

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



NGUYỄN KHUƠNG DUY

XÂY DỰNG GAME 2D “DUNGEONS AND WITCHERS”

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

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

NGUYỄN KHUƠNG DUY

XÂY DỰNG GAME 2D “DUNGEONS AND WITCHERS”

Ngành: Công nghệ thông tin

Mã số: _____

NGƯỜI HƯỚNG DẪN 1. ThS. Trương Xuân Nam

2. TS. Lê Nguyễn Tuấn Thành

HÀ NỘI, NĂM 2024

GÁY BÌA ĐỒ ÁN TỐT NGHIỆP

NGUYỄN KHUÔNG DUY

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: Nguyễn Khương Duy

Hệ đào tạo : Đại học chính quy

Lớp: 61TH1

Mã sinh viên: 1951060661

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1 TÊN ĐỀ TÀI:

XÂY DỰNG GAME 2D “DUNGEONS AND WITCHERS”

2 CÁC TÀI LIỆU CƠ BẢN:

- Quy trình giao đồ án tốt nghiệp khoa Công nghệ thông tin Đại học Thủy Lợi (2023)
- C#: <https://learn.microsoft.com/en-us/dotnet/csharp/>
- Unity: <https://docs.unity3d.com/Manual/UsingTheEditor.html>
- Assets: <https://itch.io/game-assets/free/tag-2d>
- Trợ giúp : <https://stackoverflow.com/>
- Lập trình Unity căn bản: Nhiều nguồn.

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

| Nội dung cần thuyết minh | Tỷ lệ |
|--|-------|
| 1. Chương I. Tổng quan về ngành game | 10% |
| 2. Chương II. Tổng quan về Unity | 20% |
| 3. Chương III. Mô tả thiết kế game | 25% |
| 4. Chương IV: Phân tích và thiết kế hệ thống | 20% |
| 5. Chương V: Xây dựng và demo | 23% |
| 6. Chương VI: Kết luận và hướng phát triển | 2% |

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

| Các phần | GVHD |
|--|--------------------------|
| 1. Chương I. Tổng quan về ngành game | ThS. Trương Xuân Nam |
| 2. Chương II. Tổng quan về Unity | ThS. Trương Xuân Nam |
| 3. Chương III. Mô tả thiết kế game | TS. Lê Nguyễn Tuấn Thành |
| 4. Chương IV: Phân tích và thiết kế hệ thống | TS. Lê Nguyễn Tuấn Thành |
| 5. Chương V: Xây dựng và demo | ThS. Trương Xuân Nam |
| 6. Chương VI: Kết luận | ThS. Trương Xuân Nam |

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngàytháng năm 202..

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 ... tháng ... năm 202...

Sinh viên làm Đồ án tốt nghiệp
(Ký và ghi rõ Họ tên)

Nguyễn Khương Duy



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 2D “Dungeons And Witchers”**

Sinh viên thực hiện: Nguyễn Khương Duy

Lớp: 61TH1

Mã sinh viên: 1951060661

Số điện thoại: 0355321023

Email: duytlu2001@gmail.com

Giáo viên hướng dẫn: ThS. Trương Xuân Nam

TS. Lê Nguyễn Tuấn Thành

Thời gian thực hiện: 14 tuần (từ ngày: 16/10/2023 đến ngày: 21/01/2024)

TÓM TẮT ĐỀ TÀI

"Dungeons And Witchers" là một trò chơi dựa trên cơ chế của “The Binding of Isaac” và “Noita”, tập trung vào cuộc phiêu lưu hấp dẫn của một phù thủy trong một thế giới dưới lòng đất đầy nguy hiểm. Trong game, người chơi sẽ vào vai một phù thủy và bắt đầu hành trình khám phá hầm ngục bí ẩn.

Nhiệm vụ chính của người chơi là đối mặt với những thách thức khó khăn, đánh quái vật và tiêu diệt các boss mạnh mẽ để thu thập những cây đũa phép có sức mạnh đặc biệt. Mỗi cây đũa phép sẽ mang theo một loạt các kỹ năng phép thuật độc đáo, giúp người chơi thích nghi với các tình huống khác nhau. Giữa các tầng của hầm ngục sẽ có những căn phòng trao đổi giúp người chơi nâng cấp những cây đũa phép của mình và tạo ra những phép mới lạ.

Trong suốt cuộc hành trình, người chơi sẽ phải tìm kiếm và sử dụng thông minh các vật phẩm, thuốc phép, và trang bị để cải thiện sức mạnh của họ và tăng khả năng sinh tồn. Các hầm ngục sẽ luôn thay đổi, đảm bảo sự đa dạng và tính chất ngẫu nhiên trong mỗi

lần chơi. "Dungeons And Witchers" hứa hẹn mang đến cho người chơi trải nghiệm thú vị với đồ họa độc đáo, lối chơi đa dạng và một thế giới đầy bí ẩn và kỳ quái để khám phá.

CÁC MỤC TIÊU CHÍNH

Chức năng :

1. Tìm hiểu về ngôn ngữ C#, cách sử dụng unity trong làm game.
2. Xây dựng nhân vật, tìm hiểu về cơ chế điều khiển nhân vật.
3. Xây dựng map, Tìm hiểu về các thuật toán tạo ra hàm ngẫu nhiên.
4. Xây dựng quái, tìm hiểu cơ chế của quái.
5. Xây dựng kho vật phẩm, tỉ lệ rơi vật phẩm, cấp bậc vật phẩm.
6. Xây dựng các loại phép thuật, cơ chế kết hợp phép thuật.
7. Xây dựng môi trường, cách mà các đối tượng tương tác với môi trường.
8. Thiết kế UI.

Công nghệ sử dụng :

- • C#
- • Unity, Visual Studio

KẾT QUẢ DỰ KIẾN

Sản phẩm:

- Trò chơi hoàn thiện, đáp ứng được các mục tiêu đề ra.

Kiến thức:

- Thành thạo C#.
- Kiến thức về công cụ unity.
- Kiến thức về lập trình game, quy trình để tạo nên game 2D.
- Kiến thức về phân tích và thiết kế.

Về kỹ năng:

- Sử dụng tốt công cụ unity.

- Khả năng thiết kế giao diện UI, thiết kế nhân vật.
- Nâng cao khả năng code, tư duy logic.
- Kỹ năng làm việc, tìm kiếm thông tin.

TIẾN ĐỘ THỰC HIỆN

| T T | Thời gian | Nội dung công việc | Kết quả dự kiến đạt được |
|----------------|-------------------------|---|---|
| 1 | 16/10/2023 – 05/11/2023 | Đọc tài liệu, phân tích thiết kế xây dựng lược đồ quan hệ | Phân tích thiết kế ra mô hình đầy đủ nhất của dự án |
| 2 | 06/11/2023 – 10/01/2024 | Lập trình xây dựng trò chơi | hoàn chỉnh trò chơi với các chức năng |
| 3 | 11/01/2024 – 16/01/2024 | Kiểm thử và fix lỗi hệ thống | đảm bảo hệ thống hoạt động bình thường |
| 4 | 17/01/2024 – 20/01/2024 | Hoàn thiện và chỉnh sửa báo cáo đồ án | Báo cáo đồ án hoàn chỉnh |



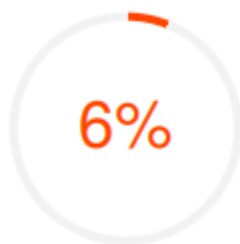
BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

| | |
|-------------------------|----------------------------------|
| Tên tài liệu: | 61TH1_1951060661_NguyenKhuongDuy |
| Tác giả: | Trương Xuân Nam |
| Điểm trùng lặp: | 6 |
| Thời gian tải lên: | 15:58 21/01/2024 |
| Thời gian sinh báo cáo: | 16:01 21/01/2024 |
| Các trang kiểm tra: | 63/63 trang |



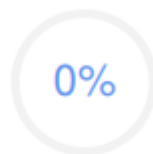
Kết quả kiểm tra trùng lặp



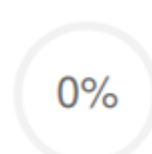
Có 6% nội dung trùng lặp



Có 94% nội dung không trùng lặp



Có 0% nội dung người dùng loại trừ



Có 0% nội dung hệ thống bỏ qua

Nguồn trùng lặp tiêu biểu

[123docz.net](#) [vi.wikipedia.org](#) [xemtailieu.net](#)

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN

Nguyễn Khương Duy

LỜI CẢM ƠN

Trong suốt thời gian học tập rèn luyện tại trường Đại học Thủy Lợi, em xin gửi lời cảm ơn đến tập thể các thầy cô trong trường nói chung và các thầy cô khoa Công Nghệ Thông Tin nói riêng. Nhờ sự dìu dắt tận tình của các thầy cô mà em mới có đủ khả năng, kiến thức để làm Đồ án tốt nghiệp lần này cũng như là kinh nghiệm để bước ra ngoài làm việc.

Em xin chân thành cảm ơn thầy giáo, Thạc sĩ Trương Xuân Nam đã hướng dẫn em hoàn thành tốt đồ án tốt nghiệp này trong thời gian vừa qua. Trong quá trình làm đồ án tốt nghiệp đã đưa cho em những lời khuyên, khuyến khích, góp ý tận tình để em hoàn thành đồ án của mình một cách tốt nhất.

Em xin chân thành cảm ơn thầy giáo, Tiến sĩ Lê Nguyễn Tuấn Thành đã cùng thầy Nam đồng hướng dẫn em hoàn thành đồ án tốt nghiệp này trong thời gian vừa qua. Thầy luôn nhiệt tình và quan tâm đến sinh viên, mỗi lần nhận được góp ý của thầy đều giúp em nhìn nhận lại lỗi sai mà em đã mắc phải trong quá trình phân tích và thiết kế trò chơi.

Em xin bày tỏ lòng biết ơn đến tất cả các bạn và thầy cô trường Đại học Thủy lợi đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường, em đã nhận được nhiều sự quan tâm, giúp đỡ, chia sẻ và động viên của thầy cô và bạn bè.

Do kiến thức còn hạn chế và thời gian hoàn thành không nhiều nên có thể còn thiếu sót, em rất mong nhận được sự góp ý của thầy cô và bạn bè để đồ án này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

| | |
|---|------|
| LỜI CAM ĐOAN | i |
| LỜI CẢM ƠN..... | ii |
| MỤC LỤC | iii |
| DANH MỤC HÌNH ẢNH..... | vi |
| DANH MỤC BẢNG BIỂU..... | viii |
| CHƯƠNG 1 TỔNG QUAN VỀ NGÀNH GAME | 1 |
| 1.1 Lịch sử và hướng phát triển..... | 1 |
| 1.2 Các công cụ làm game..... | 2 |
| 1.3 Các ngôn ngữ làm game | 4 |
| CHƯƠNG 2 TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY..... | 6 |
| 2.1 Unity là gì? | 6 |
| 2.2 Ưu điểm của Unity | 6 |
| 2.3 Các thành phần trong Unity..... | 7 |
| 2.3.1 Cửa sổ Scene | 7 |
| 2.3.2 Cửa sổ Game view..... | 7 |
| 2.3.3 Cửa sổ Hierarchy | 8 |
| 2.3.4 Cửa sổ Inspector | 9 |
| 2.3.5 Cửa sổ Project..... | 9 |
| 2.3.6 Cửa sổ Console..... | 10 |
| 2.3.7 Cửa sổ Toolbar | 11 |
| 2.3.8 Cửa sổ Animator..... | 11 |
| 2.4 Các khái niệm trong Unity | 12 |
| 2.4.1 GameObject | 12 |
| 2.4.2 Sprite..... | 12 |
| 2.4.3 Component | 12 |
| 2.4.4 Animation | 13 |
| 2.4.5 Prefab..... | 13 |
| 2.4.6 Script..... | 13 |
| 2.4.7 Assets..... | 13 |
| 2.4.8 Camera..... | 13 |

| | |
|------------------------------------|----|
| 2.4.9 Sound | 13 |
| 2.4.10 Unity Flow | 13 |
| 2.4.11 Finite State Machine:..... | 14 |
| 2.4.12 Nav Mesh..... | 15 |
| 2.5 Các lớp Unity API | 16 |
| 2.5.1 Animator Class | 16 |
| 2.5.2 AudioSource Class | 16 |
| 2.5.3 Camera Class | 16 |
| 2.5.4 Collider Class | 16 |
| 2.5.5 Component Class..... | 16 |
| 2.5.6 GameObject Class | 16 |
| 2.5.7 Input Class | 16 |
| 2.5.8 MonoBehaviour Class | 17 |
| 2.5.9 Physics Class | 17 |
| 2.5.10 Rigidbody Class..... | 17 |
| 2.5.11 ScriptableObject Class..... | 17 |
| 2.5.12 SceneManager Class..... | 17 |
| 2.5.13 SpriteRenderer Class | 17 |
| 2.5.14 Transform Class..... | 17 |
| 2.5.15 Time Class | 18 |
| CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME | 19 |
| 3.1 Ý tưởng | 19 |
| 3.2 Thiết kế game | 19 |
| 3.2.1 Thiết kế người chơi | 19 |
| 3.2.2 Thiết kế kẻ địch | 20 |
| 3.2.3 Thiết kế map | 22 |
| 3.2.4 Thiết kế vũ khí..... | 22 |
| 3.2.5 Thiết kế ngọc phép | 23 |
| 3.3 Các quy luật chơi..... | 23 |
| 3.3.1 Di chuyển..... | 23 |
| 3.3.2 Tấn công | 23 |

| | |
|--|----|
| 3.3.3 Thu thập..... | 23 |
| 3.3.4 Thanh trạng thái..... | 23 |
| CHƯƠNG 4 PHÂN TÍCH THIẾT KẾ HỆ THỐNG | 24 |
| 4.1 Xác định các tác nhân hệ thống..... | 24 |
| 4.2 Xây dựng biểu đồ Usecase | 24 |
| 4.2.1 Mô hình Usecase tổng quát | 24 |
| 4.3 Đặc tả và phân tích chức năng của hệ thống | 25 |
| 4.3.1 Usecase Chuyển đổi trạng thái | 25 |
| 4.3.2 Usecase Tương tác với kẻ địch..... | 27 |
| 4.3.3 Usecase Lựa chọn chúc phúc | 29 |
| 4.3.4 Usecase Quản lý vật phẩm | 31 |
| 4.4 Biểu đồ ER..... | 33 |
| 4.5 Xác định các lớp..... | 34 |
| 4.6 Xây dựng dữ liệu trò chơi..... | 34 |
| CHƯƠNG 5 XÂY DỰNG GAME VÀ DEMO | 36 |
| 5.1 Xây dựng game..... | 36 |
| 5.1.1 Xây dựng nhân vật..... | 36 |
| 5.1.2 Xây dựng quái vật | 36 |
| 5.1.3 Xây dựng vật phẩm: | 37 |
| 5.1.4 Xây dựng Map | 38 |
| 5.2 Hình ảnh demo..... | 39 |
| CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 43 |
| 6.1 Kết luận..... | 43 |
| 6.2 Hướng phát triển..... | 43 |
| TÀI LIỆU THAM KHẢO | 44 |

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 1. Logo Unity..... | 3 |
| Hình 2. Logo Unreal Engine | 3 |
| Hình 3. Logo Godot..... | 4 |
| Hình 4. Logo Unity..... | 6 |
| Hình 5. Cửa sổ Scene | 7 |
| Hình 6. Cửa sổ Game view..... | 8 |
| Hình 7. Cửa sổ Hierarchy | 8 |
| Hình 8. Cửa sổ Inspector | 9 |
| Hình 9. Cửa sổ Project..... | 10 |
| Hình 10. Cửa sổ Console..... | 10 |
| Hình 11. Cửa sổ Toolbar | 11 |
| Hình 12. Cửa sổ Animation..... | 12 |
| Hình 13. Thứ tự thực thi trong Unity | 14 |
| Hình 14. Finite State Machine | 15 |
| Hình 15. Ứng dụng navmesh trong trò chơi..... | 15 |
| Hình 16. Sprite người chơi | 20 |
| Hình 17. Sprite quái vật..... | 21 |
| Hình 18. Sprite quái vật..... | 21 |
| Hình 19. Sprite map..... | 22 |
| Hình 20. Sprite vũ khí | 22 |
| Hình 21. Sprite ngọc phép | 23 |
| Hình 22. Mô hình Usecase tổng quát | 24 |
| Hình 23. Biểu đồ luồng hoạt động Usecase Chuyển đổi trạng thái..... | 26 |
| Hình 24. Biểu đồ trình tự Usecase Chuyển đổi trạng thái..... | 26 |
| Hình 25. Biểu đồ luồng hoạt động Usecase Tương tác với kẻ địch | 28 |
| Hình 26. Biểu đồ luồng hoạt động Usecase Tương tác với kẻ địch | 28 |
| Hình 27. Biểu đồ luồng hoạt động Usecase Lựa chọn chúc phúc | 30 |
| Hình 28. Biểu đồ trình tự Usecase Lựa chọn chúc phúc | 30 |
| Hình 29. Biểu đồ luồng hoạt động Usecase Quản lý vật phẩm..... | 32 |
| Hình 30. Biểu đồ luồng hoạt động Usecase Quản lý vật phẩm..... | 33 |
| Hình 31. Biểu đồ ER | 33 |
| Hình 32. Ví dụ về việc xây dựng ScriptableObject..... | 35 |
| Hình 33. Ví dụ về việc Triển khai ScriptableObject | 35 |
| Hình 34. Xây dựng player | 36 |
| Hình 35. Xây dựng quái vật | 37 |
| Hình 36. Xây dựng vật phẩm | 38 |
| Hình 37. Xây dựng map | 38 |
| Hình 38. Giao diện khởi động game..... | 39 |

| | |
|---|----|
| Hình 39. Giao diện cài đặt tùy chọn | 39 |
| Hình 40. Giao diện quản lý vật phẩm..... | 40 |
| Hình 41. Tấn công quái vật | 41 |
| Hình 42. Rương thưởng và phân thưởng..... | 41 |
| Hình 43. Giao diện game over..... | 42 |

DANH MỤC BẢNG BIỂU

| | |
|--|----|
| Bảng 1. Xác định các tác nhân hệ thống | 24 |
| Bảng 2. Đặc tả Usecase Chuyển đổi trạng thái | 25 |
| Bảng 3. Đặc tả Usecase Tương tác với kẻ địch..... | 27 |
| Bảng 4. Đặc tả Usecase Lựa chọn chức phức..... | 29 |
| Bảng 5. Đặc tả Usecase Quản lý vật phẩm..... | 31 |
| Bảng 6. Một số lớp có trong trò chơi..... | 34 |

CHƯƠNG 1 TỔNG QUAN VỀ NGÀNH GAME

1.1 Lịch sử và hướng phát triển

Ngành công nghiệp game đã trải qua một hành trình dài và đang phát triển đáng kể trong suốt hơn nửa thế kỷ qua. Sau đây là một số sự kiện quan trọng trong lịch sử ngành game [1]:

- Năm 1947: Thomas T. Goldsmith Jr. và Estle Ray Mann phát minh ra một máy chơi game đầu tiên gọi là “Cathode Ray Tube Amusement Device”, có thể coi là bước đầu tiên trong lịch sử game [1].
- Năm 1958: Physicist William Higinbotham tạo ra trò chơi tennis đầu tiên trên máy tính tại Brookhaven National Laboratory [1].
- Năm 1972: Nolan Bushnell và Ted Dabney thành lập công ty Atari và phát hành trò chơi đầu tiên của họ là Pong, một trong những trò chơi arcade đầu tiên [1].
- Năm 1980: Nintendo phát hành game đầu tiên của mình, Donkey Kong, và đưa ra thị trường máy console đầu tiên của họ, Nintendo Entertainment System (NES) [1].
- Năm 1990: Sony phát hành PlayStation, máy console đầu tiên của họ và một trong những máy console phổ biến nhất của thập niên 1990 [1].
- Năm 1993: id Software phát hành game Doom, đưa thể loại game bắn súng trở nên phổ biến hơn bao giờ hết [1].
- Năm 1994: The Entertainment Software Rating Board (ESRB) được thành lập để đánh giá nội dung của các trò chơi video [1].
- Năm 2001: Microsoft phát hành máy console Xbox đầu tiên của họ [1].
- Năm 2004: Blizzard Entertainment phát hành game trực tuyến đa nền tảng đầu tiên của họ, World of Warcraft, và trở thành một trong những game trực tuyến phổ biến nhất trong lịch sử [1].
- Năm 2007: Apple phát hành iPhone, mở ra một thị trường mới cho game di động [1].
- Năm 2017: Nintendo phát hành máy console di động Nintendo Switch, được coi là một trong những thiết bị chơi game phổ biến nhất của thế hệ mới [1].

