**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

                                      ──────── \* ───────

A red and yellow logo

Description automatically generated

**PROJECT III**

**ĐỀ TÀI: Lập trình game 2D ( Spaceship over terrain )**

Giảng viên hướng dẫn: ThS. Bành Thị Quỳnh Mai

Sinh viên thực hiện: Hà Văn Quang

Mssv: 20204775

Lớp: Kĩ thuật máy tính 02 -k65

***Hà Nội, tháng 1 năm 2024***

Mục lục

[**Danh mục các kí hiệu:** 2](#_Toc156353837)

[**Lời nói đầu** 4](#_Toc156353838)

[**Chương 1: Tổng quan về đề tài:** 5](#_Toc156353839)

[**1.1.Mô tả về game** 5](#_Toc156353840)

[**1.2.Yêu cầu đối với sản phẩm.** 5](#_Toc156353841)

[**1.3.Công cụ sử dụng.** 6](#_Toc156353842)

[**Chương 2: Cơ sở lý thuyết** 6](#_Toc156353843)

[**2.1. Thư viện sdl** 6](#_Toc156353844)

[*2.1.1.Đa nền tảng(Cross-Platform)* 6](#_Toc156353845)

[*2.1.2.Quản lý cửa sổ và đồ họa* 6](#_Toc156353846)

[*2.1.3.Xử lý sự kiện* 6](#_Toc156353847)

[*2.1.4.Âm thanh và âm nhạc* 7](#_Toc156353848)

[*2.1.5.Tương tác với đồ họa hình ảnh* 7](#_Toc156353849)

[*2.1.6.Hiệu suất tối ưu hóa* 7](#_Toc156353850)

[*2.1.7.Ngôn ngữ lập trình* 7](#_Toc156353851)

[*2.1.8.Cộng đồng tài nguyện* 7](#_Toc156353852)

[**2.2.Cách tiếp cận: Hướng đối tượng.** 7](#_Toc156353853)

[**2.3.Phân tích thiết kế hướng đối tượng và UML** 8](#_Toc156353854)

[*2.3.1. View (góc nhìn):* 9](#_Toc156353855)

[*2.3.2. Diagram (bản vẽ)* 10](#_Toc156353856)

[*2.3.3. Notations (ký hiệu):* 12](#_Toc156353857)

[Notations là các ký hiệu để vẽ, nó như từ vựng trong ngôn ngữ tự nhiên. Bạn phải biết từ vựng thì mới ghép thành câu, thành bài được. Xem ví dụ từ bảng các ký hiệu. 12](#_Toc156353858)

[*2.3.4. Mechanisms (qui tắc, cơ chế):* 12](#_Toc156353859)

[**Chương 3: Phân tích và thiết kế hệ thống** 12](#_Toc156353860)

[**3.1. Phân tích game** 12](#_Toc156353861)

[**3.2. Biểu đồ use-case** 13](#_Toc156353862)

[*3.2.1. Danh sách Actor* 13](#_Toc156353863)

[*3.2.2.Danh sách use-case* 14](#_Toc156353864)

[*3.2.3.Đặt tả use-case* 14](#_Toc156353865)

[**3.3.Biểu đồ hoạt động** 17](#_Toc156353866)

[*3.3.1.Biểu đồ hoạt động usecase “chơi game”* 17](#_Toc156353867)

[*3.3.2. Biểu đồ hoạt động usecase “thoát game”* 18](#_Toc156353868)

[*3.3.3. Biểu đồ hoạt động usecase “xem hướng dẫn”* 18](#_Toc156353869)

[*3.3.4. Biểu đồ hoạt động usecase “bật/tắt âm thanh”* 19](#_Toc156353870)

[**3.4.Biểu đồ lớp.** 19](#_Toc156353871)

[**Chương 4: Giao diện game** 20](#_Toc156353872)

[**Chương 5: Kết luận** 24](#_Toc156353873)

[**5.1.Kết quả đạt được** 24](#_Toc156353874)

[**5.2. Hướng phát triển** 24](#_Toc156353875)

[**5.3.Hạn chế** 24](#_Toc156353876)

# **Danh mục các kí hiệu:**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

# 

# **Lời nói đầu**

[Trên](https://fr.slideshare.net/TTHNHT/bo-co-phn-tch-thit-k-n-game#8)thế giới cũng như Việt Nam, công nghệ thông tin đã trở thành một công nghệ mũi nhọn, nó là ngành công nghệ kỹ thuật không thể thiếu trong việc áp dụng vào các hoạt động xã hội như: các hoạt động quản lý hệ thống, kinh doanh, trong hoạt động nghiên cứu,… và đặc biệt là trong lĩnh vực game – một lĩnh vực đang phát triển mạnh hiện nay.

