|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



BÙI QUANG HÙNG

Xây dựng Game 3D Action “Lockdown Gunblade” bằng Unity

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, 2024

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

BÙI QUANG HÙNG

Xây dựng Game 3D Action “Lockdown Gunblade” bằng Unity

|  |  |
| --- | --- |
| Ngành : | Kĩ thuật phần mềm |
| Mã số: | 7480103 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | **TS. Trương Xuân nam** |
|  |  |

HÀ NỘI, NĂM 2024

**GÁY BÌA ĐỒ ÁN TỐT NGHIỆP**

**BÙI QUANG HÙNG ĐỒ ÁN TỐT NGHIỆP HÀ NỘI, NĂM 2024**

|  |  |
| --- | --- |
| Logo-WRU | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

Họ tên sinh viên: Bùi Quang Hùng Hệ đào tạo : Đại học chính quy

Lớp: 62PM1 Ngành: Kỹ thuật Phần mềm

Khoa: Công nghệ thông tin

1. TÊN ĐỀ TÀI:

XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY

1. CÁC TÀI LIỆU CƠ BẢN:

* Quy trình giao đồ án tốt nghiệp khoa Công nghệ thông tin Đại học Thủy Lợi (2021)
* Website
* <https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation>
* <https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/>
* <https://help.bizagi.com/platform/en/index.html?intro_welcome.htm>

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

|  |  |
| --- | --- |
| **Nội dung cần thuyết minh** | **Tỷ lệ** |
| 1. Mở đầu | 2% |
| 2. Chương I. Tổng quan về ngành game | 4% |
| 3. Chương II. Tổng quan về Unity | 44% |
| 4. Chương III. Game bắn súng bằng Unity | 15% |
| 5. Chương IV: Phân tích và thiết kế hệ thống | 19% |
| 6. Chương V: Xây dựng và demo | 6% |
| 7. Chương VI: Kết luận | 2% |

1. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

|  |  |
| --- | --- |
| **Các phần** | **GVHD** |
| 1. Mở đầu | TS. Trương Xuân Nam |
| 2. Chương I. Tổng quan mô hình hóa nghiệp vụ | TS. Trương Xuân Nam |
| 3. Chương II. Tổng quan về phần mềm Unity | TS. Trương Xuân Nam |
| 4. Chương III. Game bắng súng bằng Unity | TS. Trương Xuân Nam |
| 5. Chương IV: Mô hình chức năng của game | TS. Trương Xuân Nam |
| 6. Chương V: Xây dựng và demo | TS. Trương Xuân Nam |
| 7. Chương VI: Kết luận | TS. Trương Xuân Nam |

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày ..........tháng ......... năm 202..

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày. . . . .tháng. . . . .năm 202...

**Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày ... tháng ... năm 202...

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*

Bùi Quang Hùng

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

TÊN ĐỀ TÀI: **XÂY DỰNG GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY**

*Sinh viên thực hiện*: Bùi Quang Hùng

*Lớp*: 62PM1

*Mã sinh viên :* 2051063675

*Giáo viên hướng dẫn*: ThS. Trương Xuân Nam.

**TÓM TẮT ĐỀ TÀI**

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe hơn từ phía những người dùng, đặc biệt là về sản phẩm Game được nhận rất nhiều sự đánh giá từ phía các Game thủ, hay chỉ là những người chơi bình thường. Ngành công nghiệp Game hiện nay có thể nói là bùng nổ, với tốc độ phát triển đến chóng mặt, rất nhiều những Game hay và hấp dẫn đã được ra đời trong thời gian qua. Trong đấy dòng game 3D hành động theo góc nhìn thứ ba (Third-Person - TPS) có lẽ nhận được sự quan tâm nhiều nhất từ các game thủ

Game 3D hành động theo góc nhìn thứ ba đã trở thành một thể loại phổ biến và được yêu thích trong ngành công nghiệp game. Với góc nhìn từ phía sau nhân vật chính, người chơi có cái nhìn toàn cảnh về môi trường xung quanh, tạo nên một trải nghiệm chơi game đầy sống động và chân thực. Không chỉ đem lại những pha hành động mãn nhãn, các tựa game TPS còn tập trung vào việc xây dựng cốt truyện sâu sắc, phát triển nhân vật phong phú, và cơ chế chiến đấu đa dạng, tạo nên sự hấp dẫn không thể cưỡng lại. Chính nhờ những yếu tố này, game TPS không chỉ thu hút người chơi mà còn là đề tài nghiên cứu phong phú trong lĩnh vực phát triển game. Việc phát triển một tựa game hành động bắn súng theo góc nhìn thứ ba không chỉ yêu cầu sự sáng tạo trong thiết kế nhân vật và bối cảnh, mà còn đòi hỏi sự tối ưu hóa về mặt kỹ thuật để mang lại trải nghiệm mượt mà và chân thực nhất.

Với những đặc điểm nêu trên, đó là lý do em chọn đề tài này để phát triển.

**CÔNG NGHỆ SỬ DỤNG**

* Unity
* Visual Studio

**CÁC MỤC TIÊU CHÍNH**

Mục tiêu:

* Nhân vật di chuyển được theo 4 hướng và camera di chuyển theo hướng của nhân vật.
* Góc bắn của nhân vật luôn hướng vào tâm của góc nhìn.
* Xây dựng đượcc map cơ bản bao gồm:quái vật, khung cảnh, chướng ngại vật.
* Nhân vật bắn được quái vật.
* Có thể đổi được súng.
* Quái vật sẽ tăng độ khó theo từng cấp độ trò chơi.

Kỹ năng

* Sinh viên tìm hiểu kiến thức về Unity, C# và các chức năng của Unity.
* Sinh viên xây dựng thành công một game cơ bản với các chức năng đơn giản.

**KẾT QUẢ DỰ KIẾN**

Kiến thức

* Sinh viên hiểu được về những kiến thức lập trình Unity, C#.
* Sinh viên nắm được quy trình xây dựng một game bắn súng TPS.

Kỹ năng

* Báo cáo tổng hợp về Unity.
* Thu thập yêu cầu, nghiệp vụ bài toán để đưa ra các phương án giải quyết phù hợp.
* Sinh viên lập trình thành công game dựa trên nghiệp vụ bài toán bằng Unity viết bằng ngôn ngữ lập trình C# với bài toán chuyển phát nhanh.
* Xây dựng một trò chơi hợp lý và thú vị.

LỜI CAM ĐOAN

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

|  |  |
| --- | --- |
|  | **Tác giả ĐATN/KLTN**  *Chữ ký*  **Bùi Quang Hùng** |

LỜI CÁM ƠN

Báo cáo tốt nghiệp chuyên ngành Kỹ thuật phần mềm với đề tài “Xây dựng game 3D Action LockDown GunBlade bằng Unity” là kết quả của quá trình cố gắng không ngừng của bản thân và được sự khích lệ của các thầy, bạn bè và người thân. Qua trang viết này, em xin gửi lời cảm ơn tới những người đã giúp đỡ em trong thời gian học tập và nghiên cứu vừa qua.

Em xin tỏ lòng kính trọng và biết ơn sâu sắc với thầy TS.Trương Xuân Nam đã trực tiếp hướng dẫn, góp ý và cung cấp tài liệu thông tin cần thiết, giúp em hoàn thành đồ án tốt nghiệp một cách tốt nhất.

Em cũng xin chân thành cảm ơn lãnh đạo trường Đại học Thủy lợi, khoa Công nghệ thông tin và Bộ môn đã tạo điều kiện thuận lợi để em có thể học tập và rèn luyện trong suốt quãng thời gian vừa qua.

