**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------



**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHÓA: 2020 – 2025**

***ĐỀ TÀI:***

**Xây dựng game Platformer sử dụng Unity**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Nguyễn Văn Hà** |
| **Lớp:** | **20CN1** |

**HÀ NỘI 12-2024**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------



**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHÓA: 2020 – 2025**

***ĐỀ TÀI:***

**Xây dựng game Platformer sử dụng Unity**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:** | **Th.s Nguyễn Mạnh Hùng** |
| **Sinh viên thực hiện:** | **Nguyễn Văn Hà** |
| **Lớp:** | **20CN1** |

**HÀ NỘI 12-2024**

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc nhất từ tận đáy lòng đến tất cả những người đã đồng hành, hỗ trợ và tạo điều kiện cho em trong suốt hành trình học tập và thực hiện đồ án tốt nghiệp kỹ sư với đề tài: **Xây dựng game Platformer sử dụng Unity**.

Em xin bày tỏ lòng biết ơn chân thành đến Ban lãnh đạo **Trường Đại học Kiến Trúc Hà Nội**. Nhờ tầm nhìn, sự quan tâm và tạo dựng môi trường giáo dục lý tưởng của Quý Nhà Trường, em và các bạn sinh viên đã có cơ hội được học tập và phát triển trong một không gian tràn đầy tri thức và cảm hứng.

Tiếp theo, xin gửi lời cảm ơn trân trọng đến **Khoa Công Nghệ Thông Tin**, nơi đã trang bị cho em không chỉ kiến thức chuyên môn mà còn là những bài học quý giá về tư duy và cách tiếp cận thực tế. Những nỗ lực của các Thầy Cô trong Khoa đã luôn là động lực lớn giúp em vượt qua những thử thách trong hành trình học tập.

Đặc biệt, em xin gửi lời tri ân sâu sắc đến thầy **ThS. NGUYỄN MẠNH HÙNG**, giảng viên hướng dẫn đã đồng hành cùng em trong từng bước thực hiện đồ án. Những hướng dẫn tận tình, lời khuyên sâu sắc của thầy không chỉ giúp em hoàn thành tốt đề tài mà còn mở ra cho em nhiều góc nhìn mới mẻ trong lĩnh vực Công nghệ Thông tin.

Bên cạnh đó, em xin gửi lời cảm ơn đặc biệt đến cô **ThS.** **Đinh Kim Phượng**, cố vấn học tập lớp **20CN1**, người đã luôn ân cần theo sát, động viên và định hướng cho chúng em trong cả học tập lẫn cuộc sống sinh viên đầy ý nghĩa.

Là một sinh viên còn nhiều hạn chế về kinh nghiệm, em hiểu rằng đồ án này vẫn có thể tồn tại những thiếu sót. Em rất mong nhận được sự đóng góp, chỉ dẫn từ các Thầy Cô và mọi người để em có cơ hội hoàn thiện bản thân, không ngừng học hỏi và phát triển trong tương lai.

Em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em xin cam đoan rằng đây là công trình nghiên cứu của riêng em và được thực hiện dưới sự hướng dẫn khoa học tận tình của **ThS. Nguyễn Mạnh Hùng**. Các nội dung nghiên cứu, kết quả trong đề tài tốt nghiệp **“Phát triển trò chơi Unity2D Platformer”** là hoàn toàn trung thực và chưa từng được công bố dưới bất kỳ hình thức nào trước đây.

Những số liệu, dữ kiện được sử dụng trong các bảng biểu để phục vụ cho việc phân tích, nhận xét và đánh giá đều được trích dẫn từ các nguồn đáng tin cậy, đồng thời được ghi rõ trong phần tài liệu tham khảo.

Trong quá trình thực hiện, em cũng sử dụng một số nhận xét, đánh giá cũng như số liệu từ các tác giả khác, cơ quan hoặc tổ chức liên quan. Tất cả các nguồn thông tin này đều đã được trích dẫn và chú thích đầy đủ, rõ ràng.

Nếu có bất kỳ sai sót hoặc hành vi gian lận nào xảy ra, em xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình. **Trường Đại học Kiến Trúc Hà Nội** không liên quan đến bất kỳ vi phạm tác quyền hay bản quyền nào phát sinh từ công trình này (nếu có).

|  |
| --- |
| *Hà Nội, ngày ...... tháng ...... năm 202…* |
| Sinh viên |
|  |
| Nguyễn Văn Hà |

# LỜI **NÓI** ĐẦU

**Lời nói đầu**

Với sự phát triển không ngừng của công nghệ, việc sáng tạo và triển khai các ứng dụng giải trí đã trở thành một phần quan trọng trong cuộc sống hiện đại. Dự án Platformer được xây dựng nhằm mang lại một không gian trải nghiệm sáng tạo, nơi người dùng không chỉ được giải trí mà còn được khám phá những giá trị mới từ công nghệ.

**Đặt vấn đề**

Trong bối cảnh công nghệ số ngày càng phát triển, các ứng dụng và trò chơi số đã trở thành một phần không thể thiếu trong đời sống hàng ngày. Dự án Platformer ra đời nhằm mục đích mang đến một trải nghiệm sáng tạo và giải trí cho người dùng, đồng thời khuyến khích việc học hỏi và ứng dụng công nghệ hiện đại trong phát triển trò chơi.

**Mục đích dự án**

Lựa chọn đề tài xuất phát từ niềm đam mê với lĩnh vực phát triển game và mong muốn khám phá các phương pháp thiết kế hiện đại trong Unity. Tựa game này là cơ hội để kết hợp sáng tạo nội dung với các giải pháp công nghệ tiên tiến, đồng thời thử thách bản thân trong việc hiện thực hóa một sản phẩm phức tạp và hoàn chỉnh.

Dự án Platformer được phát triển với các mục đích sau:

* Xây dựng một trò chơi độc đáo, hấp dẫn và dễ tiếp cận.
* Khai thác các tính năng tiên tiến của Unity để tạo ra sản phẩm chất lượng cao.
* Tạo môi trường học tập và thực hành cho các nhà phát triển phần mềm, đặc biệt là trong lĩnh vực phát triển trò chơi.

**Nội dung dự án**

Dự án tập trung vào các nội dung chính sau:

* Phát triển các cảnh trò chơi và giao diện trực quan, sinh động.
* Tích hợp các tính năng vật lý và hoạt họa để nâng cao trải nghiệm người chơi.
* Tối ưu hóa mã nguồn và tổ chức thư mục dự án để đảm bảo khả năng mở rộng và bảo trì.

**Ý nghĩa dự án**

* + - Phát triển kỹ năng chuyên môn, chuẩn bị cho nghề nghiệp trong ngành lập trình và phát triển game.
    - Góp phần cung cấp một sản phẩm giải trí sáng tao, đáp ứng nhu cầu trải nghiệm của người chơi.

**Công nghệ Sử dụng**

Dựa trên cấu trúc thư mục, dự án có vẻ được phát triển bằng Unity, với các công nghệ và công cụ sau:

* **Unity**: Công cụ phát triển trò chơi và ứng dụng đa nền tảng.
* **TextMesh Pro**: Plugin cho việc xử lý văn bản.
* **Visual Scripting**: Công cụ kéo-thả để tạo logic trò chơi mà không cần mã hóa.

**Hướng Dẫn Cài Đặt và Chạy Dự Án**

**Yêu cầu hệ thống**:

* + Unity Editor (phiên bản khuyến nghị: [phiên bản Unity được sử dụng])
  + Hệ điều hành: Windows, macOS, hoặc Linux.

**Cài đặt**:

* + Tải mã nguồn từ [GitHub](https://github.com/nguyenlam1962c5/Fun_Trap).
  + Giải nén nếu cần thiết.
  + Mở thư mục dự án trong Unity Editor.

**Chạy dự án**:

* + Trong Unity Editor, mở một scene chính từ thư mục Scenes.
  + Nhấn nút "Play" để kiểm tra.

**Kết luận**

Dự án Platformer là một nền tảng tiềm năng cho [mô tả ngắn về mục tiêu hoặc ứng dụng thực tế của dự án]. Việc tổ chức rõ ràng các thư mục và sử dụng công cụ mạnh mẽ như Unity mang lại khả năng mở rộng và phát triển lâu dài.

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc190364561)

[LỜI CAM ĐOAN ii](#_Toc190364562)

[LỜI NÓI ĐẦU iii](#_Toc190364563)

[**MỤC LỤC** v](#_Toc190364564)

[DANH MỤC BẢNG viii](#_Toc190364565)

[DANH MỤC HÌNH ẢNH ix](#_Toc190364566)

[KÝ HIỆU VÀ CHỮ VIẾT TẮT x](#_Toc190364567)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 1](#_Toc190364568)

[1.1. Giới thiệu về trò chơi 2D Platformer 1](#_Toc190364569)

[1.1.1. Định nghĩa và đặc điểm của trò chơi platforme 1](#_Toc190364570)

[**1.1.2. Vai trò và ý nghĩa của thể loại cho chơi này** 1](#_Toc190364571)

[**1.1.3. Ví dụ nổi bật về trò chơi platformer trên thị trường** 2](#_Toc190364572)

[**1.2. Công cụ phát triển trò chơi Unity2D** 2](#_Toc190364573)

[**1.2.1. Tổng quản Unity** 2](#_Toc190364574)

[**1.2.2. Các đặc điểm và tính năng của Unity** 3](#_Toc190364575)

[**1.3. Các thành phần trong Unity** 6](#_Toc190364576)

[**1.3.1. Assets (tài nguyên)** 6](#_Toc190364577)

[1.3.2. Scenes (cảnh) 7](#_Toc190364578)

[1.3.3. Game Object (đối tượng game) 8](#_Toc190364579)

[1.3.4. Components (thành phần ) 9](#_Toc190364580)

[1.3.5. Script 11](#_Toc190364581)

[1.3.6. Prefabs 11](#_Toc190364582)

[1.3.7. Giao diện người dùng UI 12](#_Toc190364583)

[1.3.8. Camera 13](#_Toc190364584)

[**1.4. So sánh Unity với các công cụ khác** 13](#_Toc190364585)

[1.5. Tổng quan về lập trình C# trong Unity 15](#_Toc190364586)

[1.5.1.C# cơ bản và ứng dụng trong Unity 15](#_Toc190364587)

[1.5.2.Script và cách hoạt động trong Unity 16](#_Toc190364588)

[1.5.3.Các thư viện hỗ trợ hữu ích 17](#_Toc190364589)

[CHƯƠNG 2 :TỔNG QUAN VỀ ĐỀ TÀI 20](#_Toc190364590)

[**2.1. Mục tiêu và phạm vi của đề tài** 20](#_Toc190364591)

[**2.1.1. Mục tiêu chính của trò chơi** 20](#_Toc190364592)

[2.1.2. Phạm vi phát triển (thời gian ,công cụ ,nguồn lực) 20](#_Toc190364593)

[2.2. Phân tích các trò chơi platformer hiện có 22](#_Toc190364594)

[2.2.1. Phân tích các trò chơi điển hình 22](#_Toc190364595)

[2.2.2. Điểm mạnh và điểm yếu của các trò chơi này 23](#_Toc190364596)

[2.2.3. Bài học kinh nghiệm cho đề tài 23](#_Toc190364597)

[2.3. Ý tưởng phát triển trò chơi 24](#_Toc190364598)

[2.3.1. Cốt truyện và nội dung 24](#_Toc190364599)

[2.3.2. Tính năng chính và cơ chế của GamePlay 25](#_Toc190364600)

[2.4. Các công cụ và công nghệ áp dụng 27](#_Toc190364601)

[2.4.1. Unity và Visual studio 27](#_Toc190364602)

[2.4.2. Công cụ thiết kế đồ họa và âm thanh 28](#_Toc190364603)

[CHƯƠNG 3 :XÂY DỰNG ỨNG DỤNG 29](#_Toc190364604)

[3.1. Thiết kế Player 29](#_Toc190364605)

[3.1.1. Hình ảnh của Player 29](#_Toc190364606)

[3.1.2. Di chuyển nhân vật 30](#_Toc190364607)

[3.1.3. Animation của Player 31](#_Toc190364608)

[3.1.4. Chức năng tấn công của Player 32](#_Toc190364609)

[3.1.5. Chức năng hồi sinh Player 33](#_Toc190364610)

[3.1.6. Máu của Player 34](#_Toc190364611)

[3.2. Kẻ địch ,bẫy, vật phẩm và kho báu 34](#_Toc190364612)

[3.2.1. Kẻ địch 34](#_Toc190364613)

[3.2.2. Bẫy trong game 37](#_Toc190364614)

[3.2.3. Vật phẩm trong Game 38](#_Toc190364615)

[3.2.4. Kho báu 41](#_Toc190364616)

[3.3. Các màn chơi trong game 41](#_Toc190364617)

[3.3.1. Màn 1:Vượt qua các chướng ngại vật và thu thập 3 viên ngọc thủy tinh 41](#_Toc190364618)

[3.3.2. Màn 2 :Tấn công quái thú để lấy chìa khóa ma thuật 42](#_Toc190364619)

[3.3.3.Màn 3 :Tấn công quái thú và vượt qua các chướng ngại vật để tìm thấy kho báu lấy chiếc la bàn ánh sáng . 42](#_Toc190364620)

[3.3.4. Tấn công quái thú và vượt qua các chướng ngại vật để tìm thấy kho báu lấy viên ngọc rồng . 43](#_Toc190364621)

[3.3.5. Qua màn 43](#_Toc190364622)

[3.4. Thiết kế Menu ,GameOver ,GameWin 43](#_Toc190364623)

[3.4.1. Thiết kế menu chính 44](#_Toc190364624)

[**3.4.2. Thiết kế màn hình GameOver** 44](#_Toc190364625)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 47](#_Toc190364626)

[4.1. Đánh giá kết quả 47](#_Toc190364627)

[4.2. Hướng phát triển trong tương lai 47](#_Toc190364628)

[TÀI LIỆU THAM KHẢO 49](#_Toc190364629)

# DANH MỤC BẢNG

[Bảng1: Bảng phân tich gameplay 24](#_Toc190381844)

[Bảng 2: Bảng ý tưởng và tổng quan phát triển game 29](#_Toc190381845)

[Bảng 3: Bảng phân tích nhân vật 30](#_Toc190381846)

# DANH MỤC HÌNH ẢNH

[Hình 2.1:Logo của Unity Engine 3](#_Toc190381307)

[Hình 2.2. Assets trong Unity 6](#_Toc190381308)

