

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CNTT CLC



Lập Trình Top-Down 2D Shooter Bằng UNITY

Lớp : E22CQCN01-B

Tên thành viên : Trần Đức Anh - B22DCVT034

Nguyễn Nhật Hưng - B22DCDT148

Hà Nội-2025

❖ Lịch sử cập nhật:

| Tuần | Nội dung thực hiện trong tuần |
|------|--|
| 1 | <ul style="list-style-type: none"> Dành vài ngày để trải nghiệm một số tựa game top-down, qua đó phân tích cơ chế, cách thiết kế màn chơi,... Nhằm giúp nhóm tiếp thu thêm ý tưởng. Sau khi đã dành thời gian nghiên cứu, trải nghiệm, tiếp tục tiến hành dự án sử dụng framework Unity với ngôn ngữ lập trình C#(hỗ trợ bởi Visual Code Studio), thống nhất sử dụng nền tảng 2D. Hoàn thành phiên bản đầu tiên của Giao diện chính với 3 nút là Play - đưa người chơi vào thẳng màn chơi qua 1 phần mở đầu ngắn, Credits - giới thiệu người làm, Quit - thoát game. Hoàn thiện cơ chế chính đầu tiên là cơ chế di chuyển cho người chơi, hỗ trợ di chuyển đủ 8 hướng cùng với hoạt ảnh đi kèm. Xây dựng Trang bìa, Lịch sử cập nhật, mục lục, outline cho đê tài, hoàn thành Chương I và Chương II phần 1: Chương I: <ul style="list-style-type: none"> Nguyễn Nhật Hưng Chương II: <ul style="list-style-type: none"> Trần Đức Anh: 2.1, 2.2.2, 2.2.4, 2.2.6, 2.2.8 Nguyễn Nhật Hưng: 2.2.1, 2.2.3, 2.2.5, 2.2.7 |
| 2 | <ul style="list-style-type: none"> Chỉnh sửa các lỗi của tuần 1 Update lại mục lục Tiếp tục hoàn thiện Tutorial Stage, thông nhất Asset chính sẽ được dùng, bắt đầu tìm hiệu ứng âm thanh/âm nhạc thích hợp với game. Hoàn thành giao diện người chơi, hiện tại có 3 giao diện chính là Character Sheet, Inventory, System Page. Hoàn thành giao diện hiển thị người dùng Tạo thêm map, mở rộng chi tiết môi trường, thêm các file âm thanh hiệu ứng âm thanh cho phần Tutorial và tạo transition giữa các map. Chương II: <ul style="list-style-type: none"> Trần Đức Anh: 2.2.9 Nguyễn Nhật Hưng: 2.2.10 |
| 3 | <ul style="list-style-type: none"> Sửa các lỗi của tuần 2 |

- Chỉnh sửa lại phần tài liệu tham khảo và tìm thêm các tài liệu cho các phần làm việc về sau.
- Chỉnh sửa phần thiết kế và tách phần môi trường, âm thanh, chỉnh sửa lại phần map transition của tuần trước.
- Tách phần 2.2.9 ban đầu thành 2.2.9, 2.2.10 và 2.2.11 còn 2.2.10 đổi thành 2.2.12.
- Cập nhật mục lục.
- Thiết lập cơ chế Interaction tương tác với các vật phẩm, môi trường object trong game, tạo player dialogue hiện các câu thoại hướng dẫn,....
- Tao NPC cùng với các data behavior cho NPC.
- Thiết lập cơ chế vật phẩm dùng nhanh, kết nối với giao diện Inventory và giao diện hiển thị người dùng.
- Hoàn thành cơ chế bắn súng, kết nối với giao diện người dùng.
- Thiết kế địch, tạo và triển khai A.I cho kẻ địch.

⇒Hoàn thành Tutorial Stage

- Chương II:
 - Trần Đức Anh(2.2.13, 2.2.14, 2.2.17, chỉnh sửa lại phần 2.2.9, 2.2.10, 2.2.11)
 - Nguyễn Nhật Hưng(2.2.15, 2.2.16, 2.2.18, Tài liệu tham khảo)
- Lên kế hoạch làm việc cho tuần 4

4

- Chỉnh sửa lỗi còn tồn tại trong báo cáo tuần 3
- Tái cấu trúc chương 2
- Thêm phần thiết kế tổng quan
- Chỉnh sửa lại các đề mục, thêm các mục tổng quan, chia lại các phần và đề mục
- Thêm bối cảnh cụ thể cho game
- Thiết kế và viết lại mục 2.4.chức năng chuyển map và cùng di chuyển
- Thiết kế và viết lại mục 2.5.Thiết kế hiệu ứng âm thanh theo môi trường
- Hoàn thiện thiết kế bổ sung cho Tutorial stage:
 - Chỉnh sửa hoàn thiện map Tutorial
 - Tạo thiết kế map Khu rừng giới thiệu
 - Tạo thiết kế Map Chiến đấu và giải cứu
 - Tạo Map UI:
 - Hiển thị vị trí người chơi bằng sprite và cập nhật theo thời gian thực.

| | |
|--|---|
| | <ul style="list-style-type: none"> ■ Điều khiển bản đồ với chức năng zoom, pan và cập nhật vị trí; tích hợp với hệ thống chuyển cảnh và lưu trữ. ■ Cấu hình ban đầu, gán tham chiếu cần thiết và tích hợp với các script như MapTransition (cho chuyển cảnh) và SaveController (cho lưu trạng thái). ○ Generating Map UI: <ul style="list-style-type: none"> ■ Tắt các đối tượng tạo thủ công để ưu tiên đối tượng động. ■ Tạo đối tượng động tự động cập nhật dữ liệu theo thời gian thực. ■ Điều khiển toàn bộ giao diện bản đồ, bao gồm thao tác và cập nhật vị trí. ■ Tích hợp với SaveController và MapTransition. ● Hoàn thành thiết kế cơ chế lưu/tải dữ liệu ● Bổ sung các loại súng và các cơ chế của nó. ● Thiết kế thêm các loại kẻ địch ● Cập nhật mục lục ● Thêm và Thiết kế màn 1(hoàn thiện bối cảnh thiết kế nhân vật kẻ địch, thống nhất các nhiệm vụ trong màn) ● Chương 2: <ul style="list-style-type: none"> ○ Trần Đức Anh(chỉnh sửa các mục: 2.2, 2.3.1, 2.3.2, 2.3.3, 2.6, 2.8.1; Các mục mới: 2.3.5, 2.3.6, 2.3.7, 2.4, 2.5) ○ Nguyễn Nhật Hưng(chỉnh sửa các mục: 2.1, 2.7, 2.9, 2.11, 2.12.1; Các mục mới: 2.8.2, 2.10, 2.12.2) ● Thay đổi và lên kế hoạch làm việc cho tuần 5 |
|--|---|

5

| | |
|--|---|
| | <ul style="list-style-type: none"> ● Cho thêm hình vẽ minh họa hỗ trợ cho báo cáo, sử dụng Draw.io. ● Tiếp tục thiết kế tổng quan màn 1 của game. ● Tạo các cảnh chơi nhiệm vụ về đêm cho màn chơi ● Thiết kế màn hình thông số tính điểm sau các màn chơi ● Tạo thêm các nhân vật chơi được với các kỹ năng riêng biệt chuẩn bị cho các màn chơi tiếp theo ● Sửa lại toàn bộ cốt truyện. ● Thiết kế giao diện chọn nhân vật khi vào màn chơi. ● Điều chỉnh lại thứ tự các phần tại chương II sao cho hợp lý và gọn gàng: ● Cập nhật mục lục |
|--|---|

| | |
|---|--|
| | <ul style="list-style-type: none"> ● Chương II: <ul style="list-style-type: none"> ○ Trần Đức Anh: Thêm vào các hình vẽ kèm giải thích chi tiết cho phần thiết kế và UI các phần: 2.3.1, 2.3.2, 2.3.3; Bổ sung và hoàn thiện Thiết kế tổng quan màn 1: 2.3.7, bổ sung phần mở mapUI: 2.10.3. ○ Nguyễn Nhật Hưng: các mục chỉnh sửa: 2.1, 2.2; các mục mới: 2.6, 2.7, 2.8.3. ● Lên kế hoạch cho tuần 6: hoàn thiện màn 1, thiết kế tổng quan màn 2 |
| 6 | <ul style="list-style-type: none"> ● Thay thế hình vẽ thiết kế bằng ảnh thiết kế trong UNITY. ● Đổi toàn bộ mục thiết kế Chương 2 sang Chương 3. ● Chương 2 bổ sung thêm phần Unity engine: <ul style="list-style-type: none"> ○ Các đặc điểm và tính năng của UNITY. ○ Các thành phần Trong UNITY. ○ Giao diện của UNITY. ● Hoàn thành màn 2 và 3. ● Thiết kế kẻ địch cho màn 2, 3. ● Thiết lập cơ chế mới cho kẻ địch mới. ● Chỉnh sửa lại toàn bộ đề mục. ● Chỉnh sửa lại phần 3.3(3 mới). ● Làm lại phần 4(3.4 mới) và sáp nhập vào phần 3(3.3 mới) ● Thiết kế, vẽ và mô tả Usecase Diagram cho dự án bằng Visual Paradigm. ● Cập nhật lại mục lục. ● Chương II: <ul style="list-style-type: none"> ○ Trần Đức Anh: 2;3(chỉnh và sửa phần 1) ○ Nguyễn Nhật Hưng: 1(chỉnh sửa phần 2, 3) ● Chương III: <ul style="list-style-type: none"> ○ Trần Đức Anh: 3.5 màn 2, 3; thay ảnh cho phần 3.1, 3.2, 3.3; 14. ○ Nguyễn Nhật Hưng: 8.2, thay thế hình vẽ thành ảnh cho các phần 4, 5, 6, ● Lên kế hoạch cho màn 4, 5. |
| 7 | <ul style="list-style-type: none"> ● Điều chỉnh thiết kế sơ đồ USECASE lên đầu chương III ● Thiết lập màn chơi đặc biệt và các cơ chế liên quan: <ul style="list-style-type: none"> ○ Cơ chế sinh ra kẻ địch ở vị trí bất kỳ sau mỗi đợt. ○ Cơ chế thay đổi môi trường chơi trực tiếp tại màn ● Thiết kế thêm các sơ đồ USECASE chi tiết cho các module chính |

| | |
|---|---|
| | <ul style="list-style-type: none"> ● Tạo bảng scenario cho các module chính ● Tạo Entity class diagram cho các module chính ● Tạo Full class diagram cho các module chính ● Chỉnh sửa mục lục ● Chương III: <ul style="list-style-type: none"> ○ Trần Đức Anh: 1.1, 1.2.4, 1.2.5, 1.2.6 ○ Nguyễn Nhật Hưng: 1.2.1, 1.2.3, 1.2.4, 4.5 (Màn đặc biệt) ● Lên kế hoạch cho tuần sau: <ul style="list-style-type: none"> ○ Hoàn thành Sequence diagram(+ senario ver2) ○ Chuyển sang pha thiết kế |
| 8 | <ul style="list-style-type: none"> ● Sửa các lỗi đã note trong báo cáo tuần 7: <ul style="list-style-type: none"> ○ Loại bỏ tất cả các biểu đồ lớp và lớp thực thể. ○ Viết lại văn phong cho các phần 9 tới phần 14. ● Tạo Use case, scenario cho các cơ chế chính được liệt kê cụ thể trong báo cáo. ● Thiết kế biểu đồ hoạt động cho các module và mô tả. ● Sử dụng UML để phân tích ý tưởng, cụ thể là sử dụng để trình bày ý tưởng cho các cơ chế chính: <ul style="list-style-type: none"> ○ Dùng General UC để mô tả các chức năng của module chính của Game Project. ○ Dùng UC chi tiết để mô tả chi tiết các chức năng của từng module chính cùng với các mô tả kịch bản trong Game Project. ○ Dùng Biểu đồ hoạt động để mô tả cách hoạt động của từng module và các hàm hoạt động gọi nhau trong các chức năng của từng module. ● Chỉnh sửa và cập nhật mục lục và lịch sử chỉnh sửa. ● Chương III: <ul style="list-style-type: none"> ○ Trần Đức Anh: 1.2.4, 1.2.5, 1.2.6, 10, 11, 13. ○ Nguyễn Nhật Hưng: 1.2.1, 1.2.2, 1.2.4, 8.2, 9, 12, 14. |

Mục Lục

| | |
|--|-----------|
| Lời mở đầu..... | 10 |
| Chương I.Phân tích game Top-Down Shooter..... | 11 |
| 1. Giới thiệu về dự án..... | 11 |
| 1.1. Lựa chọn thể loại và hướng đi/mục tiêu chính..... | 11 |
| 1.2.Mục tiêu chính của dự án..... | 11 |
| 2.Xác định yêu cầu..... | 12 |
| 2.1.Khảo sát ứng dụng thực tế..... | 12 |
| 2.2.Ngôn ngữ lập trình..... | 15 |
| 2.3.Cơ chế..... | 15 |
| 2.4.Thiết kế..... | 15 |
| 2.5.Đối tượng nghiên cứu..... | 15 |
| 2.6.Phương pháp nghiên cứu..... | 15 |
| 3. Kết luận..... | 16 |
| Chương II.UNITY engine..... | 17 |
| 1. Đặc điểm và tính năng của Unity..... | 17 |
| 1.1. Rendering (Kết xuất hình ảnh)..... | 17 |
| 1.2. Lighting (Ánh sáng)..... | 17 |
| 1.3. Terrains (Địa hình)..... | 17 |
| 1.4. Substances (Texture thông minh)..... | 17 |
| 1.5. Physics (Vật lý)..... | 18 |
| 1.6. Pathfinding (Tim đường)..... | 18 |
| 1.7. Audio (Âm thanh)..... | 18 |
| 1.8. Programming (Lập trình):..... | 18 |
| 1.9. Networking (Mạng):..... | 18 |
| 2.Các thành phần trong UNITY..... | 18 |
| 2.1. Assets..... | 18 |
| 2.2. Scenes..... | 19 |
| 2.3. Game Object..... | 19 |
| 2.4. Components..... | 19 |
| 2.5. Scripts..... | 19 |
| 2.6. Prefabs..... | 20 |

| | |
|---|-----------|
| 3.Giao diện của UNITY..... | 20 |
| 3.1.Giao Diện..... | 20 |
| 3.2.Cửa sổ Scene và Hierarchy..... | 20 |
| 3.3.Cửa sổ Inspector..... | 20 |
| 3.4.Cửa sổ Project..... | 20 |
| 3.5.Cửa sổ Game..... | 21 |
| Chương III.Thiết kế game Top-Down Shooter và kiểm thử..... | 21 |
| 1.Pha thiết kế:..... | 21 |
| 1.1.Sơ đồ USECASE chung..... | 21 |
| 1.2.Module chi tiết:..... | 22 |
| 1.2.1. Play Game Module..... | 22 |
| 1.2.2.Access Menu Module..... | 25 |
| 1.2.3.View Credits Module..... | 27 |
| 1.2.4.Exit Game Module..... | 30 |
| 1.2.5.Gaming Screen Module..... | 32 |
| 3.Thiết kế màn chơi..... | 36 |
| 4.Thiết kế chi tiết các màn chơi..... | 37 |
| 4.1.Thiết kế Main Menu..... | 37 |
| 4.2.Thiết kế Intro..... | 38 |
| 4.3.Thiết kế CreditScene..... | 39 |
| 4.4.Thiết kế Tutorial Stage..... | 40 |
| 4.5.Thiết kế tổng quan các màn chơi:..... | 43 |
| 5. Thiết kế giao diện..... | 50 |
| 6.Cài đặt màn hình tính toán thông số..... | 54 |
| 7.Cài đặt màn hình chọn nhân vật..... | 55 |
| 8. Thiết kế và triển khai cơ chế cho nhân vật..... | 55 |
| 8.1.Thiết kế nhân vật..... | 55 |
| 8.2.Cơ chế di chuyển..... | 56 |
| 8.3.Thiết kế kĩ năng riêng cho các nhân vật..... | 57 |
| 9.Thiết kế và triển khai kẻ địch..... | 58 |
| 9.1.Thiết kế địch..... | 59 |
| 9.2. Thiết kế chi tiết kẻ địch..... | 60 |

| | |
|---|-----------|
| 10. Tạo và mở rộng chức năng cho map..... | 61 |
| 10.1.Chức năng chuyển map..... | 62 |
| 10.2.Thiết Lập Vùng Di Chuyển (Trigger)..... | 63 |
| 10.3.Tạo map UI cho đè tài..... | 64 |
| 10.4.Tạo Dynamic map cho menu UI..... | 65 |
| 11 Thiết kế tương tác, tạo player dialogue và tạo NPC..... | 66 |
| 11.1. Hệ Thống Tương Tác Đa Dạng..... | 67 |
| 11.2. Hệ Thống Hội Thoại NPC..... | 68 |
| 11.3. Quy Trình Tương Tác..... | 68 |
| 12. Thiết kế cơ chế lưu/tải dữ liệu..... | 69 |
| 13.Thiết lập cơ chế vật phẩm dùng nhanh..... | 70 |
| 14.Thiết lập cơ chế bắn súng..... | 72 |
| 14.1.Tổng quan của cơ chế bắn súng..... | 73 |
| 14.2. Cơ chế riêng của các loại súng..... | 74 |

Lời mở đầu

Đây sẽ là một báo cáo Thực tập cơ sở về dự án phát triển game sử dụng Unity với ngôn ngữ lập trình C#. Game sẽ thuộc thể loại top-down với góc nhìn từ trên xuống cùng đồ họa 2d.

Để thực hiện đề tài này, dự án sẽ áp dụng một trong các framework phổ biến trong phát triển game là Unity cùng với ngôn ngữ lập trình C#. Cách tiếp cận sẽ bao gồm các bước chính sau:

- Nghiên cứu về cơ chế và đặc điểm của thể loại Top-Down Shooter.
- Thiết kế nhân vật, môi trường và hệ thống vũ khí.
- Xây dựng gameplay, cách cơ chế cốt lõi như di chuyển nhân vật, cơ chế súng đạn, AI,...
- Tối ưu hóa hiệu suất và mở rộng tính năng.
- Kiểm thử và sửa lỗi lần cuối.

Tóm tắt nội dung các chương

- Chương 1: Phân tích
 - Phân tích về gameplay.
 - Xác định yêu cầu hệ thống và công nghệ sử dụng.
- Chương 2: UNITY engine
 - Đặc điểm và tính năng của Unity.
 - Các thành phần trong Unity.
 - Giao diện của Unity.
- Chương 3: Thiết kế và kiểm thử
 - Thiết kế tổng quan.
 - Thiết kế và xây dựng các màn chơi.
 - Thiết kế và triển khai các cơ chế.

Chương I.Phân tích game Top-Down Shooter

1. Giới thiệu về dự án

1.1. Lựa chọn thể loại và hướng đi/mục tiêu chính

- Trong suốt chiều dài lịch sử của ngành phát triển game, chúng ta đã chứng kiến sự ra đời của nhiều tựa game mang tính lịch sử, cách mạng hóa hoặc tạo ra nền móng cho sự ra đời, sự thay đổi của thể loại mà nó đại diện, ví dụ như Ocarina of Time (1998) đã làm với thể loại phiêu lưu, Halo Combat Evolved với thể loại bắn súng góc nhìn thứ nhất hiện đại và cuối cùng là thể loại mà dự án hướng đến là Top-Down Shooter với đại diện tiêu biểu là seri Hotline Miami(2012).
- Ra mắt vào cuối năm 2012 với tiền thân là một dự án chưa được hoàn thiện có tên là Super Carnage (2006), Hotline Miami đã trở thành một hiện tượng, đến tận bây giờ vẫn còn được đo lén bàn cân để so sánh và xuất hiện rộng rãi qua các phương tiện văn hóa. Đối với những ai sinh ra vào giữa những năm 2000, trò chơi này một ký ức tuổi thơ, có thể qua các video gameplay trên youtube hoặc được nhìn các anh lớn chơi. Với yếu tố chơi đi chơi lại được, hiệu ứng âm thanh chân thực cùng với thể loại nhạc synthwave độc đáo, cuối cùng là lối chơi với tốc độ nhanh, tất cả những thứ đó đã trở thành ý tưởng và truyền cảm hứng cho dự án này.
- Dự án này sẽ không chỉ là một game top-down thuần túy, nó còn là sự kết hợp của các yếu tố, các đặc trưng nhằm tạo ra một trải nghiệm tốt nhất cho người chơi.

