

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

CỜ CARO



Họ và tên: Nguyễn Đỗ Chí Thảo

MSSV: 1712159

I/ Phân tích các hàm:

1/ Lớp Common:

```
class _Common {
public:

    static void fixConsoleWindow();
    enum
    {
        KEY_ESC = 27,
        ARROW_UP = 256 + 72,
        ARROW_DOWN = 256 + 80,
        ARROW_LEFT = 256 + 75,
        ARROW_RIGHT = 256 + 77,
        ENTER = 32
    };

    static void gotoXY(int, int);
        /*Dùng để di chuyển trỏ chuột đến vị trí thích hợp
        trên bàn cờ */
    static void txtColor(int color);
        //Dùng để đổi màu
    static void re_sizeConsole(int width, int height);
        //Thay đổi kích thước của màn hình game
    static void X(int x, int y);
        //In ra kí tự X đặc biệt
    static void O(int x, int y);
        //In ra kí tự O đặc biệt
    static int Menu(int x, int y);
        //Tạo màn hình menu của game
    static void Introduction(int x, int y);
        //Tạo giới thiệu sơ bộ về game: cách chơi, người
        thực hiện ...
    static int get_code(void);
        //Dùng để nhận dạng kí tự điều khiển menu
    static int Menu1(int, int);
        //Màn hình menu sau khi chọn Play ở menu
    static int Menu3(int, int);
        //Màn hình menu sau khi chọn Sound ở menu
```

```

static int Menu4(int, int);
    //Màn hình menu sau khi chọn Sound ở menu
static void menu(int arrow, string menu_options[], int
menu_size, int x, int y);
    /*Xử lý các hiển thị đặc biệt cho các di chuyển ở
menu, khi di chuyển tới mục nào thì mục đó đổi màu và thụt
vào trong.*/
static void Caro(int X, int Y)
    //In ra dòng chữ CARO đặc biệt
};

```

2/ Lớp Point:

```

class _Point {
    int _x, _y, _check;
    // Hoành độ, tung độ, lượt hiện hành
public:
    _Point();
    // Khởi tạo giá trị một điểm
    _Point(int, int);
    // Xóa một điểm
    bool setCheck(int);
    // Khởi gán giá trị lượt hiện hành
    int getX();
    // Trả về hoành độ
    int getY();
    // Trả về tung độ
    int getCheck();
    // Trả về giá trị lượt hiện hành
    void setX(int);
    // Khởi gán hoành độ
    void setY(int);
    // Khởi gán tung độ
};

```

3/ Lớp Board:

```

class _Board {

```

```

private:
    int _size;
        // Kích thước bàn cờ
    int _left, _top;
        // Tọa độ điểm trái trên
    _Point** _pArr;
        // Mảng bàn cờ
public:
    int get_size();
        // Trả về kích thước bàn cờ
    int getLeft();
        //Trả về hoành độ điểm trái trên bàn cờ
    int getTop();
        //Trả về tung độ điểm trái trên của bàn cờ
    int getXAt(int, int);
        //Trả về hoành độ của một vị trí trên bàn cờ
    int getYAt(int, int);
        //Trả về tung độ của một vị trí trên bàn cờ
    _Board(int, int, int);
        //Khởi tạo bàn cờ
    ~_Board();
        //Xóa bàn cờ
    void resetData();
        //Reset lại tất cả dữ liệu của bàn cờ
    void drawBoard();
        //Vẽ bàn cờ
    int checkBoard(int, int, bool);
        //Kiểm tra lượt hiện hành
    int testBoard(int, int);
        //Kiểm tra thắng, thua, hòa
    void saveGame();
        // Lưu game
    void loadGame();
        // Load game
    _Point* findBestMove();
        // Tìm nước đi tốt nhất cho máy
    long Diem_TC_Doc(int x, int y);
        // Tìm điểm tấn công theo hàng dọc
    long Diem_TC_Ngang(int, int);

