128     GotoXY(x + 0, y + 6);
129     cout << "***   ***";
130     GotoXY(x + 0, y + 7);
131     cout << "***   ***";
132     GotoXY(x + 0, y + 8);
133     cout << "***   ***";
134     Textcolor(Yellow);
135     Sleep(100);
136
137     GotoXY(x + 14, y + 1);
138     cout << " ***** ";
139     GotoXY(x + 14, y + 2);
140     cout << "*****";
141     GotoXY(x + 14, y + 3);
142     cout << "***";
143     GotoXY(x + 14, y + 4);
144     cout << "***";
145     GotoXY(x + 14, y + 5);
146     cout << "***";
147     GotoXY(x + 14, y + 6);
148     cout << "***";
149     GotoXY(x + 14, y + 7);
150     cout << "*****";
151     GotoXY(x + 14, y + 8);
152     cout << " ***** ";
153     Textcolor(Pink);
154     Sleep(100);
155
156     GotoXY(x + 29, y + 1);
157     cout << "***   ***";
158     GotoXY(x + 29, y + 2);
159     cout << "*****   *****";
160     GotoXY(x + 29, y + 3);
161     cout << "*** **   ** ***";
162     GotoXY(x + 29, y + 4);
163     cout << "*** ** **   ***";
164     GotoXY(x + 29, y + 5);
165     cout << "***   ***   ***";
166 }
```

Hình 25 hàm about



```
void Help() //in ra các hướng dẫn cho người chơi
```

```
void in(int x, int y) //in ra màn hình chính
```

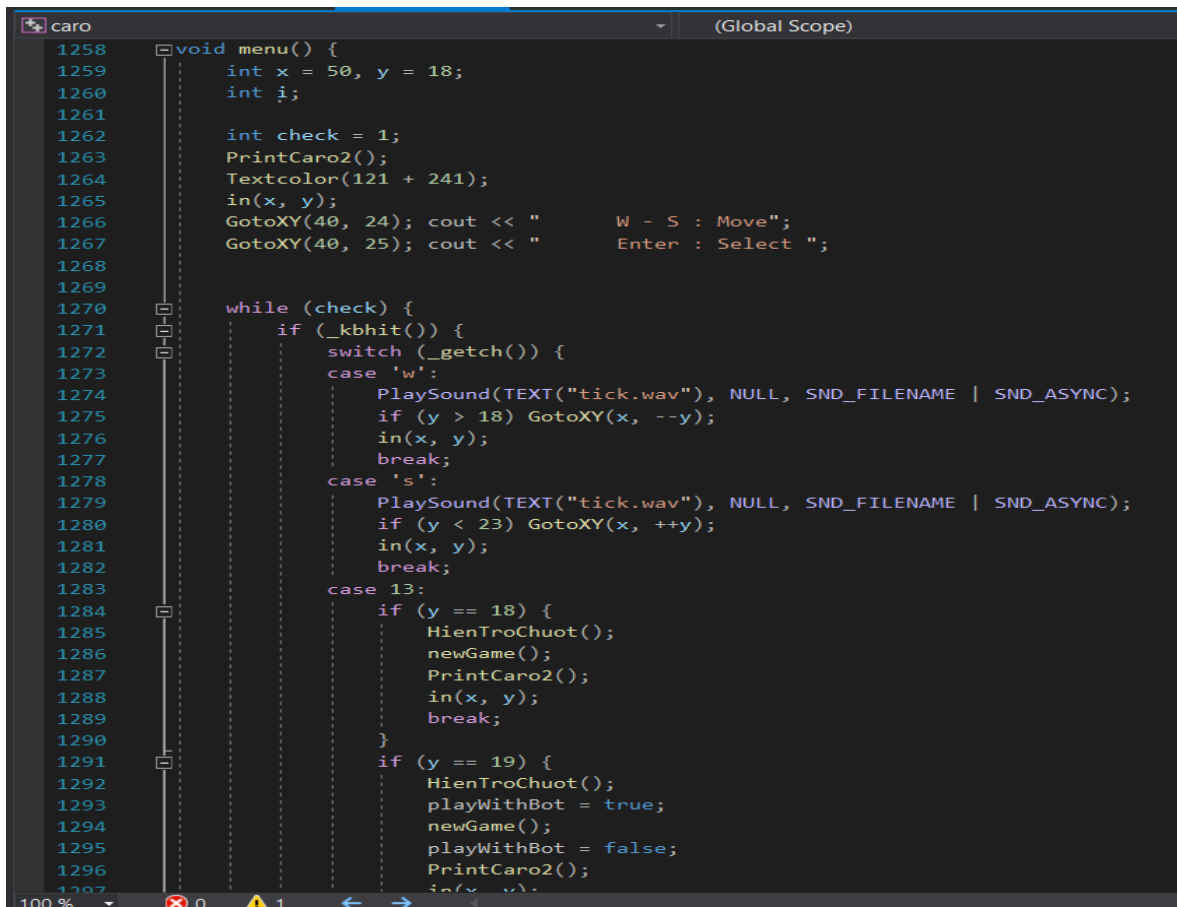
```

