

BỘ GIÁO DỤC VÀ ĐÀO TẠO

BỘ NÔNG NGHIỆP VÀ PTNT

TRƯỜNG ĐẠI HỌC THỦY LỢI



BÙI QUANG HÙNG

**XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE”
BẰNG UNITY**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI

BÙI QUANG HÙNG

**XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE”
BẰNG UNITY**

Ngành : Kỹ thuật phần mềm

Mã số: 2051063675

NGƯỜI HƯỚNG DẪN

TS. Trương Xuân Nam

HÀ NỘI, NĂM 2024

GẤY BÌA ĐỒ ÁN TỐT NGHIỆP

BÙI QUANG HÙNG

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024



CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc



NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Bùi Quang Hùng

Hệ đào tạo : Đại học chính quy

Lớp: 62PM1

Ngành: Kỹ thuật Phần mềm

Khoa: Công nghệ thông tin

- 1 TÊN ĐỀ TÀI:
XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG
UNITY
- 2 CÁC TÀI LIỆU CƠ BẢN:

- Quy trình giao đồ án tốt nghiệp khoa Công nghệ thông tin Đại học Thủy Lợi (2021)
- Website
- https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation
- <https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/>
- https://help.bizagi.com/platform/en/index.html?intro_welcome.htm

3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung cần thuyết minh	Tỷ lệ
1. Mở đầu	2%
2. Chương I. Tổng quan về ngành game	4%
3. Chương II. Tổng quan về Unity	44%
4. Chương III. Giới thiệu game 3D Action “Lockdown Gunblade” bằng unity	15%
5. Chương IV: Phân tích và thiết kế hệ thống	19%
6. Chương V: Xây dựng và demo game	6%

7. Chương VI: Kết luận	2%
------------------------	----

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Các phần	GVHD
1. Mở đầu	TS. Trương Xuân Nam
2. Chương I. Tổng quan mô hình hóa nghiệp vụ	TS. Trương Xuân Nam
3. Chương II. Tổng quan về phần mềm Unity	TS. Trương Xuân Nam
4. Chương III. Giới thiệu game 3D Action “Lockdown Gunblade” bằng unity	TS. Trương Xuân Nam
5. Chương IV: Mô hình chức năng của game	TS. Trương Xuân Nam
6. Chương V: Xây dựng và demo	TS. Trương Xuân Nam
7. Chương VI: Kết luận	TS. Trương Xuân Nam

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày 07 tháng 04 năm 2024

Trưởng Bộ môn
(Ký và ghi rõ Họ tên)

Giáo viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày.tháng.năm 202...

Chủ tịch Hội đồng
(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày 15 tháng 07 năm 2024.

Sinh viên làm Đồ án tốt nghiệp

(Ký và ghi rõ Họ tên)

Bùi Quang Hùng



TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

**TÊN ĐỀ TÀI: XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE”
BẰNG UNITY**

Sinh viên thực hiện: Bùi Quang Hùng

Lớp: 62PM1

Mã sinh viên : 2051063675

Giáo viên hướng dẫn: ThS. Trương Xuân Nam.

TÓM TẮT ĐỀ TÀI

Trong kỷ nguyên số hiện nay, các sản phẩm công nghệ phải đối mặt với những tiêu chuẩn đánh giá ngày càng cao từ người dùng, đặc biệt là các trò chơi điện tử. Game không chỉ được các game thủ nhiệt tình đánh giá mà cả những người chơi phổ thông cũng có ý kiến. Ngành công nghiệp game hiện đang bùng nổ với tốc độ phát triển vượt bậc, mang đến nhiều tựa game hấp dẫn và chất lượng trong thời gian qua. Trong số đó, thể loại game 3D hành động góc nhìn thứ ba thu hút sự chú ý đặc biệt từ cộng đồng game thủ.

Game 3D hành động theo góc nhìn thứ ba đã trở thành một thể loại phổ biến và được yêu thích trong ngành công nghiệp game. Với góc nhìn từ phía sau nhân vật chính, người chơi có cái nhìn toàn cảnh về môi trường xung quanh, tạo nên một trải nghiệm chơi game đầy sống động và chân thực. Không chỉ đem lại những pha hành động mãn nhãn, tựa game này còn tập trung vào việc xây dựng cốt truyện sâu sắc, phát triển nhân vật phong phú, và cơ chế chiến đấu đa dạng, tạo nên sự hấp dẫn không thể cưỡng lại. Chính nhờ những yếu tố này, game 3D hành động không chỉ thu hút người chơi mà còn là đề tài nghiên cứu phong phú trong lĩnh vực phát triển game. Việc phát triển một tựa game hành động bắn súng theo góc nhìn thứ ba không chỉ yêu cầu sự sáng tạo trong thiết kế nhân vật và bối cảnh, mà còn đòi hỏi sự tối ưu hóa về mặt kỹ thuật để mang lại trải nghiệm mượt mà và chân thực nhất.

Với những đặc điểm nêu trên, đó là lý do em chọn đề tài này để phát triển.

CÔNG NGHỆ SỬ DỤNG

- Unity
- Visual Studio

CÁC MỤC TIÊU CHÍNH

Mục tiêu:

- Nhân vật di chuyển được theo 4 hướng và camera di chuyển theo hướng của nhân vật.
- Góc bắn của nhân vật luôn hướng vào tâm của góc nhìn.
- Xây dựng được map cơ bản bao gồm: quái vật, khung cảnh, chương ngại vật.
- Nhân vật bắn được quái vật.
- Có thể đổi được súng.
- Quái vật sẽ tăng độ khó theo từng cấp độ trò chơi. [1]

Kỹ năng

- Sinh viên tìm hiểu kiến thức về Unity, C# và các chức năng của Unity.
- Sinh viên xây dựng thành công một game cơ bản với các chức năng đơn giản.

KẾT QUẢ DỰ KIẾN

Kiến thức

- Sinh viên hiểu được về những kiến thức lập trình Unity, C#.
- Sinh viên nắm được quy trình xây dựng một 3D Action

Kỹ năng

- Báo cáo tổng hợp về Unity.
- Thu thập yêu cầu, nghiệp vụ bài toán để đưa ra các phương án giải quyết phù hợp.
- Sinh viên lập trình thành công game dựa trên nghiệp vụ bài toán bằng Unity viết bằng ngôn ngữ lập trình C# với bài toán chuyển phát nhanh.
- Xây dựng một trò chơi hợp lý và thú vị.

LỜI CAM ĐOAN

Tôi xin cam đoan Đồ án tốt nghiệp “Xây dựng game 3D Action Lockdown GunBlade bằng Unity” là công trình nghiên cứu được thực hiện bởi chính tôi, có sự hướng dẫn hỗ trợ từ giáo viên hướng dẫn ThS. Trương Xuân Nam. Các kết quả trong Đồ án tốt nghiệp này là trung thực, những lý thuyết được trình bày trong báo cáo đều được sử dụng từ các tài liệu mà tôi đã tham khảo và được trích dẫn rõ ràng ở phần tài liệu tham khảo đã có trong báo cáo.

Tác giả ĐATN/KLTN

Chữ ký



Bùi Quang Hùng

LỜI CẢM ƠN

Báo cáo tốt nghiệp chuyên ngành Kỹ thuật phần mềm với đề tài “Xây dựng game 3D Action LockDown GunBlade bằng Unity” là kết quả của quá trình cố gắng không ngừng của bản thân và được sự khích lệ của các thầy, bạn bè và người thân. Qua trang viết này, em xin gửi lời cảm ơn tới những người đã giúp đỡ em trong thời gian học tập và nghiên cứu vừa qua.

Em xin tỏ lòng kính trọng và biết ơn sâu sắc với thầy TS.Trương Xuân Nam đã trực tiếp hướng dẫn, góp ý và cung cấp tài liệu thông tin cần thiết, giúp em hoàn thành đồ án tốt nghiệp một cách tốt nhất.

Em cũng xin chân thành cảm ơn lãnh đạo trường Đại học Thủy lợi, khoa Công nghệ thông tin và Bộ môn đã tạo điều kiện thuận lợi để em có thể học tập và rèn luyện trong suốt quãng thời gian vừa qua.

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH.....	xi
DANH MỤC BẢNG BIỂU.....	xii
DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ	xiii
MỞ ĐẦU	1
1. Đặt vấn đề	1
2. Bố cục của đồ án	1
CHƯƠNG 1 : TỔNG QUAN VỀ NGÀNH GAME	3
1.1 Lịch sử	3
1.2 Các công cụ làm game.....	4
1.3 Các ngôn ngữ làm game	5
CHƯƠNG 2 : TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY	7
2.1 . Unity là gì?	7
2.2 . Những điểm nổi bật của Unity	7
2.3 . Các thành phần trong Unity Editor.....	9
2.3.1 . Cửa sổ Scene	9
2.3.2 . Cửa sổ Hierarchy	10
2.3.3 . Cửa sổ Game	10
2.3.4 . Cửa sổ Project.....	11
2.3.5 . Cửa sổ Inspector	12
2.4 . Các khái niệm cơ bản trong Unity.....	13
2.4.1 . GameObject	13
2.4.2 . Component	13
2.4.3 . Sprite.....	13
2.4.4 . Animation	13
2.4.5 . Key Frame	14
2.4.6 . Prefabs	14
2.4.7 . Sounds	14
2.4.8 . Script.....	14
2.4.9 . Scenes	15
2.4.10 . Assets.....	15

2.4.11 . Camera.....	15
2.4.12 . Transform	16
2.4.13 . Package.....	16
2.4.14 . Console	16
2.5 . Các hàm API phổ biến thường được sử dụng trong Unity	17
2.5.1 . MonoBehaviour	18
2.5.2 . GameObject	18
2.5.3 . Transform	18
2.5.4 . Component	18
2.5.5 . Rigidbody	18
2.5.6 . Animation	18
2.5.7 . Input.....	19
2.5.8 . Audio	19
2.5.9 . Scene Manager	19
2.5.10 . NavMeshAgent.....	19
2.5.11 . Coroutine	19
2.5.12 . PlayerPrefs.....	19
2.6 . Một số Design Pattern phổ biến thường được dùng trong Unity	20
2.6.1 . Object Pool Pattern.....	20
2.6.2 . State Pattern.....	20
2.6.3 . Singleton Pattern	22
CHƯƠNG 3 : GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY	23
3.1 . Các yếu tố làm nên game.....	23
3.2 . Phân tích thiết kế game.....	24
3.2.1 . Đồ họa	24
3.2.2 . Âm thanh	24
3.2.3 . Giao diện	24
3.2.4 . Điều khiển	25
3.2.5 . Nhiệm vụ	26
3.2.6 . Kẻ thù	26
3.2.7 . Màn chơi.....	27

3.2.8 .Góc nhìn	27
3.3 . Kiểm thử và bảo trì game	28
CHƯƠNG 4 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG	30
4.1 . Xác định tác nhân	30
4.2 . Xây dựng biểu đồ Usecase	30
4.3 . Danh sách các Usecase	30
4.4 . Đặc tả Use-case	31
4.4.1 . Use-case “Play Game”	31
4.4.2 . Use-case “Select Weapon”	33
4.4.3 . Use-case “Setting”	35
4.4.4 . Use-case “Attack Enemy”	37
4.4.5 . Use-case “Exit”	39
4.5 . Cơ sở dữ liệu.....	41
CHƯƠNG 5 : XÂY DỰNG GAME VÀ DEMO.....	43
CHƯƠNG 6 : KẾT LUẬN.....	46
6.1 . Kết quả đạt được của đề tài	46
6.2 . Hạn chế của đề tài.....	46
6.3 . Hướng phát triển của đề tài	46
TÀI LIỆU THAM KHẢO	Error! Bookmark not defined.

DANH MỤC CÁC HÌNH ẢNH

Hình 2.1: Logo Unity	7
Hình 2.2: Các nền tảng hỗ trợ Unity	8
Hình 2.3: Cửa sổ Scene	9
Hình 2.4: Cửa sổ Hierarchy	10
Hình 2.5: Cửa sổ Game	11
Hình 2.6: Cửa sổ Project.....	12
Hình 2.7: Cửa sổ Inspector	13
Hình 2.8: Animation của Unity	14
Hình 2.9: Assets của Unity	15
Hình 2.10: Camera của Unity	16
Hình 2.11: Cửa sổ Console.....	17
Hình 2.12: State Pattern.....	21
Hình 3.1 Giao diện ngoài sảnh game.....	25
Hình 3.2 Giao diện trong môi trường game	25
Hình 3.3 Kẻ thù trong game	26
Hình 3.4: Camera Cinemachine	28
Bảng 4.1: Kịch bản Play Game	31
Bảng 4.2: Kịch bản Select Weapon	33
Bảng 4.4: Kịch bản Setting.....	35
Bảng 4.5: Kịch bản Attack Enemy	37
Bảng 4.6: Kịch bản Exit	39

DANH MỤC BẢNG BIỂU

Bảng 4.1: Kịch bản Play Game	31
Bảng 4.2: Kịch bản Select Weapon	33
Bảng 4.4: Kịch bản Setting.....	35
Bảng 4.5: Kịch bản Attack Enemy	37
Bảng 4.6: Kịch bản Exit	39

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

Từ viết tắt	Từ đầy đủ	Nghĩa tiếng việt
CSDL	Cơ sở dữ liệu	
BM	Bộ môn	
GVHD	Giáo viên hướng dẫn	
SV	Sinh viên	

MỞ ĐẦU

1. Đặt vấn đề

Trong những năm gần đây, ngành công nghiệp trò chơi điện tử đã chứng kiến sự phát triển vượt bậc, đặc biệt là trong lĩnh vực game hành động 3D. Với sự tiến bộ không ngừng của công nghệ đồ họa và phần cứng, các trò chơi 3D đã mang lại cho người chơi những trải nghiệm sống động và chân thực hơn bao giờ hết. Trong bối cảnh đó, việc phát triển một trò chơi hành động 3D không chỉ đáp ứng nhu cầu giải trí mà còn mở ra cơ hội kinh doanh lớn trong thị trường game đầy tiềm năng.