1.2.Mục tiêu chính của dự án

- Hành động tốc độ cao, yêu cầu người chơi di chuyển liên tục để né tránh đòn tấn công, khuyến khích lối chơi chủ động, chống lại việc đứng yên quá lâu, buộc phải đưa ra quyết định thật nhanh và tận dụng môi trường chiến đấu.
- Đảm bảo độ khó ở mức đủ dễ để chơi và đủ khó để chống úc ché. Độ thử thách sẽ tăng dần qua từng màn, tương ứng với độ thuần thục của người chơi.
- Kẻ địch đa dạng, bắt buộc người chơi phải sáng tạo trong việc tiếp cận màn chơi.

2.Xác định yêu cầu

- Khảo sát các ứng dụng thực tế: Dành vài ngày trước khi tiến hành dự án để trải nghiệm những game cùng thể loại, xác định ưu và nhược điểm và đặc trưng, nét riêng của từng game
- Làm quen với Unity: Biết được rõ giao diện và cách sử dụng chung của framework và thiết lập cơ chế Collaboration trên github để có thể làm việc nhóm với các thành viên.
- Nghiên cứu các cơ chế chính của top down shooter game (di chuyển, bắn, AI, va chạm): Có thêm kiến thức cơ bản về các cơ chế cốt lõi, tăng hiệu suất làm việc.
- Phát triển các module chức năng cơ bản và kiểm thử từng phần kỹ càng: Tách các phần chính thành nhiều module nhỏ nhằm cải thiện việc quản lý và lập trình đồng thời làm kiểm thử dễ dàng hơn.
- Tối ưu hiệu năng và trải nghiệm người chơi: Đảm bảo sự mượt mà trong quá trình chơi game, sửa những lỗi liên quan đến hình ảnh,...

2.1.Khảo sát ứng dụng thực tế

- Nhằm hiểu rõ hơn về điểm mạnh và điểm yếu của những game cùng thể loại. Dự án đã chọn lọc ra 3 game để trải nghiệm và nghiên cứu, sau vài ngày đã rút các nhận xét về đặc trưng chính, ưu điểm và nhược điểm như sau:

1. Super Carnage

- Là 1 dự án chưa được hoàn thành, tiền thân của Hotline Miami, cơ chế gameplay tốc độ cao, yêu cầu tính chiến thuật, sự chính xác trong việc xử lý. Game cũng giới thiệu cơ chế one-hit kill áp dụng cho cả ta lẩn địch, tăng tính chân thực cho một game lấy bối cảnh ở thế giới con người bình thường.
- Ưu điểm
 - Hệ thống vũ khí đầy đủ từ các loại súng đến các loại dao kiếm, hiệu ứng âm thanh đầy đủ tạo cảm giác chân thực
 - Dù là 1 bản thử nghiệm từ 20 năm trước nhưng game vẫn chạy ổn định trong các hệ máy hiện đại hơn
- Nhược điểm
 - Địch phản ứng quá nhanh, hệ thống A.I mất cân bằng làm việc qua màn gần như là bắt khả thi, phải chơi từ từ,

chậm rãi may ra mới hoàn thành màn nhưng lại đi ngược lại với ý định ban đầu của game.

- Thiết kế địch chưa đa dạng, chỉ có người và chó.
- Sau khi thất bại, thời gian quay lại xuất phát rất lâu, đôi lúc game sẽ bị sập trong thời gian quay lại.

2. Hotline Miami

- Kế thừa của Super Carnage, vẫn là một phong cách độc đáo với lối chơi tốc độ cao nhưng hệ thống đã được tinh chỉnh và hoàn thiện đến hoàn hảo. Không chỉ có vậy, lựa chọn thể loại âm nhạc synthwave cũng là một điểm cộng vì nó phù hợp với bối cảnh thời gian của game.
- Ưu điểm
 - Cơ chế one-hit kill nhanh chóng đưa người chơi về điểm xuất phát mỗi khi thất bại, khuyến khích sự sáng tạo. Thông thường, thời gian giữa việc quay trở lại màn chơi sau khi thất bại sẽ khá ngắn, nhưng trong Hotline Miami, chúng ta chỉ cần bấm một nút để ngay lập tức quay trở lại.
 - Hệ thống vũ khí đa dạng cùng với A.I địch nhạy bén, nhưng cũng thiết kế để cho không quá bất công cho người chơi.
 - Sử dụng thể loại nhạc synthwave làm tăng nhịp độ chơi game, làm người chơi cảm thấy phấn khích.
- Nhược điểm
 - Với hầu hết người chơi, lối chơi thất bại và quay lại liên tục này rất thú vị và sẽ gây nghiện, trong các trường hợp còn lại, có người có thể sẽ thấy ức chế.
 - Lối chơi bạo lực không phù hợp với một vài lứa tuổi và đối tượng.

3. Enter the Gungeon

- Top-down shooter + roguelike + bullet hell, độ khó cao và khắc nghiệt, trừng phạt mọi sai lầm và không có chỗ để sửa sai, có thể chơi cùng bạn bè.
- Ưu điểm:

- Thiết kế địch đa dạng, hoạt ảnh vui nhộn, phù hợp cho mọi lứa tuổi.
 - Hệ thống roguelike với rất nhiều nâng cấp, khuyến khích người chơi quay lại liên tục vì mỗi lần chơi sẽ không bao giờ giống nhau.
 - Game sở hữu hệ thống trùm xuất sắc với cơ chế đánh đa dạng và có chiều sâu, mang đến thử thách lớn.
- Nhược điểm
 - Cơ chế roguelike khiến người chơi mất hết tiến trình nếu thất bại, dễ gây ức chế..
 - Với việc có nhiều nâng cấp và vật phẩm, nếu đủ may mắn thì người chơi có thể tạo ra một sự kết hợp rất “lỗi” khiến game quá dễ làm mất cân bằng, đây có thể là mặt vừa tốt vừa xấu.
 - Rất nhiều vật phẩm và nâng cấp trong game sẽ trở nên vô dụng ở giai đoạn sau, khiến cho sự sáng tạo bị hạn chế với những người chơi lâu.
- Sau khoảng thời gian trải nghiệm và phân tích, dự án đã đề ra những ý kiến sau
 - Super Carnage cho thấy việc cân bằng A.I địch rất quan trọng, tránh việc kẻ địch phản ứng quá nhanh hoặc quá mạnh khiến người chơi cảm thấy bất công. Thử thách có thể khó nhưng không có nghĩa là nó không thể làm được.
 - Hotline Miami sẽ là tiêu chuẩn mà nhóm hướng đến, nhóm mong được thể hiện điều đó qua việc kết hợp giữa màn chơi, âm nhạc và vòng lặp thử lại liên tục nhanh chóng giúp người chơi luôn muôn tiến xa hơn.
 - Enter the Gungeon nhấn mạnh rằng yếu tố roguelike rất mất cân bằng, thế nhưng không vì thế mà game không thú vị. Điều này giúp nhóm cân nhắc cách triển khai các cơ chế nâng cấp tương tự nhằm tăng tính hấp dẫn.

2.2.Ngôn ngữ lập trình

- Sử dụng framework Unity kết hợp với ngôn ngữ lập trình C#(Sử dụng Visual studio code).

2.3.Cơ chế

- Cơ chế di chuyển theo góc nhìn từ trên xuống, tốc độ cao, va chạm và tương tác giữa các đối tượng, yêu cầu chiến thuật và cách tiếp cận màn chơi hợp lý.
- Người chơi có thể sử dụng vũ khí cận chiến, súng đạn, các nâng cấp tạm thời, và chính môi trường xung quanh để hạ gục kẻ địch và vượt qua màn chơi.

2.4.Thiết kế

- Pha trộn giữa các loại asset các nhau để tạo sự đa dạng giữa các cảnh, các màn.
- Giao diện người dùng tối giản.
- Nhạc nền, hiệu ứng âm thanh khi bắn súng, trúng đạn,...

2.5.Đối tượng nghiên cứu

- Những người mới bắt đầu học phát triển game 2D, sinh viên ngành công nghệ thông tin.

2.6.Phương pháp nghiên cứu

- Lý thuyết: tìm hiểu qua sách mở,các bài báo, báo cáo liên quan, các video tutorial và phân tích các game mẫu trên youtube
- Thực hành: Cài đặt và làm quen với Unity Engine, học các thuật toán của C#, kết nối với Visual studio code cho việc chỉnh sửa, tạo các dự án thử và kiểm tra các thành phần cơ bản của game(nhân vật, môi trường,...).

3. Kết luận

- Qua quá trình khảo sát và nghiên cứu, dự án đã xác định rõ ràng các mục tiêu và định hướng phát triển. Bằng việc phân tích những tựa game trước đó, nhóm đã rút ra những ưu điểm cần phát huy, nhược điểm cần rút ra và khắc phục, nhằm tạo ra một trải nghiệm tuyệt vời nhất cho người chơi.
- Dự án hướng đến một lối chơi tốc độ cao với cơ chế "one-hit kill", khi mà tất cả các đối tượng trong game đều có thể bị hạ gục bởi sát thương từ mọi nguồn. Cơ chế này giúp đẩy cao nhịp độ trận đấu và yêu cầu

người chơi phản xạ nhanh cùng việc có chiến thuật hợp lý. Để đảm bảo sự cân bằng, trí tuệ nhân tạo (AI) của kẻ địch sẽ được tinh chỉnh để phù hợp với mức độ thử thách.

- Bên cạnh đó, dự án cũng tích hợp hệ thống nâng cấp ngắn hạn, mang đến sự đa dạng trong lối chơi. Dù những nâng cấp này không hoàn toàn mang tính thực tế, nhưng chúng giúp tăng sự hấp dẫn và tạo ra nhiều chiến thuật. Việc giới hạn thời gian sử dụng nâng cấp cũng là một cách để duy trì sự cân bằng cần có trong game.
- Với nền tảng lý thuyết và thực hành được đặt ra, dự án hướng đến việc phát triển một sản phẩm hoàn chỉnh, đáp ứng đầy đủ các yêu cầu. Báo cáo này không chỉ là bản tổng kết quá trình nghiên cứu, mà còn là minh chứng cho những nỗ lực của dự án.

Chương II.UNITY engine

1. Đặc điểm và tính năng của Unity

- Dự án chọn và sử dụng nền tảng Unity 6 làm Framework để lập trình và phát triển game.
- Unity là một nền tảng đa năng, hỗ trợ cả 2D và 3D, với môi trường phát triển trực quan, dễ tiếp cận ngay cả với những lập trình viên mới.
- Phiên bản Unity 6 mang lại nhiều cải tiến đáng kể về hiệu suất, khả năng render hình ảnh, và hỗ trợ tốt hơn cho các hệ thống như vật lý, animation, các tính năng được miêu tả chi tiết như sau

1.1. Rendering (Kết xuất hình ảnh)

- Unity hỗ trợ kết xuất đồ họa chất lượng cao với High Definition Render Pipeline và công nghệ Universal Render Pipeline là công nghệ mà dự án đã chọn để tạo game 2D, có thể gọi tắt là URP-2D . Các công cụ này giúp tối ưu hiệu suất và nâng cao chất lượng đồ họa trên tất cả các nền tảng.

1.2. Lighting (Ánh sáng)

- Unity cũng cung cấp cho người dùng hỗ trợ cho các ánh sáng loại có hướng (Directional), điểm (Point), và hình nón(Spotlight).

1.3. Terrains (Địa hình)

- Hệ thống tạo địa hình mạnh mẽ của Unity cho phép xây dựng bản đồ môi trường mở với công cụ sculpting và painting chi tiết, hỗ trợ cả texture nhiều lớp và hệ thống cây cối, cỏ dại sống động.

1.4. Substances (Texture thông minh)

- Unity tích hợp tốt với các tệp Substance từ Adobe, cho phép sử dụng vật liệu thông minh có thể tùy chỉnh các thông số (như độ gồ ghề, màu sắc, vết xước...) trực tiếp trong editor để tạo ra bề mặt vật thể đa dạng và sinh động.

1.5. Physics (Vật lý)

- Unity tích hợp hệ thống vật lý mạnh mẽ thông qua PhysX (của NVIDIA), hỗ trợ va chạm, trọng lực, mô phỏng lực, rigidbody 2d, rigidbody 3d, joints, raycasting, v.v... giúp các tương tác trong game trở nên thực tế hơn.

1.6. Pathfinding (Tìm đường)

- Unity sử dụng hệ thống **NavMesh** để xử lý tìm đường cho các đối tượng AI. Điều này cho phép các NPC di chuyển một cách thông minh, tránh vật cản và phản ứng theo môi trường.

1.7. Audio (Âm thanh)

- Unity hỗ trợ phát âm thanh 2D và 3D, cung cấp các công cụ xử lý âm thanh theo không gian, âm lượng, pitch, hiệu ứng âm thanh (spatial blend, reverb zones), mang lại trải nghiệm âm thanh chân thực.

1.8. Programming (Lập trình):

- Unity sử dụng ngôn ngữ C# để lập trình logic game. Ngoài ra, hệ thống MonoBehaviour, event system và API phong phú giúp dễ dàng quản lý đối tượng, xử lý sự kiện và mở rộng tính năng.

1.9. Networking (Mạng):

- Unity hỗ trợ phát triển game nhiều người chơi qua hệ thống Unity Transport Layer, Netcode for GameObjects hoặc sử dụng các giải pháp bên ngoài như Photon, Mirror, giúp xây dựng các trò chơi online với đồng bộ hóa dữ liệu theo thời gian thực.

2.Các thành phần trong UNITY

2.1. Assets

- Assets là kho chứa toàn bộ dữ liệu và tài nguyên của dự án game, bao gồm hình ảnh, âm thanh, video, mô hình 3D, script, shader... Các file này được import vào Unity và sử dụng trong quá trình xây dựng game.

2.2. Scenes

- Scenes là các “cảnh” hay “màn chơi” của game. Mỗi scene chứa các đối tượng và môi trường tạo nên một phần của trò chơi. Người phát triển có thể tạo, sắp xếp và chuyển đổi giữa các scene khác nhau để xây dựng cấu trúc tổng thể của game.

2.3. Game Object

- Game Object là các đối tượng cơ bản trong Unity, có thể đại diện cho nhân vật, vật thể, hoặc các yếu tố khác trong game. Tất cả mọi thứ mà bạn thấy trong scene đều là Game Object. Các Game Object có thể được cấu hình và mở rộng thông qua các thành phần (Components).

2.4. Components

- Components là các thành phần gắn vào Game Object, giúp xác định hình dạng, hành vi và chức năng của chúng. Mỗi Game Object đều có thể chứa nhiều component, từ việc quản lý đồ họa, vật lý cho đến điều khiển logic trò chơi.

Ví dụ: Một Game Object có thể có component “Mesh Renderer” để hiển thị hình ảnh, “Rigidbody” để mô phỏng vật lý hoặc “Collider” để xử lý va chạm.

2.5. Scripts

- Scripts (hay mã lệnh) là các đoạn chương trình được viết bằng ngôn ngữ lập trình (thường là C#) dùng để định nghĩa hành vi, logic và tương tác của các Game Object trong game.

Ví dụ: Script điều khiển hành động của người chơi, quản lý trạng thái của game hoặc xử lý sự kiện,...

2.6. Prefabs

- Prefabs là các mẫu Game Object được lưu trữ như một asset, cho phép tái sử dụng các thiết lập cấu hình đã được định nghĩa trước. Khi sử dụng Prefabs, bạn có thể dễ dàng tạo ra nhiều bản sao của đối tượng

có cùng cấu hình mà không cần phải tạo lại từ đầu, giúp tiết kiệm thời gian và đảm bảo tính nhất quán.

3.Giao diện của UNITY

3.1.Giao Diện

- Giao diện của Unity được thiết kế dưới dạng một môi trường làm việc tích hợp (IDE) với nhiều cửa sổ chuyên dụng, giúp người dùng dễ dàng quản lý, chỉnh sửa và kiểm tra dự án của mình.

3.2.Cửa sổ Scene và Hierarchy

- Cửa sổ Scene: Đây là không gian trực quan để bạn xây dựng, đặt vị trí và chỉnh sửa các Game Object trong cảnh.
- Cửa sổ Hierarchy: Hiển thị cấu trúc cây của tất cả các Game Object trong scene, cho phép quản lý mối quan hệ giữa các đối tượng (ví dụ: cha-con).

3.3.Cửa sổ Inspector

- Cửa sổ Inspector hiển thị các thuộc tính và cấu hình chi tiết của đối tượng hiện đang được chọn.
- Từ đây, bạn có thể chỉnh sửa các component, biến đổi (transform) và thông số khác của đối tượng.
- Một vài yếu tố mới có thể xuất hiện trong cửa sổ inspector bằng cách gán script để tạo thêm.

3.4.Cửa sổ Project

- Cửa sổ Project liệt kê tất cả các asset, file, và thư mục chứa trong dự án.
- Đây là nơi bạn quản lý tài nguyên, di chuyển, nhóm hoặc tìm kiếm các tệp cần thiết cho việc phát triển game.

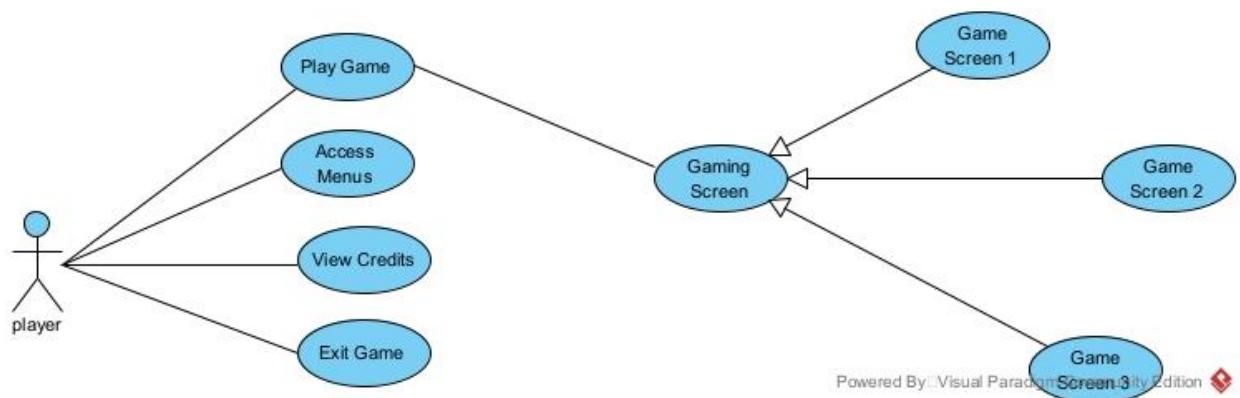
3.5.Cửa sổ Game

- Cửa sổ Game cho phép bạn xem trước kết quả của trò chơi trong thời gian thực khi đang chạy hoặc thử nghiệm.
- Đây là giao diện hiển thị game như người chơi cuối cùng sẽ thấy khi sử dụng sản phẩm hoàn thiện.

Chương III.Thiết kế game Top-Down Shooter và kiểm thử

1.Pha thiết kế:

1.1.Sơ đồ USECASE chung



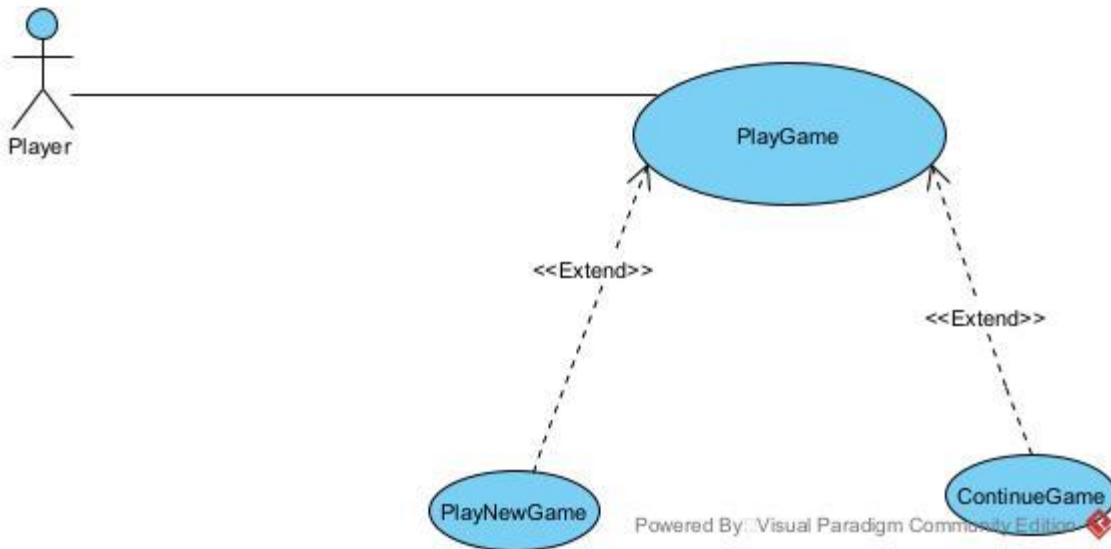
Mô tả:

- Use case "Play Game" cho phép Người chơi chơi trò chơi thông qua các màn hình game: 1, 2 và 3.
- Use case "Access Menus" cho phép Người chơi tương tác với các menu của dự án.
- Use case "View Credits" cho phép Người chơi xem thông tin về các nhà phát triển của dự án.
- Use case "Exit Game" cho phép Người chơi rời khỏi trò chơi.

