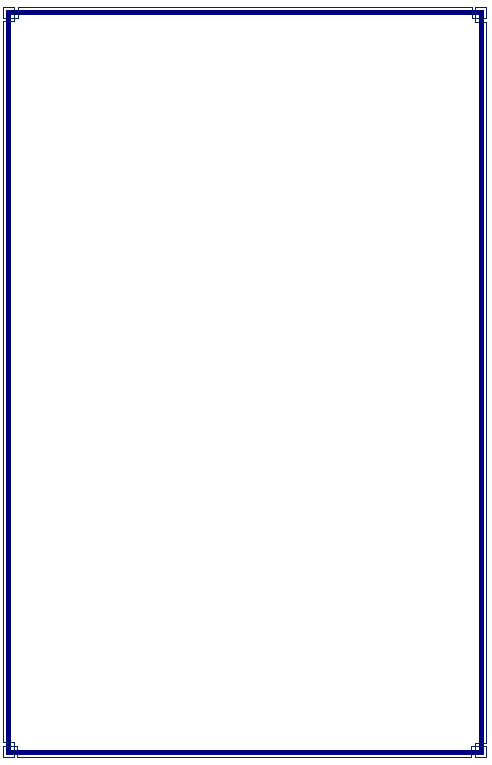
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG GAME 2D “GREAT WAR OF THE MOON”**

**BẰNG UNITY**

|  |  |
| --- | --- |
| **CBHD:** | ThS. Lê Thị Thủy |
| **Sinh viên:** | Nguyễn Quốc Khánh |
| **Mã số sinh viên:** | 2020607299 |
|  |
|  |

*Hà Nội, năm 2024*

# MỤC LỤC

MỤC LỤC 2

DANH MỤC HÌNH ẢNH 5

DANH MỤC BẢNG BIỂU 7

DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ TỪ VIẾT TẮT 8

LỜI CẢM ƠN 10

LÝ DO CHỌN ĐỀ TÀI 11

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI, MÔI TRƯỜNG VÀ CÔNG CỤ LẬP TRÌNH 13

1.1 Giới thiệu đề tài 13

1.2: Giới thiệu về Indie Game 14

1.2.1: Game Indie là gì ? 14

1.2.2: Lịch sử phát triển của game Indie 14

2.1.3: Ưu điểm 15

2.1.4: Nhược điểm 16

1.3 Giới thiệu phần mềm Unity 16

1.3.1 Ưu nhược điểm của phần mềm 17

1.3.2 Hướng dẫn tải và cài đặt (môi trường windows) 19

1.3.3 Giao diện phần mềm 21

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ GAME GREAT WAR OF THE MOON 28

1.4 Giới thiệu tổng quan 28

1.4.1 Thông tin game 28

1.4.2 Luật chơi 28

1.4.3 Đối tượng chơi 28

1.4.4 Nền tảng 29

1.4.5: Kịch bản game 29

1.4.5: Tương tác và điều khiển game 32

1.5 Storyboard 33

1.5.1 Sơ đồ các màn hình 33

1.5.2 Màn hình bắt đầu 33

1.5.3 Màn hình chơi game 34

1.5.4 Màn hình game over 35

1.6 Tài nguyên 35

1.6.1 Hình ảnh 35

1.6.2 Màu sắc 36

1.6.3 Âm thanh 36

1.6.4 Font chữ 37

1.6.5 Hiệu ứng và animation 37

CHƯƠNG 3. CÀI ĐẶT GAME VÀ KẾT QUẢ SẢN PHẨM 38

3.1. Các kỹ thuật thực hiện 38

3.1.1. Tạo các Prefab cho game (Game UI) 38

3.1.2. Sprites và Animation: 38

3.1.3. Collision Detection 39

3.1.4. Tilemap và Tilemap collider 40

3.1.5. UI và HUD 40

3.1.6. Code chức năng chính của game 41

3.2. Sản phẩm màn hình 46

3.2.1 Màn hình chính 46

3.2.2 Màn hình chọn nhân vật chơi game 46

3.2.3 Màn hình chơi game 47

3.2.4 Màn hình game over 48

3.2.5: Màn hình shop skin 48

3.3: Kiểm thử game 49

*1.* *Phân loại lỗi* 53

*2.* *Quy trình xử lý lỗi* 54

KẾT LUẬN 62

TÀI LIỆU THAM KHẢO 63

# DANH MỤC HÌNH ẢNH

[Hình 1: Hình ảnh minh họa Indie Game 11](#_Toc168177456)

[Hình 2: Các hệ điều hành Unity nhắm tới. 14](#_Toc168177457)

[Hình 3: Đăng nhập Unity Hub 16](#_Toc168177458)

[Hình 4: Kích hoạt Unity Hub 17](#_Toc168177459)

[Hình 5: Chọn phiên bản để cài đặt công cụ Unity 17](#_Toc168177460)

[Hình 6: Giao diện chính của phần mềm Unity 18](#_Toc168177461)

[Hình 7: Giao diện project 18](#_Toc168177462)

[Hình 8: Giao diện màn hình Hierarchy 20](#_Toc168177463)

[Hình 9: Giao diện màn hình Inspector. 20](#_Toc168177464)

[Hình 10: Giao diện màn hình Scene. 21](#_Toc168177465)

[Hình 11: Giao diện màn hình Gameview. 22](#_Toc168177466)

[Hình 12: Storyboard của game 30](#_Toc168177467)

[Hình 13: Màn hình bắt đầu 30](#_Toc168177468)

[Hình 14: Thanh HP của nhân vật 31](#_Toc168177469)

[Hình 15: Thanh exp của nhân vật 31](#_Toc168177470)

[Hình 16: Số quái đã tiêu diệt 31](#_Toc168177471)

[Hình 17: Số vàng đang sở hữu 31](#_Toc168177472)

[Hình 18: Màn hình chơi game 31](#_Toc168177473)

[Hình 19: Màn hình thua 32](#_Toc168177474)

[Hình 20: Hình ảnh các asset và vật phẩm 33](#_Toc168177475)

[Hình 21: Màu sắc trên nền tối thân thiện với mắt 33](#_Toc168177476)

[Hình 22: Âm thanh của trò chơi 34](#_Toc168177477)

[Hình 23: Hình ảnh font chữ 34](#_Toc168177478)

[Hình 24: Các animation có trong game 34](#_Toc168177479)

[Hình 25: Giao diện các prefab của game 35](#_Toc168177480)

[Hình 26: UI và HUD trong game 38](#_Toc168177481)

[Hình 27: Phần code thiết kế nhân vật 40](#_Toc168177482)

[Hình 28: Phần code tấn công quái 41](#_Toc168177483)

[Hình 29: Phần code camera di chuyển theo nhân vật 42](#_Toc168177484)

[Hình 30: Hình ảnh màn hình chính. 43](#_Toc168177485)

[Hình 31: Hình ảnh chọn nhân vật 43](#_Toc168177486)

[Hình 32: Hình ảnh màn hình chơi game chính 1 44](#_Toc168177487)

[Hình 33: Màn hình chọn vũ khí trong game 44](#_Toc168177488)

[Hình 34: Hình ảnh màn hình game over 45](#_Toc168177489)

[Hình 35: Hình ảnh màn hình shop skin 45](#_Toc168177490)

# DANH MỤC BẢNG BIỂU

[Bảng 1.4.3.1: Mô tả nhân vật chính 29](#_Toc168242376)

[Bảng 1.4.3.1: Mô tả các quái vật 31](#_Toc168242377)

[Bảng 1.4.3.1: Mô tả các vũ khí 31](#_Toc168242378)

[Bảng 3.3.1: Kế Hoạch Kiểm Thử 49](#_Toc168242379)

[Bảng 3.3.2.1: Yêu cầu phần cứng 49](#_Toc168242380)

[Bảng 3.3.2.2: Yêu cầu phần mềm 49](#_Toc168242381)

[Bảng 3.3.2.3: Công cụ kiểm thử 50](#_Toc168242382)

[Bảng 3.3.4.1: Kiểm thử chức năng 51](#_Toc168242383)

[Bảng 3.3.4.2: Kiểm thử giao diện 52](#_Toc168242384)

[Bảng 3.3.5: Đánh giá rủi ro 53](#_Toc168242385)

[Bảng 3.3.6.1: Phân loại lỗi 53](#_Toc168242386)

[Bảng 3.3.7.1: Testcase 1 54](#_Toc168242387)

[Bảng 3.3.7.2: TestCase 2 55](#_Toc168242388)

[Bảng 3.3.7.3: Testcase 3 56](#_Toc168242389)

[Bảng 3.3.7.4: Testcase 4 57](#_Toc168242390)

[Bảng 3.3.7.5: Testcase 5 58](#_Toc168242391)

[Bảng 3.3.7.6: Testcase 6 58](#_Toc168242392)

[Bảng 3.3.7.7: Testcase 7 59](#_Toc168242393)

[Bảng 3.7.7.8: Testcase 8 60](#_Toc168242394)

[Bảng 3.3.7.9: Testcase 9 61](#_Toc168242395)

# DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ TỪ VIẾT TẮT

 **API** - Application Programming Interface : là một tập hợp các giao diện và giao thức mà phần mềm sử dụng để giao tiếp với các thành phần khác nhau của hệ thống hoặc với các hệ thống khác. API cho phép các nhà phát triển sử dụng các chức năng có sẵn mà không cần phải viết mã từ đầu, giúp tiết kiệm thời gian và công sức.

 **IDE** - Integrated Development Environment : là một phần mềm cung cấp môi trường lập trình toàn diện với các công cụ nhằm hỗ trợ lập trình viên phát triển phần mềm hiệu quả hơn. Ví dụ: Visual Studio, Eclipse, và IntelliJ IDEA.

 **UI** - User Interface : là giao diện người dùng, bao gồm tất cả các yếu tố mà người dùng có thể tương tác trực tiếp với phần mềm hoặc thiết bị, chẳng hạn như các nút, biểu đồ, và menu. UI được thiết kế để giúp người dùng tương tác dễ dàng và hiệu quả với hệ thống.

 **UX** - User Experience : là trải nghiệm người dùng, đề cập đến cảm nhận tổng thể của người dùng khi sử dụng sản phẩm hoặc dịch vụ. UX bao gồm nhiều khía cạnh như thiết kế, chức năng, tính tiện dụng, và cảm xúc mà người dùng trải qua khi tương tác với sản phẩm.

 **SDK** - Software Development Kit : là bộ công cụ phát triển phần mềm bao gồm các công cụ, thư viện, tài liệu và mẫu mã nguồn giúp lập trình viên phát triển ứng dụng cho một nền tảng cụ thể.

 **VR** - Virtual Reality : là công nghệ thực tế ảo, tạo ra một môi trường ảo mà người dùng có thể tương tác và trải nghiệm như thật. Thực tế ảo sử dụng các thiết bị như kính VR và bộ điều khiển để tạo ra trải nghiệm chân thực và sống động.

 **AR** - Augmented Reality: là công nghệ thực tế tăng cường, bổ sung các yếu tố ảo vào thế giới thực thông qua thiết bị như điện thoại thông minh hoặc kính AR. AR nâng cao trải nghiệm người dùng bằng cách thêm các thông tin ảo vào môi trường thực tế.

 **Singleton** : là một mẫu thiết kế (design pattern) được sử dụng để hạn chế việc khởi tạo một lớp nào đó chỉ có một đối tượng duy nhất tồn tại trong suốt vòng đời của ứng dụng. Điều này có nghĩa là bất kỳ lúc nào bạn yêu cầu một đối tượng của lớp đó, bạn sẽ luôn nhận được cùng một đối tượng.

# LỜI CẢM ƠN

Em xin trân trọng cảm ơn ThS.Lê Thị Thủy đã tận tình dẫn dắt, định hướng từ những ngày đầu tiên thực hiện Đồ án tốt nghiệp. Xuất phát từ những ý tưởng ban đầu, vạch ra kế hoạch và thực hiện đều được sự tư vấn, hướng dẫn chi tiết của thầy để em có thể đưa ra bản Đồ án tốt nghiệp hoàn thiện cuối cùng.

Em xin trân trọng cảm ơn các thầy, cô giáo trong khoa Công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội đã đưa ra những góp ý, phản biện hữu ích và sâu sắc giúp cho Đồ án tốt nghiệp và sản phẩm của em được hoàn thiện và đạt giá trị ở mức cao hơn.