Từ đó 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 [1].

Dự đoán xu hướng ngành game đến năm 2030 là khá khó khăn, tuy nhiên có một số xu hướng chung có thể xảy ra trong tương lai gần [1]:

- Chơi game trực tuyến sẽ tiếp tục tăng: Với sự phát triển của mạng internet và công nghệ kết nối, chơi game trực tuyến sẽ tiếp tục tăng. Các trò chơi đa người chơi trực tuyến và các game có tính tương tác cao sẽ trở nên phổ biến hơn [1].
- Game di động sẽ tiếp tục phát triển: Đã có xu hướng tăng trưởng trong việc chơi game trên điện thoại di động, và điều này sẽ tiếp tục phát triển trong tương lai. Các game di động sẽ ngày càng được phát triển để đáp ứng nhu cầu của người chơi trên toàn cầu [1].
- Các trò chơi sử dụng công nghệ VR và AR sẽ trở nên phổ biến: Công nghệ thực tế ảo (VR) và thực tế tăng cường (AR) đang được sử dụng trong các trò chơi và có thể trở thành xu hướng chính trong ngành game trong tương lai [1].
- Sự phát triển của game trên đám mây: Với sự phát triển của công nghệ đám mây, các game có thể được truy cập trên nhiều thiết bị và người chơi có thể chơi các trò chơi yêu thích của mình mọi lúc, mọi nơi [1].
- Game chạy trên blockchain và sử dụng NFT sẽ phát triển: Game chạy trên blockchain và sử dụng NFT đang trở nên phổ biến hơn, và điều này có thể tiếp tục phát triển trong tương lai khi người chơi muốn sở hữu các tài sản trong game một cách độc đáo và minh bạch [1].
- Sự xuất hiện của game AI: Sự phát triển của trí tuệ nhân tạo (AI) sẽ mở ra cơ hội để phát triển các game có tính tự động cao hơn, hỗ trợ cho việc tạo ra các nhân vật và tạo nên các nội dung trong game [1].

1.2 Các công cụ làm game

Các công cụ làm game là các phần mềm hỗ trợ cho nhà phát triển game dễ dàng hơn trong quá trình thiết kế và phát triển game. Một số lợi ích có thể dễ nhận thấy khi sử dụng các công cụ làm game so với làm game thuần như: dễ học, dễ sử dụng, tiết kiệm thời gian phát triển, dễ dàng nâng cấp và bảo trì.

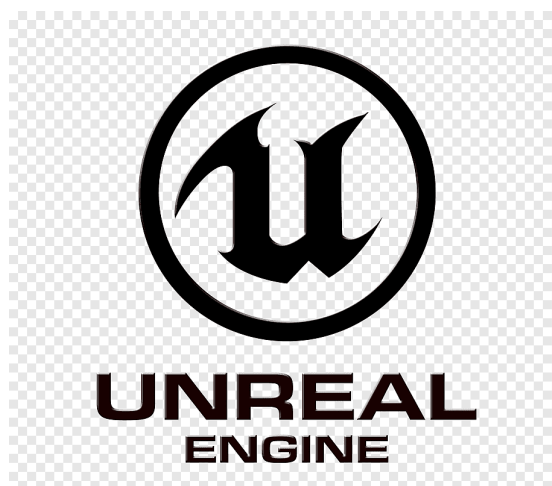
Một số công cụ phát triển game phổ biến hiện nay:

- Unity: Đây là một công cụ đa nền tảng được phát triển bởi Unity Technologies, cho phép tạo ra các game 2D và 3D cho nhiều thiết bị khác nhau như máy tính, điện thoại, máy chơi game.... Lần đầu tiên unity được công bố là được chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay unity đã mở rộng hơn 27 nền tảng [2]. Unity hỗ trợ lập trình 2D và 3D, hiện nay ngôn ngữ thông dụng được sử dụng trong unity là C#.



Hình 1. Logo Unity

- Unreal Engine: Là một công cụ đa nền tảng chủ yếu để phát triển game 3D có đồ họa nặng, các nền tảng chủ yếu là máy tính, điện thoại, máy chơi game. Được viết bằng C++, Unreal Engine có tính di động cao, hỗ trợ nhiều loại nền tảng [3].



Hình 2. Logo Unreal Engine

- Godot: là một công cụ trò chơi đa nền tảng, miễn phí và mã nguồn mở được phát hành theo giấy phép MIT cho phép. Ban đầu nó được phát triển bởi các nhà phát triển phần mềm người Argentina Juan Linietsky và Ariel Manzur cho một số công ty ở Mỹ Latinh trước khi phát hành công khai vào năm 2014. Môi trường phát triển chạy trên nhiều nền tảng và có thể xuất sang nhiều nền tảng khác. Nó được thiết kế để tạo ra cả trò chơi 2D và 3D nhắm mục tiêu đến PC, thiết bị di động và nền tảng web và cũng có thể được sử dụng để phát triển phần mềm không phải trò chơi, bao gồm cả trình chỉnh sửa [4].



Hình 3. Logo Godot

1.3 Các ngôn ngữ làm game

Các ngôn ngữ làm game là những ngôn ngữ lập trình được sử dụng để viết mã nguồn cho game. Ngôn ngữ làm game phải có các yếu tố sau:

- Dễ học và sử dụng: Các ngôn ngữ làm game thường có cú pháp đơn giản và rõ ràng, giúp cho người lập trình dễ hiểu và viết mã nguồn.
- Nhanh, hiệu quả và nhẹ: Các ngôn ngữ làm game thường có khả năng biên dịch hoặc thông dịch nhanh chóng không thể được biên dịch chậm vì đảm bảo trải nghiệm người dùng tốt nhất theo thời gian thực.
- Linh hoạt và dễ mở rộng: Dựa vào khả năng phát triển của các thiết bị chơi game, các ngôn ngữ lập trình cũng phải có khả năng tương thích với nhiều công cụ, thiết bị.

Một số ngôn ngữ làm game phổ biến hiện nay bao gồm:

- C#: Là ngôn ngữ lập trình hướng đối tượng, được sử dụng trong công cụ Unity để viết mã nguồn cho game. C# có cú pháp giống với Java và C++ nên dễ làm quen, có khả năng hỗ trợ lập trình đa luồng và xử lý sự kiện.
- C++: Là ngôn ngữ lập trình có độ trừu tượng cao, cho phép bạn kiểm soát chính xác cách phần cứng của máy tính tương tác với đoạn code của bạn, nó được sử dụng trong công cụ Unreal. Cú pháp của C++ giống với C#, có khả năng lập trình cấp thấp và khả năng xử lý bộ nhớ, đồ họa tốt.
- GML: Là một ngôn ngữ lập trình thông dịch, được sử dụng trong công cụ GameMaker Studio để viết mã nguồn cho game. GML có cú pháp đơn giản và dễ học, có khả năng hỗ trợ các hàm và biến toàn cục và cục bộ.
- Python: Là một ngôn ngữ lập trình thông dịch, được sử dụng trong các công cụ làm game như Pygame, Blender. Python có cú pháp rõ ràng và sạch sẽ, có khả năng hỗ trợ lập trình hướng đối tượng và hàm. Các nguyên lý vận hành đơn giản của nó làm cho Python trở thành một lựa chọn phù hợp cho nhiều nhà phát triển trò chơi.

CHƯƠNG 2 TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY

2.1 Unity là gì?

Unity là một công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies từ những năm 2005, hiện nay nó đã dần trở thành một game engine phổ biến nhất trên thế giới. Có rất nhiều nền tảng hỗ trợ các game được lập trình bằng unity, đó là Android, IOS, Linux, macOS, Windows, Windows Phone,...

Unity cung cấp một hệ thống toàn diện cho các lập trình viên, từ lập trình mã nguồn, xây dựng hình ảnh, âm thanh,... là công cụ tự động hóa đến trình sửa lỗi nên cũng khá dễ sử dụng. Ngôn ngữ lập trình chính của Unity là C#.

Unity cũng có chức năng của các thư viện phần mềm làm game khác 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 4. Logo Unity

2.2 Ưu điểm của Unity

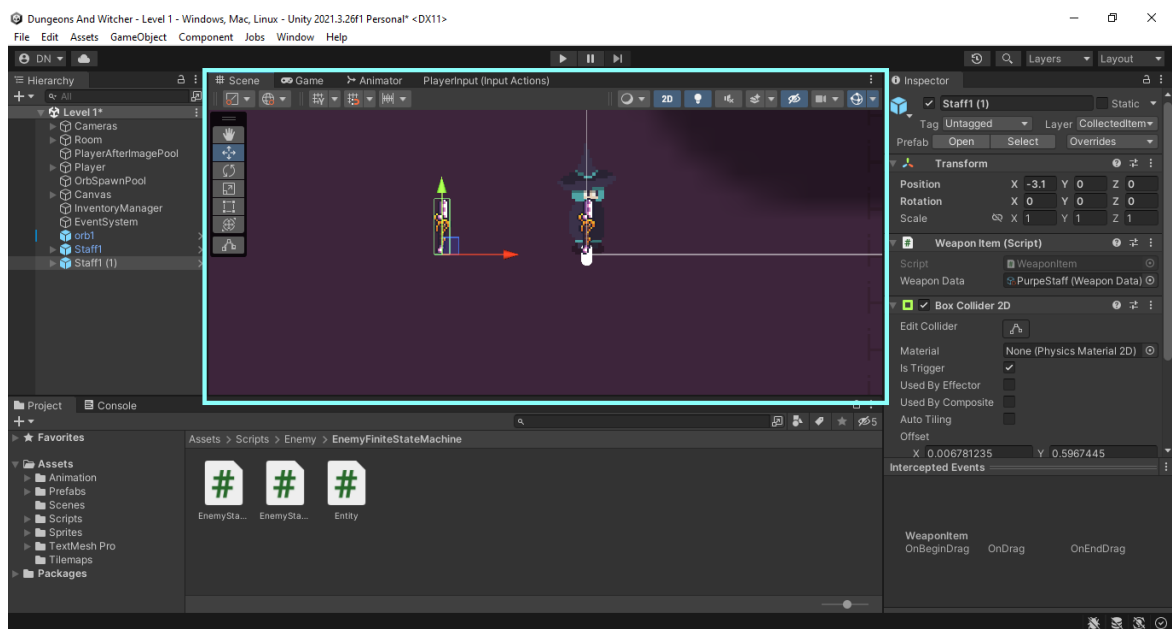
- Dễ học và sử dụng: Unity có một cộng đồng lớn và phong phú, cung cấp nhiều tài nguyên giáo trình và hỗ trợ, làm cho nó trở thành một lựa chọn lý tưởng cho cả người mới học lập trình và những nhà phát triển kinh nghiệm.
- Đồng nhất đa nền tảng: Unity cho phép phát triển một lần và triển khai trên nhiều nền tảng, giảm thiểu công sức cần thiết cho việc đa nền tảng hóa.

- Cộng đồng lớn và hỗ trợ: Cộng đồng Unity rất tích cực và hỗ trợ lẫn nhau thông qua diễn đàn, blog, video hướng dẫn, và các sự kiện cộng đồng.
- Thư viện tài nguyên phong phú: Unity Asset Store cung cấp một nguồn lực khổng lồ các tài nguyên, công cụ, và asset có sẵn để giúp nhà phát triển tăng tốc quá trình phát triển.

2.3 Các thành phần trong Unity

2.3.1 Cửa sổ Scene

Cửa sổ Scene là nơi chứa các GameObject, lập trình viên có thể thực hiện kéo thả các thành phần thay vì code giúp tiết kiệm thời gian. Một trò chơi thường có nhiều scene.

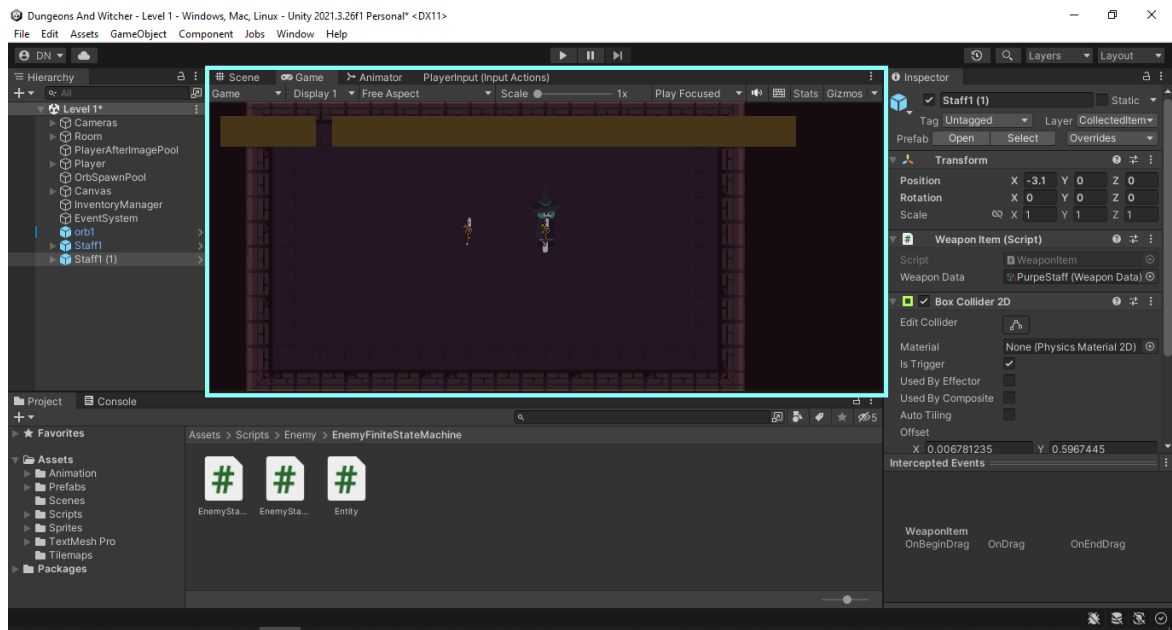


Hình 5. Cửa sổ Scene

2.3.2 Cửa sổ Game view

Game view là góc nhìn từ những Camera trong Game. Chúng ta sẽ cần ít nhất là một hoặc nhiều hơn số lượng các Camera để quyết định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game, mặc định sẽ là MainCamera.

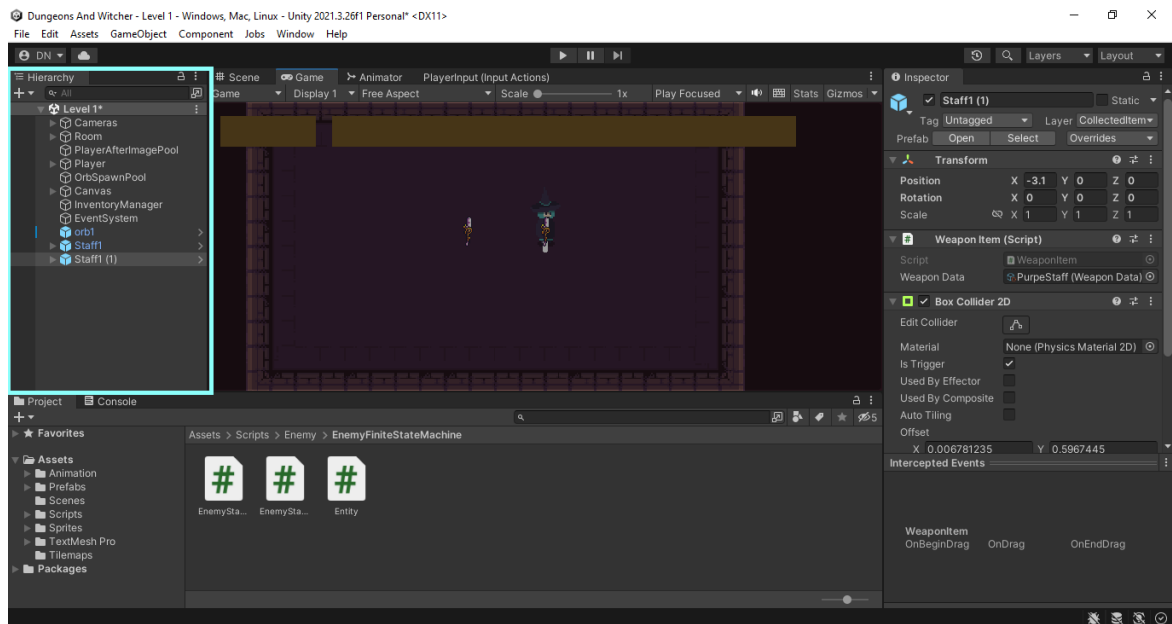
Cửa sổ Game view cho phép bạn xem trò chơi trong khi đang phát triển. Điều này giúp bạn kiểm tra và thử nghiệm trải nghiệm người chơi trực tiếp từ góc độ của người chơi.



