

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



PHẠM DUY THÁI - 51702180
LÊ TRẦN MINH TÂM - 51702174

XÂY DỰNG GAME 3D
FOR THE CROWN

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KỸ THUẬT PHẦN MỀM

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



PHẠM DUY THÁI - 51702180
LÊ TRẦN MINH TÂM - 51702174

XÂY DỰNG GAME 3D
FOR THE CROWN

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KỸ THUẬT PHẦN MỀM

Người hướng dẫn
Ths. Vũ Đình Hồng

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Vũ Đình Hồng đã góp ý và định hướng cho chúng em trước khi quá trình làm đồ án được bắt đầu. Tuy thời gian và năng lực còn hạn chế nên chúng em không thể hoàn thiện đầy đủ và trọn vẹn các yêu tố cần thiết và hấp dẫn người chơi cho một tựa game, nhưng những góp ý của thầy đã cho chúng em một hướng đi để làm theo cho đến khi hoàn thành đồ án, không để bị lạc đê mất thời gian.

Chúng em xin cảm ơn thầy rất nhiều.

TP. Hồ Chí Minh, ngày ... tháng ... năm 20..

Tác giả

(Ký tên và ghi rõ họ tên)

Phạm Duy Thái

Lê Trần Minh Tâm

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của Ths. Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày ... tháng ... năm 20..

Tác giả

(Ký tên và ghi rõ họ tên)

Phạm Duy Thái

Lê Trần Minh Tâm

Xây dựng game 3D sử dụng Unreal Engine – For The Crown

TÓM TẮT

Đề tài này là quá trình tìm hiểu cơ bản về Unreal Engine, sau đó áp dụng các hiểu biết để xây dựng demo, bao gồm các nội dung:

- Thông tin tổng quát của Engine
- Giới thiệu cơ bản giao diện và cách sử dụng Unreal Engine
- Xây dựng map từ các công cụ có sẵn trong Engine
- Xây dựng AI trong Unreal Engine
- Các level trong Game
- Các asset sử dụng

Demo được xây dựng với phiên bản Unreal Engine 5.2, gameplay Third Person. Người chơi sẽ vượt chướng ngại vật (bao gồm AI và câu đố) để đến điểm kết thúc là thành công.

Building a 3D game with Unreal Engine – FOR THE CROWN

ABSTRACT

This project is about learning basic of Unreal Engine, then apply the knowledge to build a demo, the content which included in is:

- General information of Unreal Engine
- Display and Instruction of Unreal Engine UI and Functions
- The process of building map from Unreal Engine
- The process of making AI in Unreal Engine
- The levels of Game
- The assets have been used

The demo was built from Unreal Engine 5.2, with Third Person gameplay. The Player will have to go pass through obstacles (which are AIs and quiz) to get to the final destination.

MỤC LỤC

DANH MỤC HÌNH VẼ.....	viii
DANH MỤC BẢNG BIỂU.....	xii
DANH MỤC CÁC CHỮ VIẾT TẮT	xii
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài	1
1.2 Mục tiêu thực hiện đề tài và mô tả đề tài.....	1
<i>1.2.1 Mục tiêu thực hiện đề tài</i>	<i>1</i>
<i>1.2.2 Mô tả đề tài</i>	<i>1</i>
1.3 Hướng thực hiện đề tài	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	4
2.1 Giới thiệu về Unreal Engine (“Unreal Engine 5.2 Documentation,” n.d.).....	4
<i>2.1.1 Khái niệm game engine</i>	<i>4</i>
<i>2.1.2 Lịch sử các phiên bản của Unreal Engine.....</i>	<i>4</i>
<i>2.1.3 Các lĩnh vực trong UE5</i>	<i>8</i>
<i>2.1.4 Các kiến thức nền tảng về Unreal Editor</i>	<i>14</i>
<i>2.1.5 Các thuật ngữ trong Unreal Engine.....</i>	<i>20</i>
<i>2.1.6 Các mode thường được sử dụng trong Unreal Editor</i>	<i>23</i>
2.2 Mixamo	26
2.3 Cargo Kitbash 3D	26
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ.....	27
3.1 Sơ đồ Use-case của Game.....	27
3.2 Sơ đồ ERD	31

3.3 Sơ đồ Sequence xử lý của game.....	32
3.4 Các sơ đồ flowchart của các actor trong game	33
3.4.1 <i>Flowchart của AI Scout</i>	33
3.4.2 <i>Flowchart của AI Defender</i>	34
3.4.3 <i>Flowchart của Target map1</i>	34
3.4.4 <i>Flowchart của Target map2</i>	36
3.4.5 <i>Flowchart của Target map3</i>	37
3.4.6 <i>Flowchart Shelter</i>	39
3.4.7 <i>Noisy thing</i>	40
CHƯƠNG 4. QUÁ TRÌNH THỰC HIỆN ĐỒ ÁN.....	41
4.1 Lên ý tưởng về gameplay và tìm assets phù hợp	41
4.1.1 <i>Ý tưởng</i>	41
4.1.2 <i>Plugins hỗ trợ</i>	43
4.1.3 <i>Assets</i>	43
4.2 Xây dựng map (LEVEL)	44
4.3 Xây dựng AI (BOT).....	50
4.3.1 <i>Blueprint cho Scout</i>	50
4.3.2 <i>Blueprint cho Defender</i>	55
4.4 Xây dựng thử thách	56
4.4.1 <i>Thử thách mê cung</i>	56
4.4.2 <i>Cơ chế Oxygen</i>	57
4.4.3 <i>Các nhiệm vụ của game</i>	59
CHƯƠNG 5. KẾT LUẬN	61

5.1 Ưu điểm.....	61
5.2 Nhược điểm.....	61
5.3 Hướng phát triển.....	62
TÀI LIỆU THAM KHẢO.....	63

DANH MỤC HÌNH VẼ

Hình 2.1 Logo của Unreal Engine.....	4
Hình 2.2 Hình render của phiên bản Unreal đầu tiên.....	5
Hình 2.3 Hình render của Unreal Engine 2	5
Hình 2.4 Giao diện của Unreal Engine 3.....	6
Hình 2.5 Các sản phẩm được tạo bởi Unreal Engine 4	6
Hình 2.6 Hình render từ Unreal Engine 5	7
Hình 2.7 Phiên bản render Film mode của Unreal Engine.....	9
Hình 2.8 Ảnh minh họa chuyển động trong Unreal Engine	11
Hình 2.9 Live action với Unreal Engine.....	12
Hình 2.10 Hình ảnh giả lập chuyến bay	13
Hình 2.11 Unreal Engine Editor Interface.....	14
Hình 2.12 Menu Bar	16
Hình 2.13 Main Toolbar	16
Hình 2.14 View mode trong Level Viewport	17
Hình 2.15 Content Browser	18
Hình 2.16 Bottom Toolbar.....	18
Hình 2.17 Outliner/ Hierachy	19
Hình 2.18 Details panel	20
Hình 2.19 Blueprint actor	20
Hình 2.20 Các class trong dự án	21
Hình 2.21 Blueprint Pawn	22
Hình 2.22 Tạo mới một vùng landscape trong Level Viewport	23

Hình 2.23 Foliage Mode – Brushes (bộ cọ).....	24
Hình 2.24 Fracture Mode – Chaos System (Destructable Mode).....	25
Hình 2.25 Logo mixamo.....	26
Hình 2.26 Logo Cargo Kitbash 3D	26
Hình 3.1 Sơ đồ usecase.....	27
Hình 3.2 Sơ đồ ERD.....	31
Hình 3.3 Sơ đồ sequence xử lý của game.....	32
Hình 3.4 Flow hoạt động của AI Scout trong map 1 và map 2	33
Hình 3.5 Flow hoạt động của AI Scout trong map3	33
Hình 3.6 Flow hoạt động của AI Defender	34
Hình 3.7 Gate - Target của player trong map1	35
Hình 3.8 Flow xử lý của Gate	35
Hình 3.9 Flag – Target của player trong map2	36
Hình 3.10 Flow xử lý của Flag	37
Hình 3.11 Crown – Target của player trong map3	38
Hình 3.12 Flow xử lý của Crown.....	38
Hình 3.13 Object Shelter	39
Hình 3.14 Flow Shelter xử lý khi có tương tác.....	40
Hình 3.15 Flow NoisyThing xử lý khi có tương tác	40
Hình 4.1 Kích hoạt Landmass Plugin.....	44
Hình 4.2 Plugin Brush của Landmass Plugin	45
Hình 4.3 Thông số của blueprint brush	46
Hình 4.4 Landmass Brush Details.....	47

Hình 4.5 Giao diện Foliage Mode.....	48
Hình 4.6 BP_Berrier.....	49
Hình 4.7 Các hàm xử lý Damage System.....	50
Hình 4.8 Các hàm tăng/ giảm Oxygen	51
Hình 4.9 Hàm tạo tiếng động.....	51
Hình 4.10 Các Perception của AI.....	51
Hình 4.11 BehaviorTree của Scout map1	52
Hình 4.12 Hàm xử lý Sight của Scout map2 (hình trên) và map1 (hình dưới)	53
Hình 4.13 BehaviorTree của Scout map3.....	54
Hình 4.14 Các thuộc tính cơ bản của người chơi.....	54
Hình 4.15 BehaviorTree của Defender map1	55
Hình 4.16 Hàm xử lý Sight của Defender trong ngũ cảnh đơn (hình trên) và	56
Hình 4.17 Mê cung cho map1	57
Hình 4.18 Nhà trú ẩn giúp phục hồi Oxygen và tránh bị tấn công	58
Hình 4.19 Các slot của DataAsset.....	58
Hình 4.20 Hàm sử dụng SmartObject của AI.....	59
Hình 4.21 Hàm tìm kiếm các slot của SmartObject.....	59
Hình 4.22 Cổng mục tiêu của map1	59
Hình 4.23 Hàm tương tác với mục tiêu map1	60

DANH MỤC BẢNG BIỂU

Bảng 2.1 Các thành phần giao diện.....	15
Bảng 3.1 Đặc tả use case	27

DANH MỤC CÁC CHỮ VIẾT TẮT

AI	Artificial Intelligence
AR	Augmented Reality
BIM	Building Information Modeling
CAD	Computer-Aided Drawing
E2E	End to End
MR	Mixed Reality
PC	Personal Computer
UE	Unreal Engine
VR	Virtual Reality
HMI	Human-Machine Interface
XR	Extended Reality

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Tình hình thực tế cho thấy, thời điểm 2 năm đại dịch xảy ra thì một trong những ngành nghề không những không bị ảnh hưởng mà còn có một bước tiến lớn chính là ngành công nghiệp game.

Thế hệ mới được sinh ra, đi kèm với điều kiện kinh tế và công nghệ đang phát triển giúp cho việc tiếp cận với trò chơi điện tử trong tương lai gần là dễ dàng hơn rất nhiều so với quá khứ.

Cả 2 chúng em đang làm việc tại Glassegg – A Virtuos Studio, một studio chuyên cung cấp các sản phẩm đồ họa cho các công ty game lớn trên toàn cầu. Việc định hướng của công ty hướng đến E2E biến những hiểu biết cơ bản về Game Engine trở thành những điều cần thiết cho chúng em.

So sánh giữa Unity và Unreal Engine thì nhiều nhà làm game đang muôn đưa sản phẩm của mình lên Unreal Engine hơn.

Từ những lý do đó, chúng em quyết định chọn đề tài này và đã được thầy Hồng đồng ý hướng dẫn.

1.2 Mục tiêu thực hiện đề tài và mô tả đề tài

1.2.1 Mục tiêu thực hiện đề tài

Năm bắt được các kiến thức cơ bản của Game Engine và cụ thể là Unreal Engine.

Có khả năng xây dựng một tựa game nhỏ, từ đó rút ra các yêu cầu, các khó khăn có thể gặp phải khi tham gia vào ngành công nghiệp game.

Trang bị được các kiến thức cơ bản cho mục tiêu nghề nghiệp hướng tới E2E giữa khách hàng và công ty.

1.2.2 Mô tả đề tài

- **Thể loại:** Góc nhìn thứ 3 – Battle Royale

- **Cốt truyện/ Kịch bản:** Một nhóm người được chuyển lên một hòn đảo tách biệt với thế giới, thi đấu với nhau để tìm ra người chiến thắng sẽ trở thành vua đảo.
- **Thể thức:** Một người chơi (Single Player) vs AI BOT
- **Cách chơi:** Điều khiển nhân vật tham gia trận đấu theo luật chơi nhất định để giành chiến thắng.
- **Luật chơi:**
 - **(Game 1) Đua tốc độ:** vượt qua mê cung và lính gác để tìm được mục tiêu. TOP 6 người sớm nhất sẽ được qua vòng.
 - **(Game 2) Cướp cờ:** Team-up với các người chơi khác, vượt mê cung và hàng phòng thủ của kẻ thù để đến mục tiêu tích điểm. Đội tích được 10 điểm trước qua vòng.
 - **(Game 3) Người thắng lấy tất cả:** Tất cả người chơi đều có chung 1 mục tiêu. Chiếm lấy nó và giành chiến thắng cuối cùng.
- **Các đối tượng trong game:**
 - MAP: bản đồ mà người chơi sẽ thi đấu.
 - BOT(s): sẽ được lập trình tùy context
 - FLAG: cờ - điểm đến cuối cùng
- **Tính năng của các đối tượng trong game:**

Bối cảnh môi trường nắng nóng khắc nghiệt sẽ làm giảm Oxygen của toàn bộ các đối tượng. Sau một thời gian nhất định, toàn bộ đối tượng sẽ phải đi “tránh nắng” nếu không muốn bị OUT.

 - **Defender (lính gác):** sẽ có patrol route – tuyến tuần tra. Mục đích là canh giữa tuyến đường di chuyển giữa Player và điểm đích. Khi phát hiện mục tiêu sẽ đuổi theo tấn công.
 - **Scout (người tìm kiếm):** (cùng là role của Player): di chuyển tự do trong phạm vi mê cung để thực hiện nhiệm vụ. Đối với AI Scout sẽ có cơ chế tránh khỏi Defender.

1.3 Hướng thực hiện đề tài

- Engine để thực hiện đề tài: **UNREAL ENGINE 5.2.**
- Tìm hiểu về Unreal Engine.
- Tìm hiểu về cách xây dựng MAP
- Tìm hiểu về cách xây dựng AI
- Tìm hiểu về các công nghệ sử dụng trong Unreal Engine
- Tìm assets
- Thực hiện đề tài.
- Báo cáo quá trình thực hiện đề tài.
- Báo cáo kết quả cuối cùng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Unreal Engine (“Unreal Engine 5.2 Documentation,” n.d.)

2.1.1 Khái niệm game engine

Thuở sơ khai, các trò chơi đơn giản được tạo ra từ những dòng mã máy. Nhưng khi trò chơi phát triển với những yêu cầu phức tạp hơn, ta sẽ phải tạo ra các model 3D trong 1 phần mềm, xử lý hoạt ảnh (animation) trong 1 phần mềm khác, tạo ra âm thanh trong 1 phần mềm khác nữa và hàng tá các khía cạnh khác như ánh sáng, vật lý, logic và hơn nữa đều cần những bộ xử lý riêng biệt.

Từ đó, những Game Engine ra đời, đi cùng những công cụ và các thành phần cơ bản, bao trọn các công việc trên để giúp lập trình viên phát triển game một cách dễ dàng.

Bằng cách xử lý đồ họa, vật lý, âm thanh, trí tuệ nhân tạo và đầu vào của người dùng, game engine đơn giản hóa quá trình phát triển game.

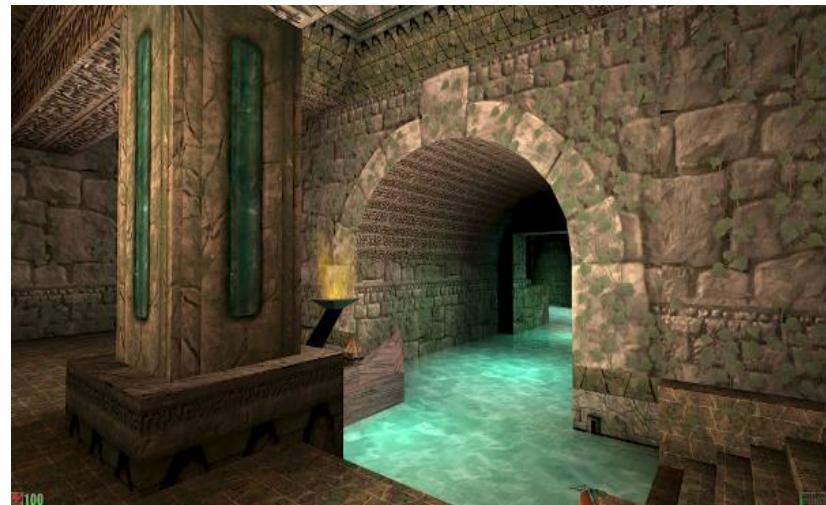
Nó cung cấp các framework và bộ dụng cụ có sẵn, cho phép lập trình viên chỉ cần tập trung vào nội dung và gameplay đặc trưng.



Hình 2.1 Logo của Unreal Engine

2.1.2 Lịch sử các phiên bản của Unreal Engine

Unreal Engine 1: là phiên bản đầu tiên của UE ra mắt đến người chơi vào năm 1998. UE ban đầu là công cụ hoàn toàn dựa vào kết xuất phần mềm, nghĩa là các tính toán đồ họa được xử lý bởi CPU. Theo thời gian, UE đã có thể tận dụng các khả năng được cung cấp bởi GPU trong quá trình xử lý.