1.2.Module chi tiết:

1.2.1. Play Game Module

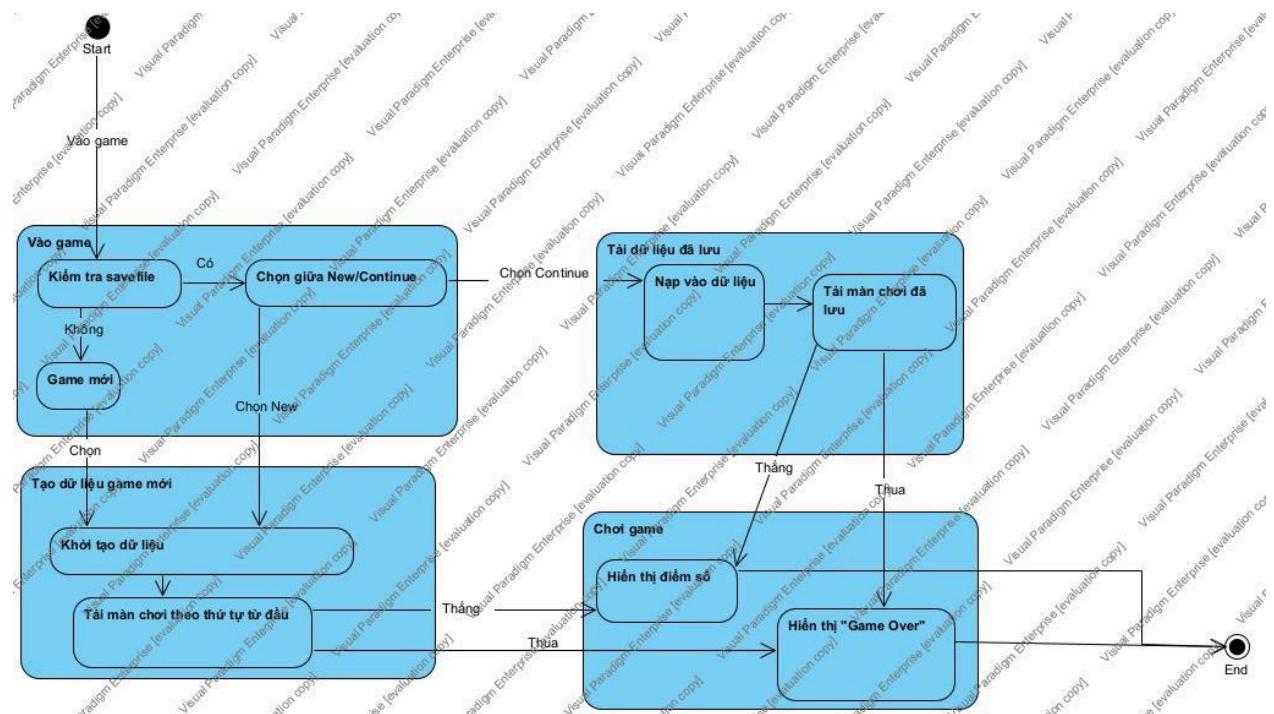
a) Use case diagram



b) Scenario

| | |
|----------------|---|
| Module | PlayGame |
| Actor | Người chơi |
| Pre-condition | Người chơi đã chọn "Play" từ menu chính. |
| Post-condition | Người chơi hoàn thành màn chơi hoặc thất bại và quay lại menu. |
| Main-event | <ol style="list-style-type: none"> 1. Người chơi nhấn "Play". 2. Hệ thống kiểm tra file JSON lưu trữ, nếu có file, hiển thị "New Game" và "Continue". 3. Người chơi chọn giữa "New Game" hoặc "Continue" để bắt đầu từ đầu hoặc tải dữ liệu đã lưu. 4. Hệ thống tải tài nguyên cho màn chơi. 5. Người chơi hoàn thành màn chơi hoặc bị hạ gục. -Hệ thống hiển thị màn hình tính điểm hoặc "màn hình Game Over" |
| Exception | <ol style="list-style-type: none"> 2.1. Không có file JSON khi chọn "Continue". 4.1. Lỗi tải tài nguyên màn chơi. 4.2 Người chơi thoát game giữa chừng. |

Biểu đồ hoạt động



1. Người chơi click chọn Play tại menu chính.
2. Menu chính gọi script SaveManager yêu cầu kiểm tra có file save hay không:
 - SaveManager thực thi CheckSaveFileExists().
3. Nếu có file save:
 - Menu chính hiển thị tùy chọn New/Continue cho người chơi chọn.
 - Người chơi chọn New hoặc Continue:
 - Nếu chọn "New":
 - SaveManager thực thi CreateNewSaveData().
 - SaveManager khởi tạo dữ liệu mới.

- SceneManager thực thi LoadScene("FirstLevel").
- Nếu chọn "Continue":
 - SaveManager thực thi LoadSaveData().
 - SaveManager nạp dữ liệu lưu.
 - SceneManager thực thi LoadScene(SaveData.currentMap).

4. Nếu không có file save:

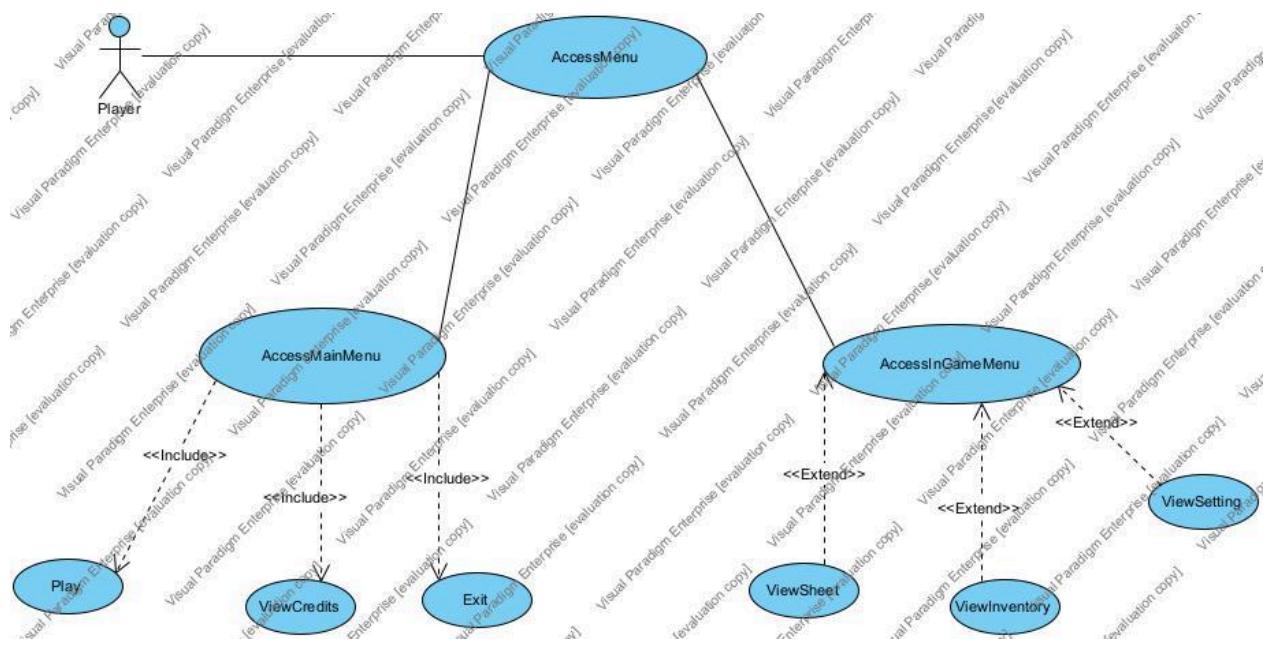
- SceneManager thực hiện LoadScene("FirstLevel").

5. Trong quá trình chơi:

- Nếu người chơi chiến thắng màn chơi:
 - Hiển thị điểm số chiến thắng (ShowVictoryScreen()).
- Nếu người chơi thất bại (bị hạ gục):
 - Màn hình hiển thị thông báo game over (ShowGameOverScreen()).

1.2.2. Access Menu Module

a) Usecase diagram

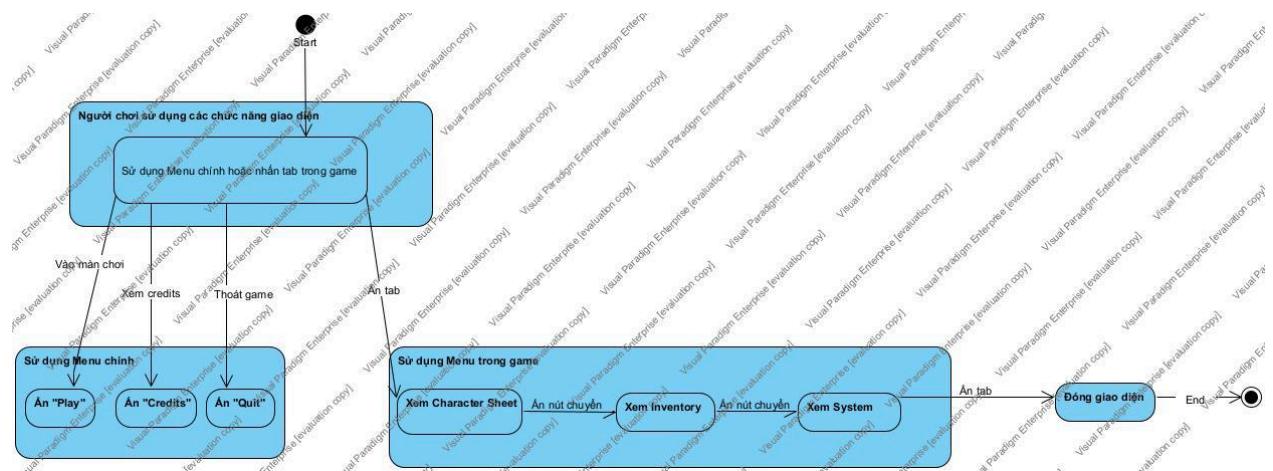


b) Kịch bản

| | |
|----------------|---|
| Scenario | Access Menu |
| Actor | Người chơi |
| Pre-Condition | Trò chơi đang ở menu chính hoặc trong giao diện UI. |
| Post-Condition | Người chơi chuyển sang module khác / tiếp tục chơi. |
| Main event | <ol style="list-style-type: none"> Người chơi chọn các tùy chọn trên menu chính. Người chơi chọn "Play" để vào Play Game Module. Người chơi chọn "Credits" để vào View Credits Module. Người chơi chọn "Exit" để vào Exit Game Module. Trong game, người chơi nhấn "Tab" để mở UI. |

| | |
|-----------|--|
| | <p>6. Người chơi chuyển đổi giữa các tab: Character Sheet, Inventory, System.</p> <p>7. Người chơi xem thông tin nhân vật, quản lý vật phẩm, lưu/tải game.</p> |
| Exception | <p>1.1 Lỗi hiển thị UI.</p> <p>2.1,2.2,2.3. Hệ thống không phản hồi khi chuyển tab.</p> |

Biểu đồ hoạt động



1. Người chơi click chọn chức năng Play, Credits, Exit hoặc nhấn Tab để mở UI trong game.
2. Khi chọn "Play":
 - Menu chính gọi đến script SceneManager.
 - SceneManager thực hiện tải và chuyển sang màn chơi (LoadScene("PlayScene")).
3. Khi chọn "Credits":
 - Menu chính gọi script SceneManager.

- SceneManager thực hiện tải và chuyển sang màn hình credits (LoadScene("CreditsScene")).

4. Khi chọn "Exit":

- Menu chính gọi hàm Application.Quit() trong script để thoát trò chơi.

5. Khi nhấn "Tab" trong game:

- Hệ thống gọi đến script UIController bật Menu UI.

6. Trong Menu UI, người chơi có thể:

- Xem Character Sheet:
 - UIController sẽ hiện CharacterSheet đầu tiên.

- Án nút chuyển sang Inventory:
 - UIController tắt CharacterSheet, bật Inventory.

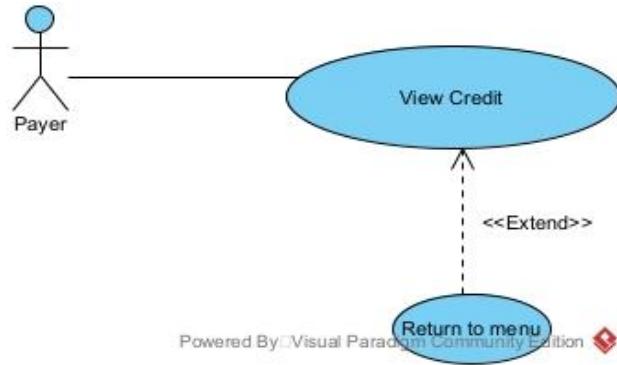
- Án nút chuyển sang System:
 - UIController tắt Inventory, bật System.

7. Người chơi nhấn lại Tab:

- UIController tắt tab cuối cùng đang được bật, quay lại trò chơi.

1.2.3. View Credits Module

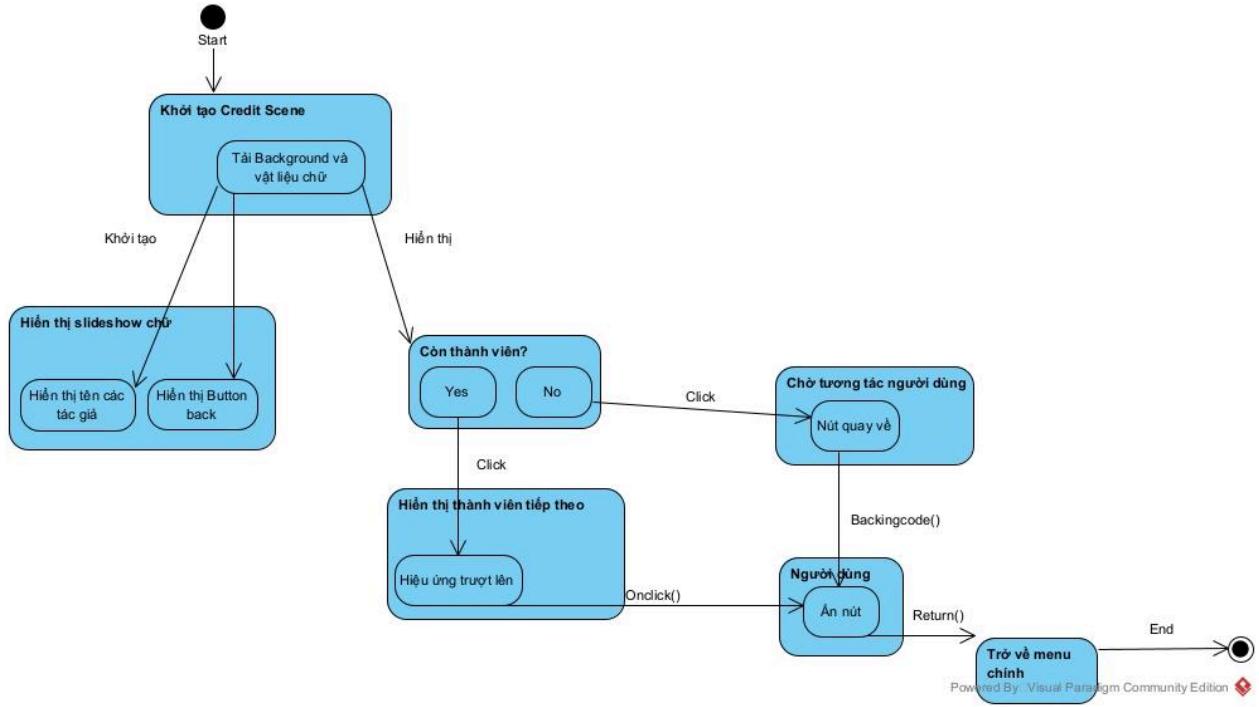
a) Usecase diagram



b) Kịch bản

| | |
|----------------|---|
| Scenario | View Credits |
| Actor | Người chơi |
| Pre-Condition | Người chơi đã chọn "Credits" từ menu chính. |
| Post-Condition | Người chơi quay lại menu chính sau khi xem Credits |
| Main event | <ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình chuyển sang Credits Scene. 2. Văn bản credits trượt từ dưới lên. 3. Hệ thống hiển thị thông tin tác giả và đồng thời cùng với nhạc nền phát kèm. 4. Người chơi nhấn "Back" để quay lại menu chính. |
| Exception | <ol style="list-style-type: none"> 1. Lỗi tải văn bản credits. 3.1. Âm thanh không phát. 4.1. Nút "Back" không hoạt động. |

c) Biểu đồ hoạt động:

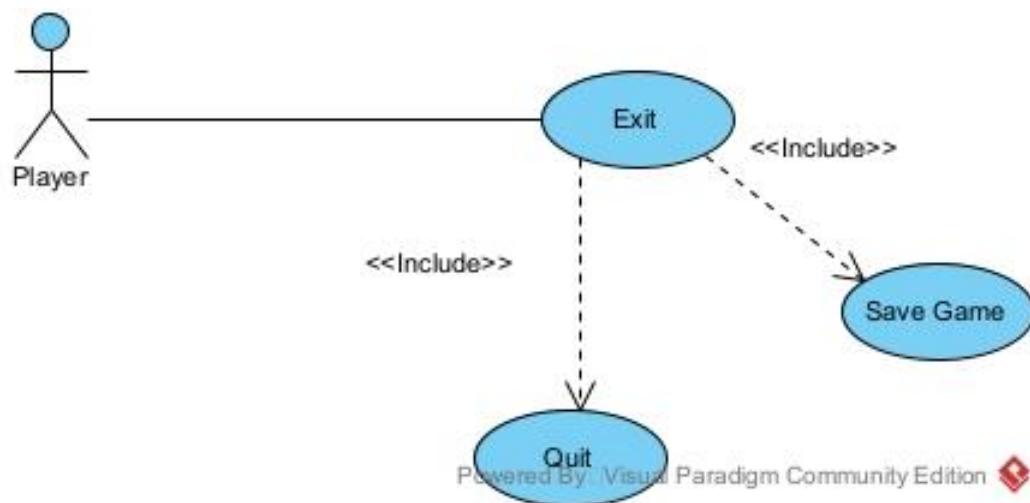


- Khởi tạo hệ thống
 - Tại giao diện chính, người dùng chọn chức năng "Xem Credit"
 - Hệ thống gọi CreditController để xử lý yêu cầu
 - CreditController khởi tạo scene hiển thị credit
- Tải dữ liệu
 - CreditController gọi CreditService.getCreditData()
 - CreditService truy vấn CSDL thông qua CreditRepository
 - Dữ liệu được đóng gói thành đối tượng CreditInfo
 - Danh sách thành viên trả về cho controller
- Thiết lập hiển thị
 - Hệ thống tải các tài nguyên:
 - Background
 - Font chữ
 - Hiệu ứng animation
 - Thiết lập timeline cho slideshow
- Hiển thị slideshow
- Vòng lặp hiển thị từng thành viên:
 - Áp dụng hiệu ứng trượt từ dưới lên
 - Hiển thị thông tin (tên, vai trò)
 - Tạm dừng 3s trước khi chuyển tiếp

- Xử lý tương tác
 - Sau khi hiển thị hết thành viên:
 - Hiển thị nút "Quay về"
 - Chờ người dùng click
- Khi click:
 - Hủy các tài nguyên đã tải
 - Trở về màn hình chính