Game hành động 3D là một thể loại mà người chơi điều khiển nhân vật từ góc nhìn phía sau, tập trung vào các cuộc chiến đấu và di chuyển linh hoạt, thể loại này thường được áp dụng trong các trò chơi có yếu tố hành động cao và kịch tính. nổi bật với khả năng cung cấp một góc nhìn rộng mở, cho phép người chơi quan sát toàn cảnh môi trường xung quanh nhân vật. Điều này không chỉ giúp người chơi định vị và điều khiển nhân vật dễ dàng hơn mà còn tăng cường khả năng phản ứng trước các tình huống bất ngờ xảy ra từ nhiều phía.

Phát triển một trò chơi hành động 3D đòi hỏi sự đầu tư lớn về thời gian, công sức. Một số thách thức chính bao gồm:

Một số thách thức trong việc phát triển game hành động 3D bao gồm:

Điều chỉnh góc quay camera để người chơi có thể theo dõi môi trường và các sự kiện xung quanh một cách hiệu quả.

Thiết kế các đối tượng và bối cảnh trong game sao cho người chơi có thể dễ dàng di chuyển và tương tác.

Tạo ra các tình huống và nhiệm vụ hấp dẫn, kích thích sự tham gia và thử thách người chơi.

2. Bố cục của đồ án

Các nhiệm vụ chính của đồ án như sau:

1. Tìm hiểu về ngành game thể hiện qua chương 1.
2. Tổng quan làm game bằng Unity qua chương 2.
3. Tiến hành làm game bắn súng bằng Unity qua chương 3;
4. Phân tích và thiết kế hệ thống qua chương 4.
5. Xây dựng và demo game qua chương 5.
6. Kết luận và hướng phát triển, chương 6.

CHƯƠNG 1 : TỔNG QUAN VỀ NGÀNH GAME

1.1 Lịch sử

Ngành làm game bắt đầu hình thành từ những năm 1950, máy tính trong thời điểm này rất hiếm nên để phát triển game một trò chơi game là rất khó, một số nhà sản xuất phải bỏ rất nhiều công sức ra nghiên cứu mới có thể cho ra được một vài tựa game đơn giản thời bấy giờ.

Nimrod là chiếc máy tính chơi game đầu tiên xuất hiện trên thế giới. Chiếc máy tính này thiết kế chơi được một trò chơi game duy nhất là Nim, phát triển bởi Ferranti, một công ty nghiên cứu thiết bị điện tử có trụ sở chính tại Anh [2].

Bước vào thập niên 70, máy tính tên The Magnavox Odyssey được ra đời. Đây là chiếc máy console đầu tiên đưa game vào một thiết bị rời, sử dụng băng từ, để có thể chơi được người dùng phải đưa băng vào từ khe cắm của console. Nhìn thấy tiềm năng của game console, nhiều nhà sản xuất đã bắt đầu đầu tư hơn để đưa ra nhiều loạt game ấn tượng khác, có thể kể đến như Pong (1972), Space Invaders (1978) [2].

Bước sang thập niên 80 của thế kỷ trước, chiếc máy tính cá nhân đầu tiên ra đời là chiếc máy có thể cắm với một màn hình TV. Với chiếc máy tính cá nhân đời đầu này, người chơi vừa chơi game và vừa có thể tạo ra được 1 tựa game cho chính mình bằng các công cụ lập trình [2].

Cuối năm 80, đầu những năm 90, chiếc đĩa CD ra đời có thể tối ưu hóa để có dung lượng lớn hơn, chứa được nhiều dữ liệu hơn và phù hợp để chứa các sản phẩm game lớn.

Bước sang thập kỷ đầu của thế kỷ 21, sự phát triển của chiếc PC ngày càng mạnh dần, cùng sự phát triển của phần cứng, thì xu hướng phát triển ra một loại game có thể liên kết game thủ từ xa mà không cần phải offline. Những hệ máy chơi game hiện đại như Xbox, PlayStation 3 và Wii cung cấp cho người chơi khả năng chơi game trực tuyến và tương tác với nhau qua mạng. Những thiết bị di động như điện thoại thông minh và máy tính bảng cũng cho phép người chơi chơi game mọi lúc mọi nơi. Những trò chơi nổi bật trong thời kỳ này bao gồm Halo, Grand Theft Auto, World of Warcraft và Angry Birds.

Cho đến nay, ngành công nghiệp game đã trở thành một ngành công nghiệp lớn, có ảnh hưởng đến nhiều lĩnh vực khác nhau và tạo ra hàng tỉ đô la doanh thu hàng năm [2].

1.2 Các công cụ làm game

Trong ngành công nghiệp game, công cụ làm game (game engine) đóng vai trò quan trọng nhất trong quá trình phát triển và sản xuất game. Game engine là một nền tảng phần mềm cung cấp các công cụ và công nghệ cần thiết để tạo ra các trò chơi điện tử. Các công cụ này bao gồm đồ họa, vật lý, âm thanh, trí tuệ nhân tạo, mạng và nhiều yếu tố khác, giúp cho việc phát triển game trở nên hiệu quả và tiết kiệm thời gian hơn. Bên cạnh đó, game engine còn hỗ trợ phát triển đa nền tảng, cho phép các nhà phát triển dễ dàng đưa trò chơi của mình lên nhiều thiết bị khác nhau như PC, console, và di động. Dưới đây là một số game engine phổ biến và mạnh mẽ nhất hiện nay.

- **Unity:** Trong số các công cụ Phát triển Di động hàng đầu, Unity luôn được nhắc đến đầu tiên - một công cụ đa nền tảng được ra mắt bởi Unity Technologies. Công cụ này giúp bạn tạo trò chơi của riêng mình bằng các tính năng phát triển 2D và 3D, Unity sử dụng ngôn ngữ lập trình C# hoặc JavaScript để viết mã nguồn cho game
- **Unreal Engine:** Một công cụ làm game đa nền tảng, cho phép tạo ra các game 3D có đồ họa cao cấp cho nhiều thiết bị khác nhau như máy tính, điện thoại, máy chơi game... Unreal Engine sử dụng ngôn ngữ lập trình C++ hoặc Blueprint để viết mã nguồn cho game.
- **GameMaker Studio:** Một công cụ làm game đơn giản, cho phép tạo ra các game 2D cho nhiều thiết bị khác nhau như máy tính, điện thoại, máy chơi game... GameMaker Studio sử dụng ngôn ngữ lập trình GML hoặc kéo thả để viết mã nguồn cho game.
- **Godot:** là một công cụ lập trình trò chơi mã nguồn mở và miễn phí, được sử dụng để tạo ra các trò chơi 2D và 3D. Nó được phát triển bởi Godot Engine, một công ty có trụ sở tại Thụy Điển. Godot hiện hỗ trợ nhiều loại ngôn ngữ lập trình khác nhau như C#, C++,... và làm việc trên VisualScript.

- **Cocos2d-x**: là một bộ công cụ lập trình game nguồn mở được sử dụng để tạo ra các trò chơi 2D. Cocos2d-x là một bộ công cụ mạnh mẽ và linh hoạt, có thể được sử dụng để tạo ra nhiều loại trò chơi khác nhau, từ trò chơi đơn giản đến trò chơi phức tạp. Có thể sử dụng ngôn ngữ lập trình C++ hoặc Lua để viết mã nguồn cho game.

Trên đây là một số công cụ phát triển game phổ biến hiện nay, ngoài ra còn rất nhiều công cụ khác những game engine đã và đang đóng góp rất lớn vào sự phát triển của ngành công nghiệp game, mang lại cho các nhà phát triển những công cụ mạnh mẽ và linh hoạt để hiện thực hóa ý tưởng sáng tạo của mình. Việc lựa chọn game engine phù hợp sẽ giúp các nhà phát triển tối ưu hóa quy trình làm việc và tạo ra những trò chơi chất lượng cao, đáp ứng được nhu cầu ngày càng đa dạng của người chơi.

1.3 Các ngôn ngữ làm game

Trong quá trình phát triển game, việc lựa chọn ngôn ngữ lập trình phù hợp là một yếu tố then chốt giúp tối ưu hóa hiệu suất và hiệu quả làm việc. Các ngôn ngữ lập trình cung cấp nền tảng cho việc xây dựng logic game, phát triển tính năng và tối ưu hóa trải nghiệm người chơi. Mỗi ngôn ngữ lập trình đều có những ưu điểm và nhược điểm riêng, phù hợp với các loại game và game engine khác nhau. Dưới đây là một số ngôn ngữ lập trình phổ biến nhất được sử dụng trong ngành công nghiệp game hiện nay.

- **C#**: Là ngôn ngữ chính được sử dụng trong Unity, C# nổi tiếng với tính dễ học và mạnh mẽ. Nó cung cấp nhiều thư viện và công cụ hỗ trợ, giúp tăng tốc quá trình phát triển game 2D và 3D.
- **C++**: Được sử dụng rộng rãi trong Unreal Engine và nhiều game engine khác, C++ là ngôn ngữ mạnh mẽ với hiệu suất cao, cho phép các nhà phát triển tạo ra các game với đồ họa phức tạp và yêu cầu tài nguyên hệ thống lớn.
- **JavaScript**: tương tự như C#, JavaScript cũng có thể làm game trên Unity nhưng so với C# thì không mạnh bằng, JavaScript cũng là một ngôn ngữ dễ học và linh hoạt, thích hợp cho việc phát triển các game nhẹ và các dự án game web.
- **GML (GameMaker Language)**: Là ngôn ngữ lập trình được sử dụng trong GameMaker Studio, GML dễ học và cho phép các nhà phát triển nhanh chóng tạo ra các game 2D với logic phức tạp.

- Lua: Được sử dụng trong Cocos2d-x và nhiều engine khác, Lua là ngôn ngữ nhẹ và dễ tích hợp, thích hợp cho việc viết kịch bản và phát triển các tính năng game nhanh chóng.
- Python: Mặc dù không phổ biến như C++ hay C#, Python vẫn được sử dụng trong một số dự án game nhờ tính dễ học và thư viện phong phú. Python thường được sử dụng cho việc tạo prototype và công cụ hỗ trợ trong quá trình phát triển game.

CHƯƠNG 2 : TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY

2.1 . Unity là gì?

Unity là một “cross-platform game engine”, tạm hiểu là công cụ phát triển game đa nền tảng, được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát triển game trên PC, consoles, thiết bị di động và trên websites [1].

Unity3D cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi nên cũng khá dễ sử dụng. Ngôn ngữ lập trình chính của Unity là C#, ngoài ra còn có hỗ trợ cho Javascript.

Unity cũng tận dụng chức năng của các thư viện phần mềm như engine mô phỏng vật lý PhysicX của Nvidia, OpenGL và Direct3D để kết xuất hình ảnh 3D, OpenAL cho âm thanh, ... nên nó hỗ trợ rất mạnh cho công việc lập trình game



Hình 2.1: Logo Unity

2.2 . Những điểm nổi bật của Unity

Chức năng cốt lõi đa dạng, bao gồm: cung cấp công cụ dựng hình (kết xuất đồ họa) cho các hình ảnh 2D hoặc 3D, công cụ vật lý (tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động, trí tuệ nhân tạo, phân luồng, tạo dòng dữ liệu xử lý, quản lý bộ nhớ, dựng ảnh đồ thị và kết nối mạng. Nhờ có các engine mà công việc làm game trở nên ít tốn kém và đơn giản hơn [1].

Hỗ trợ đa nền tảng: Một trong các thế mạnh của Unity3D chính là khả năng hỗ trợ gần như toàn bộ các nền tảng hiện có bao gồm: PlayStation 3, Xbox 360, Wii U, iOS, Android, Windows, Blackberry 10, OS X, Linux, trình duyệt Web và cả Flash. Nói cách

khác, chỉ với một gói engine, các studio có thể làm game cho bất kỳ hệ điều hành nào và dễ dàng convert chúng sang những hệ điều hành khác nhau. Đồng thời, đây cũng là giải pháp cho các game online đa nền tảng – có thể chơi đồng thời trên nhiều hệ điều hành, phần cứng khác nhau như Web, PC, Mobile, Tablet.... [1]



Hình 2.2: Các nền tảng hỗ trợ Unity

Dễ sử dụng: Unity3D được build trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ [1].

Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại [1].

Thư viện tài nguyên và cộng đồng lớn: Unity có một thư viện tài nguyên lớn với hàng ngàn mô hình 3D, vật liệu, âm thanh và các tài nguyên khác sẵn sàng sử dụng. Điều này giúp tạo ra các dự án một cách nhanh chóng và tiết kiệm thời gian [1].

Unity có một cộng đồng phát triển đông đảo. Bạn có thể tìm thấy hàng ngàn tài liệu hướng dẫn, tài liệu trực tuyến và các nguồn hỗ trợ từ cộng đồng để giúp bạn giải quyết các vấn đề phát triển [1].