1109     }
1110 }
1111
1112 string chuoi[] = { "New Game (PVP)", "New Game vs Bot", "Load Game", "Help", "About", "Exit" };
1113 void in(int x, int y) {
1114     AnTroChuat();
1115     for (int i = 18; i <= 23; i++) {
1116         if (y == i) Textcolor(12 + 241);
1117         else Textcolor(7 + 241);
1118         GotoXY(x, i); cout << chuoi[i - 18];
1119     }
1120 }
1121
1122 void Help()
1123 {
1124     PlaySound(TEXT("win.wav"), NULL, SND_FILENAME | SND_LOOP | SND_ASYNC);
1125
1126     system("cls");
1127     AnTroChuat();
1128     int x = 25, y = 0;
1129     Sleep(50);
1130     Textcolor(Red);
1131
1132     GotoXY(x + 0, y + 3);
1133     cout << "***  ***";
1134     GotoXY(x + 0, y + 4);
1135     cout << "***  ***";
1136     GotoXY(x + 0, y + 5);
1137     cout << "***  ***";
1138     GotoXY(x + 0, y + 6);
1139     cout << "*****";
1140     GotoXY(x + 0, y + 7);
1141     cout << "*****";
1142     GotoXY(x + 0, y + 8);
1143     cout << "***  ***";
1144     GotoXY(x + 0, y + 9);
1145     cout << "***  ***";
1146     GotoXY(x + 0, y + 10);
1147     cout << "***  ***";
1148     Textcolor(Yellow);

```

Hình 26 hàm in và help

`void menu()` // từng bộ phận của menu với việc chỉ cần gọi các hàm đã cài đặt trên để tạo thành một tính năng của game



```
1258 void menu() {
1259     int x = 50, y = 18;
1260     int i;
1261
1262     int check = 1;
1263     PrintCaro2();
1264     Textcolor(121 + 241);
1265     in(x, y);
1266     GotoXY(40, 24); cout << "      W - S : Move";
1267     GotoXY(40, 25); cout << "      Enter : Select ";
1268
1269
1270     while (check) {
1271         if (_kbhit()) {
1272             switch (_getch()) {
1273                 case 'w':
1274                     PlaySound(TEXT("tick.wav"), NULL, SND_FILENAME | SND_ASYNC);
1275                     if (y > 18) GotoXY(x, --y);
1276                     in(x, y);
1277                     break;
1278                 case 's':
1279                     PlaySound(TEXT("tick.wav"), NULL, SND_FILENAME | SND_ASYNC);
1280                     if (y < 23) GotoXY(x, ++y);
1281                     in(x, y);
1282                     break;
1283                 case 13:
1284                     if (y == 18) {
1285                         HienTroChuat();
1286                         newGame();
1287                         PrintCaro2();
1288                         in(x, y);
1289                         break;
1290                     }
1291                     if (y == 19) {
1292                         HienTroChuat();
1293                         playWithBot = true;
1294                         newGame();
1295                         playWithBot = false;
1296                         PrintCaro2();
1297                         in(x, y);
1298                     }
1299             }
1300         }
1301     }
1302 }
```

```

main.cpp      caro.h      caro.cpp*  # X
caro
1296          PrintCaro2();
1297          in(x, y);
1298          break;
1299      }
1300      if (y == 20) {
1301          HienTroChuat();
1302          loadGame();
1303          if (isLoad) newGame();
1304          PrintCaro2();
1305
1306          in(x, y);
1307          break;
1308      }
1309      if (y == 21) {
1310          Help();
1311          PrintCaro2();
1312
1313          in(x, y);
1314          break;
1315      }
1316      if (y == 22)
1317      {
1318          about();
1319          PrintCaro2();
1320          in(x, y);
1321          break;
1322      }
1323      if (y == 23) {
1324          exit(0);
1325      }
1326      break;
1327  }
1328  }
1329  }
1330  }
  
```

Hình 27 hàm menu

```
int main() //main
```

```

1331
1332  int main()
1333  {
1334      FixConsoleWindow();
1335      Textcolor(241);
1336      system("cls");
1337      menu();
1338  }
1339
1340
  
```

Hình 28 hàm main

## TÀI LIỆU THAM KHẢO

### TÀI LIỆU THAM KHẢO :

1/Tài liệu hướng dẫn của giảng viên: DoAnCaro.pdf

2/Hướng dẫn vẽ bàn cờ: (12/2020)

<https://www.youtube.com/watch?v=AOvXmLpucXk>

3/Hướng dẫn chen âm thanh: (12/2020)

<https://daynhauhoc.com/t/long-nhac-mp3-vao-ung-dung-console-c/34035>

4/Xử lý thắng thua: (12/2020)

<https://www.youtube.com/watch?v=ZcXuXHFZNiQ&t=54s>

5/Kỹ thuật xử lý tập tin: (12/2020)

[https://www.youtube.com/watch?v=R1H\\_McNO6LQ](https://www.youtube.com/watch?v=R1H_McNO6LQ)