

Trường Đại Học Khoa Học Tự Nhiên

ĐỀ TÀI MIỄN THI

MÃ P01

Sinh Viên Đinh Nguyên Khánh

MSSV 19120087

Hồ Chí Minh

18 6 2021

MỤC LỤC

- I. Quá trình hoàn thành đồ án
- II. Giải thích code
- III. Demo trò chơi

I. Quá trình hoàn thành đồ án:

- 9/6/2021 đến ngày 12/6/2021 hoàn thành cơ bản game
- 13/6/2021 debug chỉnh sửa
- 14/6/2021 debug
- 18/6/2021 debug và viết report

II. Giải thích code

Phần Header:

- Champion.h : thông tin 1 tướng
- Game.h : thông tin 1 game đấu
- Log.h: thông tin 1 log
- Menu.h: khai báo hàm menu()

Phần Source:

- Champion.cpp
- Game.cpp
- Log.cpp
- Menu.cpp: triển khai menu()
- DoAn.cpp: hàm main()

Phần Header:

- Champion.h:

Lưu private thông tin 1 tướng

```
private: // lưu những thuộc tính của 1 con tướng
    string ten;
    string he;
    float mau;
    float cong;
    float thu;
    float tocDanh;
```

Các constructor

```
TUONG();
TUONG(const TUONG&);
TUONG(string, string, float, float, float, float);
```

Các hàm get, set

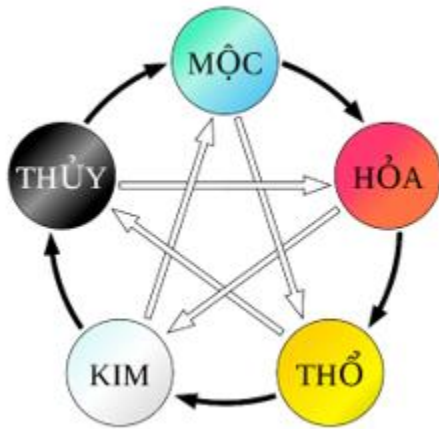
```
void setTen(string);
void setHe(string);
void setMau(float);
void setCong(float);
void setThu(float);
void setTocDanh(float);
string getTen();
string getHe();
float getMau();
float getCong();
float getThu();
float getTocDanh();
```

Lấy 1 mảng string trả về các hệ tương sinh tương khắc:

0. hệ khắc / 1. hệ sinh / 2. hệ bị khắc / 3. hệ được sinh

```
string* getHeTSTK();
```

Theo quy luật:



Ví dụ: Hệ hỏa

- 0.Hệ khắc: thủy
- 1.Hệ sinh: Mộc
- 2.Hệ bị khắc: Kim
- 3.Hệ được sinh: Thổ

Các hàm khác

```
float thoiGianGietDich(TUONG dich); // thời gian để giết chết 1 con tướng khác
bool chet(); // kiểm tra xem đã chết chưa
void print(ostream& outDev); // hàm in
friend istream& operator >> (istream& inDev, TUONG& x);
friend ostream& operator << (ostream& outDev, TUONG& x);
void nhapThongTin(); // nhập thông tin 1 con tướng
```

- Game.h

Lưu private thông tin game

```
private:
    vector<TUONG> listTuong; // danh sách tướng đọc từ file ListHero.txt
    vector<TUONG> player_1; // danh sách tướng 2 người chơi
    vector<TUONG> player_2;
    string username_1; // tên 2 người chơi
    string username_2;
    string moiTruong; // môi trường
    float thoiGian; // thời gian của hiệp đấu
    int cheDo; // 1. là chế độ 1 người chơi, 2. là chế độ 2 người chơi
```

Constructor:

```
GAME();
```

Các hàm get, set

```
void setMoiTruong(string);
void setThoiGian(float);
void setUsername1(string _us1);
void setUsername2(string _us2);
void setCheDo(int);
string getMoiTruong();
float getThoiGian();
string getUsername1();
string getUsername2();
int getCheDo();
```

