TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

PHÂN HIỆU TẠI TP. HỒ CHÍ MINH

BỘ MÔN CÔNG NGHỆ THÔNG TIN

**Logo

Description automatically generated**

BÁO CÁO LẬP TRÌNH HƯỚNG

ĐỐI TƯỢNG

ĐỀ TÀI: GAME ROGUE LITE

**Giảng viên hướng dẫn:** TRẦN THỊ DUNG

**Sinh viên thực hiện:** HỒ VĨNH TÍN

**Lớp:** CQ.62.CNTT

**Khóa:** K62

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

PHÂN HIỆU TẠI TP.HỒ CHÍ MINH

BỘ MÔN CÔNG NGHỆ THÔNG TIN

**Logo

Description automatically generated**

BÁO CÁO LẬP TRÌNH HƯỚNG

ĐỐI TƯỢNG

ĐỀ TÀI: GAME ROGUE LITE

**Giảng viên hướng dẫn:** TRẦN THỊ DUNG

**Sinh viên thực hiện:** HỒ VĨNH TÍ**N**

**Lớp:** CQ.62.CNTT

**Khóa:** K62

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI             PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH | CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM  Độc lập – Tự do – Hạnh phúc |

BÀI TẬP LỚN MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-------\*\*\*-------

**Mã sinh viên:** 6251071104 **Họ tên SV:** Hồ Vĩnh Tín

**Khóa:** K62 **Lớp:** CQ.62.CNTT

**Mã sinh viên:** 6251071105 **Họ tên SV:** Phan Ngọc Như Tranh

**Khóa:** K62 **Lớp:** CQ.62.CNTT

**Mã sinh viên:** 6251071101 **Họ tên SV:** Nguyễn Ngọc Tiểu Thư

**Khóa:** K62 **Lớp:** CQ.62.CNTT

**Mã sinh viên:** 6251071100 **Họ tên SV:** Lê Công Thuận

**Khóa:** K62 **Lớp:** CQ.62.CNTT

1. Tên đề tài

**GAME ROGUE LITE**

1. Mục đích, yêu cầu

* Mục đích:
* Tìm hiểu về một số game cơ bản trong lập trình.
* Tăng cường hiểu biết, ứng dụng linh hoạt ngôn ngữ lập trình C++ vào game cơ bản.
* Hiểu rõ hơn vê 4 tính chất quan trọng trong lập trình hướng đối tượng.
* Yêu cầu:
* Nắm rõ về game nhóm làm, 4 tính chất cơ bản trong lập trình hướng đối tượng.
* **Tinh thần teamwork, đoàn kết, công bằng.**
* **Có kiến thức cơ bản và mở rộng chuyên sâu về ngôn ngữ chuyên ngành**

1. Nội dung và phạm vi đề tài

* Nội dung báo cáo:
* Tổng quan về game Rogue lite
* Ứng dụng ngôn ngữ lập trình C++ vào game
* Áp dụng nền tảng Github hỗ trợ việc quản lý làm việc nhóm, đóng góp vào bài tập chung.
* Phạm vi đề tài:
* **Nghiên cứu, xây dựng game Rogue lite và ứng dụng trong phạm vi lớp học.**

1. Công nghệ, công cụ và ngôn ngữ lập trình

- Công cụ sử dụng: Visual Studio, Git, GitHub, …

- Ngôn ngữ lập trình: C++

- Phần mềm Tiled

- Thư viện SDL

1. Các kết quả chính dự kiến và đạt được ứng dụng

**- Quyển báo cáo đồ án tốt nghiệp.**

**- Xây dựng thành công game Rogue lite đơn giản.**

**- Ứng dụng vào thực tế giúp các sinh viên cải thiện kỹ thuật lập trình và hiểu rõ về lập trình hướng đối tượng(OOP).**

1. Giảng viên và bộ môn hướng dẫn

**Họ tên: ThS. TRẦN THỊ DUNG**

**Đơn vị công tác:** Bộ môn Công nghệ Thông tin – Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh.

**Điện thoại: 0388389579 Email:** [ttdung@st.utc2.edu.vn](mailto:ttdung@st.utc2.edu.vn)

|  |  |
| --- | --- |
| Ngày 1 tháng 11 năm 2022  Trưởng bộ BM Công nghệ Thông tin  ThS. Trần Phong Nhã | Đã giao nhiệm vụ TKTN  Giáo viên hướng dẫn  ThS. Trần Thị Dung |