[Hình 2.3. Scenes trong Unity 7](#_Toc190381309)

[Hinh 2.4. Các tài nguyên trong Level1 9](#_Toc190381310)

[Hình 2.5 Các thành phần đối tượng Player 10](#_Toc190381311)

[Hình 2.6. Các Scripts trong dự án 11](#_Toc190381312)

[Hình 2.8 Một số đối tượng trong prefabs 12](#_Toc190381313)

[Hình 3.1: Hình ảnh Player ở trong game 29](#_Toc190381314)

[Hình 3.2: Scripts di chuyển của player 30](#_Toc190381315)

[Hình 3.3: Scripts nhảy của player 31](#_Toc190381316)

[Hình 3.4: Scripts bám trên tường của player 31](#_Toc190381317)

[Hình 3.5: Animation của player 32](#_Toc190381318)

[Hình 3.6: Script tấn công của player 32](#_Toc190381319)

[Hình 3.7: Thời gian hồi chiêu của Player 33](#_Toc190381320)

[Hình 3.8: Hồi sinh Player sau khi chết trong game 33](#_Toc190381321)

[Hình 3.9: Số lượng máu của player trong game 34](#_Toc190381322)

[Hình 3.10 : Hình ảnh của kẻ địch trong game 34](#_Toc190381323)

[Hình 3.11 : Script di chuyển của kẻ địch trong game 35](#_Toc190381324)

[Hình 3.12 : Animation của kẻ địch 35](#_Toc190381325)

[Hình 3.13 : Vùng di chuyển của kẻ địch 36](#_Toc190381326)

[Hình 3.14 : Vùng tấn công của kẻ địch 36](#_Toc190381327)

[Hình 3.15 : Hình ảnh bẫy ArrowTraps đang hoạt động 37](#_Toc190381328)

[Hình 3.16 : Hình ảnh bẫy firetrap đang hoạt động 37](#_Toc190381329)

[Hình 3.17 : Hình ảnh bẫy Enyme \_Sideways 37](#_Toc190381330)

[Hình 3.18 : Hình ảnh bẫy Spikehead 38](#_Toc190381331)

[Hình 3.19 : Hình ảnh viên Ngọc Thủy Tinh 38](#_Toc190381332)

[Hình 3.20 : Hình ảnh chiếc chìa khóa ma thuật 39](#_Toc190381333)

[Hình 3.21 : Hình ảnh chiếc la bàn ánh sáng 40](#_Toc190381334)

[Hình 3.22 : Hình ảnh viên ngọc rồng 40](#_Toc190381335)

[Hình 3.23 : Hình ảnh kho báu 40](#_Toc190381336)

[Hình 3.24 : Hình ảnh animation của rương 41](#_Toc190381337)

[Hình 3.25 : Hình ảnh map màn 1 41](#_Toc190381338)

[Hình 3.26 : Hình ảnh map màn 2 42](#_Toc190381339)

[Hình 3.27 : Hình ảnh map màn 3 42](#_Toc190381340)

[Hình 3.28 : Hình ảnh map màn 4 43](#_Toc190381341)

[Hình 3.29 : Script chuyển màn 43](#_Toc190381342)

[Hình 3.30 : Màn hình Menu chính trong game 44](#_Toc190381343)

[Hình 3.31 : Màn hình MenuOver trong game 45](#_Toc190381344)

[Hình 3.32 : Màn hình Menuwin trong game 46](#_Toc190381345)

# ****KÝ HIỆU VÀ CHỮ VIẾT TẮT****

|  |  |
| --- | --- |
| **Ký hiệu, chữ viết tắt** | **Ý nghĩa** |
| UI | User Interface (Giao diện người dùng) |
| AI | Artificial Intelligence (Trí tuệ nhân tạo của kẻ thù/NPC) |
| API | Application Programming Interface |
| VR | Virtual Reality |
| AR | Augmented Reality |
| JIT | Just-In-Time Compilation |

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## 1.1. Giới thiệu về trò chơi 2D Platformer

### 1.1.1. Định nghĩa và đặc điểm của trò chơi platforme

* Định nghĩa

Platformer là một thể loại trò chơi điện tử trong đó nhân vật chính di chuyển qua các nền tảng (platform) bằng cách nhảy, chạy, leo trèo hoặc sử dụng các cơ chế di chuyển khác. Trò chơi thường yêu cầu người chơi điều hướng qua môi trường với các chướng ngại vật và kẻ thù, đồng thời thu thập vật phẩm hoặc hoàn thành mục tiêu nhất định.

* Đặc điểm của trò chơi platformer

Cơ chế di chuyển chính**:** Nhân vật thường có thể nhảy từ nền tảng này sang nền tảng khác, leo dây, bám tường, hoặc sử dụng các công cụ đặc biệt như móc câu, jetpack,...

Môi trường có nhiều chướng ngại vật: Người chơi phải tránh các hố sâu, gai nhọn, kẻ thù hoặc các bẫy khác nhau.

Góc nhìn phổ biến: Thường sử dụng góc nhìn 2D (điển hình như Super Mario Bros., Sonic the Hedgehog) hoặc 2.5D/3D (như Crash Bandicoot, Super Mario 64).

Cơ chế vật lý đơn giản: Các yếu tố như trọng lực, quán tính và va chạm đóng vai trò quan trọng trong cách nhân vật di chuyển và tương tác với môi trường.

Yếu tố hành động hoặc giải đố: Một số platformer tập trung vào hành động (bắn súng, chiến đấu với kẻ thù), trong khi một số khác có yếu tố giải đố yêu cầu người chơi suy nghĩ để vượt qua thử thách.

Cấu trúc màn chơi theo cấp độ: Trò chơi thường được chia thành nhiều màn chơi hoặc thế giới khác nhau ,mỗi màn có độ khó tăng dần và các cơ chế gameplay mới .

* Một số trò chơi platformer nổi tiếng

Super Mario Bros. (Nintendo) – Một trong những trò chơi platformer kinh điển nhất.

Sonic the Hedgehog (SEGA) – Nổi bật với tốc độ nhanh và cơ chế cuộn cảnh đặc trưng.

Mega Man (Capcom) – Kết hợp yếu tố bắn súng với platformer.

Rayman (Ubisoft) – Nổi tiếng với đồ họa vẽ tay đẹp mắt và lối chơi mượt mà.

Celeste (Maddy Makes Games) – Một platformer hiện đại tập trung vào độ khó cao và cơ chế điều khiển chính xác.

**1.1.2. Vai trò và ý nghĩa của thể loại cho chơi này**

Thể loại Game Platform đóng vai trò tối quan trọng trong quá trình hình thành và phát triển ngành công nghiệp game. Với lối chơi phụ thuộc phần lớn vào kỹ năng cá nhân của người chơi, Game Platform giúp phát triển kỹ năng của người chơi, rèn luyện phản xạ, lối tư duy logic và khả năng điều khiển chính xác nhằm giải quyết thử thách khó khăn.

Cùng với những đóng góp to lớn trong ngành công nghiệp game, đặt nền móng cho các thiết kế môi trường, cơ chế mới và đồng thời là cảm hứng cho những thể loại game bắn súng, hành động sau này. Từ đó, Game Platform đã gắn liền với tuổi thơ của nhiều thế hệ, trở thành biểu tượng trong lịch sử phát triển game, từ đó khai sinh ra các tựa game huyền thoại một thời như Mario, Contra, Sonic,…

**1.1.3. Ví dụ nổi bật về trò chơi platformer trên thị trường**

Một số tựa game platformer kinh điển đã góp phần định hình thể loại này bao gồm Super Mario Bros, đại diện tiêu biểu với thiết kế màn chơi sáng tạo và cách chơi hấp dẫn. Sonic the Hedgehog của Sega lại nổi bật với tốc độ nhanh và khả năng điều khiển mượt mà, mang lại cảm giác phấn khích cho người chơi.

Trong thời kỳ hiện đại, Celeste là một ví dụ thành công, không chỉ nhờ lối chơi thử thách mà còn bởi cốt truyện sâu sắc và thông điệp về sức mạnh vượt qua nghịch cảnh. Hollow Knight cũng là một đại diện đáng chú ý, kết hợp yếu tố platforming với phiêu lưu và thế giới mở, tạo nên một trải nghiệm phong phú và đầy cảm xúc. Những tựa game này đều mang tính biểu tượng, thể hiện sự phát triển và sáng tạo không ngừng của thể loại game đi cảnh.

**1.2. Công cụ phát triển trò chơi Unity2D**

**1.2.1. Tổng quản Unity**



Hình 2.1:Logo của Unity Engine

Unity là một công cụ phát triển trò chơi mạnh mẽ, đa nền tảng, cho phép các nhà phát triển tạo ra các trò chơi với chất lượng cao. Với giao diện thân thiện và kho tài liệu phong phú, Unity cung cấp mọi thứ từ các công cụ lập trình, thiết kế đồ họa, cho tới khả năng xuất bản trên nhiều nền tảng khác nhau.

Unity được hỗ trợ của Just-In-Time Compilation(JIT) ,sử dụng thư viện mã nguồn mở C++ Mono.Bằng việc sử dụng JIT, những Engine như Unity có thể tận dụng lợi thế của tốc độ biên dịch .Những đoạn code do chúng ta viết sẽ được Unity biên dịch ra Mono trước khi nó được thực thi .Điều này rất quan trọng trong game để thực thi code vào những thời điểm cần thiết trong suốt quá trình chạy

Ngoài thư viện Mono ,Uninty cũng tận dụng chức năng của những thư viện phần mềm khác vào chức năng của nó ,như Engine mô phỏng vật lý physic X của Nvidia ,OpenGL và DirectX cho kết xuất hình ảnh 3D ,OpenAL cho âm thanh .Tất cả các thư viện này được xây dựng thành các tính năng tự động hoặc công cụ trực quan vào Unity, vì thế chúng ta không cần phải lo lắng về việc phải học thế nào để sử dụng chúng một cách riêng lẻ.

Unity có một cộng đồng người dùng rất lớn luôn chia se những Plugins, công cụ của họ dưới hình thức gói phần mềm bổ sung.

Có thể sản xuất các sản phẩm trò chơi theo tiêu chuẩn chuyên nghiệp ,xuất bản 3D cho cả MAC và PC cũng như sở hữu riêng một Web Player của riêng mình Unity là một trong những Game Engine có tốc độ phát triển nhanh nhất .Unity cũng có phiên bản phát triển game cho hệ máy Will của Nintendo và Iphone của Apple có nghĩa là một khi chúng ta làm chủ được những vấn đề cơ bản không chỉ phát triển game cho máy tính cá nhân mà chúng ta còn có thể phát triển game cho hệ máy cosole và thiết bị di động .

**1.2.2. Các đặc điểm và tính năng của Unity**

* **Rendering (kết xuất hình ảnh)**

Giống như tất cả các Engine hoàn chỉnh khác , Unity hỗ trợ đầy đủ khả năng kết xuất hình ảnh cùng nhiều hỗ trợ cho phép áp dụng các công cụ phổ biến trong lĩnh vực đồ họa 3D nhằm cải thiện chất lượng hình ảnh .Các phiên bản gần đây nhất của Unity được xây dựng lại thuận toán nhằm cải thiện hiệu suất kết xuất hình ảnh đồng thời tăng cường chất lượng hình ảnh sau khi kết suất.

* **Lighting(ánh sáng)**

Ánh sáng là một điều thiết yếu giúp môi trường đẹp và thực tế hơn .Unity 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 ánh sáng có hướng, ánh sáng điểm,… Một số công nghệ và kỹ thuật được Unity hỗ trợ như :Lingtmapping, Realtime Shadows, hiệu ứng Sunshafs và Lens Flares.

* **Terrains (địa hình)**

Terrains còn được gọi chúng 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 .Cùng với Terrains Tools là Tree Creator một công cụ mạnh mẽ cho phép chúng ta tạo ra các cây cối với hình dạng ,kích thước và kiểu cách đa dạng .

* **Substance (Texture thông minh)**

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

* **Physics(vật lý )**

PhysX là một Engine mô phỏng và xử lý vật lý cực 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, Rigitbodies, Ragdolls, Joints ,…

* **Audio (âm thanh )**

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)**

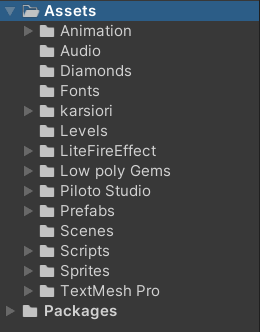
Lập trình là yếu tố quan trọng nhất trong phát triển game .Lập trình cho phép nhà phát triển tạo ra tương tác ,trí thông minh và yếu tố GamePlay trong trò chơi.

* **Ưu điểm của Unity**
* Đa nền tảng: hỗ trợ xuất bản game cho nhiều nền tảng khác nhau như Windows, macOS, Linux, Android, iOS, web (WebGL), console (PlayStation, Xbox, Nintendo Switch) và VR/AR. Điều này giúp nhà phát triển tiếp cận được nhiều đối tượng người chơi hơn.
* Dễ học và sử dụng: với giao diện trực quan, tài liệu hướng dẫn đầy đủ, cộng đồng lớn mạnh và nhiều hướng dẫn. Unity được xem là dễ tiếp cận cho người mới bắt đầu.
* Môi trường phát triển mạnh mẽ: cung cấp các công cụ cần thiết để thiết kế cấp đồ, tạo animation, lập trình, gỡ lỗi và tối ưu hóa hiệu năng.
* Linh hoạt: Unity cho phép nhà phát triển tùy biến và mở rộng chức năng thông qua việc sử dụng C# và các plungin từ Asset Store.
* Asset Store phong phú: có hàng ngàn tài sản, công cụ và plugin miễn phí và trả phí, giúp tiết kiệm thời gian và công sức phát triển.
* Cộng động lớn mạnh: cộng đồng người dùng Unity rất lớn và nhiệt tình, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức.
* Hỗ trợ VR/ AR: Unity là một trong những engine hàng đầu hỗ trợ phát triển các ứng dụng thực tế ảo (VR) và thực tế tăng cường (AR).
* Phát triển nhanh chóng: với nhiều công cụ và tài nguyên có sẵn, Unity cho phép nhà phát triển tạo ra các game và ứng dụng trong thời gian ngắn hơn so với việc viết code từ đầu.
* Cập nhật liên tục: Unity Technologies liên tục phát hành các bản cập nhật mới với nhiều tính năng và cải tiến.
* **Nhược điểm của Unity**
* **Hiệu xuất:** mặc dù đã có nhiều cải tiến, Unity đôi khi vẫn gặp vấn đề hiệu suất, đặc biệt với các game 3D phức tạp hoặc trên các thiết bị cấu hình thấp.
* **C#:** mặc dù C# là một ngôn ngữ mạnh mẽ nhưng có thể có độ dốc học tập lớn hơn đối với những người mới bắt đầu lập trình.
* **Quản lý dự án:** các dự án Unity có thể trở nên phức tạp khi có nhiều assets và script, đòi hỏi nhà phát triển phải có kỹ năng quản lý dự án tốt.
* **Asset Store không phải lúc nào cũng chất lượng:** không phải tất cả các tài sản trên Asset Store đều được kiểm duyệt kỹ lưỡng, nên người dùng cần cẩn trọng khi sử dụng.
* **Kích thước build lớn:** kích thước của game build từ Unity đôi khi có thể lớn hơn so với các engine khác.
* **Vấn đề về cập nhật:** đôi khi việc cập nhật lên các phiên bản Unity mới có thể gây ra các lỗi không tương thích với các dự án cũ.
* **Chi phí:** mặc dù có phiên bản miễn phí nhưng một số tính năng cao và việc sử dụng cho mục đích thương mại lớn hơn có thể đòi hỏi phải trả phí.
* **Tính linh hoạt hạn chế:** so với việc lập trình engine từ đầu, Unity có thể hạn chế một số tùy biến chuyên sâu.

