**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**HOÀNG QUỐC MINH**

**PHÁT TRIỂN GAME HẦM NGỤC 3D SỬ DỤNG UNREAL ENGINE**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2024BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**HOÀNG QUỐC MINH**

**PHÁT TRIỂN GAME HẦM NGỤC 3D SỬ DỤNG UNREAL ENGINE**

**Mã số sinh viên: 2151013053**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: DƯƠNG HỮU THÀNH**

**TP. HỒ CHÍ MINH, 2024**

LỜI CẢM ƠN

Đầu tiên, em xin được gửi lời cảm ơn đến Trường Đại học Mở Thành phố Hồ Chí Minh đã tạo cho em một môi trường học tập và rèn luyện trong suốt ba năm học vừa qua. Đây là quá trình mà em từ một sinh viên năm nhất, liên tục học tập trau dồi kiến thức và rèn luyện không ngừng để có thể phát triển bản thân đến mức hiện tại. Việc thực hiện môn đồ án này là một cơ hội giúp em ứng dụng được tất cả những kiến thức đã được học và có cơ hội được tìm hiểu thêm những kiến thức mới.

Em xin được gửi lời cảm ơn đến tất cả các giảng viên của khoa Công nghệ thông tin, những người thầy, người cô đã cho em những kiến thức mới, những kinh nghiệm xương máu của ngành này và đã giúp em rèn luyện nền tảng tư duy cần thiết trong suốt quá trình em theo học tại trường. Trong số đó, em xin đặc biệt cảm ơn ThS. Dương Hữu Thành, người đã dẫn dắt cũng như cho em những lời khuyên bổ ích trong suốt quá trình làm đồ án. Thầy đã có những đóng góp, chia sẻ để em có thể hoàn thiện đồ án một cách chỉnh chu nhất có thể.

Bên cạnh đó, em cũng xin cảm ơn những người bạn tốt, những người anh em, gia đình, những người đã luôn ở bên, động viên và cho em những góp ý cần thiết trong khoảng thời gian làm đồ án. Sự có mặt và quan tâm của họ đã cho em thêm động lực để có thể thực hiện đồ án một cách tốt nhất.

Tuy bản thân đã cố gắng nỗ lực để cho ra sản phẩm tốt nhất, với việc bị hạn chế về thời gian cũng như sử dụng một công nghệ không quen thuộc, em biết rằng có làm tốt, làm kĩ đến mấy thì chắc chắn cũng sẽ có những thiếu sót xảy ra. Vì vậy, em mong được nhận những góp ý và thông cảm của thầy cô cho đề tài đồ án này để em rút kinh nghiệm khi em thực hiện các đề tài tương tự trong tương lai.

Cuối cùng, em xin gửi lời chúc sức khỏe đến tất cả thầy cô của khoa và tất cả thầy cô, cán bộ công chức của trường. Em xin được gửi lời cảm ơn sâu sắc nhất đến những người thầy, cô đã đồng hành và giúp đỡ em trong suốt thời gian em theo học tại trường.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

Đề tài này tập trung vào việc xây dựng một trò chơi đánh quái và khám phá hầm ngục. Trò chơi này cần có đầy đủ các chức năng chiến đấu cơ bản như tấn công, phòng thủ, hồi máu. Về phần tấn công, phải có đa dạng các loại hình tấn công để tăng độ khó cũng như trải nghiệm khi chơi trò chơi này. Ngoài ra thì game cũng phải được tích hợp một hệ thống điều khiển kẻ thù thông minh với nhiều loại trạng thái và đa dạng kẻ thù, mỗi kẻ thù sẽ có những hành động riêng biệt để tăng độ khó cho trò chơi và không làm game bị nhàm chán. Bản đồ phải thật rộng rãi để không gây cảm giác chật hẹp khi chơi, đồng thời thì bầu không khí cũng phải mang đúng chất của một hầm ngục. Quan trọng nhất là phải đảm bảo điều khiển thân thiện với người chơi, giao diện đơn giản, không bị gò bó, không gây khó chịu khi nhìn. Công nghệ được sử dụng để xây dựng lên trò chơi này là Unreal Engine, và phiên bản được dùng là Unreal Engine 5.4. Với việc có nhiều công cụ mạnh mẽ được cung cấp như Blueprint, AI System, Rendering Engine… Unreal Engine chính là sự lựa chọn hoàn hảo để đạt được các mục tiêu đã đề ra. Trò chơi tạo ra đã gần như đáp ứng được tất cả mục tiêu đã đề ra từ ban đầu. Sau khi qua quá trình mời bạn bè chơi thử và nhận xét, em đã rút ra được những điểm mạnh của game. Hai điểm mạnh chính nằm ở việc nó có một hệ thống chiến đấu đa dạng và nhiều loại kẻ thù, điều đó giúp trò chơi có tính chơi lại cao, và tuy bản đồ mỗi lần chơi là như nhau, trải nghiệm mỗi lần chơi đều sẽ mang lại cảm giác mới mẻ cho người chơi.

ABSTRACT

This project focuses on building a monster-fighting and dungeon-exploring game. This game needs to have all the basic combat functions such as attack, defense, and healing. As for the attack, there must be a variety of attack types to increase the game’s difficulty and the user’s experience when playing. In addition, the game must also be integrated with an intelligent enemy control system with many types of states and a variety of enemies. Each enemy will have separate actions to increase the difficulty of the game and not make the game boring. The map must be spacious enough to not cause a feeling of narrowness, and at the same time, the atmosphere must also capture the true essence of a dungeon. The most important thing is to ensure that the controls are user-friendly, the interface is simple, not constrained, and does not cause annoyance to look at. The technology used to build this game is Unreal Engine, and the version used is Unreal Engine 5.4. With many powerful tools provided such as Blueprint, AI System, Rendering Engine… Unreal Engine is the perfect choice to achieve all the set goals. The created game has met almost all the goals set. After inviting friends to trial-play and getting their afterthought, I have drawn out the strengths of the game. The two main strengths lie in the fact that it has a diverse combat system and many types of enemies, which makes the game highly replayable, and although the map is the same each time you play, the experience the player gets each time is new.

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc179109081)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc179109082)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc179109083)

[ABSTRACT 4](#_Toc179109084)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc179109085)

[DANH MỤC HÌNH VẼ 8](#_Toc179109086)

[DANH MỤC BẢNG 9](#_Toc179109087)

[MỞ ĐẦU 10](#_Toc179109088)

[Chương 1. TỔNG QUAN ĐỀ TÀI 11](#_Toc179109089)

[1.1. Giới thiệu đề tài 11](#_Toc179109090)

[1.2. Lý do chọn đề tài 11](#_Toc179109091)

[1.3. Mục tiêu và phạm vi đề tài 12](#_Toc179109092)

[1.4. Phương pháp nghiên cứu 12](#_Toc179109093)

[1.5. Bố cục báo cáo 13](#_Toc179109094)

[Chương 2. CƠ SỞ LÝ THUYẾT 14](#_Toc179109095)

[2.1. Unreal Engine 14](#_Toc179109096)

[2.1.1. Giới thiệu Unreal Engine 14](#_Toc179109097)

[2.1.2. Kiến trúc của Unreal Engine 15](#_Toc179109098)

[2.1.3. Cấu trúc một project Unreal Engine 18](#_Toc179109099)

[Chương 3. HỆ THỐNG GAME HẦM NGỤC 3D 20](#_Toc179109100)

[3.1. Giới thiệu bài toán 20](#_Toc179109101)

[3.2. Phân tích hệ thống 21](#_Toc179109102)

[3.2.1. Các actor 21](#_Toc179109103)

[3.2.2. Các usecase 21](#_Toc179109104)

[3.2.3. Use case toàn cục hệ thống 22](#_Toc179109105)

[3.2.4. Đặc tả usecase 22](#_Toc179109106)

[3.3. Thiết kế hệ thống 24](#_Toc179109107)

[3.3.1. Lược đồ tuần tự 24](#_Toc179109108)

[3.3.2. Lược đồ hoạt động 28](#_Toc179109109)

[3.3.3. Thiết kế dữ liệu 31](#_Toc179109110)

[3.3.4. Thiết kế giao diện 33](#_Toc179109111)

[3.4. Kiến trúc hệ thống 35](#_Toc179109112)

[3.4.1. Tổng quan 35](#_Toc179109113)

[3.4.2. Hệ thống di chuyển 38](#_Toc179109114)

[3.4.3. Hệ thống sát thương 38](#_Toc179109115)

[3.4.4. Hệ thống AI 41](#_Toc179109116)

[3.5. Kết quả thực hiện 44](#_Toc179109117)

[3.5.1. Bầu không khí u ám, ảm đạm 44](#_Toc179109118)

[3.5.2. Đa dạng kẻ thù, đa dạng cách chiến đấu 45](#_Toc179109119)

[3.5.3. Trận đấu huy hoàng với trùm cuối 47](#_Toc179109120)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc179109121)

[4.1. Kết luận 49](#_Toc179109122)

[4.2. Hướng phát triển 50](#_Toc179109123)

[TÀI LIỆU THAM KHẢO 52](#_Toc179109124)

[PHỤ LỤC 53](#_Toc179109125)

DANH MỤC TỪ VIẾT TẮT

UE: Unreal Engine

UE5: Unreal Engine 5

API: Application Programming Interface

EQS: Environment Query System

DANH MỤC HÌNH VẼ

[Hình 2.1. Logo Unreal Engine (nguồn: https://www.unrealengine.com) 14](#_Toc179643236)

[Hình 2.2. Trình chỉnh sửa Blueprint (nguồn: https://dev.epicgames.com) 15](#_Toc179643237)

[Hình 2.3. Luồng hoạt động của Gameplay Framework khi chạy một project. Hình chi tiết luồng hoạt động ở phần phụ lục. 16](#_Toc179643238)

[Hình 2.4. Một ví dụ cho việc sử dụng Lumen để thắp sáng môi trường (nguồn: https://dev.epicgames.com) 17](#_Toc179643239)

[Hình 2.5. Một cây quyết định (Behavior Tree) đơn giản trong Unreal Engine (nguồn: https://dev.epicgames.com) 17](#_Toc179643240)

[Hình 2.6. Hệ thống định vị (Navigation System) để lựa chọn vị trí AI có thể di chuyển đến (nguồn: https://dev.epicgames.com) 18](#_Toc179643241)

[Hình 2.7. Giao diện trình chỉnh sửa một project (nguồn: https://dev.epicgames.com) 18](#_Toc179643242)

[Hình 3.1. Use case toàn cục hệ thống 22](#_Toc179643243)

[Hình 3.2. Lược đồ tuần tự của chức năng “Thực hiện đòn tấn công” 26](#_Toc179643244)

[Hình 3.3. Lược đồ tuần tự của chức năng “Đỡ đòn tấn công” 28](#_Toc179643245)

[Hình 3.4. Lược đồ hoạt động của chức năng “Thực hiện đòn tấn công” 29](#_Toc179643246)

[Hình 3.5. Lược đồ hoạt động của chức năng “Đỡ đòn tấn công”. 31](#_Toc179643247)

[Hình 3.6. Giao diện lúc bình thường 33](#_Toc179643248)

[Hình 3.7. Giao diện khi đánh boss 34](#_Toc179643249)

[Hình 3.8. Menu chiến thắng khi đánh bại trùm cuối 34](#_Toc179643250)

[Hình 3.9. Menu game over khi bị kẻ địch đánh bại 35](#_Toc179643251)

[Hình 3.10. Kiến trúc hệ thống game 37](#_Toc179643252)