Các hàm xử lý game:

```
void docFile();           // đọc file ListHero.txt vào bộ nhớ game
string randomMoiTruong(); // tạo môi trường
vector<TUONG> chonTuongBinhThuong(); // người chơi chọn
vector<TUONG> chonTuongNgauNhiem(); // máy chọn
void chonTuong();         // chọn tướng trong từng chế độ chơi
void xulyMoiTruong(vector<TUONG>&); // ảnh hưởng của môi trường
void xulyCungDoi(vector<TUONG>&); // ảnh hưởng của các thành viên cùng team
void xulyDoiKhang(vector<TUONG>&, vector<TUONG>&); // ảnh hưởng của địch
void xulyChung();         // xử lý chung cho 1 game đấu
void hienThiTranDau();
void danhNchau(TUONG&, TUONG&);
void danhNchau();        // cho các tướng đánh nhau
int xulyThangThua();      // 1. nếu player1 win, 0 nếu hòa, -1 nếu player1 thua
string ketQua();         // toString kết quả trận đấu
void print();            // in ra kết quả
void luuLog();
void themTuong();        // thêm vào file ListHero.txt
void xoaTuong();         // xóa khỏi listhero
```

- Log.h

Lưu private 1 log

```
private:
    int h; // giờ
    int m; // phút
    int s; // giây
    int dd; // ngày
    int mm; // tháng
    int yy; // năm
    string username_1;
    string username_2;
    vector<TUONG>player_1;
    vector<TUONG>player_2;
    string moiTruong;
    string ketQua;
```

Các hàm get, set

```
void setH(int);
void setM(int);
void setS(int);
void setDD(int);
void setMM(int);
void setYY(int);
void setUsername_1(string);
void setUsername_2(string);
void setListTuongPlayer_1(const vector<TUONG>);
void setListTuongPlayer_2(const vector<TUONG>);
void setMoiTruong(string);
void setKetQua(string);
int getH();
int getM();
int getS();
int getDD();
int getMM();
int getYY();
string getUsername_1();
string getUsername_2();
vector<TUONG> getListTuongPlayer_1();
vector<TUONG> getListTuongPlayer_2();
string getMoiTruong();
string getKetQua();
```


Các hàm xử lý log

```
friend ostream& operator << (ostream& outDev, LOG&);  
friend istream& operator >> (istream& inDev, LOG&);  
void docLog();           // đọc file log ra listLog  
int xemLog();            // trả về index thời gian trận đấu  
void xemLogChiTiet();    // xem chi tiết trận đấu
```

- Menu.h:

Khai báo menu

```
void menu();
```

Phần Source:

- Champion.cpp:

Hàm thời gian giết địch:

Nếu công bé hơn thủ của tướng địch thì trả về -1 (đánh mãi không mất máu)

Còn không đánh đến khi địch hết máu => return time

```

float TUONG::thoiGianGietDich(TUONG dich)
{
    float time = 0;
    float mauDich = dich.mau;
    if (cong < dich.thu) return -1; // không đánh vào máu nổi
    while (mauDich > 0)
    {
        mauDich -= (cong - dich.thu);
        time += tocDanh;
    }
    return time;
}

```

- Game.cpp

Hàm chọn tướng ngẫu nhiên:

Mỗi lần sinh r mới thì lấy đồng dư với số lượng list tướng => chọn tướng

```

vector<TUONG> GAME::chonTuongNgauNhien()
{
    vector<TUONG> x;
    srand((int)time(0));
    int r;
    for (int i = 0; i < nChamp; i++)
    {
        r = rand() % listTuong.size(); // sinh ngẫu nhiên
        for (int j = 0; j < listTuong.size(); ++j)
        {
            if (r == j)
            {
                x.push_back(listTuong.at(j));
            }
        }
    }
    return x;
}

```

Hàm setup môi trường

Nếu tương khắc với môi trường thì máu giảm 3%

Nếu tương sinh với môi trường thì tăng 5% máu và giảm 10% thời gian ra đòn

```
void GAME::xuLyMoiTruong(vector<TUONG>& x) // xử lý môi trường ảnh hưởng đến 1 danh sách tướng
{
    for (int i = 0; i < nChamp; ++i)
    {
        if (x.at(i).getHeTSTK()[0] == moiTruong || x.at(i).getHeTSTK()[2] == moiTruong)
        { // nếu hệ đối tượng khắc với môi trường
            x.at(i).setMau(x.at(i).getMau() * 97 / 100); // giảm chỉ số máu
        }
        if (x.at(i).getHeTSTK()[1] == moiTruong || x.at(i).getHeTSTK()[3] == moiTruong)
        { // nếu hệ đối tượng sinh với môi trường
            x.at(i).setMau(x.at(i).getMau() * 105 / 100); // tăng máu
            x.at(i).setTocDanh(x.at(i).getTocDanh() * 90 / 100); // giảm thời gian ra đòn
        }
    }
}
```