MỤC LỤC

[DANH MỤC CÁC HÌNH ẢNH xi](#_Toc171215933)

[DANH MỤC BẢNG BIỂU xii](#_Toc171215934)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ xiii](#_Toc171215935)

[MỞ ĐẦU 2](#_Toc171215936)

[1. Đặt vấn đề 2](#_Toc171215937)

[2. Bố cục của đồ án 2](#_Toc171215938)

[CHƯƠNG 1 : TỔNG QUAN VỀ NGÀNH GAME 4](#_Toc171215939)

[1.1 Lịch sử 4](#_Toc171215940)

[1.2 Các công cụ làm game 5](#_Toc171215941)

[1.3 Các ngôn ngữ làm game 6](#_Toc171215942)

[CHƯƠNG 2 : TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY 8](#_Toc171215943)

[2.1 . Unity là gì? 8](#_Toc171215944)

[2.2 . Những điểm nổi bật của Unity 8](#_Toc171215945)

[2.3 . Các thành phần trong Unity Editor 10](#_Toc171215946)

[2.3.1 . Cửa sổ Scene 10](#_Toc171215947)

[2.3.2 . Cửa sổ Hierarchy 11](#_Toc171215948)

[2.3.3 . Cửa sổ Game 11](#_Toc171215949)

[2.3.4 . Cửa sổ Project 12](#_Toc171215950)

[2.3.5 . Cửa sổ Inspector 13](#_Toc171215951)

[2.4 . Các khái niệm cơ bản trong Unity 14](#_Toc171215952)

[2.4.1 . GameObject 14](#_Toc171215953)

[2.4.2 . Component 14](#_Toc171215954)

[2.4.3 . Sprite 14](#_Toc171215955)

[2.4.4 . Animation 14](#_Toc171215956)

[2.4.5 . Key Frame 15](#_Toc171215957)

[2.4.6 . Prefabs 15](#_Toc171215958)

[2.4.7 . Sounds 15](#_Toc171215959)

[2.4.8 . Script 15](#_Toc171215960)

[2.4.9 . Scenes 16](#_Toc171215961)

[2.4.10 . Assets 16](#_Toc171215962)

[2.4.11 . Camera 16](#_Toc171215963)

[2.4.12 . Transform 17](#_Toc171215964)

[2.4.13 . Package 17](#_Toc171215965)

[2.4.14 . Console 17](#_Toc171215966)

[2.5 . Các hàm API phổ biến thường được sử dụng trong Unity 18](#_Toc171215967)

[2.5.1 . MonoBehaviour 19](#_Toc171215968)

[2.5.2 . GameObject 19](#_Toc171215969)

[2.5.3 . Transform 19](#_Toc171215970)

[2.5.4 . Component 19](#_Toc171215971)

[2.5.5 . Rigidbody 19](#_Toc171215972)

[2.5.6 . Animation 19](#_Toc171215973)

[2.5.7 . Input 20](#_Toc171215974)

[2.5.8 . Audio 20](#_Toc171215975)

[2.5.9 . Scene Manager 20](#_Toc171215976)

[2.5.10 . NavMeshAgent 20](#_Toc171215977)

[2.5.11 . Coroutine 20](#_Toc171215978)

[2.5.12 . PlayerPrefs 20](#_Toc171215979)

[2.6 . Một số Design Pattern phổ biến thường được dùng trong Unity 21](#_Toc171215980)

[2.6.1 . Object Pool Pattern 21](#_Toc171215981)

[2.6.2 . State Pattern 21](#_Toc171215982)

[2.6.3 . Singleton Pattern 23](#_Toc171215983)

[CHƯƠNG 3 : GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY 24](#_Toc171215984)

[3.1 . Các yếu tố làm nên game 24](#_Toc171215985)

[3.2 . Phân tích thiết kế game 25](#_Toc171215986)

[3.2.1 . Đồ họa 25](#_Toc171215987)

[3.2.2 . Âm thanh 25](#_Toc171215988)

[3.2.3 . Giao diện 25](#_Toc171215989)

[3.2.4 . Điều khiển 26](#_Toc171215990)

[3.2.5 . Nhiệm vụ 27](#_Toc171215991)

[3.2.6 . Kẻ thù 27](#_Toc171215992)

[3.2.7 . Màn chơi 28](#_Toc171215993)

[3.2.8 .Góc nhìn 28](#_Toc171215994)

[3.3 . Kiểm thử và bảo trì game 29](#_Toc171215995)

[CHƯƠNG 4 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG 31](#_Toc171215996)

[4.1 . Xác định tác nhân 31](#_Toc171215997)

[4.2 . Xây dựng biểu đồ Usecase 31](#_Toc171215998)

[4.3 . Danh sách các Usecase 31](#_Toc171215999)

[4.4 . Đặc tả Use-case 32](#_Toc171216000)

[4.4.1 . Use-case “Play Game” 32](#_Toc171216001)

[4.4.2 . Use-case “Select Weapon” 33](#_Toc171216002)

[4.4.3 . Use-case “Setting” 35](#_Toc171216003)

[4.4.4 . Use-case “Attack Enemy” 36](#_Toc171216004)

[4.4.5 . Use-case “Exit” 37](#_Toc171216005)

[4.5 . Cơ sở dữ liệu 37](#_Toc171216006)

[PHỤ LỤC 40](#_Toc171216007)

DANH MỤC CÁC HÌNH ẢNH

[Hình 2.1: Logo Unity 8](#_Toc171217019)

[Hình 2.2: Các nền tảng hỗ trợ Unity 9](#_Toc171217020)

[Hình 2.2: Cửa sổ Scene 10](#_Toc171217021)

[Hình 2.3: Cửa sổ Hierarchy 11](#_Toc171217022)

[Hình 2.4: Cửa sổ Game 12](#_Toc171217023)

[Hình 2.5: Cửa sổ Project 13](#_Toc171217024)

[Hình 2.6: Cửa sổ Inspector 14](#_Toc171217025)

[Hình 2.7: Animation của Unity 15](#_Toc171217026)

[Hình 2.8: Assets của Unity 16](#_Toc171217027)

[Hình 2.9: Camera của Unity 17](#_Toc171217028)

[Hình 2.10: Cửa sổ Console 18](#_Toc171217029)

[Hình 2.11: State Pattern 22](#_Toc171217030)

[Hình 3.1 Giao diện ngoài sảnh game 26](#_Toc171217031)

[Hình 3.2 Giao diện trong môi trường game 26](#_Toc171217032)

[Hình 3.3 Kẻ thù trong game 27](#_Toc171217033)

[Hình 3.4: Camera Cinemachine 29](#_Toc171217034)

[Bảng 4.1: Kịch bản Play Game 32](#_Toc171217035)

[Bảng 4.2: Kịch bản Select Weapon 33](#_Toc171217036)

[Bảng 4.3: Kịch bản Setting 35](#_Toc171217037)

[Bảng 4.3: Kịch bản Attack Enemy 36](#_Toc171217038)

[Bảng 4.3: Kịch bản Exit 37](#_Toc171217039)

DANH MỤC BẢNG BIỂU

[Bảng 4.1: Kịch bản Play Game 32](#_Toc171217412)

[Bảng 4.2: Kịch bản Select Weapon 33](#_Toc171217413)

[Bảng 4.4: Kịch bản Setting 35](#_Toc171217414)

[Bảng 4.5: Kịch bản Attack Enemy 36](#_Toc171217415)

[Bảng 4.6: Kịch bản Exit 37](#_Toc171217416)

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Từ đầy đủ** | **Nghĩa tiếng việt** |
| CSDL | Cơ sở đữ liệu |  |
| BM | Bộ môn |  |
| GVHD | Giáo viên hướng dẫn |  |
| SV | Sinh viên |  |

MỞ ĐẦU

## Đặt vấn đề

Trong những năm gần đây, ngành công nghiệp trò chơi điện tử đã chứng kiến sự phát triển vượt bậc, đặc biệt là trong lĩnh vực game hành động 3D. Với sự tiến bộ không ngừng của công nghệ đồ họa và phần cứng, các trò chơi 3D đã mang lại cho người chơi những trải nghiệm sống động và chân thực hơn bao giờ hết. Trong bối cảnh đó, việc phát triển một trò chơi hành động 3D không chỉ đáp ứng nhu cầu giải trí mà còn mở ra cơ hội kinh doanh lớn trong thị trường game đầy tiềm năng

Game hành động 3D là một thể loại mà người chơi điều khiển nhân vật từ góc nhìn phía sau, tập trung vào các cuộc chiến đấu và di chuyển linh hoạt, thể loại này thường được áp dụng trong các trò chơi có yếu tố hành động cao và kịch tính. nổi bật với khả năng cung cấp một góc nhìn rộng mở, cho phép người chơi quan sát toàn cảnh môi trường xung quanh nhân vật. Điều này không chỉ giúp người chơi định vị và điều khiển nhân vật dễ dàng hơn mà còn tăng cường khả năng phản ứng trước các tình huống bất ngờ xảy ra từ nhiều phía.