**1.3. Các thành phần trong Unity**

**1.3.1. Assets (tài nguyên)**

Assets là một 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). Điều này lý giải tại sao tất cả các tập tin ,thư mục và dự án Unity đều được lưu trữ trong một thư mục có tên là Assets.



Hình 2.2. Assets trong Unity

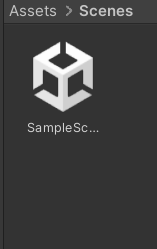
### 1.3.2. Scenes (cảnh)

Trong Unity ,chúng ta cần hiểu một cảnh (hay một phân đoạn) nghĩa là một màn chơi riêng biệt hoặc một khu vực hay thành phần có trong nội dung của trò chơi (ví dụ như Game menu) .Bằng cách tạo nên nhiều Scenes cho trò chơi ,chúng ta có thể phân phối thời gian tải hoặc kiểm tra các phần khác nhau của trò chơi một cách riêng biệt .

Các scene thường chứa một hệ thống phân cấp các GameObject, từ các đối tượng tĩnh như môi trường (địa hình, nhà cửa, cây cối) đến các đối tượng động như nhân vật, quái vật và các yếu tố tương tác. Việc tổ chức GameObject theo cấu trúc hợp lý trong scene giúp quản lý và chỉnh sửa dễ dàng hơn.

Unity cho phép bạn quản lý các scene trong dự án, thêm, xóa và sắp xếp chúng để dễ dàng thao tác thông quan Scene Management. Bạn có thể chuyển đổi giữa các scene bằng code, ví dụ như người chơi hoàn thành một màn chơi hoặc chọn một tùy chọn trong menu. Unity cung cấp các hàm để nạp scene mới một cách mượt mà. Bạn có thể nap nhiều scene cùng một lúc (Additive loading) để tạo các màn chơi phức tạp.

Scene là đơn vị tổ chức chính của game, nên việc nắm vững việc quản lý scene rất quan trọng để có thể tạo ra game có cấu trúc tốt.



Hình 2.3. Scenes trong Unity

### 1.3.3. Game Object (đối tượng game)

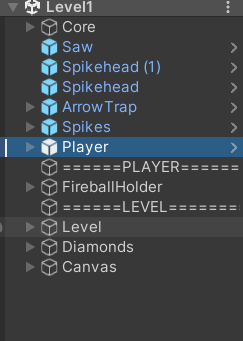
Khi Assest được sử dụng trong Scene ,chúng trở thành Game Object – một thuật ngữ được sử dụng trong Unity (đặc biệt là trong lập trình ) .Tất cả các Game Object đều chứa ít nhất một thành phần là Transform. Transform là thông tin về vị trí , góc xoay và tỷ lệ của tỷ lệ của đối tượng ,tất cả được mô tả bởi bộ 3 số X ,Y ,Z trong hệ trục tọa độ .Thành phần này có thể được tùy biến lại trong quá trình lập trình nhằm thay đổi vị trí ,góc quay và tỷ lệ của đối tượng qua các đoạn mã .Từ các thành phần cơ bản này ,chúng ta sẽ tạo ra GameObject với các thành phần khác ,bổ xung chức năng cần thiết để xây dựng nên bất kỳ một thành phần nào trong kịch bản Game mà chúng ta đã tưởng tượng.

Cấu trúc của GameObject:

* Transform: đây là component quan trọng nhất, xác định vị trí (Position), góc quay (Rotation) và kích thước (Scale) của GameObject trong không gian.
* Hierarchy: mỗi GameObject có thể có các GameObject con, tạo thành một hệ thống phân cấp. Việc này giúp bạn di chuyển, xoay, thay đổi kích thước cả một nhóm đối tượng một cách dễ dàng.

Tạo GameObject:

* Empty GameObject: GameObject rỗng, thường được dùng để làm cha các GameObject khác hoặc để chứa logic script.
* Primitive GameObject: GameObject có hình dạng sẵn như Cube, Sphere, Plane, Cylinder.
* GameObject từ Assets: tạo từ các model, ảnh, âm thanh,..



Hinh 2.4. Các tài nguyên trong Level1

### 1.3.4. Components (thành phần )

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 GameObject trong trò chơi. Bằng cách gắn chúng vào trong Game Object ,chúng ta ngay lập tức có thể áp dụng tác động chúng lên đối tượng .Những Components phổ biến trong quá trình phát triển đều được Unity hỗ trơ sẵn .Ví dụ thành phần Rigidbody đã được đề cập hay các yếu tố đơn giản khác như ánh sáng ,Camera và nhiều thành phần khác .Để tạo nên các yếu tố tương tác trong trò chơi ,chúng ta sẽ sử dụng Script , chúng cũng được xem như một Component trong Unity.

Các loại component phổ biến:

* Transform:
  + Position: vị trí
  + Rotation: góc quay
  + Scale: kích thước
* Renderer
  + Mesh Renderer: vẽ model 3D
  + Sprite Renderer: vẽ hình 2D
  + Line Renderer: vẽ đường thẳng
  + Particle System: tạo hiệu ứng hạt
* Collider: phát hiện va chạm
  + Box Collider: va chạm hình hộp
  + Sphere Collider: ca chạm hình cầu
  + Capsule Collider: va chạm hình con nhộng
  + Mesh Collider: va chạm theo hình dạng model 3D.
* Rigidbody: tạo tương tác vật lý.
* Audio Soure: phát âm thanh.
* Camera: tạo góc nhìn.
* UI Components: Button, Text, Image...
* Scrpit: code logic.

Cách component hoạt động:

* Mỗi Component có các thuộc tính và phương thức riêng.
* Bạn có thể chính sửa các thuộc tính của component trong Inspector.
* Script có thể truy cập và điều khiển các component khác của GameObject.



Hình 2.5 Các thành phần đối tượng Player

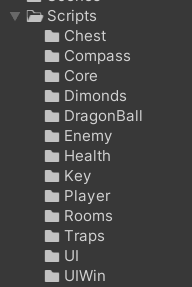
### 1.3.5. Script

Được Unity xem như một Component ,Script là một thành phần thiết yếu trong quá trình phát triển trò chơi và đáng được đề cập đến như một khái niệm “chìa khóa”.Unity cung cấp cho chúng ta khả năng viết Script bằng cả 3 loại ngôn ngữ là JavaScript, C# và Boo (một dẫn xuất của ngôn ngữ Python)

Unity không đòi hỏi chúng ta phải học như thế nào để lập trình trong Unity nhưng hầu như chúng ta phải sử dụng Script tại mỗi thành phần trong kịch bản mà chúng ta phát triển .Unity đã xây dựng sẵn một tập hợp đa dạng các lớp ,hàm mà chúng ta đã hoàn toàn có thể ứng dụng trong quá trình lập trình cho trò chơi của mình .

Để viết Script ,chúng ta sẽ làm việc với một trinh biên tập Script độc lập của Unity hoặc với chương trình Visual studio được tích hợp và đồng bộ với Unity trong những phiên bản mới nhất hiện nay .

Visual studio là một IDE khá tốt để lập trình khi cung cấp nhiều chức năng mã nguồn viết trên Visual studio sẽ được cập nhật và lưu trữ trong dự án Unity.



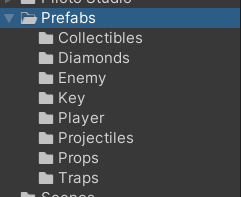
Hình 2.6. Các Scripts trong dự án

### 1.3.6. Prefabs

Prefabs cho phép chúng ta lưu trữ các đối tượng với những Component và những thiết đặt hoàn chỉnh ,prefabs chỉ đơn giản là một container (một đối tượng chưa) rỗng mà chúng ta có thể đưa bất kỳ một đối tượng hay dữ liệu nào mà chúng ta muốn tái sử dụng về sau .

Cách tạo và sử dụng Prefab:

* Kéo một GameObject từ Hierarchy vào thư mục Project để tạo Prefab.
* Kéo Prefab từ Project vào Scene để tạo ra các instance.
* Sửa Prefab để thay đổi tất cả các instance.
* Bạn có thể chỉnh sửa instance Prefab một cách độc lập bằng cách hủy liên kết với Prefab gốc.



Hình 2.8 Một số đối tượng trong prefabs

### 1.3.7. Giao diện người dùng UI

UI là các yếu tố đồ họa mà người chơi tương tác để điều khiển game, xem thông tin hoặc tùy chỉnh cài đặt. UI thường được vẽ trên Canvas (một GameObject đặc biệt) trong hệ tọa độ 2D.

Các UI Component phổ biến:

* Canvas: vùng vẽ UI.
* Text: hiển thị chữ.
* Image: hiển thị hình ảnh.
* Button: nút bấm.
* Slider: thanh trượt.
* Dropdown: danh sách thả xuống.
* Input Field: ô nhập liệu.
* Panel: container chứa các UI elements.

Cách tạo và quản lý:

* Tạo một Canvas GameObject để chứa các UI elements,
* Sử dụng các UI Component để tạo giao diện.
* Sử dụng scripts để sử lý sự kiện từ UI (click, hover, nhập liệu...).

### 1.3.8. Camera

Camera quyết định những gì người chơi thấy trong game và có nhiều loại camera khác nhau với các thiết lập và chức năng khác nhau.

Các loại Camera:

* Perspective Camera: tạo góc nhìn 3D thông thường.
* Orthographic Camera: tạo góc nhìn 2D.

Chức năng Camera:

* Vị trí và góc quay: thay đổi vị trí và góc quay của camera để thay đổi góc nhìn của người chơi.
* Projection Mode: chọn giữa perspective hoặc orthographic.
* Field of View (FOV): điều chỉnh góc nhìn rộng hẹp.
* Size: điều chỉnh kích thước của camera.
* Clipping Planes: điều chỉnh phạm vi nhìn thấy của camera.
* Clear Flags: điều chỉnh cách camera vẽ background.

Cách sử dụng Camera:

* Đặt Camera ở vị trí thích hợp trong scene.
* Gắn script vào camera để điều khiển chuyển động của camera.

**1.4. So sánh Unity với các công cụ khác**

Unreal Engine – Đồ họa đỉnh cao nhưng đầy thách thức: được phát triển bởi Epic Games, nổi bật với khả năng tạo ra đồ họa ấn tượng, phù hợp cho các dự án game AAA có yêu cầu cao về hình ảnh. Được hỗ trợ bởi công nghệ tiên tiến, Unreal Engine cung cấp các công cụ mạnh mẽ như ánh sáng động và hiệu ứng vật lý chi tiết. Điều này làm cho Unreal Engine trở thành lựa chọn hàng đầu của các nhà phát triển game chuyên nghiệp. Tuy nhiên, để tận dụng tối đa sức mạnh của Unreal Engine, người dùng cần có kiến thức sâu rộng về lập trình, đặc biệt là C++, cùng với một cấu hình máy tính đủ mạnh. Hệ thống Blueprint của Unreal giúp giảm bớt phần nào khó khăn cho những ai không rành về mã hóa, nhưng vẫn đòi hỏi thời gian để làm quen. Ngoài ra, hiệu suất và chất lượng đồ họa cao của Unreal có thể dẫn đến thời gian phát triển lâu hơn và chi phí cao hơn. Mặc dù vậy, Unreal Engine vẫn là lựa chọn lý tưởng cho các dự án cần tập trung vào chất lượng đồ họa và trải nghiệm người chơi đỉnh cao.

Godot – Công cụ mã nguồn mở linh hoạt và miễn phí: Godot là một công cụ phát triển game mã nguồn mở được cộng đồng đánh giá cao nhờ tính linh hoạt và khả năng tùy chỉnh mạnh mẽ. Đây là lựa chọn tuyệt vời cho cả các dự án 2D và 3D, đặc biệt với những ai muốn tìm kiếm một công cụ miễn phí nhưng không kém phần chuyên nghiệp. Godot sử dụng ngôn ngữ lập trình GDScript, tương tự Python, giúp việc học tập và triển khai trở nên dễ dàng hơn, đặc biệt với người mới bắt đầu. Công cụ này còn hỗ trợ các ngôn ngữ khác như C# và C++, tăng tính đa dạng trong lập trình. Tuy nhiên, khả năng xử lý đồ họa 3D của Godot vẫn chưa thể so sánh với Unity hay Unreal Engine, khiến nó phù hợp hơn với các dự án nhỏ hoặc vừa. Một điểm mạnh khác của Godot là giao diện người dùng trực quan và khả năng xuất bản trên nhiều nền tảng mà không yêu cầu phí bản quyền. Mặc dù vậy, cộng đồng người dùng của Godot vẫn đang phát triển, và tài nguyên hỗ trợ không phong phú bằng các công cụ lớn như Unity hoặc Unreal. Dù vậy, với tính linh hoạt và chi phí bằng 0, Godot là lựa chọn lý tưởng cho những nhà phát triển muốn thử nghiệm sáng tạo mà không bị ràng buộc bởi chi phí.