Hàm xử lý trong cùng 1 đội:

Vòng for để chọn chỉnh hợp

Nếu hệ khắc của i là j thì i sẽ bị tăng thời gian ra đòn đi 10%

Nếu hệ buff của con i là j thì i sẽ được j tăng 15% máu

```
void GAME::xuLyCungDoi(vector<TUONG>& x)
{
    for (int i = 0; i < nChamp; i++)
    {
        for (int j = 0; j < nChamp; j++)
        {
            if (i != j) // lấy từng đôi một có thứ tự
            {
                if (x.at(i).getHeTSTK()[0] == x.at(j).getHe()) // nếu hệ khắc của con i là con j
                {
                    x.at(i).setTocDanh(x.at(i).getTocDanh() * 110 / 100); // con i tự tăng thời gian ra đòn
                }
                if (x.at(i).getHeTSTK()[1] == x.at(j).getHe()) // nếu hệ buff của con i là con j
                {
                    x.at(i).setMau(x.at(i).getMau() * 115 / 100); // được tăng máu
                }
            }
        }
    }
}
```

Hàm xử lý đối kháng:

Nếu tướng y khắc tướng x thì thủ của x sẽ giảm 50%, công của y tăng 10%

```
void GAME::xuLyDoiKhang(vector<TUONG>& x, vector<TUONG>& y) // xử lý đối kháng ảnh hưởng đến
{
    for (int i = 0; i < nChamp; ++i)
    {
        if (x.at(i).getHeTSTK()[0] == y.at(i).getHe()) // nếu con tướng y khắc con tướng x
        {
            x.at(i).setThu(x.at(i).getThu() / 2); // x sẽ bị giảm thủ
            y.at(i).setCong(y.at(i).getCong() * 110 / 100); // y sẽ tăng công
        }
    }
}
```

Hàm xử lý thời gian:

```
if (getCheDo() == 1) // xử lý thời gian
{
    thoiGian = 1;
}
else
{
    cout << "\nINPUT MATCH TIME (1/2/5 MINUTES): ";
    cin >> thoiGian;
}
```

Bộ đếm cho 1 trận đấu:

1 là 1s

```
clock_t endwait;
endwait = clock() + 1 * CLOCKS_PER_SEC;
while (clock() < endwait) {}
```

- **Log.cpp**

Hàm đọc log

```
void LOG::docLog()
{
    listlog.clear(); // xóa listLog cũ
    fstream fileLog("Log.txt", ios::in); // đọc log
    if (!fileLog)
    {
        cout << "\nCANNOT OPEN FILE!!!";
        exit(0);
    }
    string temp;
    getline(fileLog, temp, '\n');
    if (temp != "") // xem định dạng file có dòng đầu hay không
    {
        fileLog.seekg(0);
    }
    while (!fileLog.eof())
    {
        LOG x;
        fileLog >> x;
        listlog.push_back(x);
    }
    fileLog.close();
}
```

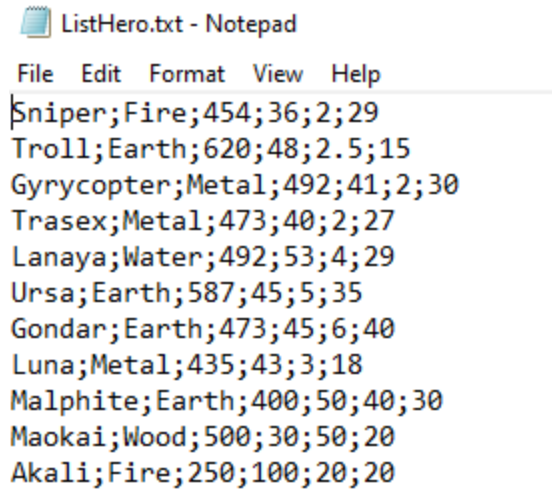
- **Menu.cpp**

Hiển thị menu

```
cout << "\n"
cout << "\n"
cout << "\n"
cout << "\n"
cout << "\n"
cout << "\n"
cout << "\n"
```

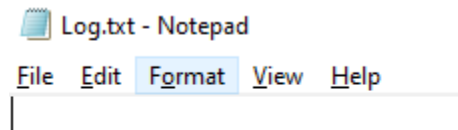
III. Demo game đấu tướng:

File list hero



```
ListHero.txt - Notepad
File Edit Format View Help
Sniper;Fire;454;36;2;29
Troll;Earth;620;48;2.5;15
Gyrycopter;Metal;492;41;2;30
Trasex;Metal;473;40;2;27
Lanaya;Water;492;53;4;29
Ursa;Earth;587;45;5;35
Gondar;Earth;473;45;6;40
Luna;Metal;435;43;3;18
Malphite;Earth;400;50;40;30
Maokai;Wood;500;30;50;20
Akali;Fire;250;100;20;20
```

File log



```
Log.txt - Notepad
File Edit Format View Help
|
```

Giao diện game



Giao diện đấu tướng

```
SUB MENU
1. (PLAYER vs PC)
2. (PLAYER vs PLAYER)
3. BACK TO MAIN MENU
INPUT OPTION:
```

Nếu chọn Player vs PC

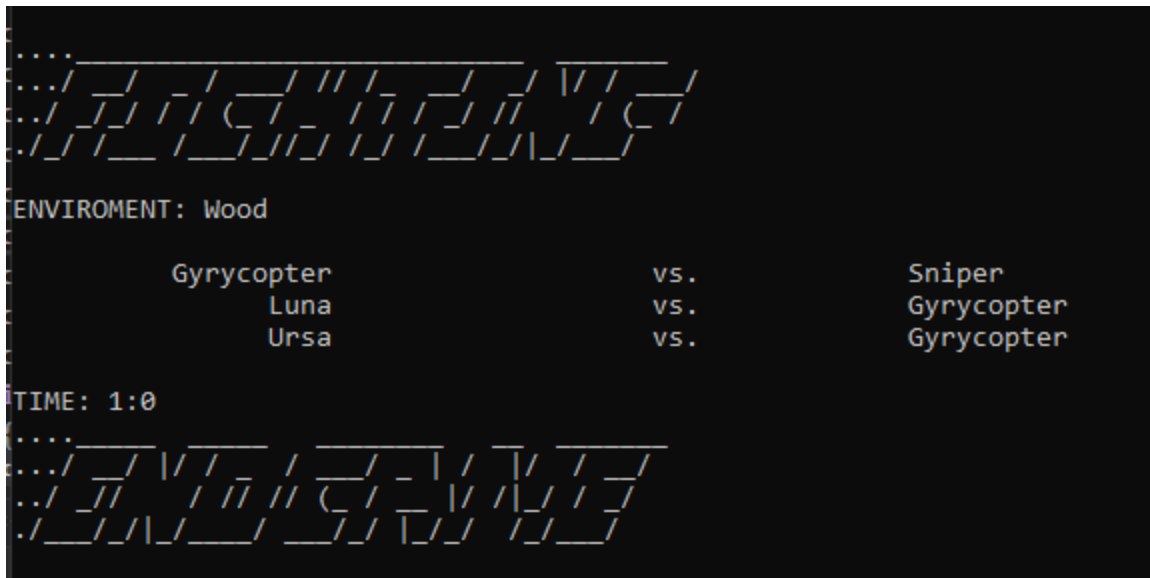
Chọn tướng

```
CHAMPION
1. Sniper;Fire;454;36;2;29
2. Troll;Earth;620;48;2.5;15
3. Gyrycopter;Metal;492;41;2;30
4. Trasex;Metal;473;40;2;27
5. Lanaya;Water;492;53;4;29
6. Ursa;Earth;587;45;5;35
7. Gondar;Earth;473;45;6;40
8. Luna;Metal;435;43;3;18
9. Malphite;Earth;400;50;40;30
10. Maokai;Wood;500;30;50;20
11. Akali;Fire;250;100;20;20

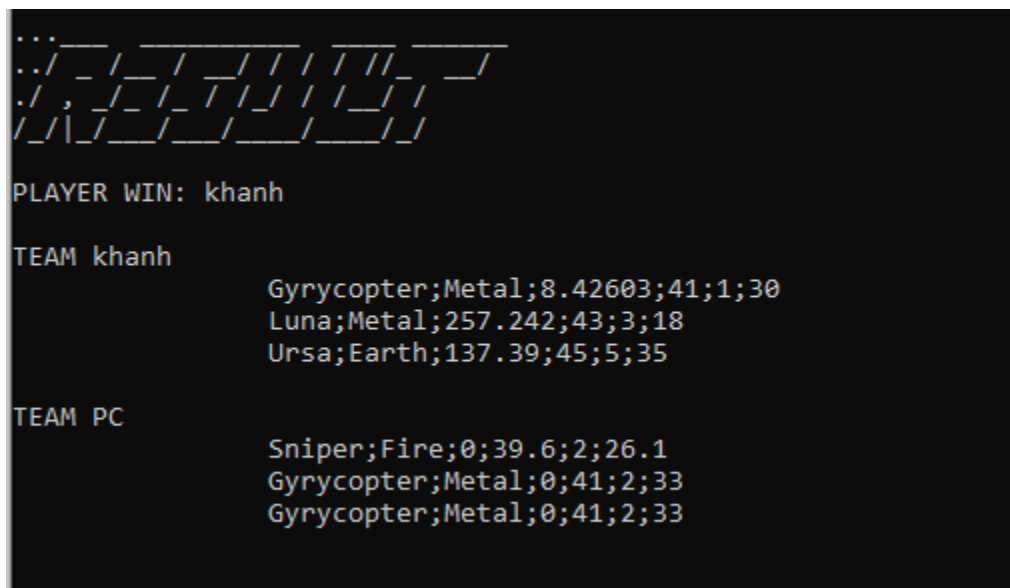
PLAYER PICK CHAMPION 1 (INDEX): 3
PLAYER PICK CHAMPION 2 (INDEX): 8
PLAYER PICK CHAMPION 3 (INDEX): 6
```