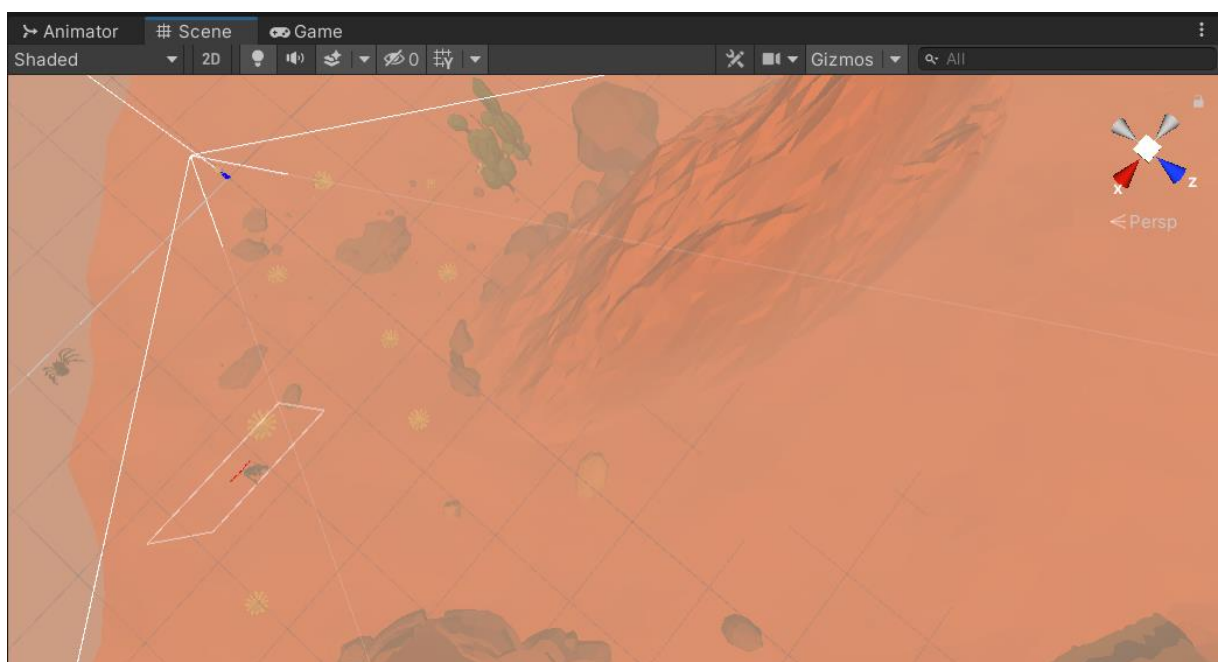
2.3 . Các thành phần trong Unity Editor

2.3.1 . Cửa sổ Scene

Phần này phân hiển thị các đối tượng trong scenes một cách trực quan, có thể lựa chọn các đối tượng, kéo thả, phóng to, thu nhỏ, xoay các đối tượng ...

Phần này có thể thiết lập một số thông số như hiển thị ánh sáng, âm thanh, cách nhìn 2D hay 3D ...

Khung nhìn Scene là nơi bố trí các GameObject như cây cối, cảnh quan, enemy, player, camera, ... trong game. Sự bố trí hoạt cảnh là một trong những chức năng quan trọng nhất của Unity [1].

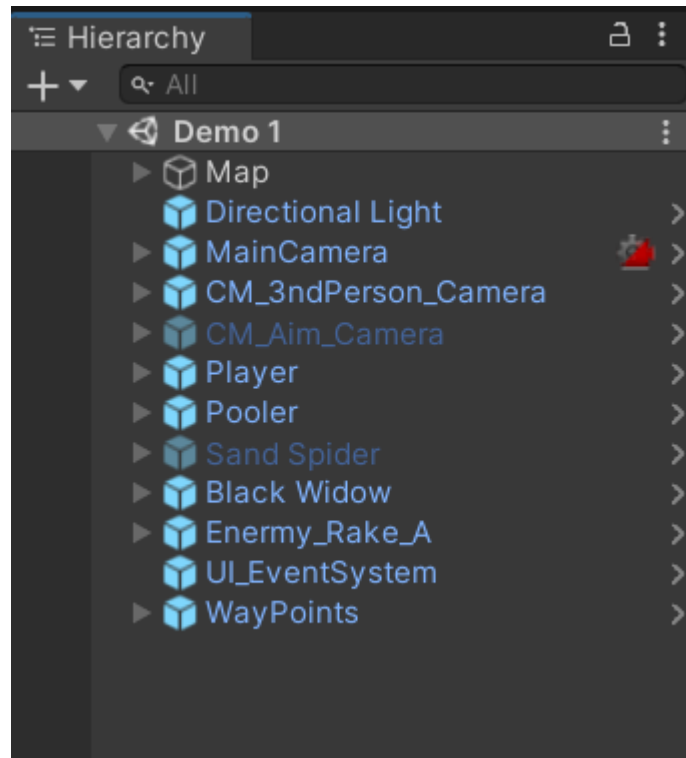


Hình 2.3: Cửa sổ Scene

2.3.2 . Cửa sổ *Hierarchy*

Tab hierarchy là nơi hiển thị các GameObject trên Scene hiện hành. Khi các đối tượng được thêm hoặc xóa trên Scene, tương ứng với các đối tượng đó trong cửa sổ Hierarchy.

Tương tự trong tab Project, Hierarchy cũng có một thanh tìm kiếm giúp quản lý và thao tác với các Game Object hiệu quả hơn đặc biệt là với các dự án lớn [1].



Hình 2.4: Cửa sổ *Hierarchy*

2.3.3 . Cửa sổ *Game*

Đây là màn hình demo Game, là góc nhìn từ camera trong game.

Thanh công cụ trong cửa sổ game cung cấp các tùy chỉnh về độ phân giải màn hình, thông số (stats), gizmos, tùy chọn bật tắt các component... [1].



Hình 2.5: Cửa sổ Game

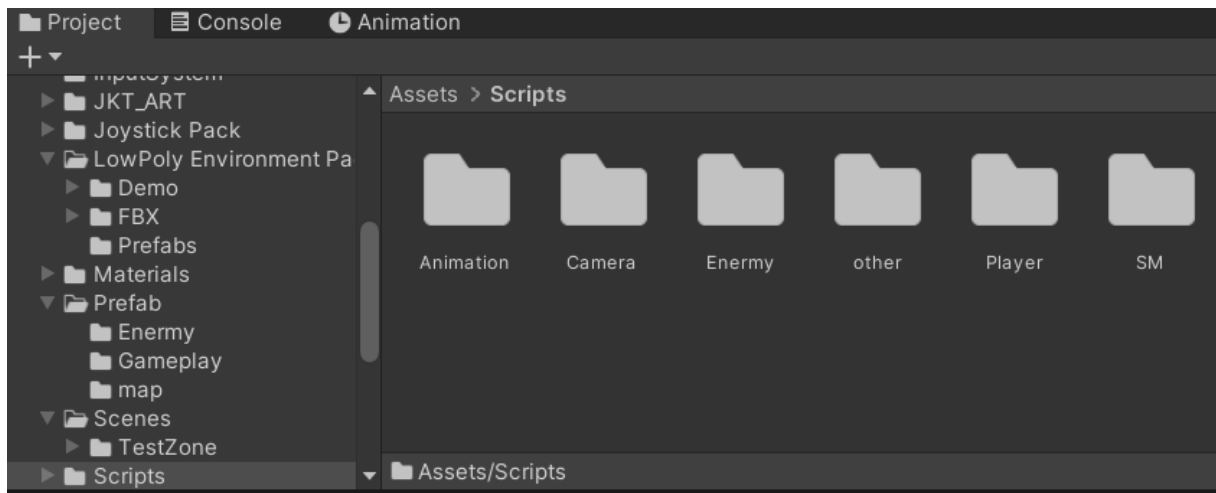
2.3.4 . Cửa sổ Project

Đây là cửa sổ explorer của Unity, hiển thị thông tin của tất cả các tài nguyên (Assets) trong game của bạn.

Cột bên trái hiển thị assets và các mục yêu thích dưới dạng cây thư mục tương tự như Windows Explorer. Khi click vào một nhánh trên cây thư mục thì toàn bộ nội dung của nhánh đó sẽ được hiển thị ở khung bên phải. Ta có thể tạo ra các thư mục mới bằng cách Right click -> Create -> Folder hoặc nhấn vào nút Create ở góc trên bên trái cửa sổ Project và chọn Folder. Các tài nguyên trong game cũng có thể được tạo ra bằng cách này [1].

Phía trên cây thư mục là mục Favorites, giúp chúng ta truy cập nhanh vào những tài nguyên thường sử dụng. Chúng ta có thể đưa các tài nguyên vào Favorites bằng thao tác kéo thả [1].

Đường dẫn của thư mục tài nguyên hiện tại. Chúng ta có thể dễ dàng tiếp cận các thư mục con hoặc thư mục gốc bằng cách click chuột vào mũi tên hoặc tên thư mục [1].



Hình 2.6: Cửa sổ Project

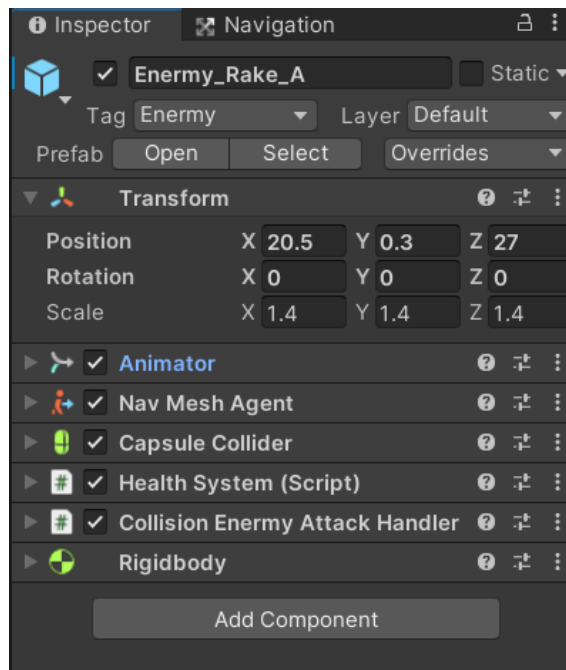
2.3.5 . Cửa sổ Inspector

Cửa sổ Inspector hiển thị chi tiết các thông tin về GameObject đang làm việc, kể cả những component được đính kèm và thuộc tính của nó. Bạn có thể điều chỉnh, thiết lập mọi thông số và chức năng của Game Object thông qua cửa sổ Inspector [1].

Mọi thuộc tính thể hiện trong Inspector đều có thể dễ dàng tùy chỉnh trực tiếp mà không cần thông qua một kịch bản định trước. Tuy nhiên Scripting API cung cấp một số lượng nhiều và đầy đủ hơn do giao diện Inspector là có giới hạn [1].

Các thiết lập của từng component được đặt trong menu. Các bạn có thể click chuột phải, hoặc chọn icon hình bánh răng nhỏ để xuất hiện menu [1].

Ngoài ra Inspector cũng thể hiện mọi thông số Import Setting của asset đang làm việc như hiển thị mã nguồn của Script, các thông số animation, ... [1]



Hình 2.7: Cửa sổ Inspector

2.4 . Các khái niệm cơ bản trong Unity

2.4.1 . *GameObject*

Đây là khái niệm dùng để chỉ những đối tượng cụ thể trong game, nó có thể là bất kỳ một nhân vật hay đồ vật nào đó được xây dựng trong trò chơi [5].

2.4.2 . *Component*

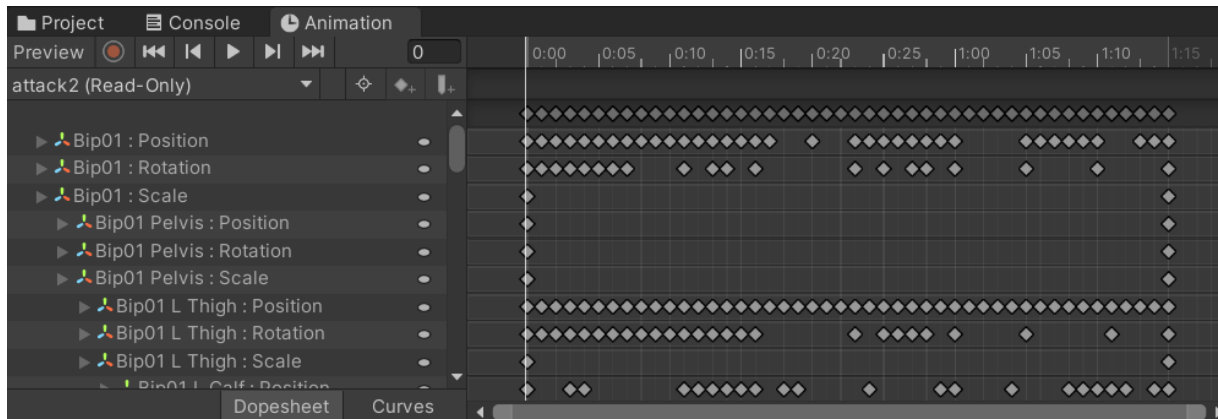
Một GameObject sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động (animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... mỗi thứ như vậy gọi là một component của GameObject [1].

2.4.3 . *Sprite*

Là một hình ảnh 2D của một gameobject có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó [5].

2.4.4 . *Animation*

Là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau [1]



Hình 2.8: Animation của Unity

2.4.5 . Key Frame

Key Frame hay Frame là một trạng thái của một animation. Có thể được tạo nên từ 1 sprite hay nhiều sprite khác nhau [1].

2.4.6 . Prefabs

Là một khái niệm trong Unity, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay từ một đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiền trong game Mario đều có xử lý giống nhau, nên ta chỉ việc tạo ra một đối tượng ban đầu, các đồng tiền còn lại sẽ sử dụng prefabs. Hoặc khi ta lát gạch cho một cái nền nhà, các viên gạch cũng được sử dụng là prefabs [1].

2.4.7 . Sounds

Âm thanh trong game.

2.4.8 . Script

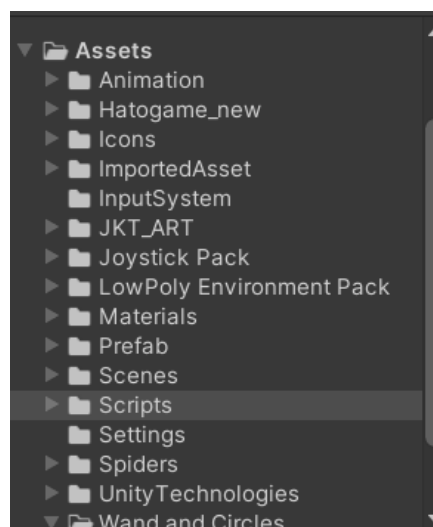
Script là tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Trong Unity có thể dùng C#, Java Script, BOO để lập trình Script [1].