Với mong muốn được học hỏi, tiếp thu kiến thức và niềm đam mê về bộ môn game, Nên em đã chọn lập trình game là đề tài để mình phát triển trong môn project3 này. Game có tên là Spaceship over terrain( là một tựa game 2D được phát triển dựa trên ý tưởng của tựa game nổi tiếng “flappy bird”.

Mặc dù đã cố gắng để hoàn thành tốt, nhưng do thời gian có hạn và thiếu kinh nghiệm, cũng như kĩ năng chưa cao nên game của em vẫn còn nhiều thiếu sót, kính mong thầy cô và các bạn góp ý để em có thể hoàn thiện và phát triển tựa game này trong tương lai. Em xin chân thành cảm ơn.

# 

# **Chương 1: Tổng quan về đề tài:**

## **1.1.Mô tả về game**

Như đã nói, game được lấy trên ý tưởng của tự game “flappy bird”, nhưng chỗ khác ở đây là chúng ta sẽ dùng chuột phải để điều khiển phi thuyền bay lên xuống và dùng chuột trái để cho phi thuyền dùng kĩ năng để vượt qua địa hình đặc biệt.

Hiện tại, game đang có 2 ải, với ải thứ 2 phải dùng tới kĩ năng thì mới vượt qua được. Phi thuyền sẽ được bay với vận tốc ổn định và phù hợp. Sau khi vượt qua màn sẽ có giao diện chiến thắng. Người chơi cũng có thể dừng game, bật tắt tiếng game và bật tắt tiếng giao diện.

Đây sẽ là tự game khiến game thủ phải sử dụng kĩ năng và độ nhanh nhạy, những chướng ngại vật sẽ khó dần lên, nên người chơi phải tập trung mới có thể chiến thắng được.

Khi phi thuyền đâm vào chướng ngại vật sẽ tạo ra vụ nổ, với âm thanh và hình ảnh kích thích. Sau đó người chơ sẽ bị out ra màn hình map game, để lựa chọn map game mình thích và phù hợp.

Rất mong “Spaceship over terrain” sẽ được mọi người yêu thích và ủng hộ.

## **1.2.Yêu cầu đối với sản phẩm.**

- Game phải có dung lượng không quá lớn, tốc độ xử lý nhanh

- Giao diện game dễ nhìn, than thiện

- Công việc tính toán ( tốc độ, vận tốc,..) phải được thực hiện chính xác

- Tạo cảm giác thích thú cho người chơi

- Phân tích game theo hướng đối tượng cụ thể, rõ rang

## **1.3.Công cụ sử dụng.**

- Sử dụng thư viện sdl, ngôn ngữ lập trình C++

- Dùng tiled map editor để tạo map.

- Lập trình hướng đối tượng

- Tài nguyên được tạo ra từ phần mềm photoshop

- Dùng internet để tìm kiếm hình ảnh và âm thanh phù hợp với trò chơi.

# 

# **Chương 2: Cơ sở lý thuyết**

## **2.1. Thư viện sdl**

SDL là một thư viện đa nền tảng (cross-platform) được thiết kế để hỗ trợ việc phát triển ứng dụng multimedia, đặc biệt là các ứng dụng và trò chơi đồ họa 2D.

### *2.1.1.Đa nền tảng(Cross-Platform)*

SDL hỗ trợ nhiều hệ điều hành như Windows, macOS, Linux, và nhiều nền tảng khác, giúp người phát triển dễ dàng di động ứng dụng của họ giữa các hệ điều hành mà không cần phải viết lại code nhiều.

### *2.1.2.Quản lý cửa sổ và đồ họa*

SDL cung cấp các hàm để tạo và quản lý cửa sổ, xử lý sự kiện cửa sổ, và hiển thị đồ họa 2D. Người phát triển có thể dễ dàng tạo cửa sổ đồ họa và vẽ đồ họa 2D trực tiếp lên cửa sổ đó.

### *2.1.3.Xử lý sự kiện*

SDL hỗ trợ xử lý sự kiện từ bàn phím, chuột, và các sự kiện khác. Người phát triển có thể dễ dàng bắt các sự kiện này để điều khiển luồng logic của ứng dụng.