1.2.4. Exit Game Module

a) Usecase diagram

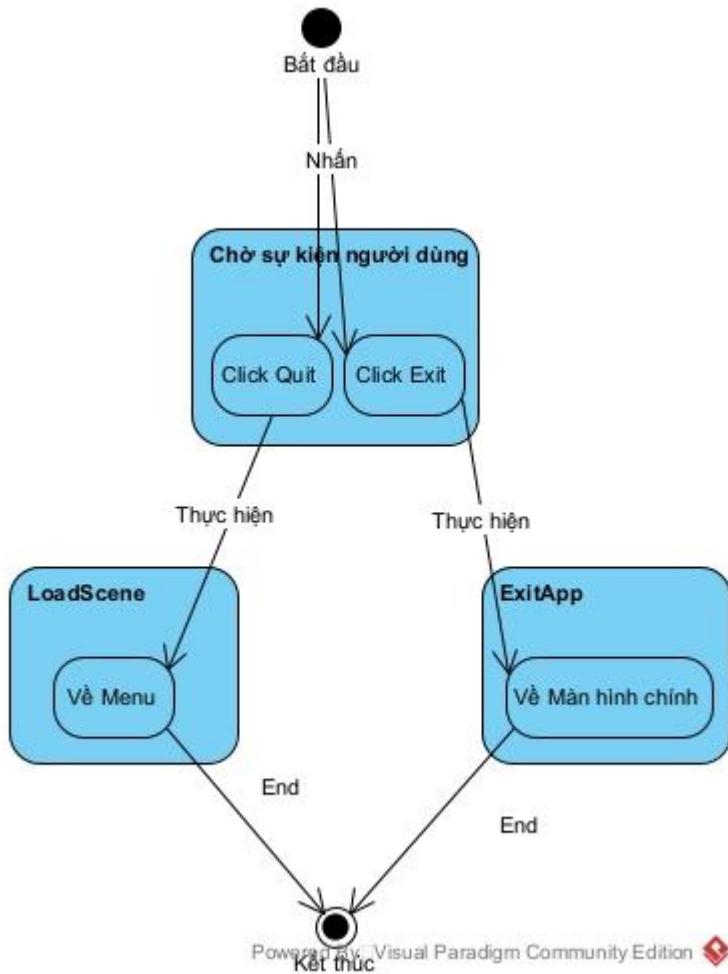


b) Kịch bản

| | |
|----------------|--|
| Scenario | Exit Game |
| Actor | Người chơi |
| Pre-Condition | Người chơi đã chọn "Exit" từ menu hoặc UI. |
| Post-Condition | Trò chơi đóng hoàn toàn. |
| Main event | 1. Hệ thống hiển thị hộp thoại xác nhận thoát. 2. Người chơi chọn "Yes" để thoát, "No" để hủy. 3. Nếu "Yes", hệ thống đóng game. |

| | |
|-----------|--|
| Exception | <p>2.1. Lỗi lưu dữ liệu khi thoát.</p> <p>2.2. Người chơi không thể hủy thoát.</p> |
|-----------|--|

c) Biểu đồ hoạt động

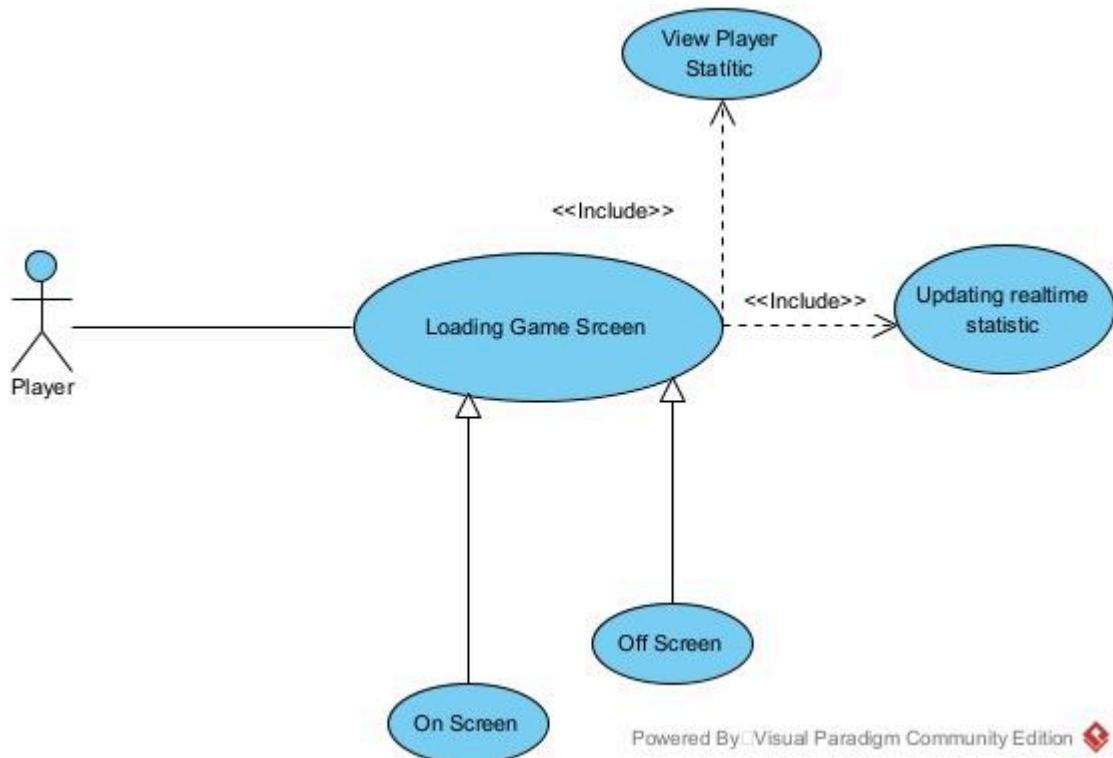


- Start → Chờ sự kiện người dùng
 - Môđun bắt đầu ở trạng thái chờ (Wait) cho đến khi có sự kiện nhấn nút từ người chơi.
- Decision Node
 - Ở trạng thái Wait, hệ thống kiểm tra hai loại sự kiện:
 - Người chơi bấm nút Quit trên Main Menu.
 - Người chơi chọn Exit trong System Menu khi đang chơi.

- Luồng “Quit” → Trả về Main Menu
- Khi phát hiện sự kiện Quit trên Main Menu, hệ thống gọi hành động SceneManager.LoadScene("MainMenu");
- Mục đích: đưa người chơi trở lại giao diện chính của trò chơi (Main Menu).
 - Luồng “Exit” → Thoát hoàn toàn ứng dụng
 - Khi phát hiện sự kiện Exit trong System Menu (trong gameplay), hệ thống gọi hành động Application.Quit();
- Mục đích: đóng hoàn toàn ứng dụng và thoát khỏi trò chơi.End
- Cả hai luồng sau khi thực thi hành động tương ứng đều kết thúc luồng hoạt động.

1.2.5.Gaming Screen Module

a) Usecase diagram

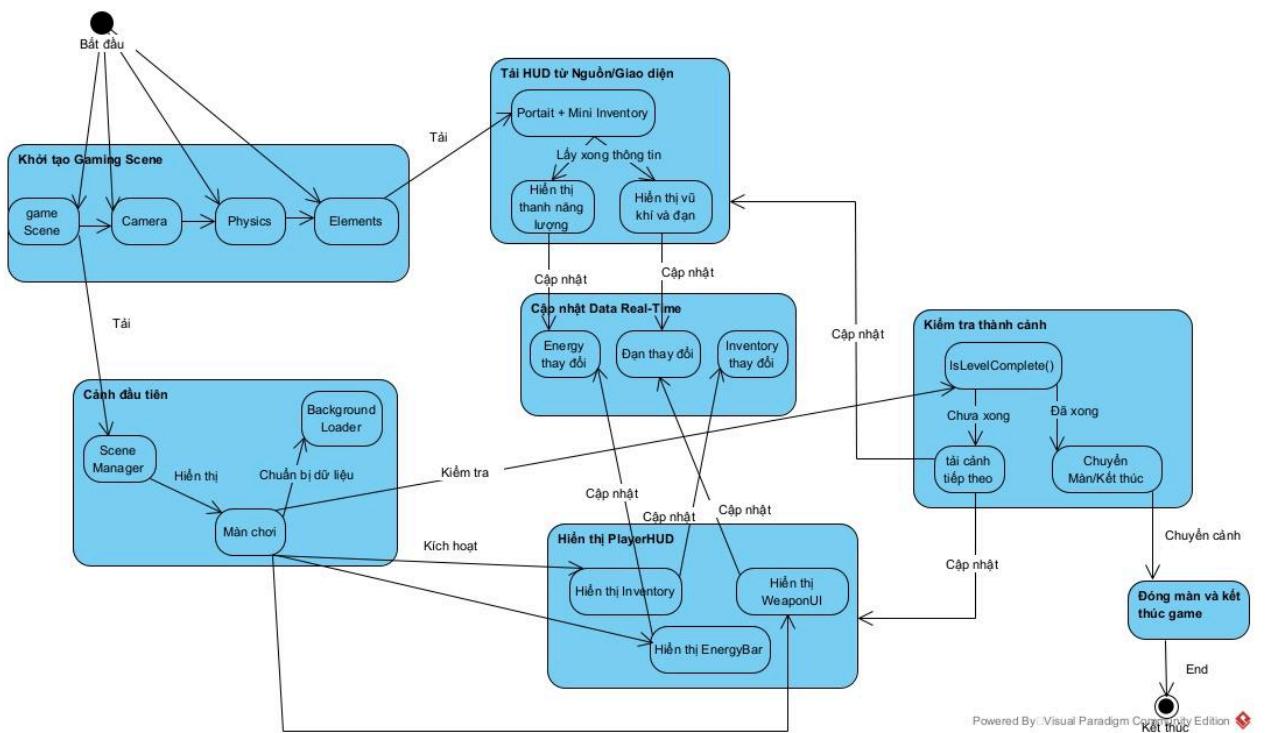


b) Kịch bản

| | |
|----------|----------------------|
| Scenario | Gaming Screen Module |
|----------|----------------------|

| | |
|----------------|---|
| Actor | Người chơi |
| Pre-Condition | Người chơi đã vào màn chơi từ Play Game Module. |
| Post-Condition | Người chơi hoàn thành hoặc thất bại trong màn chơi. |
| Main event | <p>1.Hiển thị giao diện chơi game: chân dung nhân vật, inventory, vũ khí.</p> <p>2.Người chơi điều khiển nhân vật di chuyển, bắn súng, dùng vật phẩm.</p> <p>3.Người chơi hạ gục kẻ địch, nhặt vật phẩm, sử dụng kỹ năng.</p> <p>4.Khi người chơi thất bại, màn hình "Game Over" xuất hiện.</p> <p>5.Khi người chơi thắng, màn hình "Victory" và điểm số hiển thị.</p> <p>6.Người chơi nhấn "M" để xem Map UI.</p> <p>7.Khi xong một màn chơi giao diện game sẽ lưu map cũ, tạm ẩn map cũ đi cho đến khi người chơi quay lại và load một map mới lên.</p> |
| Exception | <p>1.1.Lỗi hiển thị giao diện.</p> <p>2.1.Vũ khí không hoạt động.</p> <p>3.1.Kẻ địch không xuất hiện.</p> <p>6.1.Map UI không cập nhật vị trí.</p> |

c) Biểu đồ hoạt động:



- Khởi tạo Gaming Scene
 - Hệ thống tải scene chơi game, thiết lập camera, vật lý, và các thành phần cơ bản.
 - Gọi `GameManager.Init()` để chuẩn bị dữ liệu trò chơi.
- Tải HUD Elements
 - Load các thành phần HUD từ Resources/UI:
 - Portrait + Mini-Inventory (góc trái trên)
 - Energy Bar (thanh năng lượng)
 - Weapon UI (góc phải dưới, hiển thị vũ khí và đạn)
 - Áp dụng animation/phản hồi khi có thay đổi (ví dụ: Energy Bar giảm khi dùng skill).
- Load Cảnh Đầu Tiên
 - Sử dụng SceneManager để tải màn chơi đầu tiên (Level 1).

- Đồng thời, BackgroundLoader chuẩn bị các cảnh tiếp theo ở chế độ nền (background loading).
- Hiển Thị Player HUD
 - Các thành phần HUD được kích hoạt:
 - Mini-Inventory hiển thị vật phẩm nhanh (sử dụng InventoryManager).
 - Energy Bar cập nhật theo giá trị Player.Energy.
 - Weapon UI hiển thị icon vũ khí (WeaponManager.GetCurrentWeapon()) và số đạn (AmmoCounter).
- Cập Nhật Real-Time Data
 - Vòng lặp game liên tục kiểm tra:
 - Năng lượng (Energy) thay đổi → Cập nhật Energy Bar.
 - Đạn thay đổi → Cập nhật Weapon UI.
 - Vật phẩm thay đổi → Cập nhật Mini-Inventory.
- Kiểm Tra Hoàn Thành Cảnh
 - Nếu người chơi hoàn thành màn hiện tại (LevelManager.IsLevelComplete()):
 - Load cảnh tiếp theo (nếu có).
 - Cập nhật HUD cho màn mới (ví dụ: reset Energy Bar nếu có luật game yêu cầu).
 - Nếu là màn cuối → Chuyển sang End Game Screen.
- Kết Thúc
 - Khi người chơi thoát game:
 - Trở về menu chính.

2.Thiết kế tổng quan

- Đây sẽ là một game thể loại top-down shooter lấy cảm hứng từ các game tiêu biểu như Hotline Miami. Thể nhưng thay vì việc sử dụng đồ họa góc nhìn top-down bird view 180 độ truyền thống thì dự án sẽ đi theo hướng top down isometric 45 độ.
- Bạn tỉnh dậy ở một nơi xa lạ, không có bất cứ ký ức gì về bản thân. Một người lạ chở đến chỗ bạn, tự xưng là..., nói rằng bạn là kẻ được chọn để giải cứu thế giới. Từ giây phút đấy, bạn đã bị cuốn vào một cuộc chiến mà chính bản thân chính là người định đoạt, là trung tâm.
- Mục tiêu của người chơi là tiêu diệt toàn bộ kẻ địch, điểm số sẽ được tính sau khi màn chơi kết thúc, tỉ lệ thuận với số địch hạ được trong

thời gian ngắn, thời gian hoàn thành màn chơi và tỉ lệ nghịch với số lần thất bại.

- Người chơi sẽ phải vào màn với tay không, phải sử dụng các vũ khí được đặt rải rác trên màn chơi, có thể sử dụng cả các vật phẩm sử dụng nhanh.

3.Thiết kế màn chơi

- Dự án sẽ có 3 màn chơi chính, 1 màn chơi hướng dẫn và 1 màn chơi chế độ đặc biệt, các màn chơi với các cảnh chơi sẽ giới thiệu các cơ chế đặc biệt như màn chơi đêm, thoát khỏi khu vực trong thời gian nhất định,... cùng các âm thanh tương ứng.
- Màn chơi hướng dẫn sẽ giúp người chơi hiểu chi tiết các cơ chế của trò chơi cũng như là một phòng tập luyện để làm quen với các cơ chế điều khiển, chuẩn bị cho các màn tiếp theo
- Dự án chia ra mỗi màn chơi sẽ có 3 cảnh, người chơi sẽ phải đánh bại tất cả kẻ địch của các cảnh để chuyển tiếp màn, thứ tự chơi các màn sẽ theo trình tự tuyến tính.

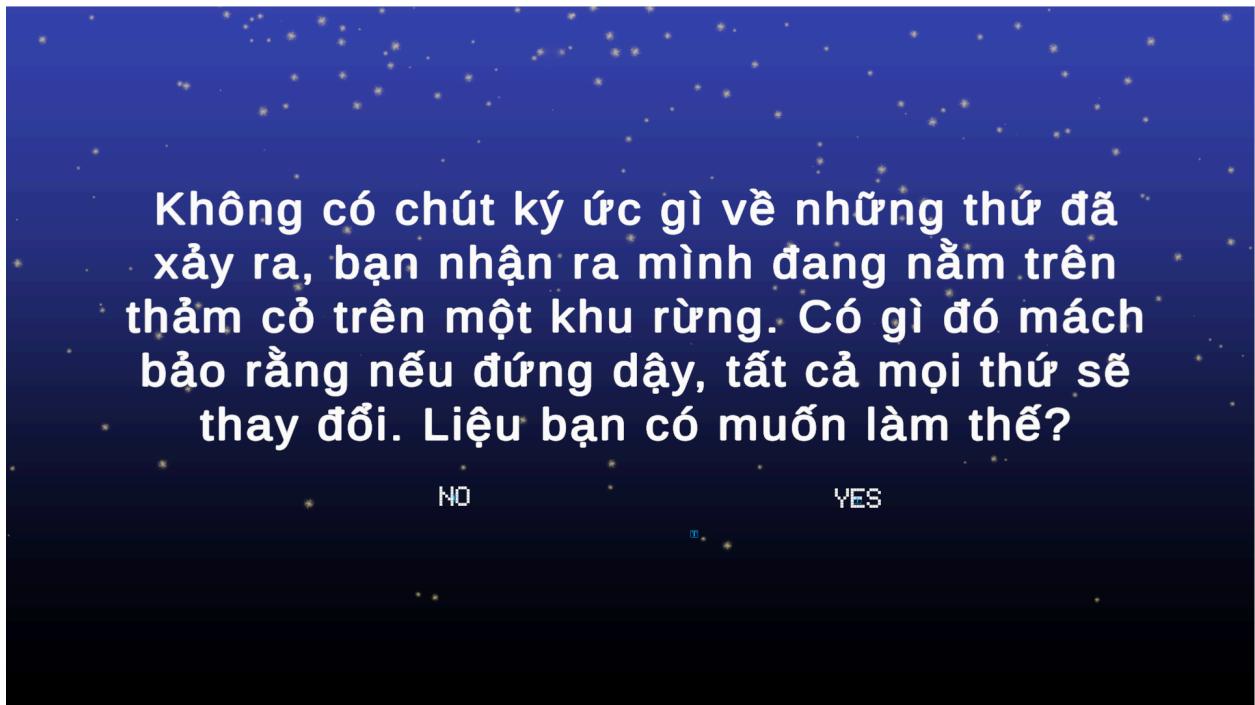
4.Thiết kế chi tiết các màn chơi

4.1.Thiết kế Main Menu



- Hiệu ứng nền:
 - Thêm hiệu ứng parallax background để tạo chiều sâu.
 - Thêm Particle System (e.g. tuyết rơi, mưa, hoặc hiệu ứng ánh sáng mờ nhẹ).
- Hiệu ứng chuyển cảnh:
 - Khi nhấn Play, thêm hiệu ứng fade-out trước khi vào game.
 - Khi vào Credits, có hiệu ứng trượt text từ dưới lên.
- Animation cho nút bấm:
 - Khi hover vào nút, thay đổi màu sắc hoặc phóng to nhẹ bằng Animation hoặc LeanTween.
 - Khi bấm vào nút, có thể thêm hiệu ứng rung nhẹ, âm thanh click và hiện màu.
- ★ Hiệu ứng tương tự với 2 nút còn lại

4.2.Thiết kế Intro



- Thiết kế intro theo kiểu slideshow chữ, màn hình sẽ có 1 dòng chữ xuất hiện để dẫn vào game.
 - Intro: "Không có chút ký ức gì về những thứ đã xảy ra, bạn nhận ra mình đang nằm trên thảm cỏ trên một khu rừng. Có gì đó mách bảo rằng nếu đứng dậy, tất cả mọi thứ sẽ thay đổi. Liệu bạn có muốn làm thế?"
 - Sẽ có 2 nút ở 2 góc dưới trái phải cho lựa chọn của bạn, chọn "có" sẽ đưa người chơi vào màn chơi còn chọn "không" đưa về MainMenu.
 - Thêm font LiberationSans SDF cho background qua inspector ô font asset.
 - Thêm image cho button và background: Skybox-default qua inspector ô image.