|  |  |
| --- | --- |
| Đã nhận nhiệm vụ TKTN  Sinh viên: Hồ Vĩnh Tín  Điện thoại: 0833402853  Sinh viên: Phan Ngọc Như Tranh  Điện thoại: 0915386629  Sinh viên: Nguyễn Ngọc Tiểu Thư  Điện thoại: 0935367698  Sinh viên: Lê Công Thuận  Điện thoại: 0773931811 | Ký tên:  Email: [6251071104@st.utc2.edu.vn](mailto:6251071104@st.utc2.edu.vn)  Ký tên:  Email: [6251071105@st.utc2.edu.vn](mailto:6251071105@st.utc2.edu.vn)  Ký tên:  Email: [6251071101@st.utc2.edu.vn](mailto:6251071101@st.utc2.edu.vn)  Ký tên:  Email: [6251071100@st.utc2.edu.vn](mailto:6251071100@st.utc2.edu.vn) |

LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi tới Quý Thầy Cô Bộ môn Công nghệ Thông tin Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh lời chúc sức khỏe và lòng biết ơn sâu sắc.

Em xin chân thành cảm ơn quý thầy cô đã giúp đỡ tạo điều kiện để em hoàn thành đồ án với đề tài “ làm game”. Đặc biệt em xin cảm ơn cô Trần Thị Dung đã nhiệt tình giúp đỡ, hướng dẫn cho em kiến thức, định hướng và kỹ năng để có thể hoàn thành đồ án tốt nghiệp này.

Cảm ơn đến toàn thể các bạn thuộc lớp Công nghệ Thông tin K62 đã luôn giúp đỡ mình trong thời gian học tập cùng nhau. Chúc sức khỏe đến toàn thể giảng viên của trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh đặc biệt là các giảng viên thuộc Bộ Môn Công nghệ Thông tin, các bạn thuộc lớp Công nghệ Thông tin K62, các bạn bè khác của tôi.

Xin chân thành cảm ơn

***TP. Hồ Chí Minh, tháng 10 năm 2022***

|  |
| --- |
| Hồ Vĩnh Tín  Phan Ngọc Như Tranh  Nguyễn Ngọc Tiểu Thư  Lê Công Thuận |

**Sinh viên thực hiện**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

***TP. Hồ Chí Minh, ngày … tháng 10 năm 2022***

**Giáo viên hướng dẫn**

**ThS.Trần Thị Dung**

**MỤC LỤC**

[**LỜI CẢM ƠN iv**](#_Toc118297745)

[**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN v**](#_Toc118297746)

[**DANH MỤC HÌNH ẢNH viii**](#_Toc118297747)

[**CHƯƠNG 1: DỊCH SÁCH ix**](#_Toc118297748)

[**1.1 Chương 18: Nạp chồng ix**](#_Toc118297749)

[1.1.1 Giới thiệu ix](#_Toc118297750)

[1.1.3 Nạp chồng hàm toán tử ix](#_Toc118297751)

[1.1.3 Một loại số phức xii](#_Toc118297752)

[**1.2 Chương 19 xii**](#_Toc118297753)

[1.2.1. Giới thiệu xii](#_Toc118297754)

[1.2.2. Các toán tử đặc biệt xii](#_Toc118297755)

[1.2.3. Các hàm hỗ trợ Một tập hợp các hàm hữu ích xiii](#_Toc118297756)

[1.2.4. Hàm bạn một hàm được khai báo friend được cấp quyền truy cập vào việc triển khai một lớp giống như một hàm thành viên nhưng độc lập với lớp đó. xiv](#_Toc118297757)

[**CHƯƠNG 2: MỞ ĐẦU 1**](#_Toc118297758)

[**2.1. Lý do chọn đề tài 1**](#_Toc118297759)

[**2.2 Mục tiêu nghiên cứu 1**](#_Toc118297760)

[**2.3. Đối tượng nghiên cứu 1**](#_Toc118297761)

[**CHƯƠNG 3: TÌM HIỂU VỀ NGÔN NGỮ LẬP TRÌNH C++ 2**](#_Toc118297762)

[**3.1. Tổng quan về ngôn ngữ lập trình C++ 2**](#_Toc118297763)

[3.1.1. Khái niệm 2](#_Toc118297764)