[Hình 3.11. Combo tấn công cận chiến của người chơi 40](#_Toc179643253)

[Hình 3.12. Behavior Tree của kẻ thù cận chiến 43](#_Toc179643254)

[Hình 3.13. Một danh sách các điểm của một có được bằng cách sử dụng EQS 44](#_Toc179643255)

[Hình 3.14. Bản đồ mê cung hầm ngục với rất nhiều kẻ thù (thanh máu màu đỏ) 45](#_Toc179643256)

[Hình 3.15. Chiến đấu với một tốp kẻ thù cận chiến 46](#_Toc179643257)

[Hình 3.16. Chiến đấu với kẻ thù pháp sư 46](#_Toc179643258)

[Hình 3.17. Trùm cuối chuẩn bị dùng chiêu thức của mình 47](#_Toc179643259)

[Hình 3.18. Tấn công trùm cuối bằng combo cận chiến 48](#_Toc179643260)

DANH MỤC BẢNG

[Bảng 3.1. Đặc tả usecase “Thực hiện đòn tấn công” 23](#_Toc179643530)

[Bảng 3.2. Đặc tả usecase “Đỡ đòn tấn công” 23](#_Toc179643531)

[Bảng 3.3. Struct DamageInfo, được dùng để lưu các thông tin của một đòn tấn công hoàn chỉnh 32](#_Toc179643532)

[Bảng 3.4. Enum MovementSpeed lưu các mức tốc độ di chuyển 38](#_Toc179643533)

[Bảng 3.5. Enum DamageType, lưu các loại sát thương 39](#_Toc179643534)

[Bảng 3.6. Enum DamageRespond, lưu các loại phản hồi khi nhận sát thương 39](#_Toc179643535)

[Bảng 3.7. Enum PlayerStance, dùng để lưu các trạng thái của người chơi 40](#_Toc179643536)

[Bảng 3.8. Enum AIState, dùng để lưu trạng thái của kẻ thù 41](#_Toc179643537)

[Bảng 3.9. Enum MeleeEnemyAttack, lưu các đòn tấn công của kẻ thù cận chiến 43](#_Toc179643538)

MỞ ĐẦU

Trong xã hội hiện nay, nhu cầu giải trí ngày càng trở nên đa dạng và phong phú, trong số đó, trò chơi điện tử (hay game) đã vươn lên và chiếm một vị trí vô cùng quan trọng, đặc biệt là với giới trẻ. Giờ đây, game không chỉ là phương tiện để giảm căng thẳng hay thư giãn, mà nó còn là một lĩnh vực nghệ thuật đòi hỏi sự sáng tạo, kết hợp giữa yếu tố kỹ thuật và thẩm mỹ. Mỗi tựa game ra đời đều mang theo những câu chuyện, cảm xúc, và thông điệp riêng, tạo nên một thế giới ảo đầy mê hoặc, thu hút hàng triệu người chơi trên toàn thế giới.

Sự phát triển của ngành công nghiệp trò chơi điện tử không chỉ đơn thuần nằm ở khía cạnh giải trí, mà đó còn là cơ hội cho các nhà làm game thể hiện những ý tưởng độc đáo, phản ánh được quan điểm cá nhân, các giá trị văn hóa và xã hội. Tuy vậy, hành trình để phát triển một tựa game hoàn chỉnh lại là một thử thách không nhỏ đối với bất kỳ ai theo đuổi đam mê này. Từ những khía cạnh tưởng chừng đơn giản như việc tạo dựng cốt truyện, thiết kế nhân vật, đến các yếu tố phức tạp như lập trình hệ thống, quản lý cơ chế gameplay, tất cả đều đòi hỏi sự tỉ mỉ và kiên trì trong từng giai đoạn. Mỗi bước tiến trong quá trình phát triển game đều cần sự cân nhắc kỹ lưỡng, bởi một thay đổi nhỏ có thể ảnh hưởng lớn đến sự thành công hay thất bại của toàn bộ tựa game.

Do đó, việc tìm hiểu và nghiên cứu phương pháp, cũng như những điều nên tránh khi tạo nên một tựa game là điều vô cùng quan trọng và đây cũng là mục tiêu của em trong lần thực hiện đồ án này.

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Chơi trò chơi điện tử (hay game) là một trong những phương thức giải trí phổ biến nhất trong giới trẻ hiện nay. Một việc khác cũng phổ biến không kém, và hơn nữa còn là mơ ước của một phần lớn giới trẻ chính là việc tự tay tạo ra những tựa game theo ý muốn của bản thân. Với sự phát triển của các công cụ làm game hiện đại, việc làm game đã trở nên càng dễ tiếp cận hơn rất nhiều cho những người có đam mê với công việc này. Sự phát triển đó đã dẫn đến sự ra đời của rất nhiều tựa game lớn nhỏ với đa dạng các thể loại và có một số tựa game cực kì thành công.

Tuy nhiên thì việc làm game cũng có các hạn chế, như việc phần lớn game làm ra đều thất bại trong một khoảng thời gian ngắn mà không thể kéo dài được lâu do nhiều lý do, có thể là do không đủ kinh phí vận hành, game không được đón nhận tốt hay các lý do nhạy cảm khác. Từ đó có thể thấy ngành công nghiệp làm game hiện nay tuy đang là một cơ hội phát triển rất tốt, nhưng nó vẫn tiềm tàng nhiều khó khăn và bất lợi cho những cá nhân, tập thể hay công ty làm game.

Tuy vậy, em vẫn mong muốn được thử sức trong ngành công nghiệp đầy tính cạnh tranh và khắc nghiệt này. Và mục đích cuối cùng của em sẽ là rút ra được những kinh nghiệm quý báu trong quá trình làm game, và dùng những kiến thức đó kết hợp với một trong những công cụ làm game đang có mặt ngoài thị trường để tạo nên một con game đầu tay cho của bản thân.

## Lý do chọn đề tài

Bản thân em là một người thích chơi game từ nhỏ, và như bao đứa trẻ khác thì em đã luôn có mong muốn được tự tay tạo ra một tựa game của riêng mình, mang đậm phong cách của mình. Thấy rằng những công cụ trong ngành công nghiệp làm game hiện nay đều đang ở mức đỉnh cao của nó, em đã nghĩ là đây chính là thời cơ để có thể biến ước mơ bấy lâu của bản thân thành hiện thực. Việc nghiên cứu và tạo nên con game đầu tay này sẽ giúp em có thể rút ra các kinh nghiệm và trải nghiệm việc làm game, điều này sẽ giúp em có thể tạo ra những tựa game chất lượng và thú vị hơn trong tương lai.

## Mục tiêu và phạm vi đề tài

Tạo ra được một game hầm ngục 3D. Game sẽ cho phép người chơi thực hiện các tác vụ như xoay camera và di chuyển để khám phá, tấn công quái, hồi máu, né, đỡ đòn, phản đòn. Bản đồ sẽ là một mê cung khổng lồ với rất nhiều quái để tạo nên thử thách cho người chơi. Ngoài ra thì còn phát triển hệ thống AI điều khiển nhiều loại kẻ thủ với đa dạng các trạng thái để khiến cho kẻ thù trở nên sinh động, đa dạng hơn và bớt sự nhàm chán. Cuối cùng là đảm bảo đồ họa đẹp, giao diện không bị rối và điều khiển không quá phức tạp nhằm nâng cao trải nghiệm của người chơi.

Game hầm ngục này sẽ là game một người chơi, nó sẽ tập trung vào yếu tố khám phá hầm ngục và việc giao chiến với quái, nhắm đến các đối tượng là người chơi trẻ, người chơi thích các loại game đánh quái hoặc người chơi yêu thích thể loại game khám phá mê cung hầm ngục.

## Phương pháp nghiên cứu

Ngành công nghiệp làm game hiện nay đang vô cùng phát triển. Nó đang trở nên phổ biến và dễ tiếp cận hơn đối với những lập trình viên và những người có đam mê làm game. Nguyên do chính dẫn đến việc này là vì các công cụ làm game hiện nay ngày càng đa dạng và đa phần đều không quá khó để bắt đầu học cách sử dụng chúng. Một vài những cái tên phổ biến có thể kể đến là Unity, Unreal Engine, RPG Maker, Godot, Ren’Py, GameMaker và còn vô vàn các công nghệ khác nữa.

Sau khi tìm hiểu các công cụ trên, em nhận ra rằng RPG Maker và Ren’Py không phải là công cụ phù hợp cho đề tài mà em đã lựa chọn. RPG Maker có rất nhiều thành phần hỗ trợ việc xây dựng hầm ngục dạng game nhập vai nhưng chỉ hỗ trợ việc lập trình game 2D, còn Ren’Py chỉ hỗ trợ các tựa game dạng tiểu thuyết 2D, vì thế nên hai công cụ này sẽ không thể đáp ứng được mục tiêu của đề tài [1,2]. GameMaker và Godot là hai công cụ vừa hỗ trợ làm game 2D và làm game 3D, nhưng chúng chỉ được chọn để làm game 2D là chính do hỗ trợ làm game 3D còn nhiều hạn chế [3,4]. Vì thế nên cũng không thể lựa chọn hai công cụ này.

Unity và Unreal Engine đã luôn là hai “ông lớn” trong dàn các công cụ làm game, cả hai đều được tin dùng bởi từ những lập trình viên đơn lẻ hay các nhóm nhỏ đến cả những công ty lớn. Cả hai đều hỗ trợ việc làm game 2D và 3D cực kì mạnh mẽ, có cả một thư viện khổng lồ chứa vô vàn thành phần có thể sử dụng trong project và một cộng đồng người dùng đông đảo sẵn sàng hỗ trợ khi gặp khó khăn. Điểm trừ của Unity khi so với Unreal Engine nằm ở chất lượng đồ họa mà cả hai công cụ này có thể cung cấp. Đồ họa của Unity tuy tốt những vẫn không thể so sánh được với đồ họa của Unreal Engine, khi nó có vô vàn các công nghệ hỗ trợ từ ánh sáng đến kết xuất hình ảnh chất lượng cao [4].

Sau khi nghiên cứu về các công cụ làm game phổ biến nhất trên thị trường hiện tại, em cho rằng Unreal Engine chính là sự lựa chọn phù hợp nhất để đạt được các mục tiêu đã đưa ra của đề tài.

## Bố cục báo cáo

Tổng quan đề tài: Chương này cho biết tình hình chung của ngành làm game hiện nay, và từ đó đưa ra đề tài và lý do lựa chọn đề tài đó của em. Sau đó chương sẽ nói về mục tiêu và phạm vi mà đề tài này bao phủ, và cuối cùng là phân tích, nghiên cứu các công nghệ hiện đang có trên thị trường để chọn ra công nghệ thích hợp nhất cho đồ án này.

Cơ sở lý thuyết: Chương này cung cấp nền tảng lý thuyết cần thiết cho công nghệ em đã lựa chọn là Unreal Engine. Trong chương này sẽ giải thích rằng Unreal Engine là gì, những lĩnh vực ứng dụng nó, những thành tựu nó đã đạt được. Sau đó sẽ đi sâu thêm vào kiến trúc của Unreal Engine và những thứ mà nó cung cấp cho người dùng. Cuối cùng là đi vào cụ thể cách vận hành một project Unreal Engine cơ bản.

