|  |  |
| --- | --- |
|  |  |
| TRẦN THÀNH CÔNG  TRẦN THÀNH CÔNG | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH CÔNG NGHỆ THÔNG TIN |
|  |
| **XÂY DỰNG GAME 2D THE PUPPET’S REVENGE** |
|  |
|  |
| **CBHD: ThS Vũ Đức Huy** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên: Trần Thành Công** |
| **Mã số sinh viên: 2020602410** |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |
|  |

# LỜI CẢM ƠN

Với tình cảm sâu sắc và chân thành nhất, em xin được phép bày tỏ lòng biết ơn đến những người và tổ chức đã đồng hành và hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài “*Xây dựng game 2D The Puppet’s Revenge*”. Đặc biệt, em xin chân thành cảm ơn thầy Vũ Đức Huy - người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành bài báo cáo này trong thời gian qua. Nhờ có sự chỉ dẫn tận tình của thầy mà em đã có thể hoàn thành đề tài này.

Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi lời cảm ơn đến quý thầy cô ở khoa công nghệ thông tin đã truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập tại trường. Và em cũng muốn gửi lời cảm ơn đến bạn bè, đồng nghiệp và gia đình đã luôn đồng hành, cổ vũ, động viên em trong suốt thời gian qua.

Đồ án của em thực hiện trong khoảng thời gian 9 tuần. Do kinh nghiệm thực tế của em còn nhiều hạn chế nên không tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của thầy cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao kỹ năng nghề nghiệp của em.

Em xin chân thành cảm ơn!

Trần Thành Công

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc167005121)

[MỤC LỤC ii](#_Toc167005122)

[DANH MỤC HÌNH ẢNH v](#_Toc167005123)

[DANH MỤC BẢNG BIỂU viii](#_Toc167005124)

[DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT ix](#_Toc167005125)

[MỞ ĐẦU x](#_Toc167005126)

[Lí do chọn đề tài x](#_Toc167005127)

[Mục tiêu và phạm vi nghiên cứu x](#_Toc167005128)

[Ý nghĩa của đề tài xi](#_Toc167005129)

[Cấu trúc báo cáo xi](#_Toc167005130)

[CHƯƠNG 1. LÝ THUYẾT THIẾT KẾ GAME 1](#_Toc167005131)

[1.1 Một số nguyên tắc cơ bản của thiết kế game 1](#_Toc167005132)

[1.2 Thiết kế ý tưởng Game 1](#_Toc167005133)

[1.3 Thiết kế phát triển Game 2](#_Toc167005134)

[1.3.1. Thiết kế kịch bản game 2](#_Toc167005135)

[1.3.2. Thiết kế âm thanh 2](#_Toc167005136)

[CHƯƠNG 2. CÔNG CỤ HỖ TRỢ XÂY DỰNG GAME 3](#_Toc167005137)

[2.1 Giới thiệu về công cụ làm game Unity 3](#_Toc167005138)

[2.1.1 Tổng quan về Unity 3](#_Toc167005139)

[2.1.2 Ưu điểm 3](#_Toc167005140)

[2.1.3 Các thành phần và khái niệm cơ bản 4](#_Toc167005141)

[2.1.4 Đặc điểm và tính năng 6](#_Toc167005142)

[2.1.5 Giao diện cơ bản của Unity 7](#_Toc167005143)

[2.1.6 Cinemachine 11](#_Toc167005144)

[2.1.7 Input System 12](#_Toc167005145)

[2.2 Ngôn ngữ lập trình C# và Mono 13](#_Toc167005146)

[CHƯƠNG 3. XÂY DỰNG GAME 2D THE PUPPET’S REVENGE 15](#_Toc167005147)

[3.1 Thiết kế Game 15](#_Toc167005148)

[3.1.1 Thiết kế ý tưởng Game 15](#_Toc167005149)

[3.1.2. Thiết kế kịch bản game 15](#_Toc167005150)

[3.1.3. Thiết kế âm thanh 29](#_Toc167005151)

[3.2 Phân tích thiết kế hệ thống 30](#_Toc167005152)

[3.2.1 Giới thiệu các chức năng trong game 30](#_Toc167005153)

[3.2.2 Đặc tả các chức năng trong game 36](#_Toc167005154)

[3.2.3 Biểu đồ trình tự và biểu đồ lớp 45](#_Toc167005155)

[3.2.4 Thiết kế cơ sở dữ liệu 50](#_Toc167005156)

[3.2.5 Thiết kế giao diện 55](#_Toc167005157)

[3.3 Xây dựng và kết quả 55](#_Toc167005158)

[3.3.1 Giải quyết một số vấn đề 55](#_Toc167005159)

[3.3.2 Kết quả 57](#_Toc167005160)

[CHƯƠNG 4. TEST GAME 2D THE PUPPET’S REVENGE 61](#_Toc167005161)

[Test Report: 62](#_Toc167005162)

[KẾT LUẬN 63](#_Toc167005163)

[TÀI LIỆU THAM KHẢO 64](#_Toc167005164)

# DANH MỤC HÌNH ẢNH

[Hình 2.1: Unity là gì? 3](#_Toc166841736)

[Hình 2.2: Scene 5](#_Toc166841737)

[Hình 2.3: Tổng quan giao diện của Unity 8](#_Toc166841738)

[Hình 2.4: Scene View trong Unity 8](#_Toc166841739)

[Hình 2.5: Game View trong Unity 9](#_Toc166841740)

[Hình 2.6: Hierarchy trong Unity 9](#_Toc166841741)

[Hình 2.7: Inspector trong Unity 11](#_Toc166841742)

[Hình 2.8: Hình ảnh Cinemachine 12](#_Toc166841743)

[Hình 2.9: Hình ảnh Input System 13](#_Toc166841744)

[Hình 3.1: Biểu đồ Use Case tổng quát 31](#_Toc166841745)

[Hình 3.2: Biểu đồ chức năng chọn chế độ chơi 32](#_Toc166841746)

[Hình 3.3: Biểu đồ chức năng chọn số lượng người chơi 32](#_Toc166841747)

[Hình 3.4: Biểu đồ chức năng chọn Màn chơi 33](#_Toc166841748)

[Hình 3.5: Biểu đồ chức năng Cửa hàng 33](#_Toc166841749)

[Hình 3.6: Biểu đồ chức năng Mua bom 34](#_Toc166841750)

[Hình 3.7: Biểu đồ chức năng Cài đặt 34](#_Toc166841751)

[Hình 3.8: Biểu đồ chức năng Tạm dừng game 35](#_Toc166841752)

[Hình 3.9: Biểu đồ chức năng Quay lại màn hình chính 35](#_Toc166841753)

[Hình 3.10: Biểu đồ chức năng Chơi lại 35](#_Toc166841754)

[Hình 3.11: Biểu đồ chức năng Chiến đấu 36](#_Toc166841755)

[Hình 3.12: Biểu đồ chức năng chiến đấu của Boss 36](#_Toc166841756)

[Hình 3.13: Biểu đồ trình tự Chọn chế độ chơi 46](#_Toc166841757)

[Hình 3.14: Biểu đồ trình tự Chọn số người chơi 47](#_Toc166841758)

[Hình 3.15: Biểu đồ trình tự Chọn màn chơi 47](#_Toc166841759)

[Hình 3.16: Biểu đồ trình tự Chức năng Cài đặt 48](#_Toc166841760)

[Hình 3.17: Biểu đồ trình tự Quay lại màn hình chính 49](#_Toc166841761)

[Hình 3.18: Biểu đồ trình tự chức năng Chơi lại 49](#_Toc166841762)

[Hình 3.19: Biểu đồ lớp chức năng Chiến đấu của người chơi 50](#_Toc166841763)

[Hình 3.20: Hình ảnh Script lớp BulletSO 51](#_Toc166841764)

[Hình 3.21: Hình ảnh Script lớp BulletSO 51](#_Toc166841765)

[Hình 3.22: Danh sách các BulletSO 52](#_Toc166841766)

[Hình 3.23: Hình ảnh lớp Pref\_Data 52](#_Toc166841767)

[Hình 3.24: Hình ảnh thuộc tính IsBought 53](#_Toc166841768)

[Hình 3.25: Dữ liệu được lưu trực tiếp trong scene 53](#_Toc166841769)

[Hình 3.26: Mô hình hóa các dữ liệu có sẵn trong scene 54](#_Toc166841770)

[Hình 3.27: Mô hình hóa các dữ liệu không có sẵn trong Scene 54](#_Toc166841771)

[Hình 3.28: Biểu đồ Flowchart 55](#_Toc166841772)

[Hình 3.29: Ví dụ hàm gọi theo sự kiện của Input System 56](#_Toc166841774)

[Hình 3.30: Giao diện màn hình chính 57](#_Toc166841775)

[Hình 3.31: Giao diện Cửa hàng 58](#_Toc166841776)

[Hình 3.32: Giao diện Cài đặt 58](#_Toc166841777)

[Hình 3.33: Giao diện Chọn số người chơi 58](#_Toc166841778)

[Hình 3.34: Giao diện Chọn màn chơi 59](#_Toc166841779)

[Hình 3.35: Giao diện Chế độ nhiều người chơi 59](#_Toc166841780)

[Hình 3.36: Giao diện Chế độ một người chơi 60](#_Toc166841781)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1: Người chơi. 17](#_Toc166841782)

[Bảng 3.2: Map 1 và 2. 18](#_Toc166841783)

[Bảng 3.3: Map 3. 19](#_Toc166841784)

[Bảng 3.4: Map 4. 20](#_Toc166841785)

[Bảng 3.5: Map Black Forest Tomb. 21](#_Toc166841786)

[Bảng 3.6: Map Spiritual Forest. 23](#_Toc166841787)

[Bảng 3.7: Map Castle In The Sky. 24](#_Toc166841788)

[Bảng 3.8: Các loại Bom. 25](#_Toc166841789)

[Bảng 3.9: Bảng thiết kế âm thanh. 29](#_Toc166841790)

[Hình 4.1: Bảng Test Plan 61](#_Toc166841791)

**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| Thuật ngữ | Ý nghĩa |
| Gameplay | Gameplay là thuật ngữ được sử dụng trong lĩnh vực trò chơi điện tử để chỉ cách thức và trải nghiệm chơi game |
| Map | Bản đồ, màn chơi của game |
| Player | Người chơi |

# MỞ ĐẦU

## Lí do chọn đề tài

Trong thời đại ngày nay, nơi áp lực công việc và cuộc sống ngày càng lớn, nhu cầu giải trí của con người cũng ngày một lớn hơn. Vì vậy, cùng với phim ảnh và ca nhạc, số lượng người chọn game làm cách để giải trí cũng ngày một tăng lên theo đó, thị trường game cũng ngày càng phát triển mạnh mẽ.

Thuận theo xu thế đó, em quyết định chọn đề tài là: “*Xây dựng game 2D The Puppet’s Revenge*”. Game thuộc thể loại Platformer và Đối kháng, hứa hẹn sẽ đem lại trải nghiệm vui vẻ và giải trí cho người chơi. Game có hai chế độ chính. Chế độ nhiều người chơi sẽ cho phép nhiều người chơi so tài với nhau trên cùng một máy. Và chế độ cốt truyên – một người chơi, nơi người chơi cũng có thể khám phá một mình, chiến đấu chống lại Boss, kiếm vàng để mua các loại vũ khí mới.

Nhân vật chính do người chơi điều khiển được xây dựng trên phong cách nhân vật Ragdoll và StickMan, nhân vật chính sẽ như những con rối được điều khiển. Điều này sẽ giảm sự chính xác đi đôi chút nhưng thay vào đó sẽ giảm sự căng thẳng, tập trung vào sự gây cười của game khi người chơi cố gắng điều khiển những con rối loạng choạng, lắc lư khi tránh né các đòn đánh, trướng ngại khác nhau. Dự án có yếu tố đối kháng nhưng không mang nặng tính thắng thua, thay vào đó tập trung vào việc mang lại niềm vui người chơi.

## Mục tiêu và phạm vi nghiên cứu

Các mục tiêu cần đạt được trong đề tài bao gồm:

* Tìm hiều quy trình phát triển game và vận dụng vào phát triển đề tài.
* Tìm hiểu và sử dụng Unity Engine, Visual Studio... và một số gói hỗ trợ của Unity như Cinemachine và Input System.
* Thực hiện phân tích, thiết kế viết tài liệu phân tích thiết kế game The Puppet’s Revenge.
* Cài đặt và kiểm thử được phần mềm.
* Chia sẻ phần mềm cho mọi người trải nghiệm.

## Ý nghĩa của đề tài

* **Ý nghĩa lý luận đề tài**

Mọi người có thể sử dụng đề tài “*Xây dựng game 2D The Puppet’s Revenge*” như một tài liệu tham khảo về lập trình game trên Unity. Qua đó có thêm nhiều bài học và ví dụ giúp sản sinh ra những ý tưởng và bài học mới.

* **Ý nghĩa thực tiễn của đề tài**

Sản phẩm được hoàn thiện có thể trở thành công cụ giải trí cho nhiều người. Đồng thời, sản phẩm cũng là một bước tiến lớn trên con đường lập trình game của em, giúp em có thêm sự tự tin để phát triển thêm nhiều sản phẩm sau này.

## Cấu trúc báo cáo

* **Chương 1: Lý thuyết thiết kế game**

Trình bày lý thuyết về các bước cơ bản để thiết kế một game

* **Chương 2: Công cụ hỗ trợ xây dựng game**

Giới thiệu các công cụ đã được sử dụng để xây dựng game.

* **Chương 3: Xây dựng game** **2D The Puppet’s Revenge**

Trình bày quá trình thiết kế lối chơi, phân tích thiết kế và xây dựng game 2D The Puppet’s Revenge.

* **Chương 4: Test game 2D The Puppet’s Revenge**

Kiểm thử các chức năng của chương trình.