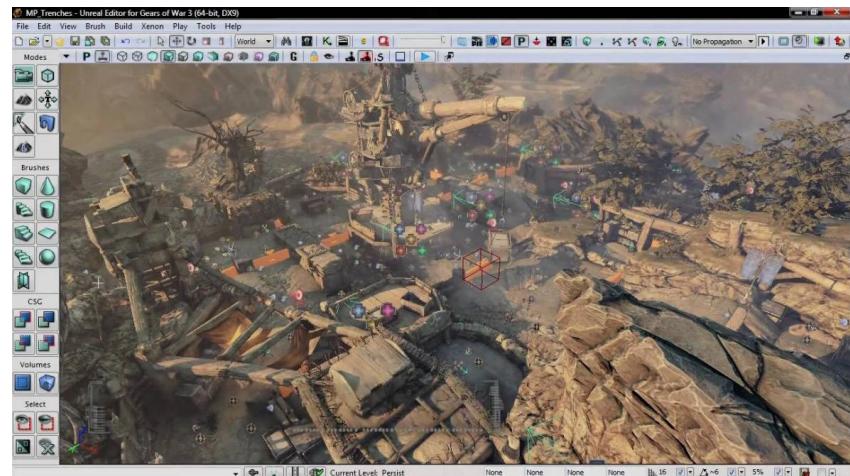
Hình 2.2 Hình render của phiên bản Unreal đầu tiên

Unreal Engine 2: là phiên bản kế nhiệm được ra mắt vào năm 2002. Epic Games kết hợp với quân đội Hoa Kỳ cho ra một tựa game bắn súng góc nhìn thứ nhất để phục vụ cho quá trình tuyển quân nhân. UE2 có những cải tiến đáng kể trong phần nhìn cũng như những cải tiến đối với các bộ tools hỗ trợ, có khả năng chạy các cấp độ chi tiết hơn gần 100 lần so với thế hệ trước. UE2 tích hợp nhiều tính năng, bao gồm công cụ chỉnh sửa điện ảnh, hệ thống hạt, trình cắm xuất cho 3D.



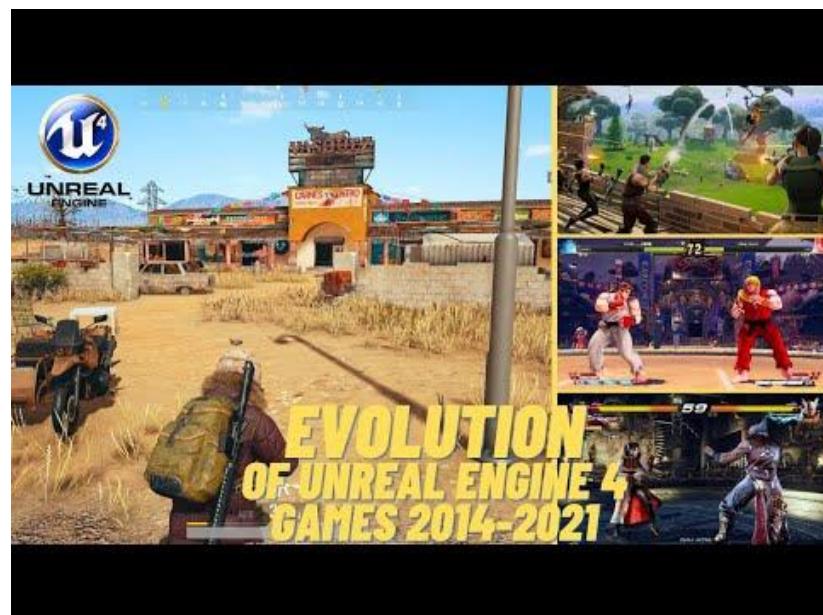
Hình 2.3 Hình render của Unreal Engine 2

Unreal Engine 3: là phiên bản nâng cấp tiếp theo của UE2 được trình làng vào tháng 7 năm 2004. UE3 có tính năng mới như hiển thị cho các lập trình viên về thiết kế hướng đối tượng, cũng như các tính năng về render, hệ thống vật lý, hệ thống âm thanh và các công cụ hỗ trợ - tất cả đều rõ ràng và mạnh mẽ hơn đáng kể so với phiên bản trước.



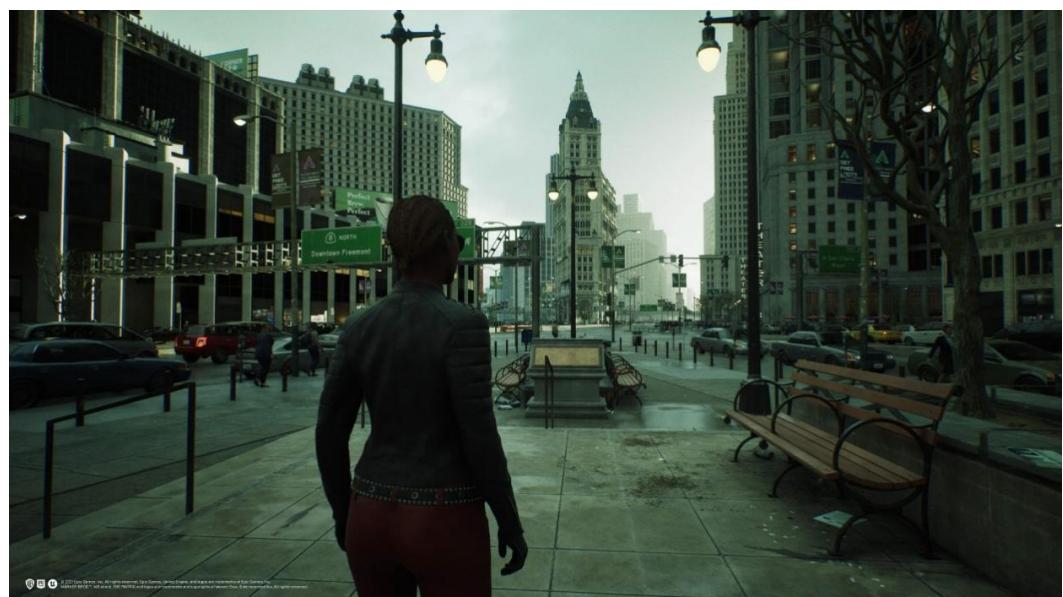
Hình 2.4 Giao diện của Unreal Engine 3

Unreal Engine 4: UE4 được công bố cho những người tham dự tại hội nghị các nhà phát triển trò chơi vào năm 2012. Phiên bản này bổ sung các tính năng chiếu sáng toàn cầu theo thời gian thực và sử dụng theo dõi hình nón voxel.



Hình 2.5 Các sản phẩm được tạo bởi Unreal Engine 4

Unreal Engine 5: UE5 được ra mắt vào năm 2020 và cũng là phiên bản được sử dụng phổ biến nhất cho đến hiện tại. Một trong những tính năng chính của UE5 là Nanite – một công cụ cho phép nhập data nguồn ảnh có độ chi tiết cao vào trò chơi. Công cụ này giúp các nhà phát triển dễ dàng tạo ra thế giới trò chơi một cách chi tiết mà không cần phải dành nhiều thời gian cho việc tạo ra các chi tiết mới.



Hình 2.6 Hình render từ Unreal Engine 5

Unreal Engine có thể tích hợp trong rất nhiều ngành công nghiệp tiêu biểu hiện nay:

1. Games.
2. Làm phim và truyền hình.
3. Diễn họa kiến trúc.
4. Tự động hóa và chuyển động.
5. Broadcast và các sự kiện trực tuyến.
6. Giả lập.

2.1.3 Các lĩnh vực trong UE5

2.1.3.1 Games

- Tổng quan

- **Mọi dự án, mọi kích cỡ:** UE cho phép người dùng thoát khỏi rào cản về mọi kích cỡ của một dự án với các công cụ và workflows trong phạm vi bạn có thể kiểm soát. Bất kể team bạn có 5 người hay 500 người, các modules hệ thống được dựng sẵn của UE, các plugin có thể tùy chỉnh và tích hợp kiểm soát sẽ giúp bạn đáp ứng được từng nhu cầu đặc biệt của dự án đó.
- **Xây dựng games trên nhiều nền tảng chỉ với một đường dẫn nội dung thống nhất:** UE cho phép người dùng kết nối nhiều người chơi với cùng một trải nghiệm chất lượng cao thông qua tất cả các nền tảng chính, từ điện thoại di động đến console, PC hoặc thậm chí XR (AR, VR và MR).
- **Made for creators, by creators:** là khẩu hiệu của Epic, họ tự sử dụng công cụ này để tạo ra các nội dung đỉnh cao và đứng đầu bảng xếp hạng và họ muốn chia sẻ những điều hay ho lại cho người dùng dưới dạng các công cụ mạnh mẽ và tối ưu nhất. Trong vòng hai tập kỉ qua, UE đã trở thành engine được tin tưởng và sử dụng nhiều nhất trên thế giới.
- **Cộng đồng lớn mạnh và chất lượng:** Các chuyên gia liên tục chọn UE để thúc đẩy nhanh giới hạn về trải nghiệm hình ảnh. Không cần biết người dùng là ai chỉ cần đang tham gia một cộng đồng tích cực như UE bạn sẽ cảm nhận có một kết quả xuất sắc.
- **Các games tiêu biểu sử dụng Unreal Engine:** Stray, Fortnite, It takes two, ...

- Các đặc trưng chính

- Dẫn đầu trong lĩnh vực đồ họa
- Multiplayer framework

- C++ source code
- Blueprints
- Metahumans

2.1.3.2 Làm phim và truyền hình



Hình 2.7 Phiên bản render Film mode của Unreal Engine

- **Tổng quan**
 - **Tự do thỏa sức sáng tạo từ đầu đến cuối**
 - **Phản hồi ngay lập tức:** UE cho phép người dùng Render thời gian thực cho phép người dùng có thể nhận phản hồi ngay lập tức, điều này giúp các team có thể test ý tưởng và ra quyết định ngay lập tức.
 - **Same asset, longer life:** UE cho phép người dùng Sử dụng cùng một asset từ ý tưởng đầu tiên đến khung hình cuối cùng, loại bỏ những thừa thãi trong suốt giai đoạn sản xuất. Sử dụng lại các asset cho trải nghiệm thị trường, tương tác và trải nghiệm.

- **Được ngành phim phê duyệt:** Epic Games đã và đang làm việc với nhiều nhà làm phim và các studios, để phát triển các tính năng được sử dụng trong các dự án tham vọng nhất bởi ngành này. Vẫn với khâu hiệu “for creators, by creators”.
- **Những con người “kỹ thuật số” sẽ tiến đến gần như thật:** MetaHuman là một framework hoàn chỉnh được phát triển bởi đội ngũ UE mà cho phép người dùng sức mạnh để tạo ra nhân vật cũng như animation cho các nhân vật “Kỹ thuật số” một cách chân thật nhất trong các dự án UE.

- **Các đặc trưng chính**

- Virtual Production
- Nonlinear Editing
- Final-pixel Output
- Pipeline

2.1.3.3 Kiến trúc

- **Tổng quan**

- **Perfect pixels – at scale:** Chuẩn chỉnh, sắc nét đến từng pixel.
- **Get in, get into it:** UE giúp cho việc mang dữ liệu có tính trung thực cao từ một loạt các ứng dụng 3D, CAD, BIM trở nên đơn giản hơn, vì vậy UE có thể cho bạn trải nghiệm về các hình ảnh sống động ngay lập tức. Với sự hỗ trợ dành cho các thiết kế có tính đồ sộ, quy mô lớn và với chủ nghĩa hình ảnh chân thực, người dùng có thể dễ dàng biến những hình ảnh táo bạo nhất thành hiện thực.
- **All together now:** UE tạo ra một cầu nối giữa các bên liên quan để giải quyết các vấn đề trước khi họ bắt đầu thực hiện một dự án nào đó, bằng cách sử dụng các “remote collaboration tools”, hoặc có thể khám phá một thành phố thông qua VR/AR.
- **Tái định nghĩa Big data:** Với UE, dữ liệu có thể có một cuộc sống mới, người dùng có thể tạo ra thế giới mở, trackers máy bay, ...

- **Twinmotion:** Chuyển đổi dữ liệu BIM, CAD và những trải nghiệm VR thành những hiệu ứng hình ảnh đẹp tuyệt vời chỉ với một vài cú click.

- **Các đặc trưng chính**

- Real time
- Góc nhìn sắc nét
- Kết hợp và chỉnh phục
- Một asset, có thể di chuyển mọi nơi

Tự động hóa và chuyển động



Hình 2.8 Ảnh minh họa chuyển động trong Unreal Engine

- **Tổng quan**

- **Design and delight:** UE có thể giúp người dùng rút ngắn quy trình thiết kế, cho các trải nghiệm nhập vai, có thể đưa cho khách hàng của họ xem mọi tùy chọn trong danh mục, cho họ lái thử xe thông qua công nghệ, hoặc thậm chí có thể bắt chước các thiết kế theo một cách chân thật nhất. Tất cả được gói gọn trong Unreal Engine.
- **Digital cockpit development (Human-Machine Interface):** Với UE người dùng có thể thiết kế các loại HMI (Human-Machine Interface), thông tin giải trí hoặc trải nghiệm buồng lái kỹ thuật số đưa họ đến nơi họ muốn theo một phong cách riêng.

- **One source of truth:** Lưu trữ tất cả các model reference của người dùng trong cùng một nơi, thúc đẩy tiến trình làm việc nhanh hơn, mọi thứ chỉ cách vài giây.
- **Automotive field guide:** Công nghệ real-time đang thúc đẩy ngành công nghiệp ô tô từ thiết kế ban đầu đến sàn đại lý. UE đang thực sự hỗ trợ sự chuyển đổi doanh nghiệp.

- **Các đặc trưng chính**

- Push the limits
- Accelerated, automated
- Multi-user collaboration
- Your data, your choice
- HMI helpers

2.1.3.4 Broadcast và các sự kiện trực tuyến



Hình 2.9 Live action với Unreal Engine

- **Tổng quan**

- **Perfect pixels:** UE giúp người dùng có thể pha trộn những hành động thực tế với đồ họa máy tính ở bất cứ nơi nào.
- **The right fit**

- **Strong & steady:** Với UE, họ tự tin là một nền tảng có thể đoán được là sẽ không bao giờ sụp đổ. Khi người dùng cần, UE luôn có mặt.
- **Các đặc trưng chính**
 - Integration
 - Real-time compositing
 - Multi-display rendering
 - Particles, effects & physics

2.1.3.5 Giả lập



Hình 2.10 Hình ảnh giả lập chuyến bay

- **Tổng quan**
 - **Critical fidelity:** Truyền thông tin thông qua những cú click chuột ngay lập tức, cho dù người dùng đang training con người

hay training máy, UE sẽ phân tích thiết bị hoặc đánh giá các lựa chọn trong tương lai. UE cung cấp cho bạn hình ảnh có độ trung thực cao và khả năng kiểm soát mà bạn yêu cầu, miễn phí.

- **Full control:** Với UE, người dùng có thể build, scale, và kiểm soát mọi khía cạnh thiết kế của họ chỉ với việc kéo thả.
- **Liberate your assets:** UE cho phép việc import và reimport asset nhiều lần nếu người dùng cần tùy chỉnh độ mô phỏng. Người dùng cũng có thể sử dụng nhanh chóng các asset miễn phí thông qua Unreal Engine Marketplace.

- Các đặc trưng chính

- Efficient data processing
- Terrain databases
- Multi-platform deployment
- A range of Realism
- XR-Ready

2.1.4 Các kiến thức nền tảng về Unreal Editor

- Unreal Editor Interface

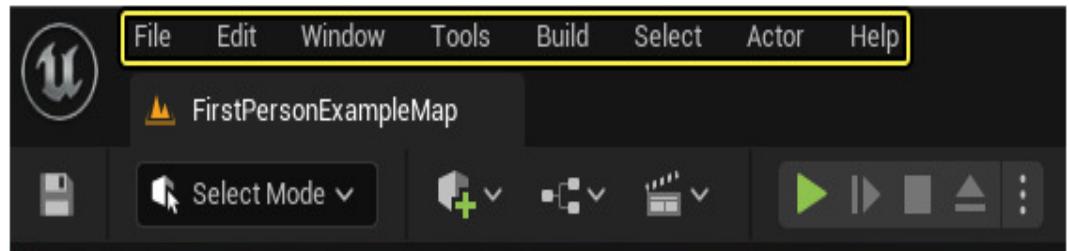


Hình 2.11 Unreal Engine Editor Interface

Bảng 2.1 Các thành phần giao diện

No.	Tên	Mô tả
1	Menu Bar	Sử dụng menu bar để truy cập những command đặc biệt cũng như những chức năng của editor.
2	Main Toolbar	Chứa các phím tắt cho một số tools và editors thông dụng nhất trong UE, cũng như các phím tắt để vào chế độ “Play” (chạy trò chơi bên trong Unreal Editor) và để triển khai dự án của người dùng trên các nền tảng khác.
3	Level Viewport	Hiển thị nội dung của Levels.
4	Content Drawer	Nơi chứa tất cả các asset mà người dùng sử dụng trong dự án.
5	Bottom Toolbar	Chứa các phím tắt đến chức năng Command Console, Output Log và Derived Data. Cũng như hiển thị trạng thái của Source Control.
6	Outliner	Hiển thị Hierarchi dạng cây của tất cả nội dung trong Level.
7	Details Panel	Xuất hiện khi bạn chọn bất kỳ 1 Actor. Hiển thị các thuộc tính khác nhau cho Actor đó. Bảng điều khiển này hiển thị các cài đặt khác nhau tùy thuộc vào những gì người dùng chọn trong Level Viewport.