Hình 6. Cửa sổ Game view

2.3.3 Cửa sổ Hierarchy

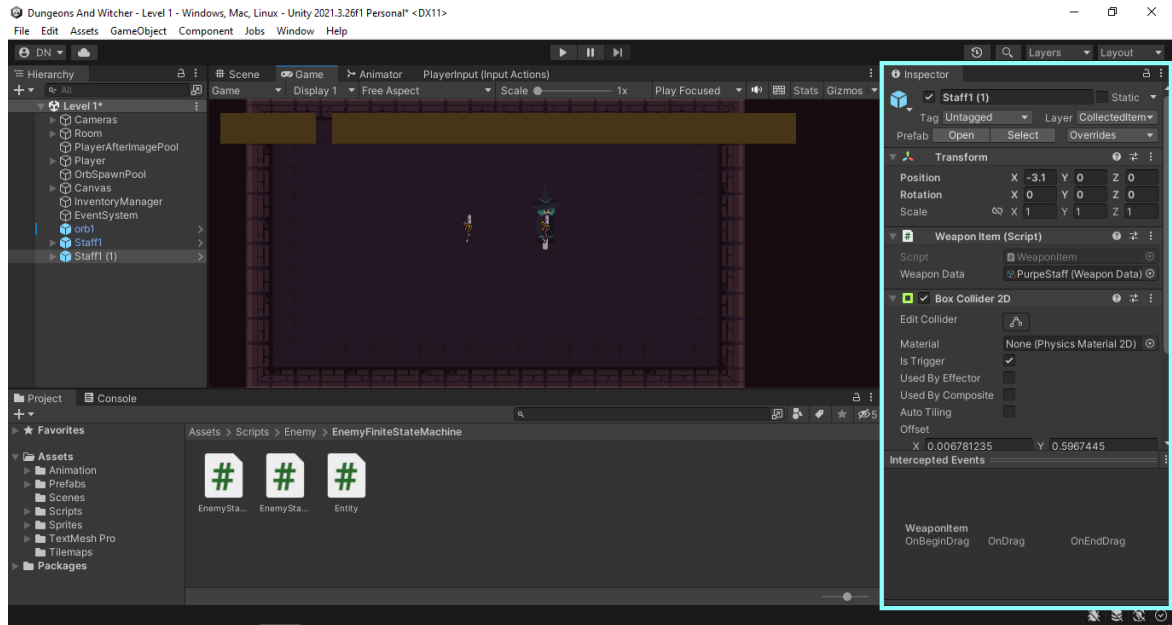
Cửa sổ Hierarchy liệt kê tất cả các đối tượng trong scene hiện thời. Đối tượng nào được hiển thị ở đây sẽ xuất hiện trong Sence. Quản lý các đối tượng bằng cách kéo và thả chúng, thay đổi thứ tự, và tạo các đối tượng (prefab).



Hình 7. Cửa sổ Hierarchy

2.3.4 Cửa sổ Inspector

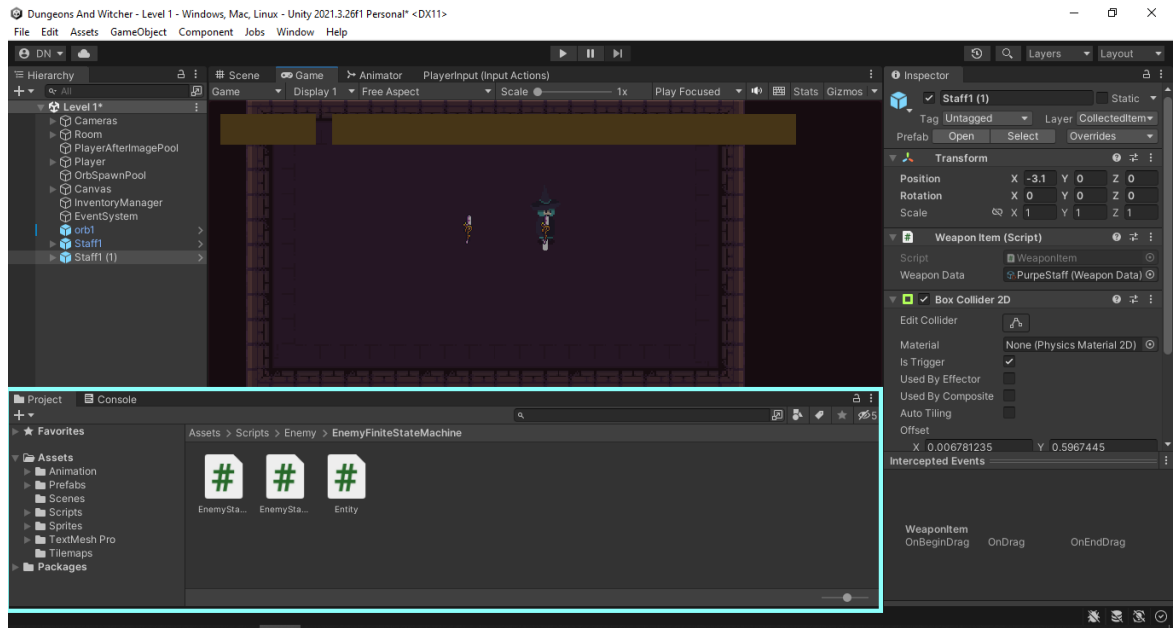
Cửa sổ Inspector hiển thị thông tin chi tiết và các thuộc tính của đối tượng được chọn trong Scene hoặc Hierarchy. Bạn có thể điều chỉnh các giá trị, thêm các thành phần, và tùy chỉnh các thuộc tính của đối tượng ở đây mà không cần thông qua code.



Hình 8. Cửa sổ Inspector

2.3.5 Cửa sổ Project

Cửa sổ Project hiển thị cấu trúc thư mục của dự án. Nó chứa tất cả các tài nguyên như hình ảnh, âm thanh, văn bản, mã nguồn và các prefab. Bạn có thể quản lý và tổ chức tài nguyên của mình từ đây. Mỗi Project của Unity đều chứa một thư mục Assets. Nó chứa tất cả các tài nguyên như hình ảnh, âm thanh, văn bản, mã nguồn và các prefab.

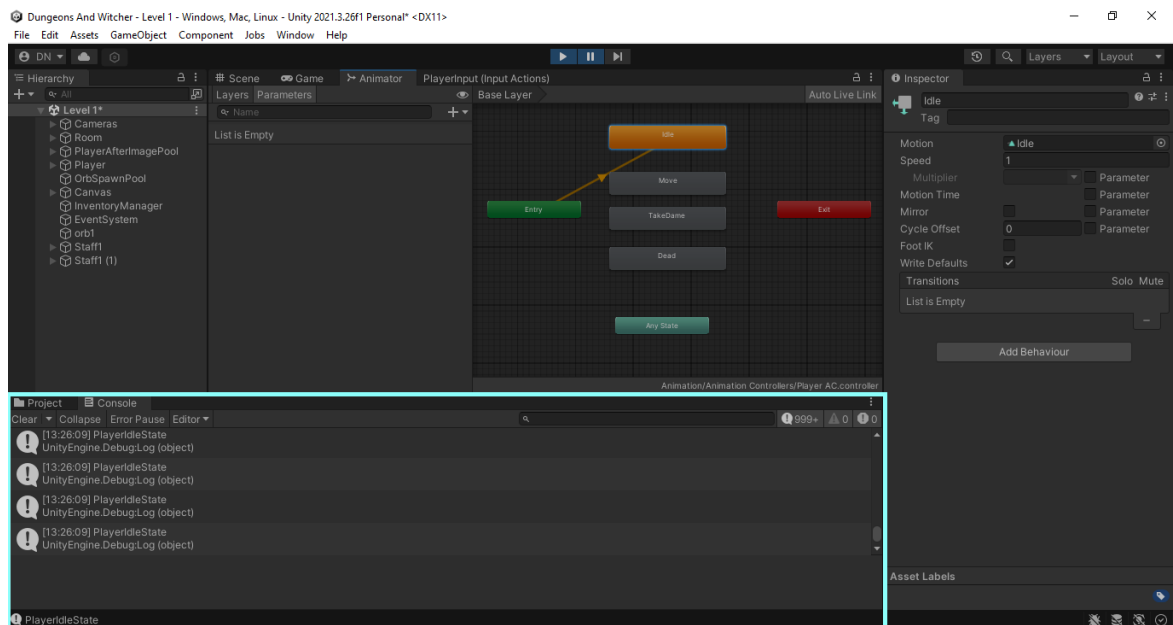


Hình 9. Cửa sổ Project

2.3.6 Cửa sổ Console

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

Ngoài ra ta có thể sử dụng Debug để hiện những test case cần thiết để xem code của mình hoạt động như nào.

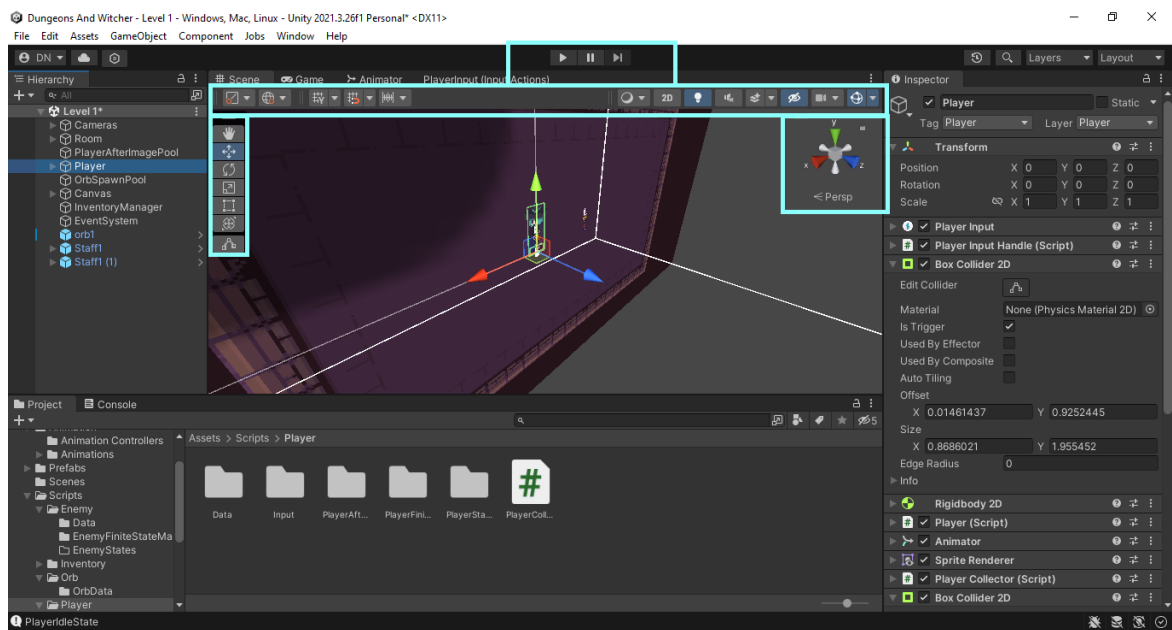


Hình 10. Cửa sổ Console

2.3.7 Cửa sổ Toolbar

Toolbar chứa các loại điều khiển cơ bản, mỗi loại giữ một vai trò quan trọng trong Editor.

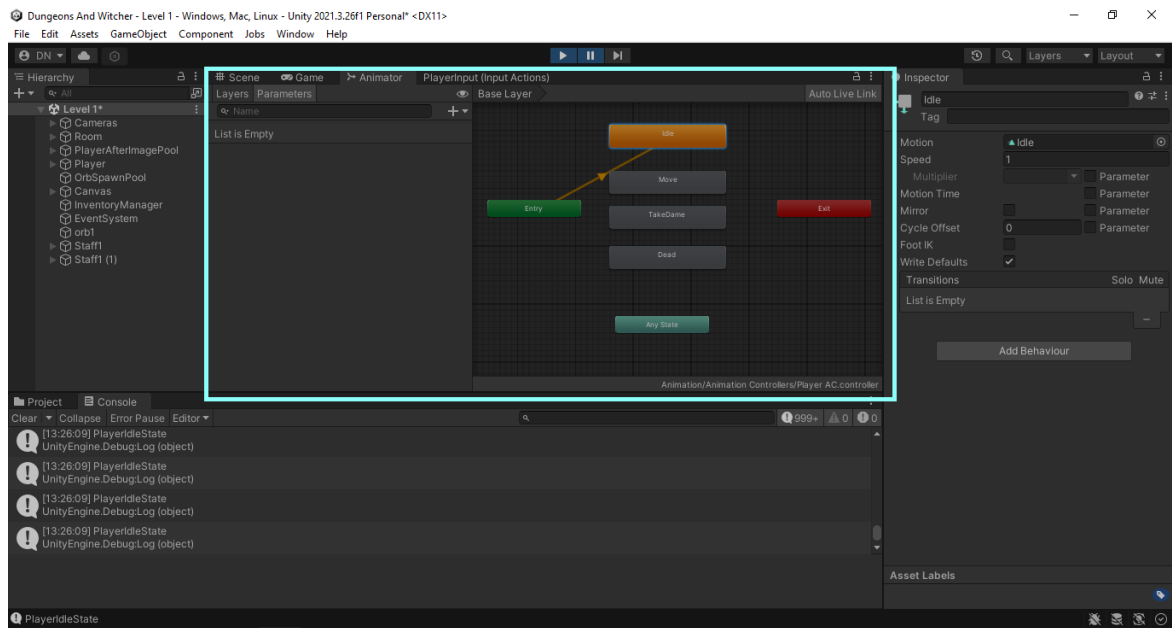
- Transform Tool: được dùng với Scene view, như quay trái, phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.
- View Options: Dùng cho việc thể hiện Scene view như 2D, 3D, audio, light,....
- Play/Pause/Step Buttons: dùng cho view game, chạy game ngay trong Editor để kiểm tra.



Hình 11. Cửa sổ Toolbar

2.3.8 Cửa sổ Animator

Cửa sổ Animator được sử dụng để tạo và quản lý các trạng thái và chuyển động của các đối tượng trong trò chơi. Bạn có thể tạo và chỉnh sửa các animation clips và điều khiển cách chúng được kích hoạt trong trò chơi.



Hình 12. Cửa sổ Animation

2.4 Các khái niệm trong Unity

Dưới đây là mô tả chi tiết về một số thành phần quan trọng có trong Unity như: GameObject, Component, Sprite, Animation, Prefab, Script, Assets, Camera, Sound, Transform, Material và Shader.

2.4.1 GameObject

GameObject là thành phần cơ bản trong Unity, đại diện cho mọi đối tượng trong cảnh (scene). Đối tượng này có thể là nhân vật, vật phẩm, ánh sáng,... GameObject không thể tồn tại một cách độc lập mà phải chứa các Component để có thể thực hiện các chức năng cụ thể.

2.4.2 Sprite

Sprite là một hình ảnh 2D được sử dụng để hiển thị trên màn hình. Trong Unity, Sprite thường được sử dụng cho các đối tượng như nhân vật, vật phẩm, và nền của trò chơi 2D.

2.4.3 Component

Component là các thành phần gắn liền với GameObject, dùng để cung cấp các chức năng và tính năng cho GameObject. Các Component bao gồm như Renderer, Collider, và Script,... Mỗi GameObject có thể có nhiều Component khác nhau để thực hiện các nhiệm vụ và hành vi cụ thể.

2.4.4 Animation

Animation là quá trình tạo ra chuyển động cho GameObject trong trò chơi. Bạn có thể tạo Animation bằng cách ghi lại các keyframe, xác định sự thay đổi trong thuộc tính Transform trong khoảng thời gian, bạn cũng có thể thêm các sự kiện kích hoạt ở các keyframe cụ thể.

2.4.5 Prefab

Prefab là một mô hình hoặc đối tượng có thể được tái sử dụng trong nhiều địa điểm khác nhau trong dự án. Prefab giúp tiết kiệm thời gian và duy trì tính thống nhất trong trò chơi.

2.4.6 Script

Script là các đoạn mã nguồn viết bằng ngôn ngữ lập trình như C# hoặc JavaScript, được gắn kết với GameObject để thực hiện các chức năng và tương tác trong trò chơi. Script là một phần quan trọng của việc lập trình trong Unity.

2.4.7 Assets

Assets là tất cả các tài nguyên (resources) như hình ảnh, âm thanh, văn bản, và Prefab có thể được sử dụng trong dự án Unity. Cửa sổ Project chứa toàn bộ các Assets.