```

```

        // Tìm điểm tấn công theo hàng ngang
long Diem_TC_CheoNguoc(int, int);
        // Tìm điểm tấn công theo đường chéo ngược
long Diem_TC_CheoChinh(int, int);
        // Tìm điểm tấn công theo đường chéo chính
long Diem_PT_Doc(int x, int y);
        // Tìm điểm phòng thủ theo hàng dọc
long Diem_PT_Ngang(int, int);
        // Tìm điểm phòng thủ theo hàng ngang
long Diem_PT_CheoNguoc(int, int);
        // Tìm điểm phòng thủ theo đường chéo ngược
long Diem_PT_CheoChinh(int, int);
        // Tìm điểm phòng thủ theo đường chéo chính
_Point Random();
        // Phát sinh ngẫu nhiên nước đi
long Mangdiemphongthu[13] = {
0,3,24,192,1536,12288,98304,786432,6291456,50331648,40265318
4,402653184,402653184};
long Mangdiemtancong[13] ={0,1, 9, 81, 729,
6561,59049,531441,4782969,43046721,387420489,387420489,38742
0489 };

```

4/ Lớp Game:

```

class _Game {
    _Board* _b;
        // một bàn cờ
int _mode;
        // Chế độ chơi (với người hay với máy)
int _winX;
        // Số ván thắng của người X
int _win0;
        // Số ván thắng người O
int _resX;
        // Số nước người X đã đi
int _res0;
        // Số nước người O đã đi
bool _turn;

```

```

        /* lượt chơi: true lượt người một & false lượt
người thứ hai */
int _x, _y;
    // Tọa độ hiện hành của nháy chuột
int _command;
    // phím gõ từ người dùng
bool _loop;
    // Biến quyết định thoát game hay không
public:
    _Game(int, int, int);
        // khởi tạo trò chơi
    ~_Game();
        // Xóa toàn bộ trò chơi
int mode() {
    return _mode;
}
    //Trả về chế độ hiện hành
void setmode(int x) { _mode = x; }
    // Khởi gán chế độ
int getCommand();
    // Trả về phím gõ từ người dùng
bool isContinue();
    // Kiểm tra có tiếp tục chơi
char waitKeyBoard();
    // Hàm nhận phím từ người dùng
char askContinue();
    /* Hàm hỏi người chơi có tiếp tục chơi sau khi
thắng/ thua/ hòa */
void startGame();
    // Hàm bắt đầu game
void exitGame();
    // Hàm thoát game
int processFinish();
    // Xử lý khi thắng/ thua/ hòa
bool processCheckBoard();
    /*Xử lý khi người chơi ấn enter( in ra X hoặc O),
đồng thời xử lý các thông tin về số nước đã đi của hai
người chơi*/
void moveRight();

```

```

        // Di chuyển sang phải
void moveLeft();
        // Di chuyển sang trái
void moveUp();
        // Di chuyển lên trên
void moveDown();
        // Di chuyển xuống dưới
void loadGame();
        // tải Game
void saveGame();
        //Lưu Game
void vsCom();
        // Chế độ đánh với máy
};

```

II/ Yêu cầu đồ án:

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

-Đề lưu trò chơi chúng ta sẽ duyệt hết bàn cờ(_pArr[i][j]), rồi lưu giá trị các quân cờ trên bàn cờ (_pArr[i][j].getCheck()), tọa độ của quân cờ (_pArr[i][j].getX(), _pArr[i][j].getY()) thế các giá trị về nhận biết tại 1 ô trên bàn cờ đã đi hay chưa (_check) vào file Save.txt, và các giá trị về lượt đi hiện hành (_turn), số bàn thắng của hai người chơi(_winX, _winY), số nước đi của hai người chơi(_resX, _resY) và chế độ chơi(_mode)

-Đề tải trò chơi, thì chúng ta sẽ xuất từ file Save.txt,các quân cờ đã được đánh ở các tọa độ nào. Sau đó sẽ in ra bàn cờ các quân cờ theo dữ liệu đã xuất ra từ file. Tiếp theo sẽ in các thông tin của từng người chơi lên màn hình giao diện chơi.

2/ Nhận biết thắng/thua/hòa:

-Ta sẽ xét người đó thắng/ thua/ hòa ngay khi người chưa vừa đánh quân cờ và chỉ xét tới trường hợp người vừa mới xong sẽ thắng hay là hòa. Bởi nếu người đó vừa mới chơi xong thì chắc chắn không thể thua.