GameMaker Studio – Lựa chọn đơn giản cho game 2D: GameMaker Studio là một công cụ phát triển game tập trung chủ yếu vào 2D, được biết đến với giao diện kéo thả thân thiện, dễ tiếp cận cho cả những người không có nền tảng lập trình. Ngôn ngữ lập trình riêng của công cụ này, GML (GameMaker Language), được thiết kế đơn giản, dễ học, giúp người dùng nhanh chóng làm quen và triển khai ý tưởng của mình. GameMaker Studio phù hợp cho các dự án nhỏ hoặc trung bình, đặc biệt là trong lĩnh vực game di động hoặc indie. Tuy nhiên, điểm yếu lớn nhất của công cụ này là khả năng hỗ trợ đồ họa 3D còn hạn chế, khiến nó không phải là lựa chọn tốt cho các dự án yêu cầu hình ảnh phức tạp. Một nhược điểm khác là người dùng phải mua giấy phép riêng nếu muốn xuất bản trò chơi trên các nền tảng khác nhau, điều này có thể làm tăng chi phí phát triển. Dù vậy, với khả năng tập trung mạnh mẽ vào 2D và sự đơn giản trong sử dụng, GameMaker Studio là một giải pháp phù hợp cho những ai muốn nhanh chóng tạo ra một sản phẩm hoàn chỉnh mà không cần đầu tư quá nhiều thời gian hoặc nguồn lực.

## 1.5. Tổng quan về lập trình C# trong Unity

### 1.5.1.C# cơ bản và ứng dụng trong Unity

C Sharp (C#) là ngôn ngữ lập trình thuần hướng đối tượng được Microsoft phát triển năm 2000 chủ yếu được thiết kế Anders Hejlsberg. Được xây dựng dựa trên 2 ngôn ngữ trước đó là C++ và Java nên C# có được sự cân bằng giữa Visual Basic, C++, Delphi và Java. Với các đặc điểm nổi bật như: khả năng hướng đổi tượng hỗ trợ đóng gói, đa hình, kế thừa; hỗ trợ lập trình bất đồng bộ giúp các tác vụ chạy song song như Java; quảng lý bộ nhớ tự đồng nhờ vào Garbage Collection giúp gaimr thiểu lỗi rò rỉ; cùng với đó là bộ thư viện phong phú và mạnh mẽ của .NET Framework giúp phát triển ứng dụng từ desktop, web đến game.

Từ những ưu điểm kể trên, C# được dùng làm ngôn ngữ chính để lập trình trong Unity với các ứng dụng cụ thể như sau:

* **Viết logic cho trò chơi**
* Quản lý các hành động của nhân vật: di chuyển, nhảy, tấn công, hoặc tương tác với môi trường.
* Xử lý sự kiện: phản hồi khi người chơi nhấn nút, thu thập vật phẩm, hoặc gặp kẻ thù.
* Điều khiển AI cho NPC (Non-Player Character).
* **Tương tác với Unity API**

Unity cung cấp các API tích hợp mạnh mẽ qua C# để quản lý các thành phần trong trò chơi, như:

* **MonoBehaviour:** Lớp cơ sở cho tất cả các script, hỗ trợ các phương thức như Start() và Update() để xử lý logic thời gian thực.
* **Transform:** Quản lý vị trí, xoay và tỷ lệ của các đối tượng trong game.
* **Physics:** Sử dụng các chức năng xử lý va chạm, trọng lực, và động lực học.
* **Phát triển công cụ trong Unity Editor**

Tạo các công cụ tùy chỉnh để hỗ trợ thiết kế game, chẳng hạn như hệ thống quản lý tài nguyên hoặc chỉnh sửa đối tượng trong môi trường làm việc.

* **Xử lý bất đồng bộ và đa luồng**

Sử dụng các tính năng như async/await trong C# để xử lý các tác vụ như tải dữ liệu, chơi nhạc, hoặc kết nối mạng mà không làm chậm trò chơi.

* **Tích hợp công nghệ hiện đại**
* Phát triển game đa người chơi (Multiplayer) với API mạng.
* Tích hợp các plugin và thư viện từ bên thứ ba để mở rộng tính năng.

### 1.5.2.Script và cách hoạt động trong Unity

Script trong Unity có nhiệm vụ chạy các chắc năng bằng mã thay vì xử lý bằng **Unity Editor UI.** Điều này cho phép bạn vượt ra khỏi những giới hạn rằng buộc bởi **Editor UI**. Ngoài ra, việc công khai tương tác trực tiếp với các API Unity cho phép chúng có quyền kiểm soát và tinh chỉnh ở mức độ cao hơn.

Trong Unity, **Script** giúp định nghĩa các logic và hành vi của các đối tượng trong trò chơi. Unity sử dụng **C# scripts** để viết mã, cho phép tương tác với Unity Engine thông qua các API mạnh mẽ nh:.

* **Tạo và Gắn Script:**
* Tạo script trong Unity thông qua Project Window, Unity sẽ tự động tạo tệp với cấu trúc cơ bản kế thừa từ lớp **MonoBehaviour**.
* Gắn script vào một **GameObject** trong Scene, script sẽ trở thành một **Component** của GameObject.
* **Cấu Trúc Cơ Bản Của Script:**
* Start(): Được gọi một lần khi GameObject được kích hoạt trong Scene, dùng để khởi tạo.
* Update(): Được gọi mỗi khung hình (frame), dùng để xử lý các thay đổi liên tục.
* **Tương Tác Với Unity API:**
* Transform: Điều khiển vị trí, xoay, và tỷ lệ của GameObject.
* Rigidbody: Thêm các tính năng vật lý như trọng lực hoặc lực.
* Collider: Phát hiện và xử lý va chạm.
* Input: Xử lý đầu vào từ bàn phím, chuột, hoặc tay cầm điều khiển.
* **Các Vòng Đời Quan Trọng Của MonoBehaviour:**
* Awake(): Gọi trước Start(), thường dùng để khởi tạo các thành phần hoặc tài nguyên.
* OnEnable()/OnDisable(): Gọi khi GameObject được kích hoạt hoặc tắt.
* FixedUpdate(): Chạy ở mỗi khung vật lý, thường dùng cho logic vật lý.
* LateUpdate(): Gọi sau Update(), dùng cho các tác vụ liên quan đến camera.
* **Tích Hợp Sự Kiện Và Tương Tác:**
* Xử lý va chạm thông qua các sự kiện như OnCollisionEnter.
* Xử lý trigger với OnTriggerEnter.
* **Quản Lý Bất Đồng Bộ (Asynchronous):**
* Unity hỗ trợ **Coroutine** để xử lý các tác vụ diễn ra theo thời gian, ví dụ: chờ hoặc lặp.
* Sử dụng các phương thức như yield return để điều khiển luồng thực thi.
* **Kết Nối Giữa Các Script:**
* Tham chiếu trực tiếp thông qua GetComponent.
* Tham chiếu công khai bằng cách kéo thả trong Unity Editor.

### 1.5.3.Các thư viện hỗ trợ hữu ích

Unity đã cung cấp rất nhiều bộ thư viện mạnh mẽ để hỗ trợ các lập trình viên và tự động quản lý các thành phần của trò chơi. Dưới đây có 6 thư viện quan trọng và được sử dụng phổ biến nhất:

Thư viện UnityEngine: Là nền tảng cốt lõi của Unity, cung cấp các lớp và chức năng cần thiết để phát triển trò chơi và ứng dụng tương tác. Nó bao gồm các thành phần quan trọng như GameObject, Transform, Vector3, và nhiều đối tượng khác, cho phép nhà phát triển tạo và quản lý các yếu tố trong thế giới ảo. Ví dụ, GameObject là đơn vị cơ bản của mọi đối tượng trong Unity, trong khi Transform xác định vị trí, xoay và tỉ lệ của đối tượng đó trong không gian. Sự kết hợp của các lớp này giúp xây dựng cấu trúc và hành vi cho các thành phần trong trò chơi một cách hiệu quả.

Ngoài các lớp cơ bản, UnityEngine còn cung cấp các công cụ để xử lý vật lý, âm thanh, hoạt hình và giao diện người dùng. Các lớp như Rigidbody và Collider hỗ trợ mô phỏng vật lý thực tế, cho phép các đối tượng tương tác thông qua va chạm và lực. Thành phần AudioSource và AudioClip giúp tích hợp âm thanh vào trò chơi, tạo ra trải nghiệm sống động cho người chơi. Hệ thống hoạt hình được quản lý thông qua các lớp như Animator và AnimationClip, cho phép tạo ra các chuyển động mượt mà và phức tạp cho nhân vật và đối tượng. Bên cạnh đó, các lớp như Canvas và UI hỗ trợ xây dựng giao diện người dùng, giúp tạo ra các menu, nút bấm và các thành phần tương tác khác.

Việc sử dụng thư viện UnityEngine đòi hỏi kiến thức về lập trình C# và hiểu biết về cấu trúc của Unity. Nhà phát triển cần nắm vững cách sử dụng các lớp và phương thức trong thư viện để tạo ra các chức năng mong muốn. Tài liệu chính thức của Unity cung cấp thông tin chi tiết về từng thành phần trong UnityEngine, bao gồm mô tả, cú pháp và ví dụ minh họa, giúp nhà phát triển dễ dàng tra cứu và áp dụng trong dự án của mình. Ngoài ra, cộng đồng Unity cũng chia sẻ nhiều tài nguyên và hướng dẫn hữu ích, hỗ trợ việc học tập và giải quyết các vấn đề trong quá trình phát triển. [4]

**UnityEngine.UI:** cũng thư viện hỗ trợ việc xây dựng giao diện người dùng (UI) cho trò chơi. Nó cung cấp các thành phần như Button, Text, Canvas, giúp bạn dễ dàng tạo các nút, hiển thị văn bản hoặc quản lý bố cục giao diện. Ngoài ra, UnityEngine.UI còn hỗ trợ tùy chỉnh hành vi giao diện, chẳng hạn như xử lý sự kiện khi người chơi nhấn nút, kéo thanh trượt hoặc điều chỉnh giao diện theo kích thước màn hình.

**Tiếp đến là UnityEngine.SceneManagement** có nhiệm vụ quản lý các Scene trong trò chơi là nhiệm vụ chính của thư viện này. Bạn có thể sử dụng LoadScene() để chuyển đổi giữa các màn chơi hoặc UnloadScene() để giải phóng bộ nhớ của Scene không còn cần thiết. Thư viện còn hỗ trợ các thao tác tải Scene theo cách bất đồng bộ, giúp giảm thời gian chờ và tối ưu hóa hiệu suất của trò chơi.

Theo đó là: **UnityEngine.Physics** là bộthư viện vật lý giúp mô phỏng các hiện tượng thực tế như trọng lực, va chạm, hoặc lực tác động giữa các đối tượng. Rigidbody cho phép các vật thể tương tác theo quy luật vật lý, trong khi Collider và Raycast hỗ trợ phát hiện va chạm hoặc kiểm tra các vật thể trong không gian 3D. Sử dụng UnityEngine.Physics, bạn có thể tạo các hiệu ứng vật lý sống động và chân thực hơn.

Trong đó, **UnityEngine.AI** lại dành riêng cho phát triển trí tuệ nhân tạo (AI), thư viện này giúp các đối tượng như NPC (Non-Player Character) tự tìm đường và di chuyển thông minh. NavMesh tạo bản đồ đường đi, NavMeshAgent kiểm soát di chuyển của nhân vật AI, còn NavMeshObstacle xử lý các chướng ngại vật trong môi trường. Với thư viện này, bạn có thể tạo ra AI phản ứng linh hoạt và phù hợp với môi trường trò chơi.

Cuối cùng là **TextMeshPro.** Đây là thư viện mạnh mẽ để xử lý văn bản trong trò chơi, cho phép hiển thị chữ với độ phân giải cao và hiệu suất tốt hơn so với các công cụ văn bản mặc định. Nó cung cấp các tùy chọn như đổi màu chữ, hiệu ứng đổ bóng, và tạo kiểu chữ 3D, giúp cải thiện chất lượng hiển thị văn bản. TextMeshPro là lựa chọn lý tưởng cho các trò chơi cần giao diện đẹp và rõ ràng.

# CHƯƠNG 2 :TỔNG QUAN VỀ ĐỀ TÀI

**2.1. Mục tiêu và phạm vi của đề tài**

**2.1.1. Mục tiêu chính của trò chơi**

Thử thách kỹ năng xử lý tình huống của người chơi: Trò chơi được thiết kế để kiểm tra khả năng tư duy nhanh và xử lý tình huống của người chơi. Trong môi trường đầy cạm bẫy, người chơi phải liên tục đưa ra quyết định trong thời gian ngắn để vượt qua các chướng ngại vật. Điều này không chỉ yêu cầu sự nhanh nhẹn mà còn đòi hỏi khả năng dự đoán và lập kế hoạch. Mỗi cấp độ trong trò chơi đều có sự gia tăng về độ khó, tạo động lực để người chơi không ngừng cải thiện kỹ năng.

Khuyến khích tính kiên nhẫn và sự bền bỉ: Trò chơi tạo ra một loạt thử thách khiến người chơi có thể thất bại nhiều lần trước khi hoàn thành mục tiêu. Tuy nhiên, chính những lần thất bại này sẽ rèn luyện tính kiên nhẫn và ý chí không bỏ cuộc. Game không chỉ là một trò chơi giải trí mà còn là một bài học về sự kiên trì trong cuộc sống.

Tăng cường khả năng ghi nhớ và phản xạ: Một phần quan trọng của trò chơi là việc người chơi phải ghi nhớ vị trí của các bẫy hoặc nguy hiểm đã gặp phải. Kết hợp với khả năng phản xạ nhanh, giúp người chơi rèn luyện trí nhớ và cải thiện khả năng phản ứng với những thay đổi bất ngờ.

Mang lại niềm vui và giải trí: Bên cạnh những yếu tố thử thách vẫn giữ mục tiêu chính là tạo ra niềm vui cho người chơi. Thiết kế đồ họa, âm thanh và các yếu tố bất ngờ trong trò chơi mang lại trải nghiệm thú vị, khiến người chơi cảm thấy hứng thú trong mỗi lần chinh phục.

### 2.1.2. Phạm vi phát triển (thời gian ,công cụ ,nguồn lực)

Kế hoạch phát triển trò chơi Platformer bắt đầu từ ngày 21/10/2024 và kéo dài 15 tuần, chia thành 7 giai đoạn cụ thể. Mỗi giai đoạn đều được làm đúng tiến độ dưới sự theo dõi và hướng dẫn của giảng viên hướng dẫn ThS. Nguyễn Mạnh Hùng.