[3.1.2. Đặc điểm 2](#_Toc118297765)

[**3.2. Các tính chất quan trọng 2**](#_Toc118297766)

[3.2.1. Tính đóng gói 2](#_Toc118297767)

[3.2.2. Tính kế thừa 2](#_Toc118297768)

[3.2.3. Tính đa hình 2](#_Toc118297769)

[3.2.4. Tính trừu tượng 3](#_Toc118297770)

[**CHƯƠNG 4: XÂY DỰNG GAME ROGUE LITE 4**](#_Toc118297771)

[**4.1. Tích hợp thư viện SDL2 vào dự án trên Visual Studio 4**](#_Toc118297772)

[**4.2. Thiết kế bản đồ 10**](#_Toc118297773)

[**4.3. Giao diện game Rogue Lite 11**](#_Toc118297774)

[**CHƯƠNG 5: KẾT QUẢ 14**](#_Toc118297775)

[**5.1. Kết quả 14**](#_Toc118297776)

[**5.2. Nhược điểm 14**](#_Toc118297777)

[**5.3. Công việc và mức độ hoàn thành của các thành viên 14**](#_Toc118297778)

[**TÀI LIỆU THAM KHẢO 15**](#_Toc118297779)

# **DANH MỤC HÌNH ẢNH**

[**Hình 4.1 Tạo dự án console app** 4](#_Toc118293135)

[**Hình 4.2 Chạy chương trình** 4](#_Toc118293136)

[**Hình 4.3 Copy thư mục SDL\_lib vào dự án** 5](#_Toc118293137)

[**Hình 4.4 Set up môi trường** 5](#_Toc118293138)

[**Hình 4.5 Set up môi trường** 6](#_Toc118293139)

[**Hình 4.6 Setup môi trường** 6](#_Toc118293140)

[**Hình 4.7 Set up môi trường** 7](#_Toc118293141)

[**Hình 4.8 Setup môi trường** 7](#_Toc118293142)

[**Hình 4.9 setup môi trường** 8](#_Toc118293143)

[**Hình 4.10 Copy file dll vào thư mục dự án** 8](#_Toc118293144)

[**Hình 4..11 Chương trình SDL cơ bản** 9](#_Toc118293145)

[**Hình 4.12 Chạy chương trình thành công** 9](#_Toc118293146)

[**Hình 4.13 sơ đồ các lớp kế thừa** 10](#_Toc118293147)

[**Hình 4.14 Sử dụng Tiled để tạo bản đồ bằng các tileset** 11](#_Toc118293148)

[**Hình 4.15** **Menu** 11](#_Toc118293149)

[**Hình 4.16** **Màng hình nhập tên người chơi** 12](#_Toc118293150)

[**Hình 4.17** **Màng hình hướng dẫn** 12](#_Toc118293151)

[**Hình 4.18** **Màng hình hiển thị score** 13](#_Toc118293152)

[**Hình 4.19** **Giao diện chơi game** 13](#_Toc118293153)

# **CHƯƠNG 1: DỊCH SÁCH**

## **1.1 Chương 18: Nạp chồng**

* + 1. **Giới thiệu**

Để thuận tiện cho việc trình bày và thảo luận liên quan đến các khái niệm được sử dụng thường xuyên, hầu hết các lĩnh vực kĩ thuật và phi kĩ thuật phát triển các kí hiệu viết tắt thông thường, ví dụ (x+y\*z) thay vì (nhân y với z rồi cộng kết quả cho x).

Giống như hầu hết các ngôn ngữ khác, C ++ hỗ trợ một tập hợp các toán tử cho các kiểu tích hợp của nó. Tuy nhiên, hầu hết các khái niệm mà các toán tử được sử dụng thông thường không phải là các kiểu tích hợp sẵn trong C ++, vì vậy chúng phải được biểu diễn dưới dạng các kiểu do người dùng định nghĩa. Ví dụ: nếu bạn cần số học phức tạp, đại số ma trận, tín hiệu logic hoặc chuỗi ký tự trong C ++, bạn sử dụng các class để biểu diễn các khái niệm này.