# CHƯƠNG 1. LÝ THUYẾT THIẾT KẾ GAME

## Một số nguyên tắc cơ bản của thiết kế game

- **Thiết kế trò chơi của dựa trên cơ chế cốt lõi**: Vì đây là điều mà người chơi sẽ lặp lại nhiều lần trong suốt quá trình chơi game. Vì vậy định hình và thiết kế một lỗi chơi cốt lõi là vô cùng quan trọng.

- **Làm cho trò chơi của bạn dễ học, nhưng khó để thành thạo**: Một trò chơi với những cơ thế quá khó chơi và nắm bắt sẽ dễ khiến người chơi chán nản và bỏ cuộc. Thay vào đó, ta nên thiết kế một lối chơi dễ làm quen, dễ hiểu. Và khi người chơi đã hiểu cách chơi, chúng ta sẽ duy trì các thách thức khiến cho người chơi phải vận dụng tối đa khả năng để giành chiến thắng.

- **Đảm bảo sự cân bằng của trò chơi**: Một game quá khó hoặc quá dễ sẽ khiến người chơi mau chóng chán nản. Vì vậy, tìm được một điểm cân bằng cho tiến trình của game là vông cùng quan trọng trong việc duy trì sự hứng thú của người chơi.

- **Phần thưởng**: Phần thưởng cũng đóng một vai trò quan trọng trong việc duy trì sự hứng thú của người chơi với game.

- **Giao tiếp và cung cấp cho người chơi nhiệm vụ rõ ràng**: Cung cấp cho người chơi một mục tiêu để hoàn thành là điều tối quan trọng của một game, và điều này được truyền đạt thông qua sự giao tiếp giữa game và người chơi.

## 1.2 Thiết kế ý tưởng Game

- Giới thiệu: Trình bày về những ý tưởng cơ bản của game.

- Thể loại game: Xác định thể loại của game. Thể loại game đóng vai trò quan trọng trong việc xây dựng lối chơi và giao diện của game.

- Tóm tắt game: Tóm tắt các cơ chế cơ bản của game.

- Khách hàng mục tiêu: Xác định nhóm khách hàng mà game hướng tới. Đối tượng khách hàng mục tiêu có ảnh hưởng quan trọng đến đồ họa, cốt truyện và lối chơi của game.

- Điểm mạnh của game: Xác định các yếu tố nổi bật, hay của game. Có thể là cốt truyện, đồ họa, cách chơi,...

- Phong cách nghệ thuật game: Xác định phong cách nghệ thuật của game, ví dụ như đồ họa pixel, 2D hay 3D,...

- Thiết bị trải nghiệm game: Xác định loại thiết bị dùng để chơi game.

## 1.3 Thiết kế phát triển Game

### 1.3.1. Thiết kế kịch bản game

**+ Cách chơi chính (Core gameplay)**

- Là tập các hành động mà người chơi sẽ thực hiện thường xuyên trong

quá trình trải nghiệm game và không thể thiếu để giành chiến thắng trong trò

chơi.

**+ Cốt chuyện của game**: Các game lớn trên thì trường thường có cốt truyện để người chơi khám phá và nắm bắt, tạo nên sự kích thích. Ví dụ: Elden Ring, God of War,...

**+ Các phần tử của game (Game Elements)**

- Mỗi phần tử game có thể là: player, enemies, objects, ...

- Mỗi phần tử game mô tả: tính chất, hoạt động, ảnh hưởng của nó khi

tương tác với player hoặc đối với các đối tượng khác; hình ảnh đồ họa.

**+ Các cơ chế của game (Game Mechanic)**

- Game Mechanic là các phương thức kết hợp các yếu tố trong game. Game Mechanic xác định trò chơi sẽ diễn ra như thế nào.

### 1.3.2. Thiết kế âm thanh

- Xác định âm nhạc, âm thanh cần dùng cho các tình huống của game

# 

# CHƯƠNG 2. CÔNG CỤ HỖ TRỢ XÂY DỰNG GAME

## 2.1 Giới thiệu về công cụ làm game Unity

### 2.1.1 Tổng quan về Unity



Hình 2.1: Unity là gì?

Unity là một nền tảng phát triển trò chơi đa năng và mạnh mẽ. Được phát triển bởi công ty Unity Technologies, Unity cung cấp một môi trường làm việc tích hợp để tạo ra các trò chơi 2D, 3D và thực tế ảo (VR) hoặc thực tế tăng cường (AR). Điều đặc biệt về Unity là khả năng hỗ trợ cho nhiều nền tảng khác nhau như máy tính, điện thoại di động, máy chơi game và thậm chí cả nền tảng web.

### 2.1.2 Ưu điểm

Unity có nhiều ưu điểm hấp dẫn, đó là lý do tại sao nền tảng này trở thành một công cụ phát triển trò chơi phổ biến:

**Đa năng:** Unity cho phép phát triển trò chơi 2D, 3D, thực tế ảo (VR) và thực tế tăng cường (AR) trên nhiều nền tảng khác nhau, bao gồm cả máy tính, điện thoại di động, máy chơi game và web.

**Dễ học và sử dụng:** Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity khá dễ sử dụng.

**Lập trình đa dạng:** Unity hỗ trợ việc lập trình bằng nhiều ngôn ngữ như C# và JavaScript, cho phép lập trình viên dễ dàng tiếp cận và lập trình.

**Cộng đồng mạnh mẽ:** Unity có một cộng đồng phát triển lớn và tích cực. Người dùng có thể chia sẻ kiến thức, tài liệu và tài nguyên miễn phí, giúp giải quyết vấn đề một cách dễ dàng hơn.

**Các công cụ sáng tạo:** Unity cung cấp nhiều công cụ hỗ trợ sáng tạo như trình chỉnh sửa hình ảnh, âm thanh và môi trường trò chơi.

**Kết xuất chất lượng cao:** Unity cho phép tạo ra đồ họa và hiệu ứng chất lượng cao, giúp trò chơi trở nên hấp dẫn hơn.

**Cross-Platform:** Khả năng phát triển trên nhiều nền tảng giúp đạt tới một lượng người chơi rộng rãi.

**Thư viện tài nguyên:** Unity có cửa hàng tài nguyên cộng đồng và cửa hàng tài nguyên trả phí, giúp tìm kiếm và tích hợp các tài nguyên dễ dàng.

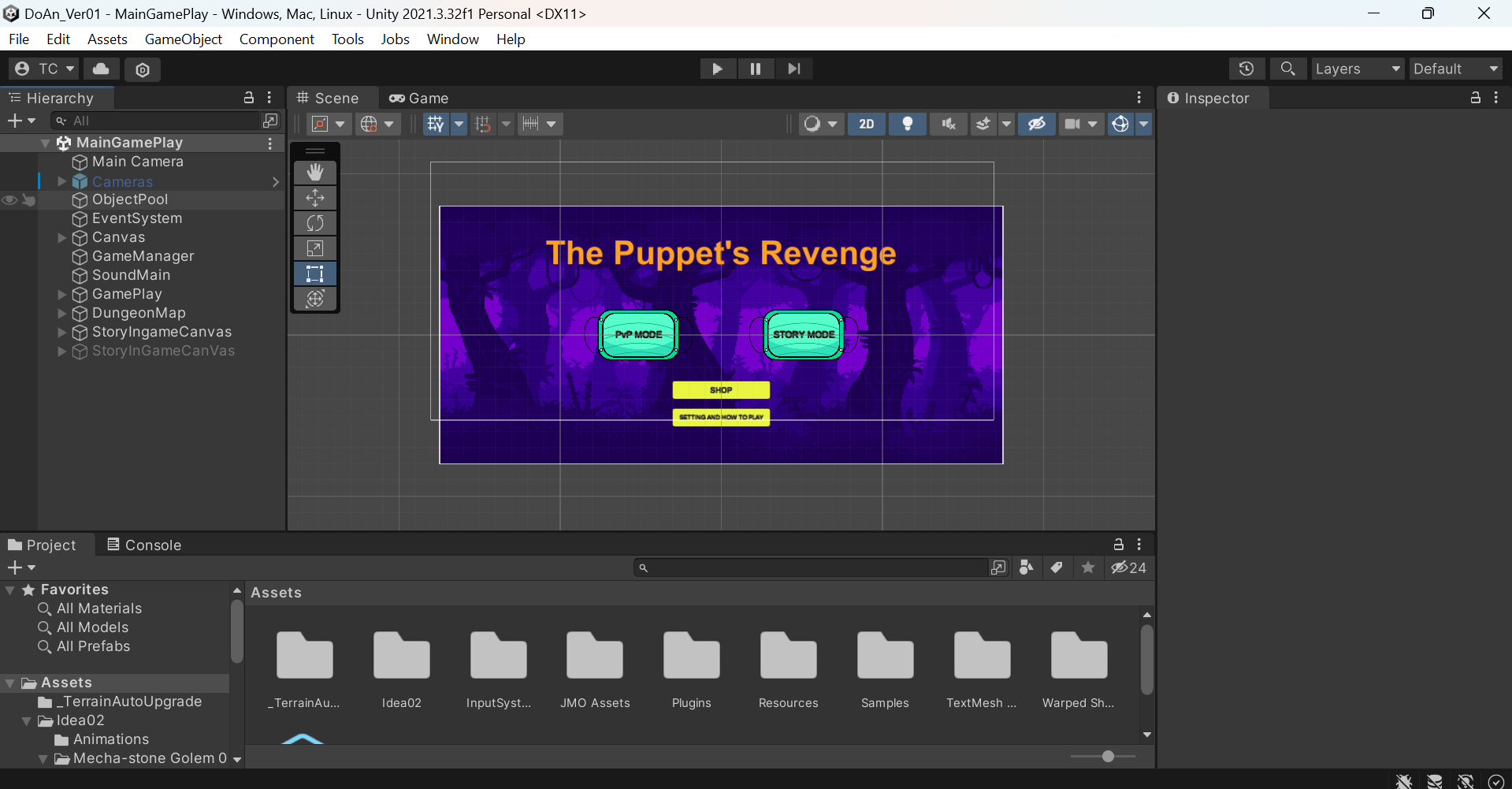
**Khả năng tùy chỉnh:** Unity cho phép tùy chỉnh các chức năng và tích hợp công cụ bên ngoài để đáp ứng nhu cầu cụ thể của dự án.

Tóm lại, Unity là một công cụ mạnh mẽ và linh hoạt, thích hợp cho cả người mới bắt đầu và các nhà phát triển chuyên nghiệp trong việc tạo ra các trò chơi chất lượng cao trên nhiều nền tảng khác nhau.

* + 1. **Các thành phần và khái niệm cơ bản**

**Asset:** là những tài nguyên xây dựng nên một dự án Unity. Từ những tập tin hình ảnh, mô hình 3D đến các tập tin âm thanh. Unity gọi các tập tin mà chúng ta dùng để tạo nên trò chơi là tài sản (Assets).

**Scenes:** là các màn chơi hoặc cảnh quan mà người chơi có thể trải nghiệm. Mỗi scene trong Unity có thể chứa các đối tượng, hình ảnh, âm thanh và các tài nguyên khác liên quan đến trải nghiệm trò chơi cụ thể. Ví dụ:



Hình 2.2: Scene

**Camera** là một Game Object đặc biệt trong Scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game.

**Transform** là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng.

**Game Object** là một thực thể cơ bản trong Unity, có thể là bất cứ thứ gì trong trò chơi như nhân vật, vật phẩm, môi trường, ánh sáng, camera... Game Object có thể có một hoặc nhiều tính năng.

**Components** Components có nhiều hình thức khác nhau. Chúng có thể xác định hành vi, cách xuất hiện,... hay ảnh hưởng đến các khía cạnh khác trong chức năng của Game Object trong trò chơi. Bằng cách “gắn” chúng vào trong Game Object, chúng ta ngay lập tức có thể áp dụng tác động của chúng lên đối tượng. Những Components phổ biến trong quá trình phát triển trò chơi đều được Unity hỗ trợ sẵn. Ví dụ như thành phần Rigidbody đã được đề cập hay các yếu tố đơn giản khác như ánh sáng, Camera và nhiều thành phần khác.

**Scripts** là các đoạn mã viết bằng ngôn ngữ lập trình (thường là C# hoặc JavaScript) được gắn vào các Game Object để thêm hành vi và logic vào trò chơi.

* + Đặc điểm: Scripts là các tập hợp các câu lệnh và chức năng được viết để thực hiện các hành động cụ thể trong trò chơi. Chúng có thể điều khiển chuyển động, tương tác, hiệu ứng và hầu hết các khía cạnh của trò chơi.
  + Kết nối với Game Object: Mỗi Script được gắn liền với một Game Object. Khi Script được gắn vào Game Object, nó có khả năng truy cập và thao tác với các thành phần và thuộc tính của Game Object đó.
  + Chức năng và Phương thức: Trong một Script, bạn có thể định nghĩa các hàm (phương thức) để thực hiện các chức năng cụ thể. Ví dụ: hàm để điều khiển di chuyển của nhân vật, hàm để kiểm tra va chạm.
  + Tương tác: Scripts cho phép tạo ra các tương tác giữa các Game Object và với người chơi. Ví dụ: khi người chơi nhấn nút, Script có thể thực hiện hành động như bắn đạn hoặc mở cửa. Sự kiện và Gọi hàm: Scripts có thể phản hồi vào các sự kiện như nhấn nút, va chạm, thời gian,... Bạn có thể kích hoạt các hàm trong Script dựa trên các sự kiện này.
  + Tùy chỉnh: Scripts cho phép bạn tạo ra các hành vi tùy chỉnh cho các Game Object. Điều này giúp tạo ra tính năng riêng biệt và độc đáo cho trò chơi của bạn.

**Prefabs** Prefabs cho phép chúng ta lưu trữ các đối tượng với những Components và những thiết đặt hoàn chỉnh. Có thể so sánh với khái niệm cơ bản là MovieClip trong Adobe Flash, Prefabs chỉ đơn giản là một Container (một đối tượng chứa) rỗng mà chúng ta có thể đưa bất kì một đối tượng hay dữ liệu mẫu nào mà chúng ta muốn tái sử dụng về sau.