Giai đoạn đầu tiên, kéo dài 1 tuần, là thời điểm phát triển bước vào thế giới của Unity – công cụ sẽ định hình toàn bộ trò chơi . Đây không chỉ là quá trình làm quen với giao diện và các tính năng cơ bản của Unity mà còn là cơ hội để khám phá những tiềm năng sáng tạo mà công cụ này mang lại. Trong tuần này, cần tập trung nghiên cứu cách Unity hỗ trợ phát triển trò chơi 2D, từ quản lý tài nguyên, tạo hình ảnh động đến tích hợp âm thanh. Đồng thời, đây cũng là lúc ý tưởng trò chơi được mài giũa, chuyển từ những phác thảo sơ khai thành một phong cách cụ thể, độc đáo. Bằng cách kết hợp sự hiểu biết về công cụ với những ý tưởng sáng tạo, nhóm đặt nền móng vững chắc cho toàn bộ dự án, đảm bảo rằng trò chơi không chỉ khả thi về mặt kỹ thuật mà còn hấp dẫn về mặt trải nghiệm.

Tiếp nối giai đoạn đầu, cần dành 2 tuần để thu thập và nghiên cứu tài liệu, một bước đi quan trọng nhằm xây dựng nền tảng lý thuyết vững chắc cho trò chơi Đây là thời điểm tập trung tìm hiểu về cơ chế trò chơi, thiết kế giao diện người dùng, và cách tạo ra những yếu tố thử thách hấp dẫn. Các tài liệu tham khảo bao gồm hướng dẫn sử dụng Unity, tài liệu về thiết kế trò chơi, và nghiên cứu các trò chơi tương tự để học hỏi ý tưởng. Đồng thời, nhóm sẽ phân tích và lập kế hoạch cho các yếu tố chính như cách di chuyển của nhân vật, cách hoạt động của cạm bẫy Kết quả của giai đoạn này không chỉ là một bản thiết kế sơ bộ mà còn là sự thống nhất về tầm nhìn sáng tạo, đảm bảo có thể hiểu rõ mục tiêu và phương hướng phát triển của dự án. Đây là bước đệm để biến ý tưởng thành hiện thực một cách hiệu quả và có tổ chức.

Trong 4 tuần tiếp theo, trọng tâm của sẽ là phát triển các thành phần cốt lõi của trò chơi .Hai tuần đầu tiên được dành riêng để xây dựng nhân vật chính, bao gồm,chuyển động và các chức năng cơ bản như di chuyển, tấn công, nhảy hoặc tương tác với môi trường. Đây là bước quan trọng nhằm đảm bảo nhân vật hoạt động mượt mà, mang lại cảm giác điều khiển tự nhiên và thú vị. Sau khi hoàn thiện nhân vật chính, tôi sẽ dành 2 tuần tiếp theo để phát triển các yếu tố thử thách như cạm bẫy và quái. Các cạm bẫy sẽ được thiết kế với mức độ khó tăng dần, tạo động lực cho người chơi cải thiện kỹ năng. Đồng thời, phần thưởng và các chức năng hỗ trợ sẽ được tích hợp để khuyến khích và giữ chân người chơi. Việc cân bằng giữa thử thách và phần thưởng sẽ giúp trò chơi trở nên hấp dẫn và đầy tính giải trí. Đây là giai đoạn quan trọng, đặt nền tảng cho tính chất đặc trưng và lối chơi độc đáo của Game.

Trong 2 tuần tiếp theo, tôi sẽ tập trung thiết kế và lập trình các kẻ thù trong trò chơi để tạo thêm thử thách và chiều sâu cho trải nghiệm người chơi. Quá trình này bắt đầu bằng việc xây dựng hình dáng và hành vi của các kẻ thù, từ những đối thủ di chuyển đơn giản đến những kẻ có khả năng tương tác phức tạp hơn với môi trường và nhân vật chính. Để đảm bảo sự đa dạng, tôi sẽ thiết kế nhiều loại kẻ thù với đặc điểm riêng, như tốc độ di chuyển khác nhau, khả năng tấn công đặc biệt, hoặc cách xuất hiện bất ngờ.

Ngoài ra, yếu tố cân bằng sẽ được đặt lên hàng đầu. Tôi sẽ điều chỉnh mức độ khó của các kẻ thù sao cho phù hợp với tiến độ của người chơi, tạo nên những thử thách vừa đủ mà không gây cảm giác quá áp lực. Việc tích hợp các kẻ thù với cạm bẫy và phần thưởng đã thiết kế trước đó cũng là một phần quan trọng, đảm bảo chúng hoạt động đồng bộ và mang lại trải nghiệm chơi game mượt mà.

Đây là giai đoạn đòi hỏi sự sáng tạo và tư duy logic, bởi tôi cần đảm bảo rằng mỗi kẻ thù không chỉ là thử thách mà còn góp phần làm nổi bật nét đặc trưng của trò chơi. Kết quả mong đợi là một hệ thống kẻ thù vừa hấp dẫn vừa thúc đẩy người chơi khám phá và chinh phục trò chơi.

Trong 4 tuần tiếp theo, tôi sẽ tập trung vào giai đoạn kiểm thử, một bước quan trọng nhằm đảm bảo trò chơi hoạt động ổn định và đạt chất lượng cao. Quá trình kiểm thử sẽ bắt đầu với việc kiểm tra các chức năng chính, như chuyển động của nhân vật, hoạt động của cạm bẫy, và hành vi của kẻ thù. Tôi sẽ sử dụng phương pháp kiểm thử thủ công kết hợp với công cụ tự động để phát hiện lỗi logic, lỗi đồ họa, và các vấn đề về hiệu suất. Đặc biệt, tôi sẽ chú ý đến trải nghiệm người chơi bằng cách thử nghiệm trên các cấu hình máy tính khác nhau để đảm bảo tính tương thích.

Bên cạnh việc xử lý lỗi, tôi sẽ xây dựng khung báo cáo chức năng, trong đó mô tả chi tiết cách các thành phần của trò chơi hoạt động và tương tác với nhau. Báo cáo này không chỉ là tài liệu kỹ thuật mà còn giúp tôi hệ thống hóa toàn bộ quá trình phát triển, từ ý tưởng đến thực thi.

Cuối cùng, 2 tuần cuối cùng sẽ được dành riêng để hoàn thiện báo cáo đồ án. Tôi sẽ tổng hợp tất cả các kết quả, bao gồm phân tích tiến trình, thành tựu đạt được, và những khó khăn đã vượt qua. Đây cũng là lúc tôi đánh giá tổng thể dự án, đưa ra những bài học kinh nghiệm và đề xuất hướng phát triển trong tương lai. Việc hoàn thiện báo cáo không chỉ giúp tôi khép lại dự án một cách chuyên nghiệp mà còn tạo tiền đề cho những dự án tiếp theo.

## 2.2. Phân tích các trò chơi platformer hiện có

### 2.2.1. Phân tích các trò chơi điển hình

Platformer là một trong những thể loại game lâu đời và phổ biến nhất ,có nhiều biến thể và cách tiếp cận khác nhau .Dưới đây là phân tích một số trò chơi tiêu biểu dựa trên lỗi chơi đồ họa ,cơ chế điều khiển và thiết kế cấp độ .

**Super Mario Bros. Series** (Nintendo)

* **Phong cách**: Platformer 2D/3D cổ điển
* **Cơ chế chính**: Nhảy qua chướng ngại vật, thu thập power-up, đánh bại kẻ thù
* **Điểm nổi bật**:
  + Level thiết kế xuất sắc, tạo cảm giác khám phá liên tục
  + Điều khiển mượt mà, dễ tiếp cận nhưng có chiều sâu
  + Power-up sáng tạo giúp mở rộng gameplay (Super Mushroom, Fire Flower...)
* **Ảnh hưởng**: Là nền tảng cho hầu hết các game platformer sau này

**Sonic the Hedgehog Series** (SEGA)

* **Phong cách**: Platformer tốc độ cao
* **Cơ chế chính**: Chạy nhanh, cuộn tròn tấn công, thu thập nhẫn
* **Điểm nổi bật**:
  + Tốc độ di chuyển cao tạo ra cảm giác phấn khích
  + Level thiết kế đa tầng, khuyến khích người chơi khám phá đường đi tối ưu
  + Nhẫn giúp bảo vệ nhân vật khỏi kẻ thù, tạo ra cơ chế rủi ro-thưởng hấp dẫn
* **Ảnh hưởng**: Định hình một nhánh platformer tập trung vào tốc độ

### 2.2.2. Điểm mạnh và điểm yếu của các trò chơi này

Super Mario Bros. Series là một trong những dòng game platformer cổ điển nhất với lối chơi dễ tiếp cận nhưng có chiều sâu, thiết kế màn chơi xuất sắc và cơ chế điều khiển mượt mà. Điểm mạnh của game nằm ở hệ thống power-up đa dạng và thương hiệu mạnh mẽ, tuy nhiên, cốt truyện đơn giản, tuyến tính và thiếu thử thách cho game thủ hardcore là những điểm yếu đáng kể.

Trong khi đó, Sonic the Hedgehog Series nổi bật với tốc độ cao, thiết kế màn chơi đa tầng và hệ thống nhẫn độc đáo, tạo ra một phong cách platformer khác biệt. Tuy nhiên, tốc độ quá nhanh có thể gây khó khăn cho người chơi mới, và chất lượng không đồng đều giữa các phiên bản khiến dòng game này gặp nhiều tranh cãi.

### 2.2.3. Bài học kinh nghiệm cho đề tài

Từ việc phân tích Super Mario Bros, Sonic the Hedgehog Series và Celeste chúng ta có thể rút ra nhiều bài học quan trọng để áp dụng vào quá trình thiết kế và phát triển trò chơi. Đầu tiên, tính dễ tiếp cận và sự cân bằng giữa thử thách và phần thưởng là yếu tố quan trọng giúp giữ chân người chơi. Super Mario đã chứng minh rằng việc thiết kế cơ chế chơi đơn giản nhưng chặt chẽ, kết hợp cùng các phần thưởng như power-ups hoặc đồng xu, có thể tạo ra động lực mạnh mẽ. Vì vậy, trong quá trình phát triển trò chơi, cần đảm bảo rằng người chơi mới có thể dễ dàng tiếp cận, nhưng vẫn có những thử thách hấp dẫn để họ tiếp tục khám phá.

Thứ hai , Sonic the Hedgehog Series dạy chúng ta về tầm quan trọng của chiều sâu trong thiết kế thế giới và cốt truyện. Một thế giới phi tuyến tính với các khu vực liên kết chặt chẽ, bí mật ẩn giấu, và những yếu tố câu chuyện được hé lộ qua môi trường sẽ tăng cường trải nghiệm người chơi. Điều này cho thấy rằng việc đầu tư vào yếu tố khám phá và xây dựng nội dung đa tầng là rất cần thiết, đặc biệt nếu muốn tạo ra trò chơi mang lại cảm giác dài lâu và ý nghĩa.

Bài học khác từ cả hai trò chơi là việc thiết kế phong cách thẩm mỹ và âm thanh đồng bộ với lối chơi. Super Mario dùng đồ họa sáng sủa và nhạc nền vui nhộn để nhấn mạnh tính giải trí, trong khi Sonic the Hedgehog Series lựa chọn phong cách gothic với tông màu tối để phù hợp với bầu không khí bí ẩn và thử thách. Điều này cho thấy sự hài hòa giữa yếu tố nghệ thuật và cơ chế trò chơi có vai trò quan trọng trong việc xây dựng thương hiệu và ấn tượng của trò chơi.

Cuối cùng, cần chú ý đến mức độ khó phù hợp với đối tượng mục tiêu. Super Mario thành công nhờ khả năng mở rộng đối tượng từ trẻ em đến người lớn, trong khi Sonic the Hedgehog Series tạo dựng tên tuổi bằng việc tập trung vào những người chơi yêu thích thử thách. Vì vậy, khi phát triển trò chơi, cần xác định rõ đối tượng người chơi và xây dựng mức độ khó phù hợp để giữ chân họ, đồng thời cung cấp trải nghiệm thú vị và đáng nhớ.

## 2.3. Ý tưởng phát triển trò chơi

### 2.3.1. Cốt truyện và nội dung

Với thời gian 3 tháng, mục tiêu của trò chơi lần này sẽ là hoàn thành với nội dung cốt truyện hướng tới như sau:

**1. Vùng San Hô sụp đổ** : Kael bắt đầu hành trình trên những vùng san hô bị vỡ , nơi các hải tặc đang canh giữ

* Mục tiêu : Thu thập “Ngọc Thủy Tinh” để mở cánh cửa dẫn đến các vùng đất tiếp theo .

**2. Hang Đá Nham Thạch** : Kael đi hòn đảo núi lửa đang hoạt động . Dòng nham thạch chảy không ngừng tạo ra những nền tảng tạm thời mà Kael phải nhảy qua .

* Mục tiêu : Đánh bại 1 con **Rồng Nham Thạch** để lấy chìa khóa ma thuật.

**3.Rừng Mộng Mị :** Một khu rừng bí ẩn nơi Kael bị lạc trong một mê cung ảo giác Cây cối ở đây sống động và có thể tấn công người chơi

* Mục tiêu : Vượt qua các thử thách để tìm thấy kho báu lấy “Chiếc La Bàn Ánh Sáng “ dẫn đường .

**4. Vùng Sương Mù Bóng Tối :** Nơi cuối cùng trước khi trở về Đảo Rồng .Sương mù đen đặc làm giảm tầm nhìn , Kael phải dựa vào âm thanh để tránh các kẻ thù vô hình

* Mục tiêu : Tìm kho báu để lấy được viên ngọc rồng .

|  |  |
| --- | --- |
| Yếu tố | Mô tả |
|  |  |
| Thể loại | Platformer 2D |
| Cơ chế di chuyển | đi ,nhảy ,bám tường ,cơ chế vật lý |
| Cơ chế chiến đấu | Tấn công từ xa |
| Cơ chế tương tác | Thu thập vật phẩm ,cơ chế vật lý |
| Cấu trúc màn chơi | 4 màn chơi thiết kế theo cốt truyện ,có chế độ checkpoint |
| Hệ thống mạng sống | Từ màn 1-> 3 chết một lần chơi lại từ checkpoint ,màn 4 thì không |
| Mục tiêu của màn chơi | Đến được đích ,tiêu diệt được tất cả kẻ thù |

Bảng1: Bảng phân tich gameplay

### 2.3.2. Tính năng chính và cơ chế của GamePlay

**Cơ chế điều khiển nhân vật**

* **Di chuyển trái/phải**: Nhân vật có thể chạy hoặc đi bộ, điều chỉnh tốc độ phù hợp.
* **Nhảy (Jumping Mechanic)**: Bao gồm nhảy cơ bản, nhảy cao khi nhấn giữ nút lâu, và nhảy kép (double jump).
* **Dash (Lao nhanh)**: Cho phép nhân vật tăng tốc trong thời gian ngắn, thường dùng để vượt chướng ngại vật.
* **Wall Jump/Wall Climb**: Nhảy từ tường hoặc leo tường giúp mở rộng khả năng di chuyển.
* **Coyote Time**: Cho phép nhân vật nhảy ngay cả khi vừa rời khỏi mặt đất trong một khoảng thời gian ngắn.
* **Jump Buffer**: Hỗ trợ nhảy nếu người chơi nhấn nút ngay trước khi nhân vật chạm đất.