Hệ thống game hầm ngục 3D: Chương này sẽ mô tả cách thức vận hành của hệ thống game. Nó sẽ bao gồm những quy luật chung của game. Ngoài ra sẽ có những thông tin như các lược đồ để miêu tả cách hoạt động của các chức năng chủ chốt, cũng như cách hệ thống hoạt động một cách tổng thể và cuối cùng là những kết quả đã thực hiện được.

Kết luận và hướng phát triển: Chương này sẽ tổng hợp lại các kết quả đã đạt được trong quá trình thực hiện nghiên cứu. Nó sẽ bao gồm những mục tiêu đã đạt được và những điều còn cần phải cải thiện của hệ thống và cá nhân em, tiếp sau đó sẽ là phương hướng phát triển cho đồ án này trong tương lai.

# CƠ SỞ LÝ THUYẾT

## Unreal Engine

### Giới thiệu Unreal Engine

Unreal Engine (UE) là một nền tảng lập trình thời gian thực 3D được phát triển bởi Epic Games và được ứng dụng trong nhiều lĩnh vực như phát triển game, phim ảnh, kiến trúc hay thậm chí là một số các lĩnh vực không đòi hỏi tính sáng tạo khác.

A black and white logo

Description automatically generated

Hình .. Logo Unreal Engine (nguồn: https://www.unrealengine.com)

UE được tích hợp rất nhiều các công cụ mạnh mẽ giúp cho việc biến những ý tưởng trở thành hiện thực đơn giản hơn nhiều lần. Đặc biệt, công cụ này không quá đòi hỏi người sử dụng phải có nhiều kinh nghiệm trong lĩnh vực lập trình, khiến cho nó dễ dàng tiếp cận với đa phần người dùng hơn.

Tính đến bây giờ, UE đã được sử dụng để làm nên rất nhiều tựa game nổi tiếng toàn cầu, có thể kể đến như “Ark: Survival”, “Fortnite”, “Only Up!”, “Dead by Daylight”, hay mới gần đây nhất là siêu phẩm game “Black Myth: Wukong” đã phá đi vô vàn các kỉ lục về người chơi trên nền tảng Steam. Ngoài game ra thì UE cũng đã được ứng dụng trong những bộ phim bom tấn, có thể kể đến một vài cái tên tiêu biểu như “Jurassic Park” hay “Independence Day”. Vào năm 2014, UE đã được Kỷ lục Thế giới Guinness công nhận là “Nền tảng lập trình game thành công nhất thế giới” [5].

Thế hệ mới nhất của UE hiện tại là Unreal Engine 5, và phiên bản mới nhất đã ra mắt của UE là Unreal Engine 5.4. UE5 ở bản 5.4 đã có thêm nhiều nâng cấp vô cùng mãnh mẽ cho các tác vụ hoạt ảnh, kết xuất, AI và máy học, các tính năng hỗ trợ cho các lập trình viên và nhiều hơn thế nữa [6].

### Kiến trúc của Unreal Engine

UE cung cấp rất nhiều các kiến trúc mạnh mẽ hỗ trợ cho công việc làm game, từ những thao tác lập trình logic hay đến cả những việc như điều chỉnh ánh sáng, chất lượng hình ảnh, âm thanh, tương tác giữa các vật thể và quản lý các thành phần trong một project:

* Trình chỉnh sửa (Editor): Đây là giao diện làm việc chính trong quá trình làm game hay ứng dụng. Nó bao gồm nhiều công cụ hỗ trợ cho các công việc như lập trình logic game, thiết kế môi trường, chỉnh sửa mô hình, điều chỉnh ánh sáng và vật lý, và còn nhiều hơn thế nữa. Một vài tab phổ biến có thể kể đến như Level Editor, nơi người lập trình viên sẽ thiết kế trò chơi của mình một cách trực quan và Blueprint Editor, nơi mà phần lớn việc lập trình logic của game sẽ diễn ra.
* Blueprint (Scripting System): Blueprint là hệ thống lập trình trực quan của UE. Blueprint hoạt động theo phương thức kéo, thả và nối các nút (node) có sẵn lại với nhau để tạo nên logic cho game. Với việc lập trình trực quan và đã có sẵn đa dạng các loại hàm, kể cả những người không biết hay không có nhiều kinh nghiệm với C++ cũng có thể sử dụng UE để làm game. Tuy nhiên, UE vẫn hỗ trợ kết hợp sử dụng Blueprint với mã C++ nếu cần.

A screenshot of a computer

Description automatically generated

Hình .. Trình chỉnh sửa Blueprint (nguồn: https://dev.epicgames.com)

* C++ API: UE được xây dựng trên ngôn ngữ lập trình C++ và nó cung cấp cho các lập trình viên một API. Các lớp, hàm và các thành phần quan trọng khác đều được viết bằng C++ nên nó mang lại khả năng mở rộng vô cùng linh hoạt. Vì thế, người lập trình có thể tự do tạo ra các chức năng tùy chỉnh sâu hơn những gì mà Blueprint không đạt đến được.
* Gameplay Framework: Gameplay Framework bao gồm các chức năng cốt lõi của game, giúp việc quản lý và tổ chức các yếu tố trong game hiệu quả hơn, bao gồm Actors (Những thực thể tương tác được trong game như nhân vật hay vật dụng), Pawn (Lớp con của Actor, đại diện cho những đối tượng người chơi có thể điều khiển được), Character (Lớp con của Pawn, nơi để thêm vào các chức năng di chuyển hay animation), Controller (Điều khiển các Actor và Pawn) và Game Mode (Quản lý các quy tắc hay điều luật của game).

A white square with black text

Description automatically generated

Hình .. Luồng hoạt động của Gameplay Framework khi chạy một project. Hình chi tiết luồng hoạt động ở phần phụ lục.

* Rendering Engine: UE có một hệ thống kết xuất đồ họa vô cùng mạnh mẽ và tiên tiến, nó có thể kết xuất được trong thời gian thực với chất lượng hình ảnh vô cùng cao. Một số tính năng có thể kể đến như Lumen (Công nghệ chiếu sáng toàn cục thời gian thực), Nanite (Hệ thống hình học vi mô giúp hiển thị chi tiết và không bị giới hạn bởi số lượng đa giác) và Post-Processing (Hỗ trợ các hiệu ứng hình ảnh như Bloom, Depth of Field, Motion Blur và nhiều hiệu ứng khác). Nanite Virtualized Geometry của Unreal Engine 5 cho phép các vật thể chi tiết hơn nhiều so với trước đây. Còn Lumen thì đạt được độ phân giải tối đa, thực hiện các thao tác tính toán ánh sáng gián tiếp ở độ phân giải thấp hơn để tăng hiệu suất cho việc thực hiện tính toán thời gian thực.

A cave with a light shining through

Description automatically generated with medium confidence

Hình .. Một ví dụ cho việc sử dụng Lumen để thắp sáng môi trường (nguồn: https://dev.epicgames.com)

* Physics and Collision System: Bắt đầu từ UE5, UE sử dụng công nghệ Chaos Physics để mô phỏng vật lý thực tế, bao gồm Rigid Body Dynamics (Mô phỏng chuyển động của các vật thể cứng), Cloth Physics (Mô phỏng chuyển động của vải và vật liệu mềm), Collision Detection (Xử lý va chạm giữa các thực thể trong game).
* Audio System: UE hỗ trợ đa dạng các tính năng về âm thanh bằng cách cung cấp các công cụ xử lý âm thanh một cách vô cùng chi tiết và đồng bộ với các sự kiện diễn ra trong game.
* AI System: UE cung cấp một hệ thống AI vô cùng mạnh mẽ bao gồm Navigation System (Điều hướng AI), Behavior Tree (Quản lý hành vi của AI) và Perception System (Cho AI khả năng cảm nhận thế giới trong game thông qua các giác quan).

A screenshot of a computer

Description automatically generated

Hình .. Một cây quyết định (Behavior Tree) đơn giản trong Unreal Engine (nguồn: https://dev.epicgames.com)

A group of people running on a green surface

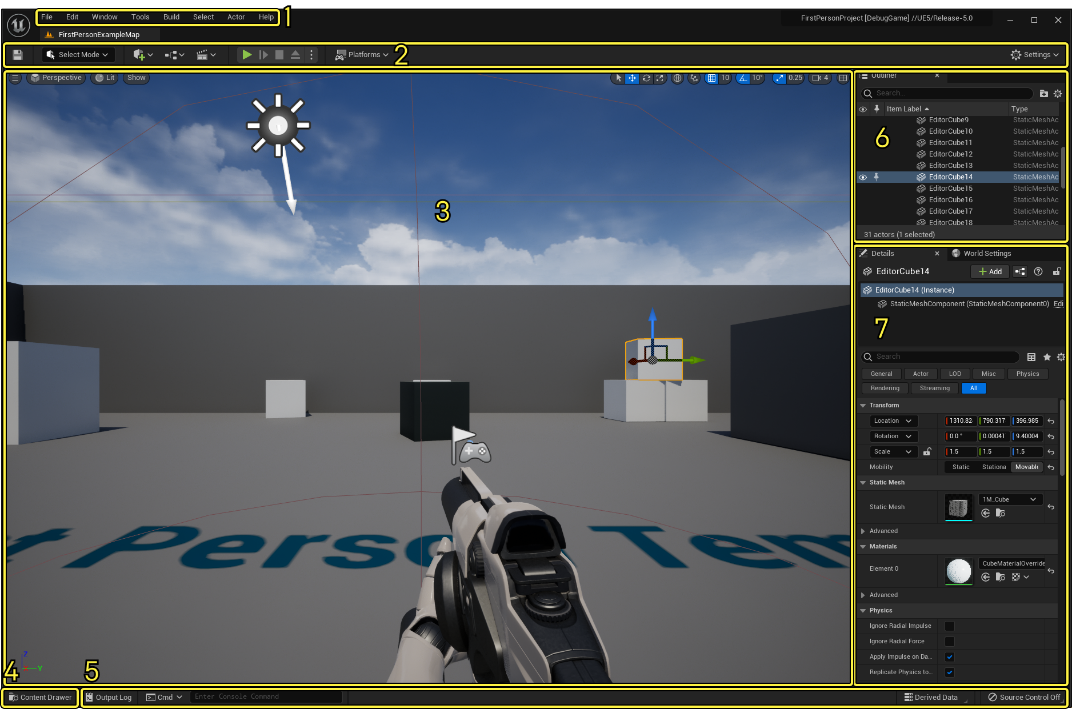
Description automatically generated

Hình .. Hệ thống định vị (Navigation System) để lựa chọn vị trí AI có thể di chuyển đến (nguồn: https://dev.epicgames.com)

* Networking and Multiplayer: UE cũng hỗ trợ phát triển game đa người chơi, và nó sử dụng mô hình client-server.
* Asset Management: UE cung cấp hệ thống quản lý tài nguyên giúp quản lý và lưu trữ các đối tượng, âm thanh, texture, vật liệu, và các thành phần khác.