### *2.1.4.Âm thanh và âm nhạc*

SDL cung cấp tính năng chơi âm thanh và âm nhạc. Điều này giúp làm giàu trải nghiệm người chơi trong các ứng dụng và trò chơi đa phương tiện.

### *2.1.5.Tương tác với đồ họa hình ảnh*

SDL cho phép người phát triển tương tác với đồ họa và hình ảnh thông qua các API đơn giản. Có thể load hình ảnh từ nhiều định dạng khác nhau và vẽ chúng lên cửa sổ đồ họa.

### *2.1.6.Hiệu suất tối ưu hóa*

SDL được thiết kế để đảm bảo hiệu suất tốt và có khả năng tối ưu hóa. Thư viện này hỗ trợ các kỹ thuật như double buffering để tránh hiện tượng giật lag.

### *2.1.7.Ngôn ngữ lập trình*

SDL thường được sử dụng với ngôn ngữ lập trình C, nhưng cũng có các liên kết với nhiều ngôn ngữ khác như C++, Python, và nhiều ngôn ngữ khác.

### *2.1.8.Cộng đồng tài nguyện*

SDL có một cộng đồng lớn và tích cực. Có nhiều tài liệu, ví dụ, và diễn đàn hỗ trợ, giúp người phát triển nhanh chóng giải quyết vấn đề và học hỏi từ cộng đồng.

=> SDL là một công cụ linh hoạt và mạnh mẽ, thích hợp cho việc phát triển ứng dụng và trò chơi đồ họa 2D trên nhiều nền tảng.

## **2.2.Cách tiếp cận: Hướng đối tượng.**

  Lấy đối tượng làm trung tâm.

Đối tượng = chức năng + dữ liệu.

Hệ thống = tập trung các đối tượng + quan hệ giữa các đối tượng.

Cách tiếp cận hướng đối tượng là một lối tư duy theo cách ánh xạ các thành phần trong bài toán vào các đối tượng vào đời thực. Với cách tiếp cận này, một hệ thống được chia tương ứng thành các thành phần nhỏ gọi là các đối tượng, mỗi đối tượng bao gồm đầy đủ cả dữ liệu và hành động liên quan đến đối tượng đó.

Ưu điểm:

• Gần gũi với thế giới thực.

• Tái sử dụng dễ dàng

• Đóng gói, che dấu thông tin làm cho hệ thống tin cậy hơn.

• Thừa kế giảm chi phí, hệ thống có tính mở cao.

• Phù hợp với hệ thống lớn và phức tạp.

## **2.3.Phân tích thiết kế hướng đối tượng và UML**

Khái niệm UML: (Unified Modeling Language)

UML là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau cũng như dùng để thi công hệ thống (phát triển), thuyết phục khách hàng, các nhà đầu tư ... (Giống như trong xây dựng người ta dùng các bản vẽ thiết kế để hướng dẫn và kiểm soát thi công, bán hàng căn hộ ...).

Phân tíchthiết kế hướng đối tượng cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung thể hiện. Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó nên chúng thường đi đôi với nhau.

UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng … trong phạm vi các bài báo cáo này chúng ta chỉ nghiên cứu cách sử dụng UML cho phân tích và thiết kế hướng đối tượng trong ngành phần mềm. Phân tích thiết kế hướng đối tượng sử dụng UML bao gồm các thành phần sau:

• View (góc nhìn)

• Diagram (bản vẽ)

• Notations (ký hiệu)

• Mechanisms (qui tắc, cơ chế)

### *2.3.1. View (góc nhìn):*

Mỗi góc nhìn như thầy bói xem voi, nó không thể hiện hết hệ thống nhưng thể hiện rõ hệ thống ở một khía cạnh. Chính vì thế trong xây dựng có bản vẽ kiến trúc (nhìn về mặt kiến trúc), bản vẽ kết cấu (nhìn về mặt kết cấu), bản vẽ thi công (nhìn về mặt thi công). Trong phần mềm cũng như vậy, Phân tích thiết kế hướng đối tượng sử dụng UML có các góc nhìn sau:

Ảnh có chứa văn bản, Phông chữ, biên lai, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 1. Sơ đồ cấu trúc góc nhìn view

• Use Case View: cung cấp góc nhìn về các case sử dụng giúp chúng ta hiểu hệ thống có gì? ai dùng và dùng nó như thế nào.

• Logical View: cung cấp góc nhìn về cấu trúc hệ thống, xem nó được tổ chức như thế nào. Bên trong nó có gì.

• Process View: cung cấp góc nhìn động về hệ thống, xem các thành phần trong hệ thống tương tác với nhau như thế nào.