2.4.8 Camera

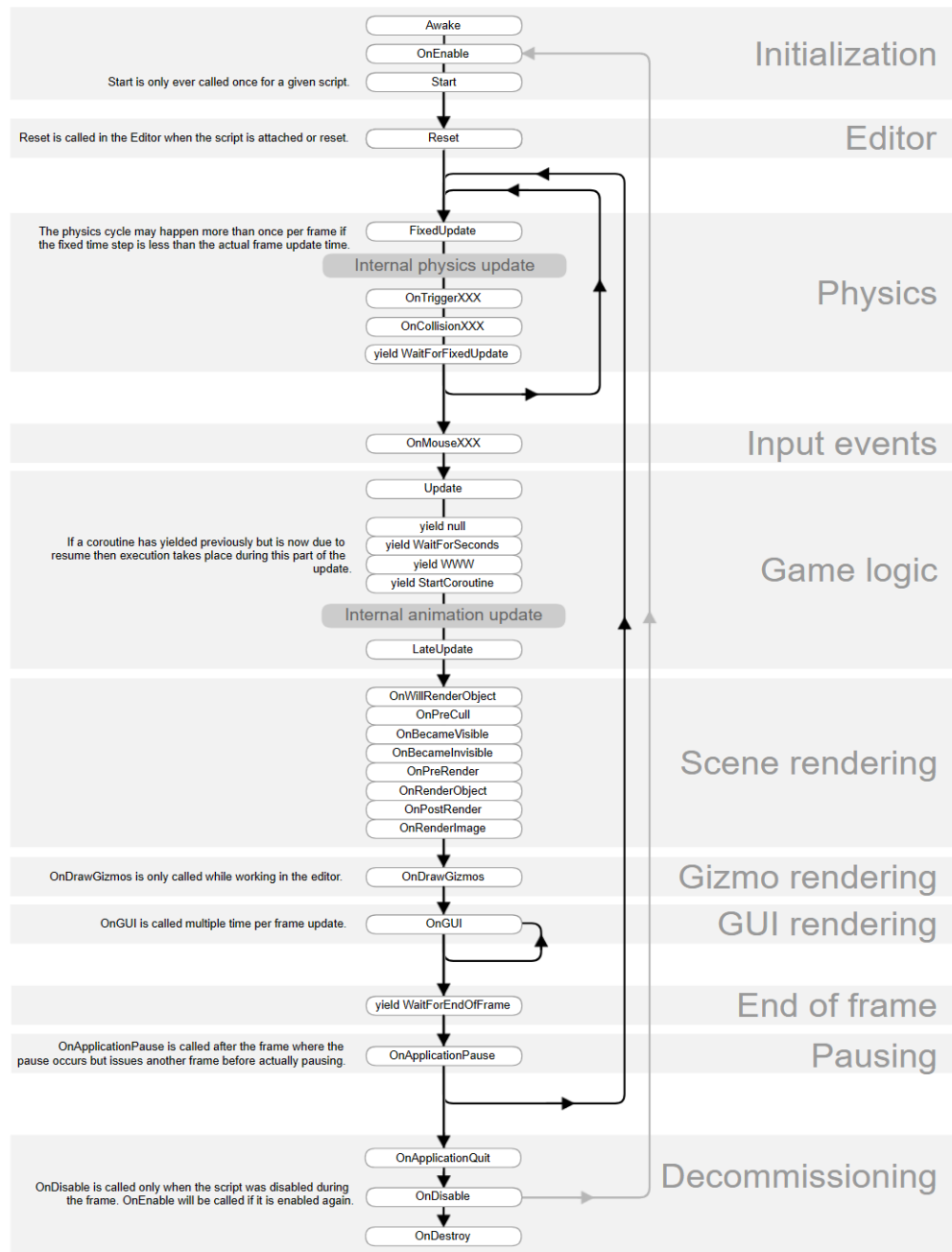
Camera quyết định góc nhìn và cách mà người chơi nhìn thấy cảnh trong trò chơi. Nó đóng vai trò quan trọng trong việc xác định cách mà trò chơi được hiển thị.

2.4.9 Sound

Sound là âm thanh trong game như nhạc nền, hiệu ứng âm thanh.

2.4.10 Unity Flow

Trong Unity, có một số event được thực thi theo thứ tự xác định, vì hầu hết các hàm sử dụng trong Unity đều kế thừa từ class MonoBehaviour. Thứ tự thực hiện này được mô tả như sau:

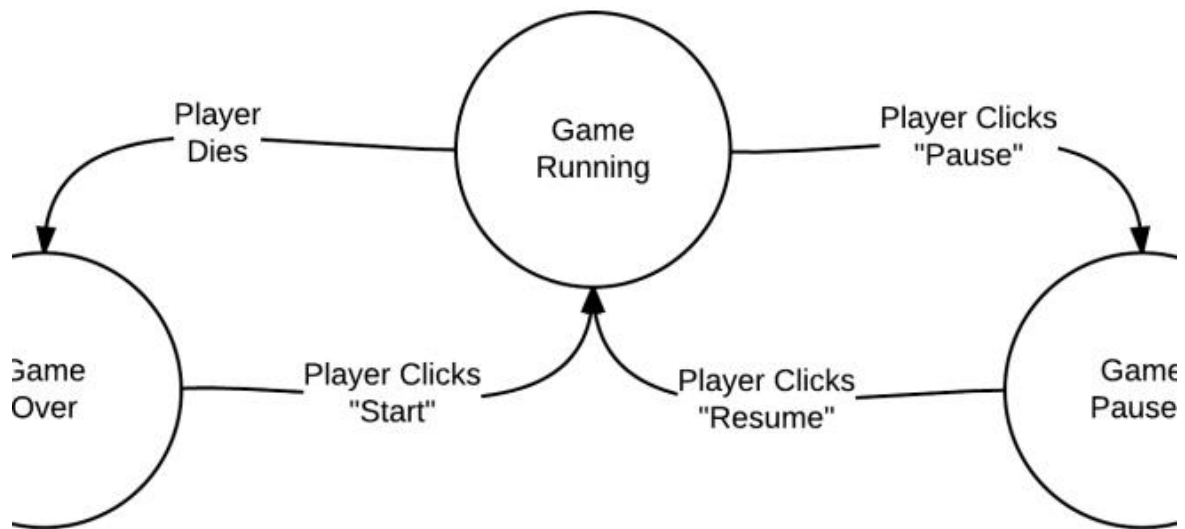


Hình 13. Thứ tự thực thi trong Unity

2.4.11 Finite State Machine:

Finite State Machine là một mô hình tính toán dựa trên một cỗ máy giả định được tạo thành từ một hoặc nhiều trạng thái của đối tượng. Chỉ một trạng thái duy nhất có thể hoạt động, do đó cỗ máy phải chuyển từ trạng thái này sang trạng thái khác để thực hiện các tác vụ khác nhau.

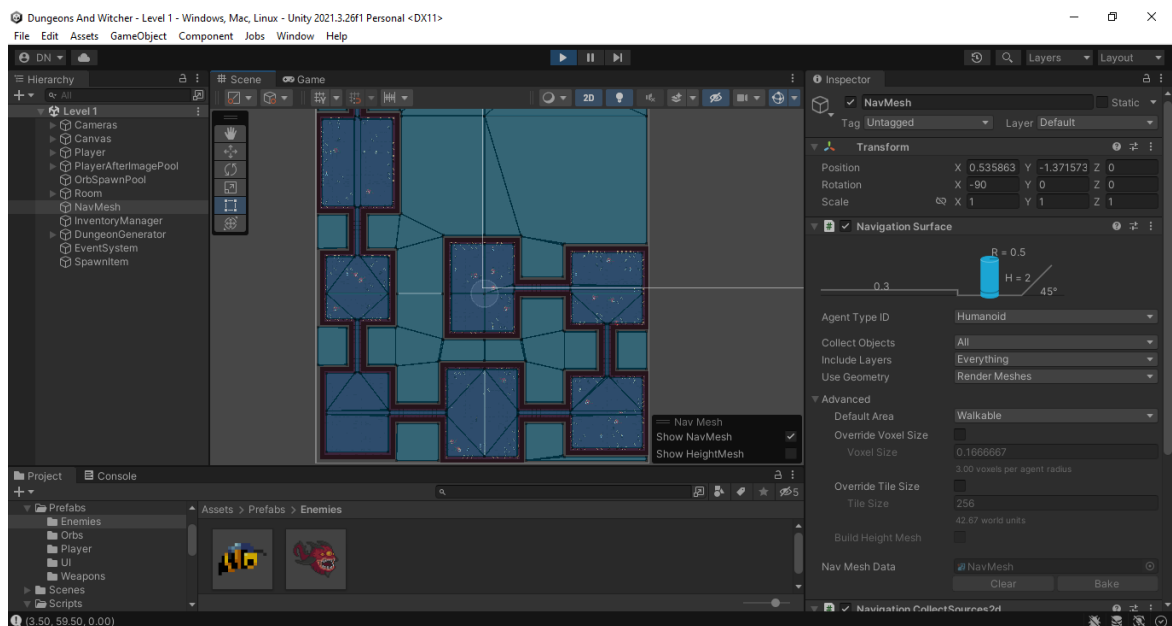
Finite State Machine thường được sử dụng để tổ chức và đại diện cho luồng thực thi, rất hữu ích để triển khai AI trong trò chơi.



Hình 14. Finite State Machine

2.4.12 Nav Mesh

Nav Mesh là thành phần có sẵn trong unity hỗ trợ nhà phát triển trong việc thiết lập AI của đối tượng. Nav Mesh sẽ tạo ra lưới đường đi với hai phần cơ bản là walkable và notwalkable. Đối tượng di chuyển tới điểm chỉ định trên walkable và né tránh các điểm notwalkable.



Hình 15. Ứng dụng navmesh trong trò chơi

2.5 Các lớp Unity API

2.5.1 Animator Class

Quản lý và phát các Animation cho GameObject.

- VD: `Animator myAnimator = myObject.GetComponent<Animator>();`

2.5.2 AudioSource Class

Quản lý và phát âm thanh trong trò chơi.

- VD: `AudioSource audioSource = myObject.GetComponent<AudioSource>();`

2.5.3 Camera Class

Quản lý cách mà trò chơi được hiển thị trên màn hình.

- VD: `Camera mainCamera = Camera.main;`

2.5.4 Collider Class

Xác định vùng va chạm của đối tượng GameObject để xử lý các sự kiện va chạm.

- VD: `BoxCollider2D myCollider = myObject.GetComponent<BoxCollider2D>();`

2.5.5 Component Class

Lớp cơ bản cho tất cả các thành phần gắn liền với GameObject.

- VD: `Rigidbody2D rb = myObject.GetComponent<Rigidbody2D>();`

2.5.6 GameObject Class

Đại diện cho đối tượng trong trò chơi và là cơ sở cho hầu hết các thành phần.

- VD: `GameObject myObject = new GameObject("ObjectName");`

2.5.7 Input Class

Xử lý đầu vào từ bàn phím, chuột, và các thiết bị khác.

- VD: `if (Input.GetKeyDown(KeyCode.Space)) { //Jump }`

2.5.8 *MonoBehaviour Class*

Là lớp cơ bản của hầu hết các script trong Unity, nó làm cho script có thể được gắn liền với GameObject, các hàm giúp script tương tác với GameObject đều cần kế thừa class này.

- VD: `public class MyScript : MonoBehaviour { /*...*/ }`

2.5.9 *Physics Class*

Cung cấp các hàm để tương tác với hệ thống vật lý trong trò chơi.

- VD: `bool isGrounded =
Physics.Raycast(transform.position, Vector3.down, 0.1f);`

2.5.10 *Rigidbody Class*

Điều khiển vật lý của GameObject, cho phép áp dụng lực và làm cho đối tượng phản ứng với lực tự do. Đây là lớp cần thiết đối với tất cả đối tượng vật lý.

- VD: `Rigidbody2D myRigidbody =
myObject.GetComponent<Rigidbody2D>();`

2.5.11 *ScriptableObject Class*

Là 1 dạng cơ sở dữ liệu đơn giản, cho phép lưu trữ dữ liệu không phải là prefab và có thể chia sẻ giữa các đối tượng.

- VD: `public class MyScriptableObject : ScriptableObject {
//code here... }`

2.5.12 *SceneManager Class*

Quản lý các scene trong dự án Unity.

- VD: `SceneManager.LoadScene("NextScene");`

2.5.13 *SpriteRenderer Class*

Hiển thị Sprite cho đối tượng trong môi trường 2D.

- VD: `SpriteRenderer SpriteR =
myObject.GetComponent<SpriteRenderer>();`

2.5.14 *Transform Class*

Định nghĩa vị trí, quay, và kích thước của GameObject.

- VD: `Transform myTransform = myObject.transform;`

2.5.15 Time Class

Cung cấp thông tin về thời gian trong trò chơi, chẳng hạn như `deltaTime`, `TimeScale`.

- VD: `float moveSpeed = 10f * Time.deltaTime;`

CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME

3.1 Ý tưởng

Ý tưởng trò chơi "Dungeons And Witcher" dựa trên cơ chế của "The Binding of Isaac" và "Noita", game tập trung vào cuộc phiêu lưu hấp dẫn của một phù thủy trong một thế giới dưới lòng đất đầy nguy hiểm. Trong game, người chơi sẽ vào vai một phù thủy và bắt đầu hành trình khám phá hàm ngục bí ẩn.

Nhiệm vụ chính của người chơi là đối mặt với những thách thức khó khăn, đánh quái vật và tiêu diệt các boss mạnh mẽ để thu thập những cây đũa phép có sức mạnh đặc biệt. Mỗi cây đũa phép sẽ mang theo một loạt các kỹ năng phép thuật độc đáo, giúp người chơi thích nghi với các tình huống khác nhau. Giữa các tầng của hàm ngục sẽ có những căn phòng trao đổi giúp người chơi nâng cấp những cây đũa phép của mình và tạo ra những phép mới lạ.

Trong suốt cuộc hành trình, người chơi sẽ phải tìm kiếm và sử dụng thông minh các vật phẩm, thuốc phép, và trang bị để cải thiện sức mạnh của họ và tăng khả năng sinh tồn. Các hàm ngục sẽ luôn thay đổi, đảm bảo sự đa dạng và tính chất ngẫu nhiên trong mỗi lần chơi.

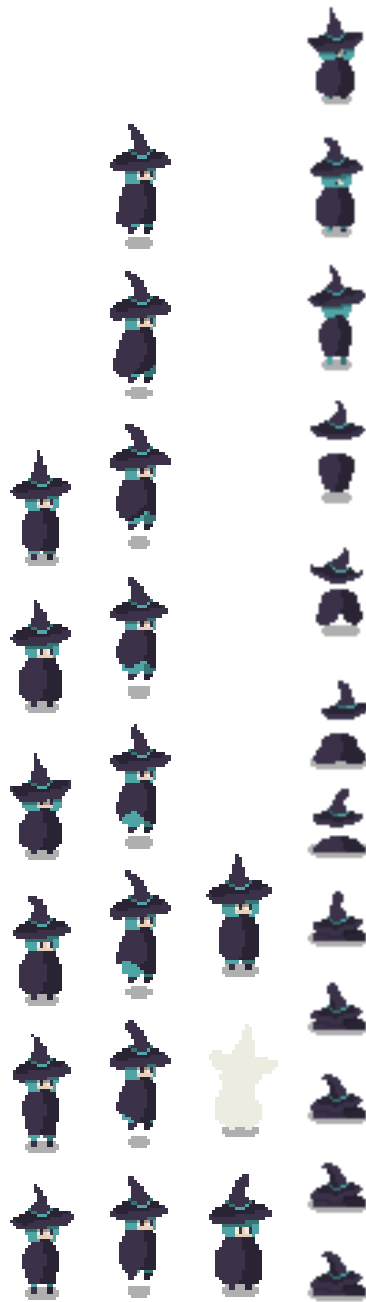
3.2 Thiết kế game

3.2.1 Thiết kế người chơi

Trạng thái người chơi: Idle, Move, Takendame, Dead.

Người chơi có chức năng:

- Di chuyển tránh chướng ngại vật, tránh kẻ địch và đạn.
- Sử dụng vũ khí để tấn công kẻ địch.
- Thu thập vật phẩm, gây phép và ngọc phép.



Hình 16. Sprite người chơi

3.2.2 Thiết kế kẻ địch

Kẻ địch có trạng thái: Idle, Move, Attack, TakenDame, Dead

Kẻ địch bao gồm: Red Monster, Bee.

- Kẻ địch sẽ di chuyển ngẫu nhiên trong phạm vi phòng.
- Kẻ địch sẽ tấn công người chơi nếu người chơi trong tầm.



Hình 17. Sprite quái vật

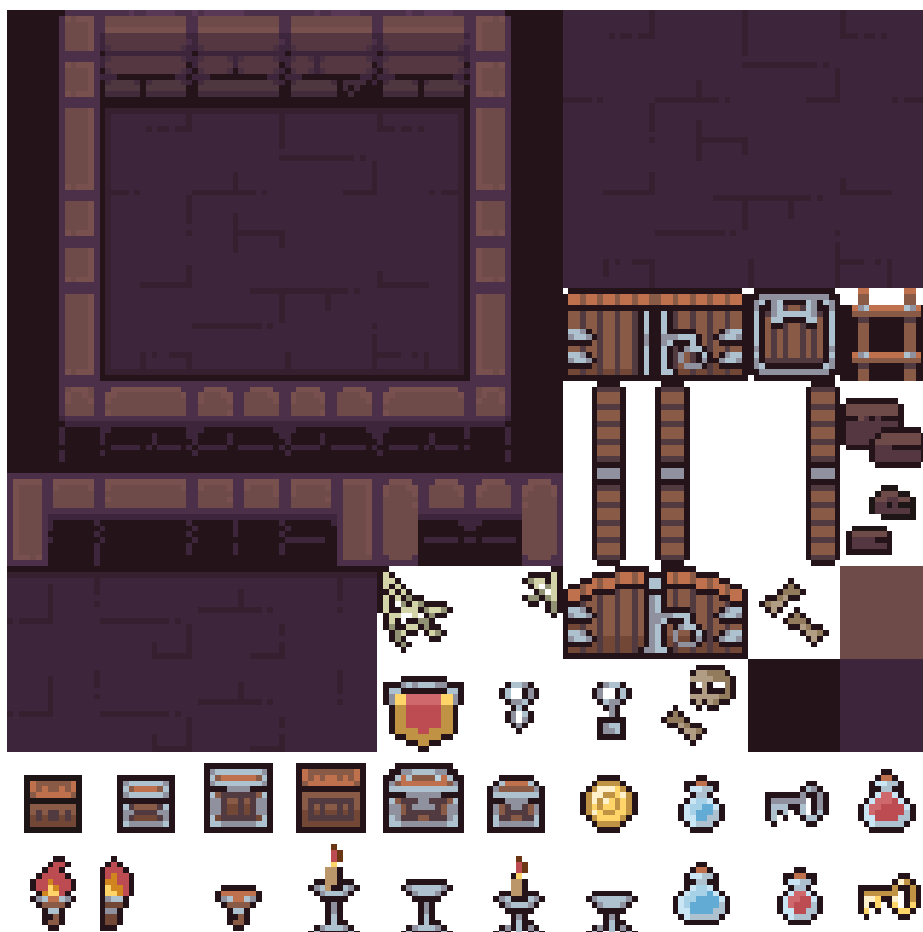


Hình 18. Sprite quái vật

3.2.3 Thiết kế map

Map gồm hành lang và phòng, quái vật sẽ xuất hiện trong phòng.