Em xin chân thành cảm ơn trường Đại học Công Nghiệp Hà Nội nói chung và khoa Công nghệ thông tin nói riêng, đã luôn tạo điều kiện tốt nhất về cả cơ sở vật chất và giảng dạy, hỗ trợ sinh viên một cách tối đa. Từ đó, sinh viên có cơ hội học tập, phấn đấu, trau dồi kiến thức trên trường lớp lẫn thực hành trong thực tiễn nhằm tạo ra những con người có ích cho đất nước, xã hội trong tương lai.

Em xin chân thành cảm ơn!

Hà Nội, ngày 01 tháng 04 năm 2024

Sinh viên thực hiện

***Nguyễn Quốc Khánh***

# LÝ DO CHỌN ĐỀ TÀI

Trò chơi điện tử, bắt đầu xuất hiện từ khoảng giữa những năm 1950 và 1960. Đã trở nên càng ngày phổ biến trong đời sống hiện đại và phát triển thành một ngành công nghiệp mạnh trong lĩnh vực giải trí của con người. Vậy trò chơi điện tử là gì?

Trò chơi điện tử, hay còn gọi là Video Game, là một loại hình giải trí tương tác trong đó người chơi tương tác với một hệ thống máy tính hoặc các thiết bị điện tử thông minh để điều khiển và tương tác với hình ảnh video hoặc đồ họa động được hiển thị trên màn hình. Trò chơi điện tử có thể chia thành nhiều thể loại, bao gồm đối kháng, phiêu lưu, chiến thuật, thể thao, và nhiều thể loại khác.

Ngoài ra trong trò chơi điện tử, người chơi thường điều khiển các nhân vật hoặc đối tượng trong môi trường ảo và tham gia vào các nhiệm vụ, thách thức, hoặc các tình huống tương tác khác. Trò chơi điện tử có thể được thiết kế cho nhiều nền tảng, bao gồm máy tính cá nhân, Console Game(như PlayStation, Xbox), điện thoại di động, máy tính bảng, và nhiều hệ thống khác.

Ngày nay, trong thời đại công nghệ 4.0. Ngành công nghiệp Game càng ngày càng phát triển mở rộng một cách lớn mạnh và ảnh hưởng một cách sâu sắc đến đời sống giải trí của con người hiện đại. Thế nhưng, cũng trong thời đại này, quyền lực và tài nguyên của ngành công nghiệp càng ngày tập trung vào tay của các nhà đội ngũ phát triển game lớn. Họ thường kiểm soát trong việc những dự án nào được phát triển và phát hành. Điều này dẫn đến việc các ý tưởng mới, phá cách và đôi khi là những trải nghiệm nghệ thuật độc đáo bị lạc lõng hoặc bị từ chối. Sự tập trung vào lợi nhuận và các mô hình kinh doanh truyền thống khiến cho các tựa game được ra mắt khó có sự đổi mới và thường đi theo một lối mòn gây nhàm chán.

Trong tình cảnh đó, đã xuất hiện một thuật ngữ mới mang tên “Indie Game”, hay còn gọi là trò chơi điện tử độc lập, là những sản phẩm được phát triển bởi các đội ngũ phát triển game nhỏ, độc lập và thường là không thuộc sở hữu của các hãng phát triển lớn. Sự xuất hiện của Indie Game là một phản kháng rõ ràng từ cộng đồng phát triển game đối với sự độc quyền và quyền lực của các nhà sản xuất game lớn, đồng thời cũng là phản ứng của cộng đồng người chơi, những người đang tìm kiếm những trải nghiệm mới, không giới hạn bởi các chuẩn mực thị trường.

Indie Game mở ra một không gian cho các nhóm phát triển nhỏ và độc lập, không bị ràng buộc bởi áp lực từ các nhà xuất bản lớn. Các nhà phát triển Indie có tự do để thử nghiệm những ý tưởng mới, thậm chí là những dự án có nguy cơ cao mà các đội lớn có thể sẽ tránh xa. Điều này tạo nên sự đa dạng không chỉ trong cách chơi mà còn trong cách người chơi tương tác với nghệ thuật và câu chuyện. Thế nên các tựa game độc lập thường nhận được sự ủng hộ mạnh mẽ từ người hâm mộ và cộng đồng, giúp chúng phát triển và tiếp tục tồn tại mặc dù không có sự hỗ trợ lớn từ các nhà sản xuất hàng đầu.

Tóm lại, Indie Game là một xu hướng mạnh mẽ và sẽ càng ngày phát triển lớn mạnh trong tương lại. Với sự giúp đỡ của các Game Engine mới phù hợp cho các nhà phát triển độc lập như Unity, Godot,.. Nó ảnh hưởng sâu sắc đến ngành công nghiệp game, không chỉ ở các nước có lịch sử phát triển game lâu năm, mà còn ở các nước đang phát triển, trong đó có Việt Nam.

Nhận thức được sức ảnh hưởng của Indie Game và mong muốn có thể đóng góp một phần sức lực của bản thân trong sự phát triển của Indie Game tại Việt Nam. Em xin được áp dụng những kiến thức đã được học và tìm hiểu để xây dựng đề tài: “**Xây dựng ứng dụng game 2D “Great war of the Moon bằng Unity**”.

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI, MÔI TRƯỜNG VÀ CÔNG CỤ LẬP TRÌNH

## Giới thiệu đề tài

Một trò chơi hay video game là một trải nghiệm điện tử tương tác, đòi hỏi người chơi tương tác với giao diện người dùng hoặc thiết bị đầu vào. Các phản hồi trực quan được tạo ra thông qua các thiết bị như bàn điều khiển, bàn phím, hay thiết bị cảm biến chuyển động. Những phản hồi này thường xuất hiện trên màn hình hiển thị video, chẳng hạn như màn hình máy tính, TV, hoặc thiết bị tai nghe thực tế ảo. Tiếng ồn từ trò chơi thường được truyền tải qua loa hoặc tai nghe, và đôi khi cảm nhận haptic được tích hợp qua công nghệ xúc giác.

Trò chơi video được xác định dựa trên nền tảng của chúng, bao gồm các thể loại như trò chơi arcade, trò chơi trên máy console và trò chơi trên máy tính cá nhân (PC). Gần đây, lĩnh vực này đã mở rộng sang trò chơi di động thông qua điện thoại thông minh và máy tính bảng, cũng như kết hợp với hệ thống thực tế ảo và thực tế tăng cường. Trò chơi Great War of the Moon là một tác phẩm độc đáo trên nền tảng PC. Với lối chơi phong phú và đa dạng, trò chơi hứa hẹn mang đến cho người chơi những trải nghiệm đáng nhớ về cả mặt hình ảnh và âm thanh.

**Great War of the Moon** là một trải nghiệm phiêu lưu đầy kịch tính, đưa người chơi vào vai anh hùng táo bạo chiến đấu với quái vật trên mặt trăng, trò chơi này thực sự tạo nên một trải nghiệm sống động, với đồ họa tuyệt vời và âm thanh sống động. Chúng ta sẽ cùng nhau khám phá và thảo luận về sự thú vị của Great War of the Moon và cách nó mang đến cho chúng ta những phút giây giải trí và thách thức đáng nhớ.

## 1.2: Giới thiệu về Indie Game



Hình 1: Hình ảnh minh họa Indie Game

### 1.2.1: Game Indie là gì ?

Indie Game, hay còn gọi là trò chơi độc lập, là một dòng thể loại game tập trung vào sự độc lập sáng tạo của các đội ngũ phát triển nhỏ, thường là các nhóm độc lập hoặc cá nhân, nằm ngoài tầm kiểm soát của các đại gia trong ngành công nghiệp game. Điều quan trọng không chỉ là sản phẩm cuối cùng mà còn là quá trình phát triển, nơi sự tự do và độ sáng tạo không bị giới hạn.

Indie Game không tuân theo các tiêu chí hay yêu cầu thị trường chung, mà thường xuyên mang lại những trải nghiệm độc đáo, dựa trên cái nhìn cá nhân của người sáng tạo. Các đội ngũ này thường tự chủ trong việc thiết kế, sản xuất và phân phối sản phẩm của mình, tạo ra một không gian mà không bị áp lực bởi thị trường hay quy định từ các công ty lớn.

### 1.2.2: Lịch sử phát triển của game Indie

Indie game không chỉ là một hiện tượng mới, mà là một tiến triển đáng chú ý trong thế giới game. Thập ky 2000 chứng kiến một bước ngoặt quan trọng khi các công cụ phát triển game trở nên dễ dàng tiếp cận hơn và internet mở ra cơ hội cho việc quảng bá không giới hạn. Các nhà phát triển có thể chia sẻ và bán các sản phẩm của họ trực tiếp với cộng đồng trực tuyến, giảm đi sự phụ thuộc vào các đối tác lớn.

Những tựa game như “Braid”(2008) của Jonathan Blow hay “Limbo”(2010) của PlayDead là những điểm mốc quan trọng, khiến người chơi và ngành công nghiệp phải chú ý đến khả năng sáng tạo của indie game. Càng về sau, các tựa game như “Undertale”(2015) của Toby Fox hay “Hollow Knight”(2017) của Team Cherry đã chứng minh rằng indie game không chỉ là những thử nghiệm nhỏ mà còn là những tác phẩm nghệ thuật có tầm ảnh hưởng lớn đối với cả ngành công nghiệp game toàn cầu.

Indie Game tại Việt Nam cũng đã góp phần vào sự sôi nổi của ngành công nghiệp game toàn cầu. Trong những năm gần đây, cộng đồng game độc lập Việt Nam đã trở thành một phần quan trọng trong hành trình phát triển của indie game trên thế giới. Một ví dụ điển hình là “Hoa”(2020) của Skrollcat Studio, một tựa game đẹp mắt với đồ họa tinh tế và âm nhạc giao hưởng độc đáo. Ngoài ra, “Thần Trùng”(2022) của DUT Studio là một ví dụ khác về sự đa dạng trong indie game ở Việt Nam. Trò chơi này sử dụng nguyên tắc của kinh dị tâm lý và đồ họa 3D để tạo ra một trải nghiệm mới mẻ và kịch tính.

### 2.1.3: Ưu điểm

* Sáng tạo và độ tự do: Indie game thường mang đến cho người chơi những trải nghiệm độc đáo do sự sáng tạo không giới hạn và độ tự do của các nhà phát triển.
* Khám phá ý tưởng mới: Các đội ngũ nhỏ thường dám thử nghiệm với các ý tưởng mới và đôi khi mang đến cho người chơi những trò chơi mà các công ty lớn không dám thử nghiệm.
* Phong cách nghệ thuật đa dạng: Indie game thường chú trọng đến nghệ thuật và thiết kế, mang đến cho người chơi những trải nghiệm đồ họa và âm nhạc đa dạng và sáng tạo.
* Mang lại giải trí và giảm căng thẳng: Indie game có thể giúp người chơi giải trí và giảm căng thẳng sau một ngày làm việc mệt mỏi.

### 2.1.4: Nhược điểm

* Ngân sách hạn chế: Do hạn chế về nguồn lực tài chính, các đội ngũ indie thường phải làm việc với ngân sách hạn chế, có thể ảnh hưởng đến chất lượng sản phẩm.
* Quảng bá khó khăn: Thiếu nguồn lực tiếp thị và quảng bá, nhiều tựa game indie có thể gặp khó khăn khi cạnh tranh với các sản phẩm từ các công ty lớn.
* Không ổn định về doanh thu: Do không có sự hỗ trợ lâu dài từ các đối tác lớn, các đội ngũ indie thường phải đối mặt với rủi ro tài chính và không ổn định về doanh thu.

## Giới thiệu phần mềm Unity

Unity là một game engine đa nền tảng được phát triển bởi Unity Technologies, mà chủ yếu để phát triển video game cho máy tính, consoles và điện thoại. Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng. Unity hỗ trợ đồ họa 2D và 3D, các chức năng được viết chủ yếu qua ngôn ngữ C#. Trong 2D games, Unity cho phép nhập sprites và một renderer thế giới Công nghệ đa phương tiện 8 2D tiên tiến. Đối với 3D games, Unity cho phép thiết lập các đập điểm kỹ thuật của các kết cấu và độ phân giải mà công cụ trò chơi hỗ trợ, cung cấp các hỗ trợ cho bump mapping, reflection mapping, parallax mapping, cảnh không gian ambient occlusion (SSAO), hiệu ứng bóng đổ bằng cách sử dụng shadow maps, render thiết lập toàn cảnh đến hiệu ứng.

Unity cung cấp các dịch vụ cho nhà phát triển, bao gồm: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity API, Unity Multiplayer, Unity Performance Reporting and Unity Collaborate.

