



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

# Báo Cáo Đồ Án

## VÕ LÂM TRUYỀN KÌ



**GV HƯỚNG DẪN:**

Bùi Văn Thành

**NGƯỜI THỰC HIỆN:**

Nguyễn Cao Thi

Đình Minh Tuấn

Nguyễn Hoàng Trung Hải

21522613

21522749

21520808

---

# Lời nói đầu

Kính chào thầy!

Lời đầu tiên, cho phép nhóm xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Bùi Văn Thành.

Nhờ những kiến thức giảng dạy của thầy trên lớp cùng với sự tìm tòi của các thành viên, chúng em cũng đã hoàn thành đề án **Võ Lâm Truyền Kỳ** theo đúng tiến độ.

Do kinh nghiệm còn ít ỏi và thời gian của nhóm hạn chế, nên đề án lần này chắc chắn không thể tránh khỏi những thiếu sót. Vì vậy chúng em rất mong nhận được sự góp ý, chỉ bảo từ thầy để những đề án tiếp theo của tụi em có thể hoàn thiện hơn.

Lời cuối cùng, chúng em xin kính chúc thầy nhiều sức khỏe và luôn thành công trên con đường giảng dạy của mình.

Chúng em xin chân thành cảm ơn!

Nhóm thực hiện

---

# Mục lục

Lời nói đầu	1
Mục lục	2
Giới thiệu đề bài	3
Yêu cầu	4
Giải quyết đề bài	4
Ý tưởng	5
Chức năng của chương trình	6
Sơ đồ lớp	7
Cài đặt code	8
• Các class chính	8
• Thiết kế đồ họa, âm thanh và thao tác	12
Lời cảm ơn	14

# Giới thiệu đề bài

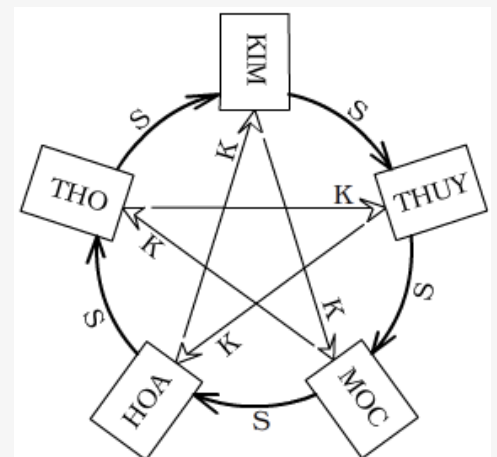
**1/** Xây dựng chương trình mô phỏng game **Võ lâm truyền kì** - game theo phong cách nhập vai kiếm hiệp. Người chơi có thể tương tác với nhau để giải trí hoặc tiêu diệt quái vật để phát triển nhân vật của mình.

**Nhân vật:** Được xem như đại diện cho một người chơi. Mỗi nhân vật thuộc về một môn phái nào đó trong tổng số 10 môn phái của game. Một nhân vật có một giá trị thể hiện cấp độ và mức sát thương. Trong đó Sát thương = Cấp độ x 5

**Quái vật:** Để gia tăng cấp độ nhân vật của mình, mỗi người chơi sẽ thông qua việc tiêu diệt các quái vật. Có hai loại quái vật : thông thường và đầu lĩnh. Các quái vật cũng sẽ có khả năng tấn công lại người chơi.

- Quái vật thông thường: Sát thương = Cấp độ x 3.
- Quái vật đầu lĩnh: Sát thương = Cấp độ x 7

**2/** Game có hệ thống ngũ hành tương sinh tương khắc, mỗi một môn phái và quái vật sẽ thuộc về một “hành” nhất định và tương tác giữa các người chơi với nhau, giữa người chơi với quái vật đều dựa trên các quy tắc về ngũ hành.