Việc xác định các toán tử cho các class như vậy đôi khi cho phép lập trình viên cung cấp một ký hiệu thông thường và thuận tiện hơn để thao tác các đối tượng hơn là có thể đạt được chỉ bằng cách sử dụng ký hiệu chức năng cơ bản.

class complex { // số phức đơn giản

double re, im;

public:

complex(double r, double i) :re{r}, im{i} { }

complex operator+(complex);

complex operator∗(complex);

};

Đây là một cách triển khai đơn giản của khái niệm số phức.

* + 1. **Nạp chồng hàm toán tử**

Các hàm xác định nghĩa cho các toán tử có thể được khai báo:

Tên của một hàm toán tử là toán tử từ khóa được theo sau bởi chính toán tử đó, ví dụ, toán tử <<. Một hàm toán tử được khai báo và có thể được gọi như bất kỳ hàm nào khác. Việc sử dụng toán tử chỉ là cách viết tắt cho một lệnh gọi rõ ràng của hàm toán tử. Ví dụ:

void f(complex a, complex b)

{

complex c = a + b; // rút gọn

complex d = a.operator+(b); // rõ ràng

}

Với định nghĩa trên đây về số phức, hai bộ khởi tạo như nhau.

* **Toán tử 2 ngôi và 1 ngôi**

Một toán tử 2 ngôi có thể được định nghĩa bởi một hàm thành viên không tĩnh mang một đối số hoặc một hàm không phải thành viên mang hai đối số.

class X {

public:

void operator+(int);

X(int);

};

void operator+(X,X);

void operator+(X,double);

void f(X a){

a+1; // a.operator+(1)

1+a; // ::operator+(X(1),a)

a+1.0; // ::operator+(a,1.0)

}

Toán tử một ngôi, dù là tiền tố hay hậu tố, nó có thể được xác định bởi một hàm thành viên không tĩnh không có đối số hoặc một hàm không phải là một đối số.

class X {

public: // hàm thành viên

X∗ operator&(); // tiền tố 1 ngôi (&)

X operator&(X); // 2 ngôi & (And)

X operator++(int); // tăng hậu tố

X operator&(X,X); // Lỗi: có 3 thứ

X operator/(); // Lỗi: có 1 ngôi

};

// hàm không là thành viên

X operator−(X); // tiền tố 1 ngôi (-)

X operator−(X,X); // 2 ngôi (-)

X operator−−(X&,int); // giảm hậu tố

X operator−(); // Lỗi: không có toán hạng

X operator−(X,X,X); // Lỗi: có 3 thứ

X operator%(X); // Lỗi: có 1 ngôi (%)

* **Ý nghĩa được xác định trước cho các toán tử**

Ý nghĩa của một số toán tử dựng sẵn được định nghĩa tương đương với một số kết hợp của các toán tử khác trên cùng các đối số.

class X {

public:

void operator=(const X&) = delete;

void operator&() = delete;

void operator,(const X&) = delete;

};

void f(X a, X b){

a = b; // Lỗi : không có toán tử =()

&a; // Lỗi : không có toán tử &()

a,b; // Lỗi : không có toán tử,()

}

Ngoài ra, chúng có thể được đưa ra các nghĩa mới bằng các định nghĩa phù hợp.

* **Toán tử và các loại do người dùng xác định**

Một hàm toán tử phải là một thành viên hoặc có ít nhất một đối số của kiểu do người dùng xác định (các hàm xác định lại toán tử new và delete không cần). Một hàm toán tử nhằm chấp nhận một kiểu dựng sẵn vì toán hạng đầu tiên của nó không thể là một hàm thành viên.

Liệt kê là kiểu do người dùng xác định để chúng ta có thể xác định các toán tử cho chúng.

enum Day { sun, mon, tue, wed, thu, fri, sat };

Day& operator++(Day& d){

return d = (sat==d) ? sun : static\_cast<Day>(d+1);

}

* + 1. **Một loại số phức**

void f(complex x, complex y, complex z){

complex r1 {x+y+z}; // r1 = operator+(operator+(x,y),z)

complex r2 {x}; // r2 = x

r2 += y; // r2.operator+=(y)

r2 += z; // r2.operator+=(z)

}

Ngoại trừ sự khác biệt về tính hiệu quả có thể xảy ra, các tính toán của r1 và r2 là tương đương

1. **Chương 19**
   * 1. **Giới thiệu**

Nạp chồng không chỉ dành cho các phép toán số học và logic. Trên thực tế, các toán tử đóng vai trò quan trọng trong việc thiết kế vùng chứa, '' con trỏ thông minh '', iterator và các class khác liên quan đến quản lý tài nguyên.

