**LỜI CẢM ƠN**

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, Trường Đại Duy Tân đã tạo điều kiện thuận lợi để em có cơ hội thực hiện và hoàn thành đề tài này.

Đặc biệt, em xin gửi lời tri ân sâu sắc đến Thầy Trịnh Hiệp Hòa, người đã tận tình hướng dẫn, định hướng và hỗ trợ em trong suốt quá trình nghiên cứu. Những kiến thức, góp ý và sự động viên của thầy là nguồn động lực to lớn giúp em vượt qua những khó khăn, hoàn thiện đề tài một cách tốt nhất.

Bên cạnh đó, em cũng xin gửi lời cảm ơn chân thành đến quý thầy cô trong khoa, những người đã không ngừng giảng dạy và truyền đạt cho em những kiến thức quý báu trong suốt thời gian học tập.

Em cũng không thể không nhắc đến gia đình – Cha, Mẹ, anh chị – những người luôn ở bên, động viên và tiếp thêm sức mạnh cho em trên hành trình này. Đồng thời, em xin gửi lời cảm ơn đến các bạn bè đã luôn đồng hành, giúp đỡ và chia sẻ cùng em trong suốt chặng đường học tập và nghiên cứu.

Dù đã nỗ lực hoàn thiện luận văn trong khả năng của mình, nhưng chắc chắn vẫn còn những thiếu sót. Em rất mong nhận được sự góp ý từ quý thầy cô và các bạn để có thể hoàn thiện hơn nữa.

Em xin chân thành cảm ơn!

Đà Nẵng, ngày 22 tháng 3 năm 2025.

Sinh viện thực hiện

Đặng Bảo Huy

LỜI CAM ĐOAN

Tôi xin cam đoan rằng:

a. Toàn bộ nội dung trong luận văn này được tôi thực hiện dưới sự hướng dẫn trực tiếp của Thầy Trịnh Hiệp Hòa.

b. Mọi tài liệu tham khảo được sử dụng trong luận văn đều được trích dẫn đầy đủ, chính xác, nêu rõ tên tác giả, tên công trình, thời gian và địa điểm công bố.

c. Tôi hoàn toàn chịu trách nhiệm trước nhà trường về mọi hành vi sao chép không hợp lệ, vi phạm quy chế đào tạo hoặc gian lận trong quá trình thực hiện luận văn này.

Tôi xin chịu hoàn toàn trách nhiệm trước những cam kết trên.

Đà Nẵng, ngày 22 tháng 3 năm 2025.

Sinh viện thực hiện

Đặng Bảo Huy

MỤC LỤC

[MỞ ĐẦU 1](#_Toc194068490)

[CHƯƠNG I 4](#_Toc194068491)

[TỔNG QUAN VỀ ĐỀ TÀI VÀ MỖI TRƯỜNG PHÁT TRIỂN 4](#_Toc194068492)

[**1.1** **GIỚI THIỆU ĐỀ TÀI GAME KINH DỊ** 4](#_Toc194068493)

[1.1.1 Game kinh dị (Horror Game) là gì? 4](#_Toc194068494)

[1.1.2 Đặc điểm của game kinh dị. 4](#_Toc194068495)

[1.1.3 Phân loại game kinh dị 5](#_Toc194068496)

[1.1.4 Một số tựa game kinh dị nổi bật 5](#_Toc194068497)

[**1.3 GIỚI THIỆU VỀ UNREAL ENGINE** 5](#_Toc194068498)

[1.3.1 Khái niệm Unreal Engine 5](#_Toc194068499)

[1.3.2 Lịch sử hình thành Unreal Engine 6](#_Toc194068500)

[1.3.3 Các phiên bản Unreal Engine 7](#_Toc194068501)

[1.3.4 Các tính năng và công dụng phổ biến của UE 11](#_Toc194068502)

[1.3.5 Những ưu thế của Unreal Engine 12](#_Toc194068503)

[**1.4 GIỚI THIỆU UNREAL ENGINE 5.4** 13](#_Toc194068504)

[1.4.1 Tổng quan về Unreal Engine 5.4 13](#_Toc194068505)

[1.4.2 Những cải tiến quan trọng trong Unreal Engine 5.4 14](#_Toc194068506)

[**1.5 GIỚI THIỆU BLUEPRINTS VISUAL SCRIPTING** 19](#_Toc194068507)

[1.5.1 Blueprints là gì? 19](#_Toc194068508)

[1.5.2 Ưu điểm của Blueprints so với lập trình truyền thống 19](#_Toc194068509)

[1.5.3 Cấu trúc và thành phần của Blueprints 20](#_Toc194068510)

[1.5.4 Nguyên lý hoạt động của Blueprints 21](#_Toc194068511)

[**1.6 GIỚI THIỆU AI SYSTEM** 23](#_Toc194068512)

[1.6.1 Giới thiệu tổng quan về AI System trong Unreal Engine 23](#_Toc194068513)

[1.6.2 Behavior Tree (Cây hành vi) trong AI 23](#_Toc194068514)

[CHƯƠNG II 26](#_Toc194068515)

[PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 26](#_Toc194068516)

[**2.1 KHẢO SÁT THỰC TRẠNG** 26](#_Toc194068517)

[**2.2 MỘT SỐ TỰA GAME KINH DỊ TƯƠNG TỰ** 27](#_Toc194068518)

[**2.3 GIỚI THIỆU GAME NYCTOPHOBIA** 29](#_Toc194068519)

[2.3.1 Khái quát về game 29](#_Toc194068520)

[2.3.2 Cốt truyện 29](#_Toc194068521)

[2.3.3 Gameplay 29](#_Toc194068522)

[2.3.4 Thiết kế màn chơi 30](#_Toc194068523)

[2.3.5 Hệ thống UI/UX 31](#_Toc194068524)

[2.3.6 Hiệu ứng môi trường 31](#_Toc194068525)

# MỞ ĐẦU

**1. Lý do chọn đề tài**

Trong những năm gần đây, thể loại game kinh dị sinh tồn ngày càng phát triển mạnh mẽ và thu hút đông đảo người chơi nhờ vào trải nghiệm căng thẳng, hồi hộp và đầy thử thách. Không chỉ đơn thuần dựa vào hiệu ứng jumpscare, các tựa game hiện đại còn tập trung vào việc xây dựng bầu không khí u ám, lối chơi sáng tạo.

Đặc biệt, với sự phát triển mạnh mẽ của công nghệ đồ họa và trí tuệ nhân tạo (AI), việc xây dựng một game kinh dị không chỉ dừng lại ở yếu tố hù dọa đơn thuần, mà còn đòi hỏi sự đầu tư vào lối chơi, bối cảnh và cơ chế hành vi của quái vật để mang đến cảm giác chân thực nhất.

Sau khi tìm hiểu nhiều tựa game kinh dị nổi tiếng, em nhận ra rằng nỗi sợ trong game không chỉ đến từ những hình ảnh đáng sợ mà còn từ cách người chơi phải đối mặt với những thử thách không lường trước. Từ đó, em đã nảy ra ý tưởng và quyết định phát triển Nyctophobia – một tựa game kinh dị góc nhìn thứ nhất, nơi người chơi phải tìm kiếm mật mã để trốn thoát khỏi một không gian tăm tối, trong khi bị săn đuổi bởi một thực thể bí ẩn.

Với đề tài “Phát triển game kinh dị 3D Nyctophobia với AI quái vật trên Unreal Engine 5”, em mong muốn không chỉ tạo ra một trải nghiệm game hấp dẫn mà còn khám phá sâu hơn về cách xây dựng bầu không khí căng thẳng trong game kinh dị, đồng thời phát triển lối chơi độc đáo, khiến người chơi luôn phải tỉnh táo và cẩn trọng trong từng bước đi. Đây không chỉ là một bài tập về lập trình, mà còn là một cơ hội để em hiểu rõ hơn về cách thiết kế trải nghiệm người chơi trong một tựa game kinh dị chân thực.

**2. Mục đích và ý nghĩa của đề tài**

**a**. Mục đích

* Xây dựng một tựa game kinh dị 3D góc nhìn thứ nhất với trải nghiệm căng thẳng, hồi hộp và đầy thử thách.
* Nghiên cứu và ứng dụng Blueprints trong Unreal Engine 5 để xây dựng một tựa game kinh dị, sử dụng AI quái vật làm trung tâm trải nghiệm.
* Nghiên cứu và triển khai Unreal Engine 5 trong việc thiết kế môi trường game chân thực.

b. Ý nghĩa

* Về mặt lý thuyết: Đề tài giúp em nghiên cứu về AI trong game, đặc biệt là cách vận hành và tối ưu hành vi của quái vật trong môi trường 3D.
* Về mặt thực tiễn: Sản phẩm có thể phát triển thành một tựa game hoàn chỉnh, đáp ứng xu hướng phát triển của ngành công nghiệp game và cung cấp một trải nghiệm mới lạ cho người chơi.

**3. Đối tượng và phạm vi nghiên cứu**

1. Đối tượng

Những người yêu thích thể loại game kinh dị, cảm giác hồi hộp,…

1. Phạm vi

* Nghiên cứu và triển khai AI quái vật bằng Behavior Tree trên Unreal Engine 5.
* Thiết kế môi trường game mang đậm chất kinh dị với hệ thống ánh sáng, âm thanh và hiệu ứng hỗ trợ tạo cảm giác căng thẳng.
* Phát triển cơ chế tương tác và thu thập manh mối để người chơi có thể mở khóa cánh cửa thoát hiểm.
* Kiểm thử AI quái vật và tối ưu hóa trải nghiệm người chơi.

**4. Phương pháp nghiên cứu**

Để thực hiện đề tài em sử dụng các phương pháp nghiên cứu sau:

* Nghiên cứu Unreal Engine 5 và Blueprint:

- Tìm hiểu cấu trúc và cơ chế hoạt động của Unreal Engine 5.

- Nghiên cứu Blueprint Visual Scripting để xây dựng gameplay.

* Nghiên cứu AI và Behavior Tree:

- Xây dựng AI quái vật Khuất Quỷ

- Ứng dụng Behavior Tree để điều khiển hành vi quái vật một cách thông minh.

- Tích hợp hệ thống Navigation Mesh để AI di chuyển trong môi trường.

* Phân tích các yếu tố thiết kế game kinh dị:

- Tìm hiểu các yếu tố gây sợ hãi trong game như ánh sáng, âm thanh, môi trường u ám.

- Nghiên cứu cách xây dựng bầu không khí căng thẳng, jumpscare và tâm lý người chơi.

* Nghiên cứu và sử dụng Asset hợp lý:

- Tìm kiếm và sử dụng Asset phù hợp (nhân vật, môi trường, hiệu ứng âm thanh).

- Tinh chỉnh, tối ưu hóa mô hình 3D, hiệu ứng ánh sáng, môi trường để đảm bảo hiệu suất.

**5. Kết quả dự kiến**

Xây dựng thành công game Nyctophobia với gameplay hoàn chỉnh trong phạm vi đề tài trong thời gian 2 tháng.