4.3.Thiết kế CreditScene



- CreditScene sẽ là 1 SlideShow chữ với đầy đủ thông tin làm việc của các thành viên trong dự án, các dòng chữ sẽ chạy từ dưới lên từ từ.
- Add material, thiết kế image cho background, button và fix code Blacking 2:
- Add material Electronic Highway Sign SDB, chỉnh color cho button và background qua inspector trong mục color và normal color chỉnh A về bên trái hết mức có thể để làm tàng hình màu trong mục color khi vào game khi di chuột vào ô button màu sẽ hiện lại.
 - Fix: Bổ sung Event System “Button” cho background để khi bấm vào sẽ lùi về Trang Main Menu.
 - Thiết kế giống với Button của phần Intro.
- Add thêm Credit, tác giả:
 - “Tác phẩm này được tạo ra bởi Nguyễn Nhật Hưng và Trần Đức Anh” như hình trên, sẽ bổ sung thêm.

4.4.Thiết kế Tutorial Stage



- Thiết kế maptile, tạo và sắp xếp các layer theo cam và character có sẵn:
- Tạo lớp ground và player set ground layer 1 và player layer 6 để character không bị map che đi.
- Tạo thêm object grid trong đó có ground, add thêm component tilemap vào.
 - Sorting layers add thêm layer ground cho phần ground, set character layer player .
- Import Tiny RPG Forest vào asset's game sau đó tìm tilemap trong folder environment và slice thành 16x16 trong sprite editor.
- Kéo thả tilemap đã slice vào Tile Palette, tạo folder tilemap.
- Thiết kế map tutorial cơ bản bằng kéo thả các ô 16x16 đã slice vào map:
 - Gồm có thác nước, hồ nước, lùm cây, vách hang động, add thêm các bụi cỏ thiết kế đường đi giữa các khu vực.
- Thiết kế và mở rộng chi tiết cho Tutorial Stage:
 - Đặt các đối tượng có Collider:
 - Đối tượng cố định: Sử dụng các asset đã có hoặc tự tạo các đối tượng như cây, đá, cầu, cổng...

- Mục đích: Tạo cảm giác thực tế cho môi trường cũng như tăng độ thử thách bằng cách cản trở đường đi và tạo các khu vực chiến thuật.
 - Kỹ thuật: Gán component Collider (BoxCollider2D, PolygonCollider2D...) cho từng đối tượng và thiết lập các layer phù hợp.
- Trang trí chi tiết:
 - Các đối tượng trang trí: Sử dụng asset cho hồ sông, cột cổng, hàng rào, đảo ếch, bọt khí... cùng với các chi tiết nhỏ như hoa và bụi lùm cây để tạo không gian sống động.
 - Layering: Sau khi bố trí các đối tượng, phân chia thành các layer riêng biệt.
 - Ground (mặt đất), Water (nước), Floor (sàn nhà), Rock (đá), Tree (cây), Player (nhân vật), UI (giao diện người dùng).
 - Mục đích: Quản lý thứ tự hiển thị và xử lý va chạm một cách hợp lý.
- Thiết kế map Khu rừng giới thiệu
 - Bối cảnh:
 - Sử dụng Tiny RPG - Forest by Ansimuz để thiết kế toàn bộ bối cảnh khu rừng xanh tươi, bao gồm cây cối, tảng lá, cầu, và cổng.
 - Thêm các chi tiết trang trí như hồ nước, bọt khí, hoa và bụi lùm cây để tạo cảm giác sống động.
 - Nhân vật và NPC:
 - Cute 2D - College Student by ClearSky Studio sẽ làm NPC hướng dẫn.
 - Nhiệm vụ:
 - Hướng dẫn người chơi các kỹ năng cơ bản như di chuyển, nhặt đồ, mở rương vật phẩm, và cơ chế dùng súng.
 - Dialogue:
 - Giới thiệu về cách thu thập và sử dụng vật phẩm.
 - Hướng dẫn cơ chế mở rương và giới thiệu các loại súng cùng cách sử dụng.
 - Hiệu ứng và âm thanh:
 - Footstep_Grass.mp3 cho bước chân trên cỏ.
 - Forest_Ambience.mp3 cho nhạc nền khu rừng.
 - Bush_Rustle.mp3 khi đi qua bụi lùm.

- Chuyển đổi giữa các khu vực:
 - Sử dụng Waypoint và Collider2D để tạo điểm dịch chuyển giữa các khu vực trong map.
- Chuyển map và âm thanh:
 - Waypoint: Đặt tại cổng ra của khu rừng.
 - Script Chuyển Map:
 - Sử dụng Collider2D (ở chế độ Trigger) để phát hiện khi nhân vật đi qua.
 - Kéo targetWaypoint đến điểm đầu vào của Map 2.
 - AudioSource: Phát Forest_Ambience.mp3.
- Script Quản Lý Âm Thanh:
 - Dùng một script quản lý duy nhất với các hàm chuyển đổi nhạc nền dựa trên sự kiện chuyển map.
 - Áp dụng hiệu ứng fade in/out để đảm bảo trải nghiệm liền mạch.
- Thiết kế map Chiến Đấu và Giải Cứu
 - Bối cảnh:
 - Sử dụng Zombie Apocalypse Tileset by Ittai Manero để thiết kế khu vực bị tàn phá, tạo cảm giác căng thẳng và nguy hiểm.
 - Nhân vật và Quái Vật:
 - Evil Zombie được sử dụng để thiết kế quái vật.
 - Vũ Nguyễn Linh là NPC cần được giải cứu.
 - Nhiệm vụ:
 - Giải cứu NPC bằng cách đánh bại quái vật.
 - Thu thập vật phẩm và mở khóa khu vực mới.
 - Dialogue:
 - Khi vừa bước vào map, xuất hiện thông báo về nhiệm vụ bảo vệ và giải cứu NPC.
 - Trước khi đánh quái, xuất hiện hướng dẫn về cách đánh bại quái, chỉ ra điểm yếu hoặc cơ chế đặc biệt.
 - Hiệu ứng và âm thanh:
 - Footstep_Rock.mp3 cho bối cảnh bê tông.
 - Gate_Lock.mp3 khi cửa bị khóa.
 - Rock_Move.mp3 khi đẩy đá hoặc vật cản.
 - Chest_Open.mp3 khi mở rương.
 - Crickets_Night.mp3 tạo không khí căng thẳng.
 - Chuyển đổi giữa các khu vực:

- Sử dụng Waypoint và Collider2D để chuyển đổi giữa các khu vực trong map và quay lại Map 1 nếu cần.
- Chuyển map và âm thanh:
 - Waypoint: Đặt tại lối ra của khu vực chiến đấu.
 - Script Chuyển Map:
 - Dùng Collider2D (ở chế độ Trigger) để phát hiện và dịch chuyển về Map 1 hoặc Map 3.
 - Kéo targetWaypoint đến waypoint của khu vực cần đến.
 - AudioSource: Phát Crickets_Night.mp3 với hiệu ứng fade in/out để mượt mà.

4.5.Thiết kế tổng quan các màn chơi:

> Màn 1 - Thành phố hoang tàn

- Bối cảnh màn chơi:
 - Sau màn giới thiệu, người chơi bước vào một thành phố hoang tàn dưới bầu trời âm u, nơi hỗn loạn và bí ẩn đang lan tràn cùng với cô gái mình vừa cứu.
 - Thành phố này từng là trung tâm của một nền văn minh phát triển, nhưng bây giờ chỉ còn lại những tàn tích của sự thịnh vượng đã qua.
 - Các khu phố bị bỏ hoang, tòa nhà đổ nát và những con đường vắng tanh tạo nên không gian đầy nguy hiểm.
 - Dưới lòng đất, một hệ thống đường hầm và căn cứ ngầm chứa đựng những bí mật kinh hoàng của một tổ chức tàn bạo, chính là nguồn gốc của sự hỗn loạn hiện nay.
- Nhân vật và quái vật:
 - Nhân vật chính: Trần Tuấn Hoàng.
 - Nhân vật phụ: Vũ Nguyễn Linh.
 - Quái: một loại quái vật Evil zombie và 2 loại quái đặc biệt là Evil wizard và Brute zombie.
- Các khu vực chính:
 - Khu dân cư bị bỏ hoang: Nhà cửa đổ nát, xe cộ cháy, rác vương vãi.
 - Phố chính: Rộng hơn, nhưng nhiều xe chắn đường, có thể di chuyển vòng vèo.
 - Trạm điện cũ: Một điểm phụ có thể bật nguồn cho một cánh cổng.

- Khu vực kho hàng: Nơi xuất hiện Brute zombie đầu tiên.
 - Cửa hầm dẫn xuống căn cứ ngầm: Phần kết màn chơi.
- Tương tác bản đồ:
 - Vật cản có thể phá vỡ (thùng gỗ, xe bẩn hoang)
 - Cửa tự động (mở khi bật nguồn)
 - Các bảng điện hoặc chìa khóa cần thu thập để mở cổng phụ.
- NPC nữ (Vũ Nguyễn Linh) được thiết kế lại với vai trò “trợ giúp tạm thời”: sau khi hoàn thành nhiệm vụ hỗ trợ ban đầu, cô sẽ rời đi khỏi chiến trường, tạo hiệu ứng bất ngờ và tăng tính chiến lược cho game.
- Nhiệm vụ chính:
 - Bắt đầu: Người chơi, với sự hỗ trợ ban đầu của Vũ Nguyễn Linh, bước vào thành phố hoang tàn qua một con hẻm và thời gian bối cảnh thành phố chuyển vào buổi đêm.
 - Kích hoạt trạm điện: Người chơi và Vũ Nguyễn Linh cùng nhau kích hoạt trạm điện để mở cửa khu kho hàng, mở ra cơ hội đối đầu với Brute Zombie.
 - Đối đầu quái đặc biệt: Trong khu kho hàng, Brute Zombie xuất hiện. Một cú bắn chính xác có thể tiêu diệt nó, nhưng cũng cần tránh bị phản công.
 - Cuộc giao tranh với Evil Wizard: Cuối màn, sau khi nhận được thông tin từ NPC nữ, Trần Tuấn Hoàng tiến hành truy đuổi và tiêu diệt Evil Wizard, phát hiện ra lối vào hệ thống đường hầm dưới lòng đất.
 - Kết thúc màn: Sau khi hoàn thành nhiệm vụ, Vũ Nguyễn Linh thực hiện báo tin nhiệm vụ hoàn thành qua giao tiếp (đoạn cắt cảnh ngắn) và rời đi, để lại Trần Tuấn Hoàng tự mình đối mặt với phần còn lại của nhiệm vụ.

➤ Màn 2 - Hầm Ngầm ám ảnh

- Bối cảnh màn chơi:
 - Sau khi Trần Tuấn Hoàng vượt qua thành phố hoang tàn và đánh bại Evil Wizard ở màn 1, anh khám phá ra lối vào hệ thống đường hầm dưới lòng đất. Màn 2 đưa người chơi vào môi trường ngầm u ám, với ánh sáng mờ ảo, bóng tối dày đặc xen lẫn những ánh đèn pin yếu ớt. Tiếng nước nhỏ giọt, tiếng cơ khí cũ kỹ vang vọng tạo nên không khí lạnh lẽo, rùng rợn nhưng cũng đầy bí ẩn.
- Nhân vật & Quái vật:

- Nhân vật chính: Trần Tuấn Hoàng tiếp tục nhiệm vụ, giờ đây phải đối mặt với môi trường ngầm phức tạp và hệ thống quái vật nguy hiểm.
- Nhân vật phụ: Mặc dù Vũ Nguyễn Linh không xuất hiện trực tiếp, cô để lại các thông tin hỗ trợ qua các tin nhắn hoặc hộp tin giấu kín khắp hệ thống hầm, giúp người chơi thu thập manh mối giải đố.
- Quái vật:
 - Evil zombie
 - Assassin
 - Hunting Dog
- Các khu vực chính:
 - Lối vào hầm ngầm: Một hành lang dài, hẹp với ánh sáng yếu đòi hỏi người chơi phải cẩn thận tránh bẫy tự động, đồng thời đối mặt với đợt tấn công nhanh chóng của Evil Zombie và Hunting Dog.
 - Phòng lưu trữ bỏ hoang: Khu vực này chứa các vật phẩm hữu ích và tài liệu về tổ chức tàn bạo. Người chơi cần tìm kiếm chìa khóa, mã số hoặc các mảnh dữ liệu để mở khóa các phần cần thiết của hệ thống an ninh.
 - Hành lang rỗng rã: Đây là nơi Assassin hoạt động chủ yếu. Với thiết kế nhiều ngã, lối đi xen kẽ và ánh sáng mờ ảo, những hiệu ứng ngụy trang của Assassin khiến người chơi cảm thấy áp lực và phải luôn cảnh giác trước bất cứ góc khuất nào.
 - Phòng kiểm soát an ninh: Thay vì đụng độ với một boss, khu vực này tập trung vào giải đố tương tác. Người chơi cần kích hoạt hệ thống an ninh của căn cứ bằng cách thu thập dữ liệu từ các khu vực trên và ghép nối thông tin để mở khóa cánh cửa dẫn ra khỏi hầm ngầm.
- Tương tác bản đồ:
 - Thu thập các mảnh dữ liệu từ phòng lưu trữ và hành lang phụ, đồng thời khai thác các tín hiệu hỗ trợ từ tin nhắn của Vũ Nguyễn Linh để hoàn thành các bảng điều khiển trong phòng kiểm soát.
 - Các cửa ra và hành lang bị khóa bằng cơ chế mã số hoặc chìa khóa điện tử yêu cầu người chơi phải tìm kiếm và giải mã, kết hợp khéo léo giữa hành động và suy luận.

- Bẫy tự động và các đoạn đường hầm khó lường đòi hỏi người chơi phải di chuyển một cách chiến thuật, sử dụng các nguồn sáng và đường tầu để tránh đụng độ không cần thiết.
- Nhiệm vụ chính:
 - Khám phá đường hầm: Bắt đầu từ lối vào hầm, đối mặt với sự xuất hiện liên tục của Evil Zombie, Hunting Dog và những cuộc đụng độ bất ngờ của Assassin.
 - Thu thập manh mối: Tìm kiếm các mảnh dữ liệu và vật phẩm hữu ích, kết hợp với các tín hiệu hỗ trợ từ Vũ Nguyễn Linh để mở khóa hành lang và phòng kiểm soát.
 - Giải mã hệ thống an ninh: Người chơi phải ghép nối các manh mối thu thập được để kích hoạt bảng điều khiển chính trong phòng kiểm soát, mở ra cánh cửa dẫn ra khỏi hệ thống hầm.
 - Quản lý đợt tấn công: đối phó với các nhóm Evil Zombie và Hunting Dog, đồng thời luôn cảnh giác để phát hiện ra Assassin đang ẩn mình trong bóng tối, từ đó sử dụng chiến thuật linh hoạt để đối phó với mọi tình huống.
 - Thông điệp bí ẩn: Các tin nhắn và tài liệu thu thập trong hầm ngầm hé lộ thêm về tổ chức tàn bạo đứng sau sự hỗn loạn, mời gọi người chơi tiến sâu vào bí ẩn của hệ thống dưới lòng đất.
 - Sự mở rộng nhiệm vụ: Sau khi kích hoạt thành công hệ thống an ninh và mở khóa cánh cửa dẫn ra, người chơi sẽ tiếp tục hành trình đến các khu vực mới với những bí mật, thử thách cao hơn và mối đe dọa đang âm ỉ từ tổ chức tàn bạo.

➤ Màn 3 - Cuộc rượt đuổi của các đám mây

- Bối cảnh màn chơi:

- Mở ra một không gian đặc biệt khác hẳn so với các màn trước, nơi bầu trời u ám chứa đầy những đám mây kinh dị. Thay vì đối đầu trực tiếp với kẻ địch hạ gục được, người chơi phải liên tục di chuyển để né tránh những đám mây truy đuổi, vì chúng không thể bị tiêu diệt thông qua các cuộc tấn công thông thường. Mỗi đe dọa chủ yếu đến từ các đám mây này sẽ chỉ giảm bớt khi Cloud Core – nguồn điều khiển bất tử của chúng – bị hạ gục.

- Nhân vật và kẻ địch:
 - Nhân vật chính: Trần Tuấn Hoàng, giờ đây phải đối mặt với thử thách chạy trốn liên tục và điều chỉnh chiến thuật để tiếp cận mục tiêu duy nhất có thể làm giảm sức mạnh của đám mây.
 - Kẻ địch:
 - Evil Zombie: Xuất hiện dưới nền của màn chơi, chúng vẫn là mối đe dọa theo script chung, hỗ trợ tạo thêm sự căng thẳng trong khi người chơi tập trung tránh né đám mây trên cao.
 - Cloud: Những đám mây hình dạng mờ ảo, liên tục rình rập và đuổi theo người chơi. Chúng hoàn toàn bất tử khi bị tấn công, do đó, việc tiếp xúc chỉ làm chậm tiến trình của người chơi hoặc gây sát thương theo thời gian.
 - Cloud Core: Vị trí chiến lược của màn chơi. Cloud Core đứng yên tại một điểm cố định, là nguồn điều khiển của các đám mây. Người chơi chỉ có thể giảm bớt sức ép từ đám mây bằng cách tiếp cận và tấn công vào Cloud Core, khiến một số lượng Cloud nhất định mất đi khả năng truy đuổi.
- Các khu vực chính:
 - Vùng ngoài trời mở: Là nơi người chơi sẽ trải nghiệm cuộc rượt đuổi của các đám mây. Những đám Cloud xuất hiện từ các góc trên cao liên tục rình rập và di chuyển theo những quỹ đạo có phần khó lường, buộc người chơi phải di chuyển liên tục để tránh va chạm.
 - Khu vực che chắn: Các mảnh đổ nát, tòa nhà sụp đổ và những điểm ẩn nấp được bố trí xung quanh cung cấp những khoảnh khắc an toàn tạm thời, nhưng cũng tạo ra những mê cung khiến lộ trình tiếp cận Cloud Core trở nên thử thách hơn.
 - Khu vực Cloud Core: Nằm ở vị trí trung tâm hoặc khu vực được bảo vệ cao, nơi đặt Cloud Core – mục tiêu duy nhất có thể làm yếu đi đội hình đám mây. Người chơi cần sử dụng những lối đi chiến thuật qua các khu vực che chắn để tiếp cận và tấn công Cloud Core.
- Tương tác bản đồ:

- Cơ chế rượt đuổi và né tránh:

- Các đám Cloud liên tục xuất hiện và đuổi theo người chơi từ phía trên, tạo ra áp lực liên tục. Do không thể tiêu diệt trực tiếp, người chơi cần khai thác tốc độ và tính linh hoạt trong việc chạy trốn.
- Va chạm với Cloud sẽ không gây ra “kill score” mà chỉ làm giảm tốc độ di chuyển hoặc gây sát thương qua thời gian, tạo thêm thử thách về việc quản lý nguồn lực và thời gian an toàn.
- Cloud Core là mục tiêu duy nhất có thể làm giảm số lượng đám mây truy đuổi. Tuy nhiên, việc tiếp cận nó đòi hỏi phải vượt qua làn đám mây liên tục rình rập, kết hợp với sự hiện diện của Evil Zombie.
- Người chơi có thể sử dụng các điểm che chắn trong môi trường để lẩn qua và tấn công Cloud Core. Một khi bị hạ gục, một phần hoặc toàn bộ các đám Cloud xung quanh sẽ mất khả năng truy đuổi, tạo ra khoảng thời gian an toàn để tiếp tục di chuyển.
- Ngoài việc tránh né, người chơi cũng phải đối phó với nhóm Evil Zombie xuất hiện ở mặt đất, tạo nên một sự kết hợp thách thức giữa việc phòng thủ và tấn công.
- Người chơi cần tính toán kỹ lưỡng giữa tốc độ di chuyển, thời gian “safe zone” khi ẩn nấp và thời điểm mở đường tấn công chính xác vào Cloud Core.

- Nhiệm vụ chính:

- Tránh né đám Cloud: Di chuyển liên tục để tránh bị dính các đám mây bất tử, trong khi vẫn phải chú ý đến sự xuất hiện của Evil Zombie từ những góc khuất.
- Tiếp cận Cloud Core: Tận dụng các khu vực che chắn để lẩn qua và tiếp cận Cloud Core mà không bị liên tục truy đuổi bởi các đám mây.
- Tấn công Cloud Core: Tập trung phá hủy Cloud Core – mục tiêu chiến lược duy nhất để làm giảm sức tấn công của các đám Cloud, từ đó mở ra khoảng thời gian an toàn cho việc di chuyển và hoàn thành nhiệm vụ.