2.4.9 . Scenes

Quản lý tất cả các đối tượng trong một màn chơi của game [1].

2.4.10 . Assets

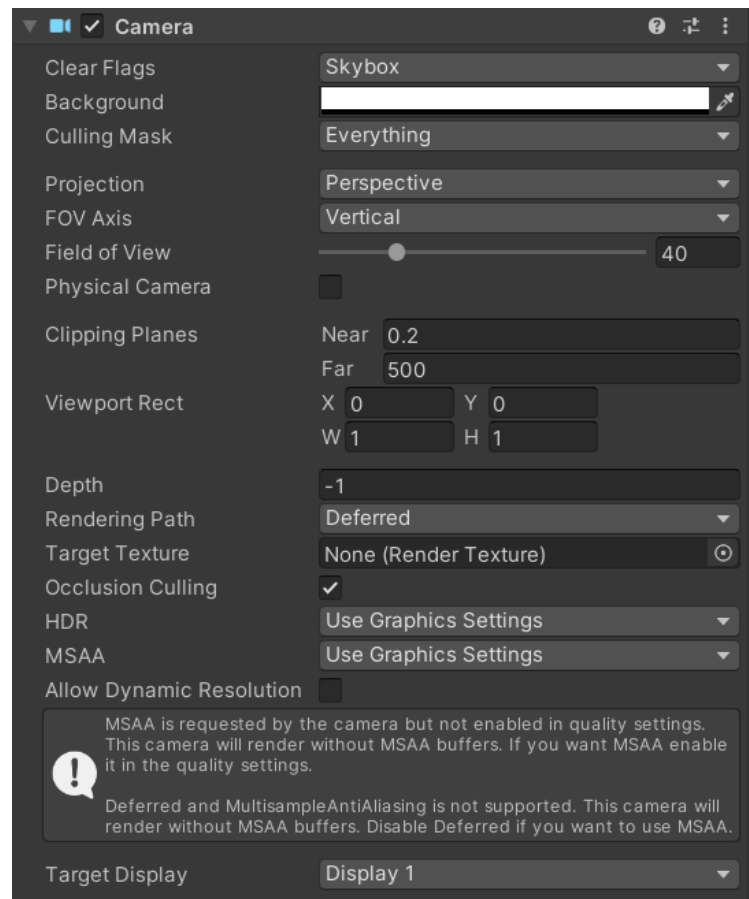
Bao gồm tất cả những gì phục vụ cho dự án game như sprite, animation, sound, script, scenes... [1]



Hình 2.9: Assets của Unity

2.4.11 . Camera

Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game [1].



Hình 2.10: Camera của Unity

2.4.12 . Transform

Là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng [5].

2.4.13 . Package

Là tập hợp những game objects, assets và các file metadata liên quan.

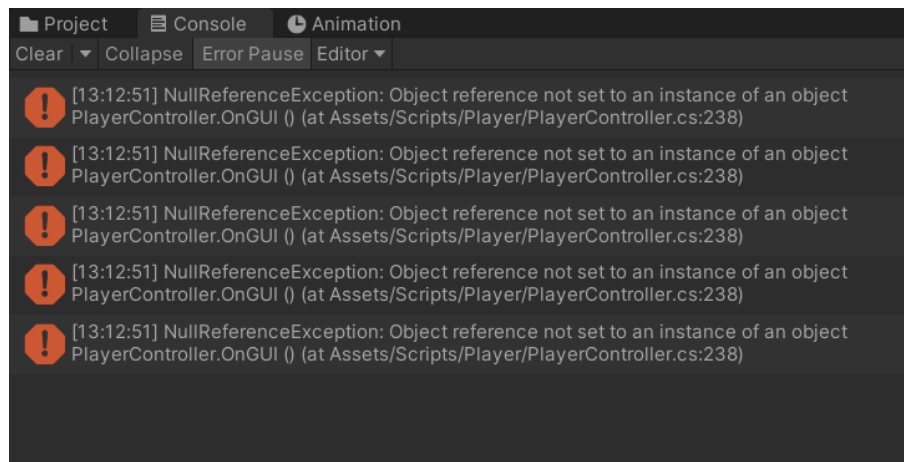
Có thể hiểu đơn giản chúng là các Object liên quan, bao gồm models, scripts, materials...

Đặc biệt, khi sử dụng Unity bạn còn được cung cấp các "standard package" tại Asset Store hoàn toàn miễn phí, bạn có thể Import trực tiếp chúng vào Project mà mình đang thực hiện dễ dàng [5].

2.4.14 . Console

Cửa sổ Console hiển thị các lỗi, cảnh báo và thông báo khác và Editor sinh ra. Những lỗi hoặc cảnh báo này giúp chúng ta tìm ra vấn đề trong dự án để sửa lại lỗi.

Ngoài ra còn giúp ta Debug để print những test case cần thiết để xem code của mình hoạt động như nào [1].



Hình 2.11: Cửa sổ Console

2.5 . Các hàm API phổ biến thường được sử dụng trong Unity

Dưới đây sẽ là một số hàm API thường gặp và một số phương thức và thuộc tính thường được dùng

2.5.1 . *MonoBehaviour*

Là lớp cơ sở cho tất cả script trong Unity, muốn gắn script vào object cần kế thừa lớp này

Start(): được gọi một lần duy nhất khi script bắt đầu kích hoạt

Update(): được gọi mỗi lần vào mỗi frame và được xem như hàm xử lý chính của vòng lặp game.

2.5.2 . *GameObject*

Đại diện cho tất cả các đối tượng trong game

GameObject.Find (string name) : Tìm GameObject theo tên và trả về GameObject đó.

gameObject.AddComponent<T>(): Thêm một component vào GameObject

2.5.3 . *Transform*

Quản lý vị trí, hướng và tỷ lệ của GameObject

transform.position: Truy cập hoặc thay đổi vị trí của đối tượng.

transform.rotation: Truy cập hoặc thay đổi góc quay của đối tượng.

transform.localScale: Truy cập hoặc thay đổi tỷ lệ của đối tượng.

2.5.4 . *Component*

Là thành phần được gắn vào Gameobject

GetComponent<T>(): Lấy một component của loại T từ GameObject.

2.5.5 . *Rigidbody*

Quản lý động lực học của các đối tượng

rigidbody.velocity: Truy cập hoặc thay đổi vận tốc của đối tượng.

rigidbody.AddForce(Vector3 force): Thêm lực vào đối tượng.

2.5.6 . *Animation*

Quản lý hoạt ảnh của đối tượng

`animator.SetBool(string name, bool value)`: Đặt giá trị của một tham số boolean trong Animator.

`SetTrigger(string name)` : Kích hoạt một trigger trong Animator.

2.5.7 . Input

Quản lý đầu vào từ người chơi

`Input.GetKey(KeyCode key)`: Kiểm tra xem phím có đang được nhấn không.

`Input.GetAxis(string axisName)`: Truy cập giá trị của trục đầu vào.

2.5.8 . Audio

Phát âm thanh trong trò chơi

`audioSource.Play()`: Phát âm thanh.

`audioSource.Stop()`: Dừng phát âm thanh.

2.5.9 . Scene Manager

Quản lý các cảnh trong trò chơi

`SceneManager.LoadScene(string sceneName)`: Tải một cảnh mới.

2.5.10 . NavMeshAgent

Quản lý AI di chuyển bằng cách sử dụng lưới điều hướng (NavMesh)

`navMeshAgent.SetDestination(Vector3 target)`: Đặt đích đến cho AI.

`navMeshAgent.speed`: Truy cập hoặc thay đổi tốc độ di chuyển của AI.

2.5.11 . Coroutine

Cho phép thực thi mã không đồng bộ

`StartCoroutine(IEnumerator routine)`: Bắt đầu một coroutine.

`StopCoroutine(IEnumerator routine)`: Dừng một coroutine.

2.5.12 . PlayerPrefs

Lưu và tải các thiết lập người chơi

PlayerPrefs.SetInt(string key, int value): Lưu giá trị số nguyên.

PlayerPrefs.GetInt(string key): Lấy giá trị số nguyên.

2.6 . Một số Design Pattern phổ biến thường được dùng trong Unity

Design Pattern hay mẫu thiết kế, là cách xây dựng một lớp dựa trên một thiết kế nhất định nào đó

2.6.1 . Object Pool Pattern

Trong Unity, để sinh ra một phiên bản copy của một GameObject (thường là prefab), ta sẽ sử dụng hàm Instantiate(), nếu không sử dụng bản copy đó nữa chúng ta sẽ sử dụng hàm Destroy() hay DestroyImmediate(). Hàm Instantiate() sẽ phân bổ, cấp phát tài nguyên bộ nhớ cho copied-GameObject đó, hàm Destroy() xóa bỏ các tài nguyên đã cấp phát và nếu việc này xảy ra liên tục, Garbage Collector (GC) sẽ hoạt động liên tục làm phân mảnh bộ nhớ đồng thời trong frame đó phải chờ GC xử lý xong.

Việc sinh ra và xóa đi các object liên tục như vậy sẽ ảnh hưởng rất lớn đến hiệu suất, Object Pooling Pattern sinh ra nhằm giúp chúng ta tái sử dụng được các object đã tạo

Về ý tưởng, Object Pooling sẽ sinh ra các objects mà ta cần sử dụng trong game trước, sau đó deactivate toàn bộ objects đó nếu chưa sử dụng, điều này có thể làm chậm quá trình khởi động của game.

Như vậy, thay vì Instantiate() các objects vào run-time, chúng ta chỉ việc lôi chúng từ Object Pool ra rồi bật active cho object đó, tương tự, thay vì Destroy(), chúng ta sẽ deactivate object đó và gửi về lại Pool [5].

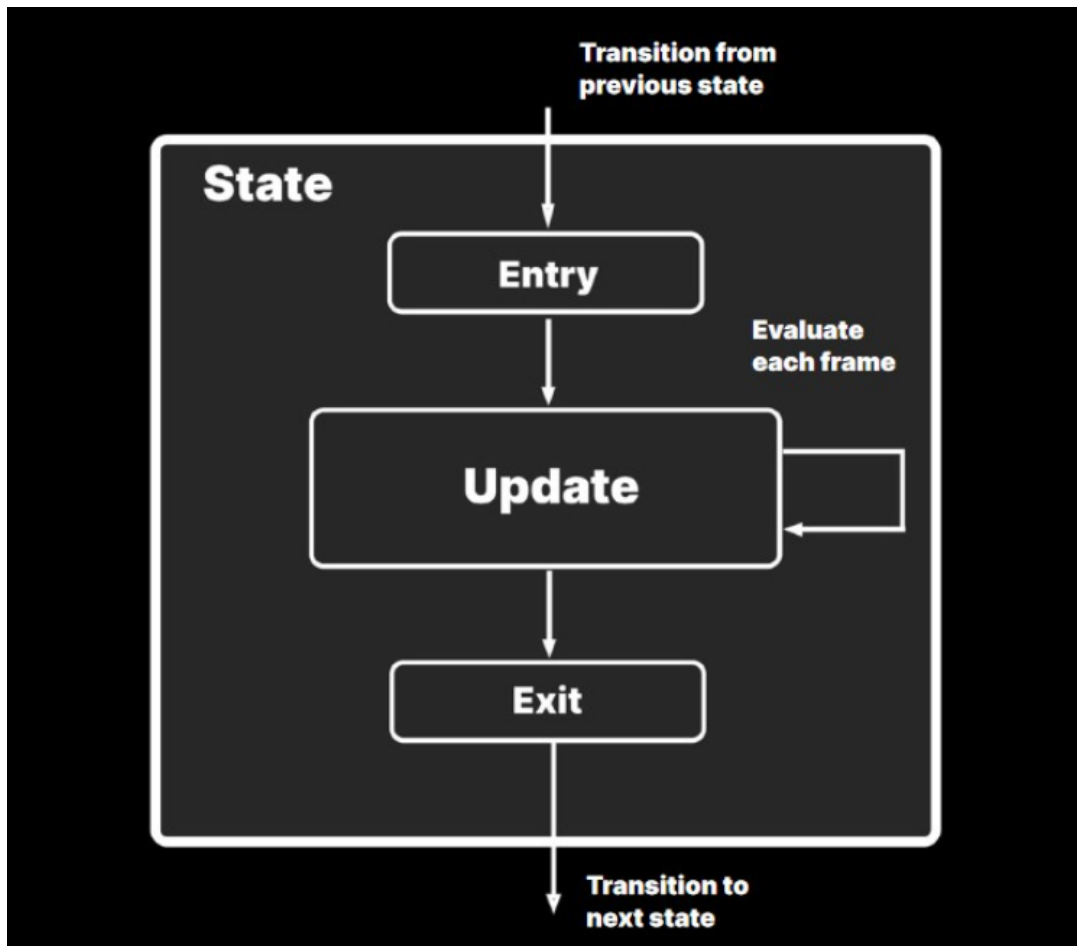
2.6.2 . State Pattern

State Pattern là một mẫu thiết kế hành vi cho phép một object thay đổi hành vi của nó khi trạng thái bên trong của nó thay đổi.

State pattern giúp giải quyết hai vấn đề sau:

- Một đối tượng sẽ thay đổi hành vi của nó khi trạng thái thay đổi
- Hành vi của trạng thái cụ thể được định nghĩa một cách độc lập. Việc thêm các trạng thái mới không ảnh hưởng đến hành vi của các trạng thái hiện có.