- **Menu Bar**



Hình 2.12 Menu Bar

Tích hợp các chức năng đặc biệt của Unreal Engine

- **Main Toolbar**



Hình 2.13 Main Toolbar

1 – Nút save

2 – Chọn Mode

3 – Các phím tắt cho phần tạo content

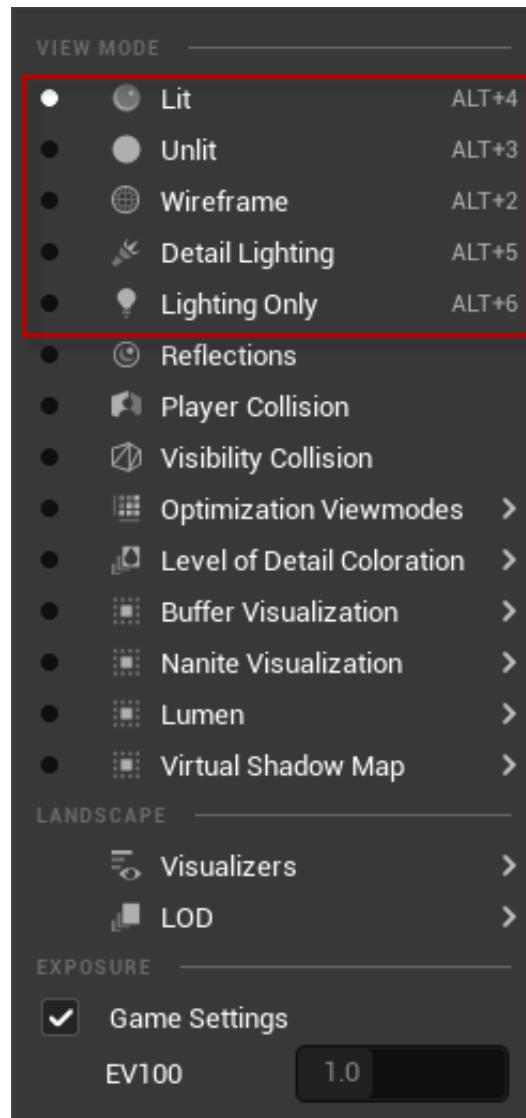
4 – Điều khiển Play Mode

5 – Platforms Menu

6 – Settings

- **Level Viewport**

- Level Viewport hiển thị nội dung của Level đang được mở.
- Level Viewport nói chung có thể hiển thị nội dung của Level theo hai cách **Perspective – 3D/ Orthographic – 2D**



Hình 2.14 View mode trong Level Viewport

Lit: Chế độ render realtime với chất liệu và ánh sáng.

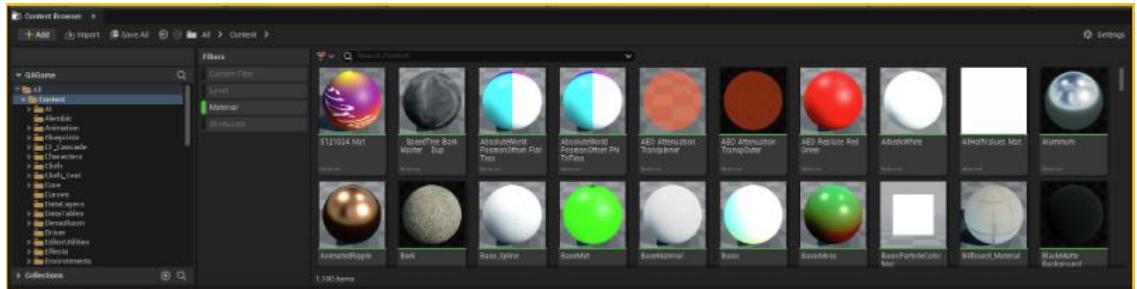
Unlit: Chế độ render không có ánh sáng, chỉ hiển thị màu của chất liệu cơ bản.

Wireframe: Chế độ chỉ show lưới.

Detail Lighting: Chế độ render chi tiết ánh sáng thông qua các map (normal, roughness, emissive).

Lighting Only: Chế độ render chỉ hiển thị ánh sáng phản xạ lên vật thể.

- Content Drawer & Content Browser



Hình 2.15 Content Browser

- Content Browser đóng vai trò như là một cửa sổ file explorer, chứa tất cả content mà người dùng sử dụng trong dự án bao gồm: assets, blueprints, những files khác có liên quan đến nhau.
- Content Drawer là một biến thể đặc biệt của Content Browser.

- Bottom Toolbar



Hình 2.16 Bottom Toolbar

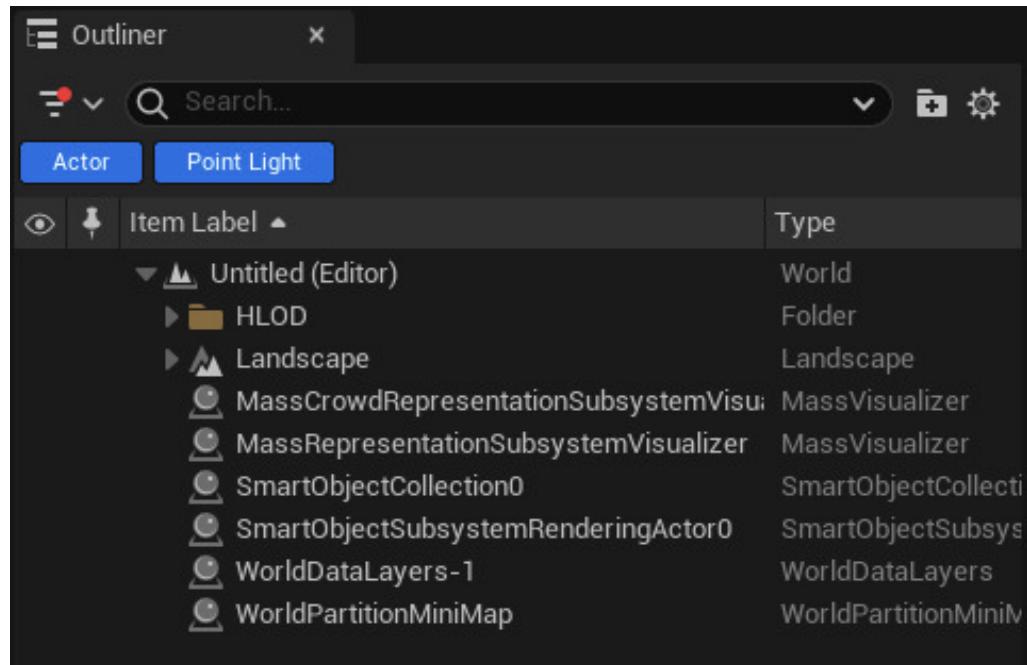
1 – Output Log: khi người dùng ánh vào nút này, một popup window sẽ được hiển thị và show ra các thông số cũng như các lệnh hoặc lỗi có thể xảy ra trong quá trình người sử dụng thực hiện dự án

2 – Command Console: khi ánh vào cũng sẽ hiển thị ra một popup window đồng thời cũng là một code editor là nơi người để các nhà phát triển có thể code dựa trên ý tưởng của mình.

3 – Derived Data: là nơi lưu bộ nhớ tạm cũng như cache của dự án mà người dùng đang làm trên đó.

4 – Source Control Status: chỉ trạng thái của Source Control.

- **Outliner**



Hình 2.17 Outliner/ Hierachy

Hiển thị nội dung của hierarchy theo dạng cây của toàn bộ nội dung trong Level đang được mở, người dùng cũng có thể thao tác trên Outliner để điều khiển được các Actors trên màn hình làm việc cũng như việc thêm, xóa, sửa các nội dung bên trong assets hoặc folders.

- **Details Panel**

Khi người dùng chọn một Actor trong Level Viewport, Details Panel sẽ hiển thị tất cả các settings cũng như thuộc tính của Actor được chọn.



Hình 2.18 Details panel

2.1.5 Các thuật ngữ trong Unreal Engine

Actor là một thực thể trong game, có thể là một nhân vật, một vật phẩm hoặc cảnh quan. Mỗi actor đều có một vị trí trong không gian 3D và có các Component (Mesh, Collision, Light, Sound, ...) để mô tả hình dạng, ánh sáng, âm thanh, chuyển động và hành vi của thực thể đó.

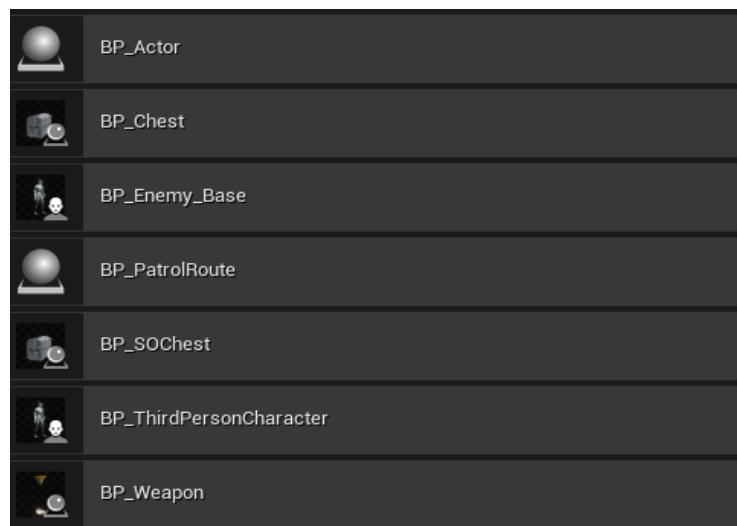


Hình 2.19 Blueprint actor

Blueprint là công cụ cho phép người dùng thiết kế và lập trình trò chơi một cách dễ dàng bằng cách sử dụng một giao diện trực quan hay vì lập trình code. Người dùng tạo ra gameplay và tương tác trong trò chơi bằng cách kết nối các nút (nodes) để tạo ra những hành động hoặc tương tác mà mình mong muốn.

- **Nodes:** Là nut được sử dụng để thực hiện những hành động khác nhau. Nodes có thể bao gồm các hàm, biến, sự kiện, điều kiện, ...
- **Wires:** Kết nối các nodes với nhau, cho phép truyền dữ liệu và giá trị giữa các nodes.
- **Variables:** Lưu trữ giá trị và trạng thái của các thành phần trong game.
- **Functions:** Là các nodes đại diện cho chức năng được thực hiện trong game. Có thể được tái sử dụng nhiều lần.
- **Events:** Các nodes đại diện cho các sự kiện trong game, chẳng hạn như nháy phím hoặc va chạm vật thể.
- **Macros:** Các khối tổng hợp nhiều nodes để tạo thành một node lớn hơn, giúp giảm độ phức tạp của Blueprint và tối ưu khả năng tái sử dụng..
- **Comments:** Là chú thích và ghi chú được thêm vào blueprint để giải thích logic hoặc chức năng của node hoặc function nào đó.

Lớp (Class) định nghĩa các hành vi và thuộc tính của một Actor hoặc một đối tượng cụ thể trong UE. **Lớp** được phân cấp, có nghĩa là một lớp kế thừa thông tin từ lớp cha của nó (có thể gọi là một lớp dẫn xuất hoặc một lớp con) và chuyển thông tin xuống cho các lớp con của nó. Lớp có thể được tạo ra bằng code C++ hoặc bằng Blueprint.



Hình 2.20 Các class trong dự án

Casting là một hành động gán một Actor vào một lớp cụ thể. Việc này có thể thành công hoặc thất bại, nếu thành công thì người dùng có thể truy cập chức năng dành riêng cho lớp bên trên của Actor.

Pawn là một lớp con của Actor và đóng vai trò là avatar hoặc nhân vật trong game. Pawn có thể được điều khiển bởi người chơi hoặc AI của trò chơi, với tư cách không phải là người chơi (NPC).



Hình 2.21 Blueprint Pawn

Master material là chất liệu được tạo ra để tối ưu hóa quy trình phát triển và quản lý tài nguyên game, có thể dùng nó cho nhiều đối tượng khác nhau thay vì phải tạo ra một material riêng cho mỗi đối tượng. **Master material** chứa các tham số và chức năng được phát triển dựa vào mục đích sử dụng, đặc điểm của texture, shader, ...

Material Instance là material phái sinh của master material, có ảnh hưởng real-time đến các đối tượng sử dụng nó.

Static meshes là các đối tượng không thể di chuyển hoặc thay đổi trong môi trường game. Static meshes có thể là tường, sàn, đồ vật, cây cối. Chúng giữ nguyên hình dáng, kích thước, material không di chuyển và thay đổi khi chạy game. Static meshes giúp tối ưu việc giảm cho hệ thống, tăng tốc độ hiển thị và hiệu suất của trò chơi.

Skeletal Meshes là một loại mesh có khả năng chuyển động trong trò chơi. Nó được kết hợp với một skeleton để tạo nên hình dáng của đối tượng và các chuyển động của chúng. Skeletal Mesh có thể được tạo bằng các phần mềm 3D Modeling hoặc được tạo trực tiếp trong UE.

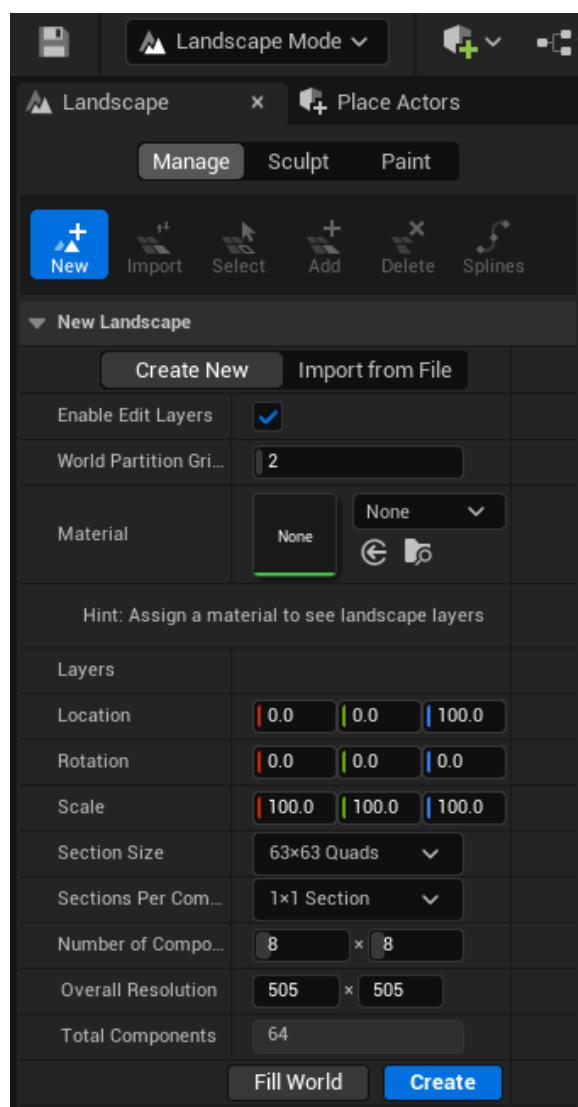
Skeleton là hệ thống xương được sử dụng để quản lý hành động của đối tượng trong game. Nó bao gồm các xương liên kết với nhau và được sắp xếp theo cấu trúc phân cấp (Hierarchy). Skeleton được sử dụng để điều khiển các skeletal mesh và tạo ra các animation cho đối tượng trong game.

2.1.6 Các mode thường được sử dụng trong Unreal Editor

Selection Mode

Với Selection Mode, UE cho phép người dùng có thể sử dụng cũng như custom các chức năng sẵn có để có thể tạo một khung cảnh hoặc một bản đồ (Level) hoàn thiện, có rất nhiều tùy chọn cho người dùng tùy vào mục đích sử dụng cho dự án.

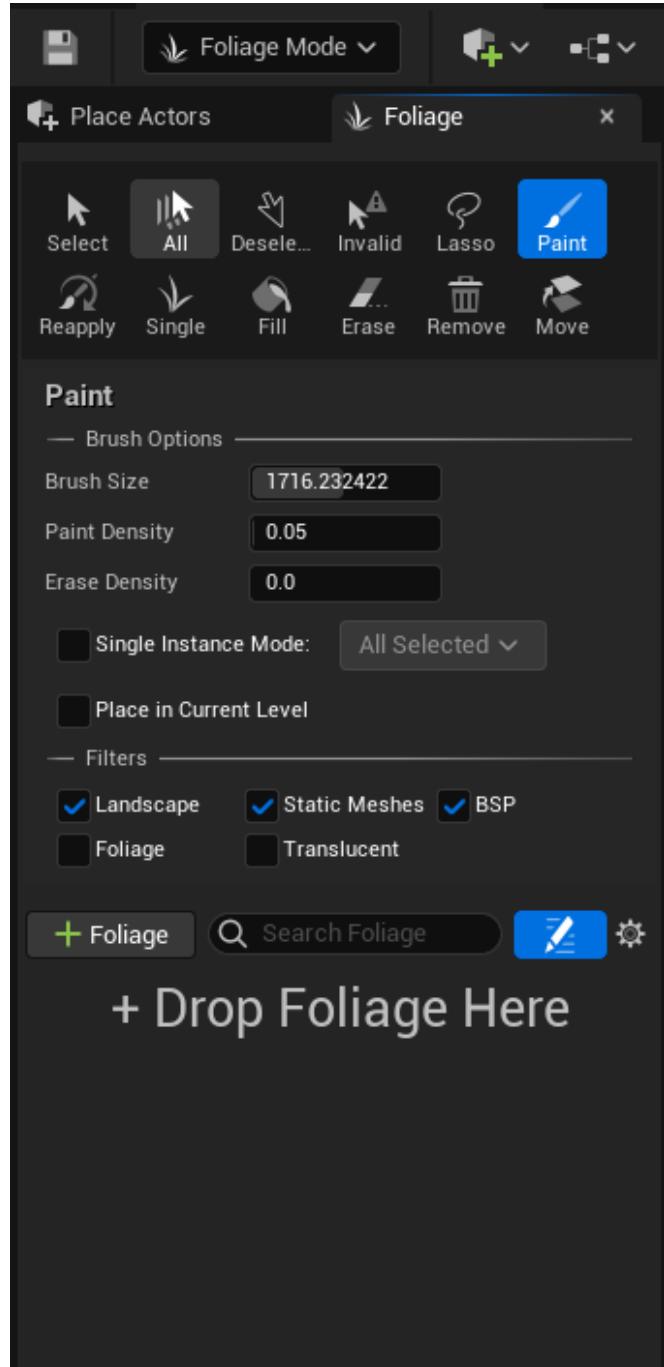
Landscape Mode



Hình 2.22 Tạo mới một vùng landscape trong Level Viewport

Với Landscape Mode, UE cho phép người dùng có thể tạo ra một thế giới mở trong game bằng cách sử dụng các chức năng như Paint, Sculpt cũng như có thể blend được nhiều loại chất liệu vào vùng đất do mình tạo ra. (“Creating Landscapes,” n.d.)