**6. Bố cục đề tài**

Nội dung của đề tài gồm các phần:

Phần 1: Giới thiệu tổng quan về đề tài và các công nghệ liên quan.

Phần 2: Phân tích và thiết kế hệ thống game Nyctophobia.

Phần 3: Cài đặt, triển khai và kiểm thử game.

Mặc dù đã rất cố gắng thực hiện đề tài nhưng vì năng lực cũng như thời gian còn hạn chế nên chương trình khó tránh khỏi những thiếu sót, rất mong thầy cô thông cảm. Những góp ý của thầy cô là bài học, là hành trang để em vững bước vào cuộc sống sau này. Qua đây, em xin gửi lời cảm ơn thầy Trịnh Hiệp Hòa, người đã nhiệt tình hướng dẫn, chỉ bảo em trong quá trình thực hiện, hoàn thành đề tài.

Chân thành cám ơn sự đóng góp ý kiến của các thầy, cô và bạn đồng nghiệp để đề tài của tôi có thể hoàn thành tốt hơn.

*Đà Nẵng, ngày 22 tháng 3, năm 2025.*

# CHƯƠNG I

# TỔNG QUAN VỀ ĐỀ TÀI VÀ MỖI TRƯỜNG PHÁT TRIỂN

## GIỚI THIỆU ĐỀ TÀI GAME KINH DỊ

### 1.1.1 Game kinh dị (Horror Game) là gì?

Game kinh dị (horror game) là một thể loại trò chơi điện tử được thiết kế để tạo ra cảm giác sợ hãi, căng thẳng và hồi hộp cho người chơi. Đây là một trong những thể loại game có ảnh hưởng mạnh đến cảm xúc, thường kết hợp giữa cốt truyện u ám, môi trường đáng sợ, âm thanh rùng rợn và các yếu tố siêu nhiên hoặc tâm lý.

Game kinh dị thường khai thác những nỗi sợ phổ biến của con người, như nỗi sợ bóng tối(Nyctophobia), nỗi sợ bị truy đuổi, nỗi sợ sinh vật siêu nhiên hoặc quái vật, nỗi sợ những điều không thể lý giải.

### Đặc điểm của game kinh dị.

Game kinh dị có nhiều phong cách khác nhau, nhưng hầu hết đều có những đặc điểm chung như :

* Không khí đáng sợ
* Sử dụng ánh sáng yếu hoặc mờ ảo.
* Âm thanh ghê rợn như tiếng bước chân, tiếng gió hú, tiếng thở dồn dập.
* Hiệu ứng môi trường như sương mù, bóng đổ, rung lắc.
* Lối chơi sinh tồn (Survival Horror)
* Người chơi thường bị hạn chế vũ khí hoặc không có vũ khí, buộc phải chạy trốn hoặc ẩn nấp.
* Tài nguyên khan hiếm, như đạn, máu hoặc ánh sáng.
* Quái vật hoặc kẻ thù có thể xuất hiện bất ngờ và đuổi theo người chơi.
* Cốt truyện bí ẩn và ám ảnh
* Thường liên quan đến những bí ẩn, lời nguyền, tai nạn, thí nghiệm thất bại hoặc sự kiện siêu nhiên.
* Người chơi dần khám phá câu chuyện thông qua ghi chép, băng cassette hoặc lời thoại nhân vật,…

### 1.1.3 Phân loại game kinh dị

Game kinh dị có thể được chia thành nhiều nhánh nhỏ tùy theo cách thiết kế và trải nghiệm mà nó mang lại:

Survival Horror (Kinh dị sinh tồn): Người chơi thường có ít tài nguyên, buộc phải tìm cách sống sót thay vì đối đầu trực diện với kẻ thù.

Psychological Horror (Kinh dị tâm lý): Tập trung vào yếu tố tinh thần, khiến người chơi cảm thấy căng thẳng thông qua không khí và câu chuyện.

Action Horror (Kinh dị hành động): Kết hợp yếu tố kinh dị với chiến đấu, cho phép người chơi sử dụng vũ khí để đối đầu với quái vật.

Stealth Horror (Kinh dị tàng hình): Người chơi cần tránh né kẻ thù thay vì đối đầu trực tiếp.

### Một số tựa game kinh dị nổi bật

* Resident Evil series – Kết hợp giữa hành động và kinh dị sinh tồn, nổi tiếng với bầu không khí căng thẳng và tài nguyên hạn chế.
* Silent Hill series – Đặt nặng yếu tố tâm lý, với môi trường ám ảnh và những câu chuyện phức tạp.
* Outlast – Người chơi không có vũ khí, buộc phải chạy trốn và ẩn nấp khỏi kẻ thù.
* Amnesia: The Dark Descent – Tập trung vào yếu tố tâm lý, yêu cầu người chơi giải đố để sinh tồn.

## 1.3 GIỚI THIỆU VỀ UNREAL ENGINE

### 1.3.1 Khái niệm Unreal Engine

Unreal Engine (UE) là một game engine – phần mềm dùng để xây dựng và phát triển trò chơi điện tử, được tạo ra bởi Tim Sweeney và phát triển bởi Epic Games. Kể từ khi ra mắt vào năm 1998, Unreal Engine đã trở thành một trong những nền tảng phát triển game mạnh mẽ và phổ biến nhất thế giới, được sử dụng rộng rãi trong nhiều thể loại game từ FPS (First-Person Shooter), RPG (Role-Playing Game), MOBA (Multiplayer Online Battle Arena) đến game kinh dị sinh tồn.

Unreal Engine cung cấp một hệ sinh thái hoàn chỉnh để phát triển game, bao gồm đồ họa 3D chất lượng cao, hệ thống vật lý tiên tiến, công cụ trí tuệ nhân tạo (AI), và hệ thống âm thanh chân thực. Nhờ vào khả năng tùy chỉnh linh hoạt, Unreal Engine không chỉ được sử dụng trong lĩnh vực phát triển game mà còn được ứng dụng trong công nghệ thực tế ảo (VR/AR), mô phỏng kiến trúc, kỹ xảo điện ảnh, và nhiều ngành công nghiệp khác.

Phiên bản mới nhất mà game *Nyctophobia* sử dụng là Unreal Engine 5.4, mang lại nhiều cải tiến đáng kể về đồ họa, hiệu suất, và công cụ phát triển.

### 1.3.2 Lịch sử hình thành Unreal Engine

Unreal được phát hành lần đầu dưới dạng trò chơi [bắn súng góc nhìn thứ nhất](https://www.thegioididong.com/game-app/game-fps-la-gi-lich-su-phat-trien-va-cac-diem-dac-trung-1356984) bởi [Epic](https://vi.wikipedia.org/wiki/Epic_Games) vào năm 1998, trong đó Tim Sweeney - nhà sáng lập đã đóng góp đến 90% trong quá trình sáng tạo. Game Unreal lúc này được phát triển bởi hệ thống công cụ Unreal Engine đã thu hút nhiều người quan tâm. Lúc này, game mang đến cho người chơi tính tùy biến bản đồ cực cao, khả năng tạo ra môi trường giàu vật thể với ánh sáng và màu sắc phong phú... Nó đã tạo nên một cơn sốt rất lớn trong thế giới game lúc bấy giờ.

Unreal Engine liên tục được cải tiến, trải qua các phiên bản lại được thêm thắt các tính năng ấn tượng và trở thành bộ công cụ không thể thiếu để sản xuất game bom tấn. Hơn nữa, Unreal Engine còn được ứng dụng trong các công đoạn sản xuất hiệu ứng cho các phim bom tấn [Hollywood](https://vi.wikipedia.org/wiki/Hollywood) như [Jurassic Park](https://en.wikipedia.org/wiki/Jurassic_Park_(film)), [Independence Day](https://vi.wikipedia.org/wiki/Ng%C3%A0y_%C4%91%E1%BB%99c_l%E1%BA%ADp:_T%C3%A1i_chi%E1%BA%BFn),...

Khả năng của Unreal Engine chính là tạo ra các [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) mới, [kết xuất đồ họa](https://vi.wikipedia.org/wiki/K%E1%BA%BFt_xu%E1%BA%A5t_%C4%91%E1%BB%93_h%E1%BB%8Da) (render) để dựng nên hình ảnh 3D của vật thể, cháy nổ trông như thật. Hiện nay Unreal Engine đã cho ra mắt phiên bản [Unreal Engine 5](https://www.unrealengine.com/en-US/unreal-engine-5) khiến cả thế giới game mãn nhãn và kinh ngạc tột độ. Những hiệu ứng chi tiết trên bề mặt vật thể và hiệu ứng ánh sáng được engine này phô diễn cho thấy ranh giới giữa game và đời thật mong manh hơn bao giờ hết. Đến năm 2014, Unreal Engine đã được [Sách Kỷ lục Thế giới (Guinness World Records)](https://vi.wikipedia.org/wiki/S%C3%A1ch_K%E1%BB%B7_l%E1%BB%A5c_Guinness) ghi nhận là game engine thành công nhất thể giới khi có đến hơn 408 tựa game trên thế giới sử dụng công nghệ này.

Kiến trúc hoạt động của UE:

Unreal Engine được xây dựng trên kiến trúc module-based, nghĩa là nó bao gồm nhiều thành phần hoạt động độc lập nhưng có thể liên kết với nhau để tạo ra trò chơi. Những thành phần quan trọng gồm:

* Core System (Hệ thống lõi): Quản lý bộ nhớ, xử lý đầu vào, vòng lặp game, đa luồng.
* Rendering Engine (Bộ xử lý đồ họa): Chịu trách nhiệm xử lý hình ảnh, ánh sáng, vật liệu, đổ bóng.
* Physics Engine (Bộ mô phỏng vật lý): Xử lý va chạm, mô phỏng lực tác động, trọng lực.
* AI System (Hệ thống trí tuệ nhân tạo): Điều khiển hành vi AI, di chuyển nhân vật NPC.
* Animation System (Hệ thống hoạt ảnh): Quản lý chuyển động của nhân vật, Blend Spaces, State Machine.
* Scripting (Lập Trình Kịch Bản): Hỗ trợ Blueprints và C++ để tạo gameplay.
* Networking (Hệ thống mạng): Đồng bộ dữ liệu giữa người chơi trong game multiplayer.
* Audio System (Hệ thống âm thanh): Mô phỏng âm thanh 3D, hiệu ứng âm thanh theo môi trường.

### 1.3.3 Các phiên bản Unreal Engine

a. Unreal Engine 1 (1998 - 2002)

Phiên bản đầu tiên của UE ra mắt đến người chơi vào năm 1998. Khi này tựa game được phát hành được phát triển trên UE là Unreal. UE ban đầu là công cụ hoàn toàn dựa vào kết xuất phần mềm, có nghĩa là các tính toán đồ họa được xử lý bởi CPU. Dần theo thời gian, nó đã có thể tận dụng các khả năng được cung cấp bởi card đồ họa trong quá trình xử lý.