Map được tạo ngẫu nhiên khi hầm ngục được tạo.



Hình 19. Sprite map

3.2.4 Thiết kế vũ khí

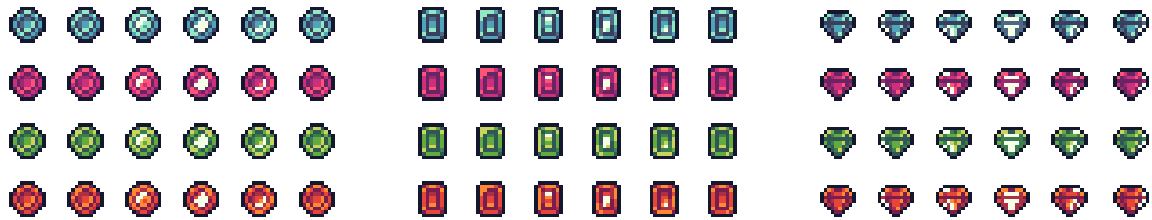
Có nhiều loại gậy phép, mỗi gậy phép lại có thông số khác nhau.



Hình 20. Sprite vũ khí

3.2.5 Thiết kế ngọc phép

Có nhiều loại ngọc phép, mỗi loại ngọc phép có thông số khác nhau.



Hình 21. Sprite ngọc phép

3.3 Các quy luật chơi

3.3.1 Di chuyển

Người chơi di chuyển bằng bàn phím, di chuyển trái, phải, lên, xuống, lướt để né tấn công của kẻ địch. Người chơi chỉ có thể di chuyển trong phạm vi camera.

3.3.2 Tấn công

Người chơi sử dụng gậy phép có thể bắn ra ngọc phép để tiêu diệt kẻ địch, mỗi kẻ địch có một lượng máu nhất định, có kẻ địch máu thấp và có kẻ địch máu cao vậy nên người chơi cần phải bắn liên tục đến khi tiêu diệt kẻ địch.

3.3.3 Thu thập

Người chơi khi vượt qua các phòng sẽ xuất hiện vật phẩm ở giữa phòng, người chơi có thể thu thập vật phẩm để nâng cấp sức mạnh của bản thân.

3.3.4 Thanh trạng thái

Người chơi có các thông số liên quan đến máu, năng lượng, chỉ số. Các thông số này thay đổi khi người chơi tấn công, nhận sát thương hoặc nhặt vật phẩm. Người chơi cần đảm bảo không làm giảm hết số máu để sống sót.

CHƯƠNG 4 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

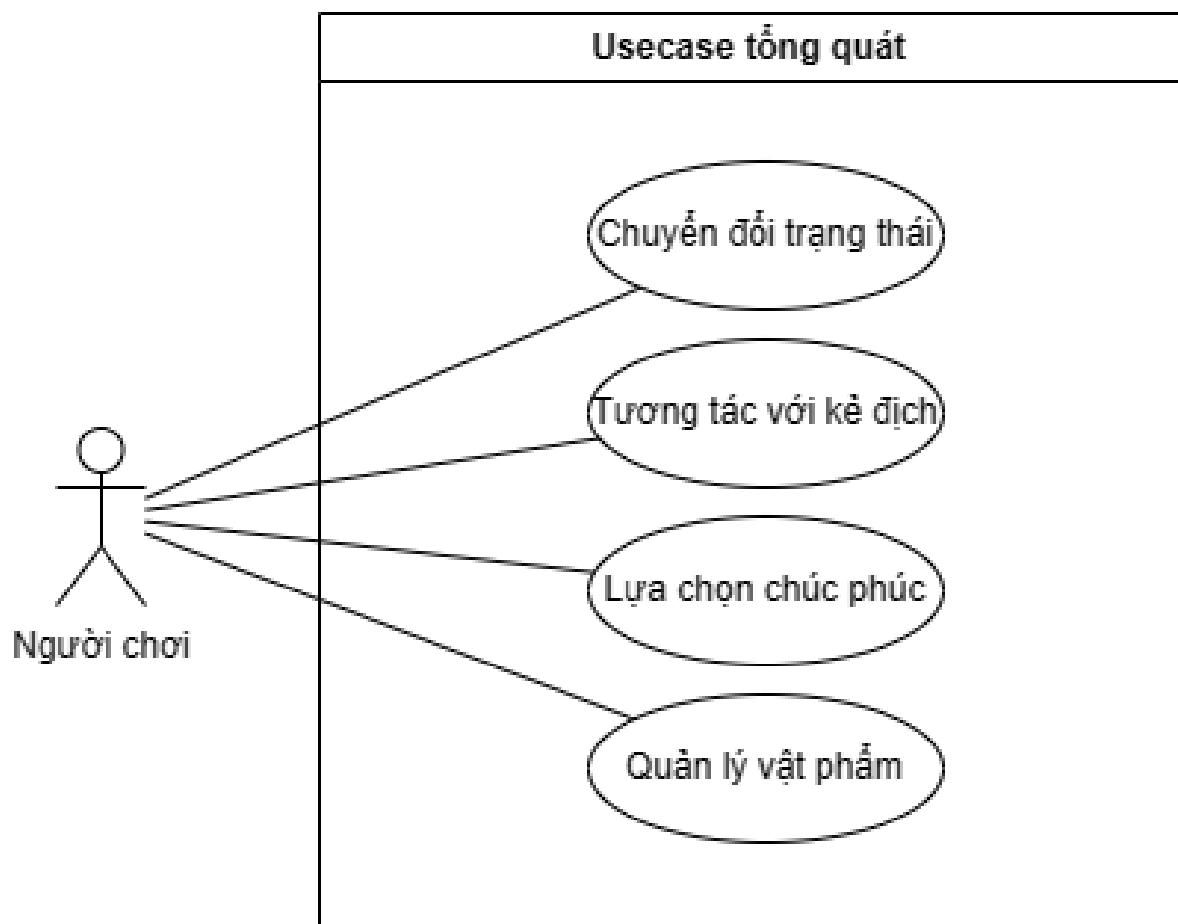
4.1 Xác định các tác nhân hệ thống

Bảng 1. Xác định các tác nhân hệ thống

| STT | Tác nhân | Nhiệm vụ |
|-----|------------|--|
| 1 | Người chơi | Điều khiển, tương tác với trò chơi thông qua dữ liệu đầu vào là các phím bấm và chuột. Các chức năng chính: <ul style="list-style-type: none">• Chuyển đổi trạng thái.• Tương tác với kẻ địch.• Lựa chọn chúc phúc.• Quản lý vật phẩm. |

4.2 Xây dựng biểu đồ Usecase

4.2.1 Mô hình Usecase tổng quát



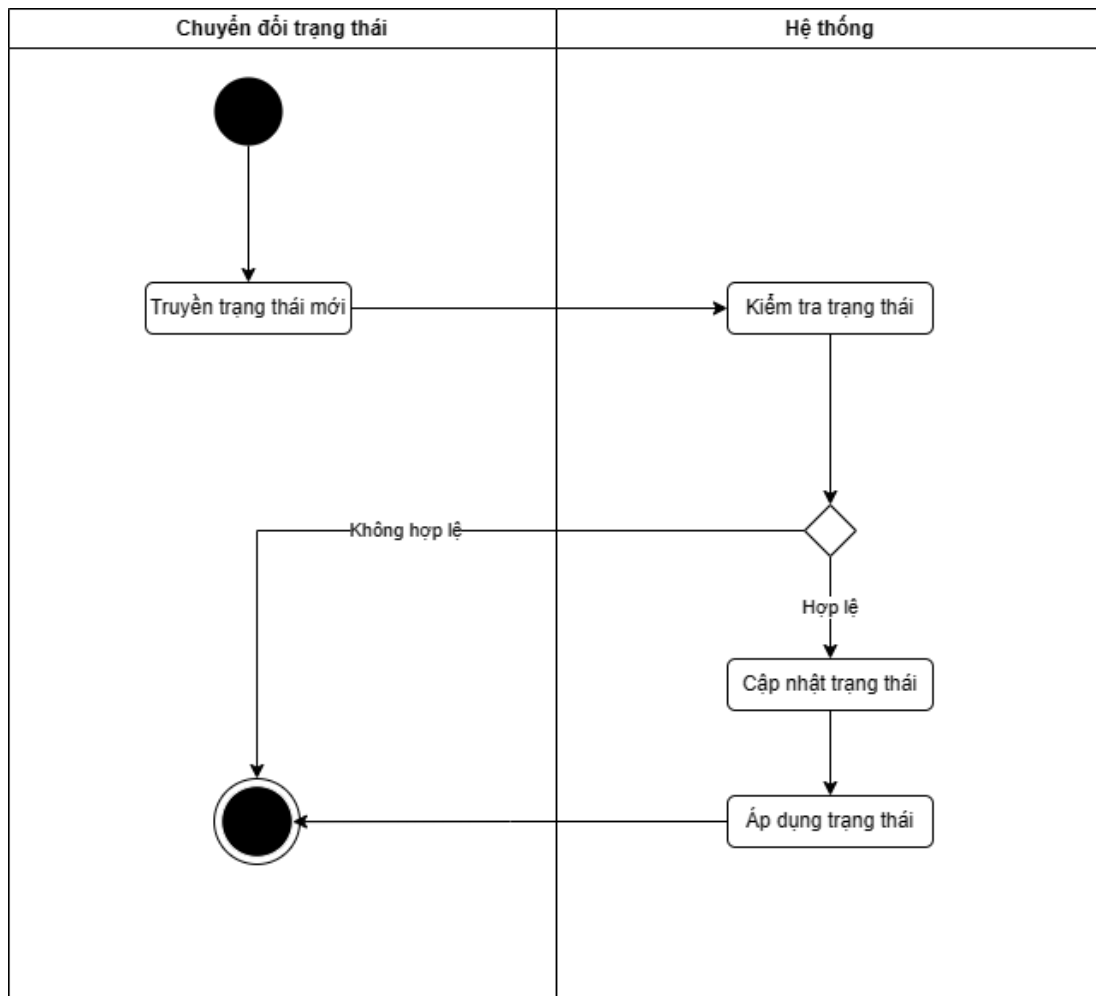
Hình 22. Mô hình Usecase tổng quát

4.3 Đặc tả và phân tích chức năng của hệ thống

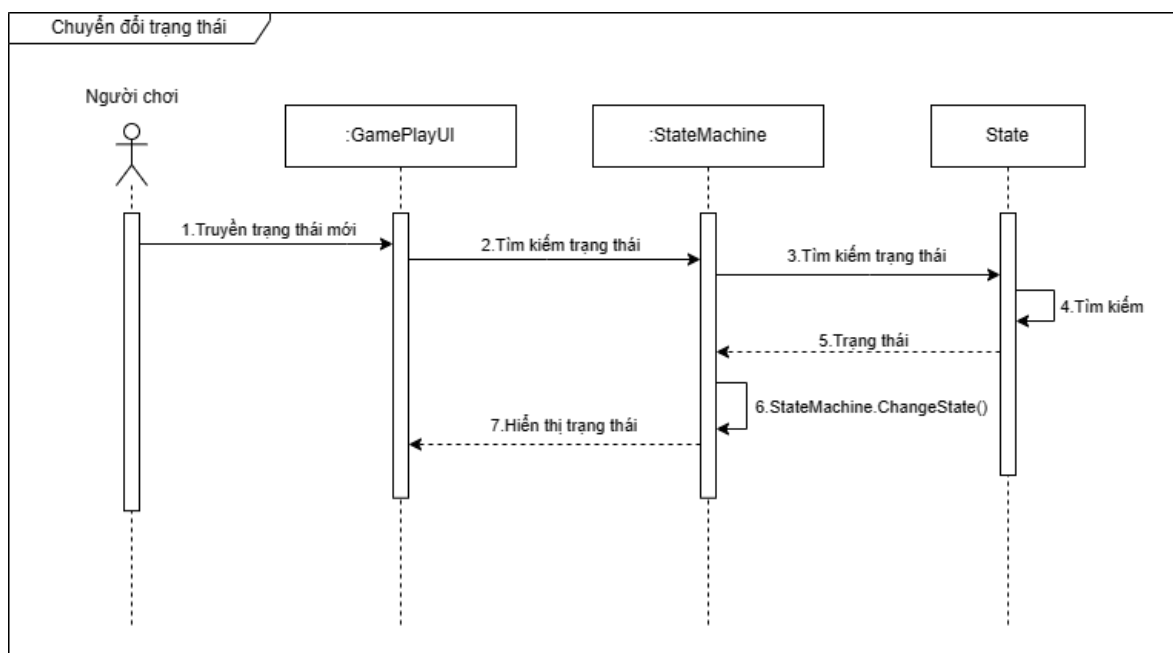
4.3.1 Usecase Chuyển đổi trạng thái

Bảng 2. Đặc tả Usecase Chuyển đổi trạng thái

| | | | |
|----------------------|--|--|---|
| Mã use-case | 01 | | |
| Tên use-case | Chuyển đổi trạng thái | | |
| Tác nhân | Người chơi | | |
| Mô tả | Chức năng cho phép tác nhân chuyển đổi trạng thái nhân vật như đứng yên, di chuyển, tấn công, tương tác,... dựa trên đầu vào tác nhân truyền đến cho hệ thống. | | |
| Điều kiện kích hoạt | Tác nhân thực hiện truyền đầu vào là các phím điều khiển chức năng di chuyển, tấn công, tương tác. | | |
| Điều kiện thành công | Trạng thái nhân vật được thay đổi. | | |
| Luồng sự kiện | | Tác nhân | Hệ thống phản hồi |
| | 1 | Tác nhân truyền trạng thái mới đến hệ thống. | |
| | 2 | | Hệ thống kiểm tra trạng thái hợp lệ, cập nhật lại thông tin và thực hiện chuyển đổi trạng thái. |
| Luồng thay thế | | | |
| Luồng ngoại lệ | 2a.1 | | Nếu trạng thái không hợp lệ, hệ thống không thực hiện chức năng. |