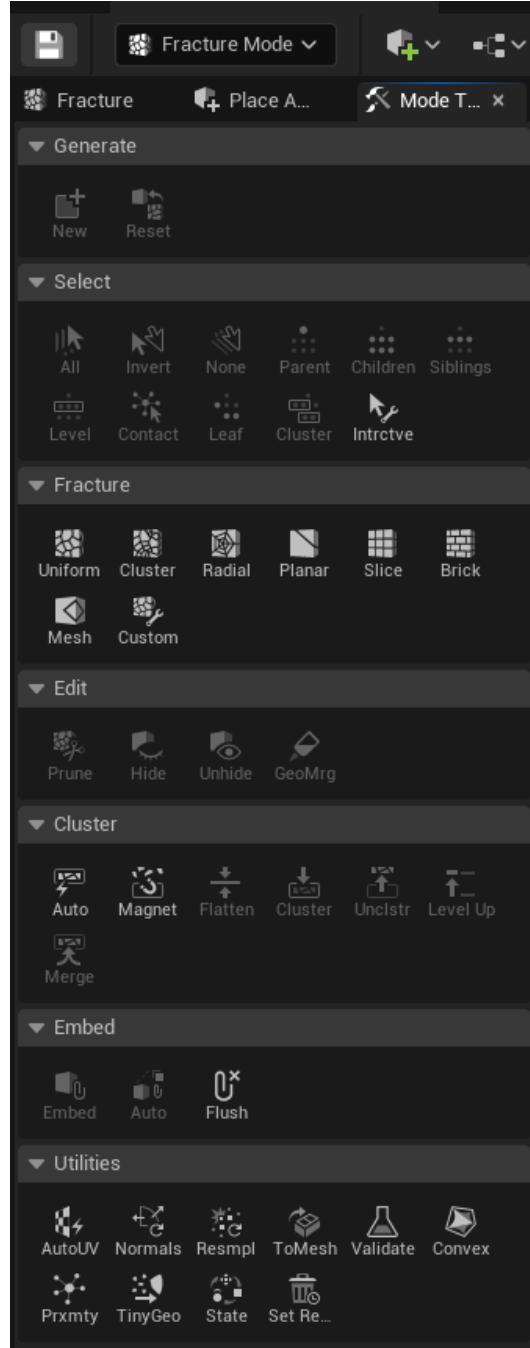
Foliage Mode



Hình 2.23 Foliage Mode – Brushes (bộ cọ)

Với Foliage Mode, UE cho phép người dùng làm cho thế giới của mình tạo ra được sinh động hơn bằng cách thêm cây cỏ, gạch, đá, các vật thể nhỏ,... một cách tùy ý và theo những tùy chọn mà người dùng có thể điều chỉnh. (“Foliage Mode,” n.d.)

Fracture Mode



Hình 2.24 Fracture Mode – Chaos System (Destructable Mode)

Với Fracture Mode, UE cho phép người dùng tạo ra các hiệu ứng đỗ vỡ kết hợp với plugin Chaos được phát triển bởi đội ngũ UE làm cho trò chơi của người dùng trở nên chân thật hơn với mỗi lần va chạm hoặc tác động vật lý lên một vật thể nào đó. (“Fracture Mode Selection Tools User Guide,” n.d.)

2.2 Mixamo



Hình 2.25 Logo mixamo

Mixamo là một phần mềm của Adobe hỗ trợ người dùng theo nhiều phương diện từ cung cấp nhân vật cho đến hành động, animations dựa trên nhân vật mình đã chọn. Chúng ta có thể áp dụng Mixamo để có được những asset ưng ý nhất về phần công dụng lẫn phần nhìn.

Mixamo có thể hỗ trợ tốt cho các phần mềm làm game như Unity cũng như Unreal Engine. (“Mixamo,” n.d.)

2.3 Cargo Kitbash 3D

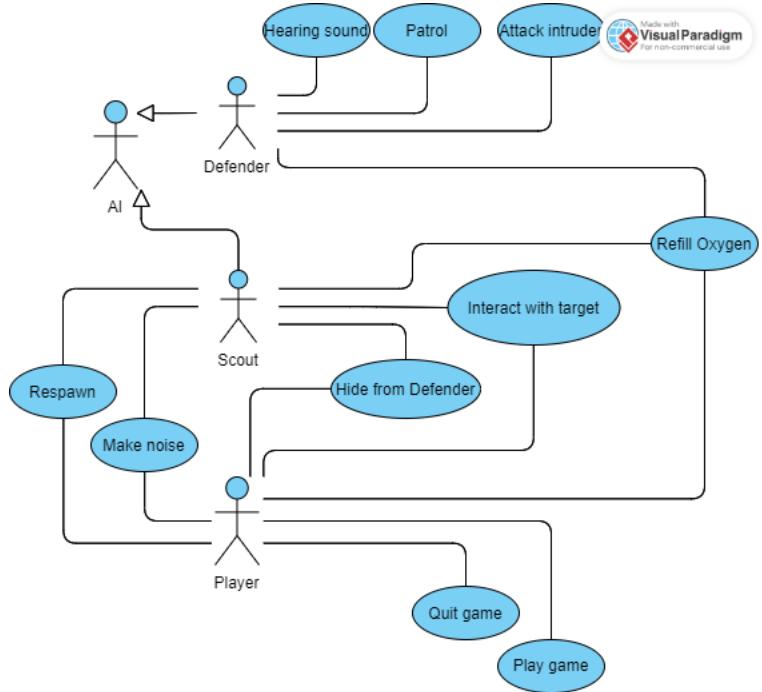


Hình 2.26 Logo Cargo Kitbash 3D

Cargo Kitbash 3D là một phần mềm hỗ trợ các nhà làm game về các assets, materials tuyệt đẹp. Các project điển hình sử dụng KitBash như: Cyberpunk 2077, Medieval, ... (“Cargo by KitBash3D | Free 3D Asset Manager for Blender, Unreal Engine 5, Cinema4D, 3ds Max, Maya, and More,” n.d.)

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ

3.1 Sơ đồ Use-case của Game



Hình 3.1 Sơ đồ usecase

Tựa game sẽ gồm 1 actor chính là Player cùng 2 actor hệ thống là các AI có nhiệm vụ cạnh tranh với Player.

Bảng 3.1 Đặc tả use case

Use case name	Play game
Description	Là người dùng, tôi muốn bắt đầu chơi tựa game
Actor(s)	Player
Trigger	Người dùng muốn chơi game
Pre-condition(s)	Người dùng đã cài đặt ứng dụng
Flow	<ul style="list-style-type: none"> Người dùng mở ứng dụng UI hiển thị các tùy chọn Người dùng chọn “Play” UI hiển thị cốt truyện và hướng dẫn Hệ thống chuyển người dùng đến map1 của game

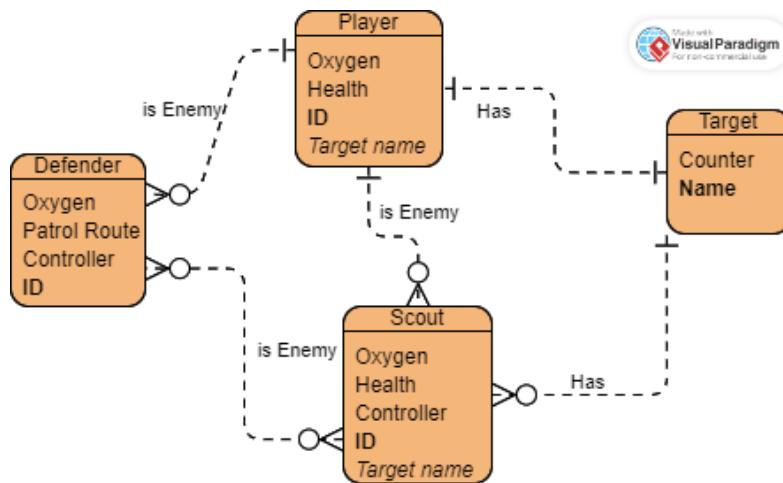
Use case name	Quit game
Description	Là người dùng, tôi muốn dừng không chơi nữa
Actor(s)	Player
Trigger	Người dùng thoát tựa game
Pre-condition(s)	Người dùng đang chơi game
Flow	<ul style="list-style-type: none"> • Người dùng mở menu tùy chọn • UI hiển thị các tùy chọn • Người dùng chọn “Quit” • Trò chơi đóng ứng dụng
Use case name	Patrol
Descripton	Defender tuần tra bảo vệ mục tiêu
Actor(s)	Defender
Trigger	Defender không có mục tiêu tấn công, xác định được tuyến tuần tra và thực hiện đi tuần
Pre-condition(s)	<p>Defender đã được spawn trên map</p> <p>Defender đang không overheated</p>
Flow	<ul style="list-style-type: none"> • Defender không có mục tiêu • Defender lấy vị trí của tuyến tuần tra • Defender thực hiện di chuyển giữa các node trên tuyến tuần tra
Use case name	Attack Intruder
Descripton	Defender tấn công mục tiêu xâm chiếm
Actor(s)	Defender
Trigger	Có kẻ địch đứng trong tầm nhìn của Defender
Pre-condition(s)	<p>Defender đang không có mục tiêu khác</p> <p>Defender đang không overheated</p>
Flow	<p>Mục tiêu bị Sight Perception của Defender phát hiện</p> <p>Defender xác định mục tiêu để tấn công</p>

	Defender di chuyển đến mục tiêu Defender thực hiện tấn công tiêu diệt
Use case name	Hearing sound
Descripton	Defender nghe âm thanh và đi điều tra
Actor(s)	Defender
Trigger	Có âm thanh phát ra trong tầm nghe của Defender
Pre-condition(s)	Defender đang không có mục tiêu khác Defender đang không overheated
Flow	<ul style="list-style-type: none"> • Hearing Perception của Defender phát hiện vị trí phát ra âm thanh • Defender di chuyển đến vị trí phát ra âm thanh • Defender thực hiện điều tra
Use case name	Interact with Target
Descripton	Player/ Scout thực hiện nhiệm vụ để qua màn
Actor(s)	Player, Scout
Trigger	Actor tìm thấy mục tiêu, tương tác với mục tiêu để qua màn
Pre-condition(s)	Actor đã được spawn trên map Actor đã tìm được mục tiêu
Flow	<ul style="list-style-type: none"> • Actor thực hiện tương tác với mục tiêu • Mục tiêu thực hiện hàm tương tác • Hệ thống kiểm tra thỏa mãn điều kiện qua màn chưa • Hệ thống chuyển đến kết quả thắng/ thua tùy trường hợp
Use case name	Hide from Defender
Descripton	Player/ Scout ẩn nấp để tránh bị phát hiện tiêu diệt
Actor(s)	Player, Scout
Trigger	Có Defender trong tầm nhìn của Actors
Pre-condition(s)	Actor đang không overheated
Flow	Sight Perception của actors phát hiện Defender

	<ul style="list-style-type: none"> Actor xác định vị trí gần nhất khuất tầm nhìn của Defender Actor di chuyển đến vị trí an toàn Actor chờ Defender đi khuất Actor tiếp tục nhiệm vụ
Use case name	Make noise
Descripton	Player/ Scout để lộ vị trí khi đạp phái bẫy
Actor(s)	Player, Scout
Trigger	Actor tương tác với bẫy trên map
Pre-condition(s)	
Flow	<ul style="list-style-type: none"> Actor đạp trúng bẫy trong quá trình làm nhiệm vụ Actor gây tiếng động tại vị trí của bẫy
Use case name	Respawn
Descripton	Player/ Scout hồi sinh sau khi bị tiêu diệt/ tương tác với mục tiêu
Actor(s)	Player, Scout
Trigger	Actor bị tiêu diệt/ tương tác với mục tiêu
Pre-condition(s)	
Flow	<ul style="list-style-type: none"> Actor bị tiêu diệt Hệ thống tìm các điểm respawn ngẫu nhiên đặt trên map Hệ thống spawn lại actor Hệ thống gán các tham số cần thiết cho actor Actor tiếp tục nhiệm vụ
Use case name	Refill oxygen
Descripton	Player/ Scout/ Defender cần phải hồi oxygen
Actor(s)	Player, Scout, Defender
Trigger	Nguồn oxygen của actor xuống mức overheat
Pre-condition(s)	

Flow	<ul style="list-style-type: none"> Actor bị overheat Actor dừng mọi hoạt động Actor tìm các điểm hòi oxygen trên map Actor di chuyển đến vị trí mục tiêu Actor thực hiện tương tác để hòi oxygen
------	---

3.2 Sơ đồ ERD



Hình 3.2 Sơ đồ ERD

Trò chơi có 4 thực thể chính:

- Player
- Scout
- Defender
- Target

Trong đó, Player chính là người dùng, với vai trò tương tự như các Scout, trong mỗi vòng chơi, Player sẽ có một mục tiêu nhất định trên map cần phải tương tác để thỏa mãn điều kiện qua màn.

Mỗi target sẽ có 1 hàm xử lý tương tác khác nhau và chỉ chấp nhận tương tác của Player và Scout

Với Player sẽ có 2 loại kẻ thù:

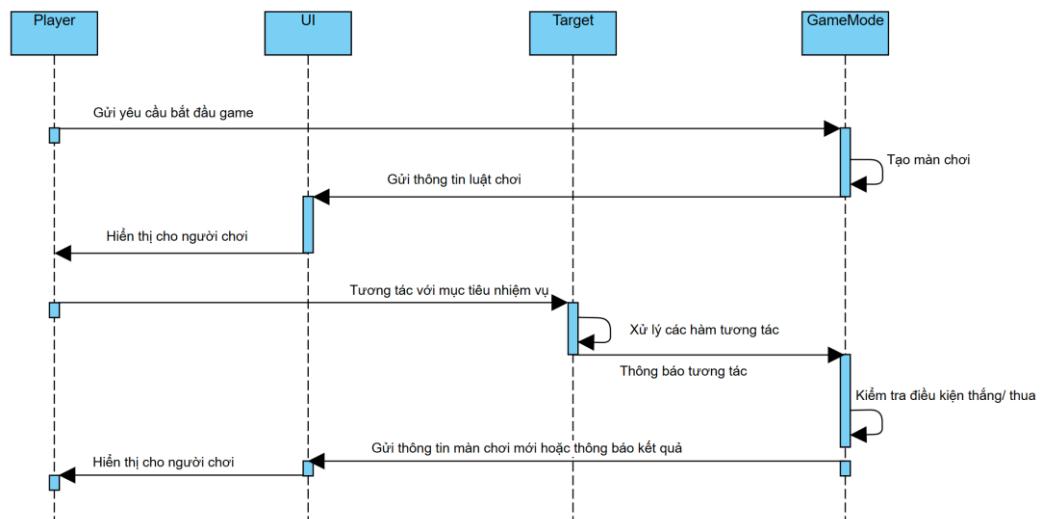
- Scout là các đối thủ cạnh tranh, mặc dù không trực tiếp gây tổn hại đến Player nhưng là các thực thể ảnh hưởng đến điều kiện qua màn của

Player khi cả hai có cùng mục tiêu và chỉ khi player thỏa mãn mục tiêu trước thì mới có thể qua màn

- Defender là các chướng ngại vật, các Defender không ảnh hưởng đến điều kiện qua màn vì chúng không thể tương tác với mục tiêu, nhưng lại cản trở player thực hiện các nhiệm vụ để qua màn.

Do Player và Scout có cùng một vai trò, nên đối với Scout, các Defender cũng là một chướng ngại và Player cũng là đối thủ cạnh tranh với Scout.

3.3 Sơ đồ Sequence xử lý của game



Hình 3.3 Sơ đồ sequence xử lý của game

Quy trình xử lý của game khá đơn giản, gồm các tác nhân chính:

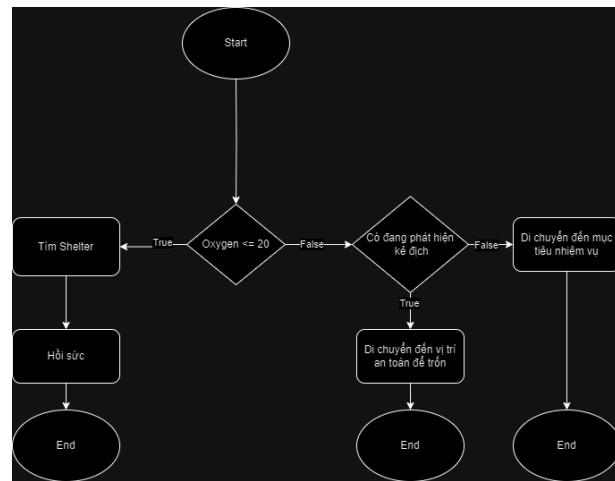
- GameMode: được xem là hệ thống xử lý của game, nơi có thể cấu hình các actor và cách xử lý các điều kiện xảy ra trong game.
- UI: phần trung gian giao tiếp giữa game và người dùng.
- Target: mục tiêu nhiệm vụ, ảnh hưởng đến kết quả trò chơi.