Các tựa game tiêu biểu: Unreal Tournament (1999), Deus Ex (2000), Rune (2000).



Hình 1. 1 Unreal Engine trong phiên bản game Unreal (1998)

b. Unreal Engine 2 (2002 - 2004)

Unreal Engine 2 là phiên bản kế nhiệm được tung ra vào năm 2002. Khi này Epic kết hợp với quân đội [Hoa Kỳ](https://vi.wikipedia.org/wiki/Hoa_K%E1%BB%B3) cho ra một tựa game bắn súng góc nhìn thứ nhất để phục vụ cho quá trình tuyển dụng quân nhân. Thế hệ này có những cải tiến đáng kể trong hiển thị cũng như những cải tiến mới đối với bộ công cụ, có khả năng chạy các cấp độ chi tiết hơn gần 100 lần so với thế hệ trước. Công cụ tích hợp nhiều tính năng, bao gồm công cụ chỉnh sửa điện ảnh, hệ thống hạt, trình cắm xuất cho 3D.

Các tựa game tiêu biểu: Splinter Cell (2002), Unreal Tournament 2003, America's Army (2002).



Hình 1. 2 Unreal Egine 2

**c**. Unreal Engine 3 (2004 - 2012)

Unreal Engine 3 được giới thiệu vào tháng 7 năm 2004. Nó có tính năng mới như hiển thị cho các [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn) về thiết kế hướng đối tượng. [Trình kết xuất, hệ thống vật lý](https://vi.wikipedia.org/wiki/K%E1%BA%BFt_xu%E1%BA%A5t_%C4%91%E1%BB%93_h%E1%BB%8Da), hệ thống âm thanh và các công cụ –  tất cả đều rõ ràng và mạnh mẽ hơn đáng kể so với phiên bản trước.

Các tựa game tiêu biểu: Gears of War (2006), Mass Effect (2007), Batman: Arkham Asylum (2009), Bioshock (2007).



Hình 1. 3 Unreal Engine 3 được sử dụng trong Airborne

d. Unreal Engine 4 (2012 - 2020)

Unreal Engine 4 được công bố cho những người tham dự tại hội nghị các nhà phát triển trò chơi 2012. Phiên bản này bổ sung tính năng chiếu sáng toàn cầu theo thời gian thực và sử dụng theo dõi hình nón [voxel](https://en.wikipedia.org/wiki/Voxel).

Các tựa game tiêu biểu: Fortnite (2017), Street Fighter V (2016), PUBG (2017), Final Fantasy VII Remake (2020).



Hình 1. 4 Unreal Engine 4 ( PUBG )

e. Unreal Engine 5 (2020 – Hiện tại)

Unreal Engine 5 được ra mắt vào năm 2020. Một trong những tính năng chính của nó là Nanite, một công cụ cho phép nhập tài liệu nguồn ảnh có độ chi tiết cao vào trò chơi. Công cụ này giúp các nhà phát triển dễ dàng tạo thế giới trò chơi một cách chi tiết mà không cần phải dành nhiều thời gian cho việc tạo các chi tiết mới.

Các tựa game tiêu biểu: The Matrix Awakens (2021), Senua’s Saga: Hellblade II (sắp ra mắt), Stalker 2 (2024), Black Myth: Wukong (2024).



Hình 1. 5 Unreal Engine 5 ( Black Myth Wukong )

### 1.3.4 Các tính năng và công dụng phổ biến của UE

* Multi-platform:

Unreal Engine có thể tạo game trên iOS và Android, tối ưu hiệu suất cho điện thoại cấu hình cao. Tích hợp công cụ tối ưu hóa đồ họa để đảm bảo game chạy mượt mà trên các thiết bị di động. Hỗ trợ Windows, PlayStation, Xbox, và VR, giúp game có thể mở rộng sang nhiều nền tảng.

* Blueprints:

[Blueprints Visual Scripting](https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/) cho phép bạn nhanh chóng chạy thử nghiệm và xuất bản ra một tựa game hoàn chỉnh một cách nhanh chóng. Bạn cũng có thể thử sức tạo ra game của riêng mình mà không nhất thiết phải biết quá nhiều về lập trình.

* Tools:

Unreal là một bộ công cụ đầy đủ để giúp bạn xây dựng hoàn hảo mọi khía cạnh của Project. Ngoài ra còn có những tính năng nâng cao như: physically-based rendering, UI, animation, visual effects, networking, và asset management.

* Source Code:

Mọi nhà phát triển Game sử dụng Unreal Engine đều được sử dụng trình chỉnh sửa mã nguồn C++ (C++ Editor Source Code). Unreal Engine có thể tùy chỉnh và gỡ lỗi cho tựa game của mình.

* Marketplace:

Cung cấp cho bạn những bảng mã về game giúp bạn có thể tăng tốc tiến độ lập trình và phát triển một Game bằng những dự án mẫu, art, audio, blueprint logic sẵn có thậm chí là những dòng code C++ đã được viết trước.

### 1.3.5 Những ưu thế của Unreal Engine

* Công cụ đa dạng:

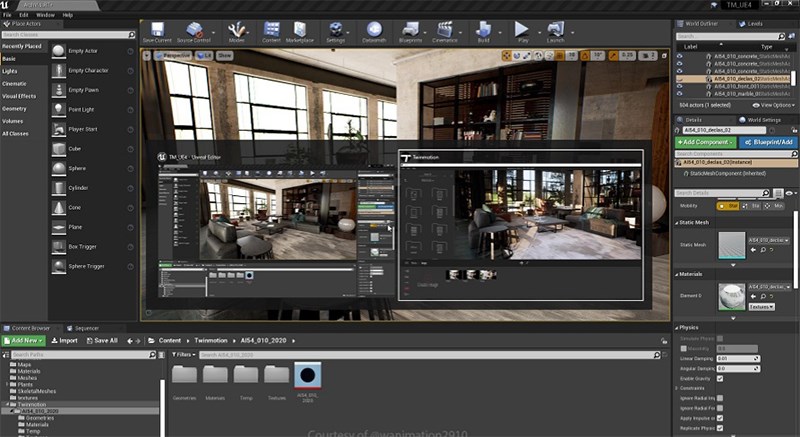
Unreal Engine cho phép sử dụng các công cụ tạo địa hình đồi núi, ao hồ,... tùy vào mục đích sử dụng của mỗi người. Không chỉ dừng lại ở đó, người dùng chỉ cần thêm các tài nguyên được tải từ [Quixel Megascan](https://quixel.com/megascans/) về và sắp xếp lên địa hình. Từ trồng cỏ, gắn cây xanh hay xếp đá thành bậc thang đều dễ dàng hơn bao giờ hết.



Hình 1. 6 Unreal Engine hỗ trợ nhiều công cụ

* Linh hoạt trong việc sáng tạo:

Người dùng sẽ được sử dụng các trình thiết kế games, [thiết kế nội thất](https://vi.wikipedia.org/wiki/Thi%E1%BA%BFt_k%E1%BA%BF_n%E1%BB%99i_th%E1%BA%A5t) (Architecture), thiết kế [TVC](https://vi.wikipedia.org/wiki/Phim_qu%E1%BA%A3ng_c%C3%A1o) cho các sản phẩm xe cộ (Automotive & Transportation), chương trình truyền hình và sự kiện trực tuyến (Broadcast & Live Events),... cho đến cả [phim điện ảnh](https://vi.wikipedia.org/wiki/Phim_%C4%91i%E1%BB%87n_%E1%BA%A3nh) và [phim truyền hình](https://vi.wikipedia.org/wiki/Phim_truy%E1%BB%81n_h%C3%ACnh) (Film & Television). Đặc biệt trong ngành kiến trúc, những bản vẽ thiết kế sẽ trở nên sống động và dễ dàng thuyết phục khách hàng hơn khi được thiết kế bằng Unreal Engine, vì tính chân thật và trực quan của nó.



Hình 1. 7 UE trong thiết kế nội thất

* Miễn phí cho mọi đối tượng:

Unreal Engine là công cụ thiết kế hoàn toàn miễn phí cho tất cả mọi người trải nghiệm và sử dụng. Unreal Engine đơn giản hóa giao diện và cách sử dụng để người dùng tự do sáng tạo mà vẫn đảm bảo được tính chuyên nghiệp. Mọi người đều có thể tự tạo cho mình một tựa game theo khả năng sáng tạo của mình một cách hoàn toàn miễn phí.

## 1.4 GIỚI THIỆU UNREAL ENGINE 5.4

### 1.4.1 Tổng quan về Unreal Engine 5.4

Unreal Engine 5.4 là phiên bản mới nhất của Unreal Engine, được Epic Games phát hành với nhiều cải tiến quan trọng. Phiên bản này mang lại những công nghệ tiên tiến giúp nâng cao chất lượng đồ họa, tối ưu hiệu suất và cải thiện khả năng phát triển game trên nhiều nền tảng khác nhau.UE 5.4 kế thừa và cải tiến các công nghệ quan trọng từ các phiên bản trước, bao gồm Lumen, Nanite, World Partition, đồng thời bổ sung nhiều tính năng mới hỗ trợ Motion Design, tối ưu hóa dò tia (Ray Tracing), Animation Retargeting, cùng nhiều công cụ giúp cải thiện hiệu suất và workflow trong quá trình phát triển.

### 1.4.2 Những cải tiến quan trọng trong Unreal Engine 5.4

a. Hệ thống Render & Lighting

* Nanite 2.0 - Hệ thống hiển thị đa giác ảo:
* Cho phép sử dụng mô hình 3D siêu chi tiết (triệu đa giác) mà không ảnh hưởng đến hiệu suất.
* Giúp tiết kiệm thời gian tối ưu hóa LOD (Level of Detail).
* Hỗ trợ Displacement Mapping, giúp mô hình có độ sâu tự nhiên hơn.