• Component View: Cũng là một góc nhìn về cấu trúc giúp chúng ta hiểu cách phân bổ và sử dụng lại các thành phần trong hệ thống ra sao.

• Deployment View: cung cấp góc nhìn về triểnkhai hệ thống, nó cũng ảnh hưởng lớn đến kiến trúc hệ thống.

• Tập hợp các góc nhìn này sẽ giúp chúng ta hiểu rõ hệ thống cần phân tích, thiết kế. Trong Hình 2.1 chúng ta thấy góc nhìn Use Case View nằm ở giữa và chi phối tất cả các góc nhìn còn lại. Chính vì thế chúng ta thường thấy các tài liệu nói về 4 view + 1 chứ không phải 5 view nhằm nhấn mạnh vai trò của Use Case View.

### *2.3.2. Diagram (bản vẽ)*

Diagram có thể dịch là sơ đồ. Tuy nhiên ở đây chúng ta sử dụng từ bản vẽ cho dễ hình dung. Các bản vẽ được dùng để thể hiện các góc nhìn của hệ thống.

Ảnh có chứa văn bản, vòng tròn, biểu đồ, hàng

Mô tả được tạo tự động

Hình 2-sơ đồ bản vẽ diagram

Trong đó:

• Use Case Diagram: bản vẽ mô tả về case sử dụng của hệ thống. Bản vẽ này sẽ giúp chúng ta biết được ai sử dụng hệ thống, hệ thống có những chức năng gì. Lập được bản vẽ này bạn sẽ hiểu được yêu cầu của hệ thống cần xây dựng.

• Class Diagram: bản vẽ này mô tả cấu trúc của hệ thống, tức hệ thống được cấu tạo từ những thành phần nào. Nó mô tả khía cạnh tĩnh của hệ thống.

• Object Diagram: Tương tự như Class Diagram nhưng nó mô tả đến đối tượng thay vì lớp (Class).

• Sequence Diagarm: là bản vẽ mô tả sự tương tác của các đối tượng trong hệ thống với nhau được mô tả tuần tự các bước tương tác theo thời gian.

• Collaboration Diagram: tương tự như sequence Diagram nhưng nhấn mạnh về sự tương tác thay vì tuần tự theo thời gian.

• State Diagram: bản vẽ mô tả sự thay đổi trạng thái của một đối tượng. Nó được dùng để theo dõi các đối tượng có trạng thái thay đổi nhiều trong hệ thống.

• Activity Diagram: bản vẽ mô tả các hoạt động của đối tượng, thường được sử dụng để hiểu về nghiệp vụ của hệ thống.

• Component Diagram: bản vẽ mô tả về việc bố trí các thành phần của hệ thống cũng như việc sử dụng các thành phần đó.

• Deployment Diagram: bản vẽ mô tả việc triển khai của hệ thống như việc kết nối, cài đặt, hiệu năng của hệ thống v.v….

Lưu ý: Ở đây chúng ta sử dụng từ hệ thống tương đương với sản phẩm phần mềm.

### *2.3.3. Notations (ký hiệu):*

### Notations là các ký hiệu để vẽ, nó như từ vựng trong ngôn ngữ tự nhiên. Bạn phải biết từ vựng thì mới ghép thành câu, thành bài được. Xem ví dụ từ bảng các ký hiệu.

### *2.3.4. Mechanisms (qui tắc, cơ chế):*

Mechanisms là các qui tắc để lập nên bản vẽ, mỗi bản vẽ có qui tắc riêng và bạn phải nắm được để tạo nên các bản vẽ thiết kế đúng.

# **Chương 3: Phân tích và thiết kế hệ thống**

## **3.1. Phân tích game**

Có một màn hình chơi game gồm các đối tượng

* Background là nột hình tĩnh
* Một chiếc phi thuyền – nhân vật game
* Sự kiện nhấp chuột phải dùng để di chuyển lên xuống và nhấp chuột trái để tốc biến
* Trong các màn chơi sẽ xuất hiện các chướng ngại vật ở cả phía trên và phía dưới, các chướng ngại này sẽ di chuyển từ trái qua phải, người chơi phải dùng chuột điều khiển nhân vật né chướng ngại. Nếu phi thuyền chạm vào chướng ngại sẽ phát nổ, kết thúc game. Và khi đến điểm cuối map, sẽ hiện màn hình chiến thắng.