Có tất cả 5 hành, trong đó mỗi hành có 2 môn phái:

## Hệ Kim:

- Thiếu Lâm
- Thiên Vương bang

## Hệ Mộc:

- Ngũ Độc giáo
- Đường Môn

## Hệ Thủy:

- Nga My
- Thúy Yên môn

## Hệ Hỏa:

- Cái Bang
- Thiên Nhẫn giáo

## Hệ Thổ:

- Côn Lôn
- Võ Đang

**Quy tắc tương sinh:(S)** Sát thương gây ra cho NV hoặc QV + 10%

**Quy tắc tương khắc:(K)** Sát thương gây ra cho NV hoặc QV ± 20%

---

# Yêu cầu

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng và xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo và quản lý một danh sách các người chơi và quái vật.
2. Cho biết phần tử có mức sát thương cao nhất trong danh sách.
3. Cho hai phần tử A và B, so sánh giá trị sát thương tác động A lên B và ngược lại.

# Giải quyết đề bài

Nhóm đã suy nghĩ: Nếu cứ nhập vào nhân vật, quái vật thì game sẽ chỉ đơn giản là 1 lệnh tính toán sự tương tác giữa các nhân vật và quái vật.

Thay vào đó, chúng em sẽ mặc định là lúc bắt đầu, người chơi sẽ ở cấp độ 1, và sẽ tăng cấp dần sau mỗi lần hạ con quái. Quái vật sẽ là sự xuất hiện ngẫu nhiên với cấp độ và loại quái ngẫu nhiên (Tất nhiên bọn em sẽ kiểm soát tỷ lệ một cách hợp lý).

Game sẽ gồm 5 round. Nếu người chơi chiến thắng 5 round liên tiếp thì sẽ thắng. Thay vào đó sẽ thất bại nếu thua 1 trong các round đó.

---

# Ý tưởng

## 1. Sơ lược:

- Sử dụng kế thừa, đa hình và những kiến thức về lập trình hướng đối tượng khác để tạo sơ đồ chi tiết các lớp đối tượng. Từ đó xây dựng source code chính thức.
- Ngôn ngữ lập trình: C++
- Phần mềm sử dụng: Visual Studio 2019
- Source code bao gồm 4 class chính: Nhân vật (class NhanVat), quái vật (class QuaiVat), xử lí (class XuLi), tương khắc (class TuongKhac).
- Code logic chương trình: Tuấn, Hải
- Code đồ họa chương trình: Thi
- Test code và chỉnh sửa: Cả nhóm.

## 2. Thiết kế đồ họa và thao tác cho chương trình:

- Tham khảo thư viện đồ họa và chỉnh sửa theo nhu cầu của chương trình.
- Xây dựng class BOX để vẽ các khung chọn nhân vật.
- Sử dụng bắt sự kiện bàn phím để chọn nhân vật
- Sử dụng thư viện âm thanh và đa luồng để tạo tiếng cho chương trình.

## 3. Hoàn thành báo cáo:

- Sử dụng phần mềm Microsoft PowerPoint để thiết kế slide báo cáo
- Sử dụng canva để viết báo cáo và xuất ra file PDF.
- Người thực hiện: cả nhóm
- Đóng góp ý kiến chỉnh sửa: cả nhóm
- Thực hiện kiểm tra, chỉnh sửa các phần và thuyết trình thử.