Đề nhận biết thắng/hòa, chúng ta sẽ xây dựng hàm **int** testBoard(**int**, **int**) trong lớp _Board. Trong hàm testBoard(**int**, **int**) chúng ta sẽ duyệt ô vừa mới đánh xong, theo các đường chéo chính, chéo phụ, thẳng, ngang. Nếu số quân cờ giống quân vừa mới đánh bằng 5 và không bị chặn hai đầu bởi quân còn lại thì sẽ trả về giá trị của quân cờ đó (_pArr[i][j].getCheck()). Nếu bàn cờ không còn ô trống mà người chơi

không thắng thì ta sẽ xuất ra là 0(hòa), ngược lại nếu thì sẽ xuất ra 2(tiếp tục đánh)

3/ Xử lý hiệu ứng thắng/thua/hòa

Nếu có người chơi nào thắng, ta sẽ xóa vị trí của hai chữ X và O hiện hành (nằm phía dưới bàn cờ). Sau đó, xuất ra màn hình dòng chữ thông báo người chơi nào thắng cùng biểu tượng quân cờ của người chơi đó được nhập nháy liên tục, đồng thời sẽ tắt nhạc nền đang chạy để chạy nhạc thông báo chiến thắng. Sau khi thông báo xong, chúng ta sẽ xóa những hiệu ứng đó, để xuất ra dòng chữ hỏi người chơi có muốn chơi tiếp hay không? Nếu người chơi bấm y(yes) thì sẽ xóa sạch màn hình, reset lại toàn bộ dữ liệu của bàn cờ, in ra màn hình một giao diện mới, bàn cờ trống.