Người chơi sẽ tương tác thông qua UI và các Object trong game, GameMode có nhiệm vụ nhận thông tin từ hành động của người dùng trên UI để hiển thị những phần được yêu cầu. Khi người dùng hoàn thành nhiệm vụ, GameMode sẽ nhận thông tin từ các mục tiêu nhiệm vụ và xác định xem người chơi đã thắng hay thua, sau đó quyết định cho người chơi chơi lại hoặc chơi tiếp hoặc kết thúc game.

3.4 Các sơ đồ flowchart của các actor trong game

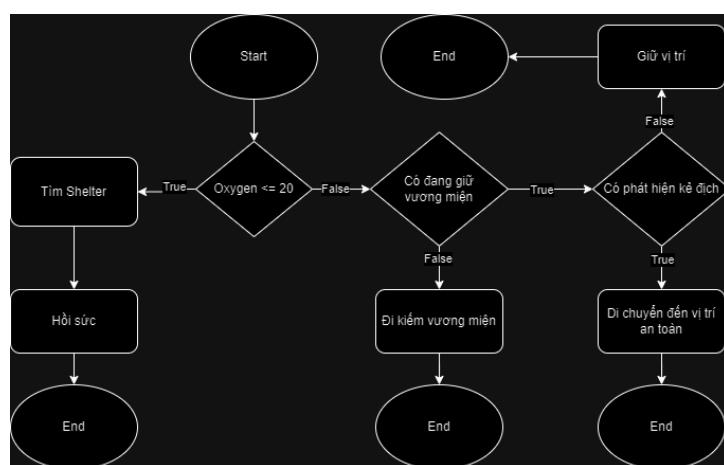
Các flow xử lý của AI sẽ được cập nhật theo từng frame để đảm bảo chính xác về mặt logic nhất.

3.4.1 Flowchart của AI Scout



Hình 3.4 Flow hoạt động của AI Scout trong map 1 và map 2

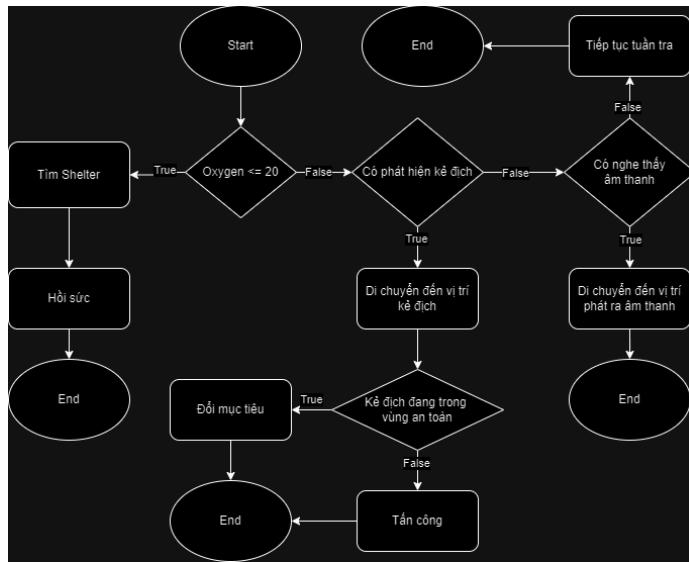
Đối với AI Scout trong map 1 và map 2, nhiệm vụ là tìm đến vị trí mục tiêu và tương tác. Tuy nhiên ưu tiên hàng đầu luôn là giữ cho bản thân có thể hoạt động được, vậy nên chỉ khi lượng oxygen không ở ngưỡng báo động, nhiệm vụ mới được thực thi. Ưu tiên thứ hai là tránh va chạm, vậy nên ngay khi phát hiện có khả năng gây giao chiến, AI Scout sẽ và tìm cách lánh mặt chứ không làm nhiệm vụ bắt cháp.



Hình 3.5 Flow hoạt động của AI Scout trong map3

Sang map3, ưu tiên hàng đầu vẫn là giữ khả năng sinh tồn, sau đó là thực hiện nhiệm vụ. Mục tiêu là trở thành người nắm giữ vương miện ở giây cuối cùng, vậy nên AI Scout sẽ luôn tìm kiếm vương miện. Khi đã có được thì sẽ chạy khỏi các mục tiêu khác để giữ vương miện không bị cướp

3.4.2 Flowchart của AI Defender



Hình 3.6 Flow hoạt động của AI Defender

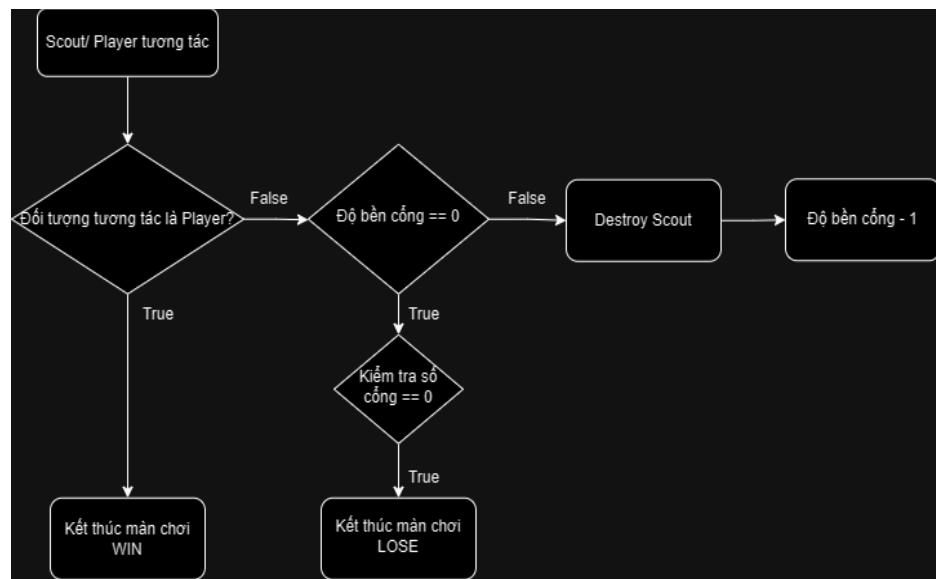
Tương tự như AI Scout, AI Defender luôn ưu tiên giữ bản thân có thể hoạt động. Ưu tiên tiếp theo là tiêu diệt kẻ xâm phạm và chỉ dừng lại khi kẻ địch đã trốn vào vùng an toàn (khu vực cấm đánh). Nếu không có mục tiêu nào để tấn công, Defender sẽ liên tục tuần tra quanh một tuyến đường nhất định và có thể rời vị trí tuần tra để đến điều tra nếu nghe thấy âm thanh đáng ngờ ở một khu vực gần đó.

3.4.3 Flowchart của Target map1

Trong map1, Player và các Scout sẽ phải tìm một cánh cửa để có thể bước qua màn sau. Màn chơi bao gồm 2 cửa và mỗi cửa chưa được 3 người trước khi bị phá hủy. Vậy nên nếu người chơi có thể tương tác được với 1 trong 2 cánh cửa là qua màn. Mỗi khi có 1 cách cửa bị phá hủy, GameMode sẽ kiểm tra số cửa còn lại. Nếu không còn cánh cửa nào, nghĩa là người chơi cũng đã hết cửa qua màn và GAME OVER.



Hình 3.7 Gate - Target của player trong map1

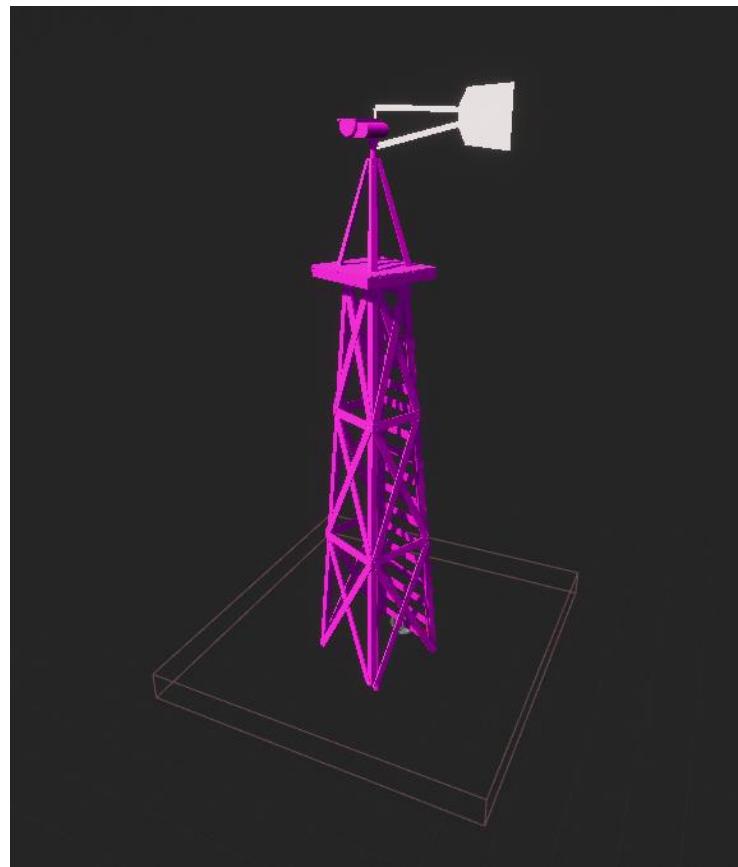


Hình 3.8 Flow xử lý của Gate

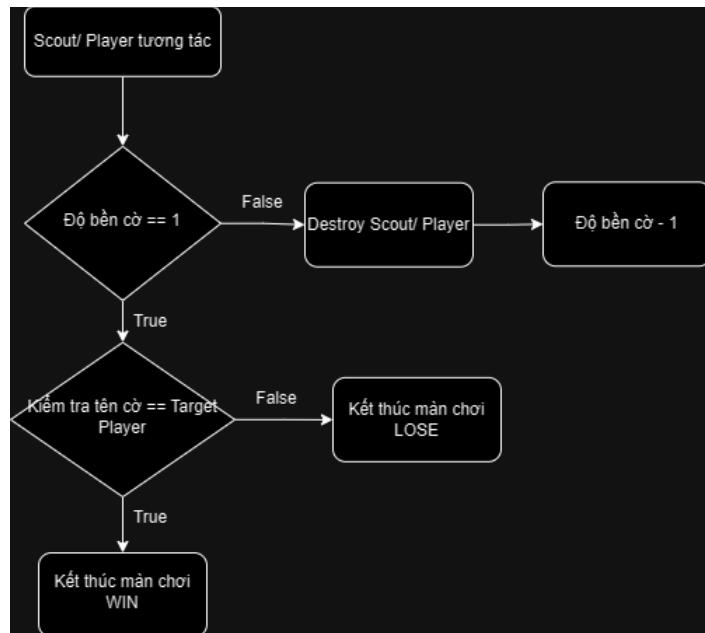
3.4.4 Flowchart của Target map2

Đối với map2, trò chơi sẽ có ngũ cành theo team và độ bền của mục tiêu cũng tăng lên nhiều. Người chơi cùng với các Scout đội mình sẽ tìm và tương tác với mục tiêu là cờ của đối phương. Đội nào có thể chạm vào cờ đối phương 10 lần trước là chiến thắng.

Mỗi lần nhân vật chạm vào cờ sẽ tăng 1 điểm cho phe ta và sẽ respawn tại một vị trí ngẫu nhiên và lặp lại nhiệm vụ. Ngay khi một bên có được 10 điểm, cờ đối thủ sẽ bị phá hủy. GameMode lúc này sẽ kiểm tra xem cờ của bên nào đã bị hủy, nếu cờ của team Player bị hủy nghĩa là GAME OVER, ngược lại thì sẽ đưa Player vào vòng cuối cùng.



Hình 3.9 Flag – Target của player trong map2



Hình 3.10 Flow xử lý của Flag

3.4.5 Flowchart của Target map3

Trong map3, số lượng mục tiêu tranh giành giảm xuống còn 1. Tất cả các đối tượng tham gia màn chơi đều có thể tương tác với mục tiêu để giành thắng lợi cuối cùng.

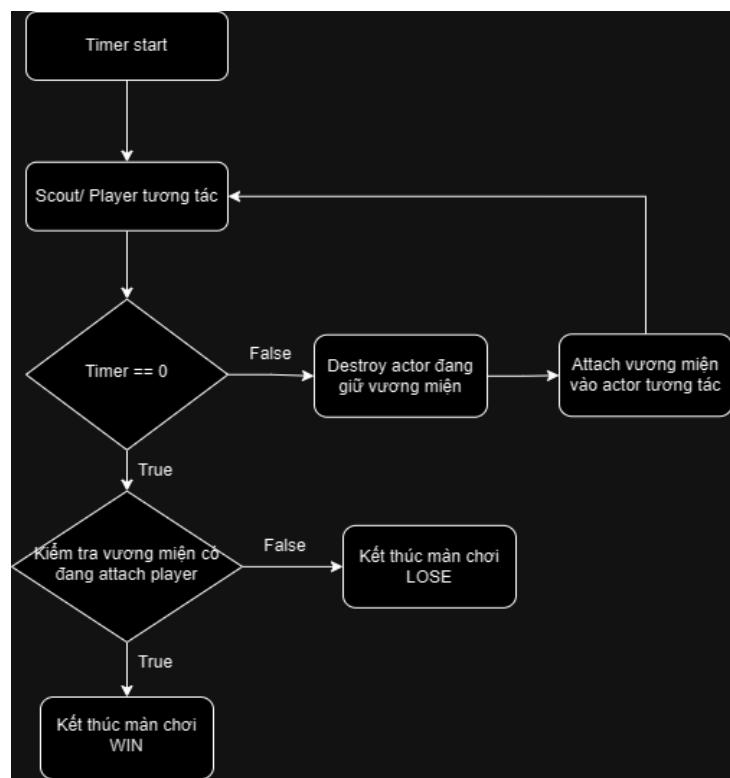
Tựa game For the Crown kết thúc với hình ảnh chiếc vương miện được tranh giành giữa các người chơi với nhau. Đồng hồ đếm ngược bắt đầu và khi con số trên đồng hồ trở thành số 0 thì người giữ vương miện sẽ là số 1 của trò chơi.

Tuy nhiên kẻ nắm giữ vương miện luôn có thể thay đổi vì chỉ cần 1 lần chạm là ngôi vương sẽ đổi chủ và kẻ bị truất ngôi sẽ bị tiêu diệt và respawn ở một nơi bất kì để lại 1 lần tìm kiếm ngôi vương.

Ngay khi timer về 0, GameMode sẽ kiểm tra nhân vật của Player nếu đang không có vương miện thì sẽ là GAME OVER, ngược lại thì người chơi sẽ hoàn thành trò chơi.



Hình 3.11 Crown – Target của player trong map3



Hình 3.12 Flow xử lý của Crown

3.4.6 Flowchart Shelter

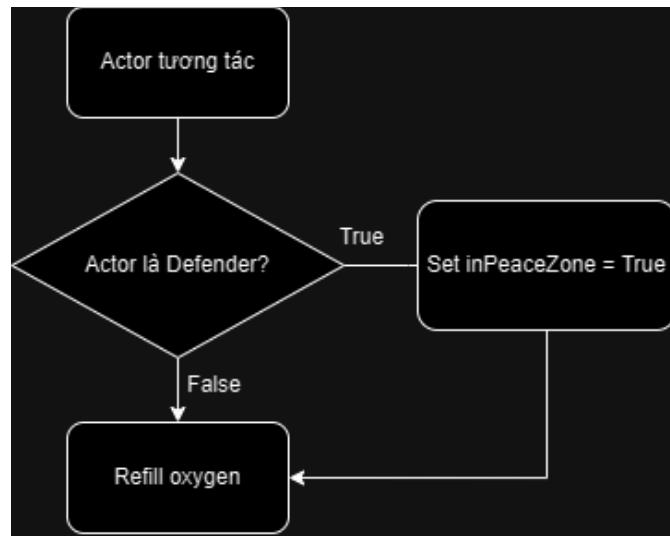
Để tạo thêm thử thách cho người chơi và tận dụng plugin SmartObject của Unreal, Shelter được tạo ra như một trạm dừng bắt buộc các actor phải ghé qua sau một khoảng thời gian nếu như không muốn phải Respawn.

Các actor sẽ liên tục bị giảm chỉ số Oxygen cho đến khi vào Shelter để hồi phục, giúp hạn chế khả năng tự do hoạt động của người chơi.

Đây cũng là nhà an toàn giúp Player và Scout cắt đuôi Defender khi bị truy đuổi vì đây là khu trung lập, các Defender sẽ không được phép tấn công khi vào đây.



Hình 3.13 Object Shelter

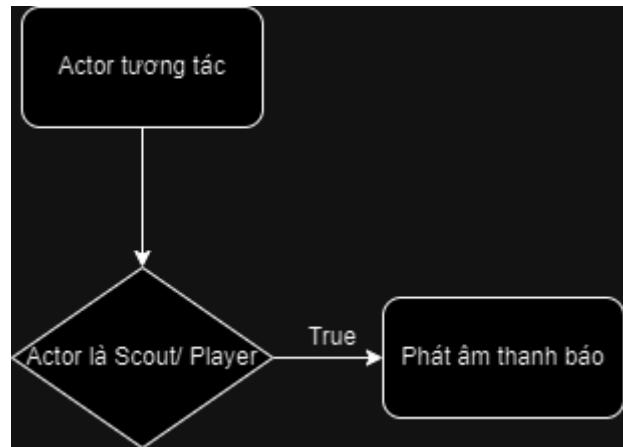


Hình 3.14 Flow Shelter xử lý khi có tương tác

3.4.7 Noisy thing

Object hoạt động như 1 trap, thông báo vị trí của Scout/ Player cho Defender.

Lại là một thử thách cho người chơi khi không biết mình sẽ gây tiếng động khi đang bước đến đâu và Defender thì không chỉ có mắt mà còn có tai.