Phát triển một trò chơi hành động 3D đòi hỏi sự đầu tư lớn về thời gian, công sức. Một số thách thức chính bao gồm:

Một số thách thức trong việc phát triển game hành động 3D bao gồm:

Điều chỉnh góc quay camera để người chơi có thể theo dõi môi trường và các sự kiện xung quanh một cách hiệu quả.

Thiết kế các đối tượng và bối cảnh trong game sao cho người chơi có thể dễ dàng di chuyển và tương tác.

Tạo ra các tình huống và nhiệm vụ hấp dẫn, kích thích sự tham gia và thử thách người chơi.

## Bố cục của đồ án

Các nhiệm vụ chính của đồ án như sau:

1. Tìm hiểu về ngành game thể hiện qua chương 1.

2. Tổng quan làm game bằng Unity qua chương 2.

3. Tiến hanh làm game bắn súng bằng Unity qua chương 3;

4. Phân tích và thiết kế hệ thống qua chương 4.

5. Xây dựng và demo game qua chương 5.

6. Kết luận và hướng phát triển, chương 6.

# : TỔNG QUAN VỀ NGÀNH GAME

## Lịch sử

Ngành làm game bắt đầu hình thành từ những năm 1950, máy tính trong thời điểm này rất hiếm nên để phát triển game một trò chơi game là rất khó, một số nhà sản xuất phải bỏ rất nhiều công sức ra nghiên cứu mới có thể cho ra được một vài tựa game đơn giản thời bấy giờ.

Nimrod là chiếc máy tính chơi game đầu tiên xuất hiện trên thế giới. Chiếc máy tính này thiết kế chơi được một trò chơi game duy nhất là Nim, phát triển bởi Ferranti, một công ty nghiên cứu thiết bị điện tử có trụ sở chính tại Anh.

Bước vào thập niên 70, máy tính tên The Magnavox Odyssey được ra đời. Đây là chiếc máy console đầu tiên đưa game vào một thiết bị rời, sử dụng băng từ, để có thể chơi được người dùng phải đưa băng vào từ khe cắm của console. Nhìn thấy tiềm năng của game console, nhiều nhà sản xuất đã bắt đầu đầu tư hơn để đưa ra nhiều loạt game ấn tượng khác, có thể kể đến như Pong (1972), Space Invaders (1978)

Bước sang thập niên 80 của thế kỷ trước, chiếc máy tính cá nhân đầu tiên ra đời là chiếc máy có thể cắm với một màn hình TV. Với chiếc máy tính cá nhân đời đầu này, người chơi vừa chơi game và vừa có thể tạo ra được 1 tựa game cho chính mình bằng các công cụ lập trình.

Cuối năm 80, đầu những năm 90, chiếc đĩa CD ra đời có thể tối ưu hóa để có dung lượng lớn hơn, chứa được nhiều dữ liệu hơn và phù hợp để chứa các sản phẩm game lớn

Bước sang thập kỷ đầu của thế kỷ 21, sự phát triển của chiếc PC ngày càng mạnh dần, cùng sự phát triển của phần cứng, thì xu hướng phát triển ra một loại game có thể liên kết game thủ từ xa mà không cần phải offline. Những hệ máy chơi game hiện đại như Xbox, PlayStation 3 và Wii cung cấp cho người chơi khả năng chơi game trực tuyến và tương tác với nhau qua mạng. Những thiết bị di động như điện thoại thông minh và máy tính bảng cũng cho phép người chơi chơi game mọi lúc mọi nơi. [Những trò chơi nổi bật trong thời kỳ này bao gồm Halo, Grand Theft Auto, World of Warcraft và Angry Birds1](https://vi.wikipedia.org/wiki/L%E1%BB%8Bch_s%E1%BB%AD_video_game). Cho đến nay, ngành công nghiệp game đã trở thành một ngành công nghiệp lớn, có ảnh hưởng đến nhiều lĩnh vực khác nhau và tạo ra hàng tỉ đô la doanh thu hàng năm

## Các công cụ làm game

Trong ngành công nghiệp game, công cụ làm game (game engine) đóng vai trò quan trọng nhất trong quá trình phát triển và sản xuất game. Game engine là một nền tảng phần mềm cung cấp các công cụ và công nghệ cần thiết để tạo ra các trò chơi điện tử. Các công cụ này bao gồm đồ họa, vật lý, âm thanh, trí tuệ nhân tạo, mạng và nhiều yếu tố khác, giúp cho việc phát triển game trở nên hiệu quả và tiết kiệm thời gian hơn. Bên cạnh đó, game engine còn hỗ trợ phát triển đa nền tảng, cho phép các nhà phát triển dễ dàng đưa trò chơi của mình lên nhiều thiết bị khác nhau như PC, console, và di động. Dưới đây là một số game engine phổ biến và mạnh mẽ nhất hiện nay.

* Unity: Trong số các công cụ Phát triển Di động hàng đầu, Unity luôn được nhắc đến đầu tiên - một công cụ đa nền tảng được ra mắt bởi Unity Technologies. Công cụ này giúp bạn tạo trò chơi của riêng mình bằng các tính năng phát triển 2D và 3D, Unity sử dụng ngôn ngữ lập trình C# hoặc JavaScript để viết mã nguồn cho game
* Unreal Engine:Một công cụ làm game đa nền tảng, cho phép tạo ra các game 3D có đồ họa cao cấp cho nhiều thiết bị khác nhau như máy tính, điện thoại, máy chơi game… Unreal Engine sử dụng ngôn ngữ lập trình C++ hoặc Blueprint để viết mã nguồn cho game.
* GameMaker Studio: Một công cụ làm game đơn giản, cho phép tạo ra các game 2D cho nhiều thiết bị khác nhau như máy tính, điện thoại, máy chơi game… GameMaker Studio sử dụng ngôn ngữ lập trình GML hoặc kéo thả để viết mã nguồn cho game.
* Godot**:** là một công cụ lập trình trò chơi mã nguồn mở và miễn phí, được sử dụng để tạo ra các trò chơi 2D và 3D. Nó được phát triển bởi Godot Engine, một công ty có trụ sở tại Thụy Điển. Godot hiện hỗ trợ nhiều loại ngôn ngữ lập trình khác nhau như C#, C++,… và làm việc trên VisualScript.
* Cocos2d-x**:** là một bộ công cụ lập trình game nguồn mở được sử dụng để tạo ra các trò chơi 2D. Cocos2d-x là một bộ công cụ mạnh mẽ và linh hoạt, có thể được sử dụng để tạo ra nhiều loại trò chơi khác nhau, từ trò chơi đơn giản đến trò chơi phức tạp. Có thể sử dụng ngôn ngữ lập trình C++ hoặc Lua để viết mã nguồn cho game.

Trên đây là một số công cụ phát triển game phổ biến hiện nay, ngoài ra còn rất nhiều công cụ khác những game engine đã và đang đóng góp rất lớn vào sự phát triển của ngành công nghiệp game, mang lại cho các nhà phát triển những công cụ mạnh mẽ và linh hoạt để hiện thực hóa ý tưởng sáng tạo của mình. Việc lựa chọn game engine phù hợp sẽ giúp các nhà phát triển tối ưu hóa quy trình làm việc và tạo ra những trò chơi chất lượng cao, đáp ứng được nhu cầu ngày càng đa dạng của người chơi.