**Cơ chế vật lý và va chạm**

* **Trọng lực (Gravity)**: Điều chỉnh trọng lực hợp lý để tạo cảm giác điều khiển tự nhiên.
* **Hitbox và va chạm**: Sử dụng **BoxCollider2D** hoặc **PolygonCollider2D** để tránh lỗi xuyên tường.
* **Động lượng (Momentum)**: Nếu muốn tạo cảm giác di chuyển trơn tru như Sonic, động lượng rất quan trọng.

**Cơ chế chiến đấu (nếu có)**

* **Tấn công cơ bản**: Nhân vật có thể nhảy lên đầu kẻ thù (giống Mario) hoặc tấn công trực tiếp.
* **Vũ khí hoặc kỹ năng đặc biệt**: Gồm đòn đánh cận chiến, vũ khí tầm xa, hoặc phép thuật.
* **Hệ thống máu và sát thương**: Có thể dùng thanh máu hoặc cơ chế nhặt nhẫn/vật phẩm bảo vệ (giống Sonic).

**Thiết kế màn chơi và thử thách**

* **Chướng ngại vật**: Gồm gai nhọn, hố sâu, vật thể di chuyển... để thử thách người chơi.
* **Checkpoint**: Lưu vị trí tiến độ để tránh người chơi phải chơi lại từ đầu khi thua.
* **Cơ chế cuộn màn hình (Camera Movement)**: Theo dõi nhân vật hoặc cuộn tự động trong một số màn chơi đặc biệt.

**Power-ups và Collectibles**

* **Vật phẩm tăng sức mạnh**: Như Super Mushroom trong Mario, tăng tốc trong Sonic hoặc dash trong Celeste.
* **Collectibles (Vật phẩm thu thập)**: Tiền xu, kim cương, chìa khóa mở cửa bí mật... giúp tăng giá trị chơi lại.

**Hệ thống AI của kẻ địch và NPC**

* **Kẻ địch có hành vi đa dạng**: Kẻ đi tuần, kẻ đuổi theo, kẻ bay... tạo sự thử thách.
* **NPC hỗ trợ hoặc cung cấp nhiệm vụ**: Giúp tạo chiều sâu cho gameplay.

**Các chế độ chơi bổ sung**

* **Speedrun Mode**: Chế độ đua thời gian dành cho game thủ thích thử thách.
* **Level Editor**: Cho phép người chơi tự tạo màn chơi (như Mario Maker).
* **Co-op/Multiplayer**: Nhiều người chơi cùng hợp tác hoặc thi đấu.

**2.2.4. Điểm khác biệt và sự sáng tạo trong trò chơi**

Trò chơi được thiết kế với những đặc điểm đồng thời gây bất ngờ và thu hút người chơi. Trước hết, các chướng ngại vật luôn thay đổi để duy trì tính tươi mới. Cơ chế gameplay được tích hợp giữa những thách thức đòi hỏi phản xạ nhanh nhạy và các khoảnh khắc vui nhộn nhẹ nhàng. Hệ thống thắng thua được thiết kế có tính động cao, đem lại cảm giác thắng đầy thành tích những không quá khó để gây nản lòng.

Một trong những yếu tố sáng tạo đồng thời làm nên sự khác biệt chính là hệ thống vật phẩm. Các vật phẩm không chỉ mang tính trang trí mà còn yêu cầu người chơi sử dụng một cách thông minh để vượt qua những khó khăn. Đặc biệt, các cấp độ thách thức thường tích hợp nhiều yếu tố độc đáo như môi trường thay đổi liên tục, hoặc chông gai vàng bất ngờ. Tính linh hoạt trong thiết kế khiến người chơi luôn phải điều chỉnh chiến lược.

## 2.4. Các công cụ và công nghệ áp dụng

### 2.4.1. Unity và Visual studio

Gameplay của dự án có thể được thiết kế xoay quanh lối chơi đơn giản nhưng đồng thời mang tính thách thức cao. Gameplay nhấn mạnh vào yếu tố phản xạ nhanh nhạy và kỹ năng điều khiển nhân vật để vượt qua các chướng ngại vật hoặc thu thập vật phẩm. Trò chơi hướng đến phong cách vui nhộn và dễ tiếp cận, nhưng vẫn đầy tính cuốn hút nhờ cân bằng khéo léo giữa thời gian chơi ngắn và mức độ thách thức. Điều này tạo động lực để người chơi quay lại, nhất là những ai tìm kiếm sự giải trí nhẹ nhàng mà vẫn kết hợp tính thống trí.

TextMeshPro (3.0.6): TextMeshPro là một plugin hỗ trợ hiển thị văn bản chất lượng cao trong Unity. Plugin này cung cấp nhiều tùy chọn để tăng tính thẩm mỹ và hiển thị văn bản rõ ràng dù trên các thiết bị có độ phân giải khác nhau. Tính năng nổi bật bao gồm đổ bóng, viền văn bản, đổ màu gradient, và hiệu ứng animation. TextMeshPro không chỉ tăng tính chuyên nghiệp mà còn tối ưu hóa hiệu suất, giúp duy trì tốc độ xử lý của trò chơi.

Timeline (1.6.5): Timeline là một công cụ trong Unity dùng để chỉnh sửa và quản lý hoạt ảnh theo dòng thời gian. Plugin này giúp nhà phát triển dễ dàng tạo ra các cảnh quay phim, sự kiện, hoặc hiệu ứng đồng. Bằng giao diện kéo thả và thao tác trực quan, người dùng có thể dựng Timeline để đồng bộ nhiều yếu tố như animation, âm thanh, và đổi tượng trong game.

Unity UI (6.0.0): Unity UI cung cấp các công cụ và thư viện để xây dựng giao diện người dùng (UI). Plugin này cho phép nhà phát triển tạo ra các thanh điều hướng, nút bấm, khung văn bản, và các thành phần UI tương tác khác. Unity UI có tính tùy biến cao và tự động tích hợp với nhiều thiết bị khác nhau, giúp trò chơi duy trì trải nghiệm tốt trên các nền tảng.

Visual Scripting (1.9.4): Visual Scripting giúp nhà phát triển tạo logic gameplay mà không cần viết mã. Thay vì dòng lệnh, plugin cung cấp giao diện kéo thả và kết nối các node logic. Visual Scripting đặc biệt phù hợp với những người mới làm quen Unity, nhất là trong việc xây dựng các prototype nhanh.

Feature 2D (2.0.1): Feature 2D cung cấp các công cụ chuyên biệt cho việc phát triển game 2D. Tính năng nổi bật bao gồm hệ thống tilemap, sprite động, và animation frame-by-frame. Plugin này giúp giảm thời gian thiết kế và tăng độ chính xác trong việc quản lý tài nguyên 2D.

IDE Plugins: Dự án bao gồm các plugin IDE như Rider, Visual Studio, và VSCode, cho phép nhà phát triển tích hợp trực tiếp các IDE và Unity. Các plugin này cung cấp tính năng gợ lỗi, gợi ý code, và quản lý dự án, tăng độ hiệu quả trong lắp trình.

Input System (1.7.0): Input System quản lý việc tương tác người chơi, hỗ trợ nhiều thiết bị như bàn phím, chuột, tay cầm, và cả màn hình cảm ứng. Hệ thống linh hoạt này giúp nhà phát triển tích hợp dễ dàng và tối đa tính tương thích trên nhiều nền tảng.

Android Logcat (1.4.4): Android Logcat cung cấp công cụ gỡ lỗi cho Android trực tiếp từ Unity. Plugin này giúp hiển thị các log runtime như các vấn đề hiệu suất, đọc logcat, và xử lý sự cố trong khi test game trên thiết bị thật.

# CHƯƠNG 3 :XÂY DỰNG ỨNG DỤNG

**Mô tả bài toán :**

Nhiệm vụ : Thiết kế và xây dựng game Platformer trên nền tảng Unity.

Tên game : Platformer.

Thể loại : Game 2D.

Cơ chế game : Điều khiển nhân vật sang trái ,sang phải ,nhảy ,nhặt kim cương ,7 viên ngọc rồng ,dùng kỹ năng vượt qua các chướng ngại vật.

**Thiết kế tính năng**

* + - Di chuyển của nhân vật
    - Tấn công của nhân vật
    - Animation
    - Quái vật
    - Nhặt viên kim cương
    - Vượt qua chướng ngại vật để tìm kho báu lấy vật phẩm
    - Qua màn

**Phân tích ý tưởng và tổng quan về phát triển game platformer trên unity**

|  |  |
| --- | --- |
| Mục | Chi tiết |
| Tên game | Platformer |
| Thể loại | Platformer(2D) |
| Gameplay chính | Nhảy qua chướng ngại vật ,đánh bại kẻ thù ,thu thập vật phẩm |
| Nhân vật chính | Tên : Dragon  Kỹ năng :nhảy cao ,tấn công ,kỹ năng đặc biệt |
| Cơ chế điều khiển | Phím mũi tên / A-D : di chuyển  Space :nhảy  Kích chuột trái :tấn công |
| Kẻ thù | Lính gác |
| Bẫy | Bẫy phun cung tên ,bẫy lửa ,máy răng cưa , Spikehead |

Bảng 2: Bảng ý tưởng và tổng quan phát triển game

## 3.1. Thiết kế Player

### 3.1.1. Hình ảnh của Player

Player được tải từ trên Unity Asset Store về dự án .Trong thư mục có các hành động di chuyển ,tấn công đầy đủ những hình ảnh để có thể làm một player hoàn chỉnh trong game platformer . Unity Asset Store là một cửa hàng trực tuyến do Unity cung cấp ,nơi các nhà phát triển game có thể mua ,bán và tải về các tài nguyên (asset) phục vụ cho việc phát triển game và ứng dụng Unity.



Hình 3.1: Hình ảnh Player ở trong game

**Phân tích nhân vật**

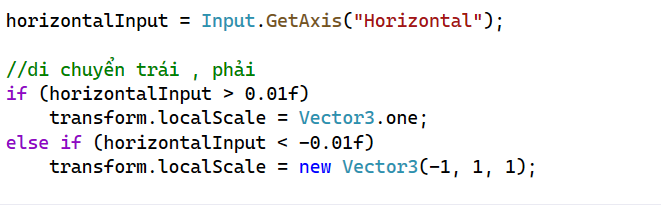
|  |  |
| --- | --- |
| Mục | Chi tiết |
| Tên nhân vật | Dragon |
| Vai trò | Nhân vật chính |
| Tính cách | Dũng cảm ,lạnh lùng và thông minh |
| Kỹ năng chính | Di chuyển , chiến đấu ,kỹ năng đặc biệt |
| Vũ khí | Phun lửa |

Bảng 3: Bảng phân tích nhân vật

### 3.1.2. Di chuyển nhân vật

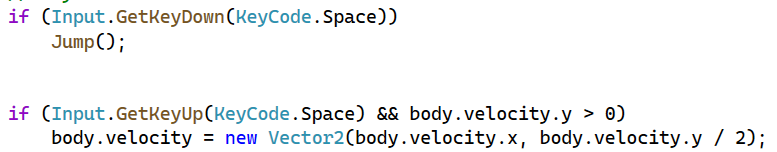
* **Di chuyển sang trái sang phải**

Player có khả năng di chuyên theo phương ngang (theo trục X) sang trái hoặc sang phải để nhặt vật phẩm ,đi tìm kho báu ,tránh né chướng ngại vật . Tại Assets tạo một folder Player trong folder đó tạo 1 scripts có tên là “PlayerMovement“ ,để viết những đoạn code tạo chức năng di chuyển cho player .Nếu như player di chuyển thì player sẽ xoay lại đúng chiều mà player đang di chuyển ,lúc này cần phải thay đổi tỷ lệ trục X của player dựa vào hướng di chuyển (trái hoặc phải) để lật hình ảnh player theo đúng chiều mà player đi .



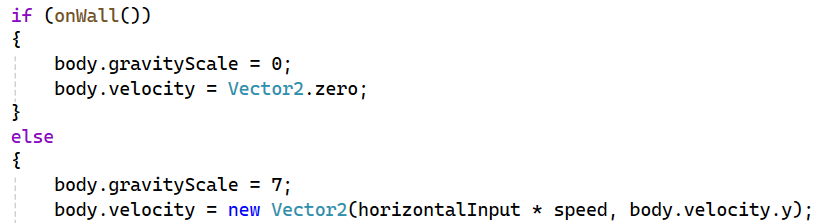
Hình 3.2: Scripts di chuyển của player

Player có thể bật nhảy lên trên không để có thể tránh được chường ngại vật ,vượt qua chướng ngại vật .Nhận phím space từ bên ngoài để bật chức năng nhảy của Pleyer kết hợp với vận tốc của player dương có nghĩa là player đang ở trên không nghĩa là player đang theo trục Y và chia vận tốc của trục Y cho 2 để có thể thấy player dơi xuống mượt mà .



Hình 3.3: Scripts nhảy của player

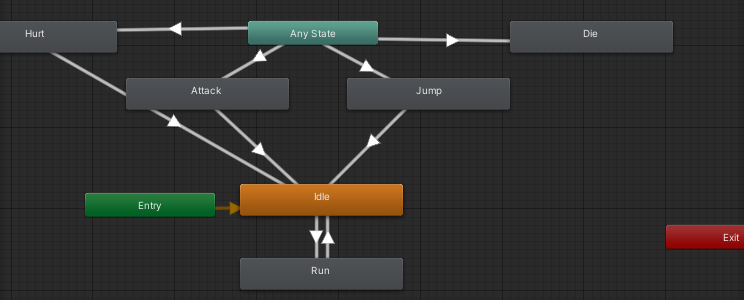
Player có thêm chức năng bám tường , leo trèo để có thể tránh được chướng ngại vật và đi tìm các vật phẩm trong game .Khi player đang bám trên tường thì trọng lực và vận tốc đều trở về 0 thì người chơi mới có thể bám được trên tường và leo trèo để có thể tránh được các chướng ngại vật di chuyển xung quanh.