---

# Giới thiệu chức năng của game

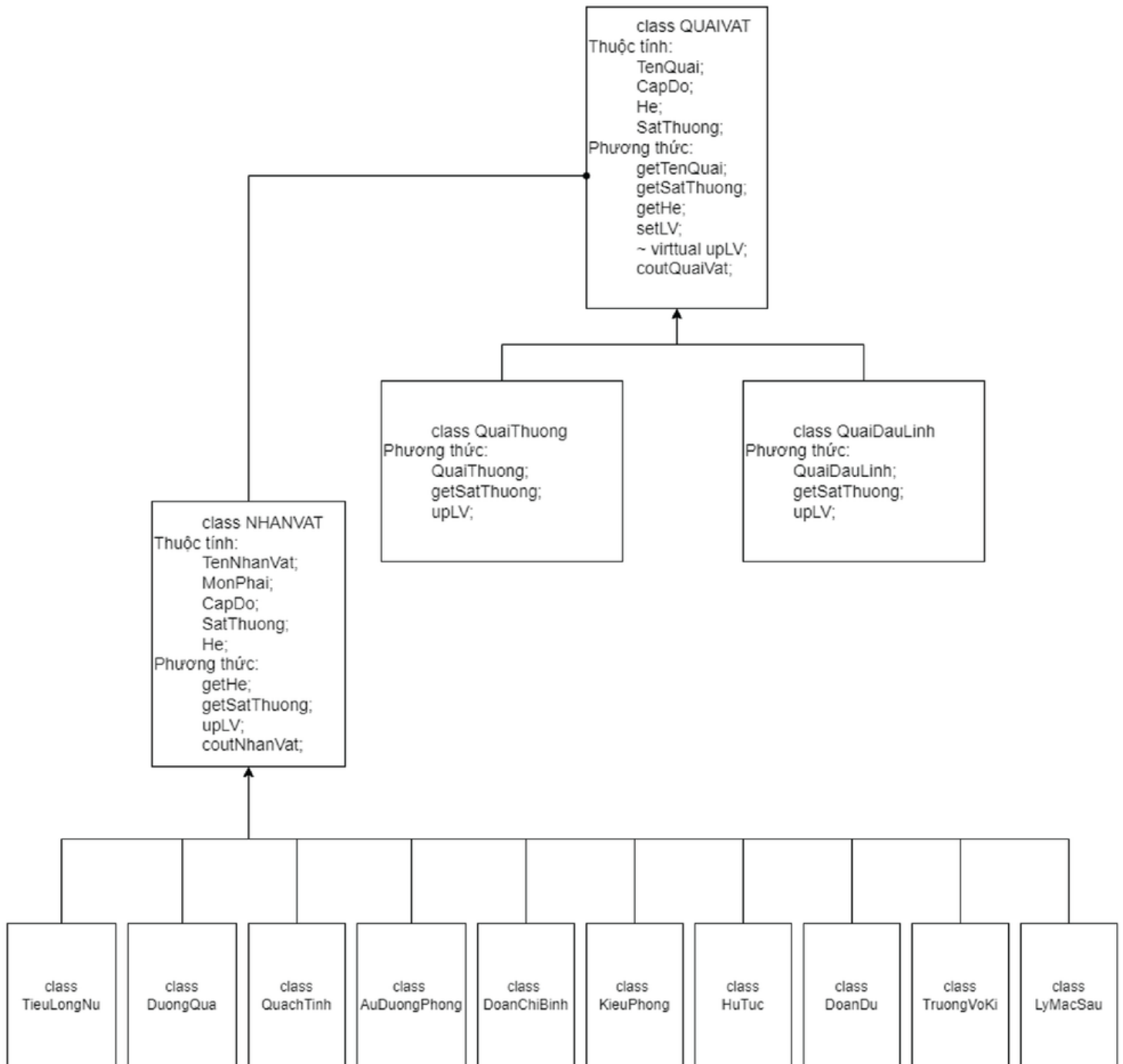
## 1. Chế độ chơi

Game này gần như là 1 game gacha khi em sẽ setup cho Quái vật tự xuất hiện và Nhân Vật sẽ lên cấp 1 cách ngẫu nhiên sau khi tiêu diệt quái vật.

## 2. Cách chơi

1. Người chơi sẽ đến với giao diện của game.
2. Nhập tên người chơi. Và chọn nhân vật game mà mình mong muốn.
3. Quái vật sẽ xuất hiện ngẫu nhiên.
4. Người chơi sẽ dành chiến thắng sau 5 vòng.
5. Và sẽ thất bại nếu 1 trong 5 vòng đó bị quái vật đánh chết.