Đây là giai đoạn của một trạng thái trong State Pattern, từ khi chuyển từ trạng thái trước đó, thực thi logic trong trạng thái hiện tại, và cuối cùng là chuyển sang trạng thái tiếp theo.



Hình 2.12: State Pattern

Mỗi trạng thái sẽ có 3 thành phần chính:

Entry : Thực thi xử lý logic khi lần đầu chuyển sang trạng thái này

Update: Thực thi logic mỗi frame(đôi khi còn được gọi là Execute hoặc Tick). Bạn có thể chia nhỏ nó thành Update, FixedUpdate hay LateUpdate giống như của MonoBehaviour. Nó cũng xử lý điều kiện chuyển sang trạng thái khác ở đây

Exit: Thực thi xử lý logic trước khi chuyển sang trạng thái khác

Điểm mấu chốt của State Pattern trong Unity là khả năng quản lý và thay đổi hành vi của đối tượng dựa trên trạng thái hiện tại một cách linh hoạt và dễ dàng mở rộng. Pattern này giúp tách biệt logic của từng trạng thái, cho phép thêm mới các trạng thái mà không làm ảnh hưởng đến các trạng thái hiện có. Việc áp dụng State Pattern trong Unity không chỉ giúp tăng tính module hóa và tái sử dụng code mà còn giảm thiểu rủi ro và đảm bảo tính nhất quán trong hành vi của đối tượng qua các trạng thái khác nhau [7].

2.6.3 . Singleton Pattern

Singleton là một Design Pattern, nó đảm bảo mỗi class sẽ chỉ có duy nhất một instance, đồng thời cung cấp một điểm truy cập toàn cục cho Instance này. Trong Unity, Singleton cũng dùng để quản lý các đối tượng mà ta muốn khởi tạo 1 lần và dùng chung cho toàn bộ game.

Singleton có thể rất tiện lợi vì chúng cho phép kết nối các phần trong trò chơi dễ dàng hơn, miễn là biến hoặc hàm tồn tại trên singleton và có thể truy cập công khai thì các tập lệnh khác sẽ có thể sử dụng nó mà không cần bạn phải thiết lập tham chiếu đến nó trước.

Tiện lợi là thế nhưng nó được xem là một bad pattern, có một vài lý do ở Singleton pattern khiến nhiều developer không thích

- Singleton quá public. Nó yêu cầu cung cấp global access, bởi vì là các global instance nên việc sử dụng nó có thể khiến nhiều dependency bị ảnh hưởng, khiến cho việc truy vết bug khó hơn.
- Singleton khiến quá trình test khó khăn. Các bài kiểm tra unit test phải độc lập với nhau nhưng singleton có thể thay đổi trạng thái của nhiều GameObject trong scene, chúng có thể ảnh hưởng đến quá trình test của bạn
- Singleton là high coupling (liên kết chặt chẽ giữa các class). high coupling khiến cho việc tái cấu trúc trở nên khó khăn, nếu bạn thay đổi một component nó có thể ảnh hưởng đến bất kỳ component nào liên kết với nó điều này dẫn đến không clean code [7].

Vì thế đừng lạm dụng Singleton pattern quá nhiều

CHƯƠNG 3 : GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY

3.1 . Các yếu tố làm nên game

Để tạo nên một trò chơi hoàn chỉnh và hấp dẫn, phải kết hợp giữa nhiều yếu tố khác nhau. Những yếu tố này không chỉ góp phần xây dựng cấu trúc của một trò chơi mà còn nâng cao trải nghiệm của người chơi, từ đó có thể giữ chân họ lâu hơn.

Những yếu tố này bao gồm:

- **Đồ họa:** Đây là những hình ảnh minh họa cho các đối tượng nhân vật môi trường và hiệu ứng trong game tạo nên bối cảnh và không gian trò chơi. Đồ họa có thể thiết kế bằng các phần mềm như Photoshop, 3D Studio Max ...
- **Âm thanh:** Đây là yếu tố đóng vai trò tạo ra không khí và cảm xúc cho trò chơi. Âm thanh từ các đối tượng, nhân vật, môi trường và hiệu ứng trong game cần được thu âm, chỉnh sửa và phối hợp một cách tinh tế bằng các phần mềm như Audacity, Sound Forge hoặc FL Studio...
- **Giao diện:** Đây là yếu tố giúp người chơi dễ tương tác với trò chơi, với các thành phần hiển thị như menu, nút bấm, biểu tượng, thông báo và điểm số có nhiệm vụ cung cấp thông tin rõ ràng và các tùy chọn điều khiển cho người chơi
- **Điều khiển:** Liên quan đến cách thức người chơi nhập lệnh và điều khiển đối tượng, nhân vật trong game. Điều khiển có thể sử dụng các thiết bị như bàn phím, chuột, tay cầm, màn hình cảm ứng
- **Nhiệm vụ:** Là mục tiêu mà người chơi phải hoàn thành để tiến triển trong game. Nhiệm vụ có thể bao gồm nhiều loại như giết địch, thu thập vật phẩm hoặc khám phá. Đây là yếu tố gây nên sự hứng thú giúp giữ chân người chơi.
- **Kỹ năng:** Liên quan đến các khả năng đặc biệt mà nhân vật có thể sử dụng để chiến đấu hoặc giải quyết các tình huống trong game. Kỹ năng có thể bao gồm các loại như tấn công, phòng thủ, chữa trị hoặc kích hoạt các khả năng đặc biệt.
- **Kẻ thù:** Là khó khăn thách thức mà người chơi gặp phải trong xuyên suốt game. Kẻ thù có thể là quái vật, robot hoặc những kẻ thù khác có ý định ngăn cản gây hại cho nhân vật trong game, đây là yếu tố tạo nên sự kịch tính trong game.

- Màn chơi: Bao gồm các giai đoạn khác nhau mà người chơi phải vượt qua trong game. Mỗi màn chơi có thể có một bối cảnh, nhiệm vụ và độ khó riêng biệt. Thiết kế màn chơi cần phải phong phú và sáng tạo bởi đây là yếu tố quan trọng khiến cho người chơi có được cảm giác tò mò muốn khám phá
- Góc nhìn: Đây là yếu tố quyết định khung cảnh và góc quay khi nhân vật di chuyển trong game. Góc nhìn là góc nhìn thứ ba, giúp người chơi dễ dàng quan sát và điều khiển nhân vật [5].

3.2 . Phân tích thiết kế game

Phần này sẽ trình bày chi tiết về các yếu tố thiết kế game “Lockdown GunBlade”

3.2.1 . Đồ họa

Nhân vật, hiệu ứng và môi trường trong game đều được thiết kế dưới dạng đồ họa 3D. Dù là một trò chơi hành động bắn súng, màu sắc trong game vẫn rất bắt mắt và dễ chịu, giúp giảm bớt sự căng thẳng khi chơi.

3.2.2 . Âm thanh

Game sử dụng âm thanh để tăng cường hiệu ứng cho các đối tượng và nhân vật, chẳng hạn như tiếng súng nổ, tiếng bước chân khi nhân vật chạy, và tiếng kêu khi nhân vật bị quái vật tấn công. Những âm thanh này giúp tạo ra một trải nghiệm sống động và chân thực hơn cho người chơi.

3.2.3 . Giao diện

Game sử dụng giao diện để hiển thị cho người chơi biết những thông tin và lựa chọn có trong game như menu cài đặt, biểu tượng tạm dừng, menu tạm dừng,...

Đây là hai giao diện chính trong game



Hình 3.1 Giao diện ngoài sảnh game



Hình 3.2 Giao diện trong môi trường game

3.2.4 . Điều khiển

Người chơi có thể điều khiển nhân vật bằng nút bấm trên màn hình đối với điện thoại hoặc bàn phím và chuột đối với máy tính.

3.2.5 . *Nhiệm vụ*

Nhiệm vụ mà người chơi phải hoàn thành trong game bao gồm:

- **Giết địch:** Đây là nhiệm vụ chính yếu trong game, nơi người chơi sẽ sử dụng vũ khí để tiêu diệt quái vật và kẻ thù xuất hiện trong suốt hành trình
- **Khám phá:** Nhiệm vụ này yêu cầu người chơi phải đi khắp bản đồ, khám phá các khu vực ẩn và truy lùng kẻ địch để tiêu diệt chúng

3.2.6 . *Kẻ thù*

Để tăng thêm phần hấp dẫn và thử thách, người chơi sẽ phải đối mặt với rất nhiều kẻ thù đa dạng và nguy hiểm trong suốt hành trình của mình trong game. Những kẻ thù này không chỉ xuất hiện với tần suất dày đặc mà còn sở hữu sức mạnh và độ khó gia tăng theo từng màn chơi.

- **Quái vật:** Đây là những kẻ thù cơ bản nhưng không kém phần nguy hiểm. Chúng thường xuất hiện thành từng nhóm và tấn công đồng loạt, đòi hỏi người chơi phải có khả năng di chuyển linh hoạt và tấn công chính xác.
- **Boss:** Mỗi màn chơi sẽ có một kẻ thù boss với sức mạnh vượt trội và kỹ năng đặc biệt. Để đánh bại được kẻ thù boss, người chơi cần phải hiểu rõ về điểm yếu của chúng và sử dụng tối đa các kỹ năng của mình.

Đây là một trong số chúng



Hình 3.3 Kẻ thù trong game

3.2.7 . Màn chơi

Game có nhiều màn chơi khác nhau mà người chơi phải vượt qua, với độ khó gia tăng theo từng màn chơi. Các màn chơi được thiết kế tỉ mỉ để mang lại trải nghiệm đa dạng và đầy thử thách, đảm bảo rằng người chơi luôn phải cải thiện kỹ năng và chiến thuật của mình để tiếp tục tiến xa hơn.

Số lượng màn chơi: Có 3 màn chơi tất cả. Mỗi màn sẽ có bối cảnh và loại kẻ thù cần tiêu diệt khác nhau, tạo nên sự phong phú và đa dạng trong gameplay.

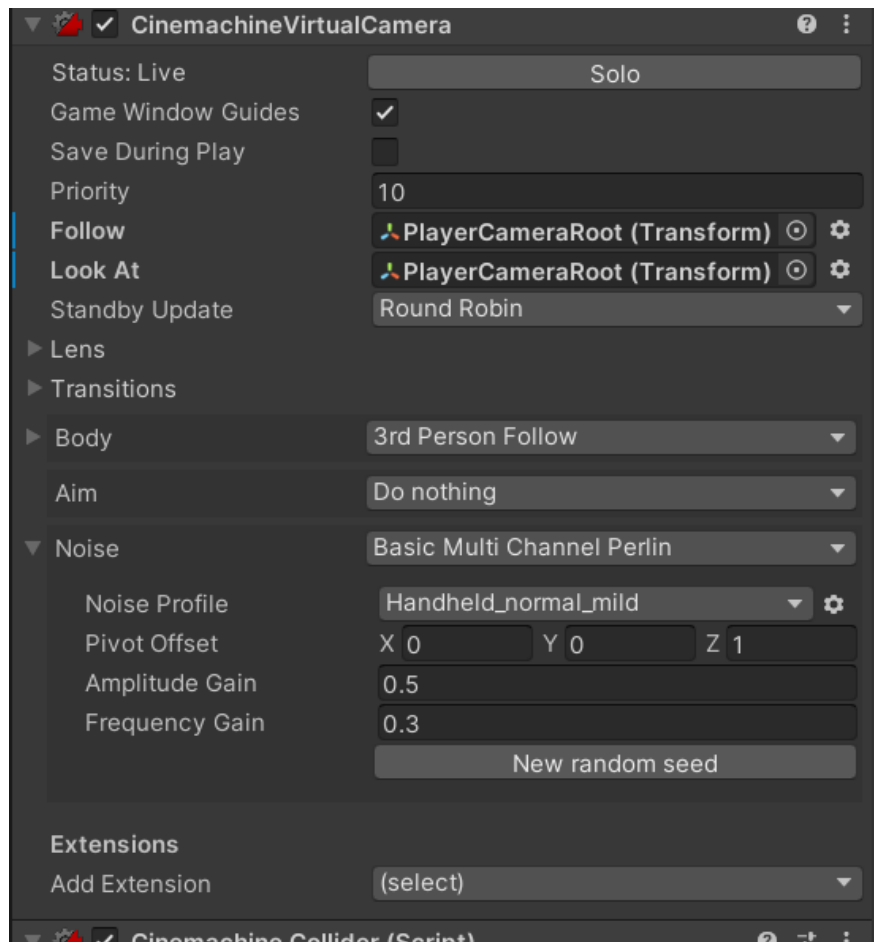
Bối cảnh: Mỗi màn chơi được đặt trong một bối cảnh riêng biệt: sa mạc, khu rừng,.... Mỗi bối cảnh không chỉ mang đến trải nghiệm thị giác mới mẻ mà còn ảnh hưởng đến chiến thuật chiến đấu của người chơi.

Loại kẻ địch: Ở mỗi màn, người chơi sẽ đối đầu với các loại quái vật khác nhau, độ khó và sức mạnh cũng khác nhau

3.2.8 .Góc nhìn

Game được thiết kế với góc nhìn thứ ba, mang lại trải nghiệm trực quan và phong phú cho người chơi. Góc nhìn thứ ba giúp người chơi có cái nhìn toàn diện về nhân vật chính và môi trường xung quanh, từ đó dễ dàng nắm bắt tình hình và đưa ra các quyết định chiến thuật.

Để hiện thực hóa góc nhìn thứ ba một cách tối ưu, game sử dụng Camera Cinemachine, một công cụ mạnh mẽ trong Unity giúp tạo ra các chuyển động camera mượt mà và chuyên nghiệp.