2 Player(X vs O)

#Người chơi X đã thắng : 1

#Người chơi X đã đi được: 5

#Người chơi O đã thắng : 0

#Người chơi O đã đi được: 4

W: Move Up

S: Move Down

A: Move Left

D: Move Right

Space: Save

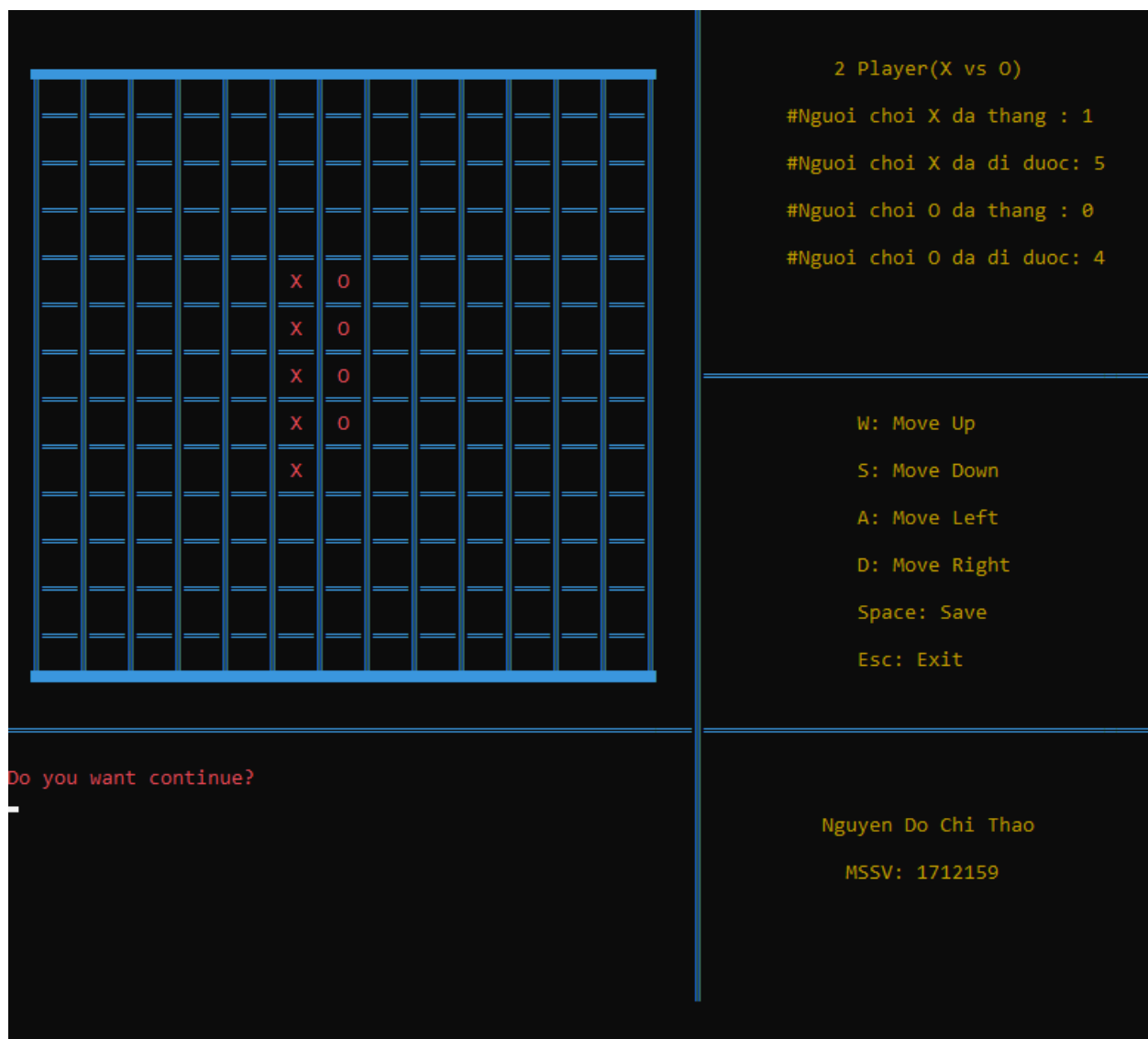
Esc: Exit

Người chơi 1 đã thắng và người chơi 0 đã thua

```
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
```