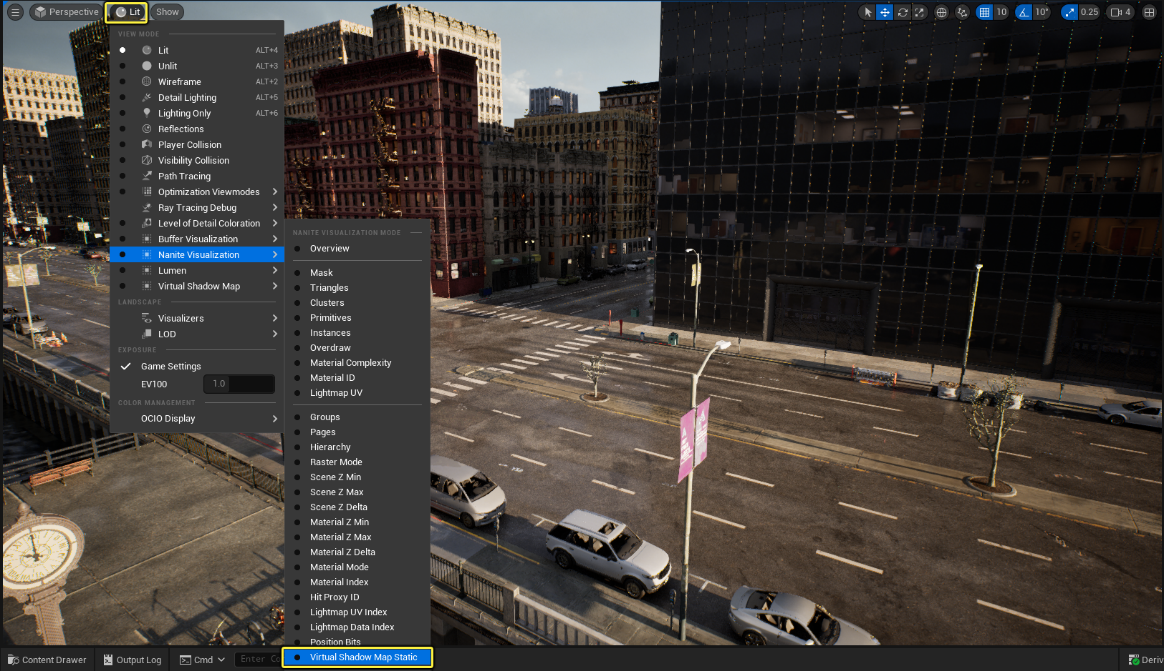
Hình 1. 8 Natie

* Lumen 2.0 - Hệ thống chiếu sáng toàn cục theo thời gian thực:
* Mô phỏng ánh sáng phản xạ từ bề mặt và môi trường xung quanh.
* Giúp cảnh game có hiệu ứng ánh sáng chân thực hơn mà không cần tính toán trước.
* Hỗ trợ Reflections (phản chiếu), Ambient Occlusion (đổ bóng môi trường) và Ray Tracing.



Hình 1. 9 Lumen

* Virtual Shadow Maps (VSM) - Hệ thống bóng đổ cải tiến:
* Cải thiện độ sắc nét và chính xác của bóng, đặc biệt là trong môi trường có nhiều chi tiết nhỏ.
* Giảm hiện tượng nhấp nháy và răng cưa của bóng động.



Hình 1. 10 Virtual Shadow Maps

* Temporal Super Resolution (TSR) 2.0:
* Hỗ trợ nâng cao độ phân giải hình ảnh với chi phí thấp hơn so với phương pháp Anti-Aliasing truyền thống.
* Giúp giữ lại chi tiết hình ảnh ngay cả khi chạy ở độ phân giải thấp hơn.



Hình 1. 11 Temporal Super Resolution

b. Hiệu suất và tối ưu hóa

* Giảm thời gian biên dịch Shader: UE 5.4 tối ưu quá trình biên dịch Shader, giúp giảm đáng kể thời gian tải dự án, mang lại trải nghiệm làm việc mượt mà hơn.
* Cải thiện Multi-Threading: Việc xử lý đa luồng được tối ưu hóa để tận dụng tối đa sức mạnh phần cứng, giúp engine vận hành hiệu quả hơn trên các hệ thống máy tính đa nhân.
* Hỗ trợ dò tia phần cứng nhanh hơn: Công nghệ Ray Tracing được tối ưu, giúp tăng tốc độ kết xuất ánh sáng và bóng đổ mà không làm giảm hiệu suất khung hình.



Hình 1. 12 Ray Tracing

c. Cải tiến hệ thống Animation & Rigging

Unreal Engine 5.4 tiếp tục hoàn thiện hệ thống Animation & Rigging, giúp các nhà phát triển dễ dàng tạo hoạt ảnh nhân vật với độ chính xác cao hơn. Những nâng cấp đáng chú ý:

Hệ thống Rig nhân vật mô-đun (Modular Character Rigging):

* Cho phép tạo và chỉnh sửa rig nhân vật trực tiếp trong Engine.
* Cung cấp thư viện rig mô-đun có thể tùy chỉnh.

Tự động chuyển hướng hoạt ảnh (Animation Retargeting):

* Hỗ trợ chuyển hướng hoạt ảnh giữa các nhân vật có tỷ lệ cơ thể khác nhau mà không cần chỉnh sửa thủ công.

Tối ưu hóa IK (Inverse Kinematics):

* Hệ thống IK được cải thiện giúp nhân vật di chuyển và phản ứng chân thực hơn với môi trường.



Hình 1. 13 Hệ thống Animation & Rigging

d. Hỗ trợ Virtual Production & MetaHuman

MetaHuman Creator: Được cải tiến để hỗ trợ tạo nhân vật chi tiết hơn, từ biểu cảm khuôn mặt đến chuyển động cơ thể.

Virtual Production Tools: Được cập nhật giúp làm phim với LED Volume trở nên chính xác hơn

A person with short hair

AI-generated content may be incorrect.

Hình 1. 14 Hệ thống Meta Human

e. Hệ thống Motion Design Mode hoàn toàn mới

UE 5.4 giới thiệu một chế độ chuyên biệt mang tên Motion Design Mode, cho phép các nhà thiết kế đồ họa chuyển động (Motion Graphics) tạo ra các hiệu ứng 2D và 3D phức tạp ngay trong Unreal Engine mà không cần đến các phần mềm bên ngoài như After Effects.



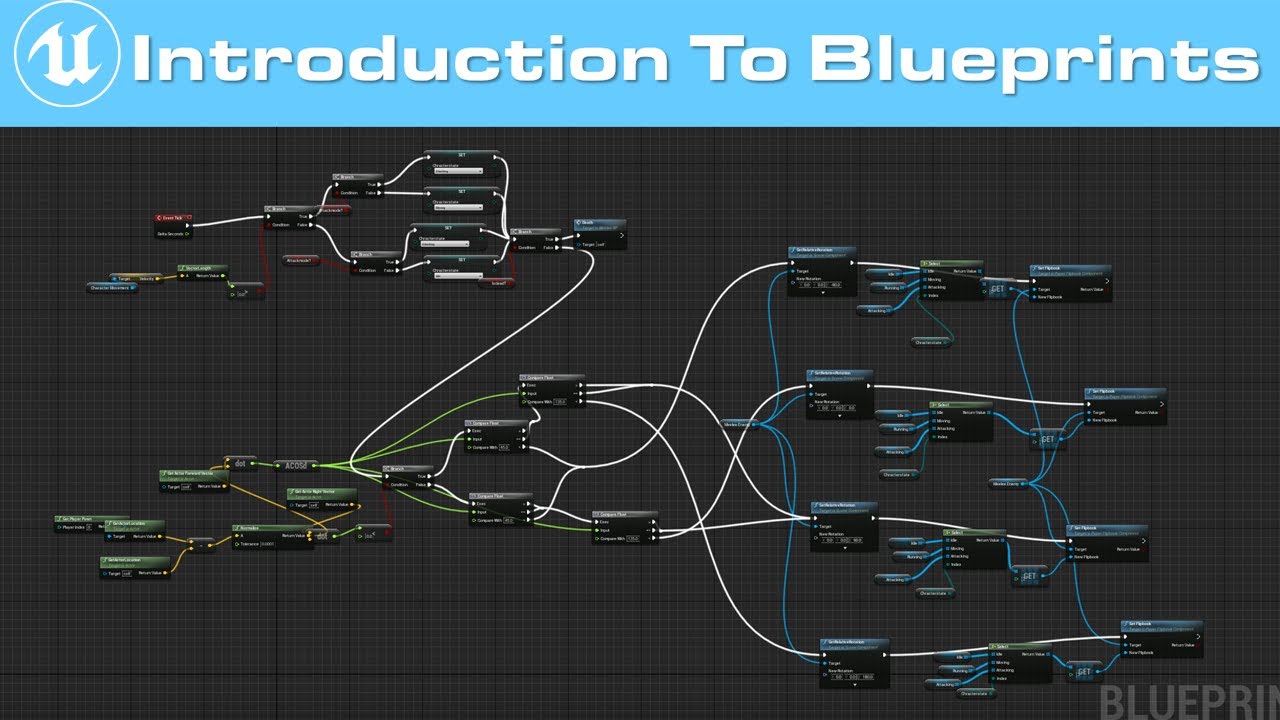
*Hình 1. 15 Hệ thống Motion Design Mode*

## 1.5 GIỚI THIỆU BLUEPRINTS VISUAL SCRIPTING

### 1.5.1 Blueprints là gì?

Blueprints là một hệ thống lập trình trực quan (Visual Scripting System) được tích hợp trong Unreal Engine, cho phép các nhà phát triển xây dựng logic cho trò chơi mà không cần sử dụng mã nguồn C++. Hệ thống này dựa trên mô hình nút kết nối (node-based programming), trong đó các hành vi, sự kiện và quy trình xử lý được thể hiện dưới dạng các khối logic kết nối với nhau.

Blueprints đóng vai trò quan trọng trong quy trình phát triển trò chơi nhờ vào tính trực quan, dễ sử dụng và khả năng tích hợp mạnh mẽ với C++, giúp tối ưu hóa quy trình phát triển và cải thiện hiệu suất vận hành của trò chơi.



Hình 1. 16 Blueprints

### 1.5.2 Ưu điểm của Blueprints so với lập trình truyền thống

Blueprints mang lại nhiều lợi ích đáng kể trong phát triển game, bao gồm:

* Tiết kiệm thời gian phát triển: Hệ thống trực quan giúp giảm thời gian viết mã và kiểm thử.
* Dễ tiếp cận: Không yêu cầu kiến thức lập trình sâu, phù hợp với nhà thiết kế trò chơi.
* Hệ thống Debug mạnh mẽ: Unreal Engine cung cấp công cụ Debug trực tiếp trong trình biên tập Blueprints.
* Tích hợp với C++: Có thể mở rộng Blueprints bằng C++ để tối ưu hiệu suất.
* Dễ dàng bảo trì: Việc chỉnh sửa logic trò chơi trở nên đơn giản hơn so với việc chỉnh sửa mã nguồn.

Tuy nhiên, Blueprints có nhược điểm là hiệu suất không cao bằng C++ do cơ chế dịch mã trung gian, điều này có thể ảnh hưởng đến các trò chơi yêu cầu xử lý nhanh như game FPS, RTS hoặc các dự án quy mô lớn.