- Mở khóa lộ trình tiếp theo: Sau khi một số đám mây bị vô hiệu hóa nhờ Cloud Core, người chơi cần kích hoạt các cơ chế an ninh hoặc các nút điều khiển trong khu vực để mở đường sáng màn chơi tiếp theo, hé lộ thêm bí ẩn về tổ chức đứng sau hỗn loạn.
- Qua màn 3, Trần Tuấn Hoàng khám phá ra phương thức điều khiển của đám mây và nhận ra rằng nguồn gốc của sự hỗn loạn không chỉ đến từ đội quân Evil Zombie mà còn liên quan mật thiết đến một công nghệ hoặc thế lực siêu nhiên đứng sau Cloud Core.
- Thông tin thu thập được sau khi làm yếu đi lực lượng đám mây sẽ mở ra những mối liên hệ mới, hé lộ dần dần về bối cảnh rộng lớn hơn của tổ chức tàn bạo và đặt nền móng cho các thử thách khó khăn ở màn chơi sau.

➤ Màn đặc biệt - Khu tập luyện mô phỏng

- Bối cảnh màn chơi:
 - Đầu đó trong khoảng thời gian làm nhiệm vụ, bạn được giới thiệu tới một cơ sở huấn luyện nhằm nâng cao kỹ năng, thoát nhìn giống một nơi huấn luyện bình thường, cơ sở này lại sở hữu một căn phòng mô phỏng với khả năng tạo ra các nhiệm vụ giả cùng với các môi trường được hệ thống máy tính tân tiến tính toán và thay đổi liên tục, bạn sẽ được sử dụng căn phòng này để luyện các kỹ năng của mình tùy ý.
- Nhân vật và kẻ địch:
 - Nhân vật chính: Trần Tuấn Hoàng, sẽ phải thích ứng và làm các nhiệm vụ giả được máy tính giao cho cùng với môi trường thay đổi.
 - Kẻ địch:
 - Tất cả các kẻ địch được giới thiệu từ các màn.
- Các khu vực chính:
 - Các khu vực cùng với các điểm sẽ được tái tạo ngẫu nhiên trong một môi trường mở, các bức tường để người chơi vận dụng sẽ xoay theo các hướng khác nhau mỗi khi màn đấu thay đổi, bối cảnh của màn cũng sẽ thay đổi sau một vài màn chơi
- Tương tác bản đồ:
 - Cơ chế:

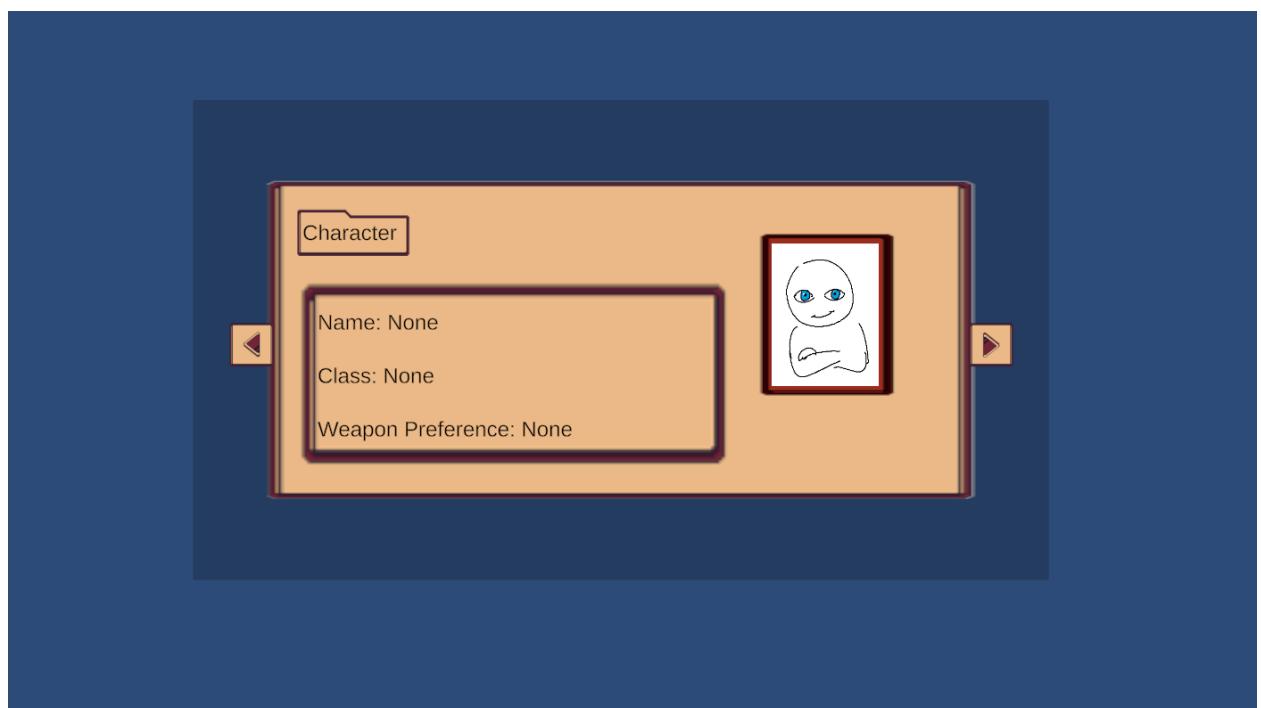
- Tận dụng tất cả các cơ chế từ các màn đã qua
- Sau khi kết thúc màn chơi có thể ấn nút tiếp tục từ màn hình tính điểm
- Thoát màn chơi bằng cách thoát ra từ giao diện System trong Menu
- Nhiệm vụ chính:
 - Diệt toàn bộ kẻ địch trong màn

5. Thiết kế giao diện

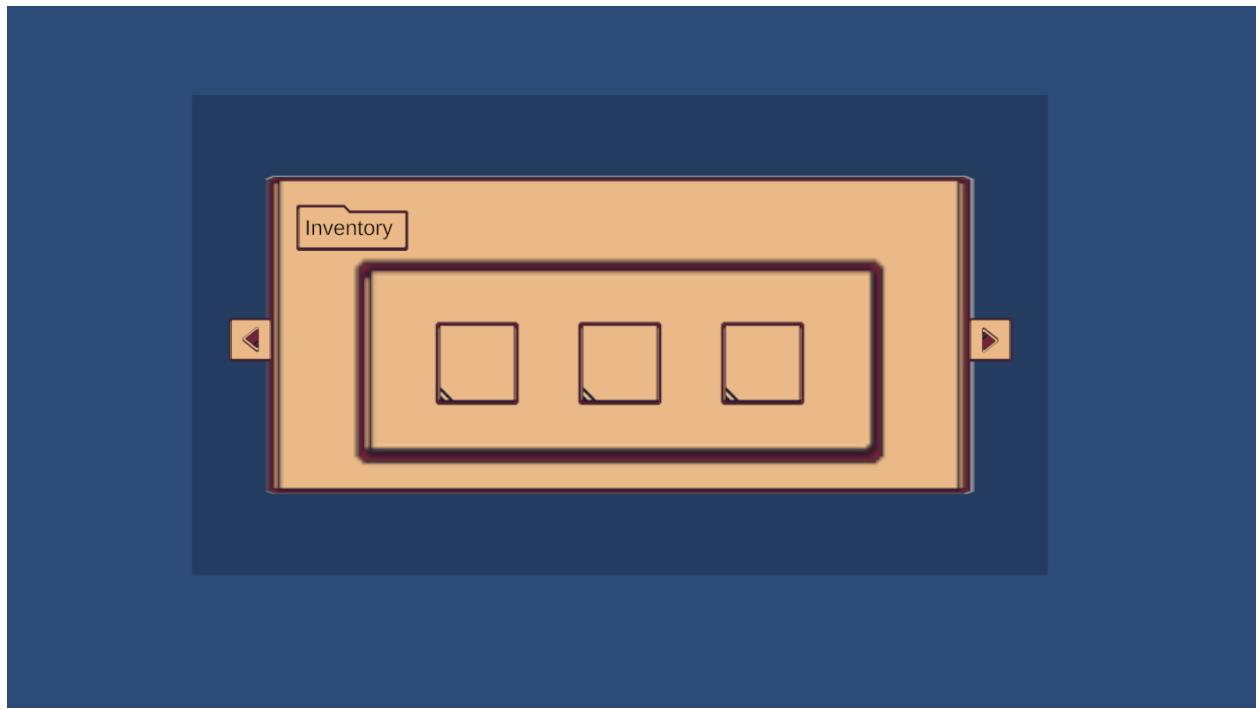


- Thiết kế giao diện hiển thị thời gian thực:
 - Góc trên bên trái sẽ là chân dung nhân vật với thanh Inventory dạng mini để người chơi có thể nhìn thấy vật phẩm mà không phải dừng game để vào giao diện người dùng để xem.
 - Bên dưới thanh Inventory mini sẽ là thanh năng lượng, khi thanh đầy có thể dùng khả năng đặc biệt của nhân vật được chọn, thanh năng lượng sẽ hồi lại sau khi dùng.
 - Góc dưới cùng bên phải sẽ là giao diện vũ khí với ảnh của vũ khí đang được sử dụng và lượng đạn còn lại.

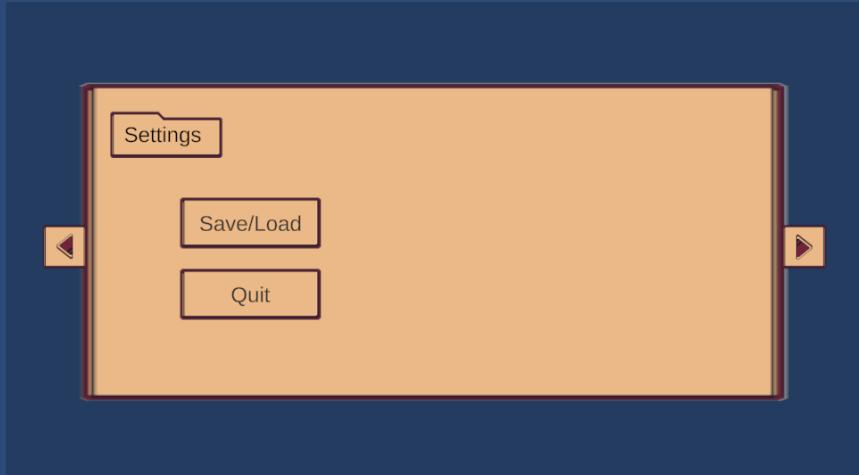
- Giao diện hiển thị sẽ biến mất khi giao diện chính được bật và ngược lại.
- Thiết kế giao diện người dùng bao gồm giao diện chính:
 - Giao diện chính hiện tại bao gồm 3 tab là Character Sheet, Inventory và System có thể chuyển đổi linh hoạt qua việc bấm phím mũi tên hiển thị kèm theo.



- Character Sheet dành cho việc hiển thị thông tin nhân vật như chân dung(portrait), tên(name), trường phái(class) và vũ khí ưa thích(prefer weaponry).

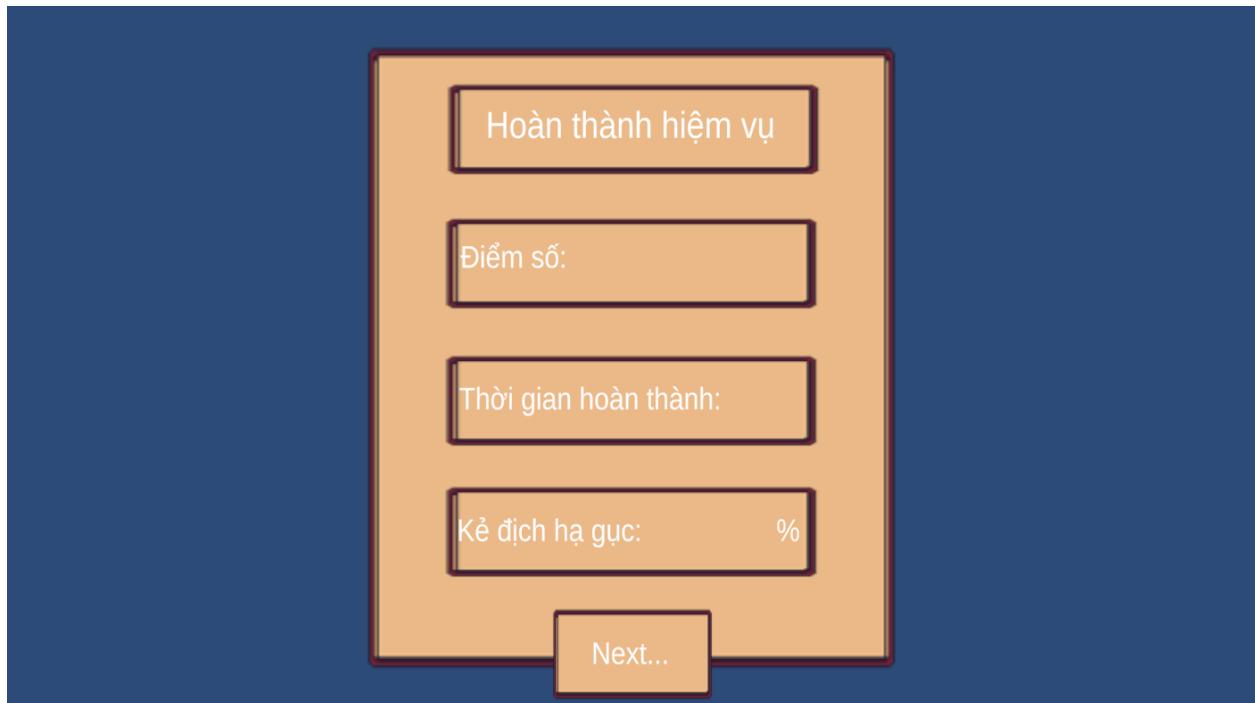


- Inventory sẽ là ô đồ của nhân vật, ấn 1 2 3 để dùng ô thứ 1 2 3 tương ứng, sau khi sử dụng thì vật phẩm trong ô biến mất.



- System sẽ là nơi chứa các nút để lưu/tải dữ liệu và nút thoát, có thể chọn giữa nút thoát về giao diện đầu tiên hoặc thoát thẳng ra khỏi game.
- Trong game sử dụng phím “Tab” để vào/ra khỏi giao diện người dùng.

6.Cài đặt màn hình tính toán thông số



- Màn hình thông số sẽ xuất hiện sau khi hoàn thiện màn chơi.
- Tính toán bằng cách cộng điểm theo số % kẻ địch đã hạ gục, thời gian hoàn thành màn.
- Sẽ bị trừ điểm dựa vào số lần người chơi bị hạ gục
- Sau khi điểm số được tính toán, người chơi có thể ấn nút xuất hiện ở góc phải bên dưới màn hình để chuyển tiếp.

7.Cài đặt màn hình chọn nhân vật.



- Màn hình chọn nhân vật sẽ xuất hiện trước màn chơi.
- Bạn có thể chọn nhân vật bằng cách ấn chuột vào màn hình có portrait nhân vật sau đó ấn nút tiếp tục ở góc phải bên dưới màn hình để vào màn chơi.
- Khi hover chuột qua nhân vật thì sẽ có chú thích kỹ năng của nhân vật, ví dụ trên cho thông tin kỹ năng của nhân vật không đội mũ.

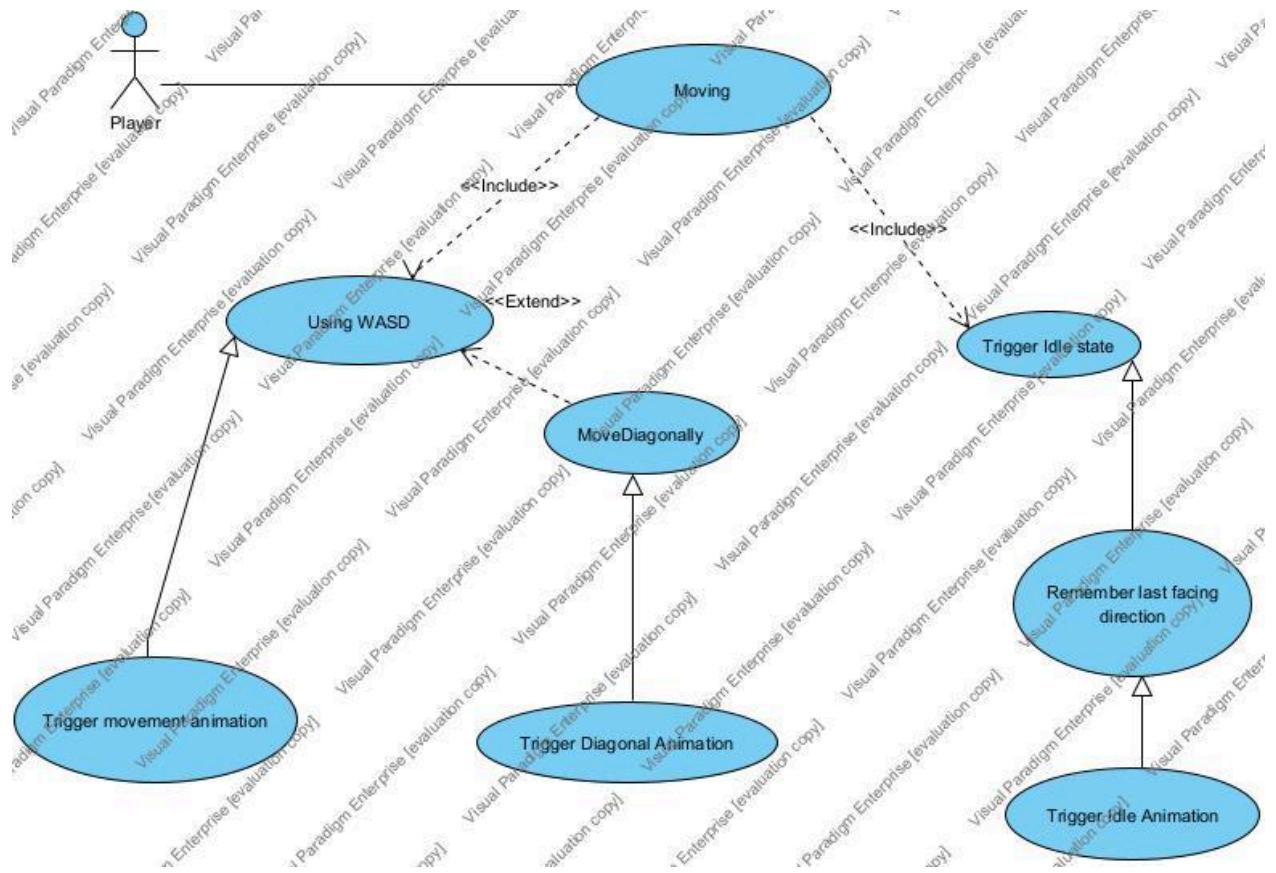
8. Thiết kế và triển khai cơ chế cho nhân vật

8.1.Thiết kế nhân vật

- Sử dụng các nhân vật từ [Small 8-direction Characters by AxulArt](#), mỗi nhân vật đại diện cho một kỹ năng khác nhau
- Các nhân vật sẽ được lựa chọn trước khi vào màn chơi, trừ màn chơi hướng dẫn sẽ chọn sẵn một nhân vật.
- 3 nhân vật trong asset sẽ được sử dụng, nhân vật trắng toàn thân sẽ có tên là WMan, nhân vật mũ xanh tên là BMan, nhân vật mũ cam tên là OMan.

8.2.Cơ chế di chuyển

Sơ đồ UseCase



Scenario:

| | |
|----------------|---|
| Module | Moving |
| Actor | Người chơi |
| Pre-condition | Người chơi vào màn chơi, có thể điều khiển |
| Post-condition | Người chơi ngừng di chuyển |
| Main-event | 1. Người chơi sử dụng tổ hợp phím WASD trên bàn phím. 2. Hệ thống nhận thông tin về phím bấm và nhận |

| | |
|-----------|--|
| | <p>vật sẽ được di chuyển theo hướng tương ứng.</p> <p>3. Hoạt ảnh được gán với hướng di chuyển sẽ được chọn.</p> <p>4. Người chơi sử dụng 2 phím di chuyển liên tiếp để di chuyển chéo, có hoạt ảnh hỗ trợ.</p> <p>5. Người chơi ngừng di chuyển, hệ thống ghi nhớ hướng di chuyển cuối cùng và sử dụng hoạt ảnh sao cho nhân vật nhìn về hướng tương ứng"</p> |
| Exception | 1.1. Người chơi bấm nhầm phím. |

- Người chơi sẽ sử dụng phím di chuyển WASD để điều khiển nhân vật, có thể kết hợp 2 phím gần nhau để nhân vật đi chéo, tất cả các hướng di chuyển đều có hoạt ảnh hỗ trợ, khi không di chuyển sẽ vào trạng thái chờ và sprite nhân vật đứng yên và nhìn về hướng di chuyển cuối cùng
- Sử dụng tính năng Animation của Unity để tạo hoạt ảnh. Kéo thả asset vào Animation để tạo dựng, sử dụng công thức vòng lặp 5 frame trong 0.45 giây: Đứng yên - di chuyển - đứng yên - di chuyển - đứng yên, tạo và đặt tên file hoạt ảnh mới theo công thức phím ấn + walk (nếu di chuyển chéo thì dùng cả 2 phím).
- Tạo thêm hoạt ảnh Idle bằng cách chỉ cho asset đứng yên tương ứng các hướng, Đặt tên theo công thức Idle + phím ấn.
- Sử dụng Animator để kiểm soát việc chuyển đổi giữa các hoạt ảnh khác nhau, sử dụng các Blend Tree cho 2 kiểu hoạt ảnh khi di chuyển và đứng im.