## Các ngôn ngữ làm game

Trong quá trình phát triển game, việc lựa chọn ngôn ngữ lập trình phù hợp là một yếu tố then chốt giúp tối ưu hóa hiệu suất và hiệu quả làm việc. Các ngôn ngữ lập trình cung cấp nền tảng cho việc xây dựng logic game, phát triển tính năng và tối ưu hóa trải nghiệm người chơi. Mỗi ngôn ngữ lập trình đều có những ưu điểm và nhược điểm riêng, phù hợp với các loại game và game engine khác nhau. Dưới đây là một số ngôn ngữ lập trình phổ biến nhất được sử dụng trong ngành công nghiệp game hiện nay.

* C#: Là ngôn ngữ chính được sử dụng trong Unity, C# nổi tiếng với tính dễ học và mạnh mẽ. Nó cung cấp nhiều thư viện và công cụ hỗ trợ, giúp tăng tốc quá trình phát triển game 2D và 3D.
* C++: Được sử dụng rộng rãi trong Unreal Engine và nhiều game engine khác, C++ là ngôn ngữ mạnh mẽ với hiệu suất cao, cho phép các nhà phát triển tạo ra các game với đồ họa phức tạp và yêu cầu tài nguyên hệ thống lớn.
* JavaScript: tương tự như C#, JavaScript cũng có thể làm game trên Unity nhưng so với C# thì không mạnh bằng, JavaScript cũng là một ngôn ngữ dễ học và linh hoạt, thích hợp cho việc phát triển các game nhẹ và các dự án game web.
* GML (GameMaker Language): Là ngôn ngữ lập trình được sử dụng trong GameMaker Studio, GML dễ học và cho phép các nhà phát triển nhanh chóng tạo ra các game 2D với logic phức tạp.
* Lua: Được sử dụng trong Cocos2d-x và nhiều engine khác, Lua là ngôn ngữ nhẹ và dễ tích hợp, thích hợp cho việc viết kịch bản và phát triển các tính năng game nhanh chóng.
* Python: Mặc dù không phổ biến như C++ hay C#, Python vẫn được sử dụng trong một số dự án game nhờ tính dễ học và thư viện phong phú. Python thường được sử dụng cho việc tạo prototype và công cụ hỗ trợ trong quá trình phát triển game.

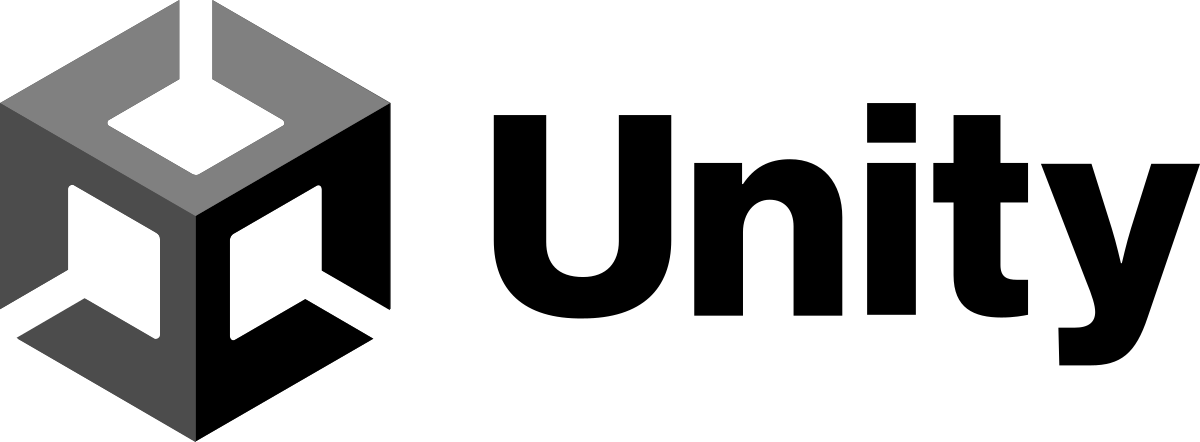
# : TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY

## . Unity là gì?

Unity là một “cross- flatform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát triển game trên PC, consoles, thiết bị di đông và trên websites.

Unity3D cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi nên cũng khá dễ sử dụng. Ngôn ngữ lập trình chính của Unity là C#, ngoài ra còn có hỗ trợ cho Javascript.

Unity cũng tận dụng chức năng của các thư viện phần mềm như engine mô phỏng vật lý PhysicX của Nvidia, OpenGL và Direct3D để kết xuất hình ảnh 3D, OpenAL cho âm thanh, … nên nó hỗ trợ rất mạnh cho công việc lập trình game

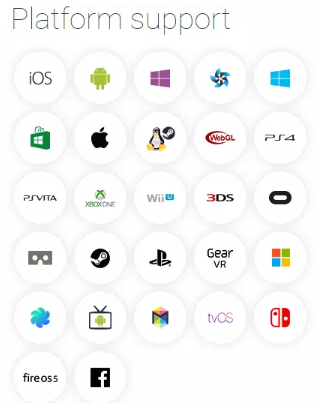


Hình 2.1: Logo Unity

## . Những điểm nổi bật của Unity

Chức năng cốt lõi đa dạng, bao gồm: cung cấp công cụ dựng hình (kết xuất đồ họa) cho các hình ảnh 2D hoặc 3D, công cụ vật lý (tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động, trí tuệ nhân tạo, phân luồng, tạo dò ng dữ liệu xử lý, quản lý bộ nhớ, dựng ảnh đồ thị và kết nối mạng. Nhờ có các engine mà công việc làm game trở nên ít tốn kém và đơn giản hơn.

Hỗ trợ đa nền tảng: Một trong các thế mạnh của Unity3D chính là khả năng hỗ trợ gần như toàn bộ các nền tảng hiện có bao gồm: PlayStation 3, Xbox 360, Wii U, iOS, Android, Windows, Blackberry 10, OS X, Linux, trình duyệt Web và cả Flash. Nói cách khác, chỉ với một gói engine, các studio có thể làm game cho bất kỳ hệ điều hành nào và dễ dàng convert chúng sang những hệ điều hành khác nhau. Đồng thời, đây cũng là giải pháp cho các game online đa nền tảng – có thể chơi đồng thời trên nhiều hệ điều hành, phần cứng khác nhau như Web, PC, Mobile, Tablet….



Hình 2.2: Các nền tảng hỗ trợ Unity

Dễ sử dụng: Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ.

Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại.

Thư viện tài nguyên và cộng đồng lớn: Unity có một thư viện tài nguyên lớn với hàng ngàn mô hình 3D, vật liệu, âm thanh và các tài nguyên khác sẵn sàng sử dụng. Điều này giúp tạo ra các dự án một cách nhanh chóng và tiết kiệm thời gian.

Unity có một cộng đồng phát triển đông đảo. Bạn có thể tìm thấy hàng ngàn tài liệu hướng dẫn, tài liệu trực tuyến và các nguồn hỗ trợ từ cộng đồng để giúp bạn giải quyết các vấn đề phát triển.