### 1.5.3 Cấu trúc và thành phần của Blueprints

a. Các loại Blueprints trong Unreal Engine

Blueprints có nhiều loại khác nhau, mỗi loại phục vụ một mục đích cụ thể trong hệ thống:

|  |  |  |
| --- | --- | --- |
| **Loại Blueprint** | **Chức năng** | **Ứng dụng thực tế** |
| Blueprint Class | Tạo các đối tượng có hành vi riêng | Nhân vật, vật phẩm, cửa, AI |
| Blueprint Interface | Tạo giao tiếp giữa các Blueprints khác nhau | Hệ thống đối thoại, tương tác NPC |
| Blueprint Function Library | Tạo các hàm chung để sử dụng lại | Tính toán vật lý, kiểm tra va chạm |
| Blueprint Macro Library | Định nghĩa các nhóm logic tái sử dụng | Xử lý sự kiện, hệ thống hiệu ứng |
| Blueprint Widget (UI) | Xây dựng giao diện người dùng | HUD, menu chính, thanh máu.,, |
| Blueprint Component | Thêm chức năng cho nhiều đối tượng | Hệ thống sức khỏe, AI di chuyển |

b. Các thành phần chính trong Blueprint Class

Mỗi Blueprint Class trong Unreal Engine có các thành phần quan trọng sau:

* Event Graph: Khu vực chứa các sự kiện và luồng xử lý chính.
* Construction Script: Chạy trước khi đối tượng xuất hiện, thường được dùng để khởi tạo dữ liệu.
* Variables: Chứa dữ liệu của Blueprint, có thể là số, chuỗi, vector, boolean hoặc object.
* Functions: Chứa các hàm thực thi logic có thể tái sử dụng.
* Macros: Tương tự như functions nhưng linh hoạt hơn, giúp nhóm các đoạn mã ngắn lại.
* Components: Các thành phần có thể tái sử dụng như Camera, Light, AI Controller...

### Nguyên lý hoạt động của Blueprints

Blueprints trong Unreal Engine hoạt động dựa trên mô hình lập trình hướng sự kiện (Event-Driven Programming), trong đó các hành động (Actions) chỉ xảy ra khi một sự kiện (Event) được kích hoạt. Ngoài ra, Blueprints còn tuân theo nguyên lý lập trình hướng đối tượng (OOP), giúp tổ chức mã một cách chặt chẽ, có thể tái sử dụng và mở rộng.

a. Mô hình hoạt động của Blueprints bao gồm ba yếu tố chính:

* Sự kiện (Events): Kích hoạt logic trong Blueprint.
* Luồng điều khiển (Execution Flow): Xác định trình tự thực hiện các khối lệnh.
* Truy xuất dữ liệu (Data Flow): Xử lý thông tin và lưu trữ trạng thái trong quá trình thực thi.

b. Sự kiện hệ thống (System Events)

Các sự kiện này do Unreal Engine cung cấp sẵn, bao gồm:

* Event BeginPlay: Được gọi khi một đối tượng xuất hiện trong Level.
* Event Tick: Chạy liên tục mỗi frame (có thể điều chỉnh tốc độ cập nhật).
* Event Destroyed: Kích hoạt khi đối tượng bị xóa khỏi game.
* Event Overlap: Xảy ra khi một vật thể đi vào vùng va chạm của một đối tượng khác.
* Event Hit: Kích hoạt khi có va chạm vật lý giữa hai vật thể.

c. Luồng điều khiển (Execution Flow)

Trong Blueprint, các nút (nodes) được liên kết bằng hai loại kết nối chính:

* Kết nối trắng (Execution Flow): Xác định thứ tự thực hiện logic.
* Kết nối màu (Data Flow): Truyền dữ liệu giữa các nút.

d. Biến (Variables) trong Blueprints

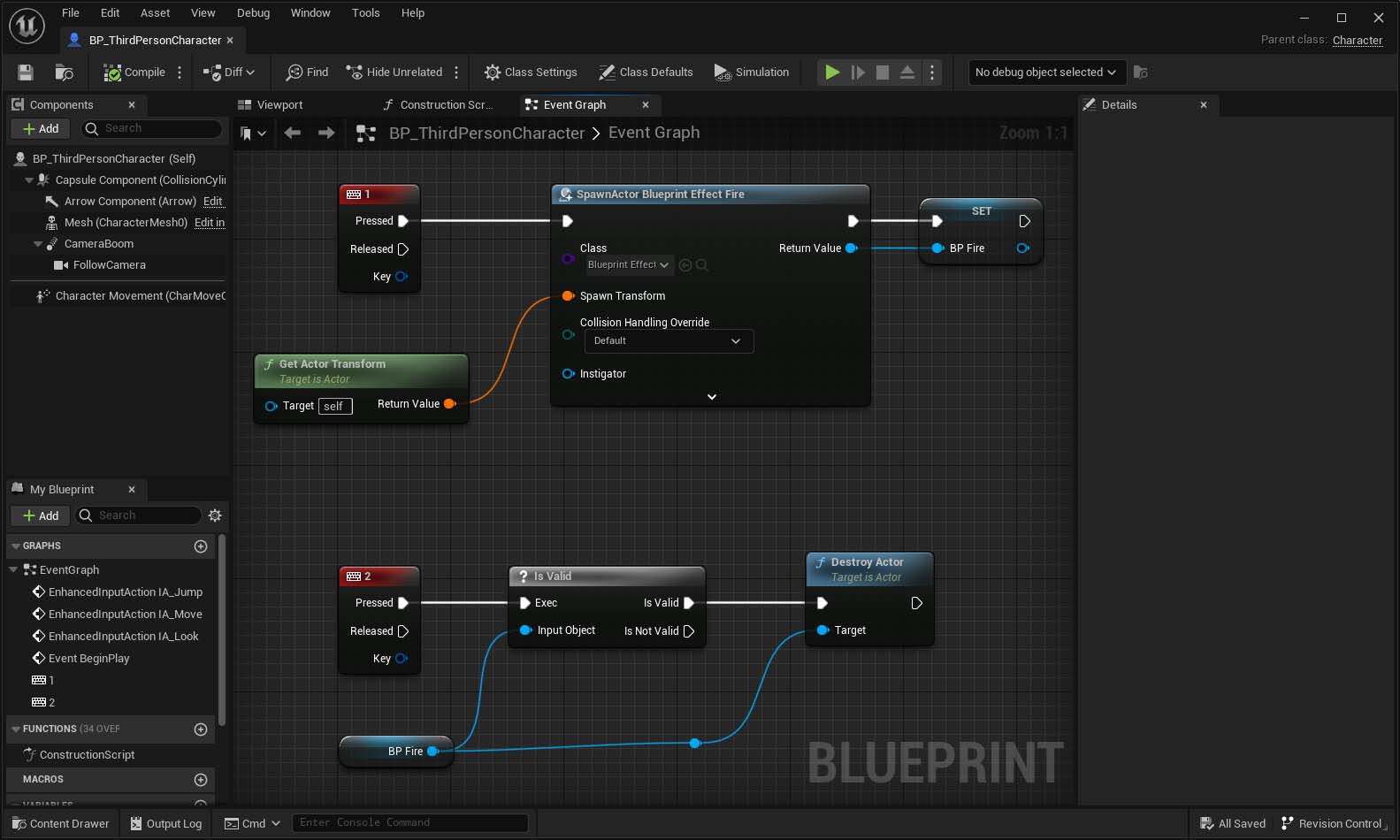
Blueprint hỗ trợ nhiều loại biến khác nhau, bao gồm:

* Boolean – True/False (bật/tắt).
* Integer – Số nguyên.
* Float – Số thực.
* Vector – Đại diện cho vị trí hoặc hướng.
* Object Reference – Tham chiếu đến một Actor khác trong game.

e. Flow Control (Điều khiển luồng chương trình)

Blueprint cung cấp nhiều công cụ để kiểm soát luồng chương trình, bao gồm:

|  |  |
| --- | --- |
| **Loại Node** | **Chức năng** |
| Branch (If-Else) | Kiểm tra điều kiện và chọn hướng thực thi phù hợp. |
| For Loop | Lặp một số lần cố định. |
| While Loop | Lặp khi điều kiện còn đúng. |
| Switch | Chọn một trong nhiều hướng đi khác nhau. |



Hình 1. 17 Giao diện Blueprints

## 1.6 GIỚI THIỆU AI SYSTEM

### 1.6.1 Giới thiệu tổng quan về AI System trong Unreal Engine

Unreal Engine cung cấp một hệ thống AI mạnh mẽ giúp xây dựng các nhân vật thông minh với hành vi phức tạp. AI System trong UE hoạt động dựa trên mô hình Agent-Based AI, trong đó mỗi NPC (Non-Player Character) được xem như một tác nhân độc lập có thể nhận thức môi trường, ra quyết định và thực hiện hành động.

Hệ thống AI trong Unreal Engine chủ yếu sử dụng:

* Behavior Tree (BT): Kiểm soát luồng suy nghĩ của AI.
* Blackboard (BB): Lưu trữ thông tin về trạng thái AI.
* AI Perception System: Giúp AI nhận thức thế giới (tầm nhìn, âm thanh, chạm).
* Navigation System: Hỗ trợ AI di chuyển thông minh.
* AI Tasks & Services: Xây dựng hành vi AI tùy chỉnh.

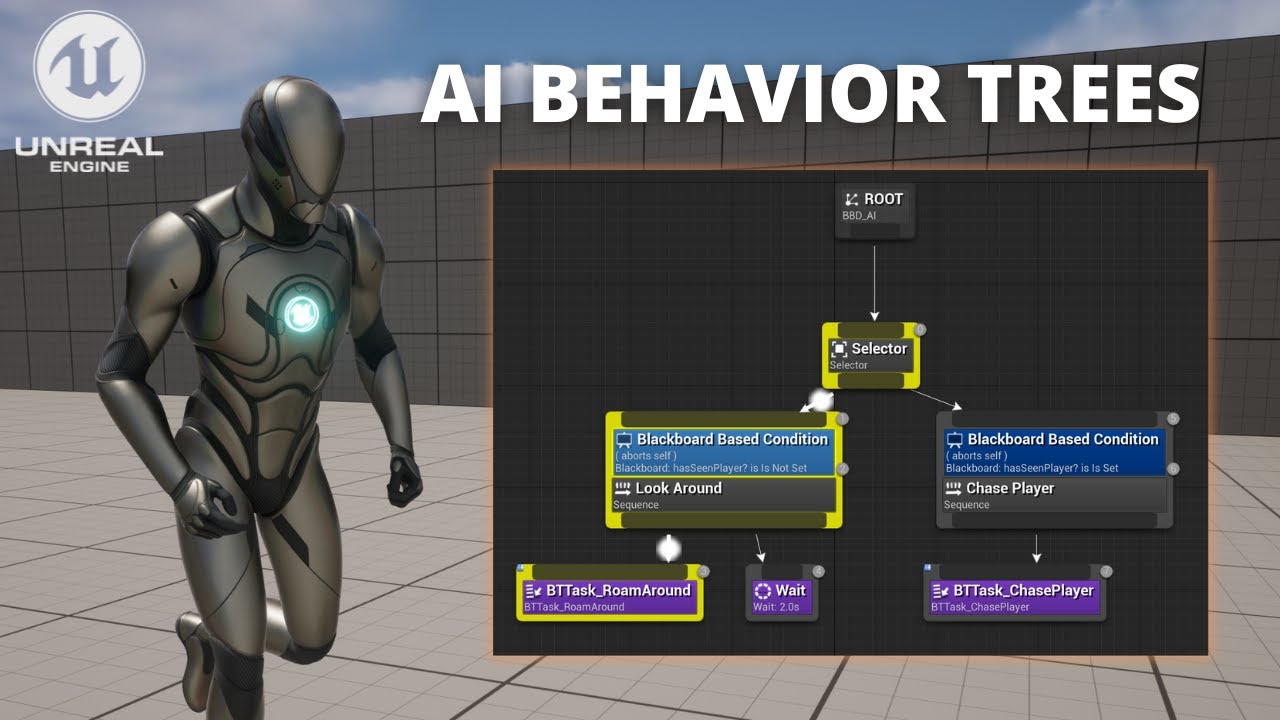
### 1.6.2 Behavior Tree (Cây hành vi) trong AI

a. Nguyên lý hoạt động của Behavior Tree

Behavior Tree (BT) là một cấu trúc phân cấp giúp điều khiển hành vi của AI theo trình tự logic. Mỗi BT bao gồm các nút (Nodes) có nhiệm vụ kiểm tra điều kiện hoặc thực hiện hành động.