Nguyen Do Chi Thao

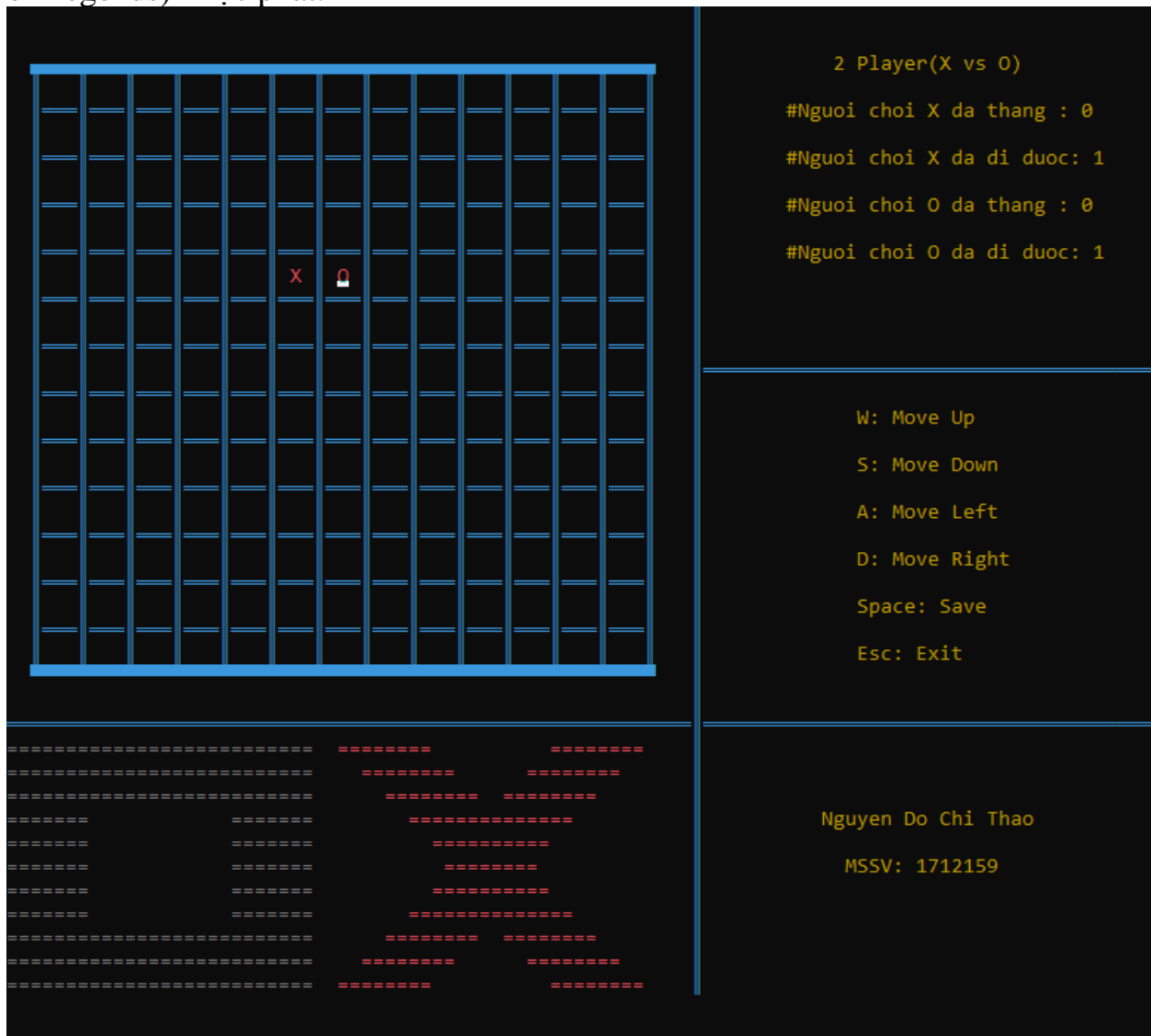
MSSV: 1712159



4. Xử lý giao diện màn hình khi chơi:

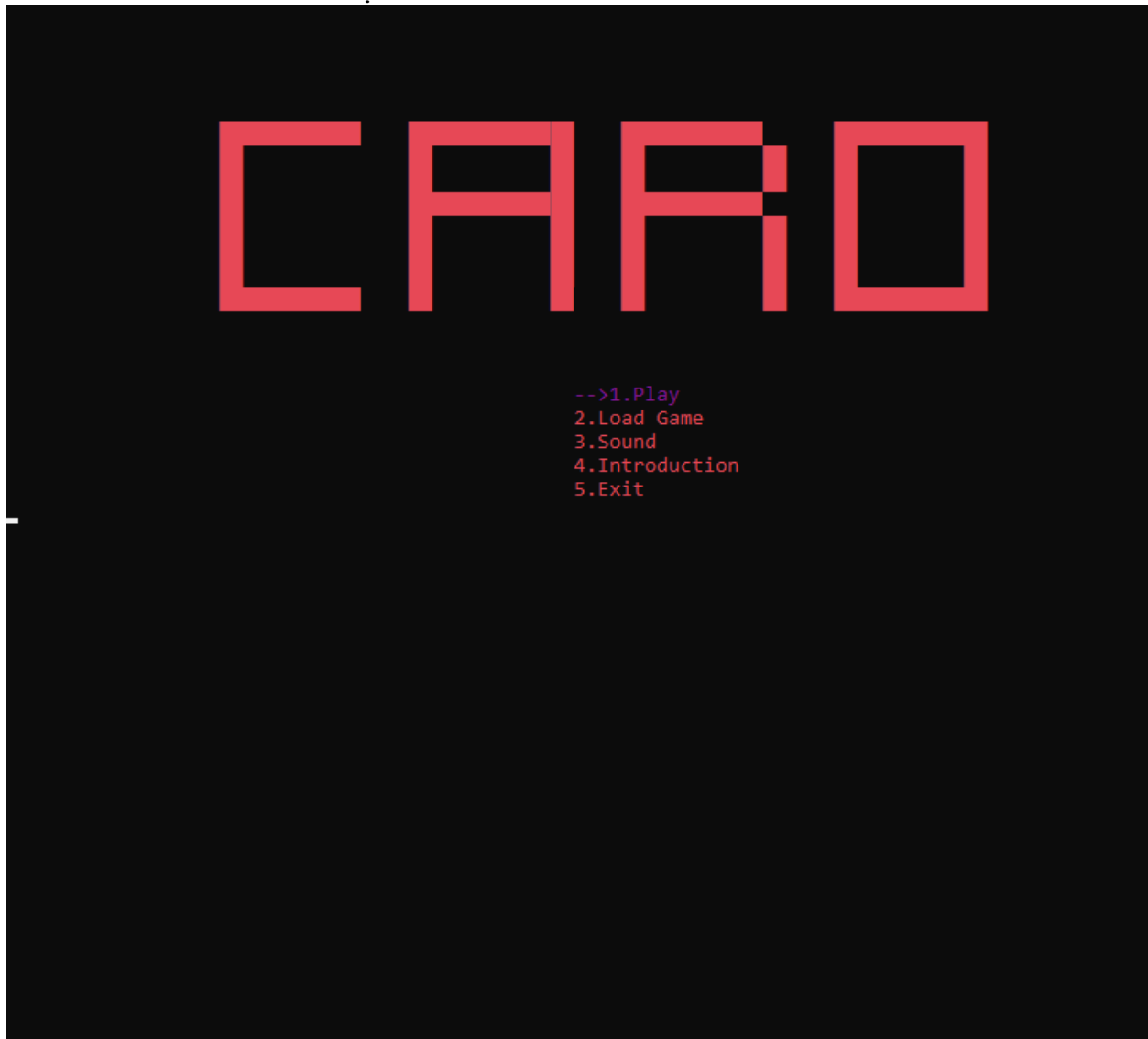
Màn hình khi chơi sẽ hiển thị các thông tin: số nước đã đi của hai người chơi, số ván thắng của hai người chơi, cách chơi ... Bên cạnh đó, ở góc dưới bàn cờ sẽ hiển thị hai kí hiệu X và O để thông báo tới lượt của người chơi nào (khi tới lượt của