Hình 23. Biểu đồ luồng hoạt động Usecase Chuyển đổi trạng thái

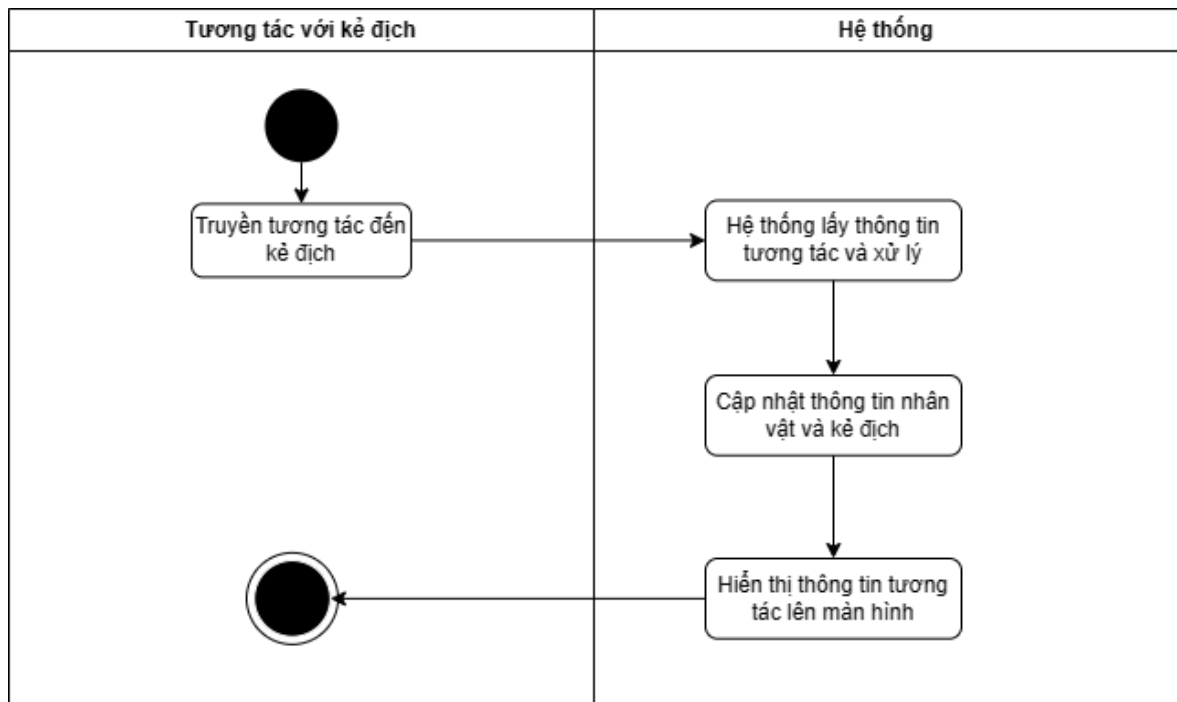


Hình 24. Biểu đồ trình tự Usecase Chuyển đổi trạng thái

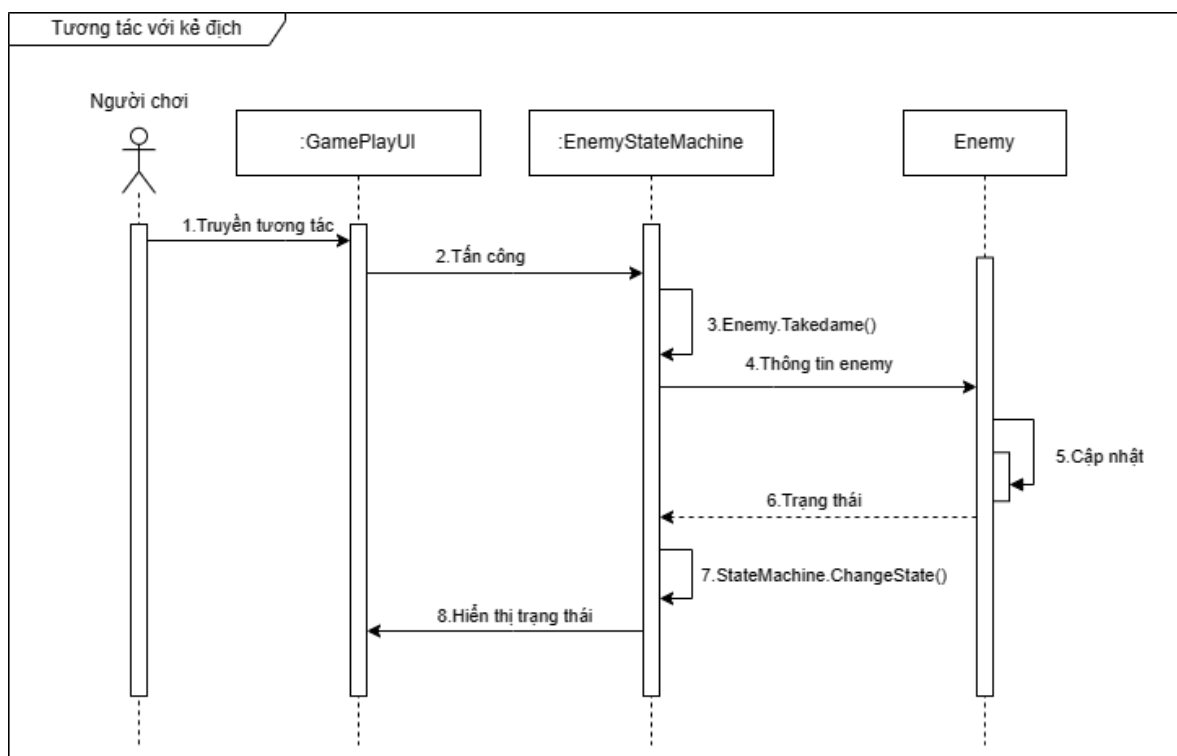
4.3.2 Usecase Tương tác với kẻ địch

Bảng 3. Đặc tả Usecase Tương tác với kẻ địch

| | | | |
|----------------------|--|---|---|
| Mã use-case | 02 | | |
| Tên use-case | Tương tác với kẻ địch | | |
| Tác nhân | Người chơi | | |
| Mô tả | Chức năng này cho phép tác nhân điều khiển nhân vật để truyền tương tác và chiến đấu với kẻ thù. | | |
| Điều kiện kích hoạt | Tác nhân thực hiện truyền đầu vào là các phím điều khiển chức năng di chuyển, tấn công, tương tác. | | |
| Điều kiện thành công | Thông tin lưu trữ của nhân vật hoặc quái vật đã được thay đổi. | | |
| Luồng sự kiện | | Tác nhân | Hệ thống phản hồi |
| | 1 | Tác nhân truyền tương tác từ nhân vật đến kẻ thù. | |
| | 2 | | Hệ thống lấy thông tin tương tác, cập nhật vào nhân vật và kẻ thù, hiển thị tương tác lên màn hình. |
| Luồng thay thế | | | |
| Luồng ngoại lệ | | | |



Hình 25. Biểu đồ luồng hoạt động Usecase Tương tác với kẻ địch

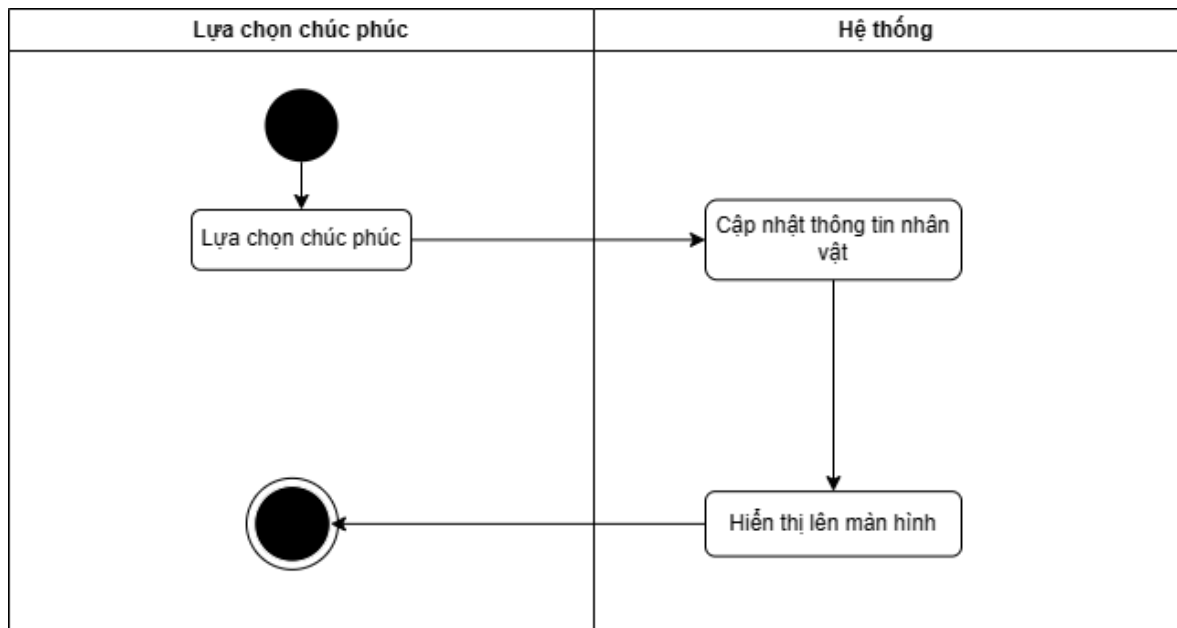


Hình 26. Biểu đồ luồng hoạt động Usecase Tương tác với kẻ địch

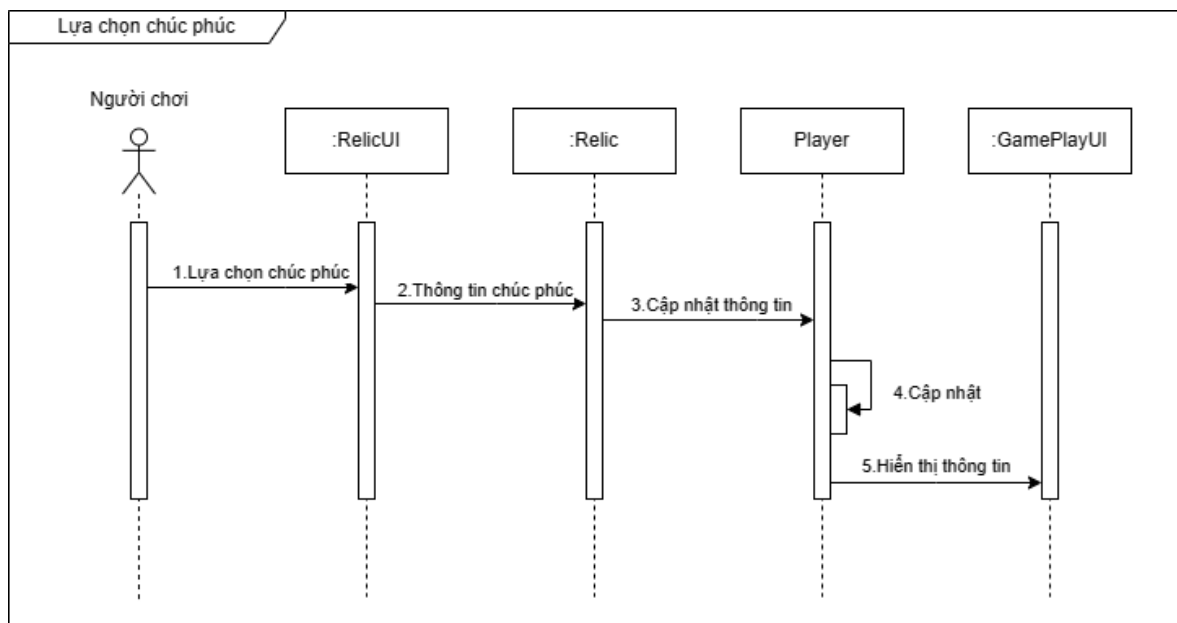
4.3.3 Usecase *Lựa chọn chúc phúc*

Bảng 4. Đặc tả Usecase Lựa chọn chúc phúc

| | | | |
|----------------------|--|------------------------------|--|
| Mã use-case | 03 | | |
| Tên use-case | Lựa chọn chúc phúc | | |
| Tác nhân | Người chơi | | |
| Mô tả | Khi kết thúc 1 phòng, có xác xuất hệ thống sẽ đưa ra các chúc phúc để tác nhân lựa chọn, tăng cường đặc biệt dành cho nhân vật mà tác nhân điều khiển. | | |
| Điều kiện kích hoạt | Tác nhân điều khiển nhân vật hoàn thành thử thách của trò chơi. | | |
| Điều kiện thành công | Thông tin lưu trữ của nhân vật hoặc quái vật đã được thay đổi. | | |
| Luồng sự kiện | | Tác nhân | Hệ thống phản hồi |
| | 1 | Tác nhân lựa chọn chúc phúc. | |
| | 2 | | Hệ thống cập nhật thông tin chúc phúc vào nhân vật và hiển thị lên màn hình. |
| Luồng thay thế | | | |
| Luồng ngoại lệ | | | |



Hình 27. Biểu đồ luồng hoạt động Usecase Chọn chức vụ

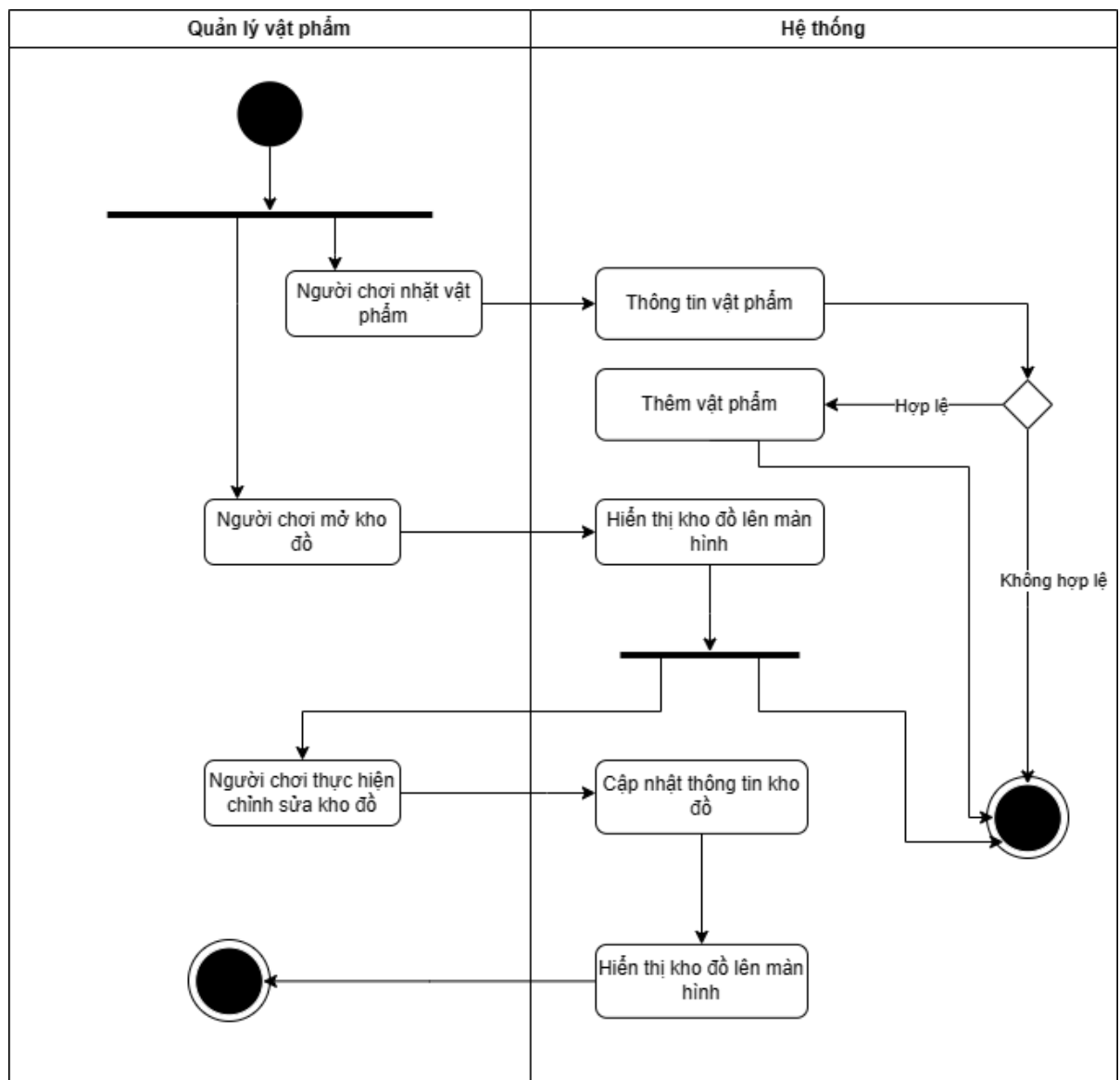


Hình 28. Biểu đồ trình tự Usecase Chọn chức vụ

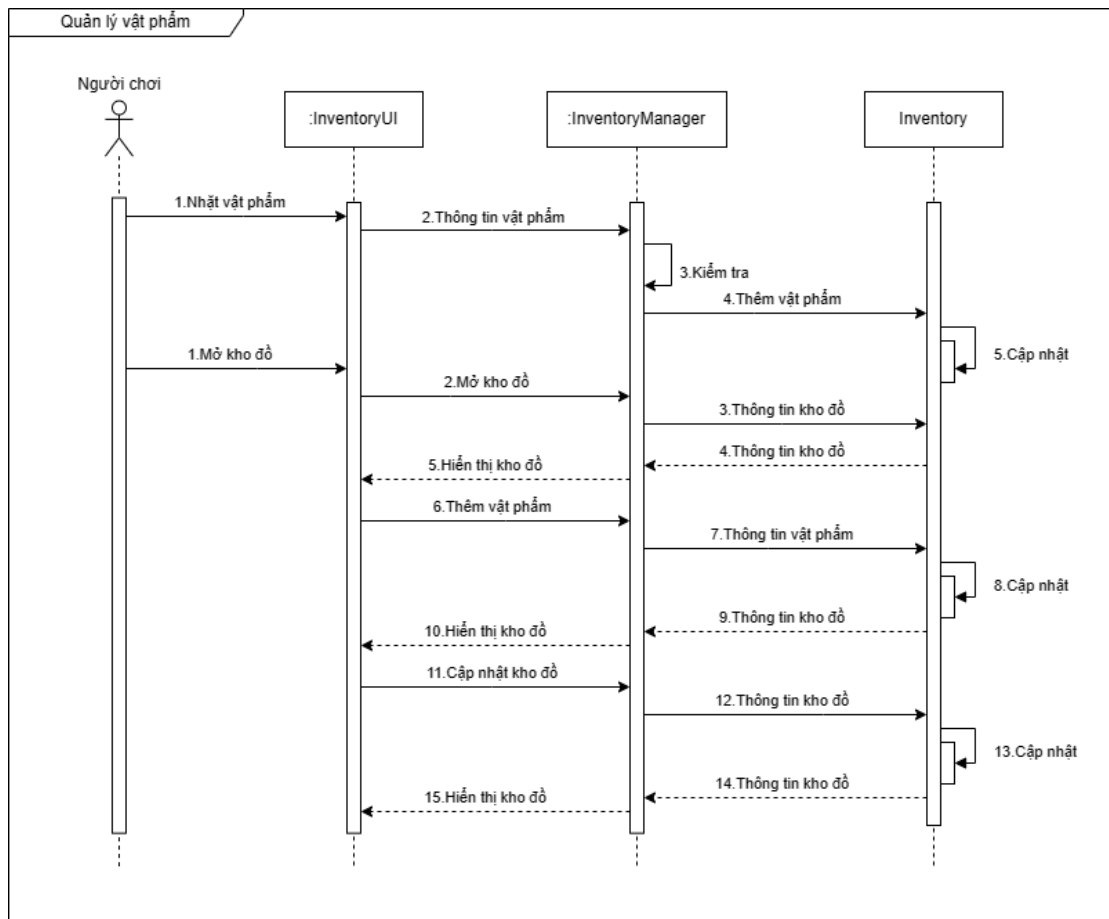
4.3.4 Usecase Quản lý vật phẩm

Bảng 5. Đặc tả Usecase Quản lý vật phẩm

| | | | |
|----------------------|--|----------------------------|--|
| Mã use-case | 04 | | |
| Tên use-case | Quản lý vật phẩm | | |
| Tác nhân | Người chơi | | |
| Mô tả | Chức năng này cho phép người chơi điều khiển nhân vật nhặt và quản lý các vật phẩm trong trò chơi. | | |
| Điều kiện kích hoạt | Tác nhân truyền tương tác nhặt hoặc quản lý vật phẩm. | | |
| Điều kiện thành công | Thông tin vật phẩm được cập nhật ở kho lưu trữ vật phẩm của nhân vật. | | |
| Luồng sự kiện | | Tác nhân | Hệ thống phản hồi |
| | 1 | Tác nhân nhặt vật phẩm. | |
| | 2 | | Kiểm tra kho đồ và thêm vật phẩm vào kho đồ. |
| | 3 | Tác nhân mở kho đồ. | |
| | 4 | | Hiển thị danh sách vật phẩm. |
| | 5 | Tác nhân chỉnh sửa kho đồ. | |
| | 6 | | Cập nhật thông tin kho đồ và hiển thị lên màn hình. |
| Luồng thay thế | | | |
| Luồng ngoại lệ | 2a.1 | | Nếu kho đồ đã đầy thì không thực hiện chức năng nhặt vật phẩm. |