Hình 3.4: Camera Cinemachine

3.3 . Kiểm thử và bảo trì game

Phần này sẽ trình bày các phương pháp kiểm thử và bảo trì game 3D action GunBlade LockDown

Kiểm thử game : là một phần trong quá trình phát triển game, nhằm kiểm soát chất lượng của sản phẩm được tạo ra, kiểm tra và đánh giá để đảm bảo rằng nó hoạt động đúng theo thiết kế, không có lỗi (bugs) và mang lại trải nghiệm tốt nhất cho người chơi [6].

Có các loại kiểm thử sau:

- Người kiểm thử chức năng: tìm kiếm các vấn đề chung trong game hoặc giao diện người dùng và đồ họa của nó, chẳng hạn như các vấn đề về : logic game, vận hành game, các vấn đề về tính ổn định và tính toàn vẹn nội dung game. Kiểm tra giao diện người dùng đảm bảo sự thân thiện với người dùng.

- Kiểm thử hiệu năng: Kiểm tra hiệu suất tổng thể của game. Thực hiện điều chỉnh hiệu suất để tối ưu hóa tốc độ của game.
- Kiểm thử khả năng tương thích: Kiểm tra xem game có tương thích trên các thiết bị khác nhau hay không và tính tương thích trên các cấu hình phần cứng và phần mềm khác nhau.
- Kiểm thử giao diện người dùng (UI Testing): Đảm bảo rằng giao diện người dùng thân thiện và không có lỗi hiển thị.

Bảo trì: Là quá trình cập nhật và sửa chữa game sau khi phát hành để khắc phục các lỗi, cải thiện hiệu năng và tăng cường tính năng của game.

Có các loại bảo trì sau:

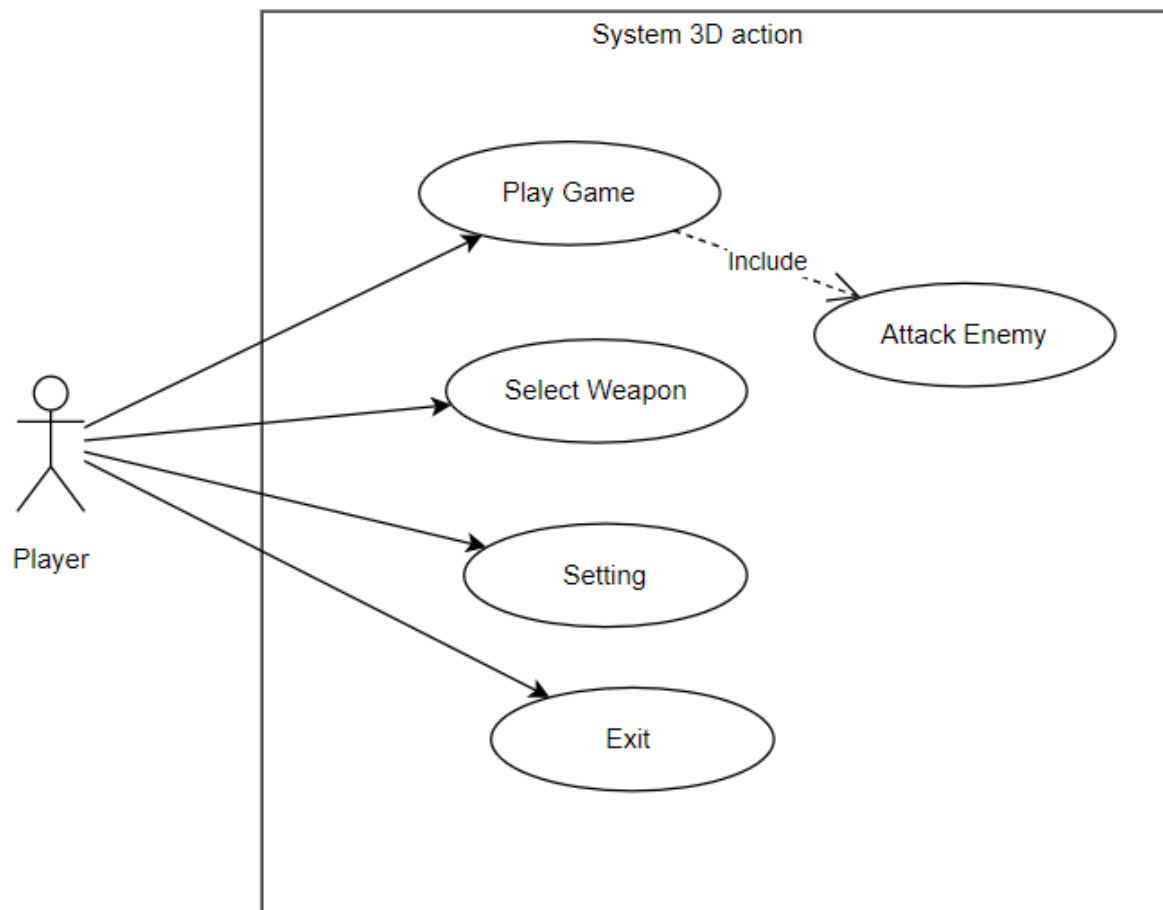
- Bảo trì sửa chữa: Là quá trình cập nhật game để sửa chữa các lỗi đã được phát hiện trong quá trình kiểm thử hoặc phản hồi từ người chơi, bao gồm việc sửa lỗi (bugs), và giải quyết các vấn đề liên quan đến hiệu suất.
- Bảo trì hoàn thiện: Đây là quá trình cải thiện và nâng cấp game dựa trên phản hồi của người chơi và những tiến bộ công nghệ mới, bao gồm việc thêm tính năng mới, cải thiện đồ họa, và tối ưu hóa gameplay.
- Bảo trì phòng ngừa: Đây là quá trình kiểm tra và bảo trì định kỳ để ngăn ngừa các vấn đề trước khi chúng xảy ra., bao gồm việc kiểm tra tối ưu hóa mã nguồn, và cập nhật hệ thống để đảm bảo game luôn chạy mượt mà.

CHƯƠNG 4 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1 . Xác định tác nhân

Player: Điều khiển nhân vật và tiêu diệt Enemy

4.2 . Xây dựng biểu đồ Usecase



Hình 4.1: Mô hình UseCase tổng quát

4.3 . Danh sách các Usecase

Play Game: Người chơi bắt đầu chơi game

Select Weapon: Chọn vũ khí đi kèm

Setting: Điều chỉnh hiệu ứng game (âm thanh, ...)

Exit: Thoát khỏi game

4.4 . Đặc tả Use-case

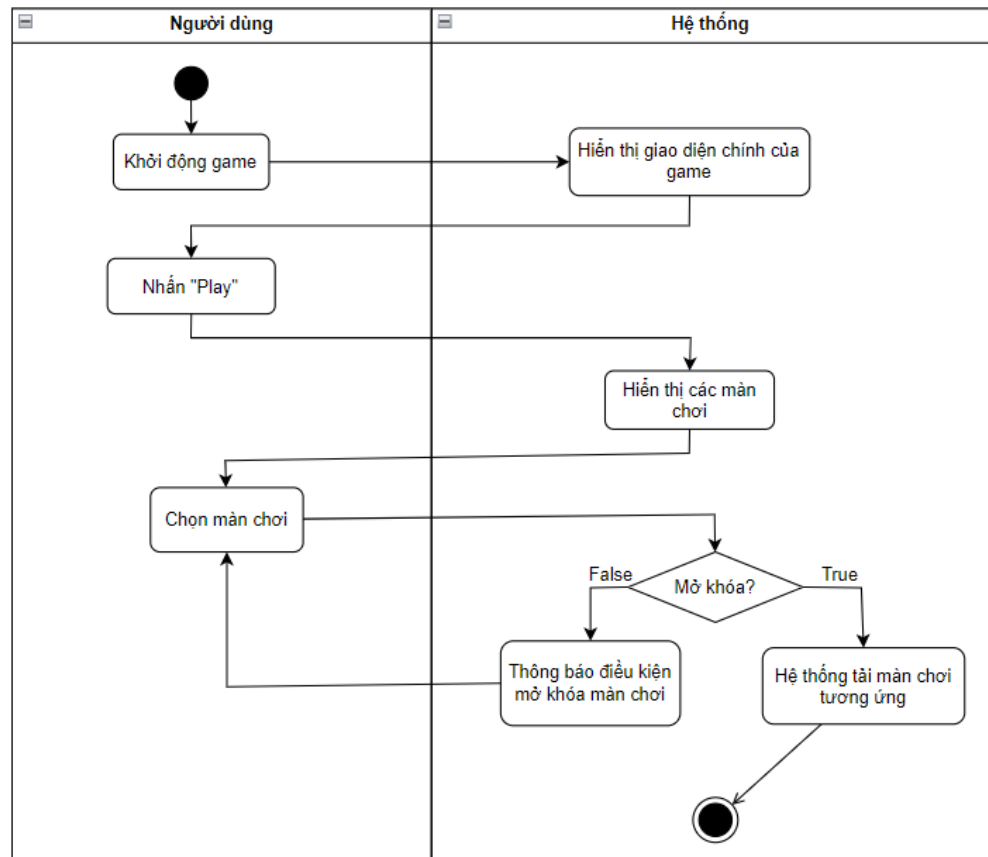
4.4.1 . Use-case “Play Game”

- Kịch bản Use-case

Bảng 4.1: Kịch bản Play Game

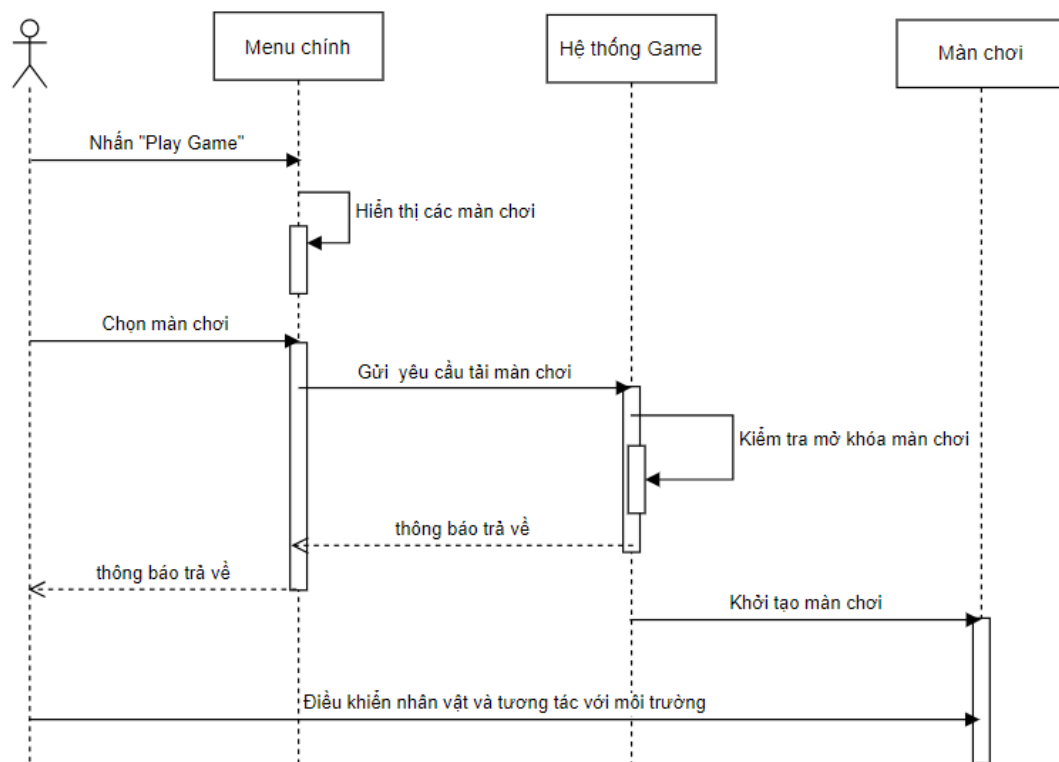
Mô tả	Bắt đầu chơi game
Tác nhân	Player
Tiền điều kiện	Người chơi đã khởi động game thành công và đã vào màn hình chính.
Hậu điều kiện	Người chơi đã vào được môi trường game chính, có thể di chuyển , tấn công kẻ địch ...
Luồng sự kiện	Người chơi khởi động game Hệ thống hiển thị giao diện chính của game Người chơi nhấn nút "Play" Màn hình hiển thị các màn chơi Người chơi chọn màn chơi Hệ thống tải màn chơi tương ứng Người chơi có thể điều khiển nhân vật và tương tác với môi trường
Luồng ngoại lệ	1a. Lỗi không thể khởi động game 4a. Người chơi chọn màn chơi chưa được mở khóa 4a1. Hệ thống thông báo điều kiện mở khóa của màn 6a. Không thể di chuyển nhân vật và tương tác với môi trường

- Sơ đồ hoạt động



Hình 4.2: Sơ đồ hoạt động Use Case Play game

- Sơ đồ tuần tự



Hình 4.3: Sơ đồ tuần tự Use Case Play game

4.4.2 . Use-case “Select Weapon”