Hình 3.4: Scripts bám trên tường của player

### 3.1.3. Animation của Player

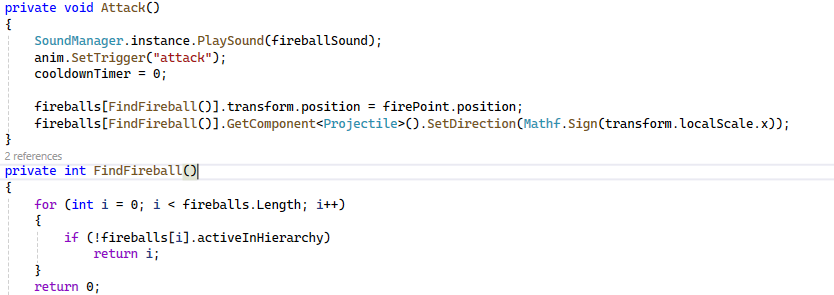
Các hành động của player ở trong game được thực hiện bằng các hoạt ảnh (Animation) mượt mà .Animation giúp cho game sống động và thu hút hơn .Các animation được tạo từ các sprite (hình ảnh) được vẽ theo thứ tự . Animation Controller (trong Unity) được sử dụng để quản lý các animation và chuyển trạng thái giữa chúng. Ở Player ta có hoạt ảnh khi đứng yên, chạy, nhảy,tấn công.



Hình 3.5: Animation của player

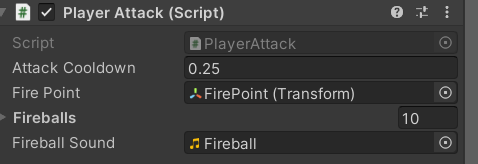
### 3.1.4. Chức năng tấn công của Player

Player có thể tấn công kẻ thù bằng cách phun ra lửa để tiêu diệt kẻ thủ .Khi người chơi nhấn chuột trái thì sẽ phun ra lửa để tấn công tiêu diệt kẻ thù .Mỗi 1 lượt phun lửa là 1 lửa tấn công ,khoảng thời gian giữa các lần tấn công này được tùy chỉnh Khi lửa va chạm với kẻ thù thì lửa biến mất và kẻ thù sẽ mất máu ,ngoài ra khi va chạm với bất vật thể nào ở trong game thì lửa cũng sẽ biến mất ,để tránh việc nó tồn tại mãi ở trong game.



Hình 3.6: Script tấn công của player

Để tránh người chơi nhấn liên tục phím chuột trái tấn công kẻ thù ,thời gian hồi chiêu sẽ được thiết lập để ngăn cản điều này .Thời gian hồi chiêu của mỗi lượt tấn công là 0.25 giây .Cái thời gian hồi chiêu này chúng ta có thể thay đổi được ở trong Unity.(Attack Cooldown)

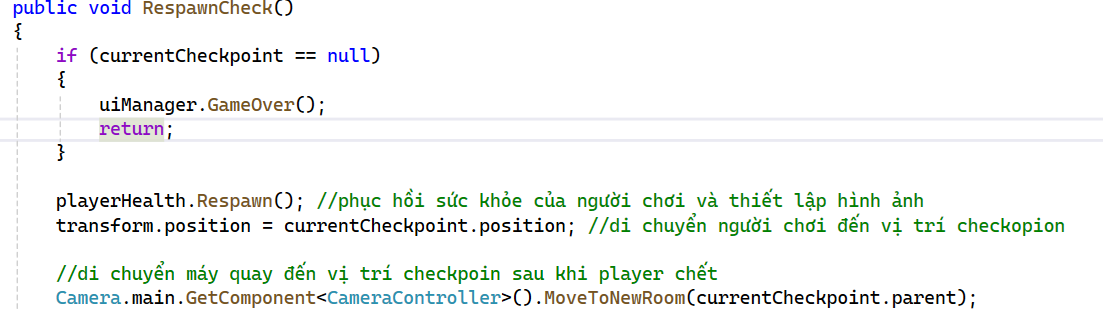


Hình 3.7: Thời gian hồi chiêu của Player

### 3.1.5. Chức năng hồi sinh Player

Khi Player chết ,nếu đã va chạm với checkpoint họ sẽ được hồi sinh tại đó .Nếu như Player không va chạm với checkpoint nào thì game sẽ kết thúc và hiện ra màn GameOver .Khi Player được hồi sinh thì sẽ hồi phục được trạng thái máu của họ , sau khi hồi sinh thì player sẽ dịch chuyển đến vị trí checkpoint. Chức năng này sẽ tạo cơ hội cho người chơi để có thể di chuyển player tránh các chướng ngại vật để có thể qua màn chơi .

Khi player chạm vào checkpoint thì collider chỉ có tác dụng 1 lần tránh việc checkpoint được ghi đè mỗi player đi qua



Hình 3.8: Hồi sinh Player sau khi chết trong game

### 3.1.6. Máu của Player

Hệ thống máu của người chơi là một yếu tố quan trọng trong game platformer, ảnh hưởng đến độ khó, trải nghiệm và cách người chơi tương tác với thử thách. Dưới đây là một số phương pháp phổ biến để thiết kế hệ thống máu cho nhân vật chính.

****

Hình 3.9: Số lượng máu của player trong game

## 3.2. Kẻ địch ,bẫy, vật phẩm và kho báu

### 3.2.1. Kẻ địch

Kẻ địch là các đối tượng di chuyển ở trong game gây trở ngại cho người chơi vượt ải .Kẻ địch sẽ chủ động tấn công kẻ địch trong phạm vi cho phép để bảo vệ rương kho báu ,khi bị người chơi tấn công lượng máu sẽ giảm và khi về 0 thì kẻ địch sẽ biến mất và hiện ra 1 vật phẩm để đủ điều kiện qua màn .

* **Thiết kế kẻ địch**

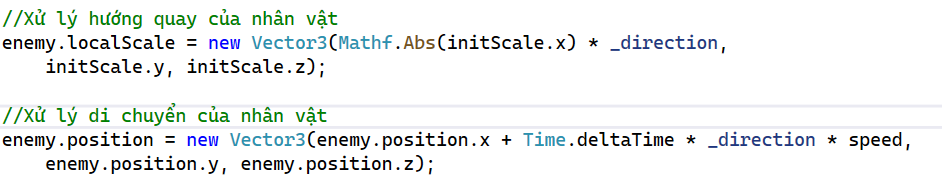
Kẻ địch được tải từ trên Unity Asset Store về dự án .Trong thư mục có các hành động di chuyển ,tấn công đầy đủ những hình ảnh để có thể làm một player hoàn chỉnh trong game platformer . Unity Asset Store là một cửa hàng trực tuyến do Unity cung cấp ,nơi các nhà phát triển game có thể mua ,bán và tải về các tài nguyên (asset) phục vụ cho việc phát triển game và ứng dụng Unity.



Hình 3.10 : Hình ảnh của kẻ địch trong game

* **Di chuyển của kẻ địch**

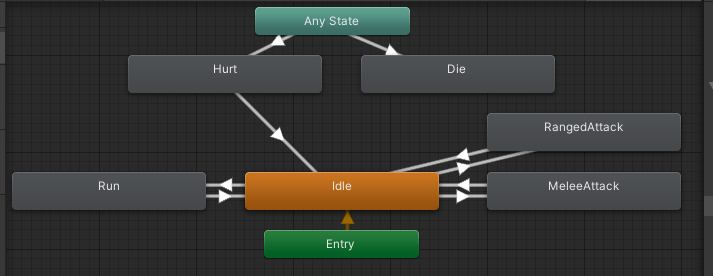
Kẻ địch có khả năng di chuyển theo phương ngang (theo trục X) sang trái hoặc sang phải để thực hiện nhiệm vụ tuần tra ,khi gặp đối tượng kẻ địch sẽ tấn công bằng vũ khí của mình .



Hình 3.11 : Script di chuyển của kẻ địch trong game

* **Animation của kẻ địch**

Các hành động của kẻ địch ở trong game được thực hiện bằng các hoạt ảnh (Animation) mượt mà .Animation giúp cho game sống động và thu hút hơn .Các animation được tạo từ các sprite (hình ảnh) được vẽ theo thứ tự . Animation Controller (trong Unity) được sử dụng để quản lý các animation và chuyển trạng thái giữa chúng.



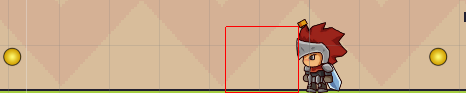
Hình 3.12 : Animation của kẻ địch

* **Chức năng tấn công của kẻ địch**

Chúng ta có 2 dạng kẻ địch : một là kẻ địch cận chiến ,hai là kẻ địch đánh xa .Kẻ địch sẽ tấn công phun ra lửa hoặc dùng đao tấn công người chơi nằm trong vùng hoạt động tấn công của nó .

* Kẻ địch cận chiến

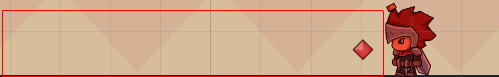
Kẻ địch sẽ di chuyển tuần tra trong vùng giới hạn trên bản đồ ,nếu gặp đối tượng nằm trong vùng tấn công ,nó sẽ dùng đao để tấn công người chơi .Người chơi chỉ cần va chạm vào kẻ địch trong vùng tấn công sẽ bị mất máu. Chúng ta có thể giới hạn được vùng kiểm soát của kẻ địch trên bản đồ nhờ LeftEdge và RightEdge .



Hình 3.13 : Vùng di chuyển của kẻ địch

Kẻ địch tấn công từ xa

Kẻ địch có thể phát hiện người chơi trong pham vi nhất định và phun lửa khi đến lượt tấn công. Khi người chơi trúng đạn của kẻ thù sẽ mất máu.



Hình 3.14 : Vùng tấn công của kẻ địch

### 3.2.2. Bẫy trong game

* **ArrowTraps (bẫy phun cung tên)**

Bẫy là các yếu tố nguy hiểm được đặt trong môi trường game để gây khó khăn cho người chơi. Chúng có thể gây sát thương hoặc làm mất máu nhất định của nhân vật khi người chơi chạm vào.

ArrowTraps đây là bẫy mũi tên tự động .Bẫy này sẽ bắn mũi tên theo chu kỳ cố định bằng cách sử dụng object pooling để tối ưu hóa hiệu suất.



Hình 3.15 : Hình ảnh bẫy ArrowTraps đang hoạt động

* **FireTrap (bẫy lửa)**

Khi người chơi chạm vào bẫy nó sẽ chuyển sang màu đỏ để cảnh cáo .Sau một thời gian 2 giây bẫy sẽ được kích hoạt (activatitioneDelay) và gây sát thương cho Player

Bẫy sẽ hoạt động trong khoảng thời gian là 2 giây (activeTime) sau đó tắt đi và quay trở lại trạng thái ban đầu .



Hình 3.16 : Hình ảnh bẫy firetrap đang hoạt động

**Enyme \_Sideways(máy răng cưa)**

Bẫy răng cưa là một loãi bẫy nguy hiểm trong trò chơi có thể di chuyển qua lại theo trục X trong phạm vi nhất định và gây sát thương khi chạm vào người chơi. Nó hoạt động liên tục ,tạo ra một mối đe dọa thường trực trong màn chơi buộc người chơi phải căn thời gian chính xác để né tránh.



Hình 3.17 : Hình ảnh bẫy Enyme \_Sideways

* **Spikehead**

Spikehead là một loại bẫy nguy hiểm có khả năng phát hiện người chơi trong phạm vi nhất định và lao nhanh về phía họ theo bốn hướng chính :trái ,phải ,lên hoặc xuống Khi xuất hiện ,bẫy sẽ đứng yên và liên tục quét xunh quanh để tìm mục tiêu .Khi phát hiện người chơi ,nó sẽ lập tức kích hoạt chế độ tân công và di chuyển với tốc độ cao đến vị trí mục tiêu .Khi va chạm với một vật thể hoặc người chơi , Spikehead sẽ dừng lại ngay lập tức .Bẫy này tạo ra sự bất ngờ và đòi hỏi người chơi phải nhanh chóng né tránh để không bị tấn công .



Hình 3.18 : Hình ảnh bẫy Spikehead

### 3.2.3. Vật phẩm trong Game

* Ngọc Thủy Tinh

Ngọc Thủy Tinh được xuất hiện ở màn 1 và được đặt rải rác trong khu vực ,buộc người chơi phải khám phá và vượt qua nhiều chướng ngại vật để thu thập chúng .Trên đường đi người chơi có thể gặp phải các cặm bẫy như máy răng cưa ,bẫy lửa hoặc là kẻ thù tuần tra ,…

****

Hình 3.19 : Hình ảnh viên Ngọc Thủy Tinh

* **Chìa khóa ma thuật**

Trong màn chơi thứ 2 này ,người chơi phải hạ gục quái vật để lấy được chìa khóa ma thuật ,vật thể duy nhất có thể mở cánh cổng dẫn đến màn tiếp theo .Chìa khóa ma thuật nằm bên trong con quái vật ,phải hạ gục được nó thì chìa khóa mới xuất hiện Người chơi cần vận dụng kỹ năng chiến đấu ,né tránh đòn tấn công và tìm ra điểm yếu của kẻ thù để giành được chiến thắng .Khi quái vật bị đánh bại ,chìa khóa sẽ xuất hiện cho phép người chơi mở cổng và tiếp tục hành trình khám phá.



Hình 3.20 : Hình ảnh chiếc chìa khóa ma thuật

* **La bàn ánh sáng**

Chiếc la bàn ánh sáng được xuất hiện trong màn 3 ,người chơi phải vượt qua hàng loạt các chứng ngại vật nguy hiểm để tìm đến kho báu bí ẩn ,nơi cất giấu chiếc la bàn ánh sáng ,chìa khóa giúp người chơi tiến vào màn tiếp theo .Hành trình không hề dễ dàng khi phía trước là các cạm bẫy chết người ,người chơi cần phải khéo léo nhảy qua các chướng ngại vật né tránh bẫy ,chỉ khi chạm tay vào chiếc la bàn ánh sáng người chơi mới có thể mở cánh cổng huyển bí bước tiếp vào thử thách đầy hấp dẫn.