Unity 3D Engine là một môi trường phát triển tích hợp, mạnh mẽ, hỗ trợ thao tác kéo thả, tuỳ biến giao diện nhanh chóng, trực quan.Cung cấp các công cụ xử lý đồ hoạ, tích hợp sẵn thư viện vật lý, tính toán va chạm...

Unity 3D Engine hỗ trợ phát triển cả game 2D và 3D, hỗ trợ nhiều nền tảng thông dụng như OSX, Linux, Window, Web, iOS, Window Phone 8, Android, PS3… với cộng đồng người dùng, hỗ trợ rộng lớn.



Hình 2: Các hệ điều hành Unity nhắm tới.

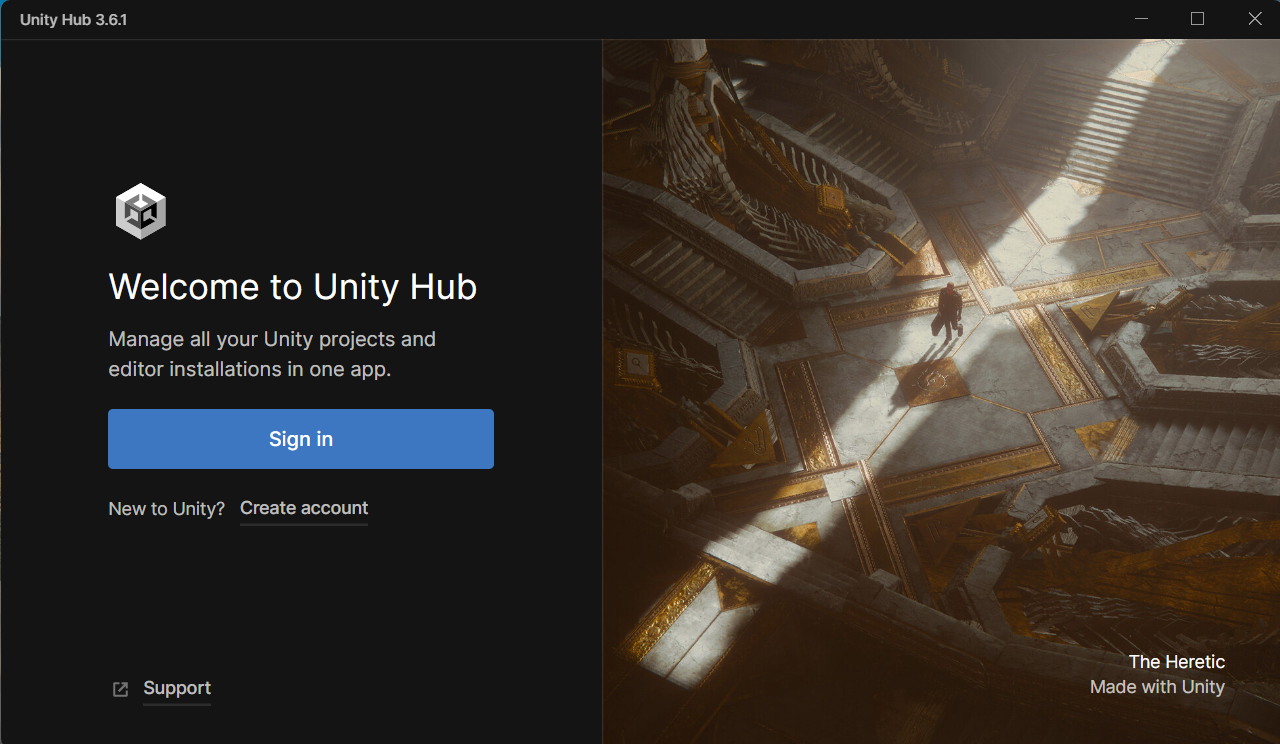
Unity 3D Engine có phiên bản miễn phí và trả phí, hỗ trợ chạy trên hệ điều hành Window và MacOS. Unity Engine hướng tới người sử dụng chuyên nghiệp và cả nghiệp dư, nên khá dễ để sử dụng. Với ngôn ngữ lập trình bằng C, ngày nay rất nhiều nhà phát triển game lựa chọn Unity 3D Engine để phát triển bởi khả năng hỗ trợ đa nền tảng và sự mạnh mẽ tiện dụng của Unity 3D Engine. Đến với Unity, các bạn sẽ không cần phải băn khoăn về các vấn đề xử lý, các khái niệm đồ hoạ phức tạp... tất cả đều trở nên dễ dàng và nhanh chóng với Unity..

### Ưu nhược điểm của phần mềm

* **Ưu điểm**
* Đa nền tảng: Unity hỗ trợ phát triển ứng dụng trên nhiều nền tảng, bao gồm máy tính, điện thoại di động, máy tính bảng, console chơi game và thực tế ảo/ thực tế tăng cường.
* Hỗ trợ đa ngôn ngữ và đa công nghệ: Unity hỗ trợ nhiều ngôn ngữ lập trình như C#, JavaScript và Boo. Ngoài ra, Unity cũng hỗ trợ các công nghệ và framework phổ biến như .NET, Mono, và Xamarin.
* Cộng đồng và tài liệu phong phú: Unity có một cộng đồng lớn và nhiệt tình, cung cấp nhiều tài liệu, hướng dẫn và nguồn tài nguyên. Bạn có thể tìm thấy các diễn đàn, trang web, video hướng dẫn, và các tài liệu chi tiết để giúp bạn trong quá trình phát triển.
* Công cụ thiết kế đồ họa và trực quan: Unity cung cấp một giao diện trực quan và các công cụ thiết kế đồ họa giúp bạn tạo và chỉnh sửa các tài nguyên như hình ảnh, âm thanh và đồ họa vector
* Hỗ trợ phát triển nhanh chóng: Unity cung cấp các khung thức và thư viện mạnh mẽ giúp bạn phát triển nhanh chóng các tính năng như vật lý, xử lý sự kiện, đa luồng, quản lý tài nguyên và trực quan hóa. Bạn cũng có thể tận dụng các giao diện người dùng và các tính năng giúp tăng tốc quá trình phát triển.
* Hỗ trợ kỹ thuật cao: Unity cung cấp hỗ trợ kỹ thuật chuyên nghiệp thông qua các kênh hỗ trợ trực tuyến, bao gồm tư vấn kỹ thuật, bản vá lỗi và cập nhật thường xuyên. Điều này giúp bạn giải quyết các vấn đề phát triển và vận hành ứng dụng một cách hiệu quả.
* **Nhược điểm**
* Unity không phù hợp với các dự án lớn: Không thể đào sâu quá vừa là điểm mạnh vừa là điểm yếu của Unity. Một mặt, nó cho phép quy trình nhanh chóng, thích ứng tốt với người mới bắt đầu, mặt khác, điều đó có nghĩa là Unity có thể không phải là thứ bạn đang tìm kiếm nếu bạn đang hy vọng tạo ra bất cứ thứ gì thật đặc biệt hoặc trên quy mô lớn.
* Unity thúc đẩy các phương pháp mã xấu: Vấn đề nằm ở chỗ Unity không nhất thiết phải trở thành một công cụ trò chơi.Ban đầu nó được dùng để phát triển web và JavaScript

### Hướng dẫn tải và cài đặt (môi trường windows)

Vào trang Download chọn Download Unity Hub, sau khi tải xuống hoàn tất double click vào file đã tải. Việc cài đặt Unity Hub diễn ra bình thường và tương tự như cài đặt các chương trình khác. Sau khi cài đặt, khởi động Unity Hub rồi đăng nhập vào tài khoản Unity, tài khoản Unity có thể được tạo miễn phí tại trang chủ.



Hình 3: Đăng nhập Unity Hub

Sau khi đăng nhập trong trường hợp chưa có License thì phải tạo 1 License, vào Preferences bên cạnh profile góc trên phải và chọn License Management và chọn như hình để kích hoạt 1 License miễn phí.

*A screenshot of a computer

Description automatically generated*

Hình 4: Kích hoạt Unity Hub

Quay ra màn hình chính, chọn Installs->ADD để cài đặt Unity.