* + 1. **Các toán tử đặc biệt**
* Toán tử gián tiếp Toán tử gián tiếp,<> (còn được gọi là toán tử mũi tên), có thể được định nghĩa là toán tử hậu tố một ngôi. Ví dụ:

class Ptr {

X∗ operator−>(); };

Việc biến đổi đối tượng p thành con trỏ p.operator  () không phụ thuộc vào thành viên m được trỏ tới. Đó là nghĩa mà toán tử  () là một toán tử hậu tố một ngôi. Tuy nhiên, không có cú pháp mới nào được giới thiệu, vì vậy tên thành viên vẫn được yêu cầu sau dấu <> void g(Ptr p){ X∗ q1 = p−>; // Lỗi X∗ q2 = p.operator−>(); // OK }

* Tăng và Giảm Khi con người phát minh ra ''con trỏ thông minh'', họ thường quyết định cung cấp toán tử tăng ++ và toán tử giảm −− để phản ánh việc sử dụng các toán tử này cho các kiểu tích hợp. Điều này đặc biệt rõ ràng và cần thiết khi mục đích là thay thế một loại con trỏ thông thường bằng một loại '' con trỏ thông minh '' có cùng ngữ nghĩa, ngoại trừ việc nó thêm một chút kiểm tra lỗi thực thi.

Ví dụ: hãy xem xét một chương trình truyền thống

void f1(X a){

// cách truyền thống X v[200]; X∗ p = &v[0];

p−−;

∗p = a;

// oops: p ngoài phạm vi ++p;

∗p = a;

// OK }

Ở đây, chúng ta có thể muốn thay thế X ∗ bằng một đối tượng của một lớp Ptr <X> chỉ có thể được tham chiếu nếu nó thực sự trỏ đến X. void f2(Ptr<X> a){ // đã kiểm tra X v[200]; Ptr<X> p(&v[0],v); p−−; ∗p = a; // Lỗi thực thi: p ngoài phạm vi ++p; ∗p = a; // OK }

* Cấp phát và giải phóng Toán tử new có được bộ nhớ của nó bằng cách gọi một toán tử new (). Tương tự, toán tử xóa giải phóng bộ nhớ của nó bằng cách gọi một toán tử delete (). void∗ operator new(size\_t); // sử dụng cho từng đối tượng void∗ operator new[](size\_t); // sử dụng cho mảng void operator delete(void∗, size\_t); // sử dụng cho từng đối tượng void operator delete[](void∗, size\_t); // sử dụng cho mảng
  + 1. **Các hàm hỗ trợ Một tập hợp các hàm hữu ích**

Luồng I/O, hỗ trợ các vòng lặp phạm vi cho, so sánh và nối. Tất cả những điều này đều phản ánh các lựa chọn thiết kế được sử dụng cho std::string. Cụ thể, << chỉ in các ký tự mà không cần thêm định dạng và >> bỏ qua khoảng trắng đầu tiên trước khi đọc cho đến khi tìm thấy khoảng trắng kết thúc. ostream& operator<<(ostream& os, const String& s){ return os << s.c\_str(); } istream& operator>>(istream& is, String& s){ s = ""; // xóa xâu mục tiêu is>>ws; // bỏ qua khoảng trắng char ch = ' '; while(is.get(ch) && !isspace(ch)) s += ch; return is; } bool operator!=(const String& a, const String& b){ return !(a==b); }

* + 1. **Hàm bạn một hàm được khai báo friend được cấp quyền truy cập vào việc triển khai một lớp giống như một hàm thành viên nhưng độc lập với lớp đó.**

Ví dụ, chúng ta có thể xác định một toán tử nhân Ma trận với một Vectơ. Tuy nhiên, thói quen nhân của chúng ta không thể là thành viên của cả hai. Ngoài ra, chúng ta không thực sự muốn cung cấp các chức năng truy cập cấp thấp để cho phép mọi người dùng có thể đọc và ghi toàn bộ biểu diễn của cả Ma trận và Vectơ. Để tránh điều này, chúng ta khai báo toán tử là bạn của cả hai constexpr rc\_max {4}; // size của hàng và cột class Matrix { Vector v[rc\_max]; friend Vector operator∗(const Matrix&, const Vector&); };

class Vector { float v[rc\_max]};

friend Vector operator∗(const Matrix&, const Vector&);