Hình 3.15 Flow NoisyThing xử lý khi có tương tác

CHƯƠNG 4. QUÁ TRÌNH THỰC HIỆN ĐO ÁN

4.1 Lên ý tưởng về gameplay và tìm assets phù hợp

4.1.1 Ý tưởng

4.1.1.1 Ý tưởng ban đầu

- **Luật chơi:**

- **Cướp cờ (MODE 7 vs 7):** Player sẽ cùng team với bot > Mục đích là chiếm được cờ đối phương - Đứng tại khu vực cờ của đội thù sẽ tăng điểm dần - Đội full điểm trước WIN > Sẽ có vũ khí để knock đội thù trong 1 khoảng thời gian - Riêng nhân vật bị knock tại khu vực cờ đội phương sẽ spawn lại ở cờ phe mình.
- **Trốn – Bắt (MODE 1 vs 6):** 1 Player và 6 bot thay phiên vai trò> Game bắt đầu sẽ random vị trí và The chosen one > Đồng hồ đếm ngược về 0, nhân vật nào đang là The chosen one thời điểm đó Win - Có thể tương tác với The Chosen one để lấy danh hiệu đó - Sẽ có các skill với cooldown - Sẽ có các object spawn random trên map để gây hiệu ứng cho nhân vật.

- **Các đối tượng/ item trong game:**

- Item_Buff: tăng oxygen/ tốc độ
- Weapon: vũ khí cho nhân vật dùng 1 lần
- Item_Trap: tạo ra area of effect bất lợi cho nhân vật khác
- Environment trap: sẽ được setup khi tạo map/ được tạo ra bởi các item Trap.
- EVENT: môi trường sẽ được thay đổi để tạo lợi thế/ bất lợi ngẫu nhiên

Vì thời gian và năng lực có hạn, các asset free trên các store không đủ để đáp ứng hoàn thành ý tưởng ban đầu nên chúng em có thay đổi một vài chi tiết để hoàn thành.

4.1.1.2 Sản phẩm thực tế

Tổng quan cốt truyện: Bạn nhận được 1 cơ hội tranh đấu để sở hữu một hòn đảo giá trị. Tham gia và giành chiến thắng tất cả các vòng chơi để trở thành người thống trị của hòn đảo.

- Luật chơi:

- (Map 1) Đua tốc độ:
 - Single-play Context
 - Số lượng: 1 Player/ 11 AI Scout/ 6 AI Defender
 - Thủ thách: Mê cung + AI Defender (lính gác)
 - Mục tiêu: tìm được đến vị trí yêu cầu trước khi bị loại khỏi danh sách TOP 7
 - Đối thủ cạnh tranh: các AI Scout
 - Nội dung: Người chơi sẽ tìm cách vượt qua mê cung, tránh né các Defender và các bẫy âm thanh thu hút Defender, tìm đến mục tiêu trong thời gian có hạn để vượt qua thử thách.
- (Map 2) Cướp cờ:
 - Team-up context
 - Số lượng: 2 Team - (1 Player + 3 AI Scout + 2 AI Defender)/ (4 AI Scout + 2 AI Defender)
 - Thủ thách: Mê cung + AI Defender đối phương
 - Mục tiêu: tìm đến cờ đối thủ và tích điểm.
 - Đối thủ cạnh tranh: Team đối phương
 - Nội dung: Tham gia vào trận đấu đội cùng với các AI. Điểm số được cộng dồn cho cả đội và được tích lũy khi có bất kỳ thành viên nào chạm được cờ đối phương. Người chạm mục tiêu sẽ respawn ở vị trí ngẫu nhiên. Đội có 10 điểm trước chiến thắng.
- (Map 3) Người thắng lấy tất:
 - Single-play context

- Số lượng: 1 Player + 5 AI Scout
- Thủ thách: mê cung
- Mục tiêu: chiếm giữ vương miện đến khi thời gian đêm ngược kết thúc
- Đối thủ cạnh tranh: các AI Scout
- Nội dung: Tìm kiếm vị trí vương miện. Tương tác để chiếm. Kẻ đang đội vương miện sẽ bị tiêu diệt khi bị truất ngôi bởi kẻ chiếm.

4.1.2 Plugins hỗ trợ

- AI Behavior + SmartObject: các Plugin có sẵn của Unreal hỗ trợ quá trình xây dựng AI trong game tiện lợi và đơn giản nhất, giúp AI trở nên gần với một người chơi hơn. Tuy nhiên trong sản phẩm này, các plugin vẫn đang là phiên bản beta và chưa mang lại sự ổn định.
- Landmass: Plugin này hỗ trợ cho người sử dụng một blueprint để người dùng có thể tạo ra các vùng đồi cũng như vùng trũng một cách nhanh chóng và tiện lợi, người dùng chỉ cần kéo thả và điều chỉnh các note trên sợi dây các notes (spline) và điều chỉnh các thông số của brush blueprint details.
- Water Plugin: Plugin này hỗ trợ cho người sử dụng một blueprint để tạo các vùng nước, ví dụ như: biển, dòng suối, ... Người dùng chỉ cần chọn đúng cây cọ của mình cần (WaterBodyOcean, WaterBodyRiver, ...), một vùng nước sẽ hiện ra trên level hiện tại và bạn chỉ có việc điều chỉnh các thông số để có thể tạo ra cho mình một vùng nước mong muốn.

4.1.3 Assets

4.1.3.1 Environment

- Cargo: plugin thu viện asset tương tự như Quixel Brigde nhưng có nhiều lựa chọn hơn về mẫu mã cũng như texture, chỉ cần tải về và đưa vào dự án để sử dụng.

- Quixel Brigde: thư viện asset được phát triển bởi Unreal Engine, chỉ cần tải về và đưa vào dự án để sử dụng.
- Unreal Marketplace: nơi các asset được bán, chỉ sử dụng được khi mua và thêm thành công vào dự án.

4.1.3.2 Characters

- Mixamo: Một thư viện assets miễn phí của Adobe dành cho ngành đồ họa 3D. Người dùng chỉ cần đăng nhập bằng gmail và có thể sử dụng các tài nguyên ở trang web này một cách miễn phí. (“Mixamo,” n.d.)

4.1.3.3 Animations

- Mixamo: Một thư viện assets miễn phí của Adobe dành cho ngành đồ họa 3D. Người dùng chỉ cần đăng nhập bằng gmail và có thể sử dụng các tài nguyên ở trang web này một cách miễn phí. (“Mixamo,” n.d.)

4.2 Xây dựng map (LEVEL)



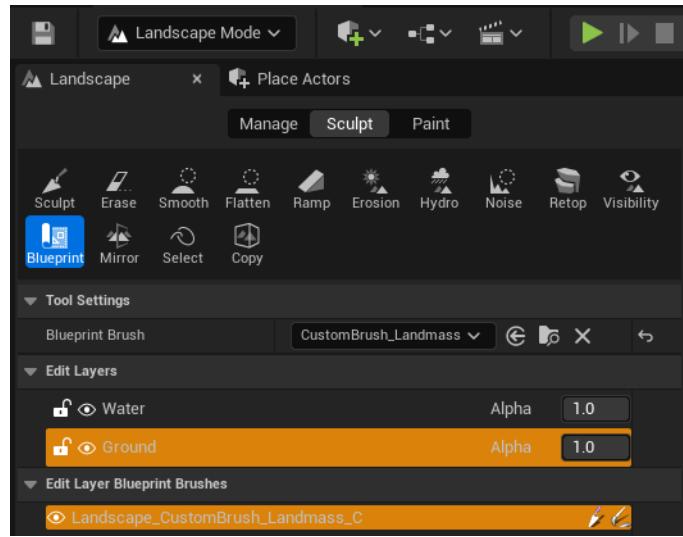
Hình 4.1 Kích hoạt Landmass Plugin

Sử dụng Landscape Mode, Foliage Mode, Blueprint và các plugins

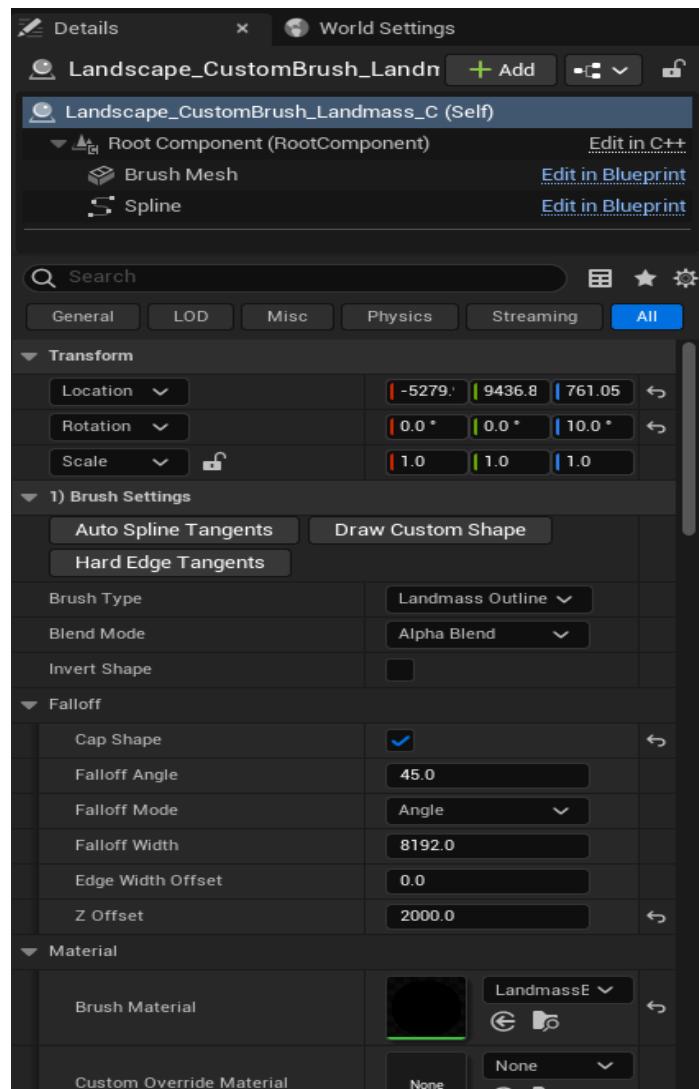
- **Landscape Mode và Landmass plugin**

- Tạo ra một bản đồ 64x64
- Kích hoạt Landmass plugin, plugin sẽ cho chúng ta một bộ blueprint về brush dạng đường spline
- Chuyển về chế độ Landscape Mode
- Án vào phần lựa chọn cọ, sau đó chọn blueprint.
- Sử dụng bộ blueprint của Landmass để có thể tạo map, điều chỉnh các thông số của map dễ dàng.

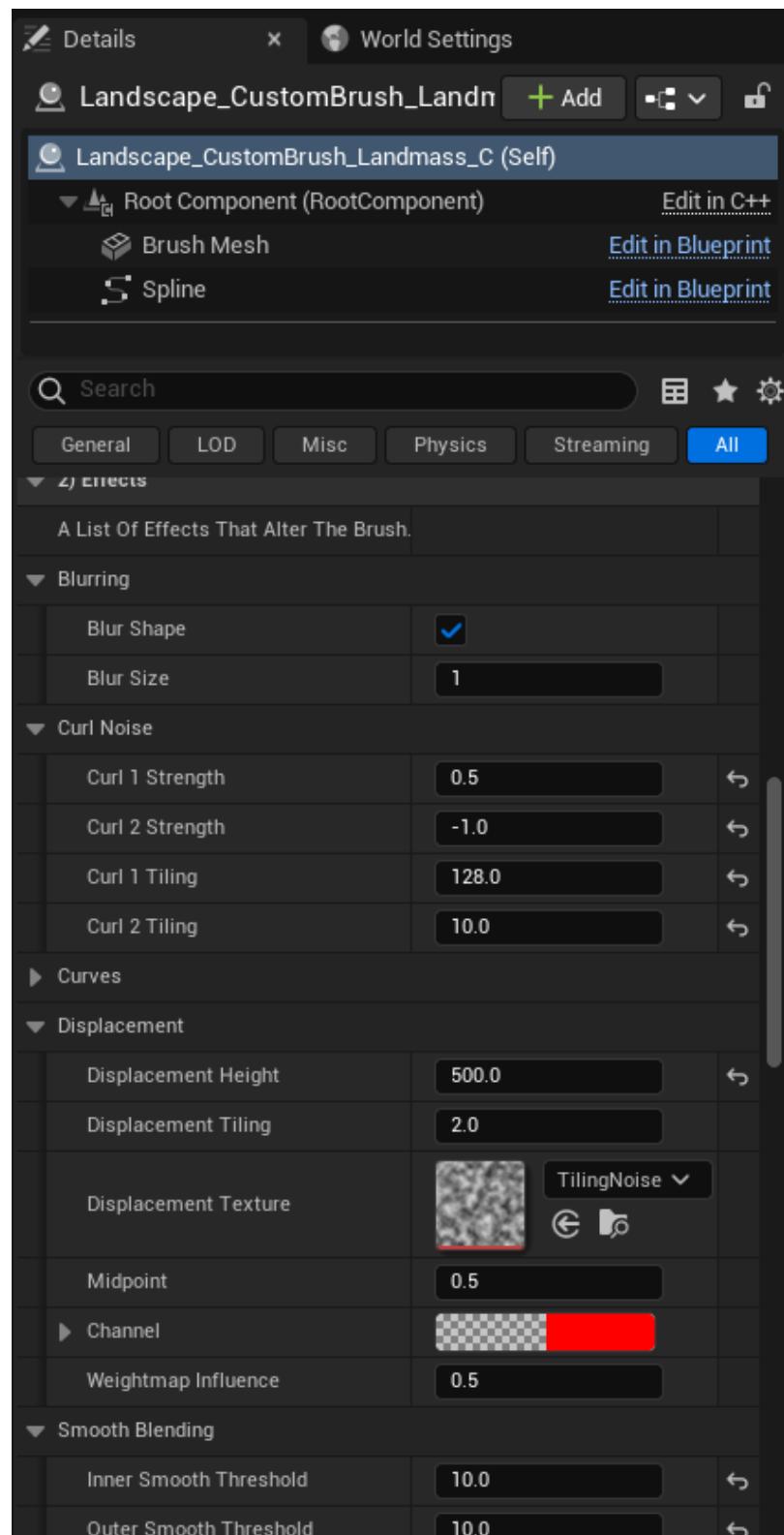
- Các thông số như: Falloff Angle, Z Offset, Capshape, Falloff width, Curl Noise, Displacement Height, Displacement Tiling, ... làm cho level (map) trở nên chân thực hơn, tự nhiên hơn mà không cần phải sculpt lại bằng cọ. Tuy nhiên chúng ta không thể điều chỉnh được độ noise trên map theo ý mà chúng ta mong muốn vì nó chỉ có thể random dựa trên thuật toán được tạo ra bởi Landmass Plugin



Hình 4.2 Plugin Brush của Landmass Plugin

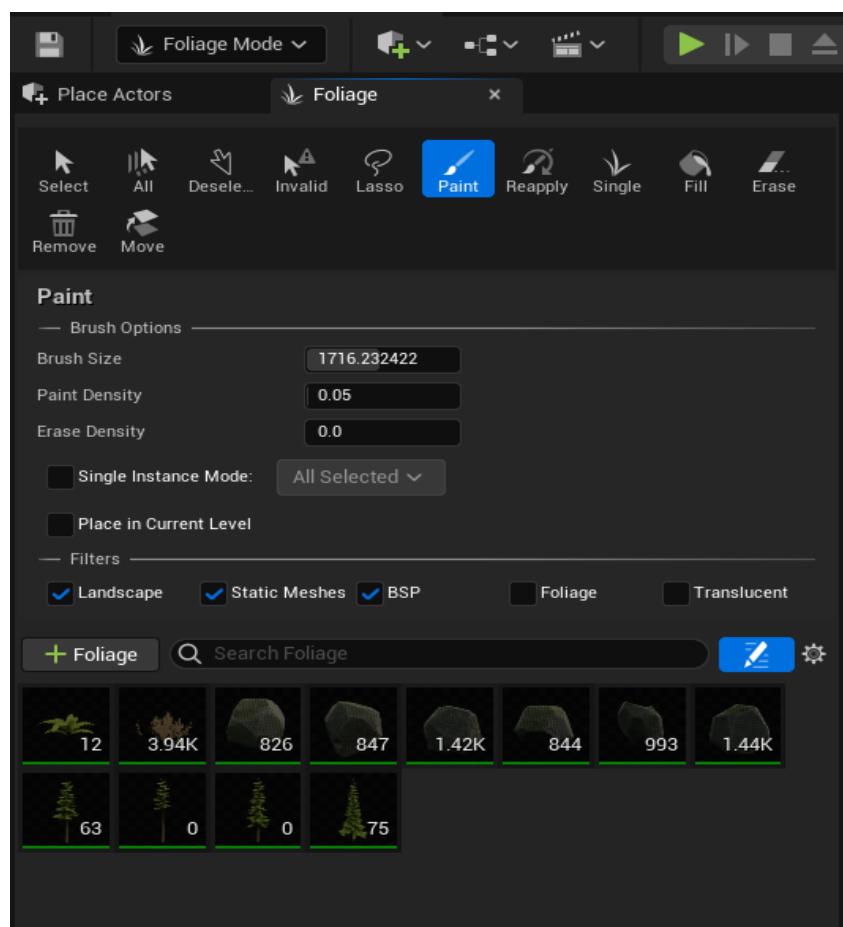


Hình 4.3 Thông số của blueprint brush



Hình 4.4 Landmass Brush Details

- Sculpt canh chỉnh lại độ gập ghènh cũng như bền mặt phẳng của map.
 - Kích hoạt Water plugin để tạo vùng nước mong muốn
 - Kéo thả Water Body Ocean vào map hiện tại, sau đó canh chỉnh để có thể tạo ra một vùng nước mà mình mong muốn.
 - Tìm các asset về vách núi, núi, đất đá, texture để có thể gán lên bề mặt của các vật thể cũng như toàn bộ playground.
- Foliage Mode
- Tạo ra môi trường (cây cối, đá, đất cát, ...) một cách ngẫu nhiên và hàng loạt.