### 2.1.4 Đặc điểm và tính năng

**Rendering:** Unity hỗ trợ đầy đủ khả năng kết xuất hình ảnh (Rendering) cùng nhiều hỗ trợ cho phép áp dụng các công nghệ phổ biến trong lĩnh vực đồ họa 3D nhằm cải thiện chất lượng hình ảnh.

**Lighting:** Ánh sáng là một điều thiết yếu giúp môi trường trở nên đẹp và thực tế hơn. Unity cũng cung cấp nhiều giải pháp đa dạng cho phép chúng ta áp dụng ánh sáng một cách tốt nhất vào môi trường trong trò chơi với nhiều loại nguồn sáng như ánh sáng có hướng (Directional Light), ánh sáng điểm (Point Light), ...

**Terrains** còn gọi chung là địa hình bao gồm phần đất nền của môi trường trong trò chơi cùng các đối tượng gắn liền như cây, cỏ, ... Unity cung cấp một công cụ hỗ trợ rất tốt khả năng này với tên gọi là Terrains Tools cho phép chúng ta thiết kế địa hình với các công cụ vẽ dưới dạng Brush có nhiều thông số tùy chỉnh để tạo hình và lát Texture cho địa hình.

**Substances** có thể hiểu đơn giản là một dạng tùy biến Textures nhằm làm đa dạng chúng trong nhiều điều kiện môi trường khác nhau. Unity cung cấp khả năng này thông qua các API dựng sẵn trong thư viện, hỗ trợ lập trình viên lập trình để tùy biến hình ảnh được kết xuất của Texture

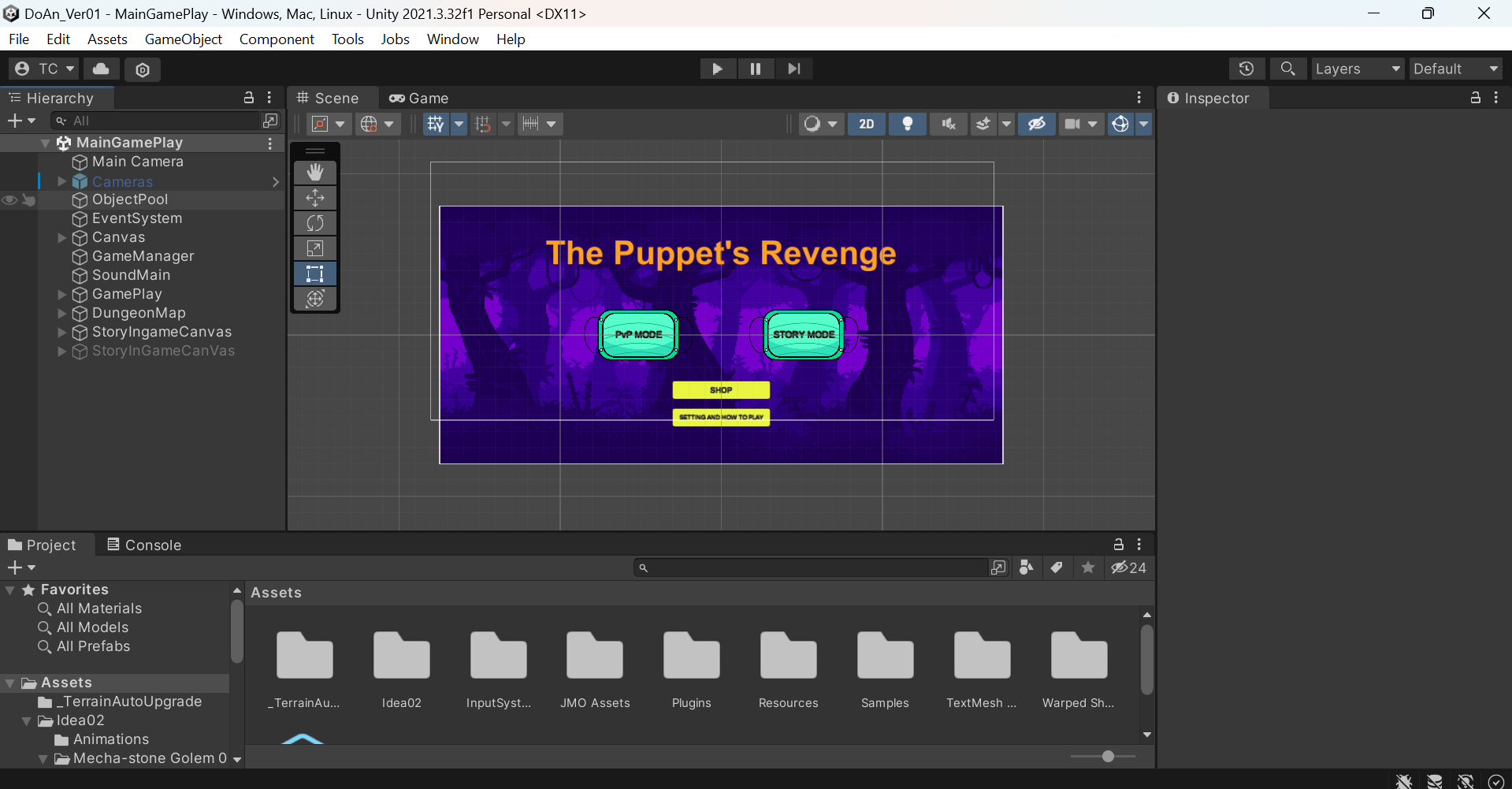
**Physics:** PhysX là một Engine mô phỏng và xử lí vật lý cực kỳ mạnh mẽ được phát triển bởi nhà sản xuất card đồ họa hàng đầu thế giới NVIDIA. Unity đã tích hợp Engine này vào để đảm nhận mọi vấn đề vật lý. Một số vấn đề vật lý được hỗ trợ bởi Unity như: Soft Bodies, Rigid Bodies, Joints, Cars, ...

**Audio:** Về âm thanh, Unity tích hợp FMOD – công cụ âm thanh thuộc hàng mạnh nhất hiện nay. Qua đó Unity hỗ trợ chúng ta nhập và sử dụng nhiều định dạng tập tin âm thanh khác nhau.

**Programming:** Lập trình là một trong những yếu tố quan trọng nhất trong phát triển Game. Lập trình cho phép nhà phát triển tạo nên khả năng tương tác, trí thông minh và yếu tố Gameplay cho trò chơi. Unity cho phép chúng ta lập trình bằng nhiều ngôn ngữ khác nhau như: C#, Javascript.

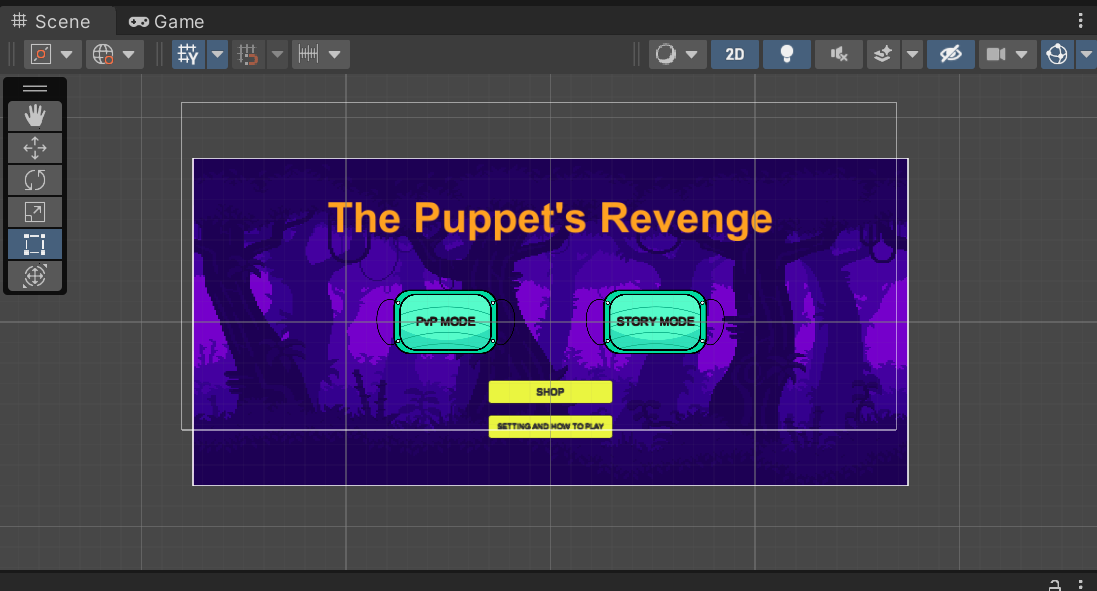
### 2.1.5 Giao diện cơ bản của Unity

Giao diện Unity được thiết kế một cách rõ ràng và dễ sử dụng để giúp bạn dễ dàng quản lý và phát triển trò chơi của mình.



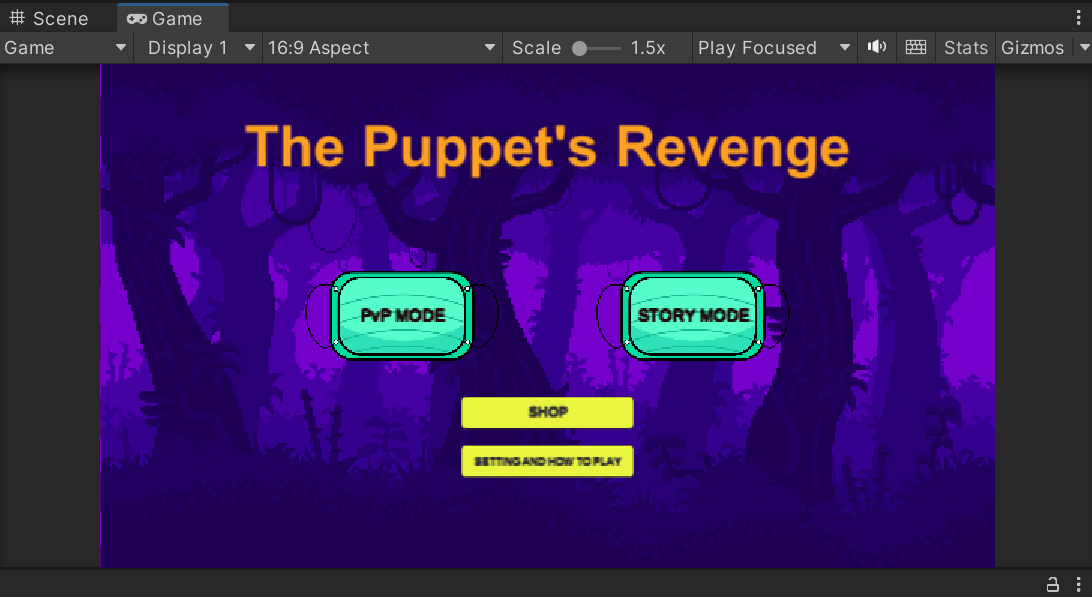
Hình 2.3: Tổng quan giao diện của Unity

**Scene View**: Phần này phần hiển thị các đối tượng trong Scenes một cách trực quan, có thể lựa chọn các đối tượng, kéo thả, phóng to, thu nhỏ, xoay các đối tượng...Phần này có để thiết lập một số thông số như hiển thị ánh sáng, âm anh, cách nhìn 2D hay 3D... Khung nhìn Scene là nơi bố trí các Game Object như cây cối, cảnh quan,enemy, player, camera,... trong game.



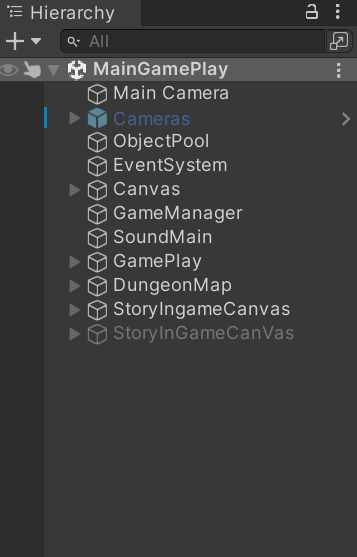
Hình 2.4: Scene View trong Unity

**Game View**: Đây là cửa sổ mà bạn có thể xem trực tiếp trò chơi của mình khi đang phát triển. Điều này giúp bạn kiểm tra trực tiếp hiệu suất và hình ảnh của trò chơi.



Hình 2.5: Game View trong Unity

**Hierarchy**: Hierarchy là nơi hiển thị các Game Object trong Sences hiện hành. Khi các đối tượng được thêm hoặc xóa trong Sences, tương ứng với các đối tượng đó trong cửa sổ Hierarchy. Tương tự trong tab Project, Hierarchy cũng có một thanh tìm kiếm giúp quản lý và thao tác với các Game Object hiệu quả hơn đặc biệt là với các dự án lớn.

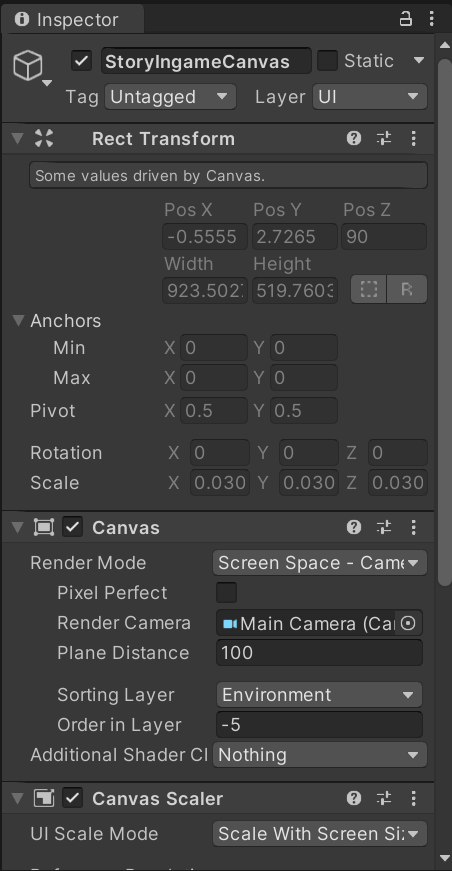


Hình 2.6: Hierarchy trong Unity

**Project**: Trong cửa sổ này, bạn có thể thấy tất cả các tài nguyên của dự án, chẳng hạn như hình ảnh, âm thanh, các tệp mã nguồn, và nhiều thứ khác. Điều này giúp bạn quản lý tất cả các tài nguyên của trò chơi.