};

Bây giờ toán tử ∗() có thể tiếp cận việc triển khai cả Vectơ và Ma trận. Điều đó sẽ cho phép các kỹ thuật triển khai phức tạp, nhưng thực hiện đơn giản:

Vector operator∗(const Matrix& m, const Vector& v){

Vector r;

for (int i = 0; i!=rc\_max; i++)

{

// r[i] = m[i] \* v;

r.v[i] = 0;

for (int j = 0; j!=rc\_max; j++)

{

r.v[i] += m.v[i].v[j] ∗ v.v[j];

}

**return r;**

**}**

# **CHƯƠNG 2: MỞ ĐẦU**

1. **Lý do chọn đề tài**

Đề tài “làm game” là đề tài chung dễ dàng thực hiện cho các khóa sinh viên năm hai. Source code gần gũi với sinh viên giúp nâng cao kỹ năng lập trình và kiến thức của sinh viên về lập trình hướng đối tượng (OOP).

Việc chọn đề tài và làm nhóm cũng giúp tăng tinh thần đoàn kết và rèn luyện khả năng làm việc nhóm giữa các sinh viên với nhau.

Từ các lợi ích nêu trên, đó là lý do tôi chọn đề tài “làm game” này.

1. **Mục tiêu nghiên cứu**

Cải thiện kỹ năng lập trình cho sinh viên, hiểu biết thêm về một số kiến thức để đi sâu vào chuyên ngành.

Mở rộng kiến thức cho sinh viên chỉ với một game đơn giản. Nâng cao tinh thần đồng đội, rèn luyện kỹ năng làm việc nhóm.

1. **Đối tượng nghiên cứu**

Tìm hiểu về game Rogue Lite đơn giản. Khái niệm game Rogue Lite và lên ý tưởng tiến hành hiện thực hóa.

Xác định các bước xây dựng game.

Xây dựng các hàm cần triển khai.

# **CHƯƠNG 3: TÌM HIỂU VỀ NGÔN NGỮ LẬP TRÌNH C++**

* 1. **Tổng quan về ngôn ngữ lập trình C++**
     1. Khái niệm

C ++ là một ngôn ngữ đa nền tảng có thể được sử dụng để tạo ra các ứng dụng hiệu suất cao. C ++ cung cấp cho lập trình viên khả năng kiểm soát cao đối với tài nguyên hệ thống và bộ nhớ. C ++ được phát triển bởi Bjarne Stroustrup, như một phần mở rộng của ngôn ngữ C.

* + 1. Đặc điểm

C ++ là một trong những ngôn ngữ lập trình phổ biến nhất thế giới. C ++ có thể được tìm thấy trong các hệ điều hành ngày nay. Giao diện người dùng đồ họa và các hệ thống nhúng. C ++ là một ngôn ngữ lập trình hướng đối tượng mang lại cấu trúc rõ ràng cho các chương trình và cho phép mã được sử dụng lại, giảm chi phí phát triển. C ++ có tính di động và có thể được sử dụng để phát triển các ứng dụng có thể thích ứng với nhiều nền tảng. C ++ rất thú vị và dễ học.

* 1. **Các tính chất quan trọng**

Ngôn ngữ lập trình C++ là ngôn ngữ lập trình hướng đối tượng. Có bốn tính chất cơ bản: đóng gói, kế thừa, đa hình và trừu tượng.

* + 1. Tính đóng gói

Tính đóng gói được thể hiện khi các thuộc tính, phương thức định tạo trong một lớp, liên kết dữ liệu và các chức năng thao tác chúng với nhau. Chúng ta không thể truy cập trực tiếp bất kỳ chức năng nào từ lớp. Chúng ta cần một đối tượng để truy cập hàm đó đang sử dụng biến thành viên của lớp đó. Hàm mà chúng ta đang tạo bên trong lớp, nó phải sử dụng tất cả biến thành viên thì chỉ nó được gọi là đóng gói. Nếu chúng ta không tạo hàm bên trong lớp đang sử dụng biến thành viên của lớp thì chúng ta không gọi nó là đóng gói.