Hình 4.5 Giao diện Foliage Mode

- Kích hoạt tính năng collision của các vật thể nêu trên để nhân vật có thể tác động vật lý lên các Foliage Instance đó.
- Án chọn vào những assets mình cần sử dụng (một hoặc nhiều) sau đó chọn cọ Paint để thêm các asset đó vào level hiện tại của mình.
- Blueprint
 - BP_Berrier: Tạo một blueprint actor đặt tên Berrier, mục đích của việc sử dụng blueprint là có thể giúp chúng ta tái sử dụng cũng như việc chỉnh sửa dễ dàng trên vật thể hơn.



Hình 4.6 BP_Berrier

- BP_Crown: Tạo ra một blueprint actor Crown để đại diện cho vật thể vương miệng được sử dụng cho gameplay ở map 3.
- Water System plugin (“Water System,” n.d.)
 - Kích hoạt water plugin sau đó reset UE.
 - Tìm kiếm trong mục GET CONTENT tên class WaterBodyOcean hoặc các class về water plugin có liên quan để sử dụng.
 - Kéo thả WaterBodyOcean và điều chỉnh kích thước cũng như thông số cho phù hợp với map.
 - Sau khi hoàn tất một vùng nước sẽ hiện lên trên map.

4.3 Xây dựng AI (BOT)

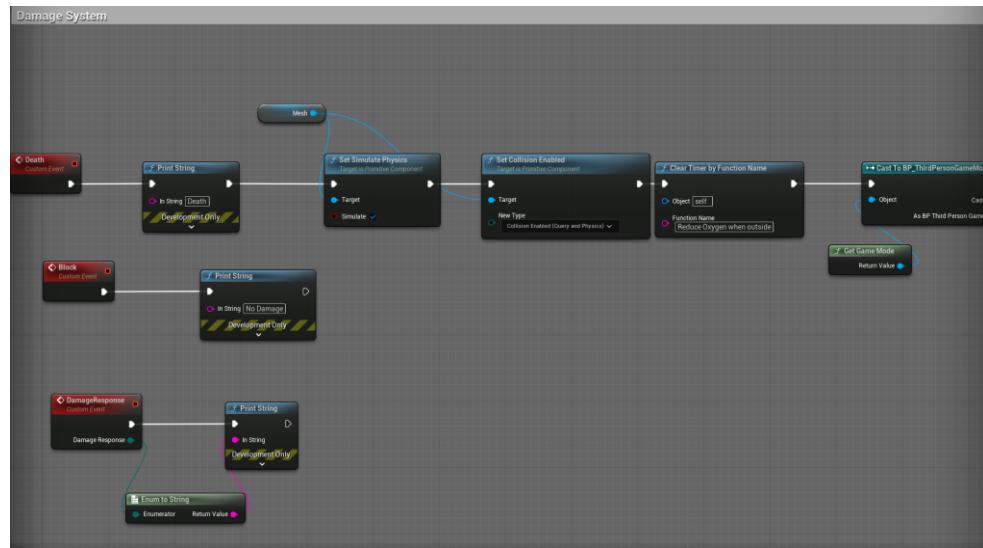
Mỗi role AI sẽ có một blueprint riêng và dùng chung cho xuyên suốt trò chơi. Blueprint này sẽ bao gồm một hình nộm giúp người chơi phân biệt được vai trò của AI và một bộ xử lý riêng cho từng trường hợp.

4.3.1 Blueprint cho Scout

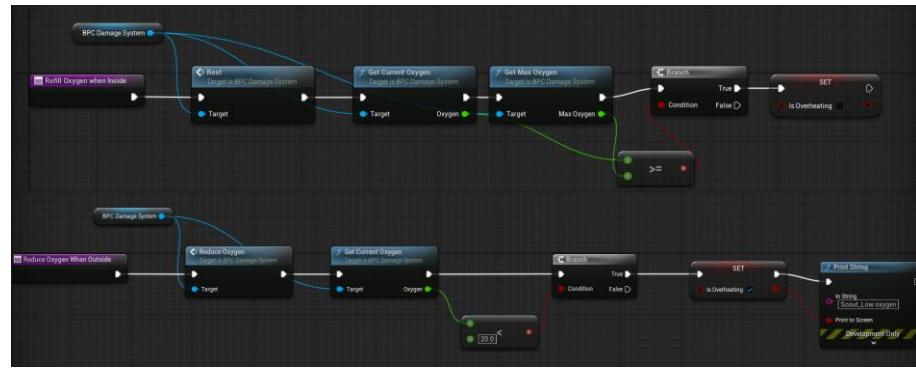
Dùng để tạo các actor thuộc role Scout. Đây cũng là vai trò của người chơi khi có các mục tiêu nhiệm vụ thay đổi xuyên suốt các màn chơi.

Bao gồm các hàm và chức năng cơ bản:

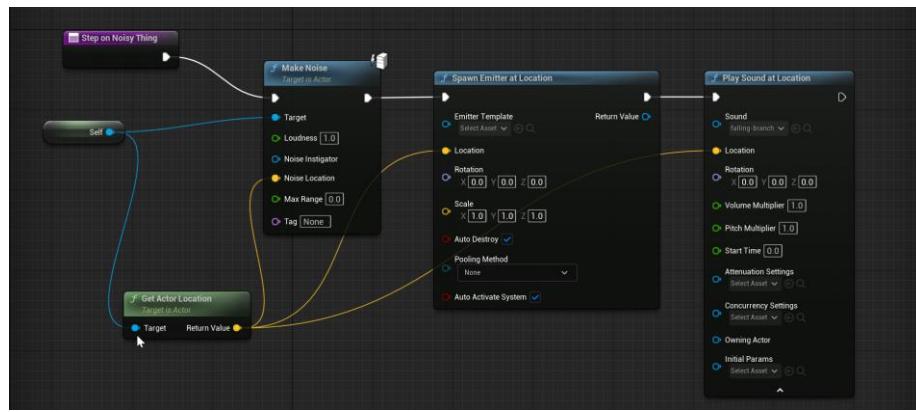
- Các hàm xử lý Damage System (sẽ được đề cập ở phần Perception)
- Các hàm tăng/ giảm Oxygen (sẽ được đề cập ở phần Gameplay)
- Hàm tạo tiếng động (sẽ được đề cập ở phần Perception)



Hình 4.7 Các hàm xử lý Damage System

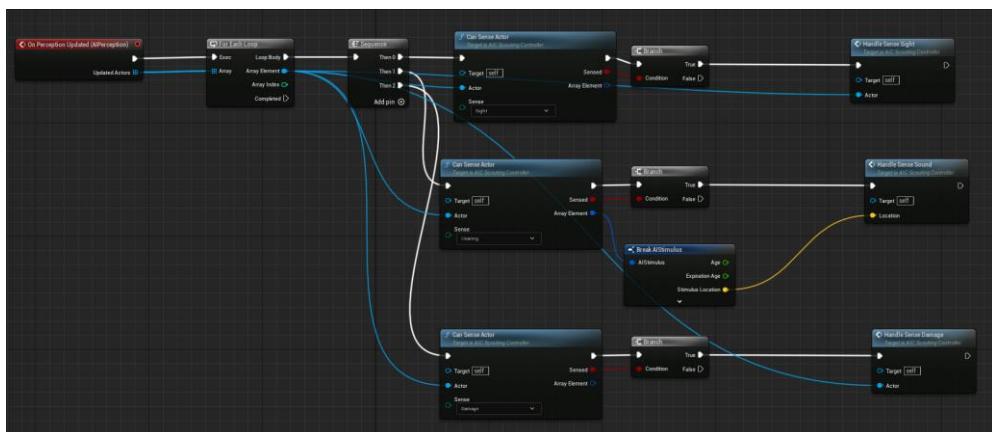


Hình 4.8 Các hàm tăng/ giảm Oxygen



Hình 4.9 Hàm tạo tiếng động

Để các AI hoạt động, Unreal Engine yêu cầu có một class AIController như bộ não, điều khiển khung xương hoạt động. Chúng ta có thể tùy chỉnh AIController để phù hợp với yêu cầu. Cụ thể, việc xây dựng Perception (tri giác) cho AI cũng được thực hiện trong AIController, giúp một actor AI tiến gần hơn với việc xử lý như một con người.

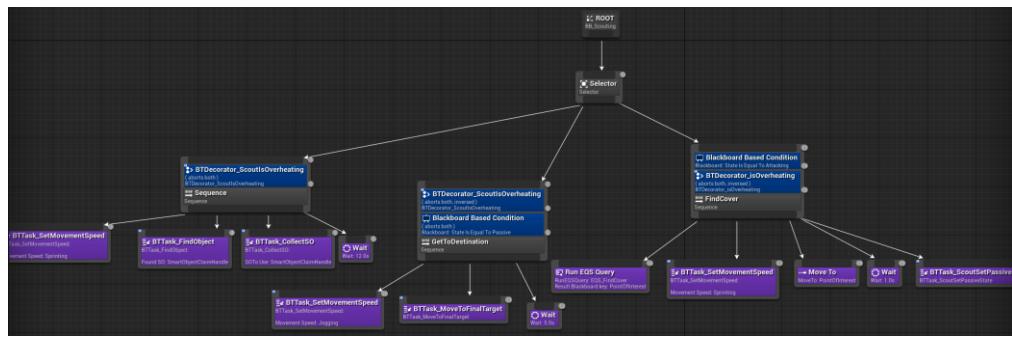


Hình 4.10 Các Perception của AI

Các AI bao gồm các tri giác: thị giác, thính giác và cảm nhận sát thương. Tùy vào ngữ cảnh màn chơi có thể chỉnh sửa cách AI xử lý khi nhận được các tri giác này.

Đối với các AI Defender, nhằm tạo điều kiện cho các Defender dễ phát hiện kẻ địch hơn, thính giác được thêm vào để không chỉ dựa vào tầm nhìn, các Defender còn có thể dùng thính giác để tiếp cận gần với mục tiêu hơn.

Mỗi AIController cần đính kèm với một BehaviorTree, hiểu đơn giản thì nếu AIController là bộ não thì BehaviorTree là cách bộ não suy nghĩ. Chúng ta có thể lập trình cách AI xử lý các sự kiện có thể xảy ra thông qua BehaviorTree.



Hình 4.11 BehaviorTree của Scout map1

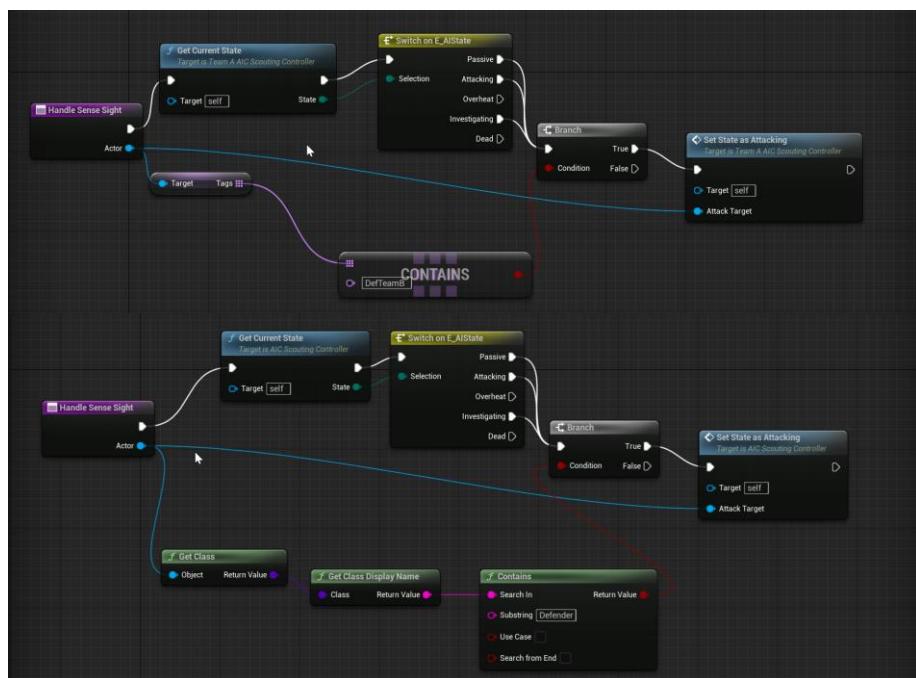
BehaviorTree sẽ thực thi từ trên xuống dưới, từ trái qua phải. Vậy nên các trường hợp xử lý tình huống cũng được xếp theo thứ tự ưu tiên như vậy. Các nhiệm vụ quan trọng nhất sẽ được sắp xếp dần từ trái qua và giảm dần độ ưu tiên khi đi qua phải.

Cơ chế Blackboard/ Decorator cũng là cơ chế hỗ trợ quá trình xử lý các tình huống xảy ra dễ dàng hơn. Chúng ta có thể gọi đây là các khóa điều kiện và chỉ khi các khóa này có sẵn, một chuỗi xử lý mới được thực thi. Ngay khi khóa không còn đúng, chuỗi xử lý sẽ ngắt và AI sẽ thực hiện kiểm tra lại BehaviorTree từ Root.

Có 3 blueprint cho AI_Scout_Controller đi kèm với 3 BehaviorTree khác nhau để xử lý tình huống trong những ngữ cảnh khác nhau:

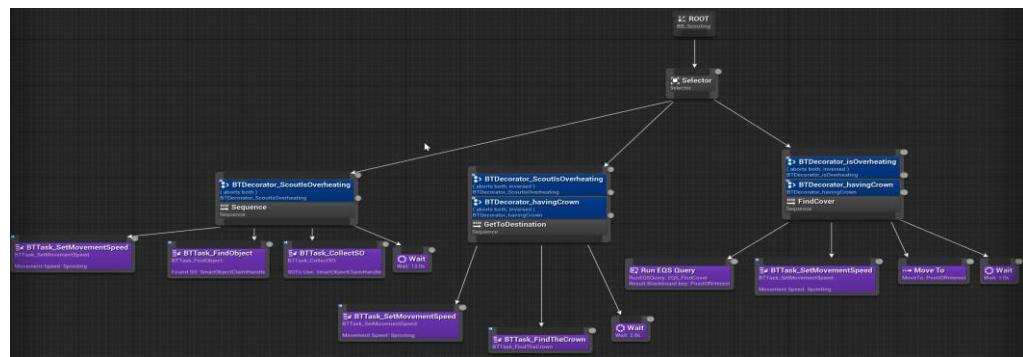
- Map 1: Yêu cầu AI Scout di chuyển đến mục tiêu trong khi tránh né các AI Defender.
 - o Lúc này AI sẽ nhận định kẻ thù chính là các Defender và kích hoạt trạng thái giao chiến (tránh né các Defender).

- Vậy nên chỉ khi đang không bị đe dọa (blackboard-key State == Passive), AI Scout sẽ tìm đến mục tiêu của mình.
- Ngay khi phát hiện nguy hiểm (blackboard-key State == Attacking), kích hoạt nhánh tìm vị trí an toàn để trốn.
- Và ưu tiên hàng đầu luôn là giữ Oxygen ở mức an toàn, vậy nên khi Oxygen xuống ngưỡng nhất định (Decorator isOverheat == True), AI Scout dừng mọi hoạt động để tìm nơi hồi phục.
- Map 2: Điểm khác biệt so với map1 là AI Scout lúc này có thể nhận biết được AI Defender đồng minh và AI Defender kẻ địch.
- Trạng thái giao chiến chỉ được kích hoạt khi đối mặt với AI Defender kẻ địch.
- Sự khác biệt này đến từ AIController của map2 khi lúc này các AI Scout sẽ chỉ nhận định các Defender của team đối địch là kẻ thù, chứ không phải toàn bộ các Defender.
- BehaviorTree sẽ không có điểm khác biệt so với map1.



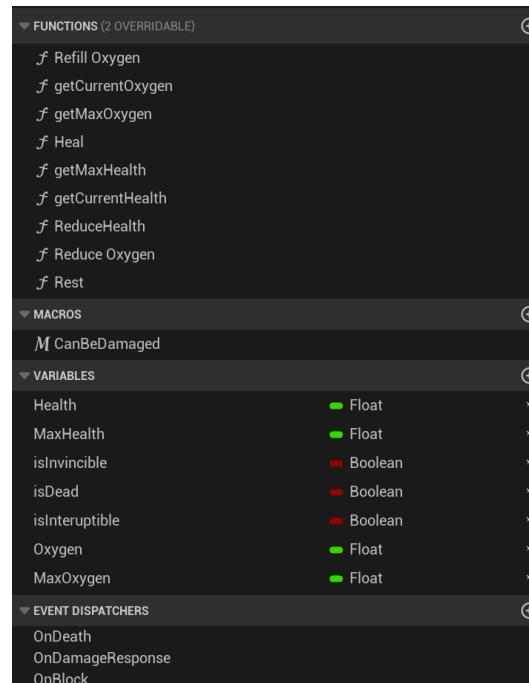
Hình 4.12 Hàm xử lý Sight của Scout map2 (hình trên) và map1 (hình dưới)

- Map3: Không còn các AI Defender, AI Scout lúc này chỉ cần nhận biết:
 - o Nếu bản thân đang không có vương miện thì tìm cách đến cướp vương miện.
 - o Nếu bản thân đang có vương miện thì phải tìm cách trốn khỏi các đối tượng đang tiến đến bản thân.
 - o Lúc này, điểm cần thay đổi chính là BehaviorTree và cần thêm một Decorator để xác định AI đang ở trường hợp nào.