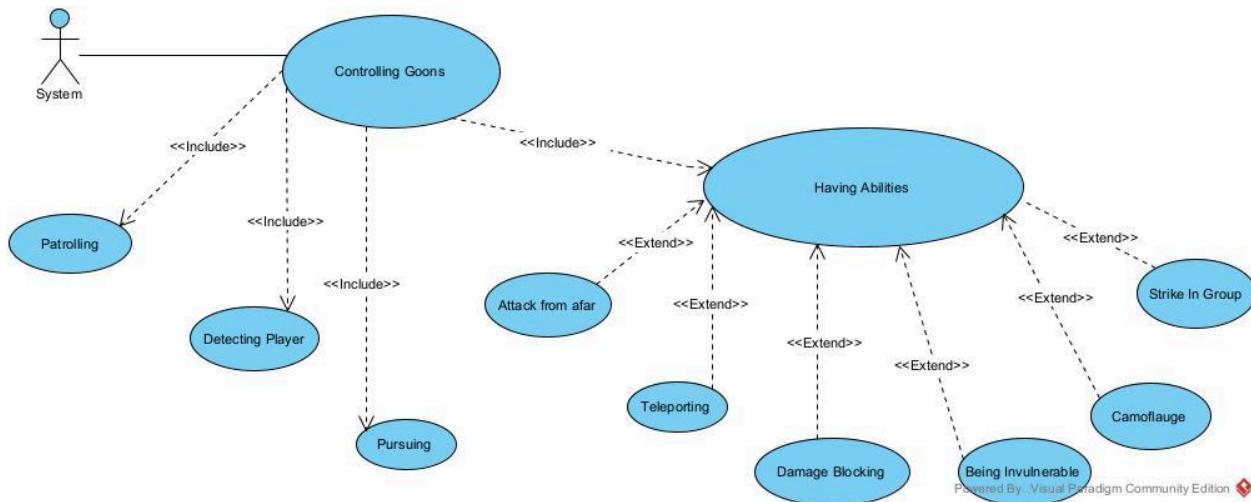
8.3.Thiết kế kỹ năng riêng cho các nhân vật

- WMan có thế mạnh trong việc hành động nhanh và lén lút, phù hợp với việc sử dụng súng lục, kỹ năng bắn sẽ là lẩn về phía trước, loại bỏ hoàn toàn sát thương trên đường lẩn.
- BMan là chuyên gia trong các cuộc chiến tranh vũ trang, ngoài việc thông thạo trong việc sử dụng các vũ khí. Kỹ năng của nhân vật là ném lựu đạn phá mảnh, gây sát thương theo hướng ngẫu nhiên – kể cả bản thân nếu đứng quá gần.

- OMan có thể đặt bẫy ở một khu vực chỉ định, khi kẻ địch đi vào sẽ gây sát thương ngay lập tức.

9.Thiết kế và triển khai kẻ địch

Sơ đồ Use Case



Scenario

| | |
|----------------|--|
| Module | Controlling Goons |
| Actor | Hệ thống |
| Pre-condition | Màn chơi bắt đầu, kẻ địch được sinh ra |
| Post-condition | Kẻ địch bị tiêu diệt. |
| Main-event | 1. Hệ thống đặt kẻ địch mới sinh ra vào trạng thái tuần tra, di chuyển xung quanh khu vực với trình tự ngẫu nhiên. 2. Hệ thống đặt trạng thái truy đuổi cho kẻ địch khi lọt vào tầm phát hiện. 3. Kẻ địch sẽ tiến hành truy đuổi với tốc độ đặt sẵn, thường là cao hơn tốc độ của người chơi |

| | |
|-----------|--|
| | 4.Kẻ địch có thể sử dụng khả năng đặc biệt 5.Hệ thống kết thúc trạng thái truy đuổi sau khi tiêu diệt người chơi hoặc hết thời gian |
| Exception | 3.1. Trạng thái truy đuổi không kết thúc, tốc độ truy đuổi không thay đổi so với tốc độ tuần tra |

9.1.Thiết kế địch

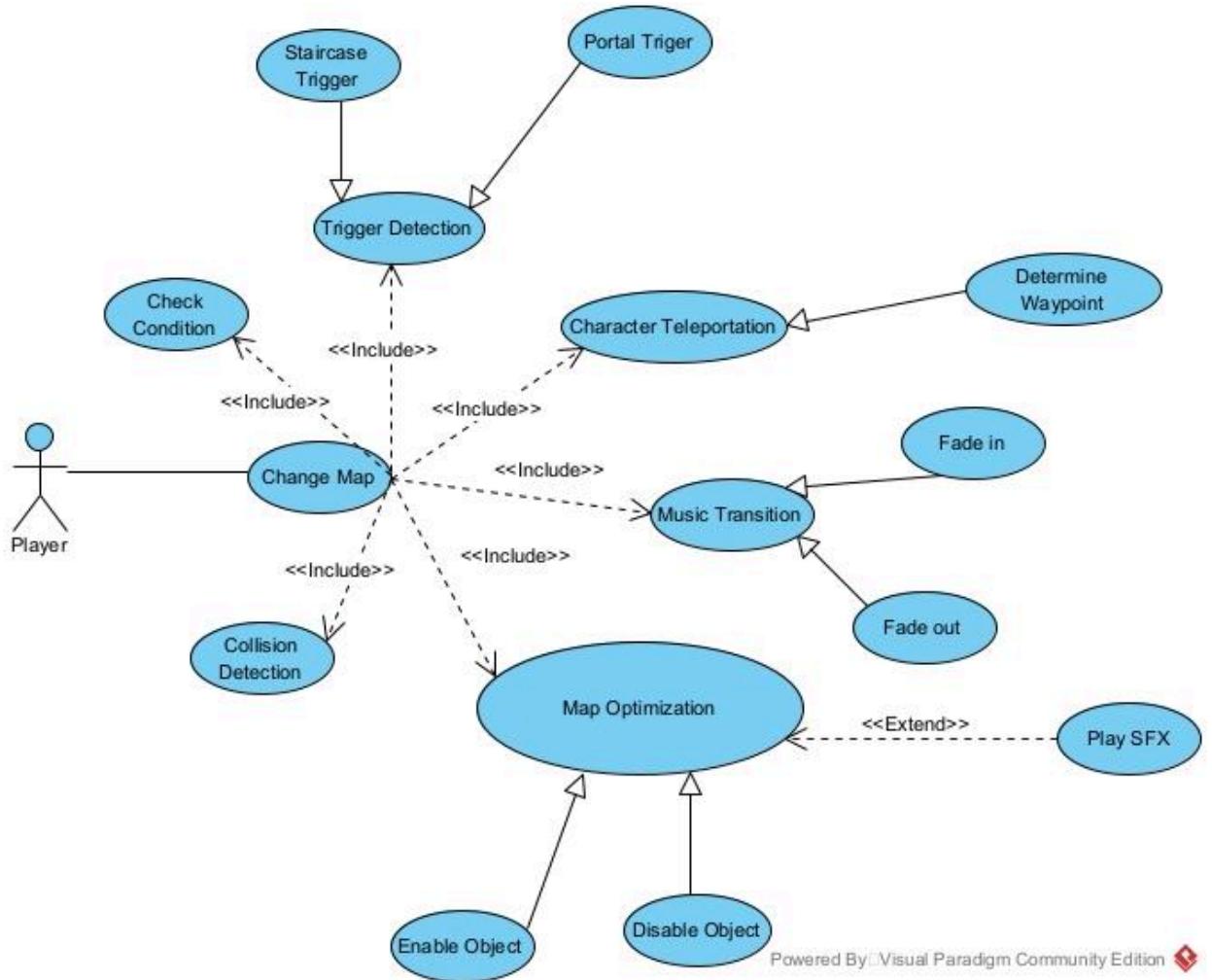
- Tạo đối tượng Enemy
 - Thiết lập GameObject: Tạo một GameObject mới đặt tên là Energy và gán Rigidbody2D để xử lý chuyển động vật lý.
 - Hình ảnh sử dụng sprite từ [Tiny RPG - Forest by Ansimuz](#) để thiết kế nhân vật phù hợp với phong cách game.
 - Cơ chế di chuyển và nhận diện người chơi(sử dụng EnemyMovement Script):
 - Tìm kiếm người chơi: Enemy sẽ liên tục quét khu vực xung quanh để phát hiện người chơi.
 - Di chuyển: Khi phát hiện, Enemy tự động di chuyển về phía người chơi với tốc độ được thiết lập.
 - Tấn công: Nếu đủ gần, Enemy sẽ kích hoạt hành động tấn công (tùy chỉnh theo thiết kế game).
 - Hệ thống nhận biết người chơi (Player Awareness)
 - Phát triển script PlayerAwareness để Enemy phản ứng linh hoạt với người chơi:
 - Xác định vị trí: Script lưu trữ Transform của người chơi để theo dõi vị trí.
 - Phạm vi nhận diện: Kiểm tra khoảng cách giữa Enemy và người chơi để quyết định hành vi (đuổi theo, tấn công, hoặc dừng lại).
 - Hành động khi phát hiện:
 - Theo dõi: Enemy tự động điều chỉnh hướng di chuyển về phía người chơi.
 - Tấn công: Khi vào phạm vi đủ gần, Enemy chuyển sang trạng thái tấn công (có thể là đánh cận chiến hoặc tầm xa)
- ➔ Những thành phần này sẽ tạo nên một hệ thống kẻ địch sống động, có khả năng tự động phản ứng với hành động của người chơi, góp phần tăng tính thách thức và làm phong phú trải nghiệm chơi game.

9.2. Thiết kế chi tiết kẻ địch

- Các màn chơi(Ngoại trừ màn chơi hướng dẫn và màn chơi đặc biệt) sẽ có 3 loại kẻ địch với mỗi màn sẽ có 1 loại địch chung và 2 loại địch riêng với các khả năng đặc biệt.
- Kẻ địch trong màn 1:
 - Evil zombie: Là kẻ địch chung xuất hiện trong tất cả các màn, chỉ có script chung dùng cho các kẻ địch.
 - Evil wizard: Có thể sử dụng các đòn đánh từ xa và dịch chuyển ở vị trí bất kỳ gần người chơi mỗi 3 giây.
 - Brute zombie:Là biến thể của evil zombie nhưng to hơn và có lớp giáp để chặn được một số đạn nhất định
- Kẻ địch trong màn 2:
 - Evil zombie: Là kẻ địch chung xuất hiện trong tất cả các màn, chỉ có script chung dùng cho các kẻ địch.
 - Assassin: Sẽ ngụy trang để ám sát người chơi, hiệu ứng ngụy trang sẽ làm cho kẻ địch trở nên khó nhìn hơn.
 - Hunting Dog: Tốc độ cao và có khả năng tấn công theo đàn.
- Kẻ địch trong màn 3:
 - Evil zombie: Là kẻ địch chung xuất hiện trong tất cả các màn, chỉ có script chung dùng cho các kẻ địch.
 - Cloud: Kẻ địch đặc biệt, có dạng như đám mây, không tính điểm khi hạ gục vì kẻ địch này bất tử, bạn chỉ có thể chạy.
 - Cloud Core: Kẻ địch điều khiển Cloud, đứng yên một chỗ, nếu bạn hạ gục kẻ thù này thì 1 Cloud sẽ biến mất.

10. Tạo và mở rộng chức năng cho map

Sơ đồ UseCase



Scenario

| | |
|--------|---------------------------|
| Module | Change Map and Transpoint |
| Actor | Player |

| | |
|----------------|--|
| Pre-condition | Player đi vào vùng trigger hợp lệ (cầu thang/cổng), game state cho phép chuyển map (không trong combat, dialogue...) |
| Post-condition | Nhân vật xuất hiện ở map mới, đổi tượng map cũ tắt, map mới bật và Camera cập nhật bound mới |
| Main-event | <ol style="list-style-type: none"> 1. Phát hiện va chạm trigger. 2. Kiểm tra điều kiện. 3. Dịch chuyển đến waypoint. 4. Thay đổi âm thanh. |
| Exception | <ol style="list-style-type: none"> 2.1. Điều kiện chuyển map không hợp lệ 3.1. Thiếu waypoint |

10.1. Chức năng chuyển map

- Thiết Kế Map và Cơ Chế Chuyển Đổi
 - Tạo và Thiết Lập Các Map
 - Ba GameObject riêng biệt đại diện cho các map M1, M2, M3, mỗi map có thiết kế và bộ cục khác nhau.
 - Khi người chơi đi vào vùng trigger, hệ thống sẽ xử lý logic chuyển map.
 - Cầu Thang và Ranh Giới Map
 - Các cầu thang được thiết kế như điểm kết nối giữa các tầng ($M1 \rightarrow M2 \rightarrow M3$).
 - Mapbound sử dụng cơ chế của UNITY để giới hạn khu vực di chuyển, ngăn người chơi đi ra khỏi phạm vi thiết kế.
- Hệ Thống Waypoint và Trigger Logic
 - Waypoint Định Vị Điểm Đến
 - Mỗi cửa chuyển map có một waypoint xác định vị trí xuất hiện của người chơi:
 - $M1 \rightarrow M2$: waypointM2
 - $M2 \rightarrow M3$: waypointM3
 - $M3 \rightarrow M2$: waypointM2
- Xử Lý Chuyển Map

- Khi người chơi vào vùng trigger, hệ thống kiểm tra va chạm và dịch chuyển nhân vật đến waypoint tương ứng ở map đích.
- Hệ thống hỗ trợ di chuyển hai chiều giữa các map.
- Các đối tượng không cần thiết được tắt/xử lý khi chuyển map để tối ưu hiệu suất.
- Hiệu Ứng Âm Thanh và Chuyển Cảnh
 - Âm Thanh Theo Map
 - Mỗi map có nhạc nền riêng (M1, M2, M3).
 - Sound effect được phát khi chuyển map (tiếng bước chân, cổng dịch chuyển).
 - Fade In/Out Âm Thanh
 - Điều chỉnh volume nhạc nền mượt mà khi chuyển map.

10.2.Thiết Lập Vùng Di Chuyển (Trigger)

- Tổng quan hệ thống
 - Hệ thống sử dụng các vùng trigger để phát hiện và kích hoạt sự kiện khi người chơi tương tác với môi trường game. Các sự kiện bao gồm hiển thị hướng dẫn, kích hoạt đói thoại NPC, mở khóa khu vực mới và chuyển đổi giữa các map.
- Cách thức hoạt động
 - Các đối tượng trong game được gắn collider dạng trigger để phát hiện khi người chơi đi vào vùng ảnh hưởng. Khi phát hiện va chạm, hệ thống sẽ thực thi các hành động tương ứng như hiển thị UI, phát âm thanh hoặc thay đổi trạng thái game.
- Ứng dụng chính
 - Hướng dẫn người chơi: Tự động hiển thị thông báo khi đến khu vực mới
 - Tương tác NPC: Kích hoạt đoạn hội thoại khi tiếp cận nhân vật
 - Mở rộng nội dung: Mở khóa khu vực mới khi đạt điều kiện
 - Chuyển cảnh: Di chuyển giữa các map khi vào vùng trigger
- Ưu điểm hệ thống
 - Linh hoạt trong việc mở rộng các loại sự kiện khác nhau
 - Dễ dàng tích hợp với các hệ thống khác như quest, inventory
 - Tạo trải nghiệm game mượt mà và tự nhiên cho người chơi
 - Hệ thống này đóng vai trò quan trọng trong việc xây dựng luồng game và dẫn dắt người chơi khám phá thế giới game một cách tự nhiên.

10.3.Tạo map UI cho đề tài



- Player Icon:

- Mục đích: Hiển thị vị trí hiện tại của người chơi trên bản đồ, giúp người dùng nắm bắt nhanh thông tin vị trí trong game.

Nội dung chi tiết:

- Tạo đối tượng: Sử dụng sprite hoặc hình ảnh vector để tạo biểu tượng cho người chơi.
 - Cập nhật vị trí: Icon sẽ được cập nhật liên tục theo dữ liệu vị trí người chơi nhận được từ hệ thống điều khiển của game.

- Một GameObject chứa sprite PNG đại diện cho người chơi sẽ tự động cập nhật vị trí khi game chạy.

- Kích Hoạt Sự Kiện:

- Canvas hệ thống được tạo làm nền tảng hiển thị giao diện người dùng

- Trong Canvas, thêm một Panel (hoặc Image) đại diện cho bản đồ.

- Bản đồ được thiết lập ẩn khi bắt đầu game để đảm bảo:

- Không làm rối giao diện ban đầu
 - Người chơi chủ động mở khi cần

- Hệ thống cho phép:

- Bật/tắt bằng phím tắt (thường là phím M)
 - Tự động hiện khi vào khu vực mới

- Ân khi di chuyển hoặc chiến đấu
- Kiểm Soát Bản Đồ Nâng Cao
 - Cập nhật vị trí thời gian thực của nhân vật trên bản đồ
 - Thao tác tương tác bao gồm:
 - Phóng to/thu nhỏ bản đồ
 - Di chuyển vùng hiển thị
 - Tích hợp đa hệ thống với:
 - Hệ thống chuyển cảnh
 - Cơ chế lưu trữ game
 - Đồng bộ tự động khi chuyển giữa các khu vực
 - Lưu trữ và khôi phục trạng thái:
 - Vị trí đánh dấu
 - Mức độ thu phóng
 - Các thiết lập hiển thị

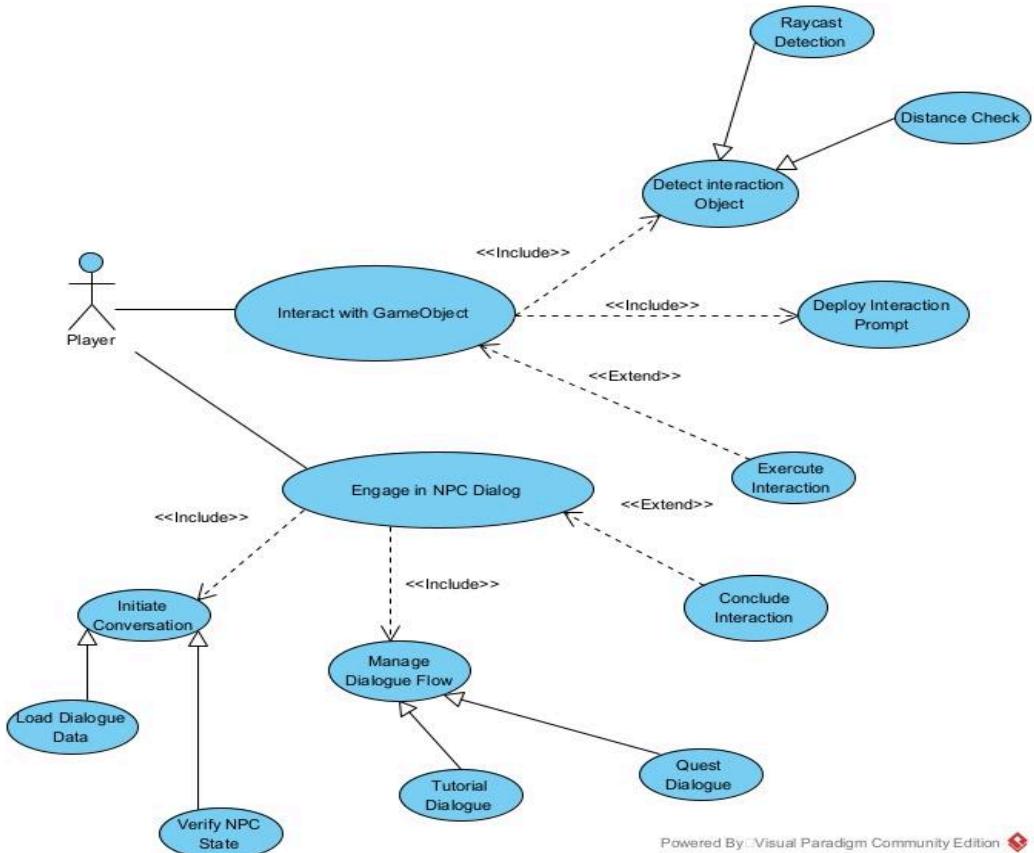
10.4.Tạo Dynamic map cho menu UI

- Disabling Map Manual Objects:
 - Ân các đối tượng Map_Manual đã được tạo sẵn nhằm ưu tiên hiển thị các đối tượng được điều khiển bởi script động.
 - Giúp hệ thống bản đồ có thể tự động điều chỉnh vị trí và hiển thị mà không bị ảnh hưởng bởi các đối tượng không đồng bộ.
- Map Dynamic Objects:
 - Tạo đối tượng động: Các đối tượng động được khởi tạo, thường bao gồm các điểm đánh dấu, vùng hoạt động và các thành phần phụ trợ được tạo ra tự động dựa trên dữ liệu.
 - Cho phép cập nhật linh hoạt khi có thay đổi trong game, giúp người dùng dễ dàng theo dõi các sự kiện cũng như vị trí của các yếu tố trong bản đồ.
- MapController Script:
 - Cấu trúc và chức năng:
 - Các hàm xử lý chính như cập nhật vị trí, zoom, pan và xử lý các sự kiện chuyển cảnh được trình bày.
 - Tích hợp điều khiển cho cả đối tượng động và các yếu tố giao diện khác, đảm bảo sự đồng bộ và mượt mà trong quá trình hiển thị.
- SaveController Script:
 - Lưu và khôi phục:

- Thêm chức năng lưu trạng thái từ MapController vào SaveController Script.
 - Các thông số như vị trí của Player Icon, mức zoom hiện tại và trạng thái các đối tượng động sẽ được lưu trữ và sau đó khôi phục khi cần thiết.
 - MapTransition Script:
 - Cập nhật bản đồ khi chuyển cảnh:
 - Sau khi người chơi kích hoạt sự kiện chuyển cảnh, MapTransition Script sẽ gọi hàm UpdateMapForNewArea() từ MapController để cập nhật toàn bộ giao diện bản đồ theo khu vực mới. Điều này giúp đảm bảo rằng các thông tin trên bản đồ luôn chính xác và đồng bộ với tiến trình của game.