# Sơ đồ lớp





# Cài đặt code

Xây dựng 2 class chính:  
class NhanVat và class QuaiVat

## Class NhanVat

```
class NhanVat {
protected:
    NhanVat() { CapDo = 1; }
    ~NhanVat() {};
    string TenNhanVat;
    string MonPhai;
    int CapDo = 1;
    int SatThuong;
    string He;
public:
    string getHe();
    int getSatThuong(); // Sat t
    friend class XuLi;
    void upLV(string);
    void coutNhanVat();
};
```

- Class **NhanVat** gồm
  - + 4 thuộc tính: Tên nhân vật, môn phái, cấp độ, sát thương, hệ
  - + 4 phương thức chính: Lấy hệ, Lấy sát thương (sát thương cơ bản), Tăng cấp độ, Xuất thông tin nhân vật.
- Class **NhanVat** có phương thức khởi tạo, khi khởi tạo nhân vật sẽ có cấp độ là 1
- Class **NhanVat** có class bạn là class **XuLi** (class xử lý sát thương giữa nhân vật và quái).

- Class **NhanVat** sẽ có 10 lớp con đại diện cho 10 nhân vật:

```
class TieuLongNu :public NhanVat {
public:
    TieuLongNu();
};
class DuongQua :public NhanVat { //Du
public:
    DuongQua();
};
class QuachTinh :public NhanVat { //Q
public:
    QuachTinh();
};
class AuDuongPhong :public NhanVat {
public:
    AuDuongPhong();
};
class DoanChi Binh :public NhanVat {
public:
    DoanChi Binh();
};
```

- Hệ: Kim
- Môn phái: Thiên Vương Bang

- Hệ: Kim
- Môn phái: Thiếu

- Hệ: Mộc
- Môn phái: Ngũ Độc Giáo

- Hệ: Mộc
- Môn phái: Đường Môn

- Hệ: Thủy
- Môn phái: Nga My

```
class KieuPhong :public NhanVat {
public:
    KieuPhong();
};
class HuTuc :public NhanVat { //Hu
public:
    HuTuc();
};
class DoanDu :public NhanVat { //Đo
public:
    DoanDu();
};
class TruongVoKi :public NhanVat {
public:
    TruongVoKi();
};
class LyMacSau :public NhanVat { //
public:
    LyMacSau();
};
```

- Hệ: Thủy
- Môn phái: Thúy Yên

- Hệ: Hỏa
- Môn phái: Cái Bang

- Hệ: Hỏa
- Môn phái: Thiên Nhẫn Giáo

- Hệ: Thổ
- Môn phái: Côn Lôn

- Hệ: Thổ
- Môn phái: Võ Đang

# Cài đặt code

## Class QuaiVat

Ở class **QuaiVat** em muốn tạo ra mỗi lượt 1 con quái vật với Loại quái, Cấp độ, Hệ ngẫu nhiên và có khả năng tăng Cấp sau mỗi round.

- 4 thuộc tính chính :

**TenQuai** : Quái Thường hoặc Quái đầu lĩnh

**CapDo** : Cấp độ của quái

**He** : Hệ của quái ;

**SatThuong**: SatThuong = CapDo x3 với Quái thường  
SatThuong = CapDo x7 với Quái Đầu Lĩnh

- 4 phương thức cơ bản:

**setLV**: Đưa vào giá trị cấp độ ban đầu ;

**getLV, getTenQuai, getHe**: Lấy các thuộc tính

**virtual getSatThuong**: lấy sát thương 2 class  
Quái Thường và Quái đầu lĩnh.

**upLV** : Tăng cấp độ của Quái Vật

*Ngoài ra Hệ và Tên Quái sẽ được khởi tạo random khi QuaiVat được tạo. Và cấp độ ban đầu của QV = 1;*

```
class QuaiVat {
protected:
    string TenQuai;
    int CapDo;
    string He;
    int SatThuong;
public:
    QuaiVat();
    ~QuaiVat();
    friend class XuLi;
    void setLV(int a);
    int getLV();
    string getTenQuai();
    string getHe();
    virtual int getSatThuong() = 0;
    virtual void upLV() = 0;
    void coutQuaiVat();
};
```