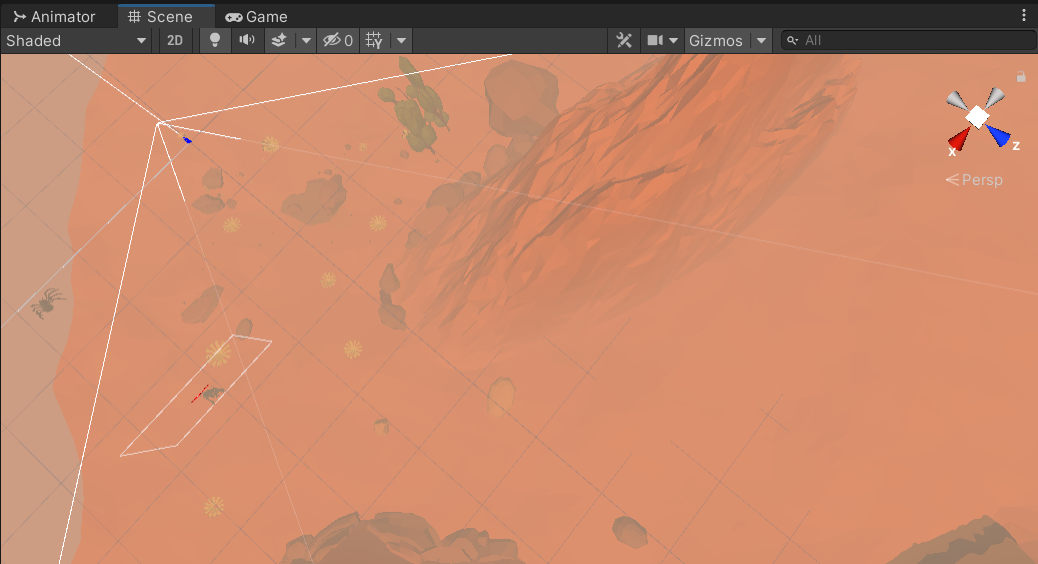
## . Các thành phần trong Unity Editor

### . Cửa sổ Scene

Phần này phần hiển thị các đối tượng trong scenes một cách trực quan, có thể lựa chọn các đối tượng, kéo thả, phóng to, thu nhỏ, xoay các đối tượng ...

Phần này có để thiết lập một số thông số như hiển thị ánh sáng, âm anh, cách nhìn 2D hay 3D ...

Khung nhìn Scene là nơi bố trí các Game Object như cây cối, cảnh quan, enemy, player, camera, … trong game. Sự bố trí hoạt cảnh là một trong những chức năng quan trọng nhất của Unity.

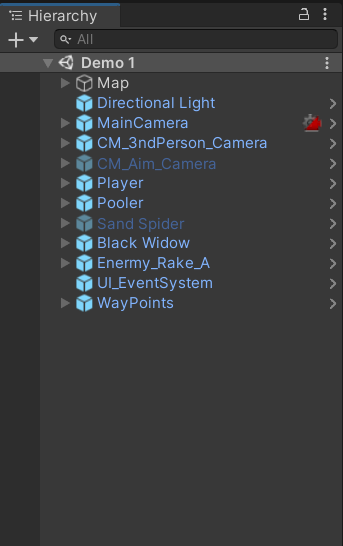


Hình 2.2: Cửa sổ Scene

### . Cửa sổ Hierarchy

Tab hierarchy là nơi hiển thị các Game Object trong Sences hiện hành. Khi các đối tượng được thêm hoặc xóa trong Sences, tương ứng với các đối tượng đó trong cửa sổ Hierarchy.

Tương tự trong tab Project, Hierarchy cũng có một thanh tìm kiếm giúp quản lý và thao tác với các Game Object hiệu quả hơn đặc biệt là với các dự án lớn.



Hình 2.3: Cửa sổ Hierarchy

### . Cửa sổ Game

Đây là màn hình demo Game, là góc nhìn từ camera trong game.

Thanh công cụ trong cửa sổ game cung cấp các tùy chỉnh về độ phân giải màn hình, thông số (stats), gizmos, tùy chọn bật tắt các component...



Hình 2.4: Cửa sổ Game

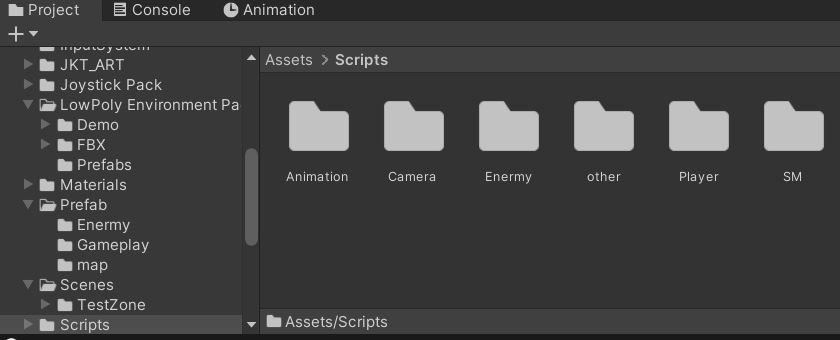
### . Cửa sổ Project

Đây là cửa sổ explorer của Unity, hiển thị thông tin của tất cả các tài nguyên (Assets) trong game của bạn.

Cột bên trái hiển thị assets và các mục yêu thích dưới dạng cây thư mục tương tự như Windows Explorer. Khi click vào một nhánh trên cây thư mục 0074hì toàn bộ nội dung của nhánh đó sẽ được hiển thị ở khung bên phải. Ta có thể tạo ra các thư mục mới bằng cách Right click -> Create -> Folder hoặc nhấn vào nút Create ở góc trên bên trái cửa sổ Project và chọn Folder. Các tài nguyên trong game cũng có thể được tạo ra bằng cách này.

Phía trên cây thư mục là mục Favorites, giúp chúng ta truy cập nhanh vào những tài nguyên thường sử dụng. Chúng ta có thể đưa các tài nguyên vào Favorites bằng thao tác kéo thả.

Đường dẫn của thư mục tài nguyên hiện tại. Chúng ta có thể dễ dàng tiếp cận các thư mục con hoặc thư mục gốc bằng cách click chuột vào mũi tên hoặc tên thư mục.



Hình 2.5: Cửa sổ Project

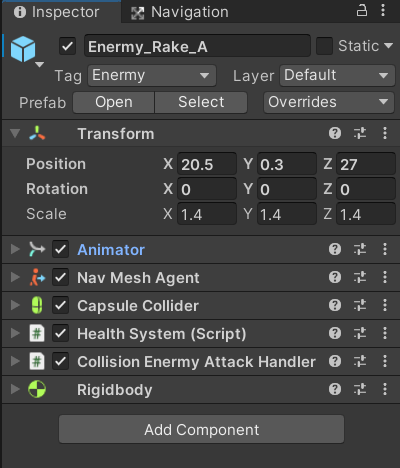
### . Cửa sổ Inspector

Cửa sổ Inspector hiển thị chi tiết các thông tin về Game Object đang làm việc, kể cả những component được đính kèm và thuộc tính của nó. Bạn có thể điều chỉnh, thiết lập mọi thông số và chức năng của Game Object thông qua cửa sổ Inspector.

Mọi thuộc tính thể hiện trong Inspector đều có thể dễ dàng tuỳ chỉnh trực tiếp mà không cần thông qua một kịch bản định trước. Tuy nhiên Scripting API cung cấp một số lượng nhiều và đầy đủ hơn do giao diện Inspector là có giới hạn.

Các thiết lập của từng component được đặt trong menu. Các bạn có thể click chuột phải, hoặc chọn icon hình bánh răng nhỏ để xuất hiện menu.

Ngoài ra Inspector cũng thể hiện mọi thông số Import Setting của asset đang làm việc như hiển thị mã nguồn của Script, các thông số animation, …



Hình 2.6: Cửa sổ Inspector

## . Các khái niệm cơ bản trong Unity

### . GameObject

Một đối tượng cụ thể trong game gọi là một game object, có thể là nhân vật, đồ vật nào đó. Ví dụ: cây cối, xe cộ, nhà cửa, người...

### . Component

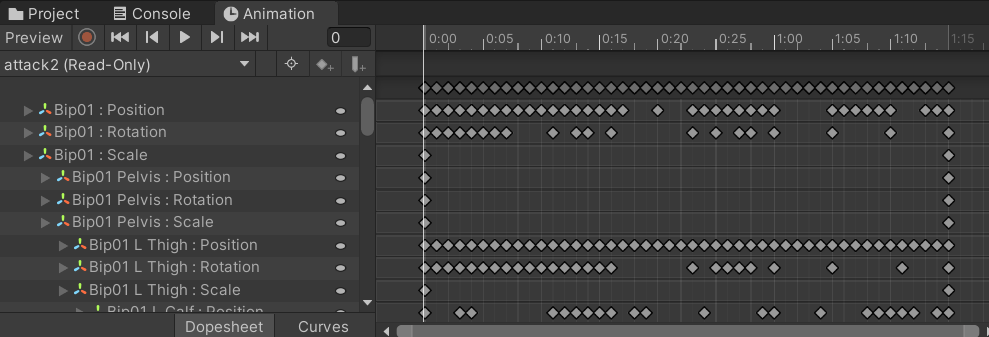
Một GameObject sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động (animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... mỗi thứ như vậy gọi là một component của GameObject.