* + 1. Tính kế thừa

Tính kế thừa được thể hiện khi một lớp khả năng của một lớp để lấy các thuộc tính và đặc điểm từ một lớp khác. Kế thừa là một trong những tính năng quan trọng nhất của Lập trình hướng đối tượng. Kế thừa là một tính năng mà các lớp mới được tạo ra từ các lớp hiện có. Lớp mới được tạo được gọi là “lớp dẫn xuất” hoặc “lớp con” và lớp hiện có được gọi là “lớp cơ sở” hoặc “lớp cha”. Lớp dẫn xuất kế thừa tất cả các thuộc tính của lớp cơ sở, mà không thay đổi các thuộc tính của lớp cơ sở và có thể thêm các tính năng mới cho riêng nó. Các tính năng mới này trong lớp dẫn xuất sẽ không ảnh hưởng đến lớp cơ sở.

* + 1. Tính đa hình

Tính đa hình hiểu nôm na là có nhiều dạng. Tính đa hình được coi là một trong những tính năng quan trọng của Lập trình hướng đối tượng. Có hai loại đa hình: đa hình thời gian biên dịch và đa hình thời gian chạy. Đa hình thời gian biên dịch đạt được bằng cách nạp chồng hàm hoặc nàm chồng toán tử. Đa hình thời gian chạy đạt được bằng tính năng ghi đè. Ghi đè hàm xảy ra khi một lớp dẫn xuất có định nghĩa cho một trong các hàm thành viên của lớp cơ sở. Hàm cơ sở đó được cho là bị ghi đè. Điều này thuận tiện khi gọi các chức năng của từng đối tượng khác nhau trong 1 danh sách, chức năng này sẽ gọi đúng đối với từng loại đối tượng.

* + 1. Tính trừu tượng

Trừu tượng có nghĩa là chỉ hiển thị thông tin cần thiết và ẩn các chi tiết. Trừu tượng hóa dữ liệu đề cập đến việc chỉ cung cấp thông tin cần thiết về dữ liệu cho bên ngoài, che giấu các chi tiết cơ bản hoặc việc triển khai. Các thành viên được khai báo là công khai trong một lớp có thể được truy cập từ bất kỳ đâu trong chương trình. Các thành viên được khai báo là riêng tư trong một lớp, chỉ có thể được truy cập từ bên trong lớp. Chúng không được phép truy cập từ bất kỳ phần nào của mã bên ngoài lớp.

# **CHƯƠNG 4:** **XÂY DỰNG GAME ROGUE LITE**

* 1. **Tích hợp thư viện SDL2 vào dự án trên Visual Studio**
* Tạo dự án console app.

**Graphical user interface, text, application, Word, email

Description automatically generated**

**Hình 4.1 Tạo dự án console app**

* Chuyển sang phiên bản x86 và chạy chương trình.

**Graphical user interface, text, application

Description automatically generated**

**Hình 4.2 Chạy chương trình**

* Vào thư mục chứ dự án, copy thư mục SDL\_lib chứa thư viện SDL2 đã chuẩn bị sẵn vào thư mục này.

**A screenshot of a computer

Description automatically generated with medium confidence**

**Hình 4.3 Copy thư mục SDL\_lib vào dự án**

* Vào thuộc tính của dự án → C/C++ → General → Additional Include Directories → Edit.

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.4 Set up môi trường**

* Một cửa sổ sẽ hiện lên, nhập vào đường dẫn như hình bên dưới, lick OK.

Graphical user interface, text, application

Description automatically generated

**Hình 4.5 Set up môi trường**

* Vào linker → General → Additional Library Directories → Edit.

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.6 Setup môi trường**

* Thêm đường dẫn các thư mục như hình dưới, click OK.

Graphical user interface, text, application

Description automatically generated

**Hình 4.7 Set up môi trường**

* Thêm các tên file lib như hình dưới, click OK.

Graphical user interface, text, application

Description automatically generated

**Hình 4.8 Setup môi trường**

* Nhấn Apply → OK.

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.9 setup môi trường**

* Copy toàn bộ file dll vào thư mục debug của dự án

**Graphical user interface, text

Description automatically generated**

**Hình 4.10 Copy file dll vào thư mục dự án**

* Khai báo thư viện SDL.h và chạy chương trình

Graphical user interface, text, application

Description automatically generated

**Hình 4..11 Chương trình SDL cơ bản**

Graphical user interface, application

Description automatically generated

**Hình 4.12 Chạy chương trình thành công**

* 1. **Thiết kế thuật toán**

**Diagram

Description automatically generated**

**Hình 4.13 sơ đồ các lớp kế thừa**

Nhìn vào sơ đồ, chúng ta thấy được hệ thống các lớp kế thừa được tổ chức như thế nào.