### Cấu trúc một project Unreal Engine

Sau khi cài đặt UE và chạy ứng dụng, người dùng sẽ vào một trang cho phép họ chọn project để mở. Người dùng có thể tạo một project mới hoàn toàn, hoặc tạo nên từ một mẫu nào đó, hoặc mở lại các project cũ. Sau khi mở project, người dùng sẽ vào trình chỉnh sửa của project. Trình chỉnh sửa của bất kì project nào cũng sẽ có những thành phần chính là Menu Bar, Main Toolbar, Level Viewport, Content Browser, Bottom Toolbar, Outliner và Details panel. Bố trí của chúng như hình sau:



Hình .. Giao diện trình chỉnh sửa một project (nguồn: https://dev.epicgames.com)

Mô tả từng phần (kèm đánh số) trong trình chỉnh sửa:

* Menu Bar (1): Chứa các menu để truy cập các lệnh và chức năng cụ thể của editor.
* Main Toolbar (2): Main Toolbar chứa các công cụ và lệnh phổ biến và được sử dụng nhiều, chúng bao gồm Save, Mode Selection (Lựa chọn các chế độ chỉnh sửa nội dung), Content Shortcuts (Tạo các Asset thường dùng, tạo Blueprint), Play Mode Controls (Nơi để bắt đầu chạy game và những chức năng khác liên quan), Platforms Menu (Hỗ trợ việc chuẩn bị, cấu hình và triển khai project lên các nền tảng khác nhau như mobile, desktop hay mobile) và Settings.
* Level Viewport (3): Level Viewport là nơi làm việc chính trong việc thiết kế game, nó hiển thị tất cả các thành phần của Level đang mở. Đây là nơi bạn sẽ xem và chỉnh sửa các thành phần của Level. Bạn có thể di chuyển camera bên trong Level đang mở trong Vỉewport bằng chuột và bàn phím.
* Content Browser (4): Là một trình quản lý tập tin chứa và hiển thị tất cả các Asset, Blueprint và tất cả những tập tin khác trong project. Bạn có thể dùng nó để tìm các thành phần trong project, kéo chúng vào trong Level Viewport, chuyển các thành phần qua lại giữa các project và nhiều chức năng khác.
* Bottom Toolbar (5): Bottom Toolbar bao gồm các phím tắt đến các chức năng Command Console, Output Log, Derived Data và cũng đồng thời hiển thị tình trạng của Source Control (ví dụ như GitHub, hoặc nếu không có thì là Off).
* Outliner (6): Outliner hiển thị một cách phân cấp tất cả các thành phần trong màn chơi của bạn, ví dụ như các vật thể, nhân vật hay thậm chí là ánh sáng. Bạn có thể chọn các thành phần trong Outliner để xem và điều chỉnh chúng ở trong Level Viewport
* Details panel (7): Khi bạn chọn một Actor trong Level Viewport hay Outliner, tất cả những thiết lập và thuộc tính ảnh hưởng đến actor đó sẽ được hiển thị lên tại đây

# HỆ THỐNG GAME HẦM NGỤC 3D

## Giới thiệu bài toán

Game sẽ có các chức năng cho phép người chơi di chuyển và xoay camera để khám phá chi tiết bản đồ game mê cung hầm ngục. Ngoài ra cũng sẽ có các chức năng liên quan đến chiến đấu với quái như tấn công, đỡ đòn, phản đòn, né và hồi phục. Người chơi sẽ có hai dạng tấn công, một dạng sử dụng kiếm và khiên, dạng còn lại sử dụng phép thuật.

Game sẽ có ba loại kẻ thù bình thường để tăng độ đa dạng khi đánh quái trong quá trình chơi game là kẻ thù đánh gần, kẻ thù đánh xa và kẻ thù pháp sư. Ngoài kẻ thù bình thường, game sẽ có một trùm cuối mạnh hơn cả với đa dạng chiêu thức để thật sự thánh thức kĩ năng của người chơi.

Kẻ thù sẽ được điều khiển bằng một hệ thống AI, làm cho kẻ thù thông minh và trông “thực” hơn. Kẻ thù cần được tích hợp 3 giác quan là thị giác, thính giác và phản xạ khi bị tấn công. Hành động của kẻ thù sẽ phụ thuộc vào trạng thái của chúng, có tổng cộng sáu trạng thái là trạng thái tuần tra (hay trạng thái bị động), trạng thái tìm hiểu, trạng thái tấn công, trạng thái truy lùng, trạng thái bất động và trạng thái chết, trạng thái mặc định của kẻ thù là trạng thái tuần tra. Đối với trùm cuối, nó chỉ có trạng thái tuần tra và tấn công, một khi nó đã phát hiện người chơi, nó sẽ không bao giờ mất tầm nhìn của người chơi. Với việc có đa dạng kẻ thù và nhiều trạng thái cho kẻ thù sẽ tăng độ khó của trò chơi và khiến trải nghiệm chơi được nâng cao.

Bản đồ game sẽ được xây dựng dưới dạng một mê cung khổng lồ, và sẽ có kẻ thù đi tuần tra ở một số tuyến đường nhất định. Người chơi phải thám hiểm mê cung, tìm ra được con đường dẫn đến sàn đấu trùm cuối. Ngoài những kẻ thù tuần tra đơn lẻ thì sẽ có những trại kẻ thù với số lượng kẻ thù đông hơn, gia tăng độ khó cho game.

Điều kiện để xem là chiến thắng chính là đánh bại được trùm cuối. Ngược lại nếu để kẻ thù đánh bại, hay có thể hiểu là thanh máu về không, thì người chơi sẽ thua và có thể lựa chọn giữa việc thoát game hoặc hồi sinh chơi lại từ đầu.

## Phân tích hệ thống

### Các actor

Actor trong hệ thống là người chơi. Người chơi sẽ tương tác với hệ thống game qua các thao tác chuột, bàn phím và sẽ nhận được phản hồi từ hệ thống thông qua hệ thống âm thanh (loa, tai nghe) và màn hình.

### Các usecase

* Usecase “Bắt đầu game”: Khi người chơi bắt đầu game, hệ thống sẽ thực hiện “Tạo bản đồ”, sau đó sẽ thực hiện “Khởi tạo kẻ thù” và những kẻ thù được tạo ra sẽ được hệ thống AI “chiếm quyền điều khiển” và bắt đầu điều khiển chúng. Sau đó, người chơi sẽ bắt đầu trò chơi ở vị trí xuất hiện đã được định sẵn.
* Usecase “Xoay camera”: Người chơi dùng chuột để xoay camera xung quanh.
* Usecase “Di chuyển nhân vật”: Actor cùng bàn phím để di chuyển nhân vật của mình. Các di chuyển bao gồm việc đi, chạy, nhảy và dịch chuyển.
* Usecase “Thực hiện đòn tấn công”: Nếu người sử dụng (người chơi và kẻ thù) đang trong trạng thái tấn công, thực hiện đòn tấn công tương ứng. Nếu đánh trúng mục tiêu sẽ tiến hành tính toán thanh máu của mục tiêu. Người chơi có một combo tấn công, kẻ thù có nhiều loại tấn công.
* Usecase “Hồi phục máu”: Người sử dụng (người chơi và kẻ thù) hồi phục máu đã mất của bản thân. Lượng máu hồi phục không vượt quá máu tối đa của người sử dụng. Khi sử dụng sẽ tính toán lại thanh máu của bản thân.
* Usecase “Đỡ đòn tấn công”: Nếu đòn tấn công có thể bị đỡ, người chơi chặn đòn tấn công của kẻ thù, bị đẩy lùi và không nhận sát thương. Khi đỡ, sẽ có một khung thời gian nhỏ mà nếu kẻ thù tấn công, nhân vật sẽ phản đòn thay vì đỡ đòn (nếu đòn tấn công đó có thể bị phản), gây choáng nhẹ và phản lại một phần sát thương lên kẻ thù, thanh máu của kẻ thù sẽ được tính toán lại.
* Usecase “Đổi trạng thái nhân vật”: Nhân vật của người chơi có ba trạng thái là bị động (mặc định), cận chiến và pháp sư. Người chơi dùng bàn phím để thay đổi trạng thái của nhân vật mình điều khiển.

### Use case toàn cục hệ thống

A diagram of a person with text

Description automatically generated

Hình .. Use case toàn cục hệ thống

### Đặc tả usecase

Hai chức năng chủ chốt và quan trọng nhất cho phép người chơi có thể điều khiển nhân vật chiến đấu là “Thực hiện đòn tấn công” và “Đỡ đòn tấn công”. Hai chức năng này tuy trông có vẻ đơn giản nhưng nó đòi hỏi người chơi cần có kĩ năng tốt trong việc điều khiển nhân vật và canh thời gian hợp lý.

#### Usecase thực hiện đòn tấn công

Bảng .. Đặc tả usecase “Thực hiện đòn tấn công”

|  |  |
| --- | --- |
| Use Case ID | UC-4 |
| Tên Use Case | Thực hiện đòn tấn công |
| Mô tả | Là người chơi, tôi muốn nhân vật trong game thực hiện đòn tấn công. |
| Actor | Người chơi |
| Trigger | Người chơi bấm nút tấn công (chuột trái). |
| Tiền điều kiện | * Trạng thái của nhân vật hiện tại phải là một trong các trạng thái tấn công. * Nhân vật phải còn sống (tức máu hiện tại phải lớn hơn 0) |
| Hậu điều kiện | * Chơi hoạt ảnh tấn công. * Tính toán máu của mục tiêu nếu đánh trúng mục tiêu. |
| Luồng hoạt động | Người chơi bấm chuột trái.  Hệ thống kiểm tra hợp lệ trạng thái của nhân vật.  Nếu nhân vật tấn công là của người chơi, hệ thống kiểm tra xem nhân vật có đang trong chuỗi thực hiện combo tấn công không.  Hệ thống chơi hoạt ảnh tấn công tương ứng cho nhân vật và tính toán thanh máu của mục tiêu nếu đánh trúng. |
| Luồng thay thế |  |
| Luồng ngoại lệ |  |

#### Usecase đỡ đòn tấn công

Bảng .. Đặc tả usecase “Đỡ đòn tấn công”

|  |  |
| --- | --- |
| Use Case ID | UC-6 |
| Tên Use Case | Đỡ đòn tấn công |
| Mô tả | Là người chơi, tôi muốn nhân vật trong game thực hiện đỡ đòn tấn công. |
| Actor | Người chơi |
| Trigger | Người chơi bấm nút đỡ đòn (chuột phải). |
| Tiền điều kiện | * Trạng thái của nhân vật hiện tại phải là trạng thái cận chiến. * Nhân vật phải còn sống (tức máu hiện tại phải lớn hơn 0) |
| Hậu điều kiện | * Chơi hoạt ảnh đỡ đòn. * Tính toán máu của mục tiêu nếu phản đòn thành công. |
| Luồng hoạt động | Người chơi bấm chuột phải.  Hệ thống kiểm tra hợp lệ trạng thái của nhân vật.  Hệ thống kiểm tra xem nhân vật có đang trong khung thời gian cho phép đỡ đòn không.  Hệ thống kiểm tra xem nhân vật có đang trong khung thời gian sử dụng phản đòn thay vì đỡ đòn không.  Hệ thống chơi hoạt ảnh đỡ đòn cho nhân vật, nếu phản đòn thành công thì làm chậm thời gian trong game một khoảng thời gian ngắn và tính toán thanh máu của mục tiêu đã tấn công nhân vật. |
| Luồng thay thế |  |
| Luồng ngoại lệ |  |

## Thiết kế hệ thống

### Lược đồ tuần tự

#### Chức năng thực hiện đòn tấn công

Các đối tượng tham gia quy trình của chức năng “Thực hiện đòn tấn công” bao gồm người chơi, nhân vật, hệ thống và kẻ thù. Tuần tự hoạt động của quy trình như sau:

Người chơi bấm nút tấn công.

Nhân vật yêu cầu hệ thống xác thực trạng thái sống hoặc chết của bản thân.

Nếu đã chết, hệ thống kết thúc quy trình. Nếu còn sống, hệ thống cho biết nhân vật còn sống và chờ yêu cầu tiếp theo.

Nhân vật yêu cầu hệ thống xác thực trạng thái của bản thân.