Cấu trúc chính của BT gồm:

* Root Node (Nút gốc): Điểm bắt đầu của BT.
* Composite Nodes (Nút tổ hợp): Xác định cách các hành vi được thực hiện.
* Decorator Nodes (Nút trang trí): Đặt điều kiện cho một hành vi.
* Task Nodes (Nút nhiệm vụ): Thực hiện hành động cụ thể.



Hình 1. 18 AI Behavivor Trees

b. Các loại Composite Nodes trong Behavior Tree

* Selector (Chọn lựa): Kiểm tra từng nhánh con theo thứ tự. Nếu một nhánh thành công, nó dừng lại.
* Sequence (Chuỗi hành động): Thực hiện từng nhánh con từ trái sang phải. Nếu một nhánh thất bại, toàn bộ Sequence thất bại.
* Parallel (Song song): Chạy tất cả nhánh con cùng lúc.
* Random Selector (Lựa chọn ngẫu nhiên): Chọn một nhánh bất kỳ để thực hiện**.**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 1. 19 Cấu trúc AI Behavivor Trees

c. Ưu điểm

* Dễ quản lý và mở rộng: Cấu trúc dạng cây giúp lập trình viên dễ dàng thêm hoặc sửa đổi hành vi của AI mà không cần viết lại toàn bộ hệ thống.
* Linh hoạt và có thể tái sử dụng: Một cây hành vi có thể được áp dụng cho nhiều NPC với các điều chỉnh nhỏ.
* Tối ưu hiệu suất: AI chỉ thực hiện các hành động cần thiết, không chạy toàn bộ mã cùng lúc.
* Dễ kết hợp với Blueprint và C++: Có thể viết AI bằng cả hai cách để tận dụng sức mạnh của từng phương pháp.
* Dễ kiểm tra lỗi: Có thể debug từng nhánh hành vi để phát hiện lỗi.

d. Nhược điểm

* Cần tối ưu hóa: Nếu không thiết kế hợp lý, AI có thể tiêu tốn nhiều tài nguyên, làm giảm hiệu suất game.
* Không phù hợp với AI học sâu: Behavior Tree chỉ dựa trên quy tắc có sẵn, không thể tự học như Machine Learning.
* Khó điều chỉnh hành vi phức tạp: Nếu NPC cần phản ứng theo nhiều điều kiện thay đổi liên tục, cần kết hợp thêm hệ thống AI khác.

# CHƯƠNG II

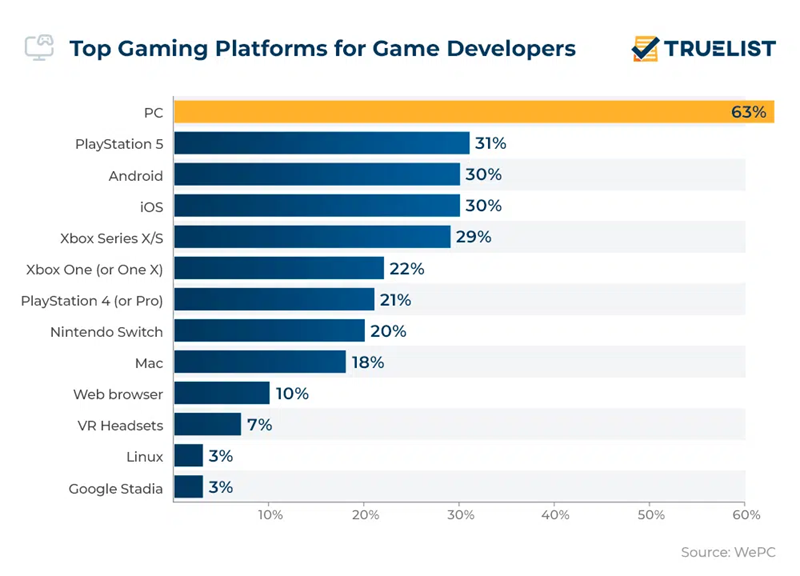
# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1 KHẢO SÁT THỰC TRẠNG

Trong những năm gần đây, thể loại game kinh dị ngày càng phát triển mạnh mẽ và thu hút đông đảo người chơi nhờ vào trải nghiệm căng thẳng, hồi hộp và đầy thử thách. Các tựa game hiện đại không chỉ tập trung vào hiệu ứng hù dọa (jumpscare) mà còn xây dựng bầu không khí ám ảnh, kết hợp với cơ chế gameplay sáng tạo để mang đến trải nghiệm độc đáo.

Tuy nhiên, một số tựa game kinh dị hiện nay vẫn gặp phải những hạn chế nhất định như: Cơ chế AI đơn giản, quái vật thường chỉ đuổi theo người chơi theo tuyến đường cố định, thiếu tính linh hoạt. Thiếu sự tương tác với môi trường, khiến trải nghiệm chưa thực sự chân thực.

Nhận thấy những điểm hạn chế đó, em quyết định nghiên cứu và phát triển game kinh dị 3D Nyctophobia trên Unreal Engine 5. Đây không chỉ là cơ hội để áp dụng những kiến thức về lập trình game mà còn giúp em tìm hiểu sâu hơn về cách xây dựng AI quái vật thông minh bằng Behavior Tree, tạo nên trải nghiệm rùng rợn và thử thách hơn cho người chơi.



Hình 2. 1 Số liệu thống kê nên tảng game được ưa chuộng toàn cầu 2023

## 2.2 MỘT SỐ TỰA GAME KINH DỊ TƯƠNG TỰ

* SCP: Containment Breach (2012)
* Nhà phát triển : Joonas Rikkonen
* Bối cảnh: Người chơi bị mắc kẹt trong một cơ sở nghiên cứu SCP và phải tìm cách trốn thoát khỏi các thực thể nguy hiểm.
* Cơ chế gameplay: Một số SCP có cơ chế đặc biệt, như SCP-173 chỉ di chuyển khi người chơi chớp mắt hoặc quay đi.
* Ưu điểm:
* AI quái vật độc đáo, mỗi SCP có cách hoạt động riêng, tạo ra nhiều thử thách khác nhau.
* Hệ thống chớp mắt giúp tăng cường sự căng thẳng và thử thách trong gameplay.
* Tính ngẫu nhiên của môi trường làm mỗi lần chơi đều khác biệt.
* Nhược điểm:
* Đồ họa không quá ấn tượng, có phần lỗi thời so với các game kinh dị hiện đại.
* Một số lỗi kỹ thuật có thể gây ảnh hưởng đến trải nghiệm.



Hình 2. 2 Game SCP: Containment Breach (2012)

* The Letter (2017)
* Nhà phát triển : Yangyang Mobile
* Bối cảnh: Game diễn ra tại thành phố Luxbourne, xoay quanh dinh thự ma ám Ermengarde, nơi ẩn chứa lời nguyền kinh hoàng và những bí ẩn chết chóc.
* Cơ chế gameplay: The Letter là game visual novel kinh dị, nơi người chơi đưa ra các lựa chọn ảnh hưởng đến cốt truyện. Game có nhiều tuyến nhân vật, mỗi người có góc nhìn riêng về lời nguyền. Một số phân đoạn yêu cầu tương tác nhanh (quick-time events) để sinh tồn.
* Ưu điểm:
* Phong cách visual novel kết hợp kinh dị, mang lại cách tiếp cận mới mẻ.
* Cốt truyện có nhiều hướng rẽ dựa trên lựa chọn của người chơi.
* Nhược điểm:
* Nhịp độ quá chậm, chủ yếu là đọc văn bản mà ít có gameplay thực sự.
* Không có cơ chế gameplay đủ hấp dẫn để tạo cảm giác sợ hãi mạnh mẽ.
* Một số đoạn hội thoại và tình tiết bị kéo dài không cần thiết, làm giảm sự căng thẳng.



Hình 2. 3 Game Letter (2017)

Sau khi nghiên cứu và phân tích các tựa game kinh dị có lối chơi tương tự, em đã rút ra những ưu, nhược điểm của từng game. Từ đó, em đã định hướng phát triển Nyctophobia với các cơ chế gameplay và yếu tố thiết kế tối ưu, nhằm mang lại trải nghiệm căng thẳng, hấp dẫn và độc đáo hơn. Các hệ thống và chức năng chính của game sẽ được trình bày chi tiết trong các phần tiếp theo.

## 2.3 GIỚI THIỆU GAME NYCTOPHOBIA

### 2.3.1 Khái quát về game

* Tên game : Nyctophobia
* Thể loại: Survival Horror, First-Person ( Kinh dị sinh tồn , góc nhìn thứ nhất )
* Nền tảng: PC (Windows)
* Engine: Unreal Engine 5 (Blueprints)
* Người chơi: Single-player
* Tông màu & Phong cách đồ họa: Tối, u ám, sử dụng ánh sáng động để tăng sự căng thẳng
* Đối tượng người chơi: 16+ (có yếu tố kinh dị và tâm lý căng thẳng).

### 2.3.2 Cốt truyện

Câu chuyện của Nyctophobia diễn ra tại một cơ sở nghiên cứu bị lãng quên dưới lòng đất—một nơi từng là trung tâm của những thí nghiệm tâm lý tàn khốc liên quan đến nỗi sợ bóng tối. Những bí mật kinh hoàng của nó đã bị chôn vùi theo thời gian, cho đến khi một nhà báo điều tra nhận được một bức thư nặc danh tiết lộ rằng nơi này che giấu một sự thật khủng khiếp.

Bị cuốn vào cơn khát khao khám phá sự thật, anh quyết định đột nhập vào cơ sở hoang tàn này. Nhưng ngay khi đặt chân vào, cánh cửa phía sau anh đóng sập lại, nhấn chìm anh trong bóng tối ngột ngạt. Khi cố gắng tìm lối thoát, anh dần nhận ra mình không hề đơn độc—một thứ gì đó ẩn mình trong bóng tối, theo dõi từng bước đi của anh. Nó không xuất hiện khi anh nhìn thẳng vào nó, nhưng chỉ cần quay lưng đi, nó sẽ lặng lẽ tiến gần hơn.