**Character** là lớp cơ sở, **Player** và **Enemy** được kế thừa từ lớp này. **Enemy\_1**, **Enemy\_2** và **Enemy\_3** được xem là 3 loại Enemy khác nhau được kế thừa từ **Enemy.**

Các lớp hầu như đều thể hiện tính đóng gói và trừu tượng khi các dữ liệu và chức năng được liên kết với nhau và thuộc tính được ẩn đi.

Ngoài ra còn một lớp **Enemies** để quản lý danh sách **Enemy**. Số lượng và mỗi loại **Enemy** được tạo ra. Khi thực hiện các chức năng mỗi Enemy trong danh sách, sẽ tùy vào từng loại **Enemy\_1**, **Enemy\_2** và **Enemy\_3** mà phương thức xử lý thêm đạn và vẽ đạn theo mỗi cách khác nhau.

* 1. **Thiết kế bản đồ**

Sử dụng phần mềm **Tiled** để tạo bản đồ.

Muốn tạo được bản đồ, chúng ta cần có các **tileset**. Tileset là một hình ảnh bình thường, tuy nhiên hình ảnh này đưa vào phần mềm nó sẽ được chia ra thành các ô vuông nhỏ 32x32 pixel tùy vào cài đặt.

Bản đồ gồm nhiều các layout xếp chồng lên nhau, mỗi layout được tạo từ các ô vuông được trong tileset. Các layout xếp chồng lên nhau sẽ tạo thành một bản đồ hoàn chỉnh.

Graphical user interface

Description automatically generated

**Hình 4.14 Sử dụng Tiled để tạo bản đồ bằng các tileset**

* 1. **Giao diện game Rogue Lite**

**A screenshot of a video game

Description automatically generated**

**Hình 4.15** **Menu**

Website

Description automatically generated

**Hình 4.16** **Màng hình nhập tên người chơi**

Shape

Description automatically generated

**Hình 4.17** **Màng hình hướng dẫn**

Graphical user interface, text

Description automatically generated

**Hình 4.18** **Màng hình hiển thị score**

Graphical user interface

Description automatically generated

**Hình 4.19** **Giao diện chơi game**

* 1. **Hướng dẫn**
* Sử dụng các phím mũi tên để di chuyển.
* Nút Space để bắn.
* Sau mỗi màng, độ khó sẽ tăng lên, đồng thời máu của nhân vật sẽ được reset lại thành 10 cho người chơi dễ dàng hơn.

**CHƯƠNG 5: KẾT QUẢ**

* 1. **Kết quả**
* Xây dựng thành công game Rogue Lite cơ bản.
* Các thành viên trong nhóm nắm được kiến thức về vấn đề làm một game cơ bản.
* Cải thiện tư duy lập trình và rèn luyện kỹ năng làm việc nhóm.
  1. **Nhược điểm**
* Còn nhiều lỗi sai sót.
* Chương trình chưa thật sự tối ưu.
* Còn hơi chậm hiểu trong quá trình là đề tài.
* Nhóm trưởng chưa phân công công việc và đánh giá thành viên rõ ràng.
  1. **Công việc và mức độ hoàn thành của các thành viên**
* Hồ Vĩnh Tín: Code chính (100%).
* Phan Ngọc Như Tranh: Phụ code và làm báo cáo (100%).
* Nguyễn Ngọc Tiểu Thư: Phụ code và làm slide (100%).
* Lê Công Thuận: Phụ code và thiết kế giao diện (100%).

# **TÀI LIỆU THAM KHẢO**

[1] Google, "SDL\_Keycode". [Online]. Available:  <https://wiki.libsdl.org/SDL_Keycode>  
[Accessed 21 10 2022].

[2] Google, "SDL Tutorials". [Online]. Available:  <https://wiki.libsdl.org/Tutorials>   
[Accessed 24 9 2022].

[3] Google, "Beginning Game Programming v2.0". [Online].  
Available:  <https://lazyfoo.net/tutorials/SDL/>   
[Accessed 01 10 2022].

[4] Google, "Pinterset" [Online]. Available:  <https://www.pinterest.com/>   
[Accessed 27 10 2022].