Nếu không phải là trạng thái tấn công thì hệ thống kết thúc quy trình. Nếu là trạng thái tấn công thì hệ thống chơi hoạt ảnh tấn công cho nhân vật.

Nhân vật tạo một vùng sát thương và khởi tạo sát thương của đòn tấn công.

Hệ thống kiểm tra xem có kẻ thù nào ở trong vùng sát thương không.

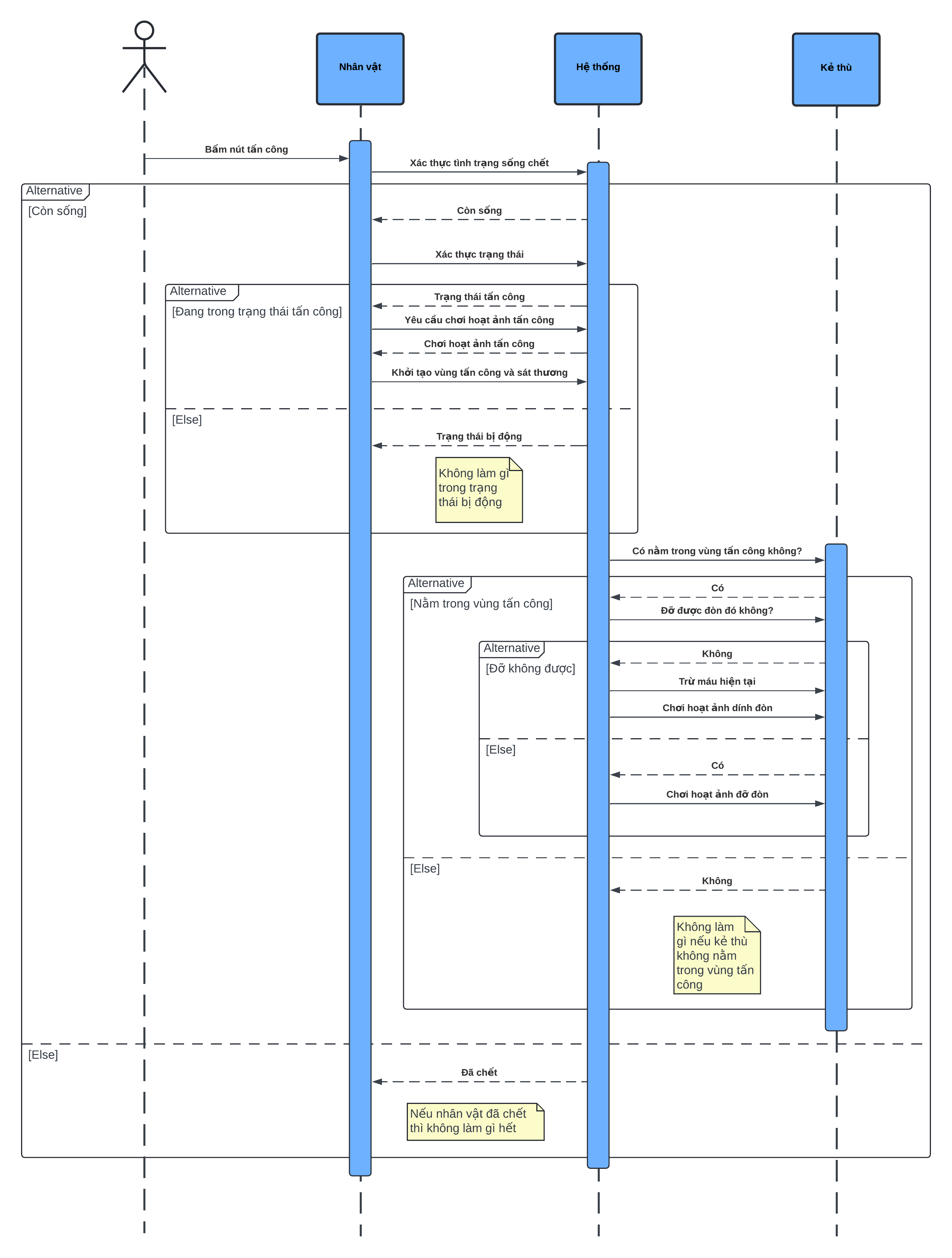
Kẻ thù trả lời lại là có hoặc không.

Nếu không có kẻ thù trong vùng sát thương thì quy trình kết thúc. Nếu có thì sẽ kiểm tra xem kẻ thù có đỡ được đòn tấn công đó không.

Kẻ thù trả lời là đỡ được hoặc không.

Nếu đỡ được, hệ thống sẽ chơi hoạt ảnh đỡ đòn cho kẻ thù và quy trình kết thúc. Nếu không, hệ thống sẽ gây sát thương lên kẻ thù và quy trình kết thúc.

Dưới đây là lược đồ tuần tự của quy trình “Thực hiện đòn tấn công”:



Hình .. Lược đồ tuần tự của chức năng “Thực hiện đòn tấn công”

#### Chức năng đỡ đòn tấn công

Các đối tượng tham gia quy trình của chức năng “Đỡ đòn tấn công” bao gồm người chơi, nhân vật, hệ thống và kẻ thù. Tuần tự hoạt động của quy trình như sau:

1. Người chơi bấm nút đỡ đòn.
2. Nhân vật yêu cầu hệ thống xác thực trạng thái sống hoặc chết của bản thân.
3. Nếu đã chết, hệ thống kết thúc quy trình. Nếu còn sống, hệ thống cho biết nhân vật còn sống và chờ yêu cầu tiếp theo.
4. Nhân vật yêu cầu hệ thống xác thực trạng thái của bản thân.
5. Nếu không phải là trạng thái cận chiến thì hệ thống kết thúc quy trình. Nếu là trạng thái cận chiến thì hệ thống chơi hoạt ảnh đỡ đòn cho nhân vật.
6. Hệ thống kiểm tra xem nhân vật có nằm trong vùng tấn công nào không.
7. Nhân vật trả lời lại là có hoặc không.
8. Nếu không, quy trình kết thúc. Nếu có, hệ thống kiểm tra xem đòn tấn công đó có đỡ được không.
9. Kẻ thù (đối tượng đã ra đòn tấn công) trả lời lại là có hoặc không.
10. Nếu không, hệ thống gây sát thương lên nhân vật và quy trình kết thúc. Nếu có, hệ thống sẽ kiểm tra xem nhân vật có đang trong khung thời gian cho phép đỡ đòn hay không.
11. Nhân vật trả lời lại là có hoặc không.
12. Nếu không, hệ thống gây sát thương lên nhân vật và quy trình kết thúc. Nếu có thì hệ thống sẽ kiểm tra xem đòn tấn công đó có thể phản lại được không.
13. Kẻ thù trả lời lại là có hoặc không.
14. Nếu không, hệ thống chơi hoạt ảnh đỡ đòn thành công cho nhân vật và quy trình kết thúc. Nếu có, hệ thống sẽ kiểm tra xem nhân vật có đang trong khung thời gian cho phép phản đòn hay không.
15. Nhân vật trả lời lại là có hoặc không.
16. Nếu không, hệ thống chơi hoạt ảnh đỡ đòn thành công cho nhân vật và quy trình kết thúc. Nếu có, hệ thống sẽ gây sát thương và làm choáng kẻ thù, sau đó quy trình kết thúc.

Dưới đây là lược đồ tuần tự của quy trình “Đỡ đòn tấn công”:

A diagram of a project

Description automatically generated with medium confidence

Hình .. Lược đồ tuần tự của chức năng “Đỡ đòn tấn công”

### Lược đồ hoạt động

#### Chức năng thực hiện đòn tấn công

Sau khi nhận được tín hiệu tấn công từ người chơi, nhân vật sẽ cần phải được kiểm tra trạng thái phù hợp trước khi thực hiện tấn công. Sau khi đã xác thực trạng thái, hệ thống cho nhân vật thực hiện hoạt ảnh tấn công, cùng lúc đó nhân vật tạo ra một vùng tấn công về hướng bản thân đã tấn công. Vùng tấn công này có thể hiểu là một khu vực mà khi có thứ gì đó đứng trong tầm thì sẽ nhận sát thương của đòn tấn công mà một đối tượng đã tạo ra. Tiếp sau đó hệ thống sẽ kiểm tra xem kẻ thù có nằm trong vùng tấn công và có đỡ đòn tấn công của nhân vật người chơi hay không để xử lý tương ứng. Nếu không sẽ kết thúc việc tấn công. Nếu có sẽ kiểm tra xem kẻ thù đó có đỡ được đòn đó hay không (việc đỡ được hay không phụ thuộc vào may mắn và đòn tấn công đó có đỡ được không). Nếu đỡ được thì sẽ chơi hoạt ảnh đỡ đòn cho kẻ thù, nếu không thì gây sát thương lên kẻ thù.

Dưới đây là lược đồ hoạt động của chức năng “Thực hiện đòn tấn công”:

A diagram of a flowchart

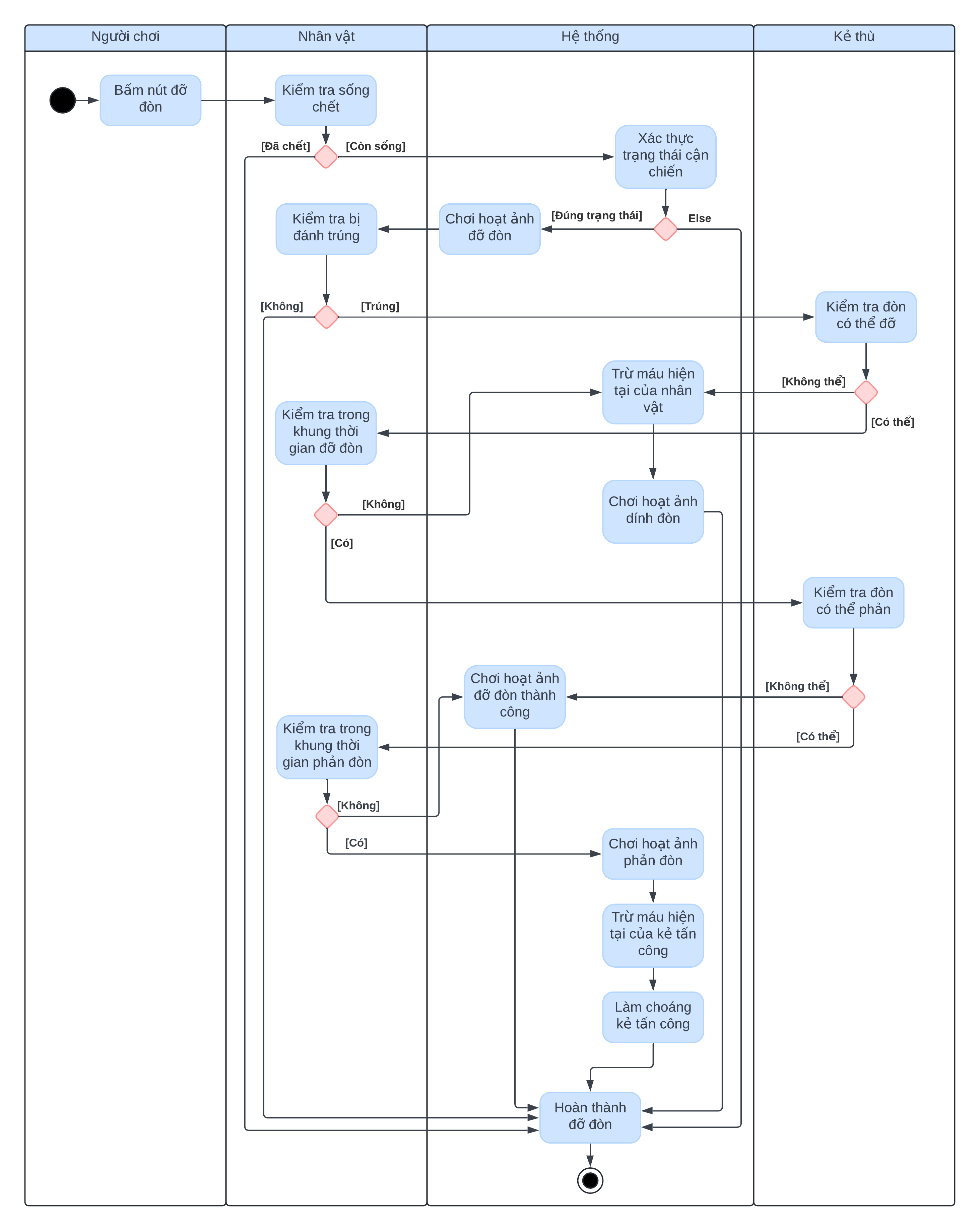
Description automatically generated

Hình .. Lược đồ hoạt động của chức năng “Thực hiện đòn tấn công”