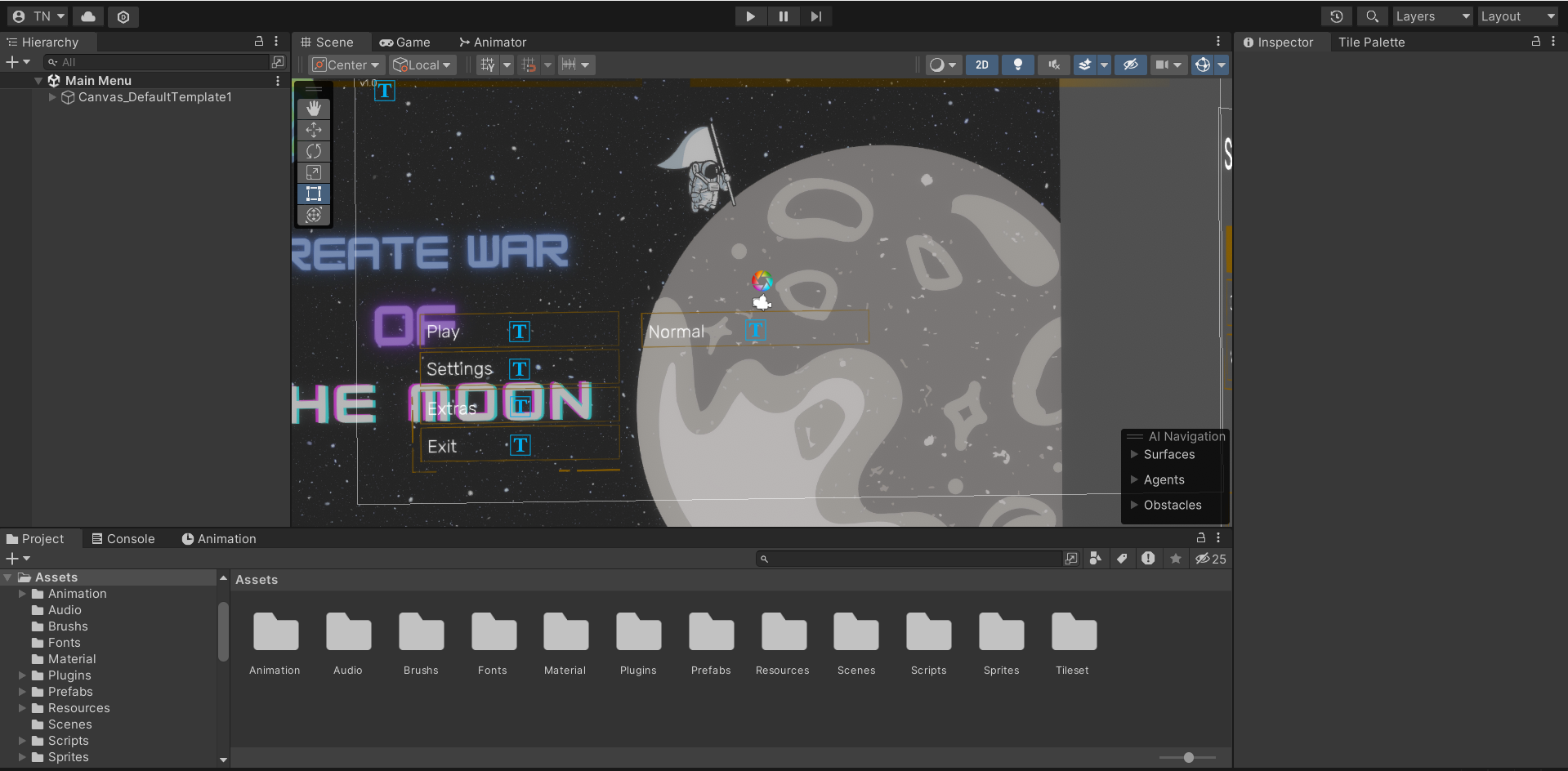
A screenshot of a computer

Description automatically generated

Hình 5: Chọn phiên bản để cài đặt công cụ Unity

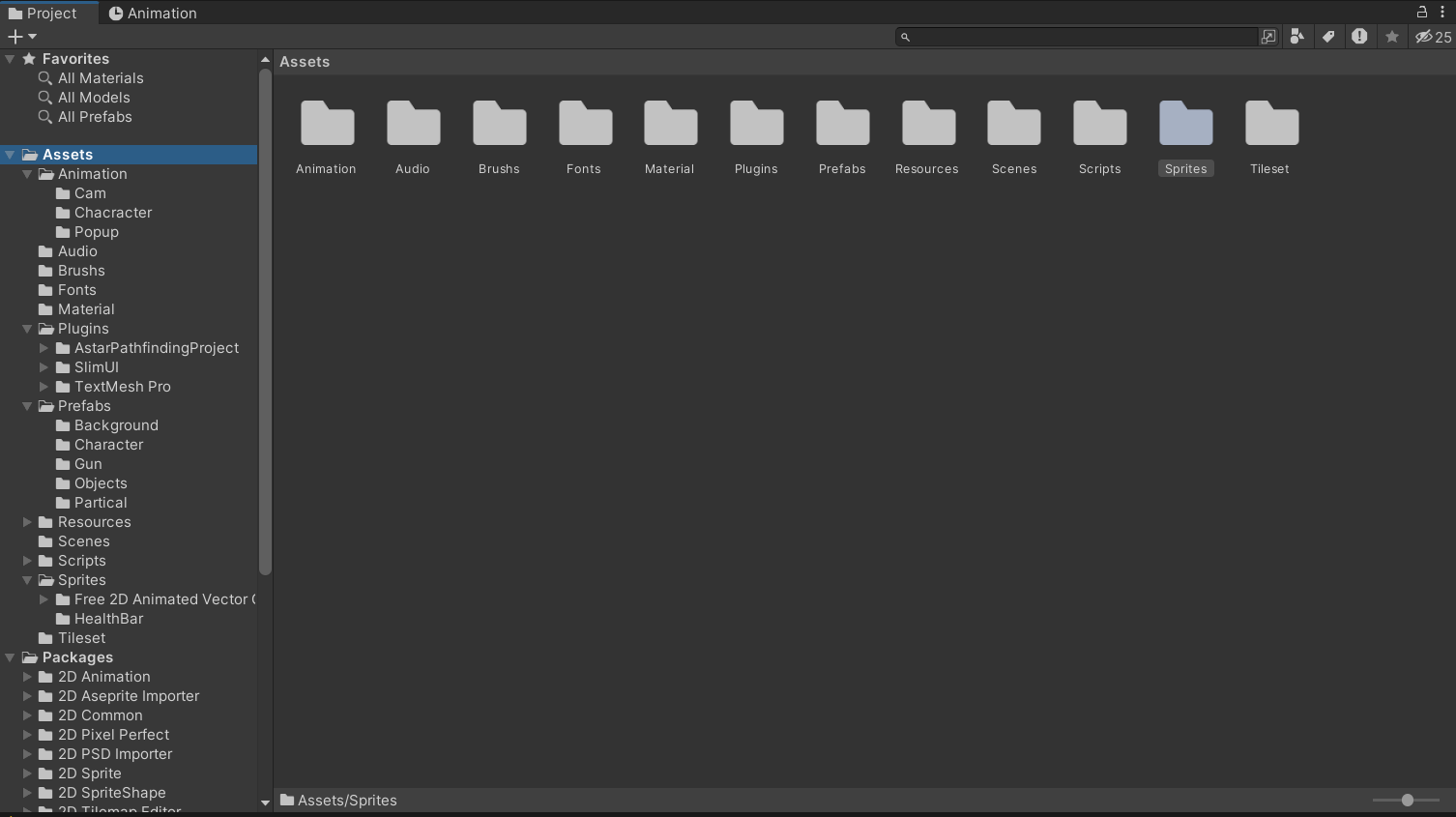
### Giao diện phần mềm

#### Giao diện chính



Hình 6: Giao diện chính của phần mềm Unity

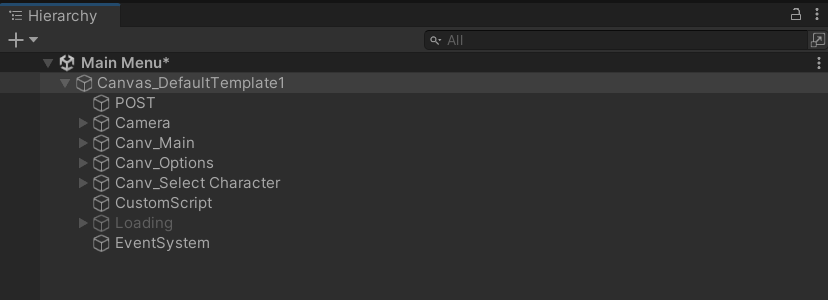
#### Giao diện project



Hình 7: Giao diện project

* Là nơi hiển thị tài sản và tổ chức của dự án: Tệp, script, kết cấu, mô hình,…
* Asset là các phần tử tồn tại dưới dạng tệp trong thư mục Assets: Kết cấu, mesh, tệp âm thanh, script,…
* Game Object là đối tượng một phần của cảnh (level).
* Có thể tạo Asset từ Game Object và có thể tạo Game Object từ Asset.
* Di chuyển Asset
  + Unity duy trì liên kết giữa các tài sản khác nhau liên quan đến các dự án.
  + Di chuyển hoặc xóa các phần tử bên ngoài Unity có thể gây ra sự cố, nên thực hiện việc quản lý tài sản bên trong Unity.
  + Khi nhấn vào một thư mục trong Project view, nội dung của thư mục được hiển thị trong phần Assets ở bên phải.
* Tổ chức Project
  + Các loại asset như cảnh, script, kết cấu,… nên có thư mục riêng.
  + Các nút Favorites cho phép chọn nhanh tất cả các asset thuộc một loại nhất định.
  + Tìm kiếm với thanh tìm kiếm sẽ thu hẹp kết quả giữa Assets và Asset Store.
  + Asset Store duyệt qua các tài sản phù hợp với tiêu chí tìm kiếm từ Unity Asset Store.
  + Có thể thu hẹp thêm kết quả theo nội dung miễn phí và trả phí

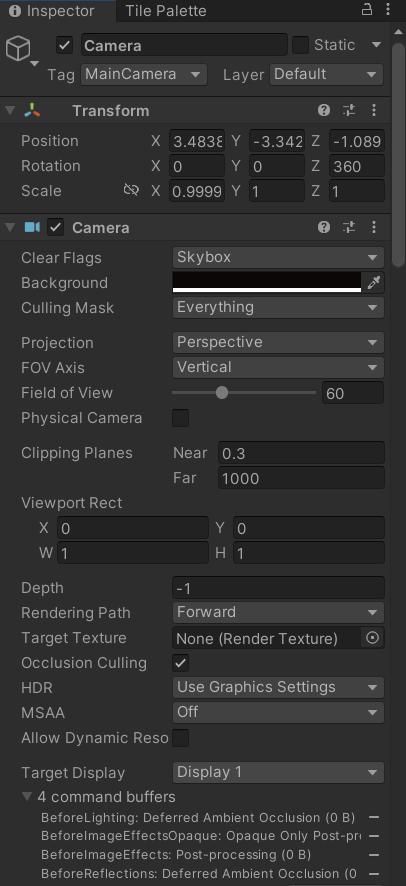
#### Hierarchy view



Hình 8: Giao diện màn hình Hierarchy

* Là nơi hiển thị tất cả các phần tử trong cảnh hiện tại thay vì toàn bộ dự án.
* Tạo dự án lần đầu tiên sẽ nhận được cảnh mặc định chỉ có hai phần tử là Main Camera và Directional Light.
* Khi thêm các phần tử vào cảnh , chúng sẽ xuất hiện trong Hierarchy View.

#### Inspector view



Hình 9: Giao diện màn hình Inspector.

* Cho phép xem tất cả các thuộc tính của một phần tử hiện đang được chọn.
* Nhấn vào đối tượng trong Project view hoặc Hierarchy view, Inspector sẽ hiển thị thông tin của đối tượng đó.
* Bỏ chọn hộp kiểm bên cạnh tên của đối tượng, nó sẽ bị vô hiệu hóa và không xuất hiện trong dự án.

#### Scene view

A screenshot of a video game

Description automatically generated

Hình 10: Giao diện màn hình Scene.

* Cho phép xem trò chơi một cách trực quan khi nó đang xây dựng.
* Sử dụng chuột, phím để di chuyển trong cảnh và thiết lập vị trí cho các đối tượng.
* Các điều khiển trong Scene view
  + Draw mode
    - Kiểm soát cách cảnh được vẽ.
    - Mặc định là Shaded, các đối tượng sẽ được vẽ với kết cấu màu sắc đầy đủ.
  + 2D/3D view
    - Thay đổi từ chế độ xem 3D sang chế độ xem 2D.
    - Ở chế độ xem 2D, scene gizmo không được hiển thị.
  + Gizmo selector
    - Cho phép chọn gizmos, nghĩa là các chỉ báo giúp đỡ lỗi trực quan hoặc hỗ trợ thiết lập xuất hiện trong scene view.
    - Xác định xem lưới vị trí có hiển thị hay không.
    - Điều khiển này hiển thị hướng hiện đang đối diện và căn chỉnh scene view với một trục.
    - Có các chỉ báo X, Y và Z phù hợp với ba trục giúp dễ dàng nhận biết chính xác đang nhìn theo hướng nào trong cảnh
    - Nhấn vào một trong các trục của gizmo, scene view ngay lập tức bám vào trục đó và theo hướng đó.
    - Nhấn vào hộp ở giữa gizmo để chuyển đổi giữa chế độ Iso (Isometric) và Persp (Perspective

#### Game view

* + - * + Cho phép “chơi” trò chơi bên trong trình editor bằng cách cung cấp mô phỏng đầy đủ về cảnh.
        + Giao diện

A screenshot of a video game

Description automatically generated

Hình 11: Giao diện màn hình Gameview.

* Nếu game view bị ẩn sau scene view hoặc không thấy tab của nó thì nhấn Play thì tab game view sẽ xuất hiện.
* Các nút điều khiển
* Play
  + Cho phép phát cảnh hiện tại.
  + Tất cả các điều khiển, hoạt ảnh, âm thanh và hiệu ứng đều hiện diện và hoạt động.
  + Để dừng trò chơi đang chạy, nhấn lại vào nút Play.
* Pause
  + Tạm dừng việc thực hiện gameview hiện đang chạy.
  + Trò chơi duy trì trạng thái và tiếp tục chính xác vị trí của nó sau khi tạm dừng.
  + Nhấn vào nút Pause một lần nữa để trò chơi tiếp tục chạy.
* Step
  + Hoạt động trong khi Game view bị tạm dừng và khiến trò chơi thực thi một khung hình duy nhất của trò chơi.
  + Cho phép “bước” qua trò chơi và gỡ lỗi gặp phải.
  + Nhấn vào nút Step trong khi trò chơi đang chạy khiến trò chơi tạm dừng.
* Aspect drop-down
  + Chọn tỷ lệ khung hình cho cửa sổ game view trong khi chạy.
  + Mặc định là Free Aspect, có thể thay đổi để phù hợp với tỷ lệ khung hình của nền tảng đang phát triển.
* Maximize on play
  + Xác định xem Game view có chiếm toàn bộ trình editor khi chạy hay không.
  + Mặc định, tính năng này bị tắt và trò chơi đang chạy chỉ có kích thước của tab Game view.
* Mute Audio
  + Tắt âm thanh khi chơi trò chơi.
  + Điều này rất hữu ích khi cảm thấy khó chịu vì phải nghe bài test game lặp đi lặp lại.
* Stats
  + Xác định số liệu thống kê kết xuất có được hiển thị trên màn hình trong khi trò chơi đang chạy hay không.
  + Những số liệu thống kê này hữu ích để đo lường hiệu suất của cảnh.
  + Mặc định, các số liệu thống kê được tắt.
* Gizmos
  + Đây vừa là nút vừa làm menu thả xuống.
  + Xác định xem gizmos có được hiển thị trong khi trò chơi đang chạy hay không.
  + Mặc định, gizmo Game view không được hiển thị.
  + Menu thả xuống trên nút này xác định gizmo nào xuất hiện nếu gizmos được bật.Kéo camera xung quanh cảnh

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ GAME GREAT WAR OF THE MOON

## Giới thiệu tổng quan

### Thông tin game

- Mô tả chung về game : Great War of the Moon là một tựa game mang phong cách “Endless run” độc đáo , bạn sẽ nhập vai vào một chiến binh loài người và chiến đấu với hàng ngàn quái vật ngoài hành tinh trên mặt trăng để bảo vệ trái đất. Mục tiêu bạn cần hoàn thành là tiêu diệt thật nhiều quái vật để đạt được điểm số cao nhất và thu thập vàng để sở hữu được các nhân vật trong game.