**Inspector**: Inspector hiển thị chi tiết các thông tin về Game Object đang làm việc, kể cả những component được đính kèm và thuộc tính của nó.

* + Bạn có thể điều chỉnh, thiết lập mọi thông số và chức năng của Game Object thông qua cửa sổ Inspector.
  + Mọi thuộc tính thể hiện trong Inspector đều có thể dễ dàng tuỳ chỉnh trực tiếp mà không cần thông qua một kịch bản định trước.
  + Các thiết lập của từng component được đặt trong menu. Các bạn có thể click chuột phải, hoặc chọn icon hình bánh răng nhỏ để xuất hiện menu.
  + Ngoài ra Inspector cũng thể hiện mọi thông số Import Setting của asset đang làm việc như hiển thị mã nguồn của Script, các thông số animation, ...



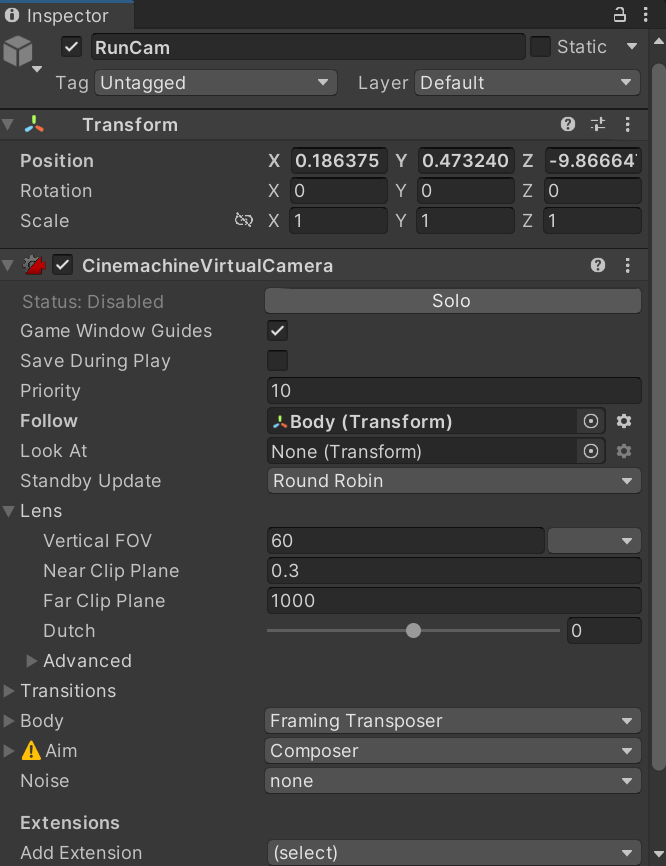
Hình 2.7: Inspector trong Unity

**Toolbar**: Thanh công cụ nằm ở phía trên cùng của giao diện, cung cấp các công cụ và chức năng phổ biến như lưu dự án, phát triển trò chơi, và quản lý các sơ đồ.

**Play Controls**: Bạn có thể nhấn vào các nút Play, Pause, và Stop để thực hiện việc chạy và kiểm tra trò chơi của mình.

### 2.1.6 Cinemachine

Cinemachine là một công cụ của Unity giúp điều khiển Camera đi theo người chơi một cách mượt mà. Trong đó, Cinemachine Camera sẽ đi theo GameObject được chọn trong Follow.



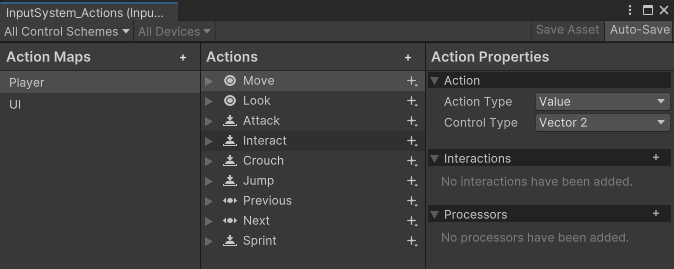
Hình 2.8: Hình ảnh Cinemachine

**Lý do lựa chọn:**

* Đa dạng các chức năng, giúp cho Camera đi theo người chơi một cách tự động, tiết kiệm tài nguyên máy. Cho phép điều khiển các thuộc tính của Camera dễ dàng và trực quan hơn.
* Tạo ra chuyển động Camera mượt mà.

### 2.1.7 Input System

Input Systemlà một hệ thống mới hơn, linh hoạt hơn, cho phép bạn sử dụng bất kỳ loại Thiết bị Đầu vào nào để điều khiển các nội dung trong project của mình. Nó có thể được sử dụng như một sự thay thế cho Input Manager cổ điển của Unity.



Hình 2.9: Hình ảnh Input System

**Lý do lựa chọn:**

* Cung cấp giao diện trực quan cho viêc thêm mới, điều chỉnh cách Game tiếp nhận thông tin truyền vào từ người chơi.
* Hỗ trợ đa dạng thiết bị đầu vào: Keyboard, GamePad,Mouse,JoyStick,...

Hỗ trợ lập trình: các hàm được gọi theo sự kiện thay vì phải liên tục chạy trong hàm Update giúp tiết kiệm tài nguyên máy, làm tăng hiệu suất của game.

## 2.2 Ngôn ngữ lập trình C# và Mono

C# là một ngôn ngữ lập trình đối tượng được phát triển bởi Microsoft. Nó là một trong những ngôn ngữ được sử dụng rộng rãi trong việc phát triển game trên Unity.

Mono là một bộ công cụ mã nguồn mở hỗ trợ việc phát triển ứng dụng chạy trên nhiều nền tảng, bao gồm cả Windows, MacOS và Linux. Mono cho phép các nhà phát triển tạo ra các ứng dụng đa nền tảng một cách dễ dàng.

Mono cung cấp một số thành phần chính để phát triển game trên Unity, bao gồm Mono Runtime, Mono C# Compiler và Mono Class Libraries.

Trong Unity, C# được sử dụng để phát triển các script điều khiển hành vi của các đối tượng trong game, xử lý những sự kiện và tương tác với các thành phần khác nhau của game. Mono cung cấp môi trường chạy các script bằng C# trong Unity, cho phép phát triển game trên đa nền tảng và tích hợp các tính năng như xử lý nhanh và bảo mật.

C# và cung cấp các tính năng và thư viện hỗ trợ mạnh mẽ cho việc phát triển game bao gồm: đa luồng, lập trình hướng đối tượng, thư viện đồ họa và âm thanh, và nhiều tính năng khác.

Tổng quan, C# và Mono là những công nghệ quan trọng và được sử dụng rộng rãi trong việc phát triển game trên Unity.

# CHƯƠNG 3. XÂY DỰNG GAME 2D THE PUPPET’S REVENGE

## 3.1 Thiết kế Game

### 3.1.1 Thiết kế ý tưởng Game

- **Giới thiệu**: Là một game đi cảnh lấy bối cảnh là hành trình báo thù của một phù thủy từng bị ba ác quỷ hãm hại. Game có sự kết hợp của yếu tố hành động, đi cảnh phiêu lưu và đối kháng.

- **Thể loại game**: Đi cảnh (platformer), Hành động (action) và Đối kháng.

- **Tóm tắt game**: Lối chơi của game tập trung vào việc người chơi điều khiển nhân vật tránh né trướng ngại, tìm kiếm lối thoát, đánh Boss và thu thập vàng. Ngoài ra, game còn có chế độ đối kháng cho phép nhiều người chơi đối đầu nhau.

- **Khách hàng mục tiêu**: Từ 10 – 30 tuổi, vì đây là lứa tuổi mà người chơi vẫn giữ được khả năng phản xạ tốt và niềm hứng thú với những cuộc phiêu lưu vẫn còn đó.

- **Điểm mạnh của game**: Lối chơi đa dạng, người chơi có thể chọn đối đầu nhau hoặc đối đầu với Boss, mỗi Boss lại có một Map và cách tấn công riêng cho người chơi khám phá.

- **Phong cách nghệ thuật game**: Đồ họa 2D cartoon kết hợp pixel, nhân vật chính mang phong cách kết hợp giữa Ragdoll và Stickman.

- **Thiết bị trải nghiệm game**: Game được thiết kế cho máy tính hệ điều hành Windows.

### 3.1.2. Thiết kế kịch bản game

**Yêu cầu về thiết kế:**

- Thiết kế trò chơi của dựa trên cơ chế cốt lõi: Người chơi điều khiển nhân vật di chuyển và tránh né các mối nguy hiểm.

- Trò chơi có lối chơi dễ học, nhưng khó để thành thạo.

- Đảm bảo sự cân bằng của trò chơi: Đàm bảo độ khó của Boss và các màn chơi. Đảm bảo cân bằng sức mạnh của các loại vũ khí.

- Phần thưởng: Có hệ thống tăng vàng khi người chơi đạt được thành tựu và cửa hàng để nâng cấp vũ khí.

- Giao tiếp và cung cấp cho người chơi nhiệm vụ rõ ràng: Có hệ thống giao diện hướng dẫn cho người chơi.

**Cách chơi chính (Core gameplay)**

- Người chơi sử dụng bộ phím AWSD để điều hướng di chuyển cũng như hướng tay của nhân vật. Phím F để ném bom. Nhấn giữ G để phóng dây vào tường, giúp cho nhân vật có thể leo lên địa hình cao hơn và di chuyển đa dạng hơn trong chiến đấu. Game có 2 cách chơi cho người chơi lựa chọn.

- **Cách chơi thứ nhất** (PvP Mode): Nhiều người chơi trên cùng một máy, sẽ có tối đa 3 người chơi đối đầu lẫn nhau.

+ Mỗi người chơi sẽ có màu sắc riêng gồm: Xanh nước biển, Xanh lá và Đỏ. Mỗi màu sẽ có các phím điều khiển tương ứng với các nút chức năng cơ bản:

+ Xanh nước biển: “AWSD” và “F”,”G”.

+ Đỏ: “JIKL” và “P”, “[“.

+ Xanh lá cây: 4 phím điều hướng và “1”, “2”.

Các người chơi sẽ cố gắng tấn công lẫn nhau, người chơi sống sót cuối cùng sẽ chiến thắng.

+ Cao trào của chế độ này đến từ sự đối đầu của nhiều người chơi và các Map chơi biến đổi liên tục.

- **Cách chơi thứ hai** (Story Mode): Khám phá Map và đấu Boss. Người chơi sẽ sử dụng nhân vật cơ bản màu xanh nước biển và các phím cơ bản để điều khiển. Đây là chế độ chơi đơn. Trong chế độ này, người chơi sẽ khám phá môi trường xung quanh, tránh trướng ngại vật, thu thập vàng, tìm kiếm lối ra ở cuối Map. Ngoài ra người chơi cũng có thể lựa chọn hạ gục Boss. Màn chơi sẽ kết thúc khi gạ gục Boss và bạn sẽ được cộng thêm 50 vàng mà ko cần thu thập.

+ Cao trào của chế độ này xuất hiện khi người chơi chạm mặt Boss và bị tấn công, truy đuổi.

Có 3 Map tương đương với 3 Boss lần lượt là:

+ Dark Forest Tomb: Boss tần công tầm gần, tốc độ di chuyển cao, tầm phát hiện người chơi trung bình. Kích thước lớn, dễ nhắm bắn.

+ Spiritual Forest: Boss tần công tầm gần, tốc độ di chuyển trung bình, tầm phát hiện người chơi trung bình. Kích thước nhỏ, khó nhắm bắn.

+ Castle in The Sky: Boss tần công tầm xa, tốc độ di chuyển thấp, tầm phát hiện người chơi xa. Kích thước trung bình, dễ nhắm bắn.

**Cốt chuyện của game**: Game là hành trình báo thù các con quỷ đã lấy đi thân xác vật lý của một phù thủy. Giờ đây anh ta đã trở lại trong hình hài của một con rối, thể rằng sẽ báo thù những con quỷ đã hãm hại mình.

**Các phần tử của game (Game Elements)**

**- PvP Mode:**

+ Player: Được tạo nên từ các bộ phận rời rạc rồi ráp nối với nhau bằng Hinge Joint 2D của Unity.

Bảng 3.1: Người chơi.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Người chơi 1 |  | Ném bom, đu dây, di chuyển trái phải  Gồm thanh máu màu đỏ mà thanh năng lượng màu xanh.  Có thể ném bom khi thanh năng lượng đầy  Có 50 máu | Va chạm với các trướng ngại vật xung quanh, Chịu sát thương khi bom nổ, Va chạm với Bom |
| Người chơi 2 |  | Ném bom, đu dây, di chuyển trái phải  Gồm thanh máu màu đỏ mà thanh năng lượng màu xanh.  Có thể ném bom khi thanh năng lượng đầy  Có 50 máu | Va chạm với các trướng ngại vật xung quanh, Chịu sát thương khi bom nổ, Va chạm với Bom |
| Người chơi 3 |  | Ném bom, đu dây, di chuyển trái phải  Gồm thanh máu màu đỏ mà thanh năng lượng màu xanh.  Có thể ném bom khi thanh năng lượng đầy  Có 50 máu | Va chạm với các trướng ngại vật xung quanh, Chịu sát thương khi bom nổ, Va chạm với Bom |

+ Map 1 và Map 2: Hai Map này chỉ khác nhau về cách sắp xếp.