#### Chức năng đỡ đòn tấn công

Khi người chơi ra tín hiệu đỡ đòn tấn công, hệ thống sẽ xác thực trạng thái của nhân vật trước khi thực hiện thao tác đỡ. Sau đó hệ thống sẽ kiểm tra xem người chơi có đang ở trong vùng tấn công nào không. Nếu không thì kết thúc việc đỡ đòn. Nếu có, hệ thống sẽ kiểm tra một loạt điều kiện để có thể xử lý tương ứng: Kiểm tra xem đòn tấn công có thể đỡ được không. Nếu không, nhân vật sẽ nhận sát thương. Nếu đỡ được, tiếp tục kiểm tra xem có đang trong thời gian đỡ hay không. Nếu không, nhân vật nhận sát thương. Nếu đang trong thời gian đỡ, tiếp tục kiểm tra xem đòn tấn công có thể phản lại không. Nếu không, nhân vật sẽ chỉ đỡ đòn. Nếu có thể phản, kiểm tra xem có đang trong thời gian phản không. Nếu không, nhân vật sẽ chỉ đỡ đòn. Nếu có, nhân vật sẽ phản lại một phần sát thương và gây choáng kẻ thù.

Dưới đây là lược đồ hoạt động của chức năng “Đỡ đòn tấn công”:



Hình .. Lược đồ hoạt động của chức năng “Đỡ đòn tấn công”.

### Thiết kế dữ liệu

Cấu trúc dữ liệu chính của game chỉ sử dụng các cấu trúc dữ liệu nội bộ mà không sử dụng cơ sở dữ liệu ngoài để quản lý thông tin liên quan đến nhân vật người chơi, kẻ địch, các trạng thái, đòn tấn công, chiêu thức. Các đối tượng trong game như nhân vật người chơi, kẻ địch, và trùm cuối được quản lý thông qua các blueprint trong hệ thống.

Đối với các đối tượng sống trong game thì có hai blueprint chính để quản lý là nhân vật người chơi và khuôn mẫu kẻ thù. Các loại kẻ thù khác nhau sẽ đều được kế thừa từ lớp cha là lớp khuôn mẫu kẻ thù. Hai blueprint đó có điểm giống nhau nằm ở việc đều có thanh máu, đều có nhiều trạng thái (các trạng thái của hai lớp nhân vật là khác nhau, nhưng cách thức lưu trữ thì giống).

Đối với các đòn tấn công và chiêu thức thì có blueprint khuôn mẫu các đòn tấn công thường, blueprint khuôn mẫu các đòn tấn công tầm xa, blueprint khuôn mẫu các chiêu thức và một blueprint các đòn tấn công giúp quản lý tấn cả đòn tấn công trong game. Các blueprint khuôn mẫu, như tên của nó, chỉ cung cấp một khuôn mẫu để tạo ra những đòn tấn công tương ứng. Chỉ một mình chúng thì không thể tạo nên một đòn tấn công. Để tạo ra một đòn tấn công mà đối tượng sống trong game có thể dùng được cần phải tạo ra các đòn tấn công từ các blueprint khuôn mẫu trên, và các đòn tấn công sẽ được quản lý bởi blueprint các đòn tấn công. Những thông tin được lưu trữ cho các đòn tấn công có thể kể đến như sát thương, loại tấn công, loại phản ứng khi dính đòn, có thể đỡ/phản không,…

Bảng .. Struct DamageInfo, được dùng để lưu các thông tin của một đòn tấn công hoàn chỉnh

|  |  |
| --- | --- |
| Tên thuộc tính | Kiểu dữ liệu |
| Lượng sát thương (Amount) | Float |
| Loại sát thương (DamageType) | Enum DamageType |
| Loại phản ứng khi nhận sát thương (DamageRespond | Enum DamageRespond |
| Có thể tấn công khi đang trong trạng thái bất bại không (ShouldDamageInvincible) | Boolean |
| Có thể đỡ không (CanBeBlocked) | Boolean |
| Có thể phản không (CanBeParried) | Boolean |
| Có thể cưỡng ép kết thúc đòn tấn công khác không (ShouldForceInterupt) | Boolean |

Về bản đồ chơi, bản đồ là một mê cung hầm ngục đã được thiết kế sẵn và se không được tạo ngẫu nhiên mỗi lần chơi, nghĩa là mỗi lần chơi đều sẽ có bản đồ và bố trí kẻ thù là như nhau. Tuy nhiên, trong công đoạn thiết kế, em cũng đã đảm bảo bản đồ tuy không tạo mới mỗi lần nhưng vẫn sẽ mang lại độ khó nhất định cho người chơi nhiều lần.

Về trạng thái của trò chơi, trò chơi sẽ không hỗ trợ việc lưu giữa chừng, nghĩa là phải chơi một lần một mạng đến khi đánh với trùm cuối.

### Thiết kế giao diện

Sau khi thử nghiệm qua nhiều mẫu giao diện khác nhau, em quyết định thực hiện loại giao diện đơn giản, không che quá nhiều tầm nhìn của người chơi. Giao diện sẽ chỉ có một thanh máu màu xanh lá ở góc dưới bên trái màn hình, khi bắt đầu trận đấu với trùm cuối sẽ có thêm một thanh máu màu đỏ cũng như tên của trùm cuối ở ngay chính giữa phía trên màn hình.

A robot standing on a tile floor

Description automatically generated

Hình .. Giao diện lúc bình thường

A screenshot of a video game

Description automatically generated

Hình .. Giao diện khi đánh boss

Thiết kế giao diện điều khiển nhân vật như trên giúp làm tăng góc nhìn của người chơi, người chơi có thể nhìn thấy nhiều hơn và cũng thoải mái mắt hơn khi không có quá nhiều thứ hiển thị lên màn hình.

Em cũng đã thiết kế thêm hai giao diện menu sẽ hiển thị khi người chơi đánh bại trùm cuối và khi người chơi bị hạ gục. Hai menu này sẽ được hiển thị khi đúng điều kiện của chúng. Việc thêm menu này sẽ làm đơn giản hóa hơn thao tác của người chơi khi muốn chơi lại hay thoát game, và cũng làm rõ ràng hơn đâu là kết thúc của một màn chơi.

A screenshot of a video game

Description automatically generated

Hình .. Menu chiến thắng khi đánh bại trùm cuối

A screenshot of a video game

Description automatically generated

Hình .. Menu game over khi bị kẻ địch đánh bại

## Kiến trúc hệ thống

### Tổng quan

Các dịch vụ của Unreal Engine (khung màu xanh dương) bao gồm những dịch vụ, công cụ và kiến trúc mà UE đã cung cấp sẵn, ví dụ như Blueprint, Gameplay Framework hay những thành phần cơ bản để tạo nên một hệ thống AI. Sử dụng những dịch vụ này, em đã xây dựng nên những hệ thống để quản lý các tác vụ và các đối tượng sống trong game. Những hệ thống đó là hệ thống sát thương, hệ thống di chuyển và hệ thống AI. Các đối tượng sống trong game bao gồm nhân vật do người chơi điều khiển và kẻ thù.

Nhân vật do người chơi điều khiển (người que màu xanh dương) là nhân vật chính của trò chơi, từ đây sẽ được gọi vắn tắt là nhân vật. Người chơi sẽ tương tác với thế giới trong game qua nhân vật này.

Kẻ thù (người que màu đó) là tổng hợp của tất cả các loại kẻ thù có trong game. Hiện tại có 4 loại kẻ thù là kẻ thù cận chiến, kẻ thù đánh xa, kẻ thù pháp sư và trùm cuối.

Hệ thống sát thương (khung màu đỏ) là hệ thống quản lý việc chiến đấu trong game. Nó bao gồm việc quản lý các đòn tấn công, quản lý thanh máu, và quản lý các vũ khí có trong game. Các đòn tấn công trong game bao gồm đòn tấn công cơ bản (vung kiếm, bắn súng,…), các đòn tấn công đặc biệt (đập đất, combo tấn công,…) và những chiêu thức ma thuật (cầu ma thuật, triệu hồi sấm sét…). Quản lý thanh máu sẽ theo dõi và cập nhật thanh máu cho nhân vật và kẻ thù, nó cũng sẽ theo dõi trạng thái sống hoặc chết của nhân vật và kẻ thù để có thể xử lý tương ứng. Các vũ khí hiện có trong game bao gồm kiếm, khiên, súng và rìu.

Hệ thống di chuyển (khung màu cam) là hệ thống quản lý việc di chuyển bên trong game. Nó bao gồm hai việc là quản lý các di chuyển bình thường và quản lý các di chuyển đặc biệt. Các di chuyển bình thường ở đây là đi tới, đi lùi, sang trái, sang phải, tổ hợp của các di chuyển trên và nhảy. Các di chuyển đặc biệt là những loại hình di chuyển còn lại, trong trường hợp của game là dịch chuyển.

Hệ thống AI của game (khung màu tím) là hệ thống quản lý tất cả hành vi của các kẻ thù trong game. Hai nhiệm vụ chính của nó là theo dõi cũng như thay đổi giữa các trạng thái của kẻ thù và từ đó thực hiện các hành động tương ứng cho từng trạng thái.

A diagram of a person's diagram

Description automatically generated

Hình .. Kiến trúc hệ thống game

Sau khi bắt đầu game, người chơi sẽ có thể điều khiển nhân vật của mình. Tất cả những kẻ thù có trong game sẽ được hệ thống AI “chiếm lấy” và điều khiển chúng thực hiện các hành động đã định sẵn.

Cả nhân vật và kẻ thù đều sử dụng hệ thống di chuyển và hệ thống sát thương. Hệ thống di chuyển cho phép nhân vật và kẻ thù đi lại xung quanh bản đồ game. Hệ thống sát thương cho phép nhân vật và kẻ thù thực hiện các đòn tấn công. Nhân vật và kẻ thù sẽ tương tác với nhau (gây sát thương, nhận sát thương, đỡ đòn, phản đòn) và với bản thân (hồi phục) thông qua hệ thống sát thương.