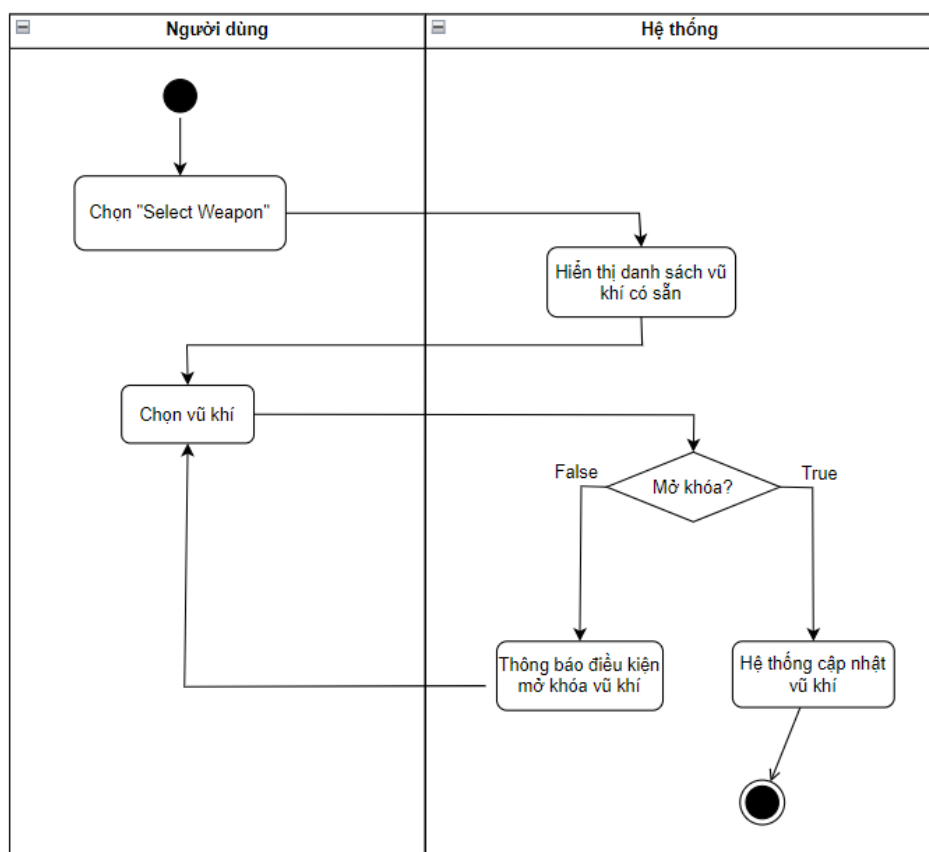
- Kịch bản Use-case

Bảng 4.2: Kịch bản Select Weapon

Mô tả	Chọn vũ khí đi kèm
Tác nhân	Player
Tiền điều kiện	Người chơi đã vào kho vũ khí Danh sách vũ khí có sẵn được hiển thị
Hậu điều kiện	Người chơi đã chọn một vũ khí đi kèm
Luồng sự kiện	Người chơi chọn “Select Weapon” ngoài giao diện Danh sách vũ khí có sẵn được hiển thị Người chơi chọn một vũ khí đi kèm

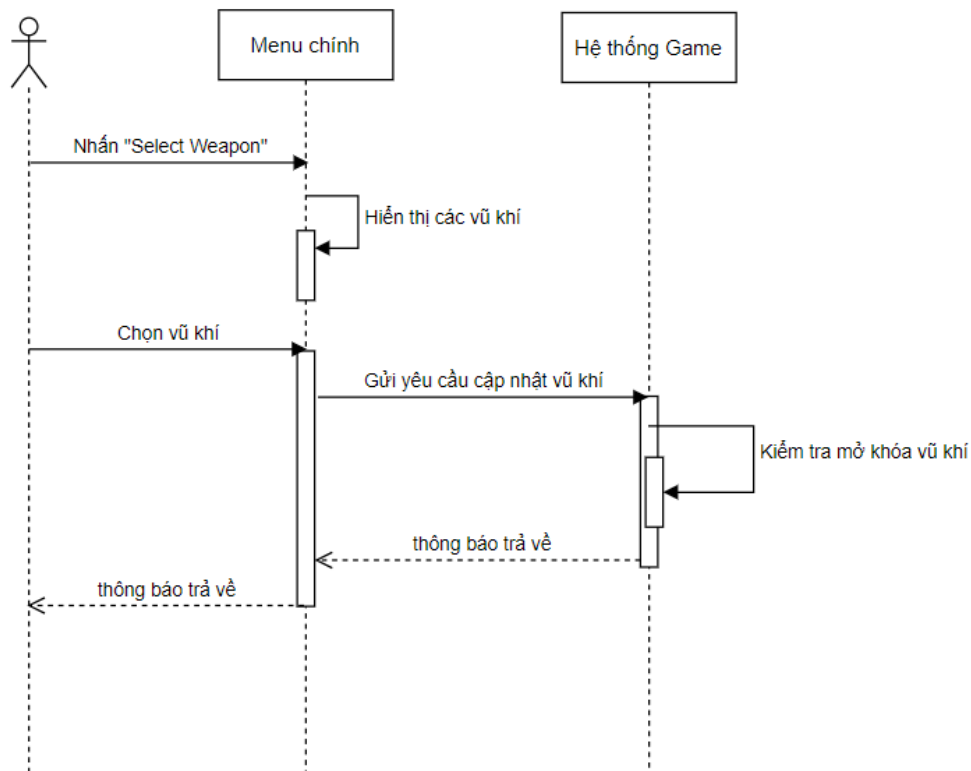
	Hệ thống xác nhận lựa chọn và cập nhật vũ khí cho nhân vật.
Luồng ngoại lệ	<p>2a. Lỗi không hiển thị danh sách vũ khí</p> <p>3a. Người chơi chọn vũ khí chưa được mở khóa</p> <p>3a1. Hệ thống thông báo điều kiện để mở khóa vũ khí</p>

- Sơ đồ hoạt động



Hình 4.4: Sơ đồ hoạt động Use Case Select Weapon

- Sơ đồ tuần tự



Hình 4.5: Sơ đồ tuần tự Use Case Select Weapon

4.4.3 . Use-case “Setting”

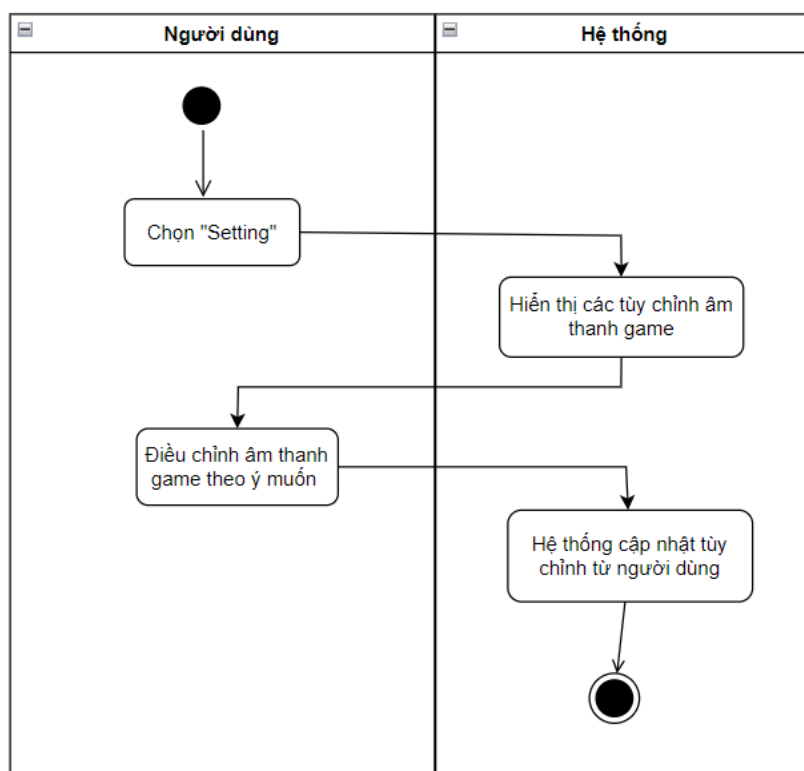
- Kịch bản Use-case

Bảng 4.4: Kịch bản Setting

Mô tả	Cài đặt âm thanh game
Tác nhân	Player
Tiền điều kiện	Người chơi đã vào môi trường game hoặc ngoài giao diện chính của game
Hậu điều kiện	Người chơi đã điều chỉnh các cài đặt âm thanh theo ý muốn
Luồng sự kiện	<p>Người chơi vào môi trường game hoặc ngoài giao diện chính của game</p> <p>Người chơi chọn “Setting” hoặc biểu tượng hình răng cưa</p>

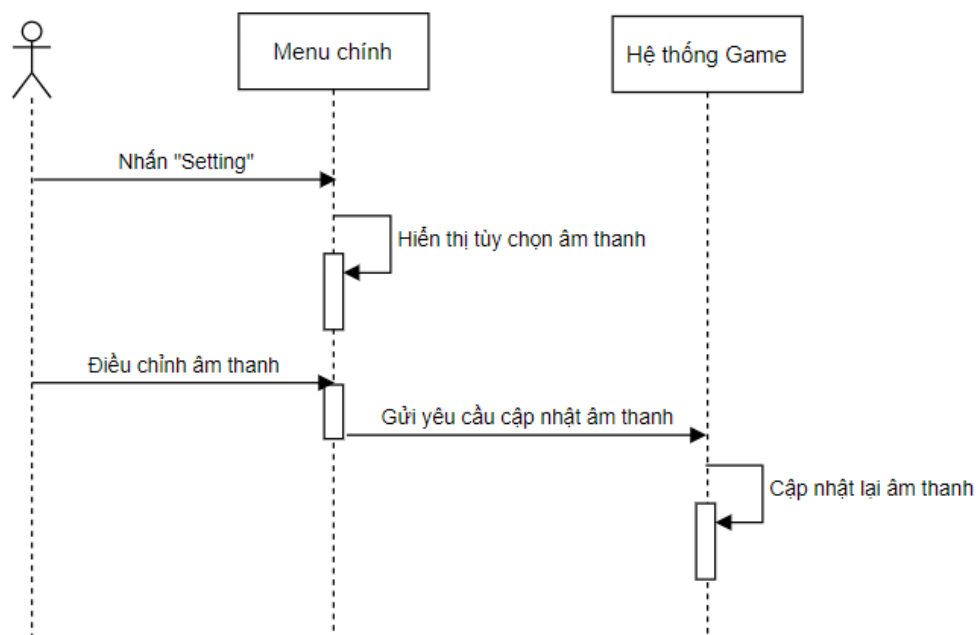
	<p>Hệ thống hiển thị các tùy chọn tùy chỉnh âm thanh game</p> <p>Người chơi điều chỉnh theo ý muốn</p> <p>Hệ thống lưu lại các cài đặt âm thanh mới và áp dụng chúng trong game.</p>
Luồng ngoại lệ	<p>5a.Lỗi tùy chọn âm thanh không áp dụng như điều chỉnh của người chơi</p>

- Sơ đồ hoạt động



Hình 4.6: Sơ đồ hoạt động Use Case Setting

- Sơ đồ tuần tự



Hình 4.7: Sơ đồ tuần tự Use Case Setting

4.4.4 . Use-case “Attack Enemy”

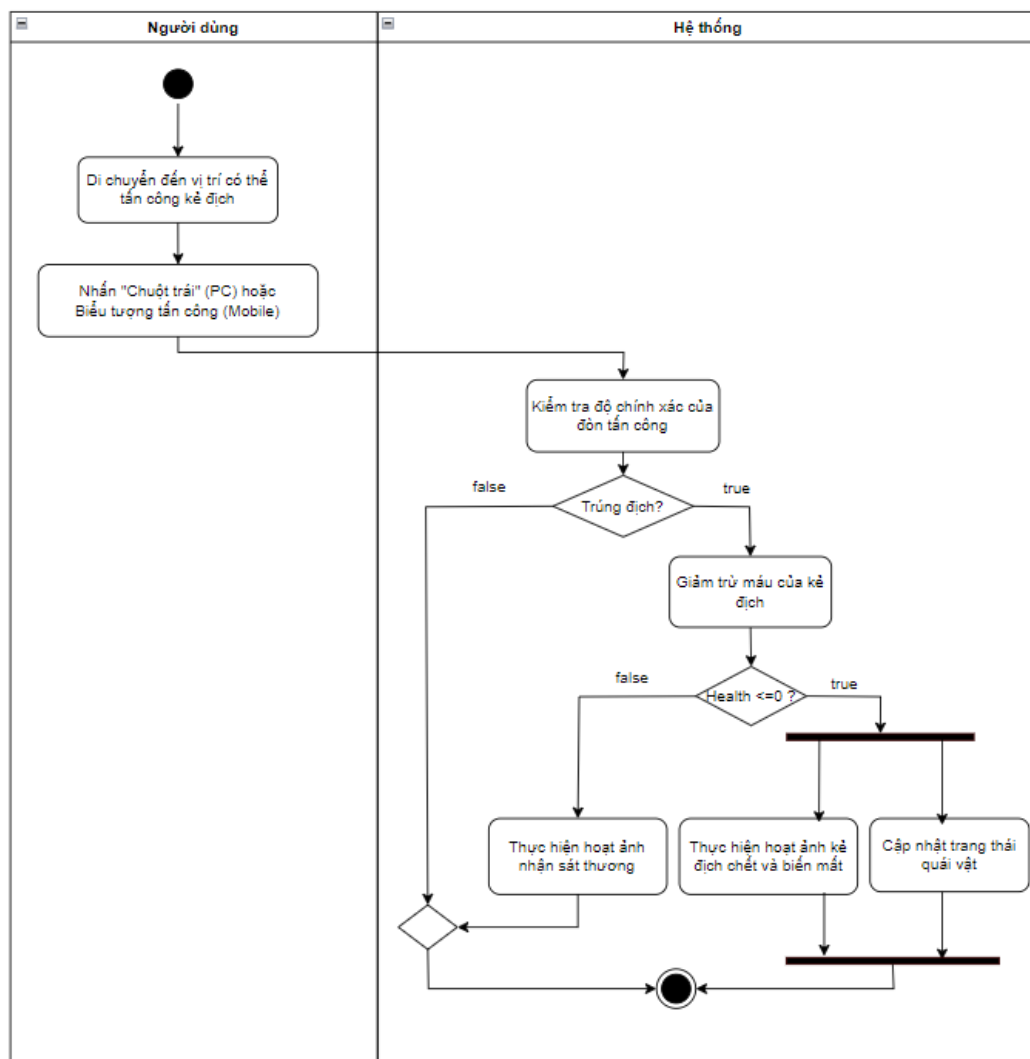
- Kịch bản Use-case

Bảng 4.5: Kịch bản Attack Enemy

Mô tả	Tấn công kẻ thù
Tác nhân	Player
Tiền điều kiện	Người chơi đã vào môi trường game chính và nhân vật của người chơi đang ở trong tình trạng có thể tấn công (không bị kẹt, không trong trạng thái vô hiệu, v.v.).
Hậu điều kiện	Kẻ địch bị tấn công và nhận sát thương (hoặc bị tiêu diệt).
Luồng sự kiện	<p>Người chơi di chuyển đến vị trí trong tầm tấn công</p> <p>Người chơi nhấn “chuột trái” đối với máy tính hoặc biểu tượng tấn công đối với điện thoại</p> <p>Hệ thống tính toán sát thương gây ra, giảm trừ lượng máu của kẻ địch</p>