### . Sprite

Là một hình ảnh 2D của một game object có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó.

### . Animation

Là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau



Hình 2.7: Animation của Unity

### . Key Frame

Key Frame hay Frame là một trạng thái của một animation. Có thể được tạo nên từ 1 sprite hay nhiều sprite khác nhau.

### . Prefabs

Là một khái niệm trong Unity, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay từ môt đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiên trong game Mario đều có xử lý giống nhau, nên ta chỉ việc tạo ra một đối tượng ban đầu, các đồng tiền còn lại sẽ sử dụng prefabs. Hoặc khi ta lát gạch cho một cái nền nhà, các viên gạch cũng được sử dụng là prefabs.

### . Sounds

Âm thanh trong game.

### . Script

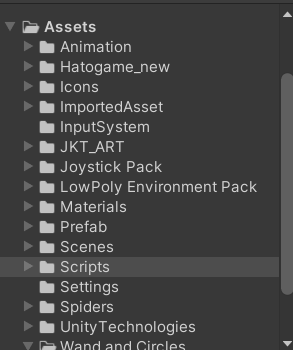
Script là tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Trong Unity có thể dùng C#, Java Script, BOO để lập trình Script.

### . Scenes

Quản lý tất cả các đối tượng trong một màn chơi của game.

### . Assets

Bao gồm tất cả những gì phục vụ cho dự án game như sprite, animation, sound, script, scenes…



Hình 2.8: Assets của Unity

### . Camera

Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game.



Hình 2.9: Camera của Unity

### . Transform

Là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng.

### . Package

Là tập hợp những game objects, assets và các file meta-data liên quan.

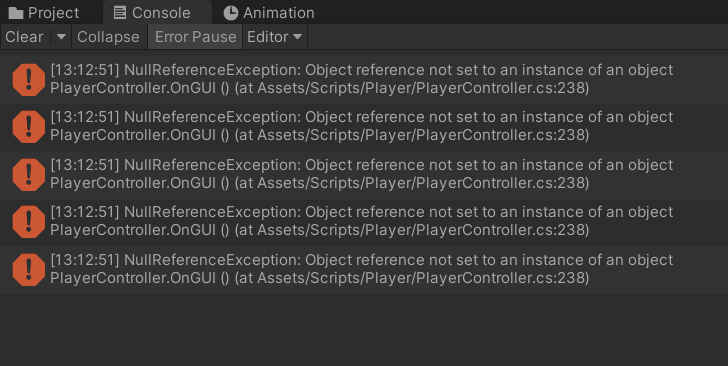
Có thể hiểu đơn giản chúng là các Object liên quan, bao gồm models, scripts, materials...

Đặc biệt, khi sử dụng Unity bạn còn được cung cấp các "standard package" tại Asset Store hoàn toàn miễn phí, bạn có thể Import trực tiếp chúng vào Project mà mình đang thực hiện dễ dàng.

### . Console

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

Ngoài ra còn giúp ta Debug để print những test case cần thiết để xem code của mình hoạt động như nào.



Hình 2.10: Cửa sổ Console

## . Các hàm API phổ biến thường được sử dụng trong Unity

Dưới đây sẽ là một số hàm API thường gặp và một số phương thức và thuộc tính thường được dùng

### . MonoBehaviour

Là lớp cơ sở cho tất cả script trong Unity, muốn gắn script vào object cần kế thừa lớp này

Start(): được gọi một lần duy nhất khi script bắt đầu kích hoạt

Update(): được gọi mỗi lần vào mỗi frame và được xem như hàm xử lý chính của vòng lặp game.

### . GameObject

Đại diện cho tất cả các đối tượng trong game

GameObject.Find (string name) : Tìm GameObject theo tên và trả về GameObject đó.

gameObject.AddComponent<T>(): Thêm một component vào GameObject

### . Transform

Quản lý vị trí, hướng và tỷ lệ của GameObject

transform.position: Truy cập hoặc thay đổi vị trí của đối tượng.

transform.rotation: Truy cập hoặc thay đổi góc quay của đối tượng.

transform.localScale: Truy cập hoặc thay đổi tỷ lệ của đối tượng.

### . Component

Là thành phần được gắn vào Gameobject

GetComponent<T>(): Lấy một component của loại T từ GameObject.

### . Rigidbody

Quản lý động lực học của các đối tượng

rigidbody.velocity: Truy cập hoặc thay đổi vận tốc của đối tượng.

rigidbody.AddForce(Vector3 force): Thêm lực vào đối tượng.

### . Animation

Quản lí hoạt ảnh của đối tượng

animator.SetBool(string name, bool value): Đặt giá trị của một tham số boolean trong Animator.

SetTrigger(string name) : Kích hoạt một trigger trong Animator.

### . Input

Quản lý đầu vào từ người chơi

Input.GetKey(KeyCode key): Kiểm tra xem phím có đang được nhấn không.

Input.GetAxis(string axisName): Truy cập giá trị của trục đầu vào.

### . Audio

Phát âm thanh trong trò chơi

audioSource.Play(): Phát âm thanh.

audioSource.Stop(): Dừng phát âm thanh.

### . Scene Manager

Quản lí các cảnh trong trò chơi

SceneManager.LoadScene(string sceneName): Tải một cảnh mới.

### . NavMeshAgent

Quản lý AI di chuyển bằng cách sử dụng lưới điều hướng (NavMesh)

navMeshAgent.SetDestination(Vector3 target): Đặt đích đến cho AI.

navMeshAgent.speed: Truy cập hoặc thay đổi tốc độ di chuyển của AI.

### . Coroutine

Cho phép thực thi mã không đồng bộ

StartCoroutine(IEnumerator routine): Bắt đầu một coroutine.

StopCoroutine(IEnumerator routine): Dừng một coroutine.

### . PlayerPrefs

Lưu và tải các thiết lập người chơi

PlayerPrefs.SetInt(string key, int value): Lưu giá trị số nguyên.

PlayerPrefs.GetInt(string key): Lấy giá trị số nguyên.

## . Một số Design Pattern phổ biến thường được dùng trong Unity

Design Pattern hay mẫu thiết kế, là cách xây dựng một lớp dựa trên một thiết kế nhất định nào đó

### . Object Pool Pattern

Trong Unity, để sinh ra một phiên bản copy của một GameObject (thường là prefab), ta sẽ sử dụng hàm Instantiate(), nếu không sử dụng bản copy đó nữa chúng ta sẽ sử dụng hàm Destroy() hay DestroyImmediate(). Hàm Instantiate() sẽ phân bổ, cấp phát tài nguyên bộ nhớ cho copied-GameObject đó, hàm Destroy() xóa bỏ các tài nguyên đã cấp phát và nếu việc này xảy ra liên tục, Garbage Collector (GC) sẽ hoạt động liên tục làm phân mảnh bộ nhớ đồng thời trong frame đó phải chờ GC xử lý xong.

Việc sinh ra và xóa đi các object liên tục như vậy sẽ ảnh hưởng rất lớn đến hiệu suất, Object Pooling Pattern sinh ra nhằm giúp chúng ta tái sử dụng được các object đã tạo

Về ý tưởng, Object Pooling sẽ sinh ra các objects mà ta cần sử dụng trong game trước, sau đó deactive toàn bộ objects đó nếu chưa sử dụng, điều này có thể làm chậm quá trình khởi động của game.

Như vậy, thay vì Instantiate() các objects vào run-time, chúng ta chỉ việc lôi chúng từ Object Pool ra rồi bật active cho object đó, tương tự, thay vì Destroy(), chúng ta sẽ deactive object đó và gửi về lại Pool.