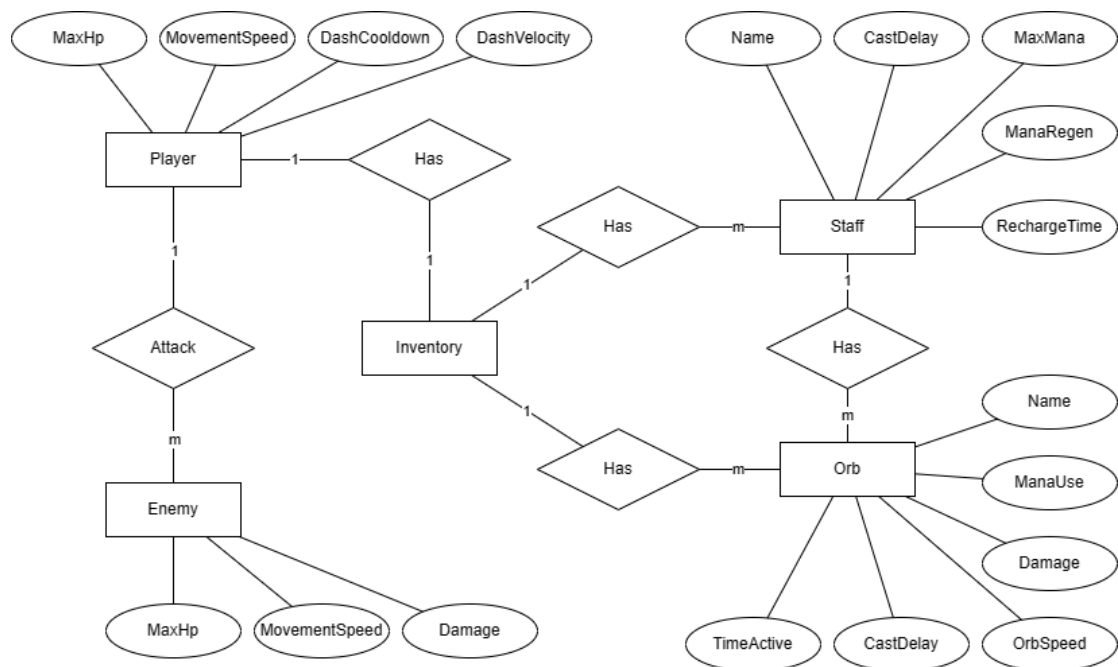


Hình 29. Biểu đồ luồng hoạt động Usecase Quản lý vật phẩm



Hình 30. Biểu đồ luồng hoạt động Usecase Quản lý vật phẩm

4.4 Biểu đồ ER



Hình 31. Biểu đồ ER

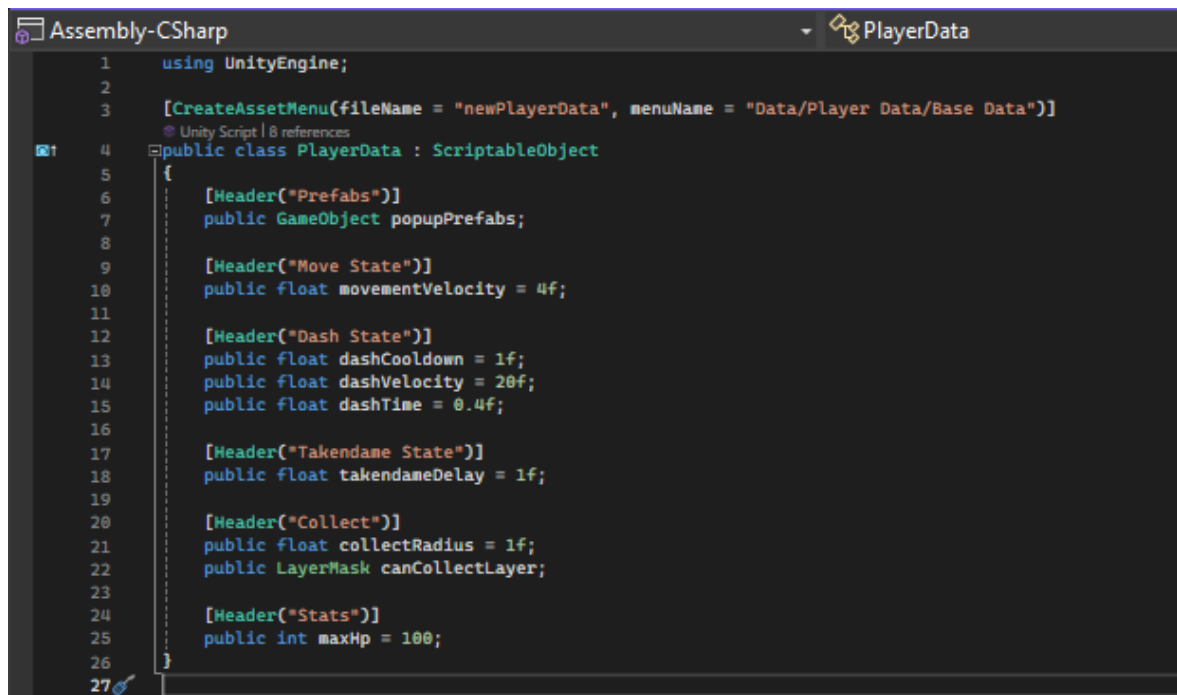
4.5 Xác định các lớp

Bảng 6. Một số lớp có trong trò chơi

| Tên lớp | Mô tả |
|-------------------|--|
| Player | Lớp người chơi, chứa code logic nhân vật, cách nhân vật tương tác với trò chơi. |
| Enemy | Lớp kẻ địch, chứa code logic kẻ địch, cách kẻ địch tương tác với người chơi và địa hình. |
| StateMachine | Lớp trạng thái, thực hiện chuyển đổi trạng thái này sang trạng thái khác của các đối tượng trong trò chơi. |
| EnemyStateMachine | Lớp trạng thái của kẻ địch, tương tự lớp StateMachine. |
| Item | Lớp vật phẩm, chứa code logic của các vật phẩm có trong trò chơi. |
| Weapon | Lớp vũ khí, mỗi vũ khí đều có thông số khác nhau, kho chứa của vũ khí, code logic của vũ khí. |
| Orb | Lớp ngọc phép, xử lý dữ liệu, logic của ngọc phép với các đối tượng khác của trò chơi. |
| Inventory | Lớp kho đồ bao gồm cả vũ khí và ngọc phép thực hiện xử lý logic liên quan đến kho đồ. |
| Map | Thực hiện chức năng tạo bản đồ, các phòng, các hành lang và môi trường. |
| SpawnerSystem | Thực hiện chức năng sinh ra quái vật và vật phẩm ngẫu nhiên trong các phòng của bản đồ. |
| Reward | Lớp phần thưởng, thực hiện xử lý khi người chơi dọn dẹp xong 1 phòng thì phần thưởng sẽ được ngẫu nhiên xuất hiện. |

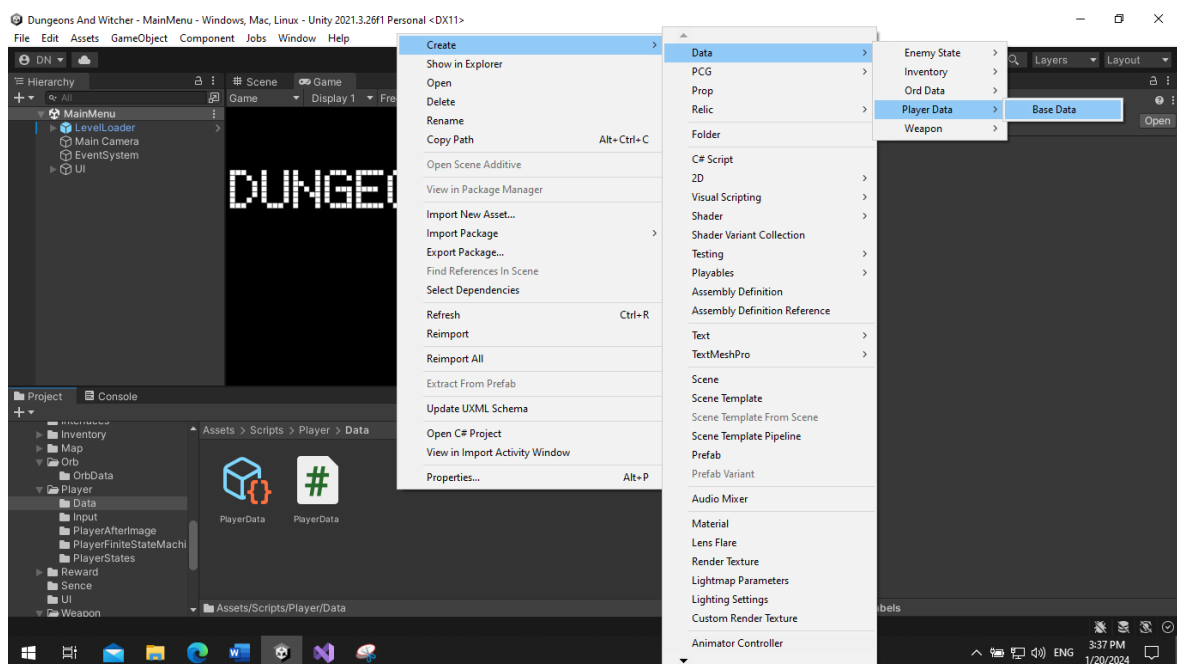
4.6 Xây dựng dữ liệu trò chơi

Unity cung cấp một lớp là ScriptableObject được sử dụng để tạo và lưu trữ dữ liệu trong trò chơi. ScriptableObject giúp ta tổ chức và quản lý dữ liệu một cách đơn giản và sử dụng chúng cho các đối tượng hoặc giữa các phiên làm việc mà không cần tạo GameObject.



Hình 32. Ví dụ về việc xây dựng ScriptableObject

Cách triển khai ScriptableObject khi đã code xong như hình ở dưới. Khi triển khai xong ta sẽ thu được 1 đối tượng là PlayerData có đuôi là “.asset”. Ta có thể chỉnh sửa trực tiếp các giá trị của đối tượng được tạo ra từ ScriptableObject thông qua editor mà không cần phải sửa lại code.



Hình 33. Ví dụ về việc Triển khai ScriptableObject

CHƯƠNG 5 XÂY DỰNG GAME VÀ DEMO

5.1 Xây dựng game

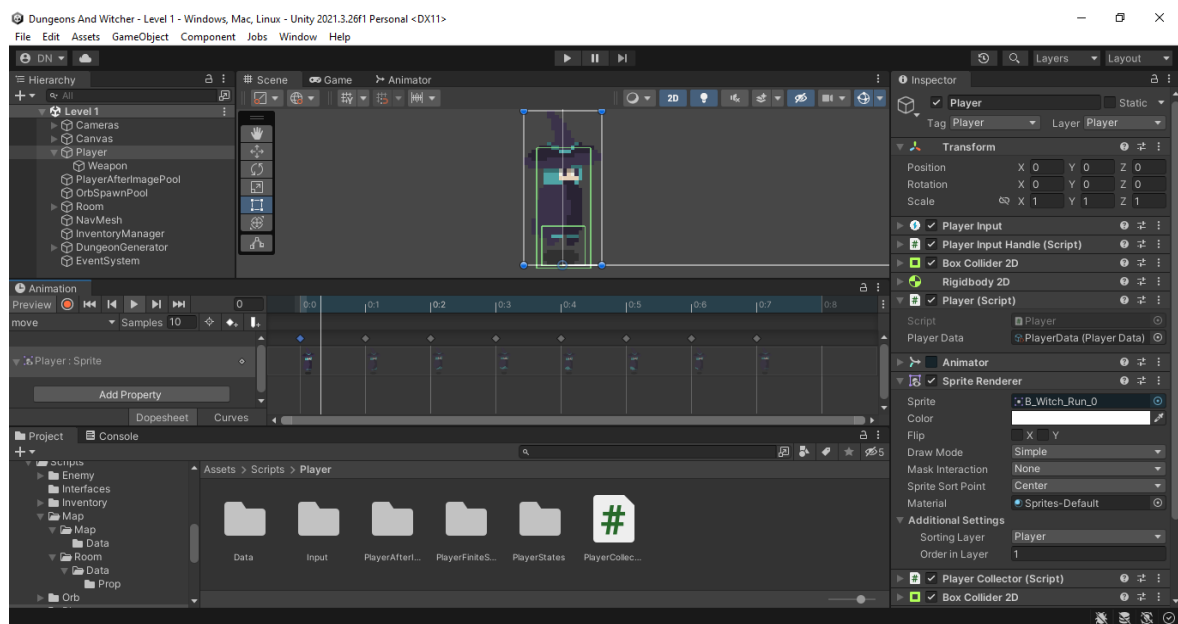
Link tải game và video demo:

https://drive.google.com/drive/folders/1-buzWCqS3Bjc6Q3HNe_m3BOV5kRX280x?usp=sharing

5.1.1 Xây dựng nhân vật

Một số công việc cần thực hiện khi xây dựng nhân vật như:

- Xử lý trạng thái, hoạt ảnh.
- Dữ liệu.
- Tương tác của nhân vật với kẻ thù.
- Xử lý logic va chạm, vật lý, viết code.



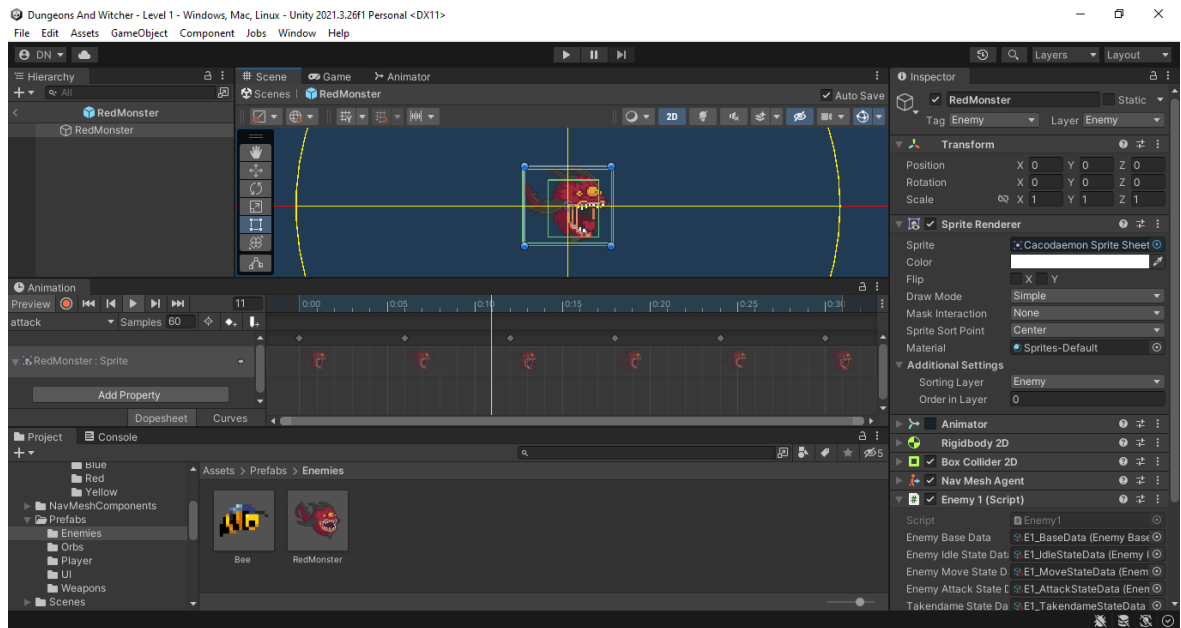
Hình 34. Xây dựng player

5.1.2 Xây dựng quái vật

Một số công việc cần thực hiện khi xây dựng quái vật như:

- Xử lý trạng thái, hoạt ảnh.
- Dữ liệu.
- Tấn công người chơi.

- Xử lý logic va chạm, vật lý, viết code.

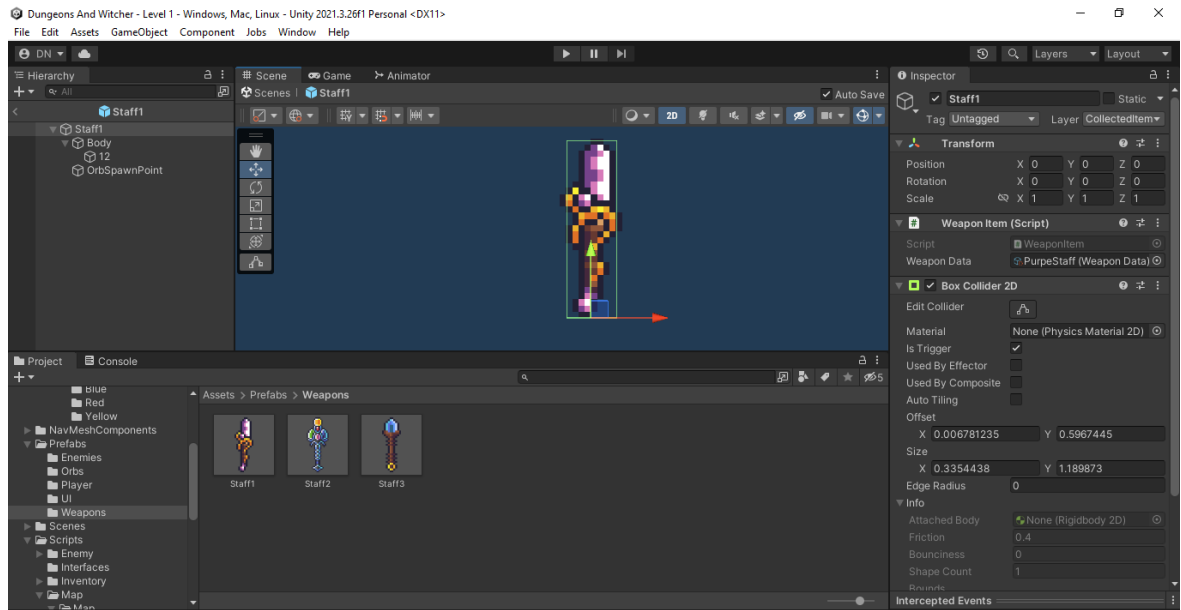


Hình 35. Xây dựng quái vật

5.1.3 Xây dựng vật phẩm:

Một số công việc cần thực hiện khi xây dựng vật phẩm như:

- Hình ảnh, vật lý, dữ liệu.
- Tương tác với vật phẩm.
- Xử lý logic va chạm, viết code.