Hình 3.21 : Hình ảnh chiếc la bàn ánh sáng

* **Viên ngọc rồng**

Trong màn chơi cuối cùng này ,người chơi phải vượt qua hàng loạt chướng ngại vật nguy hiểm và đánh bại con quái vật hung tợn để tìm thấy kho báu bí ẩn ,nơi cất dấu viên ngọc rồng .Chỉ khi chiến đấu dũng cảm ,sử dụng kỹ năng một cách điêu luyện mới có thể đánh bại được kẻ thù mở rương kho báu và sở hữu viên ngọc rồng để chiến thắng trò chơi này .



Hình 3.22 : Hình ảnh viên ngọc rồng

### 3.2.4. Kho báu

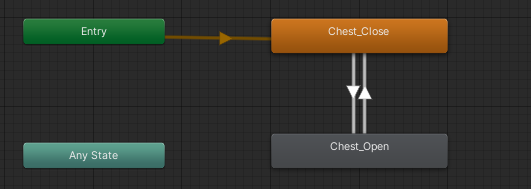
Giữa lòng hang động bí ẩn ,kho báo huyền thoại lấy lánh trong ánh sáng huyền thoại mờ ảo ,tỏa ra một vẻ đẹp mê hoặc .Những vật phẩm như : chìa khóa ma thuật ,chiếc la bàn ánh sáng và viên ngọc rồng đều được cất chứa trong đó.Tất cả vật phẩm đều quý giá , nhưng thứ quý giá nhất là viên ngọc rồng thần bí.

Hình 3.23 : Hình ảnh kho báu

**Animation của kho báu**

**Animation** là quá trình tạo ra chuyển động cho các đối tượng bằng cách hiển thị liên tiếp các hình ảnh hoặc khung hình (frames) theo thời gian. Trong game, animation giúp nhân vật, môi trường và hiệu ứng trở nên sống động, tăng tính chân thực và hấp dẫn cho trò chơi.

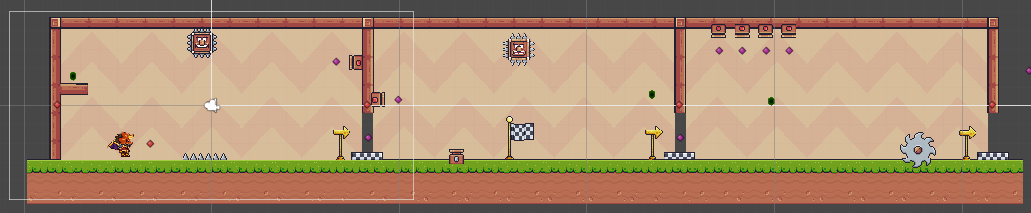


Hình 3.24 : Hình ảnh animation của rương

## 3.3. Các màn chơi trong game

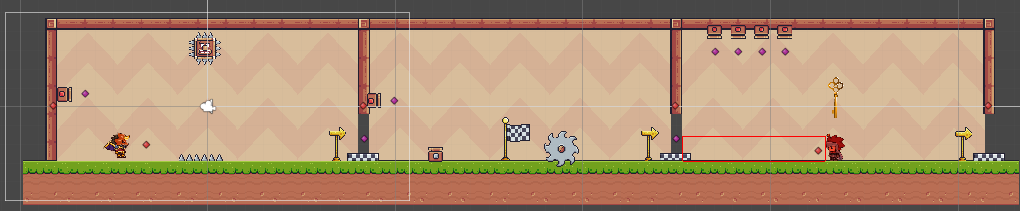
### 3.3.1. Màn 1:Vượt qua các chướng ngại vật và thu thập 3 viên ngọc thủy tinh

* Mục tiêu :Người chơi phải vượt qua các chướng ngại vật nguy hiểm và thu thập được 3 viên ngọc thủy tinh.
* Chướng ngại vật :Bao gồm các bẫy như :bẫy phun cung tên ,bẫy lửa máy răng cưa và Spikehead.
* Mở khóa cánh cổng :Khi người chơi thu thập được đủ 3 viên kim cương thì cánh cửa màn tiếp theo sẽ hiện ra .



Hình 3.25 : Hình ảnh map màn 1

### 3.3.2. Màn 2 :Tấn công quái thú để lấy chìa khóa ma thuật

* Mục tiêu : Người chơi cần đánh bại con quái mạnh mẽ để lấy chìa khóa ma thuật mở cánh cổng đi tiếp.
* Quái vật : Quái vật có sức mạnh chiến thuật phức tạp ,yêu cầu người chơi sử dụng kỹ năng và chiến thuật chiến đấu đặc biệt để hạ gục .
* Chìa khóa ma thuật :Sau khi đánh bại quái vật ,người chơi sẽ nhận được chìa khóa ma thuật ,một vật phẩm quan trọng để tiếp tục màn mới

Hình 3.26 : Hình ảnh map màn 2

### 3.3.3.Màn 3 :Tấn công quái thú và vượt qua các chướng ngại vật để tìm thấy kho báu lấy chiếc la bàn ánh sáng .

* Mục tiêu : Người chơi cần đánh bại con quái mạnh mẽ và vượt qua được các chướng ngại vật để tìm thấy kho báu lấy chiếc la bàn ánh sáng.
* Quái vật : Di chuyển tuần tra và sẽ tấn công người chơi ,quái màn này rất mạnh có kỹ năng cao
* Chiếc la bàn ánh sáng :Sau khi đánh bại quái và vượt qua các chướng ngại vật ,người chơi sẽ tìm thấy kho báu trong đó có chiếc la bàn ánh sáng ,một vật phẩm quan trọng để đi đến màn cuối cùng .



Hình 3.27 : Hình ảnh map màn 3

### 3.3.4. Tấn công quái thú và vượt qua các chướng ngại vật để tìm thấy kho báu lấy viên ngọc rồng .

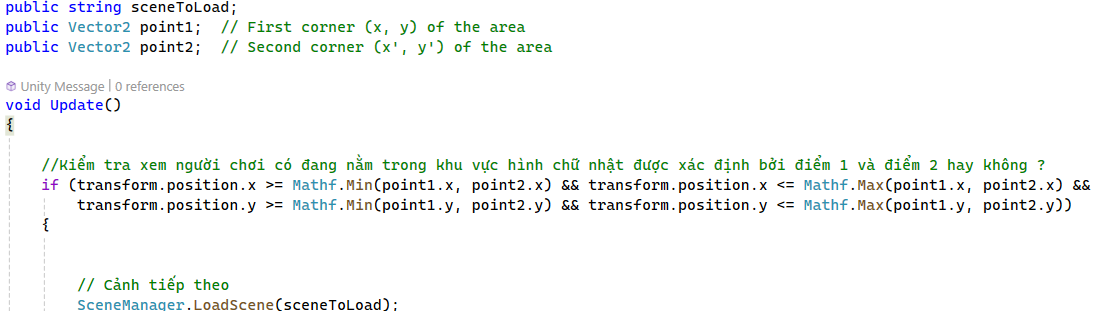
* Mục tiêu : Người chơi cần đánh bại con quái mạnh mẽ và vượt qua được các chướng ngại vật để tìm thấy kho báu lấy viên ngọc rồng.
* Quái vật : Di chuyển tuần tra và sẽ tấn công người chơi ,quái màn này rất mạnh có kỹ năng cao .
* Viên ngọc rồng : Sau khi đánh bại quái và vượt qua các chướng ngại vật người chơi sẽ tìm thấy kho báu trong đó có viên ngọc rồng và giành được chiến thắng .



Hình 3.28 : Hình ảnh map màn 4

### 3.3.5. Qua màn

Người chơi có thể hoàn thành một màn chơi và chuyển sang màn chơi tiếp theo. Khi người chơi đã đi đến cuối bản đồ hoặc tìm thấy được vật phẩm thì đấy chính là cánh cửa để người chơi dịch chuyển sang màn mới. Khi người chơi chạm vào thì sẽ được chuyển qua một bản đồ tiếp theo. Lúc này, trò chơi lấy chỉ số cấp độ từ SceneManager, nếu người chơi hoàn thành hết các cấp độ thì người chơi đã thắng.



Hình 3.29 : Script chuyển màn

## 3.4. Thiết kế Menu ,GameOver ,GameWin

Trong một trò chơi platformer, việc thiết kế các màn hình **Menu**, **Game Over**, và **Game Win** đóng vai trò quan trọng trong việc nâng cao trải nghiệm người chơi. Các màn hình này không chỉ giúp điều hướng mà còn tạo cảm xúc cho người chơi qua cách trình bày hình ảnh, âm thanh và hiệu ứng. Unity cung cấp nhiều công cụ mạnh mẽ để giúp phát triển các màn hình này một cách dễ dàng và chuyên nghiệp.

### 3.4.1. Thiết kế menu chính

Chức năng

Cung cấp các nút bấm: **Bắt đầu trò chơi, cài đặt, thoát**.

Có thể thêm hiệu ứng động hoặc nhạc nền để thu hút người chơi.

Cách triển khai trong Unity

**Sử dụng UI Canvas:** Dùng Canvas để tạo giao diện người dùng.

**Thêm nút bấm (Button):** Tạo các nút như Start, Options, Quit.

**Gắn sự kiện (Event):** Dùng OnClick() để gọi các chức năng như chuyển cảnh (SceneManager.LoadScene()).



Hình 3.30 : Màn hình Menu chính trong game

**3.4.2. Thiết kế màn hình GameOver**

Chức năng

Thông báo rằng người chơi đã thua.

Cung cấp các tùy chọn: **chơi lại, quay về menu chính ,thoát.**

Cách triển khai

**Hiển thị thông báo Game Over**: Dùng TextMeshPro hoặc Text để hiện thông điệp "Game Over".

**Nút Chơi Lại (Restart):** Dùng SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex) để reset màn chơi.

**Hiệu ứng động:** Thêm hiệu ứng xuất hiện chậm để tăng cảm xúc.

**Nhạc nền:** Dừng nhạc nền hiện tại và phát nhạc thất bại.



Hình 3.31 : Màn hình MenuOver trong game

**3.4.3. Thiết kế màn Win**

Chức năng

Chúc mừng người chơi hoàn thành màn chơi hoặc trò chơi.

Cách triển khai

**Hiển thị thông báo "You Win!":** Dùng UI Text hoặc Image.

**Hiệu ứng chiến thắng**: Thêm Particle System hoặc Animation để tạo cảm giác vui vẻ.

**Chuyển màn chơi kế tiếp:** Dùng SceneManager.LoadScene(nextLevelIndex) nếu game có nhiều màn chơi.

**Nhạc nền:** Phát nhạc chiến thắng để tạo cảm giác thành công.

****

Hình 3.32 : Màn hình Menuwin trong game

# CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1. Đánh giá kết quả

Phát triển game là một quá trình sáng tạo đòi hỏi sự kết hợp giữa tư duy kỹ thuật và nghệ thuật. Sử dụng Unity để làm game mang lại nhiều lợi ích đáng kể, từ giao diện thân thiện, hỗ trợ đa nền tảng, đến cộng đồng người dùng rộng lớn. Unity cung cấp các công cụ mạnh mẽ giúp nhà phát triển dễ dàng xây dựng và tối ưu hóa game của mình, dù là game 2D đơn giản hay 3D phức tạp. Với sự hỗ trợ từ ngôn ngữ C# và kho tài nguyên phong phú từ Unity Asset Store, các nhà phát triển có thể nhanh chóng tạo ra các trò chơi đa dạng, hấp dẫn và đầy thử thách.

Sau khi hoàn thành đề tài “Xây dựng game platformer sử dụng Unity” ,chúng ta đã đạt được những kết quả quan trọng :

* Xây dựng thành công một trò chơi platformer với đầy đủ các tính năng cơ bản như di chuyển, nhảy, va chạm và cơ chế vật lý.
* Áp dụng kỹ thuật lập trình C# trên Unity để quản lý đối tượng, xử lý sự kiện và tạo gameplay mượt mà.
* Tối ưu hiệu suất và trải nghiệm người chơi
* Thiết kế màn chơi hợp lý, cân bằng độ khó để giữ chân người chơi.

## 4.2. Hướng phát triển trong tương lai

Dựa trên nền tảng đã có, có thể tiếp tục mở rộng dự án theo các hướng sau:

* Mở rộng nội dung game: Thêm nhiều màn chơi, nhân vật, kỹ năng và vật phẩm mới.
* Hỗ trợ đa nền tảng: Xuất bản game lên PC, mobile hoặc console.
* Thương mại hóa game: Đăng tải lên Steam, Google Play hoặc App Store để tiếp cận người dùng rộng hơn.
* Lưu điểm của người chơi sau khi chết hay dừng game.
* Phát triển thêm chức năng dừng game .
* Bổ xung thêm các loại bẫy và kẻ địch
* Hệ thống nhiệm vụ xếp hạng

# TÀI LIỆU THAM KHẢO

1. Unity Documentation: https://docs.unity3d.com/
2. Unity 2D Tutorials: https://unity.com/learn/pathways/unity-2d-game-creation
3. Unity, “Animation System Overview,” 25 04 2018. [Trực tuyến]. Available:https://docs.unity3d.com/2020.1/Documentation/Manual/AnimationOverview.html https://docs.microsoft.com/en- us/dotnet/csharp/
4. Brackeys (YouTube Channel - Unity Tutorials): https://[www.youtube.com/c/Brackeys/](http://www.youtube.com/c/Brackeys/)
5. GameDev.tv (Unity Courses): https://[www.gamedev.tv/p/unity-2d-developer](http://www.gamedev.tv/p/unity-2d-developer)
6. Asset Store (Unity): https://assetstore.unity.com/
7. Unity Forums: https://forum.unity.com/
8. 2D Toolkit (Asset Store): https://assetstore.unity.com/packages/tools/2d/2d- toolkit-2914
9. Cinemachine (Unity): [https://docs.unity3d.com/Packages/com.unity.cinemachine@2.9/manual/index.html](https://docs.unity3d.com/Packages/com.unity.cinemachine%402.9/manual/index.html)
10. Tiled Map Editor: https://[www.mapeditor.org/](http://www.mapeditor.org/)
11. Aseprite (Pixel Art Software): https://[www.aseprite.org/](http://www.aseprite.org/)

# 