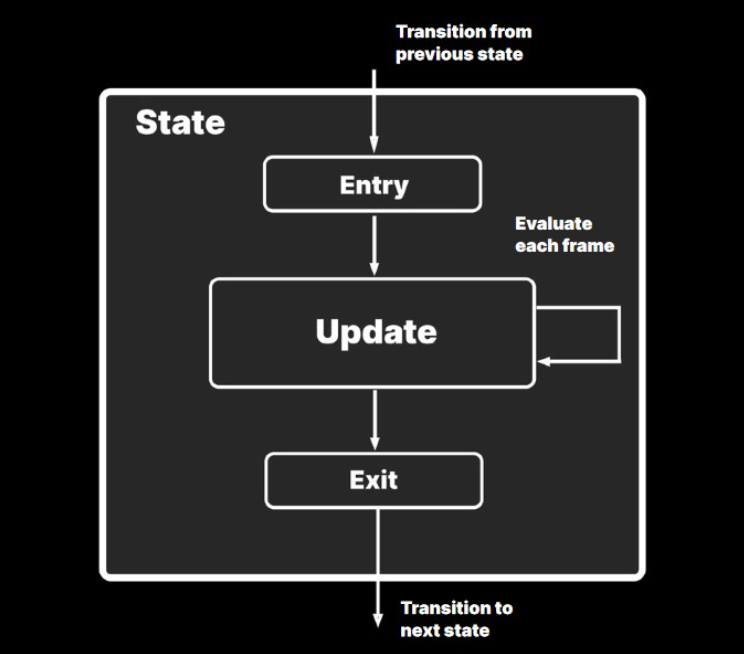
### . State Pattern

State Pattern là một mẫu thiết kế hành vi cho phép một object thay đổi hành vi của nó khi trạng thái bên trong của nó thay đổi.

State pattern giúp giải quyết hai vấn đề sau:

* Một đối tượng sẽ thay đổi hành vi của nó khi trạng thái thay đổi
* Hành vi của trạng thái cụ thể được định nghĩa một cách độc lập. Việc thêm các trạng thái mới không ảnh hưởng đến hành vi của các trạng thái hiện có.

Đây là giai đoạn của một trạng thái trong State Pattern, từ khi chuyển từ trạng thái trước đó, thực thi logic trong trạng thái hiện tại, và cuối cùng là chuyển sang trạng thái tiếp theo.



Hình 2.11: State Pattern

Mỗi trạng thái sẽ có 3 thành phần chính:

Entry : Thực thi xử lý logic khi lần đầu chuyển sang trạng thái này

Update: Thực thi logic mỗi frame( đôi khi còn được gọi là Execute hoặc Tick). Bạn có thể chia nhỏ nó thành Update, FixedUpdate hay LateUpdate giống như của MonoBehaviour. Nó cũng xử lý điều kiện chuyển sang trạng thái khác ở đây

Exit: Thực thi xử lý logic trước khi chuyển sang trạng thái khác

Điểm mấu chốt của State Pattern trong Unity là khả năng quản lý và thay đổi hành vi của đối tượng dựa trên trạng thái hiện tại một cách linh hoạt và dễ dàng mở rộng. Pattern này giúp tách biệt logic của từng trạng thái, cho phép thêm mới các trạng thái mà không làm ảnh hưởng đến các trạng thái hiện có. Việc áp dụng State Pattern trong Unity không chỉ giúp tăng tính module hóa và tái sử dụng code mà còn giảm thiểu rủi ro và đảm bảo tính nhất quán trong hành vi của đối tượng qua các trạng thái khác nhau.

### . Singleton Pattern

Singleton là một Design Pattern, nó đảm bảo mỗi class sẽ chỉ có duy nhất một instance, đồng thời cung cấp một điểm truy cập toàn cục cho Instance này. Trong Unity, Singleton cũng dùng để quản lý các đối tượng mà ta muốn khởi tạo 1 lần và dùng chung cho toàn bộ game.

Singleton có thể rất tiện lợi vì chúng cho phép kết nối các phần trong trò chơi dễ dàng hơn, miễn là biến hoặc hàm tồn tại trên singleton và có thể truy cập công khai thì các tập lệnh khác sẽ có thể sử dụng nó mà không cần bạn phải thiết lập tham chiếu đến nó trước.

Tiện lợi là thế nhưng nó được xem là một bad pattern, có một vài lí do ở Singleton pattern khiến nhiều developer không thích

* Singleton quá public. Nó yêu cầu cung cấp global access, bởi vì là các global instance nên việc sử dụng nó có thể khiên nhiều dependency bị ẩn dấu, khiến cho việc truy vết bug khó hơn.
* Singleton khiến quá trình test khó khăn. Các bài kiểm tra unit test phải độc lập với nhau nhưng singleton có thể thay đổi trạng thái của nhiều GameObject trong scene, chúng có thể ảnh hưởng đến quá trình test của bạn
* Singleton là high coupling (liên kết chặt chẽ giữa các class). high coupling khiến cho việc tái cấu trúc chở nên khó khăn, nếu bạn thay đổi một component nó có thể ảnh hưởng đến bất kỳ component nào liên kết với nó điều này dẫn đến không clean code

Vì thế đừng lạm dụng Singleton pattern quá nhiều

# : GAME 3D ACTION “LOCKDOWN GUNBLADE” BẰNG UNITY

## . Các yếu tố làm nên game

Để tạo nên một trò chơi hoàn chỉnh và hấp dẫn, phải kết hợp giữa nhiều yếu tố khác nhau. Những yếu tố này không chỉ góp phần xây dựng cấu trúc của một trò chơi mà còn nâng cao trải nghiệm của người chơi, từ đó có thể giữ chân họ lâu hơn.

Những yếu tố này bao gồm:

* Đồ họa: Đây là những hình ảnh minh họa cho các đối tượng nhân vật môi trường và hiệu ứng trong game tạo nên bối cảnh và không gian trò chơi. Đồ họa có thể thiết kế bằng các phần mềm như Photoshop, 3D Studio Max ...
* Âm thanh: Đây là yếu tố đóng vai trò tạo ra không khí và cảm xúc cho trò chơi. Âm thanh từ các đối tượng, nhân vật, môi trường và hiệu ứng trong game cần được thu âm, chỉnh sửa và phối hợp một cách tinh tế bằng các phần mềm như Audacity, Sound Forge hoặc FL Studio...
* Giao diện: Đây là yếu tố giúp người chơi dễ tương tác với trò chơi, với các thành phần hiển thị như menu, nút bấm, biểu tượng, thông báo và điểm số có nhiệm vụ cung cấp thông tin rõ ràng và các tùy chọn điều khiển cho người chơi
* Điều khiển: Liên quan đến cách thức người chơi nhập lệnh và điều khiển đối tượng, nhân vật trong game. Điều khiển có thể sử dụng các thiết bị như bàn phím, chuột, tay cầm, màn hình cảm ứng
* Nhiệm vụ: Là mục tiêu mà người chơi phải hoàn thành để tiến triển trong game. Nhiệm vụ có thể bao gồm nhiều loại như giết địch, thu thập vật phẩm hoặc khám phá. Đây là yếu tố gây nên sự hứng thú giúp giữ chân người chơi.
* Kĩ năng: Liên quan đến các khả năng đặc biệt mà nhân vật có thể sử dụng để chiến đấu hoặc giải quyết các tình huống trong game. Kỹ năng có thể bao gồm các loại như tấn công, phòng thủ, chữa trị hoặc kích hoạt các khả năng đặc biệt.
* Kẻ thù: Là khó khăn thách thức mà người chơi gặp phải trong xuyên suốt game. Kẻ thù có thể là quái vật, robot hoặc những kẻ thù khác có ý định ngăn cản gây hại cho nhân vật trong game, đây là yếu tố tạo nên sự kịch tích trong game.
* Màn chơi: Bao gồm các giai đoạn khác nhau mà người chơi phải vượt qua trong game. Mỗi màn chơi có thể có một bối cảnh, nhiệm vụ và độ khó riêng biệt. Thiết kế màn chơi cần phải phong phú và sáng tạo bởi đây là yếu tố quan trọng khiến cho người chơi có được cảm giác tò mò muốn khám phá
* Góc nhìn: Đây là yếu tố quyết định khung cảnh và góc quay khi nhân vật di chuyển trong game. Góc nhìn là góc nhìn thứ ba, giúp người chơi dễ dàng quan sát và điều khiển nhân vật.