### Hệ thống di chuyển

Để tăng độ đa dạng trong tốc độ di chuyển cho kẻ thù và người chơi, em đã tạo ra một enum để lưu trữ các mức tốc độ di chuyển. Các mức tốc độ này có thể được gán giá trị trong Blueprint, điều này giúp cho việc dù có ở mức tốc độ giống nhau, nhưng những đối tượng sống khác nhau vẫn có thể có tốc độ di chuyển khác nhau, tùy vào ý định của người làm game. Các mức tốc độ hiện tại ở trong game là Idle (đứng yên), Walking (đi bộ), Jogging (đi bộ nhanh), Running (chạy bộ).

Bảng .. Enum MovementSpeed lưu các mức tốc độ di chuyển

|  |
| --- |
| Tên các mức tốc độ di chuyển |
| Idle |
| Walking |
| Jogging |
| Running |

Về khả năng dịch chuyển, một số kẻ thù và người chơi có khả năng dịch chuyển. Cơ chế dịch chuyển cơ bản là lấy phương hướng hiện tại đang di chuyển, và cộng một đoạn ngắn vào tọa độ vị trí đang đứng, ngoài ra trong lúc dịch chuyển sẽ không nhận sát thương.

Ví dụ: Trong hệ trục tọa độ Oxy, bạn đang ở vị trí là A(10, 5). Khi bạn bước về hướng tăng x và bấm nút dịch chuyển, giả sử khoảng cách dịch chuyển (do người làm game quy định sẵn) là 5, thì bây giờ bạn đang ở vị trí B(15, 5).

Trong game, đối với nhân vật người chơi thì phím Shift trái là phím dịch chuyển, và bạn chỉ có thể dịch chuyển khi nhân vật của bạn đang di chuyển về một hướng nào đó.

### Hệ thống sát thương

#### Loại sát thương và phản hồi khi nhận sát thương

Như đã nói ở phần “Thiết kế dữ liệu”, các thông tin của các đòn tấn công ở trong game được lưu trong một struct là DamageInfo. Ngoài struct đó ra thì em còn tạo thêm hai enum để hỗ trợ trong việc lưu trữ thông tin sát thương là DamageType (loại sát thương) và DamageRespond (phản hồi khi nhận sát thương).

Bảng .. Enum DamageType, lưu các loại sát thương

|  |
| --- |
| Tên các loại sát thương |
| None |
| Melee |
| Projectile |
| Explosion |
| Environment |

DamageType lưu các giá trị thể hiện các loại sát thương hay đòn tấn công khác nhau. Điều này là vì một đòn đánh cận chiến hiển nhiên sẽ khác với một đòn tấn công tầm xa hoặc đòn tấn công nổ. Việc lưu trữ dưới dạng enum sẽ giúp cho hệ thống sát thương dễ dàng mở rộng, và người làm game cũng có thể tùy chỉnh sâu hơn cho từng loại tấn công. Hiện tại trong game đang có 4 loại tấn công, None ở đây là giá trị mặc định, Melee là đòn tấn công cận chiến, Projectile là đòn tấn công tầm xa, Explosion là đòn tấn công nổ và Environment là đòn tấn công từ môi trường.

Bảng .. Enum DamageRespond, lưu các loại phản hồi khi nhận sát thương

|  |
| --- |
| Tên các phản hồi |
| None |
| HitReaction |
| Stagger |
| Stun |
| KnockBack |

DamageRespond lưu các giá trị thể hiện cách mà đối tượng sống sẽ phản hồi khi nhận sát thương. Tùy đòn tấn công mà sẽ có phản hồi tương ứng, và hiển nhiên ta không thể để hai loại sát thương khác nhau có cùng một phản hồi được. Hiểu một cách đơn giản hơn, ta không thể để cả sát thương cận chiến và sát thương nổ chỉ đẩy lùi nhẹ đối tượng được. Cũng như DamageType, việc dùng enum sẽ thuận tiện cho cả việc phát triển và mở rộng trong tương lai. Hiện tại trong game có 4 loại phản hồi, None là giá trị mặc định (nếu không tìm thấy phản hồi tương ứng do lỗi hoặc gõ sai tên thì sẽ về đây), HitReaction là phản hồi khi bị đánh trúng, Stagger là phản hồi gây choáng nhẹ, Stun là phản hồi gây choáng lâu và KnockBack là phản hồi gây đẩy lùi

#### Các đòn tấn công của người chơi

Người chơi có ba trạng thái là bị động (Unarmed), cận chiến (Melee) và pháp sư (Magic). Khi nhấn nút tấn công, hệ thống sẽ tùy thuộc vào trạng thái hiện tại của người chơi để xử lý chơi hoạt ảnh và thực hiện đòn tấn công tương ứng.

Bảng .. Enum PlayerStance, dùng để lưu các trạng thái của người chơi

|  |
| --- |
| Tên các trạng thái của người chơi |
| Unarmed |
| Melee |
| Magic |

Người chơi có một combo 3 đòn tấn công ở trạng thái cận chiến. Combo này có thực hiện được hay không tùy thuộc vào input từ người chơi, nếu người chơi canh thời gian bấm nút tấn công chuẩn xác, combo sẽ được tiếp tục.

A person in a blue garment holding a sword

Description automatically generated

Hình .. Combo tấn công cận chiến của người chơi

Khi đang ở trong trạng thái cận chiến, người chơi có thể thực hiện đỡ đòn tấn công. Khi canh chuẩn thời gian để bấm nút đỡ đòn, người chơi có thể chặn đòn tấn công của kẻ thù, hoặc tốt hơn nữa là phản lại đòn tấn công đó.

Ở trạng thái pháp sư, khi bấm nút tấn công, người chơi bắn ra một cầu ma thuật, gây sát thương lên mục tiêu đầu tiên mà nó va phải.

Điều khiển trong game cho các đòn tấn công của nhân vật người chơi như sau:

* Chuột trái: Tấn công
* Chuột phải: Đỡ đòn
* Phím 1: Đổi sang trạng thái cận chiến, nếu đã ở trong trạng thái cận chiến thì sẽ về trạng thái bị động
* Phím 2: Đổi sang trạng thái pháp sư, nếu đã ở trong trạng thái pháp sư thì sẽ về trạng thái bị động
* Phím 0: Trở về trạng thái bị động
* Phím F: Hồi máu (thời gian hồi chiêu cho mỗi lần hồi máu là 5 giây)

### Hệ thống AI

#### Tổng quan về trạng thái AI

Tương tự với nhân vật người chơi, trạng thái của kẻ thù cũng được lưu trữ dưới dạng enum.

Bảng .. Enum AIState, dùng để lưu trạng thái của kẻ thù

|  |
| --- |
| Tên các trạng thái của kẻ thù |
| Passive |
| Investigating |
| Attacking |
| Seeking |
| Frozen |
| Dead |

Kẻ thù hiện tại có tất cả 6 trạng thái, chúng lần lượt là trạng thái bị động (Passive; cũng có thể xem là Patrol), trạng thái tấn công (Attacking), trạng thái bất động (Frozen), trạng thái tìm hiểu (Investigating), trạng thái truy lùng (Seeking) và trạng thái đã chết (Dead). Các trạng thái của kẻ thù hoạt động như sau:

* Trong trạng thái bị động (hoặc trạng thái tuần tra), tức chưa phát hiện ra người chơi, kẻ thù sẽ được điều khiển để đi tuần tra theo một lộ trình nhất định, nếu không được chỉ định lộ trình, kẻ thù sẽ đứng yên.
* Khi kẻ thù nghe thấy tiếng động, kẻ thù sẽ vào trạng thái tìm hiểu, trong trạng thái này kẻ thù sẽ di chuyển đến vị trí mà âm thanh phát ra.
* Khi phát hiện ra người chơi, kẻ thù đổi sang trạng thái tấn công, liên tục đuổi theo người chơi để thực hiện các đòn tấn công của chúng.
* Khi mất tầm nhìn của người chơi, kẻ thù sẽ vào trạng thái truy lùng, trong trạng thái này kẻ thù tiến hành tìm ở những nơi mà người chơi có thể trốn như các góc tường, và sau một khoảng thời gian không tìm thấy người chơi, kẻ thù sẽ quay lại trạng thái tuần tra.
* Khi kẻ thù bị dính một đòn tấn công có gây hiệu ứng choáng, kẻ thù sẽ vào trạng thái bất động. Trong trạng thái này, kẻ thù sẽ đứng im trong một khoảng thời gian, đồng thời cũng không thể ra đòn tấn công.
* Khi máu của kẻ thù về không thì kẻ thù vào trạng thái chết. Trong trạng thái này, kẻ thù sẽ ngã ra đất, không thể bị chọn làm mục tiêu, không thể gây thêm và cũng không thể nhận thêm sát thương.

#### Cách hoạt động của mỗi loại kẻ thù

Mỗi kẻ thù đều được kế thừa từ một lớp khuôn mẫu kẻ thù, mỗi trong số chúng có một enum để lưu các đòn tấn công mà loại kẻ thù đó có (nếu như có từ 2 đòn tấn công trở lên) và một cây hành vi (Behavior Tree). Mỗi một loại kẻ thù nên có một cây hành vi riêng để có thể đa dạng hơn trong hành động và làm tăng tính độc lập giữa các loại kẻ thù. Ta lấy ví dụ là kẻ thù cận chiến như sau:

Bảng .. Enum MeleeEnemyAttack, lưu các đòn tấn công của kẻ thù cận chiến

|  |
| --- |
| Tên các loại tấn công của kẻ thù cận chiến |
| Default |
| ShortRangeAttack |
| LongRangeAttack |
| SpinningAttack |
| GroundSmashAttack |

Tùy vào trạng thái của kẻ thù trong enum trên mà ta có cây hành vi đưa ra hành động tương ứng:

A graph with lines and dots

Description automatically generated

Hình .. Behavior Tree của kẻ thù cận chiến

#### EQS

Khác với chuyển động thông thường, khi kẻ thù đi theo các điểm định sẵn hay đuổi theo người chơi, thì còn có một cách nữa để tạo nên một danh sách các điểm mà kẻ thù có thể đi lại, và cách đó chính là sử dụng EQS. Hiểu một cách đơn giản, tùy vào điều kiện bạn cho như hình dạng, góc nhìn, ở đâu được đi và ở đâu không được đi, khoảng cách giữa các điểm,… bạn có thể tạo ra một danh sách các điểm mà đối tượng có thể di chuyển đến.

A screenshot of a computer generated image

Description automatically generated

Hình .. Một danh sách các điểm của một có được bằng cách sử dụng EQS

Trong game này, EQS đóng một vai trò rất lớn. Nó được sử dụng khi kẻ thù đang ở trong trạng thái tấn công (vây quanh người chơi chờ cơ hội phù hợp và tìm kiếm chỗ nấp), trọng thái truy lùng (tìm các điểm mà không trông tầm nhìn của kẻ thù), chọn điểm dịch chuyển cho các kẻ thù có khả năng dịch chuyển (kẻ thù pháp sư và trùm cuối).

## Kết quả thực hiện

### Một hệ thống game thú vị có thể chơi lại nhiều lần

Chung quy lại, tuy bản đồ không thay đổi qua mỗi lần chơi, giá trị chơi lại của game vẫn là rất cao, khi sự phối hợp và chiến lược giữa những kẻ thù với nhau ở mỗi lần chơi là vô cùng khác nhau. Chúng tạo lên những lần trải nghiệm khác biệt ở mỗi lần chơi, điều này giúp cải thiện một yếu tố có thể coi là điểm yếu của game là bản đồ game là như nhau ở mọi lần chơi, không hề thay đổi.