người chơi nào thì kí hiệu cờ của người đó sáng lên). Ngoài ra, trong quá trình kể từ khi bắt đầu vào game thì sẽ có nhạc nền: “ K/DA POP/STARS” (nguồn: League of Legends) được phát.



5. Xử lý màn hình chính:

Màn hình chính sẽ hiển thị menu của trò chơi:



Menu sẽ được điều khiển bằng 4 phím mũi tên, nhấn Enter để chọn. Trong mỗi mục của menu sẽ có những chức năng khác nhau:

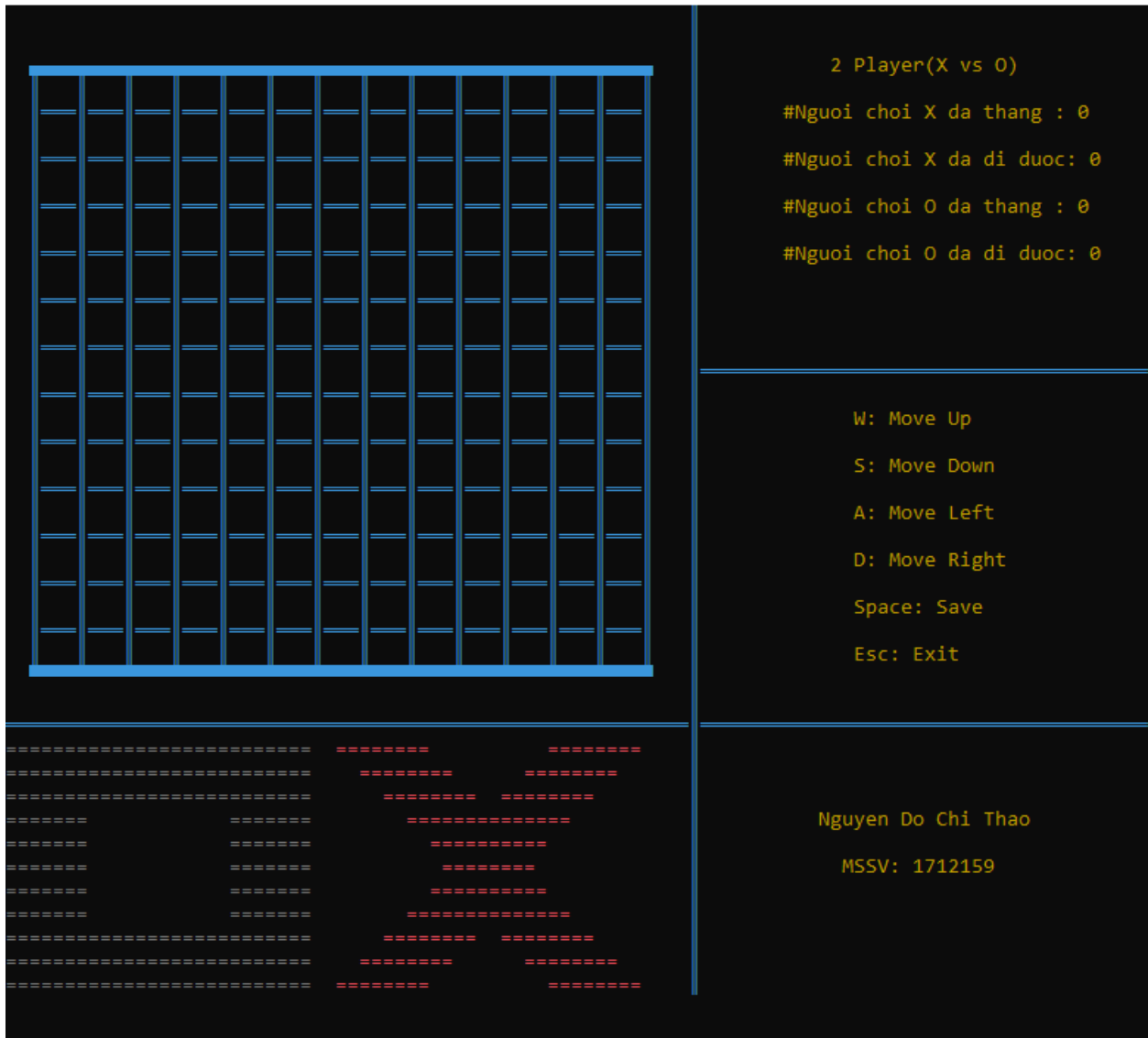
a/ Play:

CARDO

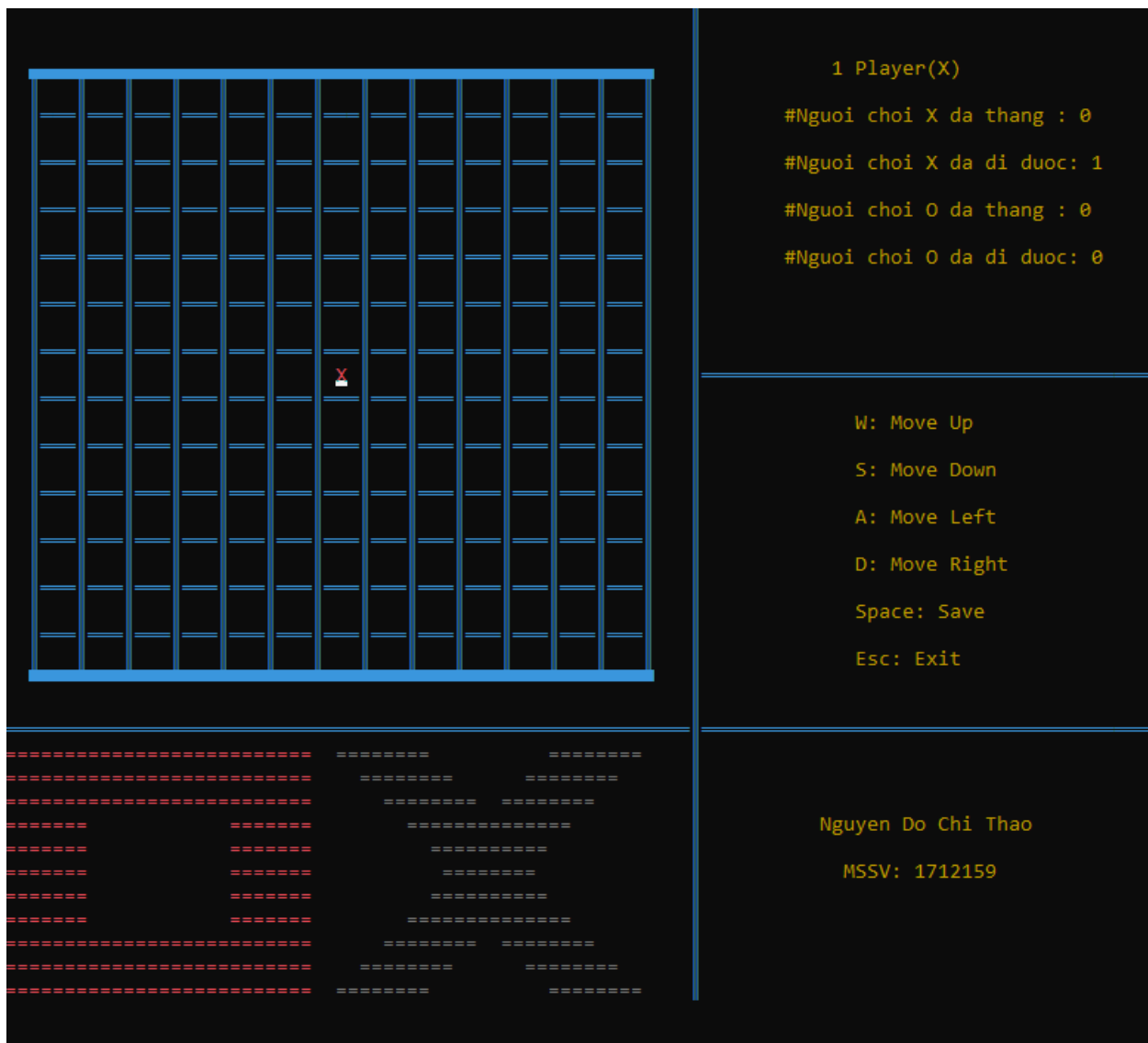
-->1.X vs 0
2.Vs Com

Chọn Play để bắt đầu chơi, trong Play sẽ có 2 chế độ:

- X vs O (chế độ 2 người chơi)



- Vs computer (chế độ 1 người chơi – đánh với máy): Chế độ này thì máy sẽ chơi quân X và đi trước



b/ Load game:

Sẽ tải lại dữ liệu của ván đã lưu rồi in lên màn hình.

c/ Sound:

Dùng để điều chỉnh âm thanh, người dùng chọn **On** để bật âm thanh, **Off** để tắt và **Return menu** để trở lại màn hình chính

CARO

-->1.On
2.Off
3. Return Menu

d/ Introduction:

-Giới thiệu luật chơi và người thực hiện đồ án. Chọn **Menu** để trở về màn hình chính

CARO

Người thực hiện: Nguyễn Do Chí Thao
MSSV: 1712159

Hướng dẫn:

Bạn hãy sử dụng 4 phím A: qua trái, D: qua phải, S: đi xuống, W: đi lên

Nếu bạn đạt được 5 quân trên một đường thẳng mà không bị chặn hai đầu, bạn sẽ chiến thắng

-->Menu

e/ Exit:

Chọn Exit để thoát game

6. Xử lý chơi với máy:

-Để xử lý chơi với máy, chúng ta sẽ dùng thuật toán tính điểm bàn cờ.

- Đầu tiên cần phải xây dựng hai mảng điểm tấn công và phòng thủ theo số quân X và O hiện có trên bàn cờ (dựa trên thực nghiệm để có mảng điểm chính xác).

```
long Mangdiemphongthu[13] = {  
0,3,24,192,1536,12288,98304,786432,6291456,50331648,402653184,402653184,402653184};  
long Mangdiemtancong[13] = {0,1, 9, 81, 729,  
6561,59049,531441,4782969,43046721,387420489,387420489,387420489 };
```


-Đối với mỗi ô cờ trên bàn cờ, chúng ta sẽ xét xem theo các đường chéo chính, chéo phụ, thẳng, ngang, có bao nhiêu quân giống với quân của máy, sau đó xuất ra điểm tấn công tương ứng trong mảng điểm tấn công. Tuy nhiên nếu bị chặn hai đầu thì điểm tấn công sẽ là 0. Tương tự cho trường hợp điểm phòng thủ.

Vd: Nếu hiện tại trên bàn cờ trên đường chéo chính tại một ô có 2 quân X và 3 quân O, không bị chặn hai đầu thì sẽ xuất ra `Mangdiemphongthu[2]` và `Mangdiemphongthu[3]`

-Tiếp theo, sẽ xây dựng một hàm `_Point* findBestMove()` trong lớp `_Board`. Hàm này sẽ làm nhiệm vụ duyệt hết bàn cờ, đối với mỗi ô sẽ so sánh tổng điểm tấn công và tổng điểm phòng thủ, rồi chọn điểm cho ô đó là điểm tấn công hay phòng thủ, giá trị bao nhiêu. Chọn ô có điểm cao nhất rồi cho máy đánh vào ô đó.

-Thuật toán này em đã tham khảo trên myclass.vn: <https://myclass.vn/tim-khoa-hoc-lap-trinh.html?k=caro>

Tài liệu tham khảo:

<https://myclass.vn/>

<https://www.youtube.com/watch?v=UOxkGD8qRB4>

<http://devnt.org/lap-trinh-giao-dien-console-ncurses-menu/>