Bảng 3.2: Map 1 và 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Trần của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Nền của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Hai bên tường |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Vật cản hình chữ nhật |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Vật cản hình tròn |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |

+ Map 3:

Bảng 3.3: Map 3.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Trần của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Nền của Map |  | Đứng yên  Nằm ngoài tầm nhìn của người chơi, người chơi coi như đã rơi xuống vực | Người chơi nào va chạm với nền sẽ thua ngay lập tức |
| Hai bên tường |  | Đứng yên | Người chơi nào va chạm sẽ thua ngay lập tức |
| Các khối đá hình chữ nhật |  | Di chuyển lần lượt trái phải | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào  Đổi hưỡng di chuyển khi va chạm với 2 bên tường |

+ Map 4:

Bảng 3.4: Map 4.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Trần của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Nền của Map |  | Đứng yên  Nằm ngoài tầm nhìn của người chơi, người chơi coi như đã rơi xuống vực | Người chơi nào va chạm với nền sẽ thua ngay lập tức |
| Hai bên tường |  | Đứng yên | Người chơi nào va chạm sẽ thua ngay lập tức |
| Các khối đá hình chữ nhật |  | Đứng yên  Xoay quanh tâm | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào  Xoay khi bị người chơi đu vào bằng dây móc hoặc đứng lên. |

**- Story Mode**: Chế độ chơi đơn, sẽ có 3 Map với 3 Boss khác nhau cho người chơi khám phá.

**+ Black Forest Tomb**:

Bảng 3.5: Map Black Forest Tomb.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Tường bao của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Cưa |  | Đứng yên | Người chơi nếu va chạm với thành phần này sẽ bị mất máu |
| Vàng |  | Đứng yên | Người chơi nào va chạm vào sẽ được tăng vàng  Vàng có thể được dùng để mua vũ khí mới. |
| Boss |  | Di chuyển từ đầu đến cuối Map  Đuổi theo và tần công nếu người chơi ở trong tầm | Phát hiện ra người chơi trong một khoảng cách nhất định và đuổi theo ( 12m )  Quay lại lối đi cũ khi người chơi ra khỏi tầm phát hiện và hồi máu đã mất theo thời gian  Tấn công tầm gần, tốc độ di chuyển cao, kích thước lớn – dễ bắn trúng  Làm người chơi mất máu khi chém trúng, dùng vũ khí liềm chém  Kích hoạt chức năng chiến thắng khi người chơi thành công đưa máu Boss về 0. |
| Lối thoát |  | Đứng yên | Người chơi chạm vào sẽ được coi là hạy trốn thành công. |

**+ Spiritual Forest**:

Bảng 3.6: Map Spiritual Forest.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Tường bao của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Cưa |  | Đứng yên | Người chơi nếu va chạm với thành phần này sẽ bị mất máu |
| Vàng |  | Đứng yên | Người chơi nào va chạm vào sẽ được tăng vàng  Vàng có thể được dùng để mua vũ khí mới. |
| Boss |  | Di chuyển từ đầu đến cuối Map  Đuổi theo và tần công nếu người chơi ở trong tầm | Phát hiện ra người chơi trong một khoảng cách nhất định và đuổi theo ( 10m )  Quay lại lối đi cũ khi người chơi ra khỏi tầm phát hiện và hồi máu đã mất theo thời gian  Tấn công tầm gần, tốc độ di chuyển cao, kích thước lớn – dễ bắn trúng  Làm người chơi mất máu khi chém trúng, dùng vũ khí liềm chém  Kích hoạt chức năng chiến thắng khi người chơi thành công đưa máu Boss về 0. |
| Lối thoát |  | Đứng yên | Người chơi chạm vào sẽ được coi là hạy trốn thành công. |

**+ Castle In The Sky**:

Bảng 3.7: Map Castle In The Sky.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Tường bao của Map |  | Đứng yên | Cho phép người chơi bắn dây vào và đu lên  Cản trở người chơi  Bom sau khi kích hoạt sẽ nảy lại khi chạm vào |
| Vàng |  | Đứng yên | Người chơi nào va chạm vào sẽ được tăng vàng  Vàng có thể được dùng để mua vũ khí mới. |
| Boss |  | Di chuyển từ đầu đến cuối Map  Đuổi theo và tần công nếu người chơi ở trong tầm | Phát hiện ra người chơi trong một khoảng cách nhất định và đuổi theo ( 25m )  Quay lại lối đi cũ khi người chơi ra khỏi tầm phát hiện và hồi máu đã mất theo thời gian  Tấn công tầm xa, tốc độ di chuyển chậm  Làm người chơi mất máu khi bắn trúng, vũ khí là tia nhiệt.  Kích hoạt chức năng chiến thắng khi người chơi thành công đưa máu Boss về 0. |
| Lối thoát |  | Đứng yên | Người chơi chạm vào sẽ được coi là hạy trốn thành công. |

**- Các loại Bom:**

Bảng 3.8: Các loại Bom.

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đối tượng | Hình ảnh | Hoạt động | Tương tác |
| Normal Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(0.2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |
| Fire Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(0.2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |
| Light Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(0.2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |
| Electric Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |
| Blood Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |
| Poison Skull Bomb |  | Bay theo đường thẳng  To lên sau khi rời khỏi tay người chơi, phát nổ sau một thời gian (10s)  Dính vào người chơi hoặc Boss khi chạm vào và phát nổ sau một thời gian(2s)  Nảy lại khi chạm phải tường | Dính vào người chơi hoặc Boss khi va chạm  Phát nổ sau một thời gian gây sát thương cho người chơi hoặc Boss dính phải |

**+ Các cơ chế của game (Game Mechanic)**

- Game có 2 cách chơi chính: Nhiều người chơi đối đầu nhau và một người chơi khám phá Map và đấu với Boss(Máy).

- **Chế độ chơi đối kháng với người**:Tối đa 3 người. Các người chơi sẽ cố gắng ném bom về phía nhau nhằm gây sát thương cho đối phương. Ai bị đưa máu về 0 sẽ bị loại và biến mất. Người sống sót cuối cùng sẽ giành chiến thắng. Các quả bom sau khi được ném ra khỏi tay khi được kích hoạt sau 1 khoảng thời gian ngắn, mỗi quả bom sẽ dính vào người chơi nếu chạm phải và phát nổ. Mỗi loại bom có thời gian nổ khác nhau. Ngoài ra, các Map sẽ có tính chất khác nhau.

- **Chế độ chơi với máy**: 1 người chơi. Người chơi sẽ điều khiển nhân vật di chuyển khám phá Map được chọn, thu thập vàng đồng thời tránh né các cạm bẫy và sự truy đuổi của Boss. Màn chơi sẽ kết thúc trong 3 trường hợp, mỗi trường hợp sẽ có thông báo khác nhau cho người chơi:

+ Người chơi hết máu: Bị xử thua.

+ Người chơi tìm thấy cửa thoát và chạm vào nó: Trốn thoát thành công.

+ Hạ gục Boss: Chiến thắng.

Boss sẽ lần lượt di chuyển xung quanh Map. Tuy nhiên hắn sẽ đổi mục tiêu truy đuổi sang người chơi khi anh ta ở trong tầm phát hiện. Hắn sẽ quay lại tuần tra sau khi người chơi thoát ra khỏi tầm phát hiện của hắn. Mỗi Boss sẽ có tầm phát hiện khác nhau, người chơi cần phải để ý và di chuyển khéo léo để thoát khỏi hoặc tấn công hắn. Mỗi Map sẽ chỉ có một Boss.

**- Các chức năng khác**:

+ Người chơi sẽ có những chức năng cơ bản như: Điều hướng tay, di chuyển, đu dây, ném bom không thay đổi giữa 2 cách chơi. Sự khác biệt của 2 cách chơi nằm ở mục tiêu để giành chiến thắng. Game sẽ không có cơ chế nhảy mà thay vào đó sẽ là đu dây, tạo sự mới lạ cho lối chơi của game.

+ Vàng kiếm được từ Chế độ chơi với máy sẽ được tích lũy.

+ Tính năng cửa hàng: Người chơi có thể mua bom mới và trang bị nó từ của hàng bằng vàng kiếm được.

### 3.1.3. Thiết kế âm thanh

Bảng 3.9: Bảng thiết kế âm thanh.

|  |  |
| --- | --- |
| Tên | Mô tả |
| Juhani Junkala [Retro Game Music Pack] Ending | Âm nhạc nền của game khi ở Menu |
| creepy-devil-dance-166764 | Âm nhạc nền khi chơi game |
| Âm thanh nổ của Normal Bomb | Âm thanh phát ra khi bom nổ |
| Âm thanh nổ của Fire Bomb | Âm thanh phát ra khi bom nổ |
| Âm thanh nổ của Light Bomb | Âm thanh phát ra khi bom nổ |
| Âm thanh nổ của Electric Bomb | Âm thanh phát ra khi bom nổ |
| Âm thanh nổ của Blood Bomb | Âm thanh phát ra khi bom nổ |
| Âm thanh nổ của Poison Skull Bomb | Âm thanh phát ra khi bom nổ |
| Tiếng lưỡi hái chém của Boss | Âm thanh phát ra khi Boss chém lưỡi hái |
| Tiếng tia nhiệt của Boss | Âm thanh phát ra khi Boss bắn tia nhiệt |
| mixkit-bonus-earned-in-video-game-2058 | Âm thanh khi thu thập được 1 đồng vàng |

## 3.2 Phân tích thiết kế hệ thống

### 3.2.1 Giới thiệu các chức năng trong game

**Các chức năng chính trong game:**

* Chọn chế độ chơi (CheDoChoi), Chọn số người chơi (SoNguoiChoi), Chơi lại(PlayAgain), Chọn màn chơi(ManChoi), Cửa hàng trong trò chơi(Cuahang), Phần thưởng(Thuong), Mua bom(MuaBom), Cài đăt(CaiDat), Tạm dừng game(Pause), Quay lại(Back), Chiến đấu(ChienDau).



Hình 3.1: Biểu đồ Use Case tổng quát

**3.2.1.1 Chức năng chọn chế độ chơi**

Khi ở màn hình chính, sẽ có 2 chế độ chơi cho người chơi lựa chọn. Tùy vào chế độ được chọn mà người chơi sẽ có những trải nghiệm và chức năng khác nhau.



Hình 3.2: Biểu đồ chức năng chọn chế độ chơi

**3.2.1.2 Chức năng chọn số lượng người chơi**

Sau khi chọn chế độ nhiều người chơi, sẽ có 2 lựa chọn số lượng người chơi là 2 người và 3 người. Sau khi chọn số lượng người, hệ thống sẽ hiển thị các nút điều khiển tương ứng của 2 người chơi. Khi người chơi chọn chơi game, sẽ có số lượng nhân vật tương ứng với số lượng được chọn xuất hiện trong game.

****

Hình 3.3: Biểu đồ chức năng chọn số lượng người chơi

**3.2.1.3 Chức năng chọn Màn chơi**

Sau khi chọn chế độ một người chơi, sẽ xuât hiện 3 màn chơi cho người chơi lựa chọn. Khi người chơi lựa chọn màn chơi nào thì trò chơi sẽ chạy màn chơi đó cho người chơi trải nghiệm.

****

Hình 3.4: Biểu đồ chức năng chọn Màn chơi

**3.2.1.4 Chức năng Cửa hàng**

Sau khi chọn vào nút cửa hàng ở màn hình chính, Cửa hàng sẽ xuất hiện cùng với danh sách các loại Bom trong đó. Người chơi có thể bấm vào từng loại Bom để xem giá mua và thuộc tính của nó. Người chơi có thể thay đổi loại Bom đang sử dụng bằng cách mua và kích nút “Equip” để trang bị loại Bom mới.

****

Hình 3.5: Biểu đồ chức năng Cửa hàng

**3.2.1.5 Chức năng Mua bom**

Sau khi chọn chế độ Cửa hàng, người chơi có thể chọn loại Bom mới và mua bằng cách click chọn nút Buy nếu như lượng vàng hiện tại của người chơi lớn hơn giá của Bom. Sau đó, lượng vàng hiện có của người chơi sẽ bị trừ đi bằng với giá của Bom. Bom sẽ được coi là đã được mua và có thể được trang bị bởi người chơi khi kích nút Equip.

****

Hình 3.6: Biểu đồ chức năng Mua bom

**3.2.1.6 Chức năng Cài đặt**

Sau khi chọn nút Setting(Cài đăt) ở màn hình chính, giao diện cài đặt xuất hiện. Người chơi có thể điều chỉnh âm lượng của game hoặc xóa toàn bộ tiến trình của game, đưa game về như lúc mới tải nếu như có nhu cầu trải nghiệm lại game.

****

Hình 3.7: Biểu đồ chức năng Cài đặt

**3.2.1.7 Chức năng Tạm dừng**

Khi ở trong màn chơi, người chơi có thể kích nút Tạm dừng để tạm dừng màn chơi và giao diện tạm dừng xuất hiện. Ở đây, người chơi có thể điều chỉnh âm lượng của game, chọn tiếp tục màn chơi hoặc thoát ra màn hình chính.

****

Hình 3.8: Biểu đồ chức năng Tạm dừng game

**3.2.1.8 Chức năng Quay lại màn hình chính**

Khi ở trong giao diện Chọn số lượng người chơi hoặc Chọn màn chơi, người chơi có thể kích nút Back để thực hiện chức năng Quay lại màn hình chính, đưa người chơi về màn hình chính.

****

Hình 3.9: Biểu đồ chức năng Quay lại màn hình chính

**3.2.1.9 Chức năng Chơi lại**

Khi kết thúc một màn chơi ở chế độ Nhiều người chơi, người chơi sẽ có lựa chọn quay lại màn hình chính hoặc tiếp tục một trận đấu mới.