Thứ tự thực thi các hàm:

- Khởi tạo quái vật: Loại Quái , Hệ ngẫu nhiên bằng hàm **randHe()** ; **randLoaiQuai**
- Cấp độ ban đầu của quái vật là 1 ;
- Lớp **Xuli** sẽ gọi các phương thức get để lấy các chỉ số thực hiện chiến đấu với nhân vật.
- Và khi con quái vật đó bị đánh bại. Cấp độ của nó sẽ được giữ lại để chuẩn bị tạo ra con quái vật mới.
- Khi 1 con quái vật bị đánh bại ta sẽ tạo ra 1 con quái vật mới với Loại quái và Hệ random. Quái vật mới sẽ nhận cấp độ của con quái vật đã chết là cấp độ ban đầu. QV sẽ được tăng cấp độ để mạnh hơn con quái vật đã chết bằng hàm **upLV** - Tăng cấp độ Quái Vật. Số cấp độ tăng tùy thuộc vào Loại Quái cũng như May mắn của người chơi. Quái Thường (tăng 1-2 cấp) , Quái đầu lĩnh (Mặc định tăng 1).
- Cứ lặp lại như thế. Và nếu con quái vật thứ 5 bị đánh bại Người chơi sẽ dành chiến thắng

# Cài đặt code

Class Quaithuong : public QuaiVat

```
class Quaithuong :public QuaiVat
{
public:
    Quaithuong();
    int getSatThuong();
    //void coutThongTin() ;
    void upLV();
};
```

Class QuaiDauLinh: public Quaivat

```
class QuaiDauLinh :public QuaiVat
{
public:
    QuaiDauLinh();
    int getSatThuong();
    //void coutThongTin() ;
    void upLV();
};
```

Ý tưởng của em khi tạo ra 2 class con này là sử dụng upcasting từ lớp QuaiVat đến 2 lớp con là **QuaiThuong** và **QuaiDauLinh**.

=> Vậy nên 2 class con này sẽ chỉ để thay đổi các thuộc tính phù hợp với với LoaiQuai mong muốn.

## Phân tích code:

- Hàm khởi tạo sẽ khởi tạo tên tương ứng cho Loại Quái vật;
- Hàm **getSatThuong** ở lớp cha (class **QuaiVat**) là hàm ảo sẽ được định nghĩa lại ở đây.
- Tương ứng với class **QuaiThuong** thì Sát thương = Cấp độ x3 còn class **QuaiDauLinh** thì Sát thương = Cấp độ x7 và nó sẽ cập nhật trực tiếp vào thuộc tính **SatThuong**.
- Ngoài ra hàm **upLV** ở lớp cha cũng được định nghĩa ảo.

Ở đây chúng ta sẽ thực hiện tăng Cấp độ cho Quái:

Quái Thường : 20% tăng 2 cấp và 80% tăng 1 cấp

Quái Đầu Lĩnh : Mặc định tăng 1 cấp

# Class xử lí logic

- Class xử lí logic của chương trình gồm 2 class: class XuLi và class TuongKhac.

## Class XuLi

```
class XuLi {
    NhanVat* NV;
    QuaiVat* QV;
    int ak;
public:
    XuLi() {
        ak = 1;
    }
    ~XuLi(){
        delete[]QV;
    };
    void intro();
    bool Win();
    void KhoiTaoNhanVat(string TenNhanVat);
    void resetQuai();
    void KhoiTaoQuaiVat();
    void Start();
    int getak();
};
```

- Class **XuLi** dùng để xử lí sát thương tác dụng giữa Nhân vật và Quái vật, khởi tạo quái mới khi quái bị tiêu diệt và tăng cấp cho nhân vật