Game gồm 4 màn hình:

* Màn hình lúc mới bắt đầu game ( menu ): Gồm nút bắt đầu, hướng dẫn, tắt game và bật/ tắt âm thanh.
* Màn hình hướng dẫn chơi
* Màn hình hiển thị map: Sẽ hiển thị tất cả các map chơi
* Màn hình chơi game: gồm phi thuyền, background và chướng ngại vật.

## **3.2. Biểu đồ use-case**

Ảnh có chứa văn bản, biểu đồ, vòng tròn, hàng

Mô tả được tạo tự động

Hình 3- biểu đồ use-case

### *3.2.1. Danh sách Actor*

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

### *3.2.2.Danh sách use-case*

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

### *3.2.3.Đặt tả use-case*

3.2.3.1> Đặc tả use-case “chơi game”

1.Tóm tắt

Người chơi sử dụng use-case này để chơi game

2.Các dòng sự kiện

Use-case bắt đầu khi người chơi nhấn nút play và chọn màn chơi

Hệ thống hiển thị màn hình chơi game, nhân vật game- phi thuyền, background, chướng ngại vật

Người chơi phải sử dụng chuột để điều khiển phi thuyền vượt chướng ngại vật.

Khi phi thuyền chạm chướng ngại vật thì use-case này sẽ kết thúc

3.Các yêu cầu đặc biệt

Không có

4.Trạng thái hệ thống khi bắt đầu thực hiện use-case

Người dùng trong giao diện chơi game

5.Trạng thái hệ thống sau khi thực hiện use-case

Người dùng vẫn trong màn hình chơi game

6.Điểm mở rộng

Không có

3.2.3.2> Đặc tả use-case “Thoát game”

1.Tóm tắt

Người chơi sử dụng use-case này để thoát game khi không muốn chơi nữa

2.Các dòng sự kiện

Use-case bắt đầu khi người chơi nhấn nút exit

Hệ thống tắt game

3.Các yêu cầu đặc biệt

Game đang được mở trên màn hình máy tính

4.Trạng thái hệ thống khi bắt đầu thực hiện use-case

Người dùng trong giao diện chơi game

5.Trạng thái hệ thống sau khi thực hiện use-case

Người dùng vẫn trong màn hình chơi game

6.Điểm mở rộng

Không có

3.2.3.3> Đặc tả use-case “xem hướng dẫn”

1.Tóm tắt

Người chơi sử dụng use-case này để xem hướng dẫn chơi.

2.Các dòng sự kiện

Use-case bắt đầu khi người chơi nhấn nút “?”

Hệ thống mở giao diện hướng dẫn

3.Các yêu cầu đặc biệt

Game đang bật trên máy

4.Trạng thái hệ thống khi bắt đầu thực hiện use-case

Người dùng trong giao diện hướng dẫn

5.Trạng thái hệ thống sau khi thực hiện use-case

Người dùng vẫn trong màn hình hướng dẫn

6.Điểm mở rộng

Không có

3.2.3.4> Đặc tả use-case “bật/tắt âm thanh”

1.Tóm tắt

Người chơi sử dụng use-case này để bật hoặc tắt âm thanh

2.Các dòng sự kiện

Use-case bắt đầu khi người chơi nhấn nút bật/tắt âm thanh

Hệ thống điều chỉnh âm thanh theo yêu cầu

3.Các yêu cầu đặc biệt

Game đang bật

4.Trạng thái hệ thống khi bắt đầu thực hiện use-case

Người dùng trong giao diện menu

5.Trạng thái hệ thống sau khi thực hiện use-case

Người dùng vẫn trong màn hình menu

6.Điểm mở rộng

Không có

## **3.3.Biểu đồ hoạt động**

### *3.3.1.Biểu đồ hoạt động usecase “chơi game”*

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 4- biểu đồ hoạt động use-case "chơi game"

### *3.3.2. Biểu đồ hoạt động usecase “thoát game”*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, vòng tròn

Mô tả được tạo tự động

Hình 5- biểu đồ hoạt động usecase "thoát game"

### *3.3.3. Biểu đồ hoạt động usecase “xem hướng dẫn”*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, vòng tròn

Mô tả được tạo tự động

Hình 6- biểu đồ hoạt động usecase "xem hướng dẫn"

### *3.3.4. Biểu đồ hoạt động usecase “bật/tắt âm thanh”*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu tượng, thiết kế

Mô tả được tạo tự động

Hình 7- biểu đồ hoạt động usecase "bật/tắt âm thanh"