****

Hình 3.10: Biểu đồ chức năng Chơi lại

**3.2.1.10 Chức năng Chiến đấu**

Cho phép người dùng điều khiển nhân vật của mình chiến đấu.

****

Hình 3.11: Biểu đồ chức năng Chiến đấu

**3.2.1.11 Chức năng chiến đấu của Boss**

Cho phép Boss di chuyển và tự động tấn công người chơi.

****

Hình 3.12: Biểu đồ chức năng chiến đấu của Boss

### 3.2.2 Đặc tả các chức năng trong game

**3.2.2.1 Chức năng chọn chế độ chơi**

**Mô tả vắn tắt:** Use case này cho phép người dùng chọn giữa 2 chế độ chơi của game.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng mở màn hình chính của game. Hệ thống sẽ hiển thị 2 nút bấm đại diện cho 2 chế độ chơi để cho người dùng chọn.

2. Người dùng chọn nút bấm có tên tương ứng với chế độ chơi mình mong muốn. Hệ thống sẽ dựa vào thông tin của nút bấm được chọn để đưa người dùng đến màn hình chuẩn bị tương ứng của từng chế độ chơi. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 2 trong luồng cơ bản nếu người dùng tắt trò chơi thì Use case kết thúc.

2. Tại bước 2 trong luồng cơ bản nếu người dùng chọn chế độ nhiều người chơi (PvPMode) thì hệ thống sẽ đưa người chơi đến màn hình chọn số lượng người chơi. Use case kết thúc.

3. Tại bước 2 trong luồng cơ bản nếu người dùng chọn chế độ một người chơi (StoryMode) thì hệ thống sẽ đưa người chơi đến màn hình chọn màn chơi. Use case kết thúc.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.2 Chức năng chọn số lượng người chơi**

**Mô tả vắn tắt:** Use case này cho phép người dùng chọn số lượng người chơi trong chế độ nhiều người chơi.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng chọn chơi chế độ nhiều người chơi. Hệ thống sẽ hiển thị hai nút bấm thể hiện cho số lượng người chơi là 2 người hoặc 3 người cho người chơi lựa chọn.

2. Sau khi chọn số lượng người chơi, người dùng kích nút Play để bắt đầu màn chơi. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1, 2 trong luồng cơ bản nếu người dùng nếu người dùng chọn quay lại màn hình chính thì hệ thống sẽ đưa người dùng về màn hình chính. Use case kết thúc.

2. Tại bước 2 trong luồng cơ bản nếu người dùng chọn 2 người chơi thì sẽ có 2 người chơi xuất hiện trong màn chơi. Nếu người dùng chọn 3 người chơi thì sẽ có 3 người chơi xuất hiện trong màn chơi.

Các yêu cầu đặc biệt: Nếu người dùng không chọn số lượng người chơi thì sẽ không thể chơi game

Tiền điều kiện: Phải chọn chế độ nhiều người chơi.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.3 Chức năng chọn Màn chơi**

**Mô tả vắn tắt:** Use case này cho phép người dùng chọn Màn chơi trong chế độ chơi một người.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng chọn chơi chế độ một người chơi. Hệ thống sẽ hiển thị giao diện chọn màn cho người chơi lựa chọn gồm danh sách các nút có tên tương ứng với các màn chơi.

2. Người dùng chọn nút có tên tương ứng với màn chơi mình muốn, hệ thống sẽ đưa người dùng đến màn chơi được chọn. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1, 2 trong luồng cơ bản nếu người dùng nếu người dùng chọn quay lại màn hình chính thì hệ thống sẽ đưa người dùng về màn hình chính. Use case kết thúc.

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Phải chọn chế độ một người chơi.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.4 Chức năng Cửa hàng**

**Mô tả vắn tắt:** Use case này cho phép người dùng mở cửa hàng và xem các loại vũ khí trong đó.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng chọn nút Shop ở màn hình chính. Hệ thống sẽ hiển thị giao diện cửa hàng và danh sách các loại vũ khí trong đó cho người chơi.

2. Người dùng bấm chọn vào từng loại Bom để xem thông số của nó. Các loại Bom đã được mua thì sẽ hiển thị lựa chọn trang bị (nút Equip) còn nếu chưa mua thì sẽ là (Buy).

3. Người dùng có thể mua loại bom mới hoặc trang bị cho mình một trong số các loại bom đã mua.

* Luồng rẽ nhánh:

1. Tại bước 1, 2 và 3 trong luồng cơ bản nếu người dùng nếu người dùng chọn quay lại màn hình chính thì hệ thống sẽ đưa người dùng về màn hình chính. Use case kết thúc.

2. Tại bước 3 trong luồng cơ bản, người dùng chọn trang bị loại bom bằng cách kích nút Equip bên phần thông số của nó. Hệ thống sẽ sử dụng loại Bom đó cho cả hai chế độ chơi.

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.5 Chức năng Trao thưởng**

**Mô tả vắn tắt:** Use case này giúp người dùng gia tăng lượng vàng khi thu thập vàng hoặc đánh bại Boss.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người chơi tương tác với gameobject vàng trong màn chơi hoặc khi đánh bại Boss.

2. Khi người chơi thu thập vàng, hệ thống sẽ hiển thị thông báo và tăng lượng vàng người chơi đang có, hệ thống cập nhật lại lượng vàng của người chơi trong cơ sở dữ liệu. Use case kết thúc.

3. Nếu người chơi hạ gục Boss, thông báo phần thưởng và thông báo chiến thắng cùng hiển thị, người chơi tắt thông báo. Use case kết thúc.

* Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Phải chọn chế độ một người chơi.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.6 Chức năng Mua Bom**

**Mô tả vắn tắt:** Use case này cho phép người dùng mua Bom trong cửa hàng của trò chơi.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dung kích chọn mua Bom ở trong cửa hàng. Hệ thống sẽ trừ đi lượng vàng của người chơi tương đương với giá của loại Bom.

2. Hệ thống cập nhật trạng thái của Bom là đã mua. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1 trong luồng cơ bản nếu người dùng nếu lượng vàng của người chơi nhỏ hơn giá của loại Bom thì Use case kết thúc.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.7 Chức năng Cài đặt**

**Mô tả vắn tắt:** Use case này cho phép người dùng điều chỉnh cài đặt của trò chơi.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dung kích chọn chức năng Cài đặt ở màn hình chính. Giao diện cài đặt sẽ xuất hiện.

2. Người dùng có thể điều chỉnh âm lượng của game.

3. Người dùng có thể chọn chức năng xóa hết tiến trình của game để chơi lại từ đầu, hệ thống sẽ hiển thị liệu người chơi có chắc chắn, nếu người chơi đồng ý, hệ thống sẽ xóa hết tiến trình của game.

4. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1 trong luồng cơ bản nếu người dùng chọn quay lại màn hình chính thì Use case kết thúc.

2. Tại bước 2 trong luồng cơ bản nếu người dùng chọn quay lại màn hình chính thì Use case kết thúc.

3. Tại bước 3 trong luồng cơ bản nếu người dùng chọn quay lại màn hình chính thì Use case kết thúc.

4. Tại bước 3 trong luồng cơ bản nếu người dùng không đồng ý thì tiến trình game sẽ không bị xóa.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.8 Chức năng Tạm dừng**

**Mô tả vắn tắt:** Use case này cho phép người dùng tạm dừng màn chơi để có những điều chỉnh.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dung kích chọn chức năng Tạm dừng khi ở trong màn chơi. Giao diện tạm dừng sẽ xuất hiện.

2. Người dùng có thể điều chỉnh âm lượng của game.

3. Người dùng có thể chọn tiếp tục màn chơi. Use case kết thúc.

4. Người dùng có thể chọn thoát màn chơi. Use case kết thúc.

* Luồng rẽ nhánh: Không có.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.9 Chức năng Quay lại màn hình chính**

**Mô tả vắn tắt:** Use case này cho phép người dùng quay lại màn hình chính

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dung kích chọn nút Back, chức năng quay về màn hình chính được kích hoạt và đưa người dùng về lại màn hình chính của game.

* Luồng rẽ nhánh: Không có.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.10 Chức năng Chơi lại**

**Mô tả vắn tắt:** Use case này cho phép người dùng lựa chọn chơi lại ván đấu hoặc thoát ra màn hình chính khi chơi chế độ nhiều người chơi.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi màn chơi kết thúc, giao diện kết thúc xuất hiện.

2. Nếu người chơi chọn chơi lại (kích nút PlayAgain). Hệ thống sẽ cho phép người chơi tiếp tục chơi với số lượng người chơi đã chọn từ trước mà không phải thoát ra màn hình chính.

3. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1 trong luồng cơ bản nếu người chơi chọn chức năng Thoát màn chơi bằng cách chọn nút Quit, hệ thống sẽ thoát màn chơi về màn hình chính.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Chỉ có khi chơi chế độ nhiều người chơi.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.11 Chức năng Chiến đấu**

**Mô tả vắn tắt:** Use case này cho phép người dung chiến đấu và trải nghiêm trò chơi.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người chơi chơi một trong hai chế độ chơi. Hệ thống sẽ sinh ra Màn chơi tùy thuộc vào chế độ chơi được chọn

2. Người chơi sẽ sử dụng các phím được quy định để cố gắng điều khiển nhân vật vừa tấn công kẻ địch vừa tránh né các đòn tấn công và trướng ngại vật.

3. Người chơi nhận sát thương khi bị dính đòn tất công và mất máu. Khi máu của người chơi về 0, nhân vật của họ sẽ bị xử thua và biến mất.

4. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 3 trong luồng cơ bản nếu người chơi chọn chế độ Nhiều người chơi thì người chơi bị xử thua sẽ biến mất và những người chơi còn lại sẽ tiếp tục chơi đến khi chỉ còn một người chơi. Use case kết thúc.

2. Tại bước 3 trong luồng cơ bản nếu người chơi chọn chế độ Một người chơi thì khi người chơi bị xử thua sẽ biến mất và hệ thống sẽ hiện thông báo thua cuộc. Use case kết thúc.

3. Tại bước 2 trong luồng cơ bản nếu người chơi chọn chế độ Một người chơi thì khi người chơi chạm vào gameobject lối thoát hiểm trong màn chơi thì hệ thống sẽ hiện thông báo trốn thoát. Use case kết thúc.

Các yêu cầu đặc biệt: Có sự khác biệt về luật chơi giữa 2 chế độ chơi.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

**3.2.2.12 Chức năng Chiến đấu của Boss**

**Mô tả vắn tắt:** Use case này cho phép Boss di chuyển và chiến đấu.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người chơi chế độ Một người chơi. Hệ thống sẽ sinh ra Boss tùy thuộc vào Màn chơi được chọn.

2. Boss sẽ di chuyển lần lượt giữa các điểm được đặt sẵn từ đầu Map đến cuối Map. Đối hướng đuổi theo người chơi khi ở trong tầm phát hiện của Boss. Chuyển sang tấn công khi người chơi ở trong tầm tấn công.

3. Boss nhận sát thương khi bị dính đòn tất công và mất máu. Khi máu về 0, Boss sẽ biến mất và hệ thống sẽ hiển thị thông báo chiến thắng cho người chơi và kích hoạt chức năng Trao thưởng.

4. Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 1 trong luồng cơ bản nếu người chơi chọn thoát màn chơi thì Use case kết thúc.

2. Tại bước 2 trong luồng cơ bản nếu người chơi rời khỏi tầm phát hiện thì Boss sẽ quay lại lộ trình đã được cài sẵn.

3. Tại bước 3 trong luồng cơ bản nếu người chơi rời khỏi tầm phát hiện thì Boss sẽ hồi máu theo thời gian.

4. Tại bước 2 trong luồng cơ bản nếu người chơi rời khỏi tầm tấn công nhưng vẫn trong tầm phát hiện thì Boss sẽ đuổi theo nhưng không tấn công.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Chỉ có khi chơi chế độ Một người chơi.

Hậu điều kiện: Không có.  
Điểm mở rộng: Không có.

### 3.2.3 Biểu đồ trình tự và biểu đồ lớp

**Biểu đồ trình tự:**



Hình 3.13: Biểu đồ trình tự Chọn chế độ chơi



Hình 3.14: Biểu đồ trình tự Chọn số người chơi



Hình 3.15: Biểu đồ trình tự Chọn màn chơi



Hình 3.16: Biểu đồ trình tự Chức năng Cài đặt



Hình 3.17: Biểu đồ trình tự Quay lại màn hình chính



Hình 3.18: Biểu đồ trình tự chức năng Chơi lại

Biểu đồ lớp:



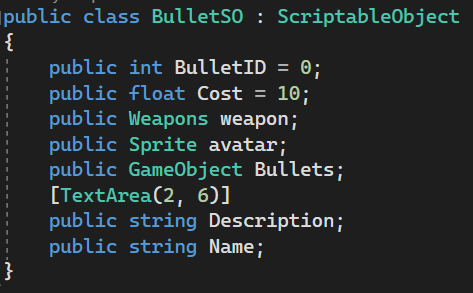
Hình 3.19: Biểu đồ lớp chức năng Chiến đấu của người chơi

### 3.2.4 Thiết kế cơ sở dữ liệu

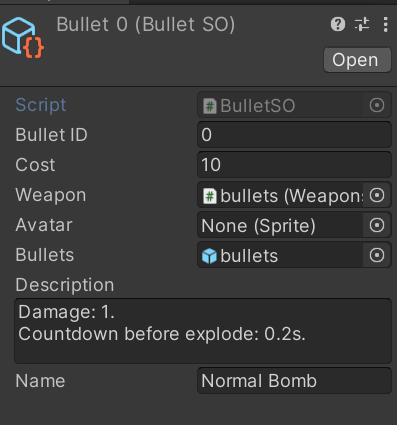
Mô hình thiết kế của trò chơi không sử dụng cơ sở dữ liệu mà lưu trữ dữ liệu theo dạng file là ScriptableObjects hay lưu trữ trực tiếp vào đối tượng trên Scene hoặc bằng PlayerPrefs của Unity.