## Class TuongKhac

```
class TuongKhac {
private:
    string check(string He1, string He2);
public:
    friend class XuLi;
    TuongKhac() {
    };
};
```

```
string TuongKhac::check(string He1, string He2) {
    string s = "Sinh";
    string k = "Khac";
    if (He1 == "Moc" && He2 == "Hoa") return s;
    if (He1 == "Hoa" && He2 == "Tho") return s;
    if (He1 == "Tho" && He2 == "Kim") return s;
    if (He1 == "Kim" && He2 == "Thuy") return s;
    if (He1 == "Thuy" && He2 == "Moc") return s;
    if (He1 == "Moc" && He2 == "Tho") return k;
    if (He1 == "Hoa" && He2 == "Kim") return k;
    if (He1 == "Tho" && He2 == "Thuy") return k;
    if (He1 == "Kim" && He2 == "Moc") return k;
    if (He1 == "Thuy" && He2 == "Hoa") return k;
    return "";
}
```

- Class **TuongKhac** dùng để xử lí tương khắc hệ giữa nhân vật và quái vật

# Thiết kế đồ họa, thao tác và âm thanh cho chương trình

## Thư viện đồ họa

Chúng em đã tham khảo thư viện đồ họa và chỉnh sửa theo ý của mình

```
#define ColorCode_Back      0
#define ColorCode_DarkBlue  1
#define ColorCode_DarkGreen 2
#define ColorCode_DarkCyan  3
#define ColorCode_DarkRed   4
#define ColorCode_DarkPink   5
#define ColorCode_DarkYellow 6
#define ColorCode_DarkWhite  7
#define ColorCode_Grey      8
#define ColorCode_Blue       9
#define ColorCode_Green     10
#define ColorCode_Cyan       11
#define ColorCode_Red        12
#define ColorCode_Pink       13
#define ColorCode_Yellow     14
#define ColorCode_White      15

extern int WidthConsole;
extern int HeightConsole;
```

```
void clrscr();
void gotoXY(int, int);
int whereX();
int whereY();
void textColor(WORD);
void backgroundColor(WORD);
void printTextColor(int, int, string, WORD);
void resizeConsole(int, int);
void hideCur();
void showCur();
void clearLine(int, int);
void fullScreen();
void removeScrollBar();
void getWHConsole();
```

**Thư viện đồ họa có các chức năng như nhau:**

- Xóa màn hình.
- Di chuyển tọa độ và lấy tọa độ
- Tô màu nền console và màu chữ
- Điều chỉnh kích cỡ màn hình console và lấy kích cỡ màn hình
- Xóa thanh cuộn và xóa con trỏ

---

## Class Box và sử dụng bắt sự kiện bàn phím

### Class Box

```
class Box{
private:
    int x, y, w, h, tColor;
    string text;
public:
    Box();
    ~Box();
    void Set(int, int);
    // x, y, w, h, tColor, text
    void Set(int, int, int, int, int, string);
    void VeBox();
    void VeBoxChon();
    void InChuoai();
    void XoaChon();
    int getX();
    int getY();
    void inNhanVat(int x, int y, string file);
};
```

**Chức năng:** Vẽ khung chọn nhân vật và xóa khung chọn bằng cách dùng di chuyển tọa độ và in thông tin của 10 nhân vật từ các file .txt như TieuLongNu.txt, DuongQua.txt, QuachTinh.txt,...

=> Sử dụng bắt sự kiện bàn phím để chọn các khung nhân vật đã vẽ

Chương trình còn có sử dụng thư viện âm thanh để thêm phần sinh động  
Ngoài ra chúng em còn sử dụng thêm một số hàm liên quan như: Random  
quái vật, vẽ hình ảnh dạng text và lưu lịch sử đấu, thầy có thể ở source  
code chúng em

---

# Lời cảm ơn

Trên đây là toàn bộ báo cáo của nhóm, cảm ơn thầy đã xem qua. Mặc dù còn nhiều thiếu sót, chúng em mong thầy chỉ bảo và góp ý để chương trình của nhóm được hoàn thiện hơn. Em xin chân thành cảm ơn!

Chúc thầy một ngày tốt lành!