Mỗi góc khuất, mỗi nơi tối tăm đều có thể là nơi ẩn náu của thực thể bí ẩn ấy. Không còn lựa chọn nào khác, anh buộc phải lần mò trong bóng tối, giải mã những mảnh ghép rời rạc của quá khứ và tìm cách thoát ra trước khi quá muộn.

### 2.3.3 Gameplay

a. Điều khiển nhân vật

|  |  |
| --- | --- |
| **HÀNH ĐỘNG** | **PHÍM TẮT** |
| **Di chuyển** | |  | | --- | |  |  |  | | --- | | WASD | |
| **Nhìn xung quanh** | |  | | --- | |  |  |  | | --- | | Chuột | |
| **Chạy nước rút** | |  | | --- | |  |   Shift |
| **Bật/Tắt đèn pin** | F |
| **Tương tác với vật phẩm** | |  | | --- | |  |  |  | | --- | | E | |
| **Mở menu tạm dừng** | P |

b. Tài nguyên trong trò chơi

Tài liệu, mảnh giấy câu đố, chìa khóa, đèn pin, cửa, các đồ vật môi trường,..

c. Mục tiêu

Khám phá được câu chuyện đang xảy ra trong trò chơi, tìm kiếm mật mã, tránh né quái vật tìm cách mở cánh cửa chính để trốn thoát.

d. Cơ chế chính

* Cơ chế sinh tồn: Trốn vào tủ, góc khuất để tránh bị phát hiện
* Cơ chế quái vật: Khuất Quỷ chỉ di chuyển khi không bị nhìn thấy, người chơi phải liên tục kiểm soát góc nhìn để không bị bất ngờ.
* Giải đố và khám phá: Tìm kiếm chìa khóa, mật mã, ghi chú để mở cửa, một số câu đố cần suy luận để tìm ra đáp án.

### 2.3.4 AI trong game

a. Hành vi AI trong game

Roam Around (Di chuyển ngẫu nhiên):

* Quái vật sẽ di chuyển ngẫu nhiên trong một vùng nhất định hoặc đi đến các điểm ngẫu nhiên.
* Hệ thống sẽ chọn một vị trí ngẫu nhiên trong môi trường và cho quái vật di chuyển đến đó.
* Nếu quái vật không có mục tiêu rõ ràng, nó sẽ tiếp tục hành vi này cho đến khi có sự thay đổi.

Chase Player (Truy đuổi người chơi):

* Khi quái vật nhìn thấy người chơi nó sẽ chuyển sang trạng thái truy đuổi.
* Quái vật sẽ liên tục di chuyển về phía người chơi cho đến khi đạt được một khoảng cách nhất định.
* Nếu người chơi thay đổi vị trí hoặc di chuyển nhanh hơn, quái vật sẽ điều chỉnh hướng di chuyển.

Attack Player (Tấn công người chơi):

* Khi quái vật đến đủ gần người chơi, nó sẽ chuyển sang trạng thái tấn công và thực hiện hành động tấn công.

Freeze State (Trạng thái Bị Đóng Băng):

* Khi người chơi sử dụng khả năng Freeze State, quái vật bị tạm thời đóng băng và không thể thực hiện hành động như Roam Around, Chase Player, hoặc Attack Player.

b. Kỹ thuật lập trình

Behavior Tree (Cây Hành Vi):

* Quản lý các hành vi của AI theo cách có cấu trúc, dễ bảo trì và mở rộng.
* Mỗi hành vi của quái vật (di chuyển ngẫu nhiên, truy đuổi, tấn công, v.v.) được biểu diễn dưới dạng các node trong một cây hành vi.

Pathfinding và NavMesh:

* Quái vật có thể tìm đường đi an toàn và hiệu quả trong môi trường phức tạp.
* Sử dụng NavMesh (mạng lưới điều hướng) để quái vật tính toán đường đi từ vị trí hiện tại đến mục tiêu.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2. 4 Cây hành vi của AI

### 2.3.5 Thiết kế màn chơi

Cấu trúc level:

* Khu 1 - Tầng hầm: Người chơi tỉnh dậy và bị mắc kẹt, tìm đèn pin.
* Khu 2 – Hành lang: Khám phá khu vực, tìm kiếm mảnh ghi chú
* Khu 3 – Phòng 1 : Tìm chìa khóa mở các căn phòng khác và thu thập manh mối , mật mã rải rác.
* Khu 4 - Phòng điều khiển: Tìm kiếm manh mối câu đố.
* Khu 5 – Phòng trung tâm: Chạm trán quái vật. Mở cửa chính thoát khỏi khu vực.

A diagram of a company

AI-generated content may be incorrect.

Hình 2. 5 Cấu trúc bản đồ game

### 2.3.6 Hệ thống UI/UX

* Màn hình chính
* Màn hình cài đặt
* Màn hình Credit
* Màn hình tạm dừng
* Màn hình thua cuộc
* Màn hình chiến thắng

### 2.3.7 Hiệu ứng môi trường

* Môi trường tối tăm, cũ kĩ, tạo cảm giác rùng rợn
* Hiệu ứng ánh sáng: Đèn sáng yếu, màu sắc tùy theo khu vực.
* Nhạc nền: Tiếng rùng rợn, không khí căng thẳng.
* Hiệu ứng âm thanh : Tiếng bước chân, tiếng thở, tiếng quái vật, tiếng cánh cửa, tiếng tương tác với đồ vật,..

## 2.4 CÁC ĐỐI TƯỢNG TƯƠNG TÁC VỚI HỆ THỐNG

Hệ thống chỉ có một tác nhân chính duy nhất, đó là Người chơi. Vai trò của tác nhân được thể hiện trong bảng sau:

|  |  |  |
| --- | --- | --- |
| STT | Tên tác nhân | Mô tả |
| 1 | Người chơi | Là người sử dụng game. Người chơi trực tiếp điều khiển nhân vật chính, thực hiện các hành động như di chuyển, tương tác với vật thể, giải đố, tránh né quái vật, sử dụng vật phẩm và trải nghiệm nội dung game. |
| 2 | AI quái vật | Là thực thể do hệ thống điều khiển; tự động di chuyển (Roam Around), truy đuổi (Chase Player), tấn công (Attack Player) hoặc bị bất động khi bị “Freeze” tùy theo logic Behavior Tree, nhằm tạo thử thách cho người chơi. |

## 2.5 SƠ ĐỒ USECASE SỬ DỤNG (USECASE DIAGRAM)

### 2.5.1. Use case tổng quát

A diagram of a game

AI-generated content may be incorrect.

Hình 2. 6 Use case tổng quát

### 2.5.2. Use case chi tiết

**2.5.2.1. Use Case New game**

a. Biểu đồ Use Case New game.

A diagram of a network

AI-generated content may be incorrect.

Hình 2. 7 Biểu đồ Use Case New game

b. Đặc tả Use Case Bắt đầu game.

|  |  |
| --- | --- |
| **Use case** | New game. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi bắt đầu một trò chơi mới. |
| **Điều kiện trước** | Người chơi đã cài đặt game, mở ứng dụng game. |
| **Điều kiện sau** | Người chơi đã bắt đầu một trò chơi mới. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. **Người chơi** nhấn vào nút **"New game"** trong menu chính của game. | 1. **Hệ thống** hiển thị màn hình Loading. |
|  | 1. **Hệ thống** bắt đầu tải trò chơi. |
| 1. **Người chơi** bắt đầu chơi. 2. **Kết thúc use case.** |  |

**2.5.2.2. Use Case Load game**

a. Biểu đồ Use Case Load game.

A diagram of a load game

AI-generated content may be incorrect.

Hình 2. 8 Biểu đồ Use Case Load game

b. Đặc tả Use Case Tiếp tục.

|  |  |
| --- | --- |
| **Use case** | Load game. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi tải một trò chơi đã lưu trước đó. |
| **Điều kiện trước** | Người chơi đã bắt đầu một trò chơi mới, và đã lưu tiến trình ở điểm lưu. |
| **Điều kiện sau** | Người chơi đã tải vào game từ điểm lưu trước đó. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi nhấn vào nút "Load Game" trong menu chính. | 1. Hệ thống hiển thị danh sách các file lưu (tối đa 6 slot). |
| 1. Người chơi chọn file lưu mong muốn. | 1. Hệ thống kiểm tra trạng thái file lưu được chọn. |
|  | 5a. Nếu file lưu tồn tại: - Hệ thống hiển thị màn hình Loading. - Hệ thống tải dữ liệu game từ file lưu. - Chuyển người chơi vào game tại vị trí đã lưu.  5b. Nếu không có dữ liệu lưu: - Hệ thống thông báo: "Chưa có dữ liệu lưu." |
| 1. Kết thúc use case. |  |

**2.5.2.3. Use Case Setting**

a. Biểu đồ Use Case Setting.

A diagram of a computer

AI-generated content may be incorrect.

Hình 2. 9 Biểu đồ Use Case Setting

b. Đặc tả Use Case Setting.

|  |  |
| --- | --- |
| **Use case** | Setting. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi thay đổi các thiết lập trò chơi. |
| **Điều kiện trước** | Người chơi đã vào menu chính và nhấn vào nút "Setting". |
| **Điều kiện sau** | Các thiết lập mới được áp dụng hoặc lưu lại tùy theo nhu cầu. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi nhấn nút "Setting" từ menu chính. | 1. Hệ thống hiển thị màn hình thiết lập. |
| 1. Người chơi thay đổi các thiết lập ( Graphic, Window Mode, Resolution, Music, SFX ) |  |
| 1. Người chơi nhấn nút "Apply" | 1. Hệ thống áp dụng các thay đổi |
| 1. Người chơi quay lại menu chính hoặc tiếp tục chơi. | 1. Kết thúc Use Case. |

**2.5.2.4. Use Case Save game.**

a. Biểu đồ Use Case Save game.

A diagram of a game

AI-generated content may be incorrect.

Hình 2. 10 Biểu đồ Use Case Save game

b. Đặc tả Use Case Save game.

|  |  |
| --- | --- |
| **Use case** | Save game. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi lưu lại tiến trình chơi hiện tại. |
| **Điều kiện trước** | Người chơi đã vào trong game và mở Pause Menu. |
| **Điều kiện sau** | Tiến trình game hiện tại được lưu thành file save. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi mở Pause Menu bằng cách nhấn phím Pause. | 1. Hệ thống hiển thị Pause Menu. |
| 1. Người chơi chọn "Save Game" trong Pause Menu. | 1. Hệ thống mở màn hình chọn file lưu (slot save). |
| 1. Người chơi chọn slot và xác nhận lưu. | 1. Hệ thống ghi lại tiến trình hiện tại vào slot save. |
| 1. Kết thúc Use Case. |  |

**2.5.2.5. Use Case Pause menu.**

a. Biểu đồ Use Case Pause menu.