### Luật chơi

Trò chơi sẽ có một bản đồ rất rộng lấy bối cảnh từ mặt trăng, người chơi sẽ di chuyển xung quanh bản đồ và tiêu diệt quái vật.

Mỗi khi tiêu diệt được một lượng quái vật nhất định thì nhân vật sẽ tăng level của bản thân và nhận được 1 trang bị vũ khí mới.

Quái vật sẽ liên tục được sinh ra theo thời gian.Quái vật sẽ tăng giới hạn sinh ra theo thời gian.

Màn chơi sẽ giới hạn thời gian là 30 phút , người chơi cần khéo léo di chuyển, thu thập tiền vàng và chọn vũ khí phù hợp để tiêu diệt được nhiều quái vật và đạt điểm số cao nhất có thể.

### Đối tượng chơi

Với lối chơi game đơn giản bằng các thao tác di chuyển và luật chơi dễ hiểu game hiện tại đang phát triển tới đối tượng :

* + Người chơi từ 8 tuổi trở lên muốn rèn luyện tăng cường sự nhanh nhẹn quan sát.
  + Người lớn muốn xả stress.

### Nền tảng

Game được xây dựng bằng Unity nên có khả năng phát triển tốt trên các nền tảng lớn. Nhưng hiện tại game tập trung chủ yếu vào nền tảng Window (Laptop và PC) và Android (Mobile).

## 1.4.5: Kịch bản game

#### 1.4.5.1: Mô tả

Great War of the Moon là một tựa game hành động phiêu lưu đặt người chơi vào vai một người hiện đại bị dịch chuyển vào một hầm ngục tối, phải khám phá các hang động nguy hiểm, đánh bại quái vật, gặp gỡ các nhân vật trong hầm ngục, từ đó tìm cách thoát khỏi hầm ngục tối tăm và nguy hiểm.

#### 1.4.5.2: Điều kiện thắng và thua

**Điều kiện thắng:**

Người chơi sống sót được càng lâu càng tốt và đạt được điểm số cao nhất có thể.

**Điều kiện thua:**

Người chơi bị quái vật tấn công hết HP => Game Over.

#### 1.4.5.3: Thiết kế game (Game Design)

##### **1.4.5.3.1: Nhân vật chính (Player)**

Nhân vật chính là một nhà thám hiểm mặt trăng có trang bị vũ khí.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | HP | EXP | SPEED | Hình Ảnh |
| Jack | 10 | 0 | 5 |  |

Bảng 1.4.3.1: Mô tả nhân vật chính

##### **1.4.5.3.2: Quái vật (Enemy).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | HP | ATK | SPEED | Hình Ảnh |
| Venom | 40 | 3 | 3 |  |
| Seaker | 60 | 3 | 4 |  |
| Bat | 30 | 3 | 5 |  |
| Minotaur | 100 | 5 | 2 |  |
| Final Boss | 1000 | 8 | 4 |  |

Bảng 1.4.3.1: Mô tả các quái vật

##### **1.4.5.3.3: Vũ Khí (Weapons)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | Sát Thương | Tốc độ bắn | Số lượng đạn | Hình Ảnh |
| AK-47 | 10 | 30 | 1 |  |
| SG-31 | 10 | 10 | 3 |  |
| M4-A1 | 15 | 20 | 1 |  |

Bảng 1.4.3.1: Mô tả các vũ khí

##### **1.4.5.3.4: Thiết kế level và Tính điểm**

Trong game người chơi sẽ có thanh kinh nghiệm để lên cấp và nhận vũ khí mới. Để lên cấp độ người chơi cần tiêu diệt quái vật và nhận về exp. Lượng exp cần thiết để lên cấp được tính bằng công thức sau:

**Exp\_char = monsterCount\*rand(1-4) exp**

|  |
| --- |
| **Current EXP = BaseExp x 2** |

Trong đó Current EXP là lượng kinh nghiệm cần thiết để lên cấp.

BaseExp là lượng kinh nghiệm cần thiết để lên cấp ở cấp trước.

Ở level 1 baseEXP = 30. Vì vậy :

**Exp\_char = currentEXP =>Level = Level +1**

Great war of the moon cũng cung cấp điểm số cho người chơi để tăng cảm giác đạt được thành tựu qua mỗi lần chơi , người chơi sẽ được thỏa mãn khi có điểm số cao hơn lần chơi trước. Công thức tính điểm như sau:

**TotalPoint = EnemyCount\*100**

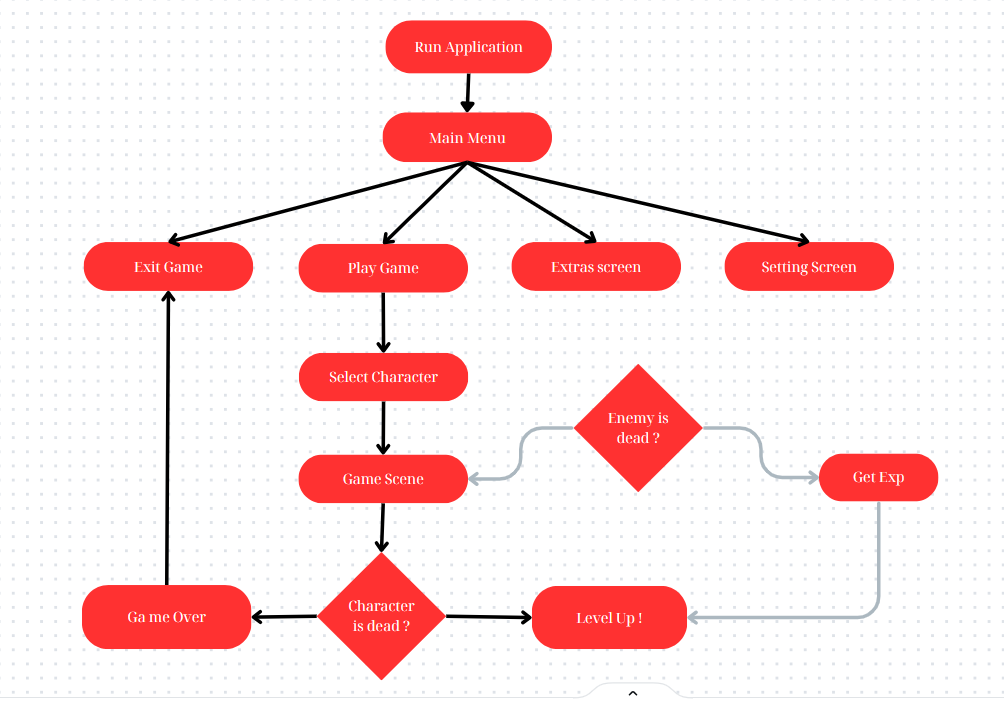
### 1.4.5: Tương tác và điều khiển game

- Tất cả các thao tác đều được điều khiển bởi bàn phím và chuột (PC),người chơi dùng bàn phím để thao tác di chuyển và dùng chuột để lựa chọn các vũ khí.

- Đối với Mobile thì người chơi di chuyển bằng joystick và chọn vũ khí bằng cách chạm vào lable trên màn hình

## Storyboard

### Sơ đồ các màn hình



Hình 12: Storyboard của game

### Màn hình bắt đầu

* Nút START bắt đầu vào màn hình chơi game.

A screenshot of a video game

Description automatically generated

Hình 13: Màn hình bắt đầu

### Màn hình chơi game

Màn hình trong chơi game:

* Các UI trong game :

**

Hình 14: Thanh HP của nhân vật

**

Hình 15: Thanh exp của nhân vật

**

Hình 16: Số quái đã tiêu diệt

*A yellow coin with white text

Description automatically generated*

Hình 17: Số vàng đang sở hữu

* Màn hình khi chơi game

A screenshot of a video game

Description automatically generated

Hình 18: Màn hình chơi game

### Màn hình game over

**Màn hình kết quả (LOST).**

Nếu bạn bị hết máu:

* Quay về nơi khi bắt đầu game.

*A screenshot of a video game

Description automatically generated*

Hình 19: Màn hình thua

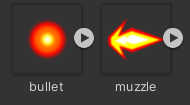
## Tài nguyên

### Hình ảnh

* Các button và vật phẩm:
* https://rgsdev.itch.io/free-cc0-modular-animated-vector-characters-2d
* https://assetstore.unity.com/packages/2d/characters/furball-2d-v1-2-mobile-optimized-40588
* https://opengameart.org/content/zelda-like-tilesets-and-sprites





**

Hình 20: Hình ảnh các asset và vật phẩm

### Màu sắc

Các hình ảnh có màu sắc sặc sỡ trên nền background màu tối giúp làm nổi bật vật thể và kích thích người chơi.

A screenshot of a video game

Description automatically generatedĐa dạng màu sắc nhưng loại bỏ các cặp màu gây chói như: đỏ - xanh, trắng – đỏ, … gây khó chịu cho mắt.

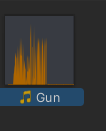
Hình 21: Màu sắc trên nền tối thân thiện với mắt

### Âm thanh

Sử dụng các âm thanh nhạc nền , click , âm thanh kèm theo màn thắng , màn thua giúp game sinh động hơn.

Âm thanh phát mỗi combo được tăng cao độ dựa trên số combo giúp nhịp chơi nhanh hơn, người chơi cảm thấy kịch tính hơn.

Âm thanh lấy từ assets store của unity và https://www.patreon.com/

****

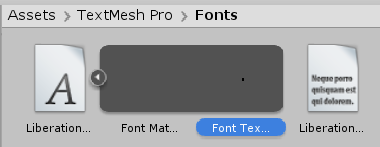
Hình 22: Âm thanh của trò chơi

### Font chữ

**Font** : LiberationSans

**Kích thước font** : 30pt.

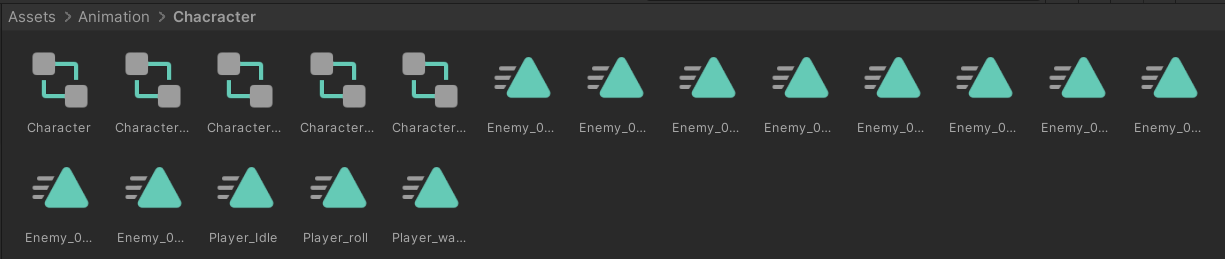
**Màu chữ** : trắng.



Hình 23: Hình ảnh font chữ

### Hiệu ứng và animation

* Ở phần thiết kế bẫy thêm animation khi bẫy xuất hiện để tăng độ sinh động.
* Sử dụng animation bị tiêu diệt và tấn công ở người chơi, quái.
* Một số animation:



Hình 24: Các animation có trong game

# CHƯƠNG 3. CÀI ĐẶT GAME VÀ KẾT QUẢ SẢN PHẨM

## Các kỹ thuật thực hiện

### Tạo các Prefab cho game (Game UI)

* Sử dụng các GameObject đã tạo kéo vào asset của Unity để tạo prefab cho GameObject đó.

A screenshot of a computer

Description automatically generated

A screenshot of a video game

Description automatically generatedA screenshot of a video game

Description automatically generated

Hình 25: Giao diện các prefab của game

### Sprites và Animation:

Sprites là các đối tượng 2D đơn giản có hình ảnh đồ họa (được gọi là kết cấu) trên chúng. Unity sử dụng các họa tiết theo mặc định khi động cơ ở chế độ 2D. Khi được xem trong không gian 3D, các họa tiết sẽ có vẻ mỏng như tờ giấy, vì chúng không có chiều rộng Z.

Sprites luôn đối mặt với máy ảnh ở góc vuông góc trừ khi xoay trong không gian 3D.

Bất cứ khi nào Unity tạo ra một sprite mới, nó sử dụng một kết cấu. Kết cấu này sau đó được áp dụng trên GameObject mới và thành phần Sprite Renderer được đính kèm. Điều này làm cho trò chơi của chúng tôi hiển thị với kết cấu của chúng tôi, cũng như cung cấp cho nó các thuộc tính liên quan đến giao diện trên màn hình.