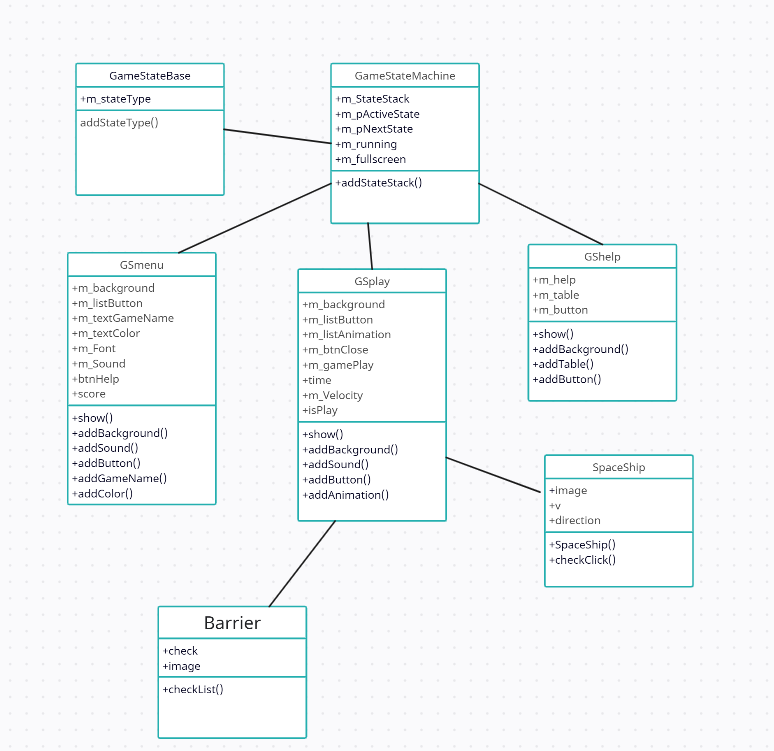
## **3.4.Biểu đồ lớp.**

Game gồm 2 màn hình chính:

* Màn hình Menu
* Màn hình chơi game

Và gồm 1 số đối tượng:

* Spaceship: Phi thuyền – nhân vật game
* Barrier: Vật cản



Hình 8- biểu đồ lớp

# **Chương 4: Giao diện game**

* Sau khi vào game, sẽ mở ra màn hình Menu

Ảnh có chứa văn bản, ảnh chụp màn hình, màu tím, hoa tím

Mô tả được tạo tự động

Hình 9- màn hình Menu

* Màn hình hướng dẫn

Ảnh có chứa văn bản, ảnh chụp màn hình, hoa tím, màu tím

Mô tả được tạo tự động

Hình 10- màn hình hướng dẫn

* Sau khi bấm play, xuất hiện nơi chọn màn chơi

Ảnh có chứa màu tím, hoa tím, Đỏ tía, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 11- giao diện màn chơi

* Giao diện chơi game

Ảnh có chứa ảnh chụp màn hình, điểm ảnh, tác phẩm nghệ thuật

Mô tả được tạo tự động

Hình 12- giao diện chơi game

* Khi phi thuyền va chạm

Ảnh có chứa văn bản, ảnh chụp màn hình, hình vuông, mẫu

Mô tả được tạo tự động

Hình 13- khi phi thuyền bị va chạm

# **Chương 5: Kết luận**

## **5.1.Kết quả đạt được**

Qua quá trình thực hiện đồ án lập trình game sử dụng thư viện sdl với sự hướng dẫn của cô Bành Thị Quỳnh Mai, em đã cài đặt thành công game Spaceship over terrain và chạy trên máy tính mình. Em cũng nắm được cơ sở lý thuyết về lập trình hướng đối tượng, biết cách dùng thư viện sdl và cải thiện lập trình C++. Đồ án 3 sẽ tạo tiền đề cho em phát triển đồ án tốt nghiệp cũng như các sản phẩm sau này.

## **5.2. Hướng phát triển**

- Tạo thêm nhiều map chơi mới với những cách chơi mới lạ hơn

- Phát triển trên nền tảng mobile

- Tăng độ khó lên

- Thiết kế đồ họa đẹp, sinh động hơn

## **5.3.Hạn chế**

Hạn chế lớn nhất là rất khó tìm asset, quá trình tìm asset mất rất nhiều thời gian mà không có, phải tự tạo bằng nhiều ứng dụng photoshop. Ngoài ra, việc chưa thành thạo lập trình hướng đối tượng cũng như làm việc với thư viện mới khiến quá trình làm quen gặp khó khăn.

Cách chơi và map chơi chưa được phong phú, code chưa được tối ưu.