A diagram of a person

AI-generated content may be incorrect.

Hình 2. 11 Biểu đồ Use Case Pause menu

b. Đặc tả Use Case Pause menu.

|  |  |
| --- | --- |
| **Use case** | Pause menu. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi tạm dừng trò chơi và thực hiện các thao tác. |
| **Điều kiện trước** | Người chơi đang ở trong game. |
| **Điều kiện sau** | Người chơi thực hiện xong thao tác mong muốn. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi nhấn phím Pause ( P ) | 1. Hệ thống tạm dừng game và hiển thị Pause Menu. |
| 1. Người chơi chọn một trong các tùy chọn trong Pause Menu   - Resume  - Save Game  - Load Game  - Credit  - Exit  - Back to Menu | 4.1. Hệ thống tiếp tục game.  4.2. Hệ thống chuyển sang màn hình Save Game.  4.3. Hệ thống chuyển sang màn hình Load Game.  4.4. Hệ thống hiển thị thông tin Credit.  4.5. Hệ thống đóng game và thoát về Desktop.  4.6. Hệ thống thoát về màn hình Menu chính. |
| 5. Kết thúc Use Case. |  |

**2.5.2.6. Use Case Credit.**

a. Biểu đồ Use Case Credit.

A diagram of a credit

AI-generated content may be incorrect.

Hình 2. 12 Biểu đồ Use Case Credit

b. Đặc tả Use Case Credit.

|  |  |
| --- | --- |
| **Use case** | Credit. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi xem thông tin về nhà phát triển game và có thể xem hướng dẫn cách chơi. |
| **Điều kiện trước** | Người chơi đang ở trong menu chính hoặc Pause Menu. |
| **Điều kiện sau** | Người chơi đọc xong Credit hoặc cách chơi, và quay lại màn hình trước đó. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi chọn mục "Credit" từ menu hoặc Pause Menu. | 1. Hệ thống hiển thị màn hình Credit với thông tin nhà phát triển. |
| 3. Người chơi nhấn nút "How to Play" trong màn hình Credit. | 4. Hệ thống hiển thị màn hình hướng dẫn cách chơi. |
| 5. Người chơi đọc xong và nhấn nút "Back". | 6. Hệ thống quay lại màn hình Credit hoặc quay về menu/pause menu trước đó. |
| 7. Kết thúc Use Case. |  |

**2.5.2.7. Use Case Control Character.**

a. Biểu đồ Use Case Control Character.

A diagram of a control character

AI-generated content may be incorrect.

Hình 2. 13 Biểu đồ Use Case Control Character

b. Đặc tả Use Case Control Character.

|  |  |
| --- | --- |
| **Use case** | Control Character. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi điều khiển nhân vật di chuyển, tương tác, hành động trong thế giới game. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Người chơi đang ở trong game. (đã qua menu) | |
| **Điều kiện sau** | Nhân vật phản hồi các thao tác điều khiển của người chơi. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi sử dụng các phím di chuyển (WASD). | 1. Nhân vật di chuyển theo hướng nhập liệu. |
| 3. Người chơi điều chỉnh camera bằng chuột. | 4. Camera quay theo góc nhìn người chơi điều khiển. |
| 5. Kết thúc Use Case. |  |

**2.5.2.8. Use Case Object Interaction.**

a. Biểu đồ Use Case Object Interaction.

A blue oval with black text

AI-generated content may be incorrect.

Hình 2. 14 Biểu đồ Use Case Object Interaction

b. Đặc tả Use Case Object Interaction.

|  |  |
| --- | --- |
| **Use case** | Object Interaction. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Cho phép người chơi tương tác với các đối tượng trong game bằng phím E. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Người chơi đang ở trong game (không ở menu hay pause).  Nhân vật ở gần một đối tượng có thể tương tác. | |
| **Điều kiện sau** | Hành động tương tác được thực thi. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi di chuyển đến gần một đối tượng có thể tương tác. | 1. Hệ thống hiển thị icon tương tác. |
| 3. Người chơi nhấn phím **E**. | 4. Hệ thống xác định loại đối tượng và gọi event tương tác con tương ứng. |
|  | 5. Hệ thống thực thi hành động. |
|  | 6. Kết thúc Use Case. |

**2.5.2.9. Use Case Freeze State.**

a. Biểu đồ Use Case Freeze State.

A diagram of a frozen state

AI-generated content may be incorrect.

Hình 2. 15 Biểu đồ Use Case Freeze State

b. Đặc tả Use Case Freeze State.

|  |  |
| --- | --- |
| **Use case** | Freeze State. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Khi người chơi nhìn thẳng vào quái vật, quái vật sẽ bị bất động (không di chuyển). |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Người chơi ở trong game và có quái vật xuất hiện.  Quái vật nằm trong tầm nhìn của camera người chơi. | |
| **Điều kiện sau** | Quái vật chuyển sang trạng thái bất động cho đến khi người chơi rời tầm nhìn. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi điều khiển camera/cursor để nhìn thẳng vào quái vật. | 1. Hệ thống phát hiện hướng nhìn (gaze) của người chơi đang hướng đến quái vật. |
|  | 1. Hệ thống kích hoạt trạng thái Freeze cho quái vật. |
|  | 5. Hệ thống thực thi hành động. |
|  | 6. Kết thúc Use Case. |

**2.5.2.9. Use Case Exit game.**

a. Biểu đồ Use Case Exit game.

A blue oval with black text

AI-generated content may be incorrect.

Hình 2. 16 Biểu đồ Use Case Exit game

b. Đặc tả Use Case Exit game.

|  |  |
| --- | --- |
| **Use case** | Exit game. |
| **Tác nhân** | Người chơi. |
| **Mục đích** | Người chơi thoát khỏi game. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Người chơi đang trong game hoặc menu chính. | |
| **Điều kiện sau** | Game được đóng và thoát. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Người chơi nhấn phím Exit hoặc chọn tùy chọn "Exit" trong menu chính hoặc Pause Menu. | 1. Hệ thống thực thi lệnh thoát. |
| 1. Kết thúc Use Case. |  |

**2.5.2.10. Use Case Roam Around.**

a. Biểu đồ Use Case Roam Around.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2. 17 Biểu đồ Use Case Roam Around

b. Đặc tả Use Case Roam Around.

|  |  |
| --- | --- |
| **Use case** | Roam Around. |
| **Tác nhân** | AI quái vật. |
| **Mục đích** | Quái vật di chuyển ngẫu nhiên trong khu vực. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Quái vật đã được khởi tạo trên bản đồ. | | Quái vật không bị đóng băng. | |
| **Điều kiện sau** | Quái vật chuyển về trạng thái Idle và tiếp tục Roam theo chu kỳ. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Quái vật ở trạng thái Idle. | 1. Hệ thống khởi tạo task “Roam Around” cho quái vật.   2a. Nếu đến đích → chuyển quái vật về Idle.  2b. Nếu gặp chướng ngại → hủy di chuyển. |
| 1. Kết thúc Use Case. |  |

**2.5.2.11. Use Case Chase Player.**

a. Biểu đồ Use Case Chase Player.

A diagram of a game

AI-generated content may be incorrect.

Hình 2. 18 Biểu đồ Use Case Chase Player

b. Đặc tả Use Case Chase Player.

|  |  |
| --- | --- |
| **Use case** | Chase Player. |
| **Tác nhân** | AI quái vật. |
| **Mục đích** | Quái vật theo dõi và truy đuổi người chơi khi phát hiện. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | Quái vật đang trong trạng thái Roam Around hoặc Idle. | | Người chơi nằm trong phạm vi tầm nhìn.  Quái vật chưa bị đóng băng (Freeze). | |
| **Điều kiện sau** | Quái vật chuyển sang trạng thái Chase và di chuyển về phía người chơi.  Quái vật tiếp tục bám sát người chơi cho đến khi không còn nhìn thấy hoặc không còn ở trong tầm gần. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Quái vật ở trạng thái Chase. | 1. Hệ thống kiểm tra xem người chơi có trong tầm nhìn của quái vật không. Nếu có, quái vật chuyển sang chế độ Chase Player. |
| 1. Quái vật sẽ di chuyển về phía người chơi bằng cách sử dụng node Move To của UE5 Behavior Tree. | 1. Cập nhật liên tục vị trí người chơi. |
| 1. Kết thúc Use Case. |  |

**2.5.2.12. Use Case Attack Player.**

a. Biểu đồ Use Case Attack Player.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2. 19 Biểu đồ Use Case Attack Player

b. Đặc tả Use Case Attack Player.

|  |  |
| --- | --- |
| **Use case** | Attack Player. |
| **Tác nhân** | AI quái vật. |
| **Mục đích** | Quái vật tấn công người chơi khi nó đến gần và đủ điều kiện. |
| **Điều kiện trước** | |  | | --- | |  |   Quái vật đang trong trạng thái Chase Player và đã đến gần người chơi trong phạm vi tấn công.  Người chơi không thoát khỏi phạm vi tấn công của quái vật. |
| **Điều kiện sau** | Quái vật thực hiện hành động tấn công. |
| **Luồng sự kiện chính** | |
| Tác nhân | Hệ thống |
| 1. Quái vật ở trạng thái Attack. | 1. Kiểm tra khoảng cách giữa quái vật và người chơi. Nếu khoảng cách đủ gần, chuyển sang task tấn công. |
| 1. Quái vật sẽ gây sát thương cho người chơi nếu tấn công thành công. | 1. Hệ thống chuyển sang màn hình thua cuộc. |
| 1. Kết thúc Use Case. |  |

## 2.6 SƠ ĐỒ HOẠT ĐỘNG (ACTIVITY DIAGRAM)

**2.6.1. Sơ đồ hoạt động New game**

**A diagram of a game

AI-generated content may be incorrect.**

Hình 2. 20 Sơ đồ hoạt động New game

* + 1. **Sơ đồ hoạt động Load game**

**A diagram of a flowchart

AI-generated content may be incorrect.**

Hình 2. 21 Sơ đồ hoạt động Load game

* + 1. **Sơ đồ hoạt động Setting**

**A diagram of a process

AI-generated content may be incorrect.**

Hình 2. 22 Sơ đồ hoạt động Setting

* + 1. **Sơ đồ hoạt động Save game**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2. 23 Sơ đồ hoạt động Save game

* + 1. **Sơ đồ hoạt động Pause menu**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2. 24 Sơ đồ hoạt động Pause menu

* + 1. **Sơ đồ hoạt động Credit**

**A diagram of a workflow

AI-generated content may be incorrect.**

Hình 2. 25 Sơ đồ hoạt động Credit

* + 1. **Sơ đồ hoạt động Exit game**

**A diagram of a computer

AI-generated content may be incorrect.**

Hình 2. 26 Sơ đồ hoạt động Exit game

* + 1. **Sơ đồ hoạt động Control Character**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2. 27 Sơ đồ hoạt động Control Character

* + 1. **Sơ đồ hoạt động Object Interaction**