Sử dụng ScriptableObject: là một lớp đặc biệt trong Unity được sử dụng để lưu trữ dữ liệu tĩnh giữa các phiên chơi và các tài nguyên trong game. Điều này có nghĩa là dữ liệu của ScriptableObject được lưu trữ trong tệp và có thể được tải lên bất cứ lúc nào trong suốt quá trình chơi game.

+ Dữ liệu về vũ khí của game được lưu trong các ScriptableObject: BulletSO. Trong lớp BulletSO các thông tin của vũ khí: ID, giá, prefab của vũ khí, Ảnh, mô tả và tên.

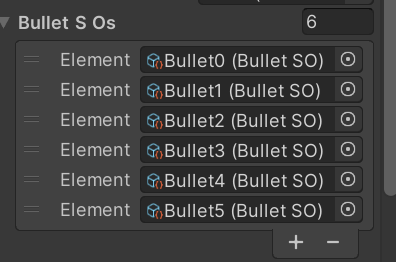


Hình 3.20: Hình ảnh Script lớp BulletSO



Hình 3.21: Hình ảnh Script lớp BulletSO

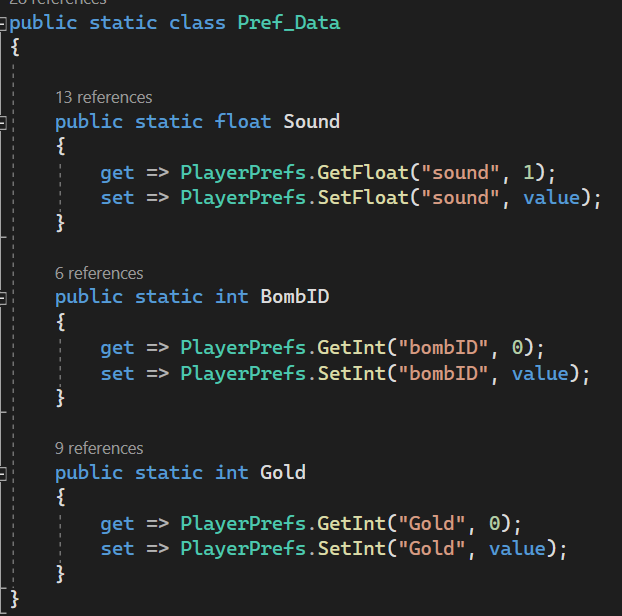
+ Dữ liệu được truy cập thông qua 1 list BulletSO trong phần tử game GameManager.



Hình 3.22: Danh sách các BulletSO

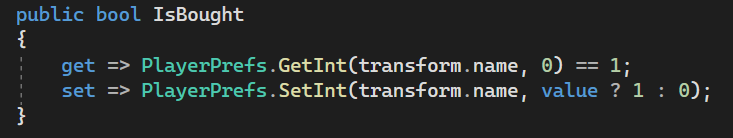
Sử dụng PlayerPrefs: PlayerPrefs là một lớp trong Unity được sử dụng để lưu trữ dữ liệu. Các giá trị được lưu trữ bởi PlayerPrefs sẽ được lưu trữ trong bộ nhớ của thiết bị và được giữ nguyên giá trị cho đến khi nó được ghi đè hoặc bị xóa. Game có 2 lớp sử dụng PlayerPrefs để lưu trữ dữ liêu là Pref\_Data và Weapons. Dữ liêu được lưu bằng PlayerPrefs là những dữ liêu động có thể được người chơi thay đổi mà không mất đi khi thoát màn chơi hoặc trò chơi.

+ Lớp Pref\_Data lưu trữ các dữ liệu của người chơi về: Cài đặt âm thanh, ID của vũ khí đang sử dụng, Số vàng của người chơi.



Hình 3.23: Hình ảnh lớp Pref\_Data

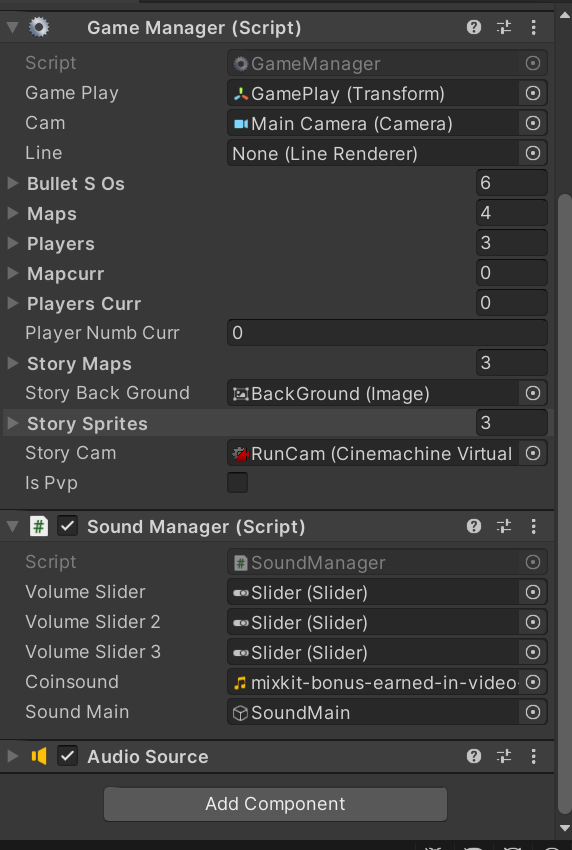
+ Lớp Weapons có thuộc tính IsBought đại diện cho tình trạng “Đã được mua” của vũ khí, giúp cho người chơi có thể mua và lưu trữ vũ khí theo thời gian. Ngoài thuộc tính này ra, các thuộc tính khác của lớp đều được lưu trực tiếp trong prefab.



Hình 3.24: Hình ảnh thuộc tính IsBought

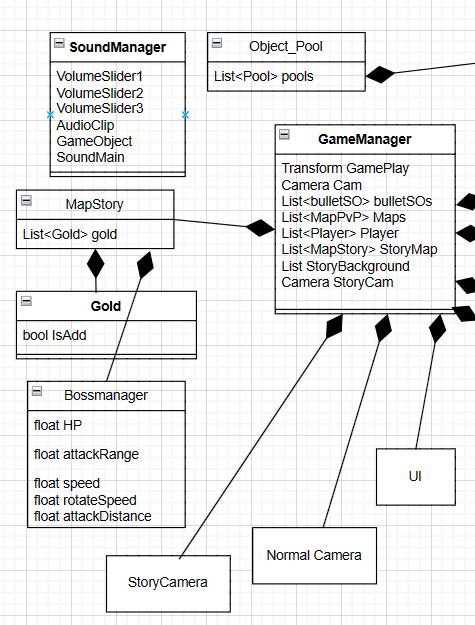
Lưu dữ trực tiếp vào object trên Scene và Prefab: là dữ liệu động có thể thay đổi trong qua trình chơi game như sẽ bị reset lại về giá trị mặc định khi người chơi thoát ra. Ngoại trừ các lớp và thuộc tính đã được lưu bằng ScriptableObject và PlayerPrefs, các dữ liệu còn lại của game đều được lưu bằng cách này.

+ Dữ liệu của game được sử dụng liên kết với nhau thông qua Script của game khi Scene chạy.



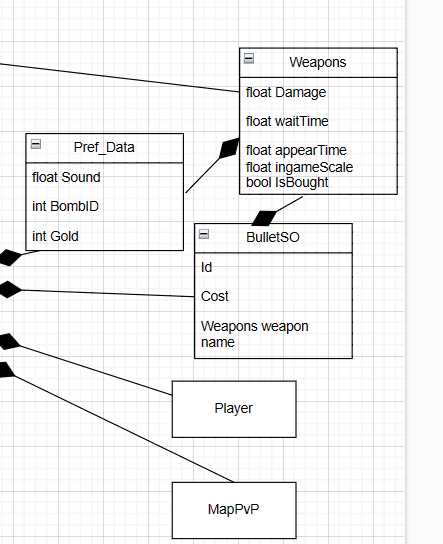
Hình 3.25: Dữ liệu được lưu trực tiếp trong scene

Ngoài ra, một số dữ liệu đã có sẵn trong scene khi scene chạy và được bật tắt thông qua Script của game.



Hình 3.26: Mô hình hóa các dữ liệu có sẵn trong scene

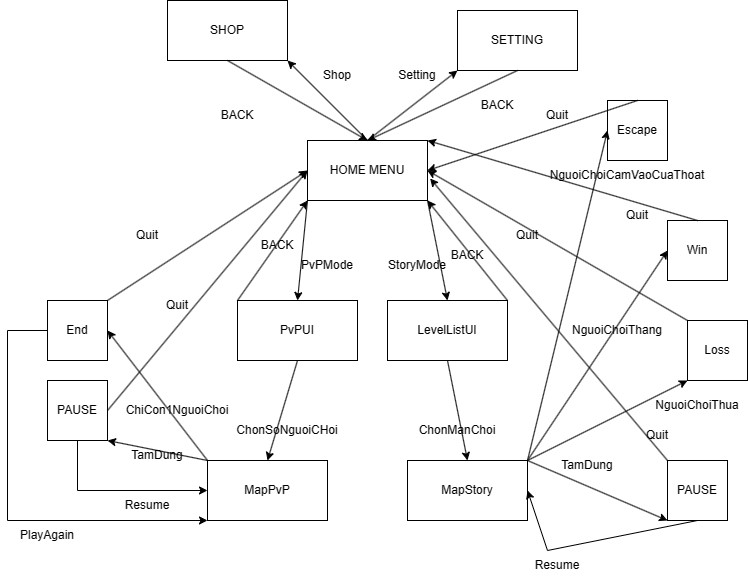
Một số dữ liệu được lưu dưới dạng Prefab, ScriptableObject,... không có sẵn trong scene nhưng có thể được gọi hoặc ánh xạ thông qua một số Script như GameManager,...



Hình 3.27: Mô hình hóa các dữ liệu không có sẵn trong Scene

### 3.2.5 Thiết kế giao diện

Biểu đồ:



Hình 3.28: Biểu đồ Flowchart

Các hình ảnh minh họa cho các thiết kế của giao diện chính như: Home Menu, Shop, Setting, PvP, LevelList, Pause và giao diện màn chơi được sẽ được giới thiệu ở mục 3.3.2.

## 3.3 Xây dựng và kết quả

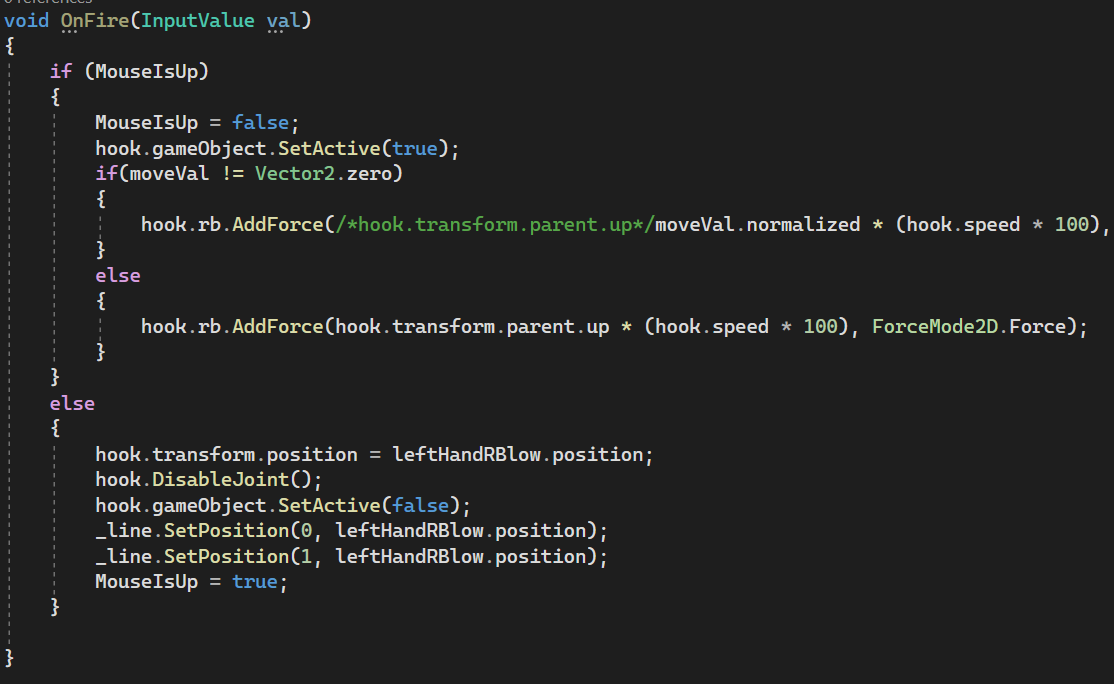
### 3.3.1 Giải quyết một số vấn đề

**3.3.1.1 Vấn đề tối ưu:**

* Hàm Update là hàm được gọi liên tục trong suốt quá trinh chơi. Vì vậy, nếu gọi quá nhiều phương thức phức tạp trong hàm Update có thể gây tốn rất nhiều tài nguyên máy khi chạy, khiến cho game giật, lag.
* Game có yếu tố đối kháng và bắn đạn (hành động ném bom của người chơi) dẫn đến việc các object Bom được sinh ra liên tục. Chúng có đặc điểm chung là giống nhau và sau khi sinh ra sẽ bị xóa đi. Tuy nhiên, việc liên tục sinh ra và hủy đi các dữ liệu mới có thể dẫn tới phân mảnh bộ nhớ và giảm đáng kể hiệu năng. Vì vậy, cần có phương pháp phù hợp để tránh tình trạng bộ nhớ bị phân mảnh và giảm hiệu năng của game mà không gây ảnh hưởng đến lối chơi.