Hình 4.13 BehaviorTree của Scout map3

Damage System là phần không thể thiếu vì nó giúp áp dụng cơ chế sống còn cho người chơi, vì nếu người chơi bất khả chiến bại thì trò chơi sẽ không còn độ khó



Hình 4.14 Các thuộc tính cơ bản của người chơi

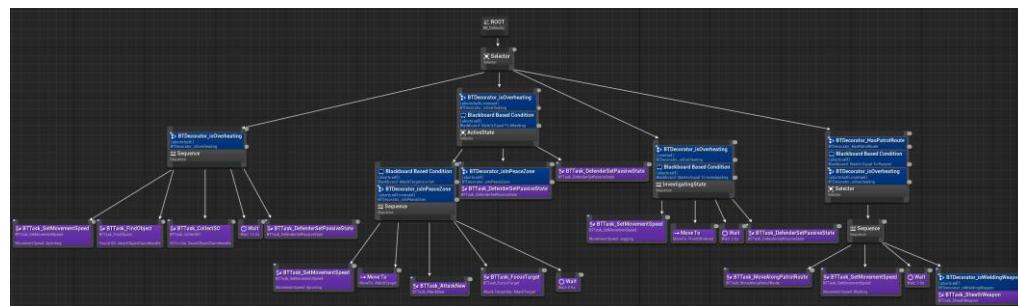
Trong đó, quan trọng nhất có thể kể đến:

- Oxygen: cơ chế gameplay thêm thử thách khi buộc người chơi phải hồi sức sau một khoảng thời gian nhất định
- Health: chỉ số sức khỏe, quyết định người chơi có thể tiếp tục hoạt động hay sẽ phải respawn.

4.3.2 Blueprint cho Defender

Tương tự như Scout, các Defender cũng có hệ thống tri giác Perception, cũng cần 2 AIController và 2 BehaviorTree cho 2 map khác nhau. Tuy nhiên, do các Defender được xây dựng với mục đích là các thử thách cản trở người chơi, nên đây là loại đối tượng bất khả chiến bại, vậy nên Damage System không có nhiều ứng dụng.

Điểm khác biệt có thể kể tới là AI Defender có một tuyến tuần tra nhất định và có thể kết hợp thính giác để điều tra phát hiện kẻ thù.

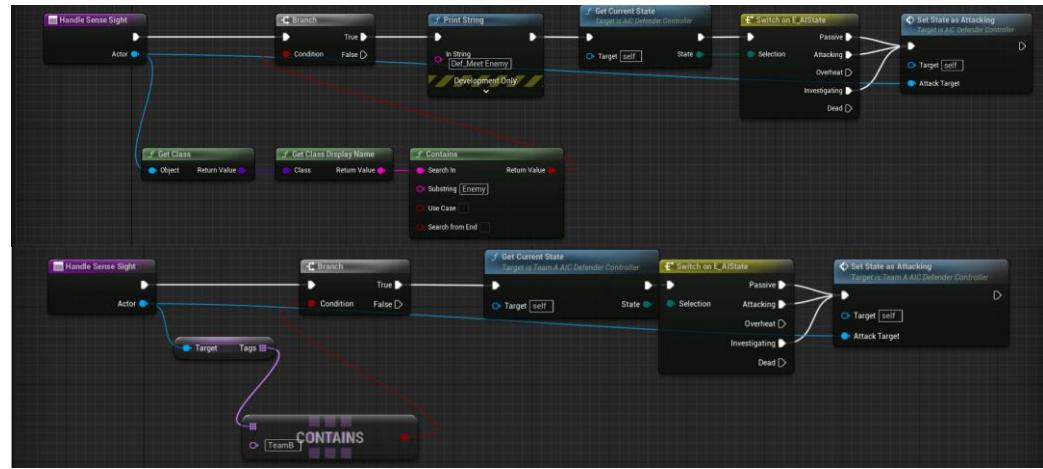


Hình 4.15 BehaviorTree của Defender map1

Trình tự xử lý của AI Defender như sau:

- Đầu tiên hàng đầu là giữ Oxygen ở mức an toàn.
- Khi nhìn thấy kẻ địch sẽ tiến hành tấn công tiêu diệt.
- Nếu đang tuần tra và nghe thấy âm thanh (khi AI Scout hoặc người chơi dẫm vào các bẫy) thì sẽ đi đến vị trí đó để điều tra.
- Nếu không có gì bất thường, Defender sẽ tiếp tục tuần tra.

Cũng giống như Scout, các Defender cũng cần thay đổi cơ chế nhận diện kẻ thù trong map2 (có đồng minh) so với map1 (có thể tiêu diệt tất cả) để tránh trường hợp quân ta lo đánh quân mình mà bỏ quên quân địch dẫn đến thua trận.



Hình 4.16 Hàm xử lý Sight của Defender trong ngũ cảnh đơn (hình trên) và
trong ngũ cảnh đội (hình dưới)

Trong tựa game này, người chơi có những thử thách và nhiệm vụ cần phải vượt qua để có thể giành được chiến thắng cuối cùng. Vì vậy chúng em xây dựng các cơ chế gameplay khác biệt giữa các map.

4.4 Xây dựng thử thách

4.4.1 Thử thách mê cung

Mục tiêu: gây khó khăn cho người chơi trong quá trình thực hiện nhiệm vụ.

Nội dung:

- Mê cung sẽ có thay đổi tùy theo từng map, giúp người chơi không bị chán vì đã biết đường đi từ trước.
- Kết hợp với các tuyến tuần tra khác nhau, người chơi sẽ không biết trước được các AI Defender có thể xuất hiện từ hướng nào.
- Tạo ra các vật cản giúp người chơi và AI Scout tránh né tầm mắt của AI Defender



Hình 4.17 Mê cung cho map1

4.4.2 Cơ chế Oxygen

Mục tiêu: hạn chế khả năng hoạt động của người chơi ngoài map, tạo ra các biến số cho game.

Nội dung: tất cả các actor trong game đều có một lượng giới hạn Oxygen.

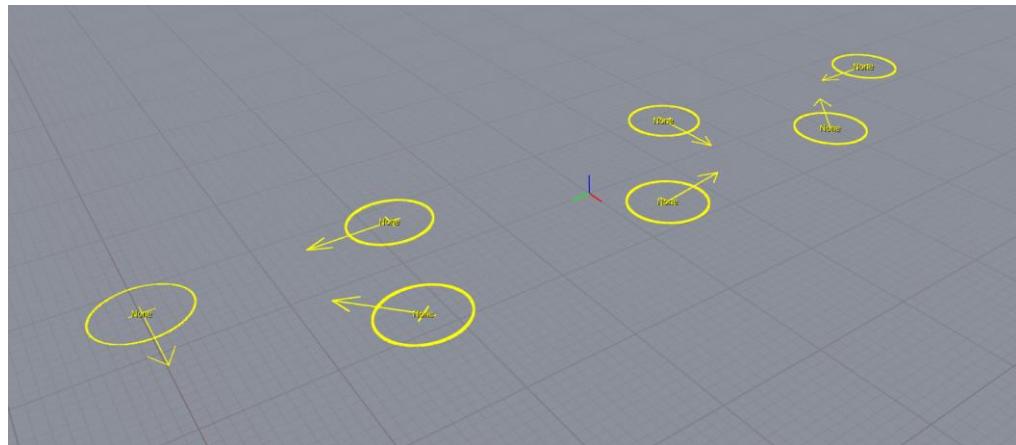
- Nếu chỉ số này về 0, đồng nghĩa actor đó phải respawn, gián đoạn quá trình chơi.
- Để tránh việc chỉ số về 0, cần phải tìm đến các trạm trú ẩn được đặt trên bản đồ để hồi phục.
- Để áp dụng đồng bộ cơ chế này cho cả người chơi và các AI, chúng em đã nghiên cứu thêm về plugin SmartObject của Unreal Engine.



Hình 4.18 Nhà trú ẩn giúp phục hồi Oxygen và tránh bị tấn công

SmartObject là cơ chế giúp AI tìm và sử dụng các Object giống như cách mà người chơi sử dụng một Object. Để một SmartObject hoạt động cần:

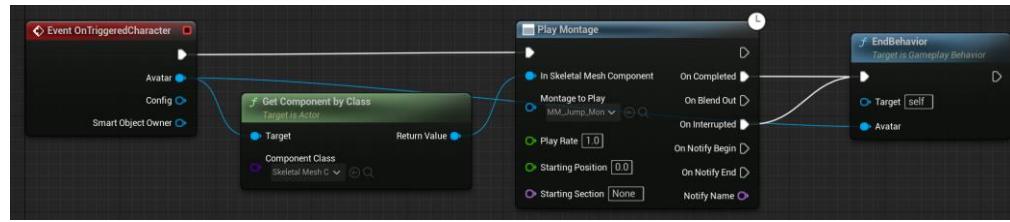
DataAsset: thứ quyết định các slot dành cho AI. AI sẽ tìm đến các slot này để sử dụng object.



Hình 4.19 Các slot của DataAsset

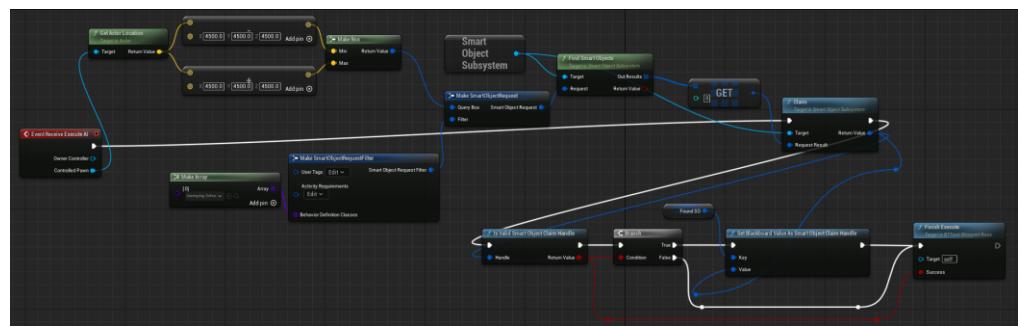
Để AI biết cần làm gì với các slot này, ta sẽ có 1 file Config, thứ sẽ quyết định hành vi của AI đối với các slot SmartObject.

GameplayBehavior: đối với người chơi, chúng ta biết được mình tương tác với một object để làm gì. Vậy còn với AI, làm cách nào để ta biết chúng đang sử dụng object đó? Câu trả lời là thông qua Behavior.



Hình 4.20 Hàm sử dụng SmartObject của AI

Trong game này, khi AI phát hiện chỉ số Oxygen đang thấp báo động, chúng sẽ dừng mọi công việc đang làm, tìm đến 1 slot của SmartObject nơi trú ẩn, và thực hiện anim ngồi nghỉ. SmartObject giúp hiện thực cơ chế game dễ dàng và thực tế hơn.



Hình 4.21 Hàm tìm kiếm các slot của SmartObject

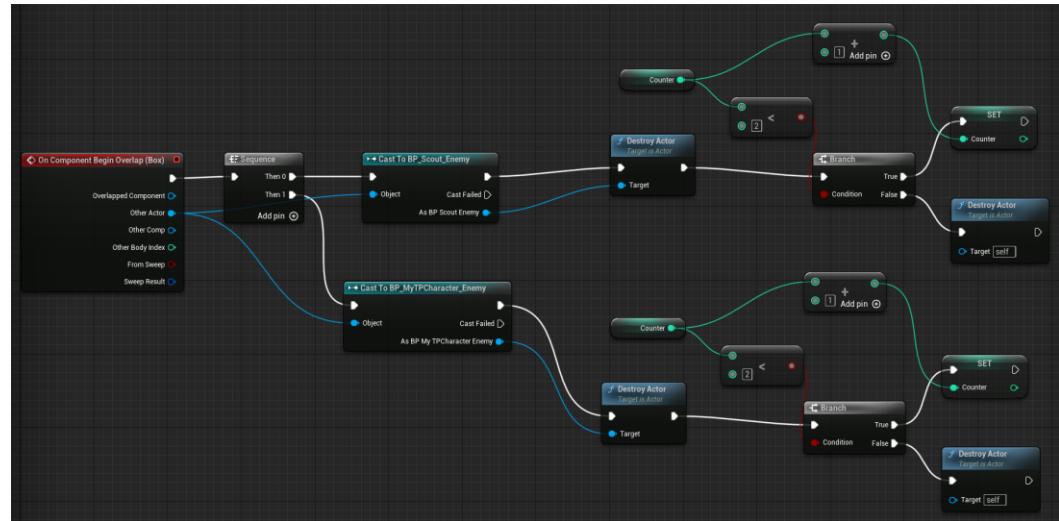
4.4.3 Các nhiệm vụ của game

Map1: Yêu cầu người chơi vượt qua mê cung và tìm đến cổng



Hình 4.22 Cổng mục tiêu của map1

Cổng chỉ cho phép tối đa 3 người qua, ngay khi đã nhận đủ 3 người, cổng sẽ bị hủy. Sẽ có 2 cổng, nhận tối đa 6 người. Ngay khi cả 2 cổng bị phá hủy, nếu người chơi không phải một trong những người bước qua cổng thì trò chơi kết thúc.



Hình 4.23 Hàm tương tác với mục tiêu map1

Map2: yêu cầu người chơi tìm đến cờ đối phương để tích điểm. Điểm sẽ được tăng lên mỗi khi có thành viên của team bắt kịp chạm đến cờ đối phương.

Đội tích lũy được 10 điểm trước sẽ phá hủy cờ đối phương. Ngay khi cờ của 1 bên bị phá hủy, bên còn lại chiến thắng.

Map3: nắm giữ vương miện đến khi thời gian đếm ngược kết thúc là giành chiến thắng.

Nhận biết vị trí kẻ chiếm vương miện thông qua hình ảnh vương miện trên đầu. Tương tác với vương miện để chiếm ngôi

Kẻ đang có vương miện nếu bị truất ngôi sẽ bị tiêu diệt và respawn ngẫu nhiên.

CHƯƠNG 5. KẾT LUẬN

Sau quá trình nghiên cứu và thực hiện đề tài, chúng em đã đúc kết được kinh nghiệm và rút ra được kết luận sau:

5.1 Ưu điểm

Có cơ hội học hỏi và sử dụng một trong những game engine phổ biến nhất hiện tại.

Năm được quy trình cơ bản của việc xây dựng một tựa game, các công đoạn quan trọng cũng như các khó khăn mắc phải.

Có cơ hội tiếp xúc với một cách thức lập trình mới ngoài các dòng lệnh là lập trình bằng blueprint.

Biết cách sử dụng bộ tool của Unreal Engine để tự xây dựng map.

Xây dựng được một cốt truyện tương đối hoàn chỉnh.

Có thể lập trình được một bộ AI đơn giản.

Ứng dụng được các plugin hỗ trợ của Unreal vào trong quá trình thực hiện đề tài.

Tự mình làm được 1 game đơn giản.

5.2 Nhược điểm

Chưa thể đầu tư xây dựng một bộ AI phức tạp và giống người nhất có thể để tăng độ thử thách cho game.

Chưa hoàn thành được bộ animation cho các nhân vật để tạo trải nghiệm tốt nhất cho người chơi.

Chưa làm được một bộ item tương tác giúp gameplay thú vị hơn.

Chưa code theo mô hình OOP, khiến việc mở rộng và thay đổi tính năng còn khó khăn.

Chưa đóng gói được tựa game thành một sản phẩm độc lập.

5.3 Hướng phát triển

Cải thiện chièu sâu và xây dựng một bộ AI hoàn chỉnh hơn, bao gồm thêm các tính năng:

- Scout:

- Có khả năng đánh lạc hướng Defender.
- Báo động cho đồng đội nếu phát hiện kẻ địch.
- Ra quyết định giữa trốn hoặc tiếp tục nhiệm vụ.

- Defender:

- Có khả năng tìm mục tiêu tối ưu hơn để tấn công.
- Kêu gọi hỗ trợ.

Xây dựng thêm nhiều trạng thái (Choáng, đóng băng, rối loạn, ...) cho các nhân vật.

Xây dựng bộ Animation hoàn chỉnh cho các actor.

Mở rộng quy mô và mức độ chi tiết của các map/ mê cung.

Cải thiện visual của các actor.

Xây dựng thêm cơ chế tương tác với Item cho AI.

Cấu trúc lại các actor theo hướng OOP để mở rộng và thay đổi.

Xây dựng thêm màn chơi cho phép Player có vai trò như Defender.

Thêm cơ chế thay đổi địa hình và mê cung thay đổi.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Không.

Tiếng Anh

Cargo by KitBash3D | Free 3D Asset Manager for Blender, Unreal Engine 5, Cinema4D, 3ds Max, Maya, and more. (n.d.). Retrieved August 2, 2023, from KitBash3D website: <https://kitbash3d.com/pages/cargo>

Creating Landscapes. (n.d.). Retrieved July 8, 2023, from <https://docs.unrealengine.com/5.2/en-US/creating-landscapes-in-unreal-engine/>

Foliage Mode. (n.d.). Retrieved July 25, 2023, from <https://docs.unrealengine.com/5.2/en-US/foliage-mode-in-unreal-engine/>

Fracture Mode Selection Tools User Guide. (n.d.). Retrieved July 25, 2023, from <https://docs.unrealengine.com/5.2/en-US/fracture-mode-selection-tools-user-guide/>

Mixamo. (n.d.). Retrieved August 2, 2023, from <https://www.mixamo.com/#/?page=1&type=Character>

Unreal Engine 5.2 Documentation. (n.d.). Retrieved August 2, 2023, from <https://docs.unrealengine.com/5.2/en-US/>

Water System. (n.d.). Retrieved August 2, 2023, from <https://docs.unrealengine.com/5.0/en-US/water-system-in-unreal-engine/>