Trong Unity, animation được sử dụng để tạo ra các hiệu ứng chuyển động cho các đối tượng trong trò chơi. Animation có thể bao gồm việc di chuyển, xoay, thay đổi kích thước và thậm chí áp dụng các hiệu ứng hình ảnh đặc biệt để làm tăng tính tương tác và hấp dẫn.

### Collision Detection

Collision Detection (Phát hiện va chạm) là một khía cạnh quan trọng trong phát triển game, đặc biệt là trong game 2D và 3D. Nó cho phép bạn xác định khi nào các đối tượng trong game va chạm với nhau để thực hiện các hành động phản ứng như phản đòn, đánh thắng, hay xử lý các tương tác khác

Discrete Collision Detection: Trong phát hiện va chạm rời rạc, va chạm được kiểm tra ở một số điểm cố định trong một khung hình cố định. Nó thường nhanh và phù hợp cho hầu hết các trò chơi 2D.

Continuous Collision Detection: Trong phát hiện va chạm liên tục, va chạm được kiểm tra liên tục trong suốt quá trình di chuyển của đối tượng. Điều này đảm bảo rằng không có "xuyên qua" (tunnelling) trong trường hợp di chuyển nhanh. Tuy nhiên, nó có thể yêu cầu tính toán tốn kém hơn.

Rigidbody2D : là các thành phần cho phép xử lý vật lý, bao gồm cả va chạm. Khi bạn sử dụng Rigidbody, Unity sẽ tự động xử lý va chạm và tương tác vật lý với các collider.

BoxCollider2D là một thành phần trong Unity được sử dụng để xác định hình dạng và vùng va chạm của một GameObject trong không gian 2D. Nó là một loại collider phổ biến trong phát triển game 2D và thường được sử dụng cho các đối tượng có hình dạng hộp chữ nhật. Bạn có thể xử lý va chạm bằng cách sử dụng các hàm trình quản lý va chạm và callbacks trong code của bạn. Điều này cho phép bạn thực hiện các hành động cụ thể khi va chạm xảy ra.

### Tilemap và Tilemap collider

Tilemap trong Unity là một công cụ cho phép bạn tạo và quản lý các môi trường game dưới dạng một lưới ô gạch (grid of tiles). Tilemap cho phép bạn sử dụng các ô gạch (tiles) để xây dựng các phần của môi trường game như sàn nhà, tường, cỏ, cây cối, và nhiều đối tượng khác trong game 2D.

Tilemap Collider là một thành phần quan trọng trong Unity cho phép bạn xác định va chạm và xử lý sự va chạm giữa nhân vật và các ô gạch (tiles) trên Tilemap.

### UI và HUD

UI (User Interface) và HUD (Heads-Up Display) là hai phần quan trọng trong game để tạo ra giao diện người dùng và hiển thị thông tin quan trọng cho người chơi. Dưới đây là chi tiết về cả hai:

UI elements là các thành phần hoặc đối tượng dùng để hiển thị thông tin và tương tác với người chơi. Các UI elements thường bao gồm nút, hình ảnh, văn bản, thanh trượt, hộp thoại, và nhiều thành phần khác.

Canvas là một thành phần chứa tất cả các UI elements. Unity hỗ trợ ba loại Canvas: Screen Space Overlay, Screen Space Camera, và World Space. Loại phổ biến nhất là Screen Space Overlay, nơi Canvas nằm trên màn hình và không bị ảnh hưởng bởi không gian trong game.

Event System là một thành phần quan trọng để xử lý sự kiện tương tác của người chơi với UI elements. Nó quản lý các sự kiện như nhấn nút, di chuyển con trỏ, và đối thoại. Bạn có thể kiểm soát và tương tác với UI elements bằng cách sử dụng scripts. Điều này cho phép bạn thay đổi nội dung và hành vi của UI theo logic của game.

HUD elements là các thành phần của giao diện người dùng được hiển thị trực tiếp trên màn hình chơi game để cung cấp thông tin quan trọng cho người chơi. Chúng bao gồm thông tin như thanh máu, điểm số, vật phẩm hiện tại, bản đồ, thời gian còn lại, và nhiều thông tin khác.

HUD elements thường cố định ở một vị trí trên màn hình và không di chuyển theo camera hoặc nhân vật trong game. Điều này giúp người chơi có thể theo dõi thông tin quan trọng mà không cần chú ý đến việc di chuyển camera.

HUD elements thường được thiết kế để truyền tải thông tin một cách trực quan và dễ hiểu. Ví dụ, thanh máu có thể được hiển thị bằng hoạt ảnh trái tim, và số điểm có thể hiển thị bằng văn bản số.

A screenshot of a video game

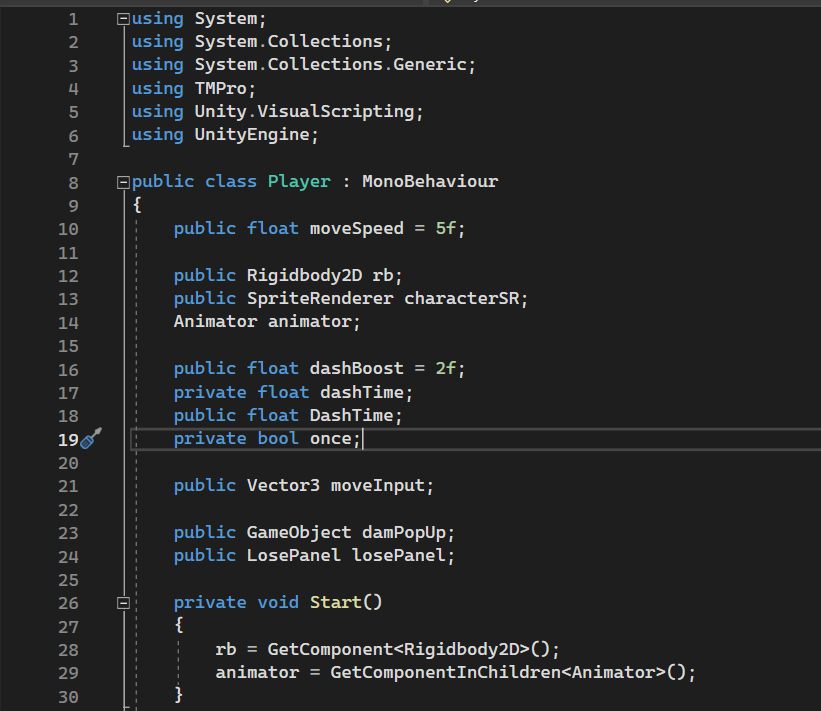
Description automatically generated

Hình 26: UI và HUD trong game

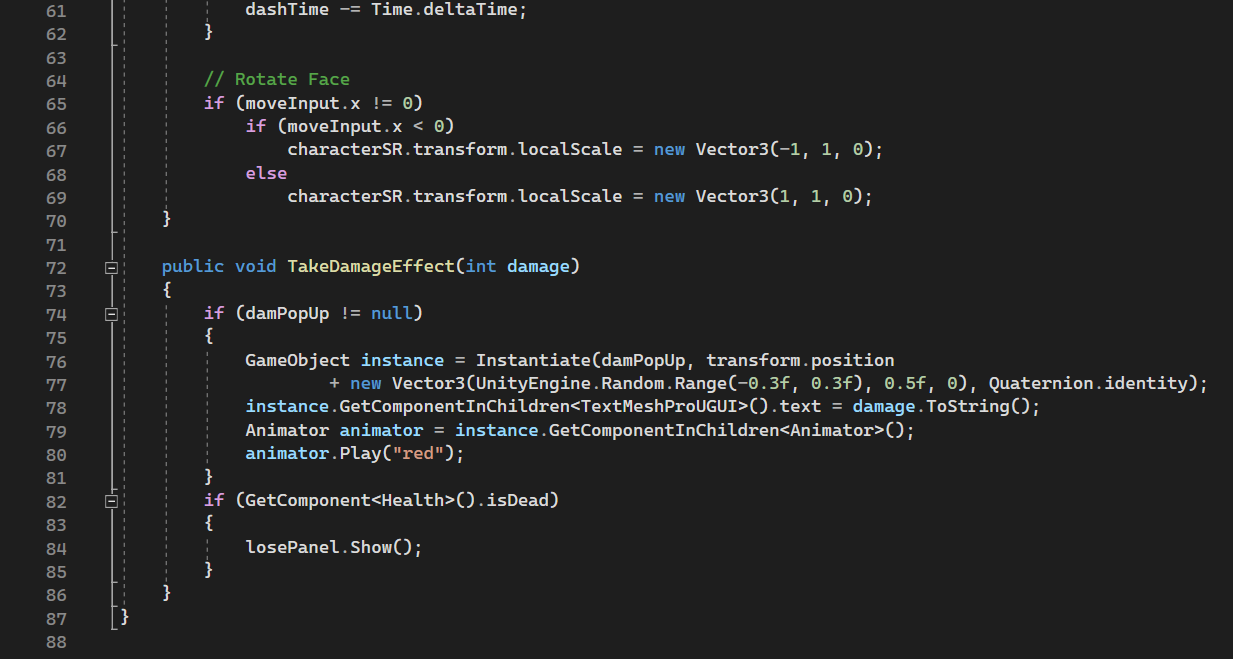
### Code chức năng chính của game

Các đoạn code cho các chức năng chính của game:

* Phần code thiết kế nhân vật:

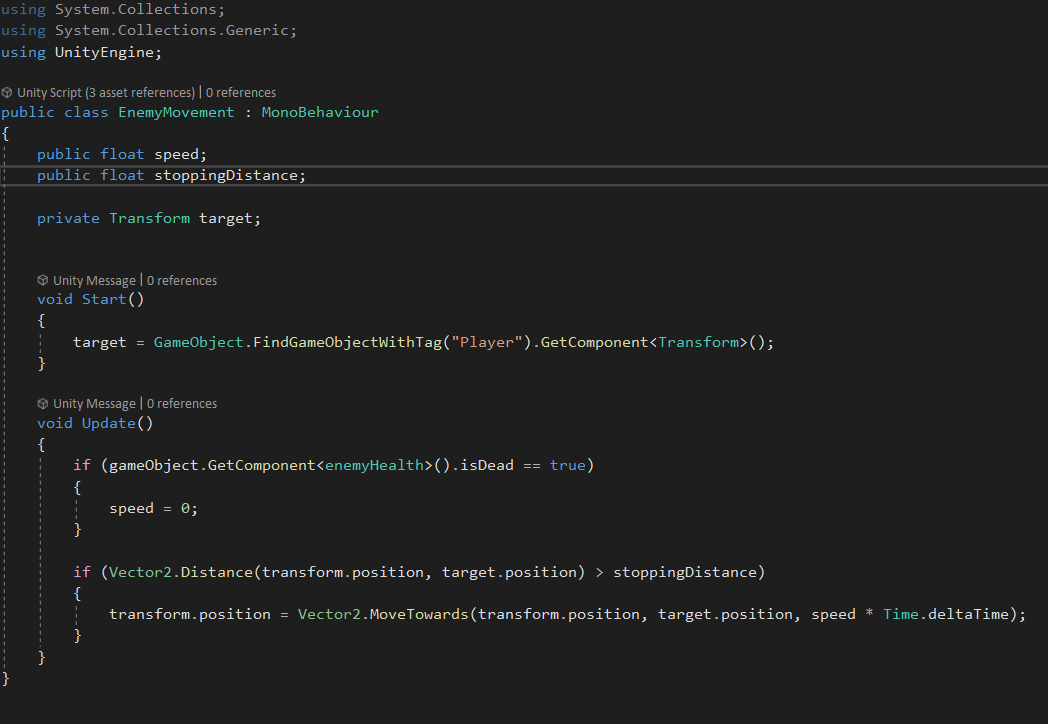
**

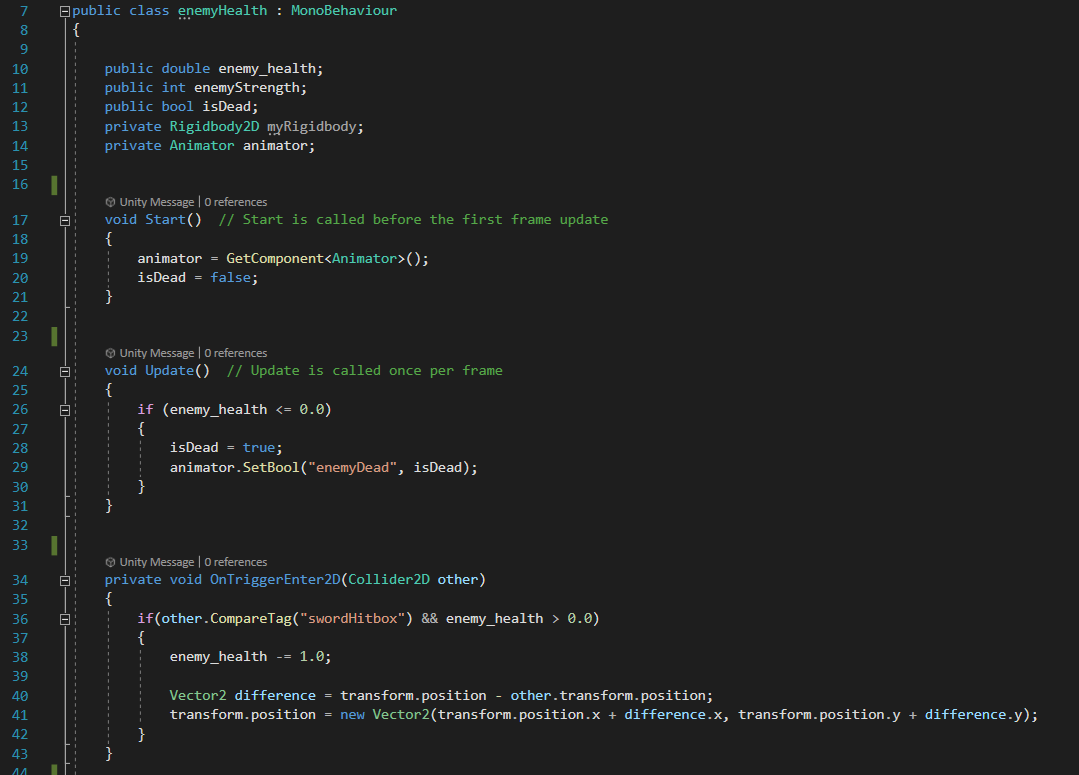
**

**

Hình 27: Phần code thiết kế nhân vật

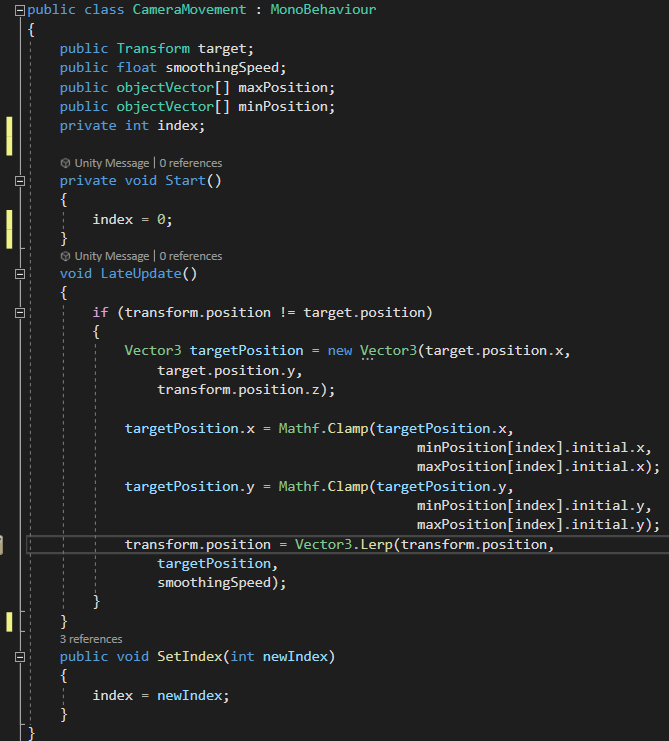
* Phần code thiết kế quái:





Hình 28: Phần code tấn công quái

* Phần code camera di chuyển theo nhân vật:



Hình 29: Phần code camera di chuyển theo nhân vật

## Sản phẩm màn hình

### Màn hình chính

A cartoon characters on the moon

Description automatically generated

Hình 30: Hình ảnh màn hình chính.

### Màn hình chọn nhân vật chơi game

A screenshot of a video game

Description automatically generated

Hình 31: Hình ảnh chọn nhân vật

### Màn hình chơi game

A screenshot of a video game

Description automatically generated

Hình 32: Hình ảnh màn hình chơi game chính 1

A screenshot of a video game

Description automatically generated

Hình 33: Màn hình chọn vũ khí trong game

### Màn hình game over

A screenshot of a video game

Description automatically generated

Hình 34: Hình ảnh màn hình game over

### 3.2.5: Màn hình shop skin

A screenshot of a video game

Description automatically generated

Hình 35: Hình ảnh màn hình shop skin

### 3.3: Kiểm thử game

#### 3.3.1: Kế hoạch kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mốc công việc** | **Sản phẩm** | **Thời gian** | **Bắt đầu** | **Kết thúc** |
| Lập kế hoạch kiểm thử | Test plan | 2 ngày | 15/05/2024 | 16/05/2024 |
| Xem lại các tài liệu | Test plan | 2 ngày | 16/05/2024 | 17/05/2024 |
| Thiết kế các testcase | Test case | 1 ngày | 17/05/2024 | 18/05/2024 |
| Viết các testcase | Test case | 2 ngày | 18/05/2024 | 19/05/2024 |
| Xem lại các testcase | Test case | 1 ngày | 19/05/2024 | 20/05/2024 |
| Thực thi các testcase | Test case | 1 ngày | 20/05/2024 | 21/05/2024 |
| Ghi nhận và đánh giá kết quả kiểm thử | Test report | 2 ngày | 21/05/2024 | 22/05/2024 |

Bảng 3.3.1: Kế Hoạch Kiểm Thử

#### 3.3.2: Các yêu cầu về tài nguyên

**3.3.2.1: Phần cứng**

Máy tính cá nhân có kết nối mạng Internet.

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **SSD** | **Architecture** |
| Intel Core i7, 2.3 GHz | 8 GB | 250 GB | 64 bit |

Bảng 3.3.2.1: Yêu cầu phần cứng

***3.3.2.2: Phần mềm***

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Unity Engine | 2021.2.6f1 | Công cụ phát triển Game |
| Microsoft Windows 10 Professional | 10 | Hệ điều hành |

Bảng 3.3.2.2: Yêu cầu phần mềm

**3.3.2.3: Công cụ kiểm thử**

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| Quản lý Test Case | Microsoft Office Excel | Microsoft | 2019 |
| Quản lý phiên bản | Microsoft Office Word | Microsoft | 2019 |
| Theo dõi lỗi | Microsoft Office Excel | Microsoft | 2019 |

Bảng 3.3.2.3: Công cụ kiểm thử

**3.3.2.4: Môi trường kiểm thử**

* Máy tính cá nhân có kết nối mạng.
* Phần mềm Unity Engine để kiểm tra ứng dụng game.
* Hệ điều hành được sử dụng là Microsoft Windows 11 Professional.

**3.3.3: Phạm vi kiểm thử**

- Những chức năng được kiểm thử:

* Điều khiển: Kiểm tra xem người chơi có thể điều khiển nhân vật hoặc các đối tượng trong game một cách trơn tru, không bị giật lag hay gặp lỗi.
* Giải đố: Kiểm tra xem các cơ chế giải đố trong game có logic, dễ hiểu và có thể giải được không.
* Giao diện: Kiểm tra xem giao diện người dùng của game có dễ sử dụng, thân thiện và phù hợp với người chơi không.
* Đồ họa: Kiểm tra xem đồ họa của game có đẹp mắt, sắc nét và phù hợp với bối cảnh của game không.
* Âm thanh: Kiểm tra xem âm thanh của game có sống động, chân thực và phù hợp với bối cảnh của game không.
* Lưu dữ liệu Game: Kiểm tra xem game có hỗ trợ lưu game không, có bị mất dữ liệu hay không.

- Những giao diện được kiểm thử:

* Giao diện Menu.
* Giao diện Cài đặt.
* Giao diện Gameplay.

v.v…

**3.3.4: Chiến lược kiểm thử**

- Các giai đoạn kiểm thử

* Kiểm thử ở mức hệ thống (ST) và kiểm thử chấp nhận (UAT)
* Dùng kiểu kiểm thử thủ công (manual test) bao gồm kiểm thử GUI và từng chức năng.
* Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử GUI và chức năng.
* Thiết kế test case theo phương pháp phân vùng tương đương.
* Chỉ thực hiện kiểm thử hồi quy, không thực hiện kiểm thử lại.

**Các loại kiểm thử :**

* Kiểm thử chức năng:

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:  - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng  - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các test case đã được thiết kế đều được thực thi.  Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho lập trình viên khắc phục. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Kiểm thử bằng tay thủ công, tuần tự theo các bước được định nghĩa trong test case |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

Bảng 3.3.4.1: Kiểm thử chức năng

* Kiểm thử giao diện:

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các giao diện được hiển thị đúng với thiết kế về kiểu hiển thị, tính hợp lệ về dữ liệu, tính thích ứng khi chạy trên các trình duyệt khác nhau |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể cho các trình duyệt khác nhau, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:  - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng.  - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng. |
| Tiêu chuẩn dừng | Tất cả các test case đã được thiết kế đều được thực thi.  Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho lập trình viên khắc phục. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Kiểm thử bằng tay thủ công, tuần tự theo các bước được định nghĩa trong test case |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

Bảng 3.3.4.2: Kiểm thử giao diện

**3.3.5 Đánh giá rủi ro :**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Rủi ro** | **Cách khắc phục** | **Mức độ rủi ro** |
| 1 | Thay đổi yêu cầu làm ảnh hưởng đến nguồn nhân lực và chiến lược test. | Lặp lại plan sao cho phù hợp với lịch trình thực tế khi thay đổi yêu cầu, có thể chọn cách tăng thêm nguồn nhân lực cho dự án, hoặc tăng thời gian làm việc ngoài giờ cho nhân viên. | Cao |
| 2 | Sản phẩm mà lập trình viên thực hiện không kịp theo thời gian như lịch trình đề ra. | Yêu cầu cập nhật tiến độ công việc thường xuyên để quản lý kịp thời các thay đổi về thời gian và kĩ thuật. | Cao |
| 3 | Thiếu nguồn nhân lực do có thành viên đột ngột xin nghỉ. | Bổ sung thêm nguồn nhân lực mới hoặc tăng giờ làm việc ngoài giờ. | Thấp |

Bảng 3.3.5: Đánh giá rủi ro

**3.3.6: Điều kiện chấp nhận kiểm thử**

Tỉ lệ test case đạt yêu cầu: 100%.

Tỉ lệ test case gây lỗi: 0%.

Ứng dụng Game chạy ổn định trên công cụ Unity Engine và trên hệ điều hành Windows

### *Phân loại lỗi*

| **Mức độ nghiêm trọng** | **Đặc tả lỗi** |
| --- | --- |
| High | * Không thể khởi động game. * Không thể load màn hình chính. * Không thể điều khiển nhân vật. * Không thể tương tác với các vật thể trong game. * Không thể hoàn thành các nhiệm vụ trong game. |
| Medium | * Lỗi đồ họa. * Lỗi logic, gameplay. |
| Low | * Lỗi nhỏ về giao diện người dùng. * Lỗi chính tả, ngữ pháp. |

Bảng 3.3.6.1: Phân loại lỗi

### *Quy trình xử lý lỗi*

* Ghi nhận lại các lỗi được tìm thấy trong quá trình kiểm thử.
* Xác nhận lại lỗi.
* Sửa lỗi.
* Kiểm thử hồi quy.
* Cập nhật trạng thái lỗi.