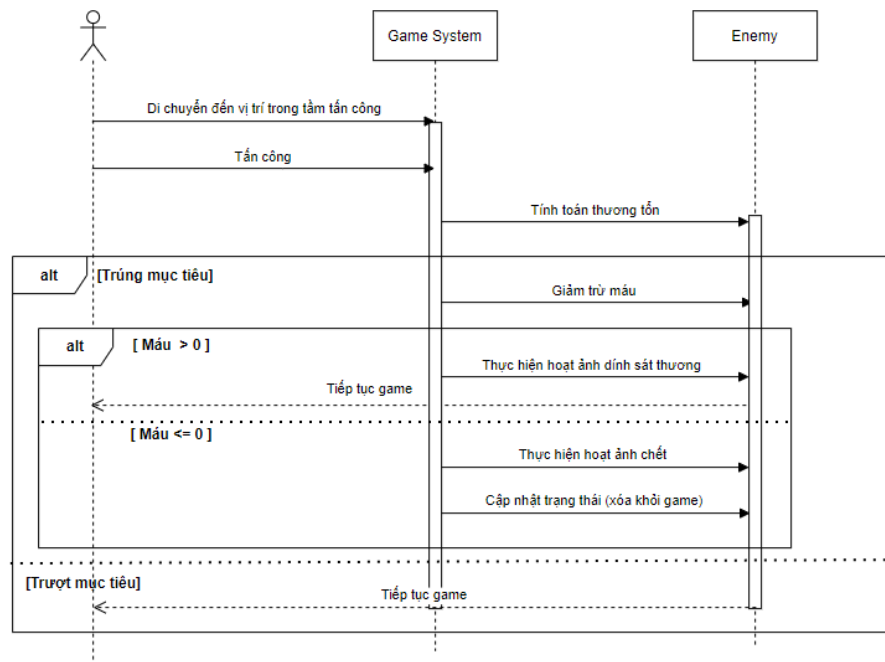
Luồng ngoại lệ	3a. Người chơi tấn công không trúng mục tiêu
	3a1. Kẻ địch không bị mất máu, hệ thống tiếp tục hoạt động bình thường
	3b. Kẻ địch hết máu (bị tiêu diệt)
	3b1. Hệ thống hiển thị hoạt ảnh bị tiêu diệt 3b2. Hệ thống cập nhật trạng thái kẻ địch (loại bỏ kẻ địch khỏi game)

- Sơ đồ hoạt động



Hình 4.8: Sơ đồ hoạt động Use Case Attack Enemy

- Sơ đồ tuần tự



Hình 4.9: Sơ đồ tuần tự Use Case Attack Enemy

4.4.5 . Use-case “Exit”

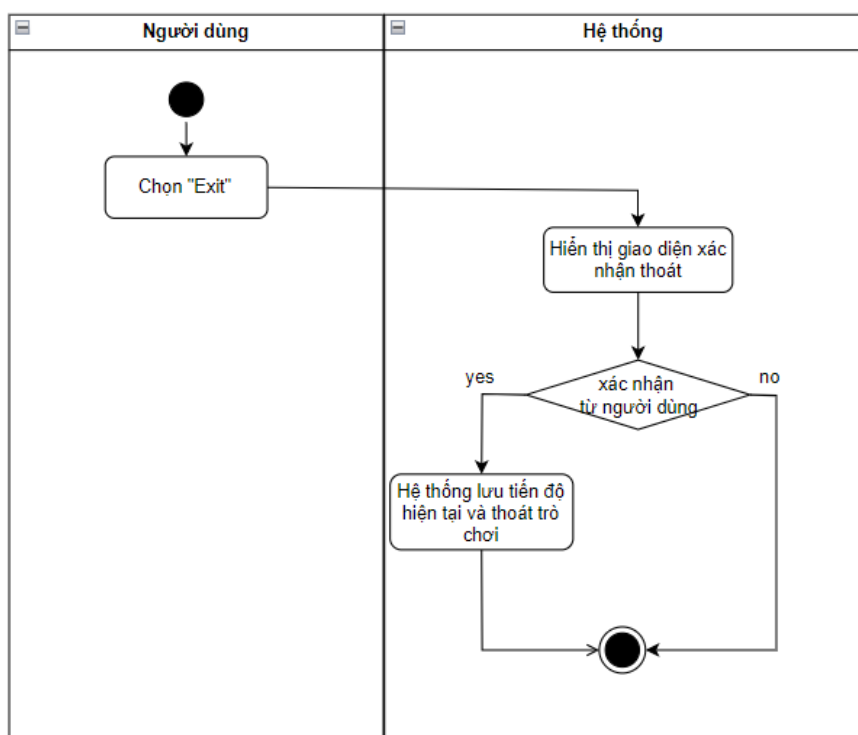
- Kịch bản Use-case

Bảng 4.6: Kịch bản Exit

Mô tả	Thoát game
Tác nhân	Player
Tiền điều kiện	Người chơi đã vào môi trường game hoặc ngoài giao diện chính của game
Hậu điều kiện	Game được thoát ra
Luồng sự kiện	<p>Người chơi nhấn nút “Exit” ngoài giao diện chính hoặc trong giao diện setting trong môi trường game</p> <p>Trong menu, người chơi chọn tùy chọn "Thoát game"</p>

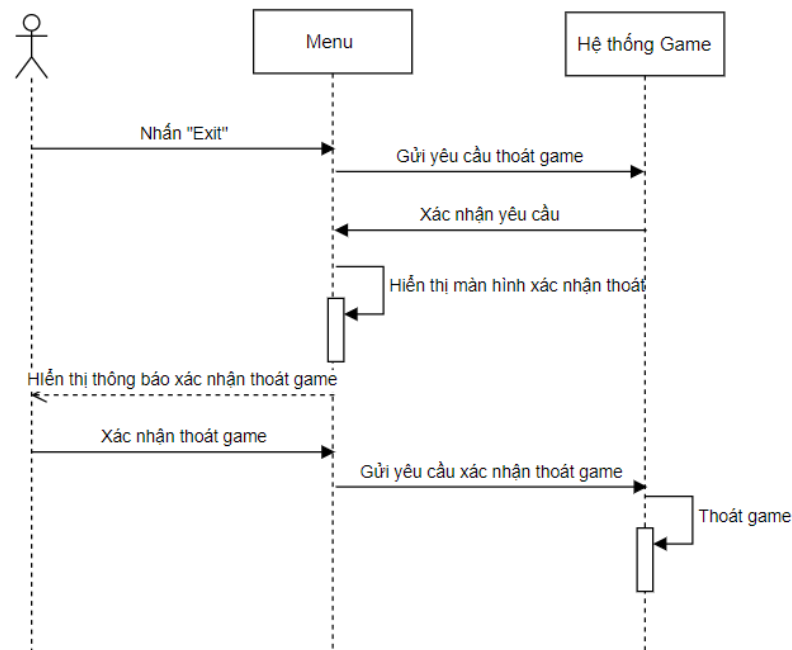
	<p>Hệ thống hiển thị thông báo xác nhận để đảm bảo người chơi thực sự muốn thoát game.</p> <p>Người chơi xác nhận lựa chọn.</p> <p>Game được đóng và hệ điều hành trở về màn hình chính.</p>
Luồng ngoại lệ	<p>2a. Người chơi chọn “Trở lại” thay vì thoát ra</p> <p>2a1. Game không đóng, hệ thống trở về giao diện setting</p>

- Sơ đồ Hoạt động



Hình 4.10: Sơ đồ hoạt động Use Case Exit

- Sơ đồ tuần tự



Hình 4.11: Sơ đồ hoạt động Use Case Exit

4.5 . Cơ sở dữ liệu

Trong game, cơ sở dữ liệu đóng vai trò quan trọng trong việc lưu trữ và quản lý thông tin về tiến trình của người chơi. Để đơn giản hóa việc lưu trữ và truy xuất dữ liệu, game sử dụng PlayerPrefs, một công cụ tích hợp sẵn trong Unity. PlayerPrefs cho phép lưu trữ các thông tin cần thiết một cách dễ dàng và hiệu quả.

PlayerPrefs truy xuất dữ liệu dưới dạng cặp giá trị khóa (key-value). Dữ liệu này được lưu trữ trên hệ thống của người chơi và có thể được truy cập lại khi người chơi quay trở lại trò chơi.

Trong game này, PlayerPrefs để lưu trữ level cao nhất mà người chơi đã vượt qua

```
PlayerPrefs.SetInt("MaxLevelCompleted", currentLevel);
```

Để truy xuất level đã hoàn thành :

```
int maxLevelCompleted = PlayerPrefs.GetInt("MaxLevelCompleted", 0);
```

PlayerPrefs để lưu trữ vũ khí mà người chơi đã mở khóa

```
PlayerPrefs.SetInt("WeaponUnlock",idWeapon);
```

CHƯƠNG 5 : XÂY DỰNG VÀ DEMO GAME

Link video demo game: [Google Drive](#)

Màn hình chính



Màn hình GamePlay



Màn hình chọn vũ khí



Màn hình dừng game



Màn hình Setting



CHƯƠNG 6 : KẾT LUẬN

6.1 . Kết quả đạt được của đề tài

- Trình bày được tổng quan về công nghệ Unity Engine
- Hiểu rõ được tác dụng của Animation.
- Hiểu rõ được cách làm game trên Unity.
- Xử lý được các lỗi cơ bản trong unity.
- Xử lý được âm thanh trong game.
- Xử lý được nhân vật trong game.
- Xử lý được các quái vật trong game.
- Xử lý được các vật phẩm trong game.
- Hoàn thành được game một game 3D action đơn giản.

6.2 . Hạn chế của đề tài

- Chưa xử lý tối ưu được code
- Còn nhiều tính năng chưa hoàn thiện
- Chưa đa dạng nhân vật và vũ khí trong game

6.3 . Hướng phát triển của đề tài

- Bổ sung các kỹ năng cho nhân vật
- Bổ sung thêm vật phẩm, vũ khí, nhân vật và quái trong game
- Cập nhật thêm màn chơi mới
- Tối ưu phần điều khiển cho điện thoại
- Nâng cấp AI của quái vật để xử lý linh hoạt hơn trong nhiều tình huống.

TÀI LIỆU THAM KHẢO

- [1] L. Chi, "Unity là gì? Hướng dẫn cách tải và cài đặt Unity 3D chi tiết, nhanh chóng nhất," 10 2023. [Online]. Available: <https://fptshop.com.vn/tin-tuc/thu-thuat/unity-la-gi-162181>.
- [2] T. X. Chu, "Giới thiệu về Unity Engine - Game Engine phổ biến nhất hiện nay.," 17 08 2018. [Online]. Available: https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay-V3m5WBj8lO7#_4-uu-diem-cua-unity-4.
- [3] DAISY, "Danh sách các khái niệm cơ bản trong Unity," 17 06 2022. [Trực tuyến]. Available: <https://tuhoclaptrinh.edu.vn/danh-sach-cac-khai-niem-co-ban-trong-unity-663.html>.
- [4] P. Le, "Awake, Start và một số hàm cơ bản trong MonoBehaviour," 29 07 2023. [Trực tuyến]. Available: <https://viblo.asia/p/awake-start-va-mot-so-ham-co-ban-trong-monobehaviour-qPoL77nNLvk>.
- [5] P. Le, "Tối ưu với Object Pooling Pattern trong Unity," 26 03 2020. [Trực tuyến]. Available: <https://viblo.asia/p/toi-uu-voi-object-pooling-pattern-trong-unity-EoW4oRyBVml>.
- [6] H. V. Tuan, "Kiểm thử Game Android," 27 03 2016. [Trực tuyến]. Available: <https://viblo.asia/p/kiem-thu-game-android-oZVRg4YxMmg5>.
- [7] yenmoc, "Design Pattern," 8 10 2022. [Trực tuyến]. Available: <https://www.pancakellc.com/tags/design-pattern/>.
- [8] N. Đ. A. Khoa, "Sử dụng Design Pattern mẫu Singleton trong Unity để tạo các đối tượng xây "một lần dùng nhiều lần"," 23 12 2023. [Trực tuyến]. Available: <https://caodang.fpt.edu.vn/tin-tuc-poly/tp-ho-chi-minh-tin-sinh-vien/su-dung-design-pattern-mau-singleton-trong-unity-de-cao-cac-doi-tuong-xay-mot-lan-dung-nhieu-lan.html>.
- [9] N. T. Phương, "Báo cáo đồ án tốt nghiệp XÂY DỰNG GAME BẮN SÚNG DƯỚI GÓC NHÌN THỨ 3 BẰNG UNITY," Đại học Thủy Lợi, 2023.
- [10] 24hComputer, "LỊCH SỬ PHÁT TRIỂN CỦA NGÀNH CÔNG NGHIỆP GAME," 16 06 2020. [Trực tuyến]. Available: <https://lapdatquangame.vn/tin-cong-nghe/lich-su-phat-trien-cua-nganh-cong-nghiep-game.html>.