Giao diện chiến đấu

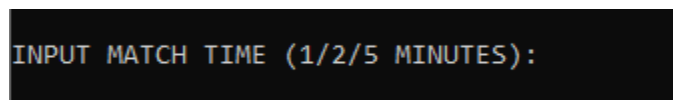
Time 1 phút



Kết quả



Nếu chọn player vs player sẽ có thêm chọn thời gian



Menu xem log

```
SUB MENU
      1. EXIT
      2. LOG VIEW
INPUT OPTION:
```

List Log

```
HISTORY MATCH
      0. EXIT
      1. 13:13:22 14/6/2021
      2. 13:13:39 14/6/2021
      3. 13:14:6 14/6/2021
      4. 13:14:29 14/6/2021
      5. 13:17:10 14/6/2021
      6. 13:17:41 14/6/2021
      7. 13:18:15 14/6/2021
      8. 13:45:10 14/6/2021
      9. 13:46:11 14/6/2021
     10. 13:46:33 14/6/2021
     11. 22:0:2 18/6/2021
     12. 22:4:6 18/6/2021
INPUT TIME (INDEX):
```

Nhập time để xem chi tiết log

```
MATCH DETAIL:
PLAYER WIN: cr7

TEAM abc

Sniper;Fire;0;39.6;2;29
Troll;Earth;466.61;48;2.5;15
Gyrycopter;Metal;0;41;2;29.7

TEAM cr7

Gyrycopter;Metal;92;41;1;27
Trasex;Metal;0;40;2;24.3
Lanaya;Water;243.67;53;4;29
```

Menu input/delete champion

```
SUB MENU
```

1. EXIT
 2. INPUT HERO
 3. DELETE HERO
- INPUT OPTION:

Input champ

```
INPUT CHAMPION!!!  
INPUT NAME: Jarvan IV  
INPUT ORIGIN: Earth  
INPUT HEALTH: 500  
INPUT ATTACK DAMAGE: 50  
INPUT ARMOR: 5  
INPUT ATTACK SPEED: 30
```

Delete champ

```
DELETE CHAMPION!!!
```

```
CHAMPION
```

1. Sniper;Fire;454;36;2;29
2. Troll;Earth;620;48;2.5;15
3. Gyrycopter;Metal;492;41;2;30
4. Trasex;Metal;473;40;2;27
5. Lanaya;Water;492;53;4;29
6. Ursa;Earth;587;45;5;35
7. Gondar;Earth;473;45;6;40
8. Luna;Metal;435;43;3;18
9. Malphite;Earth;400;50;40;30
10. Maokai;Wood;500;30;50;20
11. Akali;Fire;250;100;20;20
12. Jarvan IV;Earth;500;50;5;30

```
INPUT INDEX:
```

HẾT