11 Thiết kế tương tác, tạo player dialogue và tạo NPC

Sơ đồ UseCase



Scenario:

| | |
|----------------|---|
| Module | Cơ chế tương tác với môi trường và NPC Dialogue |
| Actor | Player |
| Pre-condition | Player trong phạm vi trigger, NPC ở trạng thái idle |
| Post-condition | UI prompt ẩn đi và hiệu ứng tương tác kích hoạt (mở rương/nhặt đồ), UI dialogue đóng và NPC trở lại trạng thái ban đầu |
| Main-event | 1. Player di chuyển vào vùng trigger 2. Hiện icon "E" 3. Nhấn E gần NPC sẽ hiện portrait + tên 4. Text chạy từ từ + âm thanh 5. Player nhấn Next để tiếp tục/dóng |
| Exception | 3.1. Player di chuyển xa Dialogue tự động đóng 4.1. Player không đủ điều kiện (ví dụ: thiếu key) Play sound "error" |

11.1. Hệ Thống Tương Tác Đa Dạng

- Game tích hợp cơ chế tương tác linh hoạt giữa người chơi với môi trường xung quanh thông qua:
 - Hệ thống phát hiện tương tác sẽ nhận diện các đối tượng có thể tương tác
 - Giao diện trực quan hiển thị biểu tượng khi có đối tượng tương tác trong phạm vi
 - Cơ chế input cho phép kích hoạt bằng phím E
 - Các đối tượng tương tác bao gồm:
 - Rương đồ và vật phẩm
 - NPC với hệ thống hội thoại
 - Các điểm tương tác đặc biệt trong môi trường

11.2. Hệ Thống Hội Thoại NPC

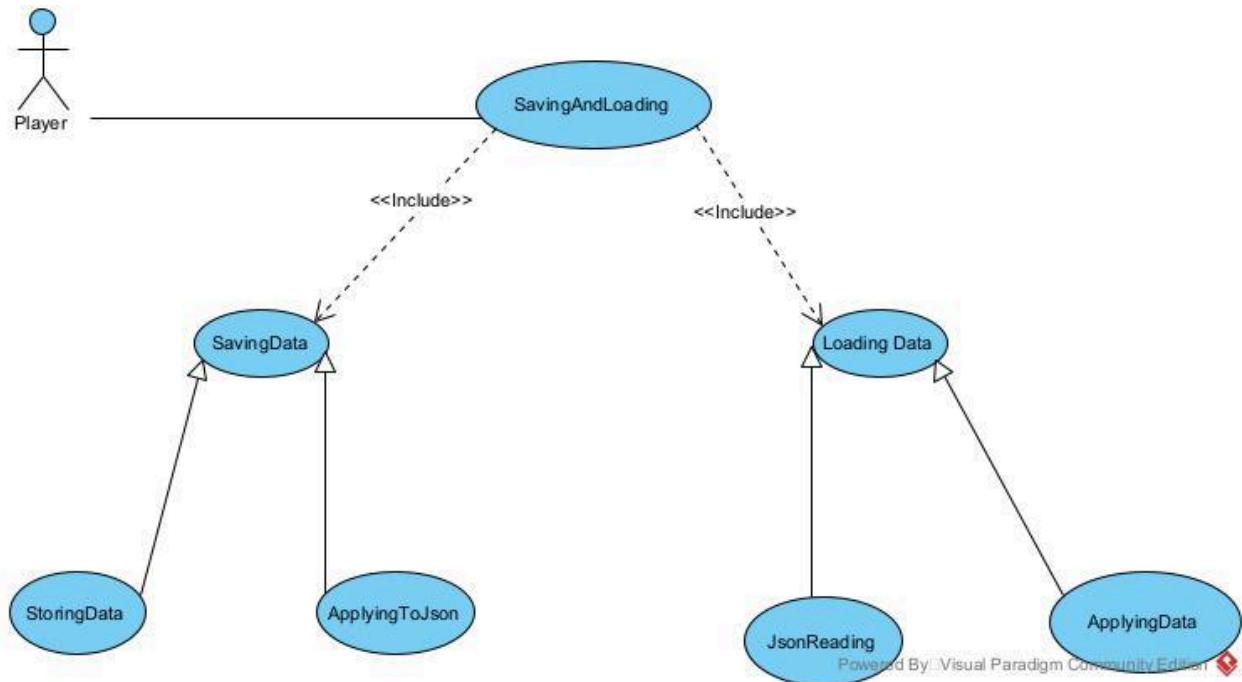
- NPC trong game được thiết kế với:
 - Giao diện hội thoại chuyên nghiệp hiển thị:
 - Tên nhân vật
 - Ảnh đại diện
 - Nội dung thoại với hiệu ứng chạy chữ
 - Hệ thống âm thanh đồng bộ với lời thoại
 - Cơ chế tương tác thông qua phím E

11.3. Quy Trình Tương Tác

- Người chơi tiếp cận đối tượng có thể tương tác
 - Hệ thống hiển thị biểu tượng nhắc nhở
 - Nhấn phím E để kích hoạt tương tác
 - Thực hiện hành động tương ứng (mở rương, bắt đầu hội thoại...)
 - Kết thúc tương tác và trở lại trạng thái bình thường
- Những chi tiết này không chỉ giúp người chơi dễ dàng tiếp cận và hiểu về các tính năng của game mà còn góp phần xây dựng một trải nghiệm tutorial trực quan, mượt mà và hấp dẫn.

12. Thiết kế cơ chế lưu/tải dữ liệu

Sơ đồ Use Case



Scenario

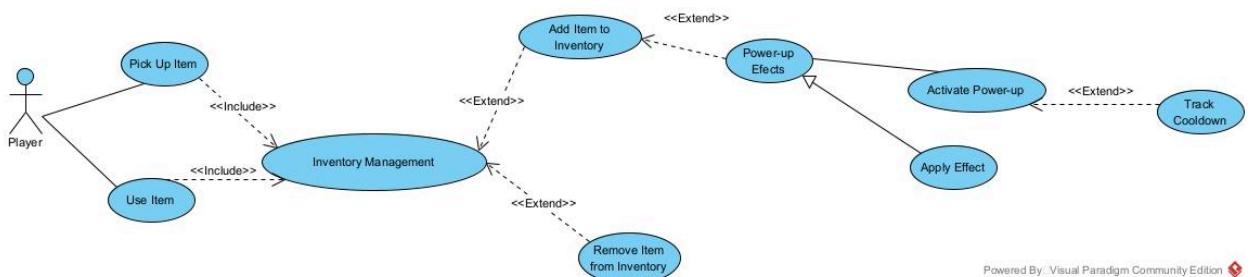
| | |
|----------------|--|
| Module | Saving And Loading |
| Actor | Người chơi |
| Pre-condition | Người chơi mở giao diện hệ thống |
| Post-condition | Kết thúc lưu/tải, người chơi tiếp tục chơi hoặc thoát trò chơi |
| Main-event | 1. Người chơi sau khi mở giao diện hệ thống sẽ chọn giữa lưu(save) dữ liệu hoặc tải(load) dữ liệu 2. Người chơi chọn lưu, hệ thống sẽ lấy thông tin màn chơi, vị trí đứng, vũ khí hiện tại, nhân vật hiện tại,... rồi lưu lại thành một lớp data. Sau đó lớp data này sẽ được áp dụng vào một file json |

| | |
|-----------|---|
| | 3. Người chơi chọn tải, hệ thống sẽ đọc file json cuối cùng được tạo ra, lấy dữ liệu từ file và tái tạo lại màn chơi với thông tin đã có. |
| Exception | 3.1. Không có file json nào để tải dữ liệu |

- Thiết lập cơ chế qua việc sử dụng định dạng file json lưu trữ các dữ liệu cần thiết như vị trí đứng, item có trong inventory, vũ khí đang cầm và số đạn còn lại.
- Sẽ chỉ có 1 file lưu dữ liệu, những lần sau sẽ viết đè lên file cũ.
- Trong giao diện người dùng, khi ấn nút 'Save/Load', sẽ hiện ra 2 nút tương ứng Save và Load. Khi ấn "Save" dữ liệu sẽ được lưu vào một lớp lưu trữ. Một script sẽ lấy dữ liệu từ lớp này và tạo một file JSON để lưu lại. Khi ấn nút 'Load', script sẽ đọc file JSON, lấy dữ liệu và áp dụng vào game.
- Tương tự, trong giao diện menu chính, khi ấn play, nếu có file JSON, tồn tại, sẽ hiển thị thêm 2 nút là "New game" và "Continue", "New game" vẫn sẽ có nhiệm vụ như nút "Play" nhưng "Continue" sẽ đọc dữ liệu file JSON và đưa người chơi về dữ liệu đã lưu.

13. Thiết lập cơ chế vật phẩm dùng nhanh

Sơ đồ Usecase



Scenario

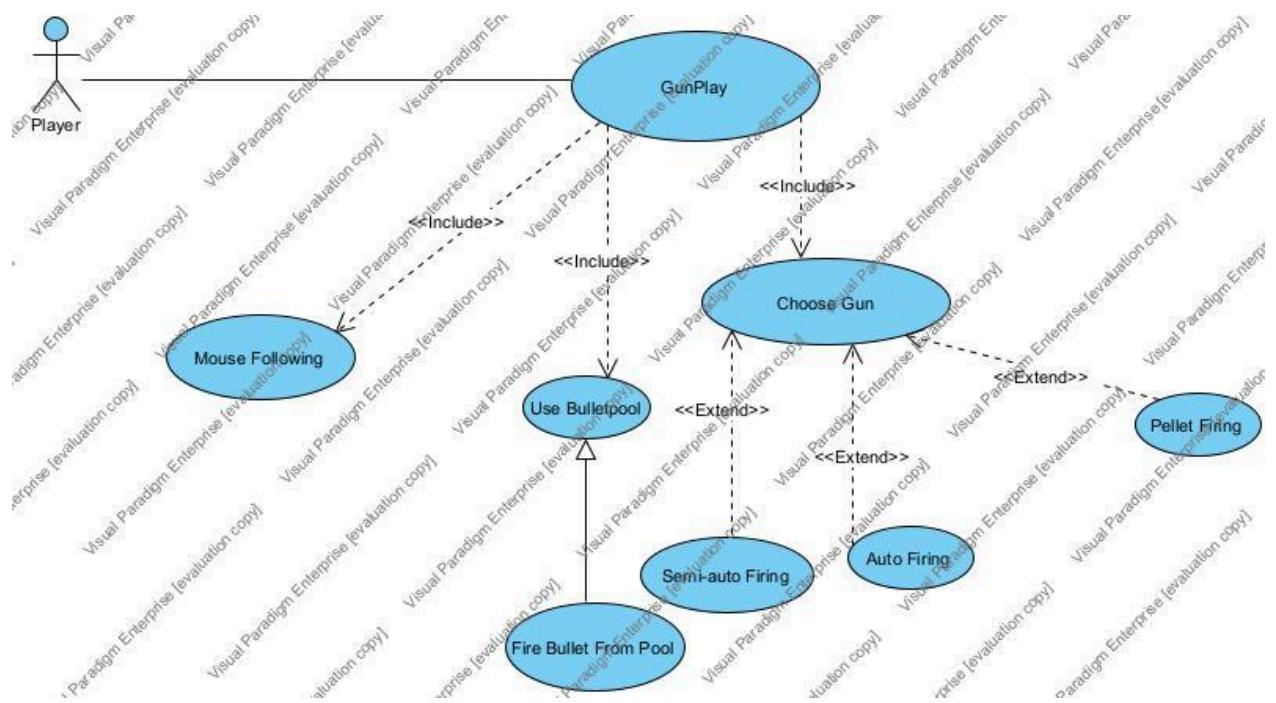
| | |
|----------------|--|
| Module | Pick up and Use Item |
| Actor | Player |
| Pre-condition | Vật phẩm tồn tại trong môi trường game, trong inventory có ít nhất 1 vật phẩm hợp lệ để player dùng |
| Post-condition | Vật phẩm được thêm vào Inventory, ngay khi được kích hoạt cooldown và player nhận được hiệu ứng |
| Main-event | <ol style="list-style-type: none"> 1. Player di chuyển đến vật phẩm. 2. Hệ thống kiểm tra ô trống. 3. Thêm vật phẩm, cập nhật UI. 4. Nhấn phím 1/2/3. 5. Kiểm tra cooldown. 6. Áp dụng hiệu ứng. 7. Cập nhật UI và bộ đếm 30s. 8. Nhận loại vật phẩm từ Inventory: Cà chua: Bất tử, Thịt: Vô hạn đạn, Đậu: Tăng tốc. 9. Đếm ngược 30s. 10. Hiển thị thanh cooldown trên UI. 11. Gửi sự kiện "Cooldown Finished" |
| Exception | <p>3.1. Inventory đầy → Hiển thị "Full", không nhặt.</p> <p>4.1. Ô trống không phản ứng khi nhấn phím.</p> <p>5.1 Vật phẩm cooldown bị lỗi vẫn dùng được trong cooldown.</p> |

- Các vật phẩm sử dụng sẽ đóng vai trò như một nâng cấp ngắn hạn cho người chơi, thời gian hiệu lực của các nâng cấp sẽ trong 30 giây.
- Sử dụng [Canned foods pixel art 32x32 asset pack](#) làm các vật phẩm.
- Với giao diện Inventory, các vật phẩm được nhặt sẽ tìm đến ô trống từ bên trái nhất, nếu đầy các ô thì sẽ không nhặt, mỗi ô chỉ chứa duy nhất 1 vật phẩm, áp dụng tương tự với giao diện Inventory mini.

- Sử dụng các vật phẩm bằng cách ấn 1, 2 và 3 tương ứng với 3 ô Inventory.
- Các vật phẩm và các ô vật phẩm sẽ được gắn script, khi vật phẩm được nhặt, thông tin sẽ được gửi tới qua nguồn trung gian là hệ thống quản lý giao diện chung - script quản lý toàn bộ các tính năng liên quan đến giao diện. Từ nguồn trung gian đó, thông tin sẽ được gửi đến các ô vật phẩm để thực hiện chức năng chứa vật phẩm.
- Sẽ có 3 loại nâng cấp cho người chơi:
 - Cà chua hộp: Bất tử.
 - Thịt hộp: Vô hạn đạn.
 - Đậu Hà Lan hộp: Tăng tốc độ.
- Thanh trạng thái ở giao diện hiển thị người dùng sẽ đóng vai trò như một bộ đếm sau khi người chơi sử dụng một nâng cấp bất kì, bộ đếm sẽ trong 30 giây. Sau khi thời gian đếm kết thúc, người chơi mới có thể sử dụng tiếp các nâng cấp khác, đảm bảo sự cân bằng của trò chơi.

14.Thiết lập cơ chế bắn súng

Sơ đồ Use Case



Scenario

| | |
|----------------|---|
| Module | GunPlay |
| Actor | Người chơi |
| Pre-condition | Người chơi nhặt được súng |
| Post-condition | Người chơi bắn hết đạn và ném súng đi |
| Main-event | <p>1. Người chơi sử dụng chuột để điều khiển hướng súng, hướng 360 độ.</p> <p>2. Đầu súng tự động đi theo hướng chuột, quyết định hướng bắn.</p> <p>3. Người chơi nhấn chuột trái, đạn được bắn ra từ BulletPool có sẵn.</p> <p>4.. Người chơi sử dụng súng lục, bắn tốc độ bình thường mỗi lần nhấn chuột là một lần bắn.</p> <p>5. Người chơi sử dụng súng trường liên thanh, có thể ấn giữ để bắn</p> <p>6.Người chơi sử dụng súng săn, bắn nhiều viên đạn trong một lần ấn chuột, tốc độ chậm.</p> <p>7. Hệ thống kiểm trả số đạn trong BulletPool , nếu hết đạn thì sẽ không bắn đạn tiếp.</p> <p>8. Người chơi ném súng để tiếp tục nhặt vũ khí</p> |
| Exception | <p>7.1. BulletPool bị lỗi, súng không có đạn.</p> <p>8.1. Người chơi bấm nhầm nút, không ném được vũ khí để thay vũ khí mới.</p> |

14.1.Tổng quan của cơ chế bắn súng

- Người chơi sử dụng súng làm vũ khí chính, ấn chuột trái để bắn và ấn r để ném, sẽ không có cơ chế nạp đạn để tăng tính chiến thuật, giúp mở ra nhiều cách hoàn thành màn chơi.
- Súng tương tự như các vật phẩm, nhưng khi nhặt sẽ không vào ô vật phẩm nào mà sẽ sử dụng ngay lập tức, thông tin về súng như

số đạn còn lại sẽ được hiển thị trên thanh vũ khí trong giao diện hiển thị người dùng.

- Cơ chế súng xoay hướng theo chuột, điểm xoay ở tay cầm súng.
- Đầu súng sẽ tạo script để sinh ra đạn, mỗi loại súng sẽ có tốc độ, cơ chế bắn riêng.
- Sử dụng phương pháp Object Pooling [...] để tránh việc khi sinh ra đạn sẽ làm đầy bộ nhớ RAM.
- Tạo Object đạn làm lưu trữ, gắn script Pooling và liên kết tới các script liên quan.
- Khi bắn các Object đạn trong Pool sẽ liên tục ở trạng thái Active/Inactive, tiết kiệm không gian và đảm bảo trải nghiệm cho người chơi.

14.2. Cơ chế riêng của các loại súng.

- Sẽ có ba loại súng dùng chơi game: Súng lục, súng trường liên thanh, súng săn tầm gần.
- Súng lục sẽ bắn phát một, tốc độ bắn tiêu chuẩn.
- Súng trường liên thanh bắn liên tục, có thể ẩn giữ để bắn.
- Súng săn bắn phát một, bắn nhiều đạn trong 1 lần bắn