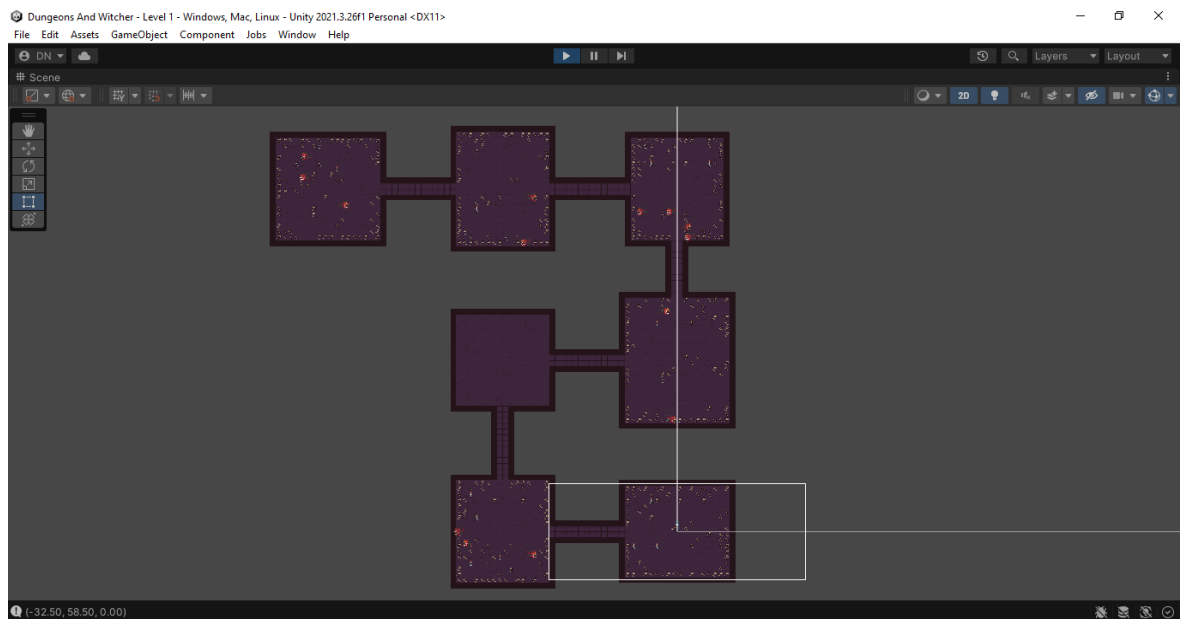


Hình 36. Xây dựng vật phẩm

5.1.4 Xây dựng Map

Map được xây dựng một cách ngẫu nhiên, mỗi phòng có 1 kích thước khác nhau, thông tin trong phòng cũng khác nhau. Một số công việc cần thực hiện khi xây dựng map:

- Hình ảnh, dữ liệu phòng, hành lang, môi trường.
- Code xử lý logic và ngẫu nhiên.



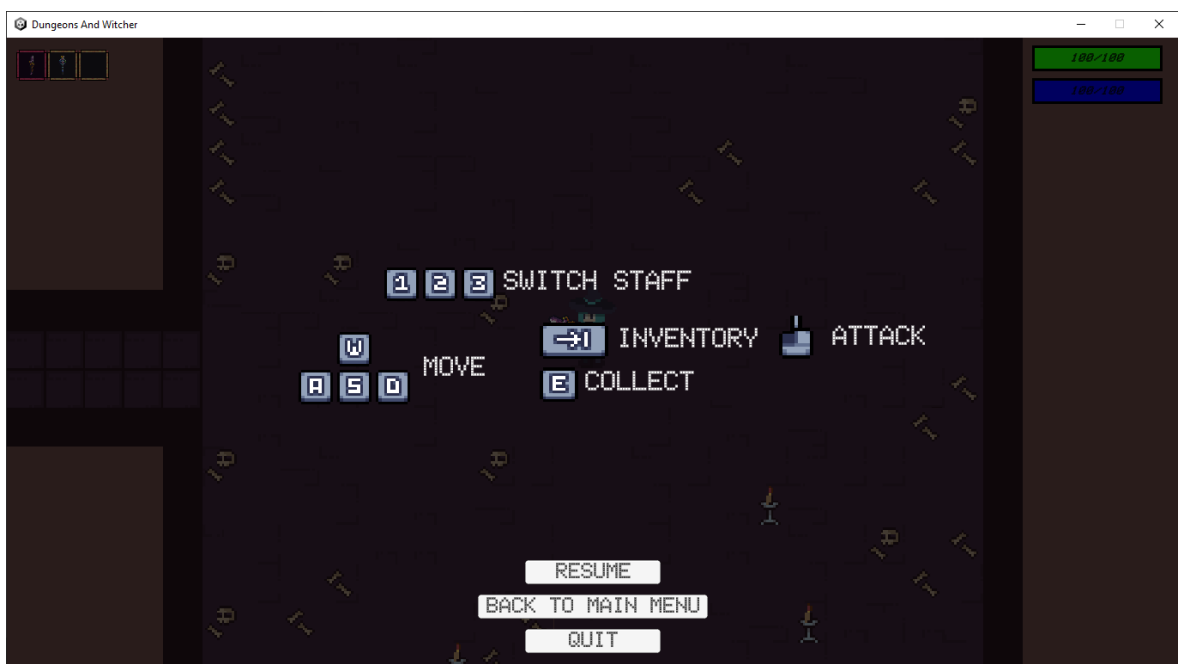
Hình 37. Xây dựng map

5.2 Hình ảnh demo

Khi thực hiện bắt đầu trò chơi, Hệ thống sẽ thực hiện sinh ra các phòng, quái vật và môi trường. Người chơi được sinh ra tại phòng được đánh dấu là phòng khởi đầu. Ở phòng này sẽ không có quái vật và người chơi có thể xem xét các trang bị khởi đầu mà hệ thống sinh ra trong kho đồ.



Hình 38. Giao diện khởi động game



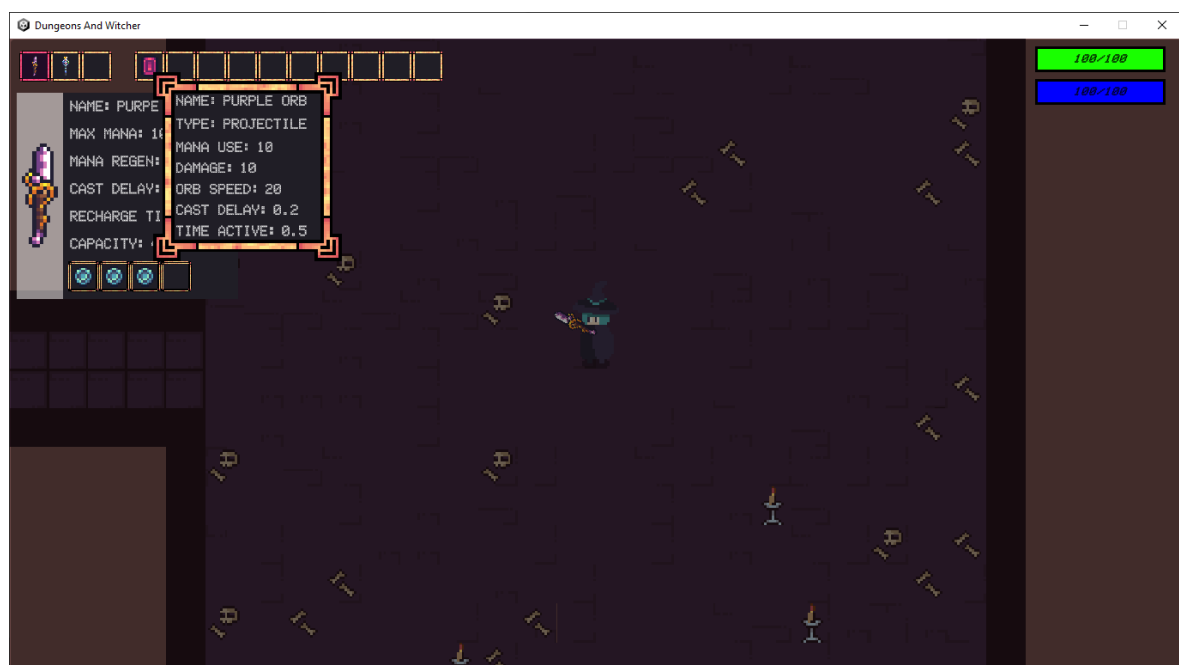
Hình 39. Giao diện cài đặt tùy chọn

Khi mở kho đồ người chơi có thể thấy được thông tin của vũ khí, ngọc phép mà mình đang có, người chơi thực hiện chỉnh sửa để có được vũ khí tối ưu nhất giúp cho quá trình vượt qua dungeon dễ dàng hơn.

Kho đồ có các chức năng cơ bản như hiển thị thông tin vũ khí đang được chọn, thay đổi vị trí vũ khí, ngọc phép, thực hiện nhặt vũ khí và ngọc phép trên sàn.

Các trường hợp đặc biệt có thể xảy ra:

- Khi số lượng vũ khí đã tối đa, vũ khí được nhặt sẽ thay thế vũ khí đang được chọn.
- Khi số lượng ngọc phép đã tối đa, ngọc phép được nhặt sẽ thay thế ngọc phép ở vị trí đầu tiên.



Hình 40. Giao diện quản lý vật phẩm

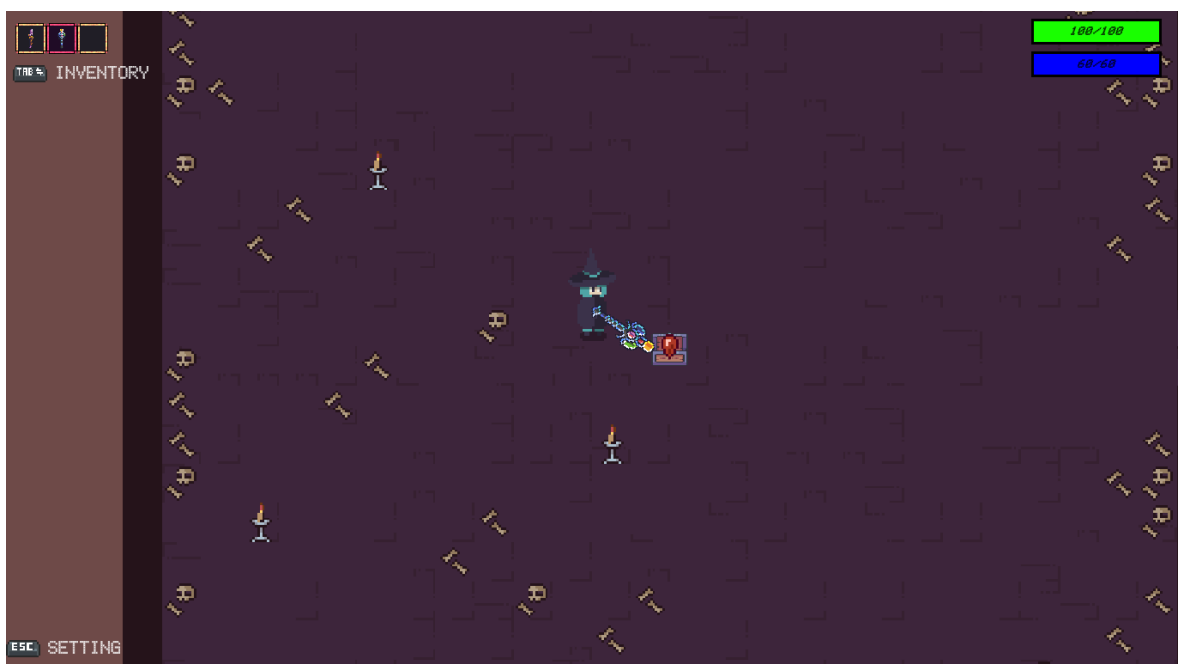
Khi người chơi đi vào khu vực phòng có kẻ địch thì kẻ địch sẽ đi về hướng người chơi nếu người chơi nằm trong tầm nhìn của kẻ địch. Nếu người chơi nằm trong tầm tấn công của kẻ địch thì chúng sẽ lao tới tấn công người chơi.

Người chơi sử dụng vũ khí hiện tại để tấn công quái vật, mỗi vũ khí sẽ có lượng mana tối đa nhất định, sử dụng ngọc phép để tấn công sẽ tiêu thụ lượng mana. Người chơi cần đảm bảo lượng mana luôn ổn định để không xảy ra trường hợp không đủ mana để tấn công quái vật.



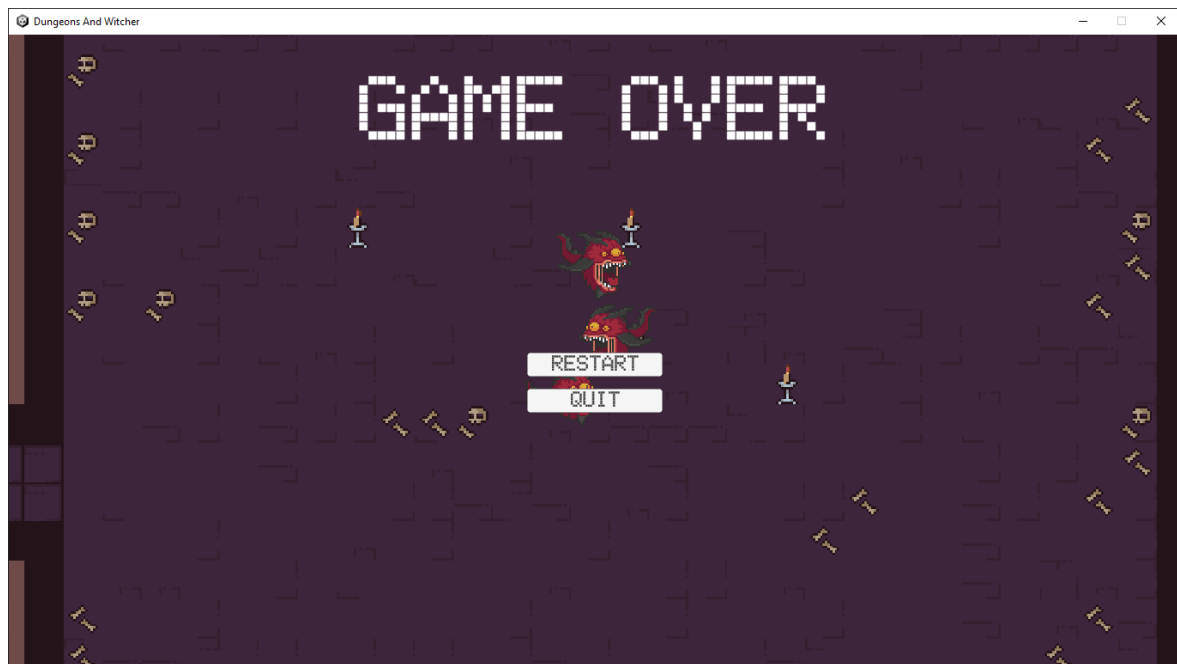
Hình 41. Tấn công quái vật

Khi tiêu diệt toàn bộ quái vật trong phòng thì rương thưởng sẽ xuất hiện tại trung tâm của phòng đó. Rương thưởng bao gồm: Vũ khí, ngọc phép, chúc phúc, máu. Mỗi loại phần thưởng đều có tỉ lệ ra khác nhau dựa vào may mắn của người chơi.



Hình 42. Rương thưởng và phần thưởng

Trò chơi sẽ kết thúc nếu người chơi để máu của mình về 0.



Hình 43. Giao diện game over

CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Trong tài liệu đồ án này, em đã trình bày về các yếu tố cơ bản về ngành game như: lịch sử, hướng phát triển của ngành trong tương lai, các công cụ và ngôn ngữ làm game. Em đang từng bước tiếp cận được với công nghệ làm game, cách thức cũng như phương pháp, biết cách kết hợp chúng lại tạo ra một trò chơi. Trong quá trình xây dựng game em nhận ra mình còn nhiều thiếu sót, code vẫn chưa chín chu và không theo đúng ý tưởng ban đầu, rất nhiều bug trong code C# cần sửa lỗi. Lập trình game rất tốn thời gian cho các công việc như design các nhân vật, bản đồ, các tài nguyên về âm thanh, hình ảnh,... Do đó để làm được trò chơi hoàn chỉnh cần có đội ngũ xây dựng phải đồng đội hơn, cần sự kiên trì và sáng tạo hơn.

Ngoài kiến thức về lập trình game em cũng đã được củng cố các kiến thức về cách viết tài liệu, lập trình hướng đối tượng, khả năng phân tích và thiết kế,...

6.2 Hướng phát triển

- Thiết kế hoàn thiện hơn về Player.
- Thiết kế thêm về vũ khí, ngọc phép tăng thêm tính đa dạng cho game.
- Thiết kế màn chơi, điều chỉnh độ khó.
- Thiết kế boss, lên màn chơi.
- Các chức năng liên quan đến cài đặt game.

TÀI LIỆU THAM KHẢO

- [1] GAMBA Team, "Tổng quan ngành game: Lịch sử, xu hướng phát triển," 04 04 2023. [Online]. Available: <https://gambaru.io/en/blog/tong-quan-nganh-game-lich-su-xu-huong-phat-trien>.
- [2] B. k. t. t. m. Wikipedia, "Unity (phần mềm làm game)," 10 07 2023. [Online]. Available: [https://vi.wikipedia.org/wiki/Unity_\(ph%E1%BA%A7n_m%E1%BB%81m_l%C3%A0m_game\)](https://vi.wikipedia.org/wiki/Unity_(ph%E1%BA%A7n_m%E1%BB%81m_l%C3%A0m_game)).
- [3] B. k. t. t. m. Wikipedia, "Unreal Engine," 02 09 2023. [Online]. Available: https://vi.wikipedia.org/wiki/Unreal_Engine.
- [4] Wikipedia, "Godot (game engine)," 06 01 2024. [Online]. Available: [https://en.wikipedia.org/wiki/Godot_\(game_engine\)](https://en.wikipedia.org/wiki/Godot_(game_engine)).
- [5] M. Quang, "Top 7 ngôn ngữ lập trình game phổ biến nhất hiện nay," 08 06 2023. [Online]. Available: <https://glints.com/vn/blog/top-ngon-ngu-lap-trinh-game-nen-hoc/#:~:text=7%20ng%C3%B4n%20ng%E1%BB%AF%20l%E1%BA%ADp%20tr%C3%ACnh%20game%20ph%E1%BB%95%20bi%E1%BA%BFn,ch%C6%A1i%20nh%E1%BB%8F.%20...%203%20HTML5%20...%20More%20items>.