**3.3.7: Sản phẩm kiểm thử**

Test Case :

* Kiểm thử chức năng:

|  |  |
| --- | --- |
| **Test case 1** | |
| **Mô tả:** | Kiểm tra Menu và các tùy chọn trong cài đặt |
| **Các bước** | 1. Truy cập vào menu chính.  2. Kiểm tra các chức năng của Menu và các tùy chọn trong cài đặt có thể truy cập được và hình ảnh hiển thị của chúng.  3. Thay đổi các tùy chọn cài đặt và kiểm tra xem các thay đổi có được ứng dụng chính xác hay không. |
| **Dữ liệu đầu vào** | Các chức năng của menu và các tùy chọn cài đặt. |
| **Kết quả mong đợi** | 1. Tất cả các chức năng của menu đều có thể truy cập được.  2. Các tùy chọn trong cài đặt có thể thay đổi và hoạt động chính xác.  3. Tất cả các chức năng của menu và các tùy chọn đều được hiển thị rõ ràng. |
| **Kết quả thực tế** | Hoạt động đúng như kết quả mong đợi |

Bảng 3.3.7.1: Testcase 1

|  |  |
| --- | --- |
| **Test case 2** | |
| **Mô tả:** | Kiểm tra các màn chơi trong game |
| **Các bước** | 1.Chọn màn chơi để chơi.  2. Kiểm tra xem các đối tượng trong cấp độ có được hiển thị chính xác hay không.  3. Kiểm tra xem các tương tác giữa nhân vật và môi trường có hoạt động chính xác hay không.  4. Kiểm tra xem các cơ chế giải đố có hoạt động chính xác hay không. |
| **Dữ liệu đầu vào** | Các cấp độ khác nhau trong game. |
| **Kết quả mong đợi** | 1. Tất cả các đối tượng trong cấp độ đều được hiển thị rõ ràng và chính xác.  2. Các tương tác giữa nhân vật và môi trường trong cấp độ hoạt đông chính xác và mượt mà.  3. Các cơ chế nhặt vàng và bắn súng hoạt động tốt. |
| **Kết quả thực tế** | 1. Một số đối tượng chưa được hiển thị rõ ràng.  2. Các tương tác giữa nhân vật và môi trường hoạt động mượt mà.  3. Các cơ chế nhặt vàng và gây sát thương hoạt động tốt. |

Bảng 3.3.7.2: TestCase 2

|  |  |
| --- | --- |
| **Test case 3** | |
| **Mô tả:** | Kiểm tra các cơ chế trong game |
| **Các bước** | 1.Chọn nhân vật để chơi.  2. Kiểm tra xem các cơ chế tấn công trong game chính xác hay không.  3. Kiểm tra xem các tương tác giữa nhân vật và cơ chế tấn công và nhận sát thương có hoạt động chính xác hay không.  4. Kiểm tra xem các cơ chế nhặt xu có hoạt động chính xác hay không. |
| **Dữ liệu đầu vào** | Nhân vật và enemy trong game. |
| **Kết quả mong đợi** | 1. Tất cả các cơ chế trong game đều được hoạt động chính xác.  2. Các tương tác giữa nhân vật và cơ chế hoạt đông chính xác và mượt mà. |
| **Kết quả thực tế** | Hoạt động đúng như kết quả mong đợi. |

Bảng 3.3.7.3: Testcase 3

|  |  |
| --- | --- |
| **Test case 4** | |
| **Mô tả:** | Kiểm tra các tương tác giữa nhân vật và môi trường trong game |
| **Các bước** | 1.Kiểm tra xem có thể di chuyển nhân vật qua lại 4 phía trong môi trường một cách tự do không.  2. Kiểm tra xem có thể hiển thị khả năng tác động của nhân vật khi muốn tương tác với các đối tượng trong môi trường một cách chính xác hay không.  3. Kiểm tra xem có thể di chuyển nhân vật tác động đến các địa hình một cách đặc biệt hay không. |
| **Dữ liệu đầu vào** | Các đối tượng có thể tương tác và địa hình trong game. |
| **Kết quả mong đợi** | 1. nhân vật có thể di chuyển qua lại tự do.  2. Các hiển thị về tương tác giữa nhân vật và các đối tượng trong môi trường chính xác và rõ ràng.  3. Khi nhân vật di chuyển trên các địa hình đặc biệt thì có hiệu ứng đặc biệt như hình ảnh, âm thanh. |
| **Kết quả thực tế** | Hoạt động đúng như kết quả mong đợi. |

Bảng 3.3.7.4: Testcase 4

* Kiểm thử hiệu năng:

|  |  |
| --- | --- |
| * **Test case 5** | |
| **Mô tả:** | Kiểm tra hiệu năng của game trên các thiết bị có cấu hình khác nhau. |
| **Các bước** | 1. Chơi game trên các thiết bị lần lượt có cấu hình thấp, trung bình và cao.  2. Kiểm tra xem game có chạy ổn định và không bị giật trên các thiết bị có cấu hình thấp, trung bình và cao hay không. |
| **Dữ liệu đầu vào** | Các thiết bị có cấu hình khác nhau. |
| **Kết quả mong đợi** | Game chạy ổn định và không bị giật trên các thiết bị có cấu hình khác nhau. |
| **Kết quả thực tế** | 1. Ở thiết bị có cấu hình thấp, game đôi khi có tình trạng bị giảm fps và giật lag.  2. Ở thiết bị có cấu hình trung bình và cao thì game chạy ổn và mượt mà. |

Bảng 3.3.7.5: Testcase 5

|  |  |
| --- | --- |
| **Test case 6** | |
| **Mô tả:** | Kiểm tra hiệu năng của game khi chơi ở các độ phân giải khác nhau. |
| **Các bước** | Kiểm tra xem game có chạy ổn định và không bị giật khi chơi ở các độ phân giải khác nhau không. |
| **Dữ liệu đầu vào** | Các độ phân giải khác nhau trên thiết bị. |
| **Kết quả mong đợi** | Game chạy ổn định và không bị giật ở các độ phân giải khác nhau trên thiết bị. |
| **Kết quả thực tế** | Hoạt động đúng với kết quả mong đợi. |

Bảng 3.3.7.6: Testcase 6

* Kiểm thử khả năng sử dụng của người chơi

|  |  |
| --- | --- |
| * **Test case 7** | |
| **Mô tả:** | Kiểm tra khả năng điều khiển nhân vật của người chơi trong game |
| **Các bước** | 1. Di chuyển nhân vật sang trái, phải, trên , dưới bằng 4 phím bấm ADWS và joystick, kiểm tra xem nhân vật có di chuyển đúng cách khi nhấn phím không.  2. Thực hiện thao tác nhấn phím bấm Space để làm cho nhân vật lăn, kiểm tra xem nhân vật có lăn lên khi nhấn nút không. |
| **Dữ liệu đầu vào** | Các thao tác từ người chơi |
| **Kết quả mong đợi** | 1. Nhân vật di chuyển đúng cách và chính xác khi người chơi di chuyển bằng 4 phím bấm.  2. Nhân vật lăn khi người chơi nhấn phím Space. |
| **Kết quả thực tế** | Hoạt động đúng với kết quả mong đợi. |

Bảng 3.3.7.7: Testcase 7

|  |  |
| --- | --- |
| **Test case 8** | |
| **Mô tả:** | Kiểm tra khả năng kết nối game với các thiết bị đầu vào khác từ người chơi. |
| **Các bước** | 1. Kết nối thiết bị chơi game với một thiết bị đầu vào khác như tay cầm.  2. Kiểm tra xem trò chơi có nhận thiết bị đầu vào không  3. Thực hiện điều khiển nhân vật và tương tác với các đối tượng trong game. |
| **Dữ liệu đầu vào** | Các thao tác từ người chơi |
| **Kết quả mong đợi** | 1. Trò chơi nhận thiết bị đầu vào.  2. Người chơi điều khiển nhân vật và tương tác thành công bằng thiết bị đầu vào. |
| **Kết quả thực tế** | 1. Trò chơi nhận thiết bị đầu vào thành công.  2. Xuất hiện lỗi không thao tác được với trò chơi khi sử dụng thiết bị đầu vào. |

Bảng 3.7.7.8: Testcase 8

* Kiểm thử khả năng ổn định

|  |  |
| --- | --- |
| **Test case 9** | |
| **Mô tả:** | Kiểm tra khả năng của game khi chạy trong thời gian dài |
| **Các bước** | 1. Thực hiện chơi game trong thời gian dài.  2. Kiểm tra xem game có bị lỗi hoặc crash trong thời gian chơi dài.  3. Kiểm tra xem thiết bị chơi game có vượt mức nhiệt khi chạy game trong thời gian dài không. |
| **Dữ liệu đầu vào** | Thiết bị chơi game và trò chơi |
| **Kết quả mong đợi** | 1. Trò chơi không bị lỗi hay crash khi chạy game trong thời gian dài.  2. Thiết bị chơi game duy trì mức nhiệt bình thường khi chạy game trong thời gian dài. |
| **Kết quả thực tế** | Hoạt động đúng với kết quả mong đợi. |

Bảng 3.3.7.9: Testcase 9

**Kết quả:**

Nhìn chung, dựa vào các kết quả thực tế từ các ca kiểm thử, ứng dụng game Great War Of The Moon đã đáp ứng được các yếu tố yêu cầu về chức năng, hiệu năng, khả năng sử dụng, và độ ổn định.

Tuy nhiên, vẫn còn một số lỗi cần khắc phục, bao gồm:

* Ca kiểm thử thứ 2: Một số đối tượng trong game chưa được hiển thị rõ ràng ở trong game.
* Ca kiểm thử thứ 5: Vẫn xảy ra hiện tượng giật lag và giảm FPS khi chạy game ở thiết bị có cấu hình thấp và sinh ra nhiều đối tượng trong game.
* Ca kiểm thử thứ 10: Chưa thể kết nối thiết bị đầu vào như tay cầm để chơi game một cách hoàn thiện.

**Đề xuất cải thiện:**

Cần triển khai các phương pháp sau để hoàn thiện game:

* Kiểm tra và rà soát toàn bộ đối tượng đang hiển thị chưa rõ ràng ở các cấp độ của game. Thực hiện xử lý các thành phần của chúng.
* Xây dựng, sửa đổi lại các đoạn mã liên quan đến kết nối thiết bị để có thể sử dụng thiết bị đầu vào
* Tìm hiểu và sử dụng các design pattern thiết thực hơn cho việc điều chỉnh, vận hành các tài nguyên và dữ liệu của game nhằm tối ưu hóa hiệu suất của game trên các thiết bị chơi game có cấu hình thấp.

# KẾT LUẬN

**Kết quả đạt được**

* **Về công nghệ**
* Vận dụng thành thục Unity engine và các thư viện của Unity
* Thành thạo các thuật toán và thư viện hỗ trợ trong C#
* Hiểu và nắm bắt được cách sử dụng script, inspector, SingleTon …
* **Về ứng dụng sản phẩm**
* Hoàn thành kịch bản game và phân tích thiết kế trò chơi.
* Hoàn thành việc cài đặt và phát triển trò chơi Great War Of The Moon.
* Trò chơi hiện tại đã xây dựng được cơ bản các màn game và chức năng cơ bản.
* Màn chơi được thiết kế rõ ràng, chi tiết

**Hạn chế**

* Hình ảnh chưa được tối ưu. Hình ảnh một số đồ vật còn vỡ
* Chuyển màn chưa được tối ưu, chưa có hiệu ứng chuyển màn
* Một số UI đang chưa hợp phong cách game
* Chưa thể kết nối với các thư viện nền tảng trực tuyến bên thứ ba như Steam, Google Play Store.

**Hướng phát triển**

* Tiếp tục tìm hiểu để hoàn thiện các chức năng và tài nguyên còn thiếu sót
* Thêm các chức năng mới cho game như: xem quảng cáo để hồi sinh, xem quảng cáo để mở khóa nhân vật, …
* Nâng cấp ứng dụng để có thể kết nối với nhiều thiết bị chơi game và chạy trên nhiều nền tảng.
* Tích hợp các thư viện bên thứ ba của các nền tảng trực tuyến như Steam, Google Play Store,… để sử dụng các tính năng hỗ trợ và thương mại ứng dụng trên các nền tảng đó.

# TÀI LIỆU THAM KHẢO

* Tìm hiểu về Unity và xây dựng game nhằm phát triển trí tuệ cho trẻ từ 4-6 tuổi, Learn about Unity and build games to develop children's minds from 4-6 years old.
  + Loại tài liệu: Tài liệu giấy - ĐA:Đồ án
  + Tác giả: Nguyễn Trọng Văn (2017600448)
  + Nhà xuất bản: Đại học Công nghiệp Hà Nội
* Năm xuất bản: 2021
* <https://assetstore.unity.com/>
* [https://docs.unity.com](https://docs.unity.com/)/
* <https://learn.unity.com/>
* Ebook Game programming with Unity and C#: A complete beginner’s guide - Casey Hardman - Thư viện điện tử Đại học công nghiệp hà nội
* <https://youtu.be/F5sMq8PrWuM>
* [Control Sprite Rendering Order (Which 2D Objects Show in Front) | Unity 2018 Tutorial - YouTube](https://www.youtube.com/watch?v=t1UwAGFLmrk)
* [HOW TO MAKE HEART/HEALTH SYSTEM - UNITY TUTORIAL](https://www.youtube.com/watch?v=3uyolYVsiWc)
* Ze-Nian Li , Mark S. Drew , Jiangchuan Liu - 2014 - Fundamentals of Multimedia