### Bầu không khí cũng như bản đồ tràn đầy sự u ám, ảm đạm

Về mặt thiết kế, em đã thiết kế được một bản đồ với bầu không khí ảm đạm, toát lên vẻ hắc ám của phe kẻ thù. Mê cung khổng lồ làm cho người chơi cảm thấy nhỏ bé, rất nhiều lối đi cùng với những kẻ thù phục kích và những trại địch vô cùng đông đảo kẻ thù. Những yếu tố trên sẽ giúp cho việc khám phá mê cung hầm ngục trở nên vô cùng chân thật, giúp người chơi nhập vai dễ dàng hơn vào nhân vật của mình.

A video game with a person standing on a rock

Description automatically generated

Hình .. Bản đồ mê cung hầm ngục với rất nhiều kẻ thù (thanh máu màu đỏ)

### Đa dạng kẻ thù, đa dạng cách chiến đấu

Với việc có tới 3 loại kẻ thù thông thường và chúng hoàn toàn có thể phối hợp với nhau, việc chiến đấu với chúng trở nên khó khăn nhưng cũng thú vị hơn. Kẻ thù cận chiến giữ chân bạn đánh tay đôi với chúng, kẻ thù tầm xa liên tục hạ thấp máu của bạn, và kẻ thù pháp sư có thể sử dụng những chiêu thức vô cùng ảo diệu, những kẻ thù này giúp trải nghiệm chơi game được tăng cao.



Hình .. Chiến đấu với một tốp kẻ thù cận chiến

A video game screen with purple flames and lights

Description automatically generated

Hình .. Chiến đấu với kẻ thù pháp sư

### Trận đấu huy hoàng với trùm cuối

A screenshot of a video game

Description automatically generated

Hình .. Trùm cuối chuẩn bị dùng chiêu thức của mình

Một game hầm ngục hay luôn kết thúc bằng một trận đấu hoành tráng với trùm cuối, và game này cũng không ngoại lệ. Trận đấu với trùm cuối này không chỉ thử thách kỹ năng chiến đấu của người chơi, mà còn yêu cầu họ vận dụng mọi chiến lược và kỹ năng đã tích lũy suốt quá trình khám phá. Mỗi đòn tấn công, mỗi bước di chuyển, và từng hành động đều đòi hỏi sự chính xác và phản xạ nhanh nhạy, bởi trùm cuối không chỉ mạnh hơn mà còn sở hữu những chiêu thức phức tạp hơn, phối hợp khéo léo giữa sức mạnh và tốc độ. Người chơi sẽ phải quan sát, phân tích từng động thái của kẻ địch, tận dụng các khoảng trống hiếm hoi để phản công, và chỉ khi thật sự hiểu rõ những quy luật ngầm của trận chiến, họ mới có thể giành được chiến thắng cuối cùng. Hơn thế nữa, với sàn đấu thể hiện rõ đây là một trận đấu tay đôi lẫn nhạc nền hoành tráng, có thể nói rằng trận đấu này chính là đỉnh cao của game.

A screenshot of a video game

Description automatically generated

Hình .. Tấn công trùm cuối bằng combo cận chiến

Tuy nhiên, trận đánh này còn có ý nghĩa sâu sắc hơn nữa trên mặt học vấn. Trận đánh với trùm cuối này là kết tinh của tất cả những kiến thức em đã học được từ lúc bắt đầu xây dựng game cho đến giờ. Những gì học được em đều đã ứng dụng trong trận chiến cuối cùng này, với mong ước biến nó thành một điểm xuất phát vững chãi trên con đường em sẽ đi trong tương lai.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Sau 10 tuần phát triển và hoàn thiện, game hầm ngục 3D này đã đáp ứng được hầu hết các mục tiêu ban đầu đã đề ra. Trò chơi mang đến một trải nghiệm hấp dẫn với các yếu tố khám phá mê cung, hệ thống chiến đấu đa dạng bao gồm di chuyển, tấn công, hồi máu, né tránh, đỡ đòn và phản đòn. Hệ thống AI điều khiển kẻ thù đã được xây dựng với sự đa dạng trong trạng thái và hành vi, thêm vào đó nữa là có nhiều loại kẻ thù, cùng với một trận chiến cuối cùng hoành tráng, góp phần tạo nên các thử thách thú vị, giữ cho trải nghiệm không trở nên nhàm chán. Bản đồ là mê cung rộng lớn cùng bầu không khí u ám và nhiều kẻ thù cũng đã giúp game đạt được mục tiêu về sự thử thách. Đồ họa đẹp mắt, hiệu ứng chân thực, giao diện được thiết kế vô cùng gọn gàng và điều khiển đơn giản, dễ làm quen giúp tăng cường trải nghiệm người chơi. Qua giai đoạn chơi thử với đối tượng là những người đam mê thể loại game này và người trẻ tuổi, game cũng đã nhận được nhiều ý kiến tích cực, cho thấy rằng game cũng đã đạt được mục tiêu đề ra về đối tượng mà game nhắm đến. Từ những điều trên, em có thể kết luận rằng game đã đạt gần như đủ các mục tiêu đã đề ra.

Việc có cơ hội được học và sử dụng Unreal Engine, một trong những công cụ phát triển game mạnh mẽ nhất thế giới, đã mang lại cho em rất nhiều kinh nghiệm quý báu. Không chỉ dừng lại ở việc giúp em hiểu sâu hơn về quy trình thiết kế và phát triển một game, em còn được học và sử dụng những công cụ tiên tiến mà Unreal Engine cung cấp để tối ưu hóa đồ họa, gameplay và trí tuệ nhân tạo, từ đó nâng cao chất lượng sản phẩm cuối cùng. Trong số đó, em đặc biệt ấn tượng với hệ thống AI, khi em được cung cấp vô vàn các thành phần vô cùng hữu ích như Behavior Tree, AI Senses, EQS, Navigation System, đã cho em thấy một bầu trời bao la vô vàn các khả năng mà AI có thể được thiết lập trong Unreal Engine. Qua trải nghiệm này, em thấy rằng bản thân không chỉ cải thiện tư duy khi phát triển game mà còn hiểu rõ hơn về quy trình tạo ra một trò chơi hoàn chỉnh, đáp ứng những yêu cầu khắt khe về thẩm mỹ, trải nghiệm người dùng, và tính thực tiễn trong thiết kế game hiện đại.

Về bản thân, em đã dành phần lớn thời gian tự học và tự nghiên cứu trong suốt quá trình thực hiện đồ án. Em học thông qua những người làm game khác, cũng như học thông qua các tài liệu đã có sẵn. Sau khi học thì em liên tục thử nghiệm, ứng dụng những gì đã học được, kết hợp cái này với cái kia, nhờ thế mà em đã rút ra được những gì nên và không nên làm khi sử dụng Unreal Engine nói riêng và phát triển game nói chung. Tuy nhiên thì vẫn có những điều em chưa làm được, và chủ yếu là ở đồ họa và chi tiết trong game. Để tự tạo các hiệu ứng, vật phẩm hay chỉ đơn giản là chỉnh sửa các vật phẩn là vô cùng khó, việc xây dựng lên một bản đồ trông tự nhiên và đẹp cũng không hề đơn giản. Do hạn chế về thời gian, em đã không thể nghiên cứu sâu thêm về cách để làm tốt hơn ở mặt đồ họa. Tất nhiên những công cụ có sẵn như ánh sáng và kết xuất hình ảnh vẫn đã giúp đồ họa của game trông rất ổn, nhưng những chi tiết hiện tại vẫn có thể cải thiện thêm.

Hệ thống của game mang đến trải nghiệm chiến đấu với kẻ thù rất tốt, bao gồm đầy đủ các tính năng cơ bản như tấn công, phòng thủ, né đòn, và hồi máu. Việc tích hợp các yếu tố này giúp người chơi cảm thấy hài lòng với sự đa dạng trong cách tiếp cận chiến đấu. Ngoài ra thì kẻ thù được thiết kế thông minh, với AI có nhiều trạng thái khác nhau, tạo nên thử thách hấp dẫn và không kém phần khó khăn, giúp người chơi cảm nhận rõ nét về sự tiến bộ và khả năng đối đầu trong trò chơi. Điểm trừ lớn nhất của hệ thống nằm ở việc thiếu một hệ thống lưu trữ tiến trình game. Điều này có nghĩa là người chơi phải hoàn thành game trong một lần chơi duy nhất, và nếu thoát ra, mọi tiến độ sẽ bị mất. Trong bối cảnh các game hiện đại ngày nay, việc không có tính năng lưu trữ có thể làm giảm trải nghiệm người chơi, đặc biệt là với những game có độ khó cao. Điều này dễ dẫn đến sự chán nản và mất động lực khi người chơi không thể tạm dừng và tiếp tục lại ở thời điểm họ mong muốn.

## Hướng phát triển

Để nâng cao trải nghiệm người chơi và theo kịp xu thế của các trò chơi hiện đại, một hệ thống lưu trữ tiến trình cần được phát triển, cho phép người chơi lưu và tiếp tục game bất kỳ lúc nào. Ngoài ra, việc mở rộng thêm các tính năng chiến đấu như kết hợp đòn tấn công, sử dụng nhiều loại phép thuật hoặc kỹ năng đặc biệt sẽ tạo chiều sâu cho hệ thống chiến đấu, làm cho game trở nên đa dạng và cuốn hút hơn. Một hệ thống phần thưởng hoặc vật phẩm cũng có thể được thêm vào, cho phép người chơi thu thập và nâng cấp trang bị qua các trận đấu. Cuối cùng, để tăng thêm tính thách thức và duy trì sự hấp dẫn lâu dài, việc phát triển thêm các cấp độ khó khác nhau, các màn chơi mới, chế độ chơi mới, và các kẻ thù hoặc trùm độc đáo sẽ là những bước phát triển tiềm năng. Hướng đi này sẽ giúp trò chơi trở nên hấp dẫn hơn và phù hợp với nhiều đối tượng người chơi.

TÀI LIỆU THAM KHẢO

[1] Mike Wolfe, "Ren’Py: A Simple Solution To Building Visual Novel Games", 17/04/2021. [Online]. Available: <https://python.plainenglish.io/renpy-a-simple-solution-to-building-visual-novel-games-32d6179a7840>. [Accessed 18/08/2024]

[2] Vinod Ravisankar, "RPG maker game engines review", 02/05/2023. [Online]. Available: <https://vionixstudio.com/2020/01/11/rpg-maker-game-engines-review/>. [Accessed 18/08/2024]

[3] Jordan Minor, "GameMaker Review", 11/12/2023. [Online]. Available: <https://www.pcmag.com/reviews/gamemaker>. [Accessed 18/08/2024]

[4] Amit Yadav, "Godot vs Unity vs Unreal", 06/07/2024. [Online]. Available: <https://medium.com/@amit25173/godot-vs-unity-vs-unreal-775fa7d91b2c>. [Accessed 18/08/2024]

[5] Guinness World Records, "Most successful game engine", 16/07/2014. [Online]. Available: <https://www.guinnessworldrecords.com/world-records/most-successful-game-engine>. [Accessed 21/08/2024]

[6] Unreal Engine, "Unreal Engine 5.4 is now available", 23/04/2024. [Online]. Available: <https://www.unrealengine.com/en-US/blog/unreal-engine-5-4-is-now-available>. [Accessed 21/08/2024]

PHỤ LỤC

A computer screen shot of a computer

Description automatically generated

Chi tiết luồng hoạt động của Gameplay Framework khi chạy một project (nguồn: https://dev.epicgames.com)