**3.3.1.2 Giải quyết vấn đề tối ưu:**

* Cách tối ưu hàm Update: Sử dụng Input System của Unity. Input System của Unity cho phép chúng ta gọi các phương thức theo sự kiện bàn phím của người chơi, giúp tiết kiệm đáng kể gánh nặng trong hàm Update.



Hình 3.29: Ví dụ hàm gọi theo sự kiện của Input System

* Giải quyết vấn đề bộ nhớ bị phân mảnh khi phải liên tục sinh ra và hủy đi các object giống nhau: Sử dụng kĩ thuật Pooling Object.

Pooling object là kỹ thuật cho phép tái sử dụng các đối tượng đã được tạo ra trước đó thay vì tạo ra các đối tượng mới khi cần. Điều này giúp giảm thiểu việc tạo và xóa các đối tượng liên tục trong game, giảm thiểu tình trạng bộ nhớ bị phân mảnh và tăng hiệu năng của game.

Đối với game, em sử dụng kĩ thuật để quản lí số lượng Bom trong chiến đấu. Khi cần sử dụng Bom, em sẽ lấy chúng từ pool đối tượng đã được tạo sẵn, thay vì tạo mới. Khi không sử dụng nữa sẽ trả lại các đối tượng này vào pool để sử dụng lại trong tương lai. Nhờ đó, game đã trở nên mượt mà hơn và hiệu năng cũng được cải thiện đáng kể.

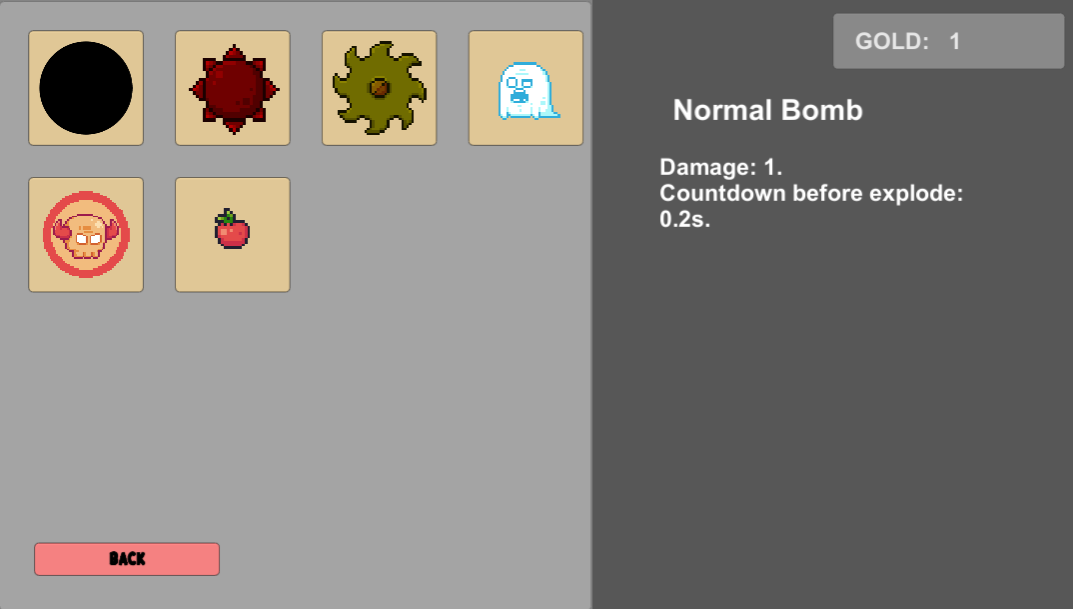
### 3.3.2 Kết quả

- **Giao diện Màn hình chính**: Gồm các 2 chế độ chơi, Cửa hàng và Setting cho người chơi lựa chọn.



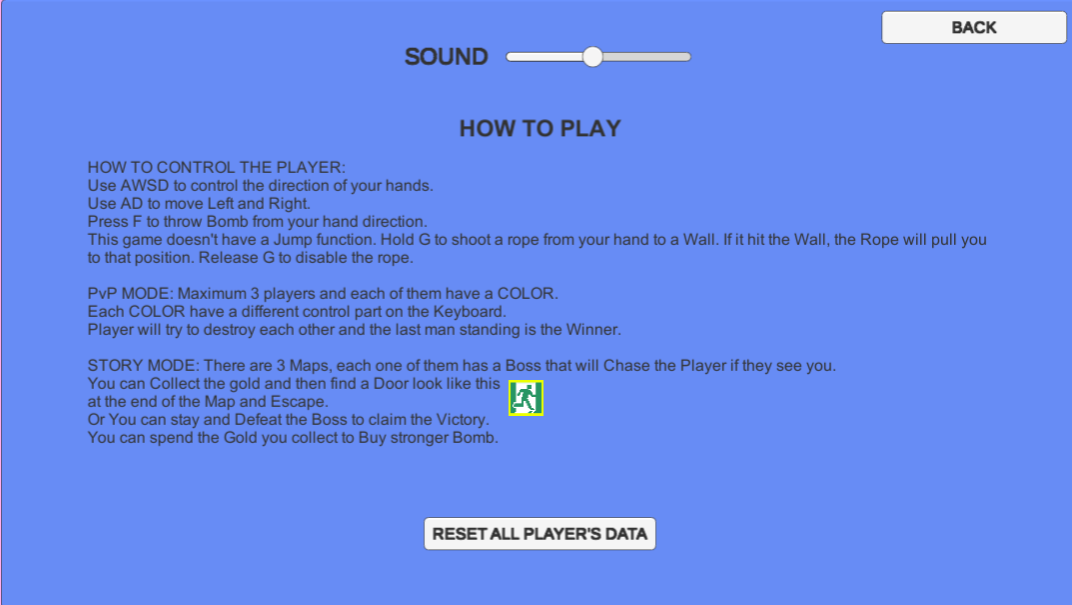
Hình 3.30: Giao diện màn hình chính

**- Giao diện Cửa hàng**: Gồm danh sách các vũ khí, phía bên phải là nơi hiển thị thông tin của vũ khí khi được chọn.



Hình 3.31: Giao diện Cửa hàng

- Giao diện Cài đặt: Gồm các 2 chức năng điều chỉnh âm thanh và reset trò chơi cho người chơi lựa chọn.



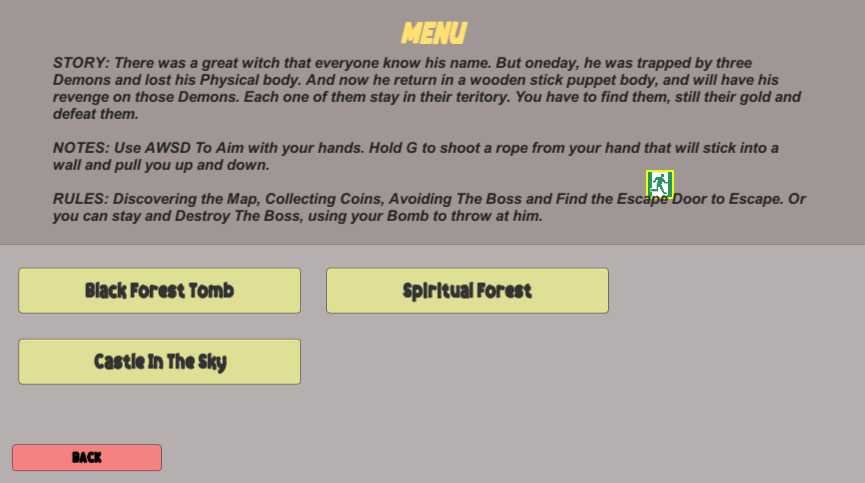
Hình 3.32: Giao diện Cài đặt

- Giao diện Chọn số người chơi: Gồm các 2 chế độ cho người chơi lựa chọn là 2 người chơi và 3 người chơi.



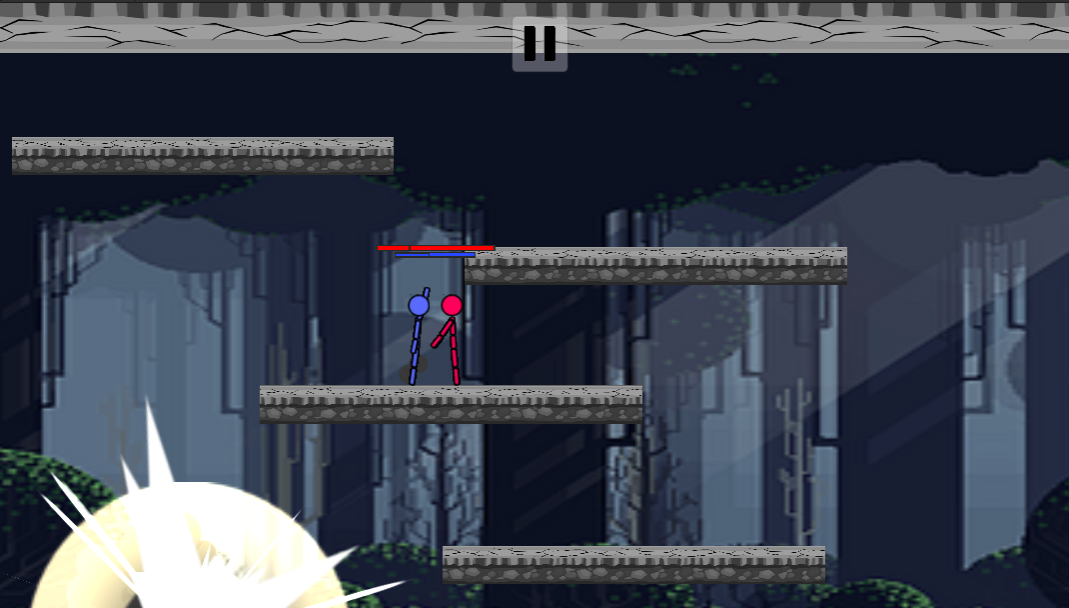
Hình 3.33: Giao diện Chọn số người chơi

- Giao diện Chọn Màn chơi: Gồm 3 màn chơi với đặc trưng riêng. Khi người chơi chọn màn chơi nào sẽ được đưa tới màn chơi đó.



Hình 3.34: Giao diện Chọn màn chơi

- Giao diện Màn chơi chế độ nhiều người chơi:



Hình 3.35: Giao diện Chế độ nhiều người chơi

- Giao diện Màn chơi chế độ một người chơi:



Hình 3.36: Giao diện Chế độ một người chơi

# CHƯƠNG 4. TEST GAME 2D THE PUPPET’S REVENGE

**Môi trường kiểm thử:**

Máy tính cá nhân có cài phần mềm Unity và Visual Studio. Các giao diện của hệ thống được kiểm tra trên các hệ điều hành Windows.

**Test Plan:**

**-Người kiểm thử và trải nghiệm game:**

**+** Trần Phan Công Minh, sinh viên năm nhất.

+ Trần Đức Thuận.

Hình 4.1: Bảng Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Nội dung kiểm thử | Thời gian kiểm thử | Cách thức kiểm thử | Kết quả |
| Chức năng chọn chế độ chơi | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng chọn số người chơi | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng chọn màn chơi | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng chơi lại | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng cửa hàng | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng thưởng | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng Cài đặt | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng tạm dừng | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng Quay lại màn hình chính | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng Chiến đấu của người chơi | Sau khi hoàn tất phần mềm | Thủ công | PASS |
| Chức năng Chiến đấu của Boss | Sau khi hoàn tất phần mềm | Thủ công | PASS |

### Test Report:

Thông qua việc tổng hợp và phân tích các test case mà các người kiểm thử và trải nghiệm cung cấp thì hiện tại chưa có lỗi nào phát sinh và đã hoàn thành kiểm thử các test case thành công. Có thể khẳng định rằng chương trình phần nào hoạt động hiệu quả đúng với mong muốn của người thực hiện.

# KẾT LUẬN

**Kết quả đạt được**

* Chương trình đã đạt được các mục tiêu đề ra. Các chức năng ở phần khảo sát đều hoạt động đầy đủ và không có lỗi phát sinh.
* Hiểu được quá trình phát triển game bao gồm từ khâu lên ý tưởng, thiết kế đến lập trình và kiểm thử.
* Tận dụng sự hỗ trợ từ các công cụ như Visual Studio, Unity, GitHub.
* Có thể sử dụng các công cụ hỗ trợ của Unity như là Input System, Cinemachine để phục vụ cho đề tài.
* Biết sử dụng các ứng dụng và kĩ thuật để tối ưu cho hoạt động của game.

**Hạn chế**

* Hệ thống vũ khí vẫn còn đơn điệu, chưa đa dạng.
* Các Map chơi và Boss còn ít.
* Người chơi chưa thể tự cài đặt phần điều khiển nhân vật.

**Hướng phát triển**

Trong tương lai, em sẽ tiếp tục cải tiến các chức năng của game, mở rộng cốt truyện, Map chơi và kho vũ khi, nhằm nâng cấp trải nghiệm của người dùng khi chơi game.

* Tạo ra thêm nhiều vũ khí khác nhau, Thêm nhiều âm thanh, hình ảnh sống động hơn cho game.
* Mở rộng cốt truyện, tạo ra thêm nhiều màn chơi khác nhau.
* Cho phép người chơi tự điều chỉnh phần điều khiển.

# 

# TÀI LIỆU THAM KHẢO

1. Microsoft, C# documentation. [Online]. https://learn.microsoft.com/en-us/dotnet/csharp/
2. Mono, Mono documentation. [Online]. https://docs.unity3d.com/Manual/Mono.html
3. Cinemachine. [Online]. <https://unity.com/unity/features/editor/art-and-design/cinemachine>
4. Input System. [Online]. <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.8/manual/index.html>
5. <https://www.kelleherbros.com/blog/2024/3/4/defining-the-platformer-genre>
6. Game Design Principles. [Online]. <https://www.evercast.us/blog/game-design-principles>
7. <https://docs.unity3d.com/Manual/index.html>
8. Pooling Object. [Online]. https://learn.unity.com/tutorial/introduction-to-object-pooling#