## . Phân tích thiết kế game

Phần này sẽ trình bày chi tiết về các yếu tố thiết kế game “Lockdown GunBlade”

### . Đồ họa

Nhân vật, hiệu ứng và môi trường trong game đều được thiết kế dưới dạng đồ họa 3D. Dù là một trò chơi hành động bắn súng, màu sắc trong game vẫn rất bắt mắt và dễ chịu, giúp giảm bớt sự căng thẳng khi chơi.

### . Âm thanh

Game sử dụng âm thanh để tăng cường hiệu ứng cho các đối tượng và nhân vật, chẳng hạn như tiếng súng nổ, tiếng bước chân khi nhân vật chạy, và tiếng kêu khi nhân vật bị quái vật tấn công. Những âm thanh này giúp tạo ra một trải nghiệm sống động và chân thực hơn cho người chơi.

### . Giao diện

Game sử dụng giao diện để hiển thị cho người chơi biết những thông tin và lựa chọn có trong game như menu cài đặt, biểu tượng tạm dừng, menu tạm dừng,...

Đây là hai giao diện chính trong game



Hình 3.1 Giao diện ngoài sảnh game



Hình 3.2 Giao diện trong môi trường game

### . Điều khiển

Người chơi có thể điều khiển nhân vật bằng nút bấm trên màn hình đối với điện thoại hoặc bàn phím và chuột đối với máy tính.

### . Nhiệm vụ

Nhiệm vụ mà người chơi phải hoàn thành trong game bao gồm:

* Giết địch: Đây là nhiệm vụ chính yếu trong game, nơi người chơi sẽ sử dụng vũ khí để tiêu diệt quái vật và kẻ thù xuất hiện trong suốt hành trình
* Khám phá: Nhiệm vụ này yêu cầu người chơi phải đi khắp bản đồ, khám phá các khu vực ẩn và truy lùng kẻ địch để tiêu diệt chúng

### . Kẻ thù

Để tăng thêm phần hấp dẫn và thử thách, người chơi sẽ phải đối mặt với rất nhiều kẻ thù đa dạng và nguy hiểm trong suốt hành trình của mình trong game. Những kẻ thù này không chỉ xuất hiện với tần suất dày đặc mà còn sở hữu sức mạnh và độ khó gia tăng theo từng màn chơi.

* Quái vật: Đây là những kẻ thù cơ bản nhưng không kém phần nguy hiểm. Chúng thường xuất hiện thành từng nhóm và tấn công đồng loạt, đòi hỏi người chơi phải có khả năng di chuyển linh hoạt và tấn công chính xác.
* Boss: Mỗi màn chơi sẽ có một kẻ thù boss với sức mạnh vượt trội và kỹ năng đặc biệt. Để đánh bại được kẻ thù boss, người chơi cần phải hiểu rõ về điểm yếu của chúng và sử dụng tối đa các kỹ năng của mình.

Đây là một trong số chúng



Hình 3.3 Kẻ thù trong game

### . Màn chơi

Game có nhiều màn chơi khác nhau mà người chơi phải vượt qua, với độ khó gia tăng theo từng màn chơi. Các màn chơi được thiết kế tỉ mỉ để mang lại trải nghiệm đa dạng và đầy thử thách, đảm bảo rằng người chơi luôn phải cải thiện kỹ năng và chiến thuật của mình để tiếp tục tiến xa hơn.

Số lượng màn chơi: Có 3 màn chơi tất cả. Mỗi màn sẽ có bối cảnh và loại kẻ thù cần tiêu diệt khác nhau, tạo nên sự phong phú và đa dạng trong gameplay.

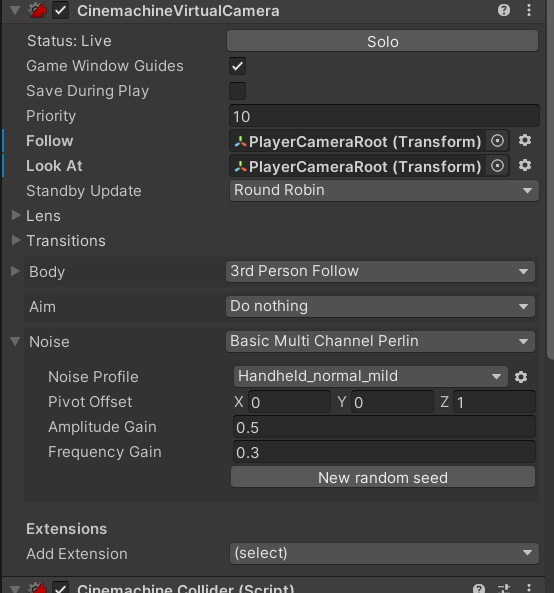
Bối cảnh: Mỗi màn chơi được đặt trong một bối cảnh riêng biệt: sa mạc, khu rừng,.... Mỗi bối cảnh không chỉ mang đến trải nghiệm thị giác mới mẻ mà còn ảnh hưởng đến chiến thuật chiến đấu của người chơi.

Loại kẻ địch: Ở mỗi màn, người chơi sẽ đối đầu với các loại quái vật khác nhau, độ khó và sức mạnh cũng khác nhau

### .Góc nhìn

Game được thiết kế với góc nhìn thứ ba, mang lại trải nghiệm trực quan và phong phú cho người chơi. Góc nhìn thứ ba giúp người chơi có cái nhìn toàn diện về nhân vật chính và môi trường xung quanh, từ đó dễ dàng nắm bắt tình hình và đưa ra các quyết định chiến thuật.

Để hiện thực hóa góc nhìn thứ ba một cách tối ưu, game sử dụng **Camera Cinemachine**, một công cụ mạnh mẽ trong Unity giúp tạo ra các chuyển động camera mượt mà và chuyên nghiệp.



Hình 3.4: Camera Cinemachine

## . Kiểm thử và bảo trì game

Phần này sẽ trình bày các phương pháp kiểm thử và bảo trì game 3D action GunBlade LockDown

Kiểm thử game : là một phần trong quá trình phát triển game, nhằm kiểm soát chất lượng của sản phẩm được tạo ra, kiểm tra và đánh giá để đảm bảo rằng nó hoạt động đúng theo thiết kế, không có lỗi (bugs) và mang lại trải nghiệm tốt nhất cho người chơi.

Có các loại kiểm thử sau:

* Người kiểm thử chức năng: tìm kiếm các vấn đề chung trong game hoặc giao diện người dùng và đồ họa của nó, chẳng hạn như các vấn đề về : logic game, vận hành game, các vấn đề về tính ổn định và tính toàn vẹn nội dung game. Kiểm tra giao diện người dùng đảm bảo sự thân thiện với người dùng.
* Kiểm thử hiệu năng: Kiểm tra hiệu suất tổng thể của game. Thực hiện điều chỉnh hiệu suất để tối ưu hóa tốc độ của game.
* Kiểm thử khả năng tương thích: Kiểm tra xem game có tương thích trên các thiết bị khác nhau hay không và tính tương thích trên các cấu hình phần cứng và phần mềm khác nhau.
* Kiểm thử giao diện người dùng (UI Testing): Đảm bảo rằng giao diện người dùng thân thiện và không có lỗi hiển thị.

Bảo trì: Là quá trình cập nhật và sửa chữa game sau khi phát hành để khắc phục các lỗi,

cải thiện hiệu năng và tăng cường tính năng của game.

Có các loại bảo trì sau:

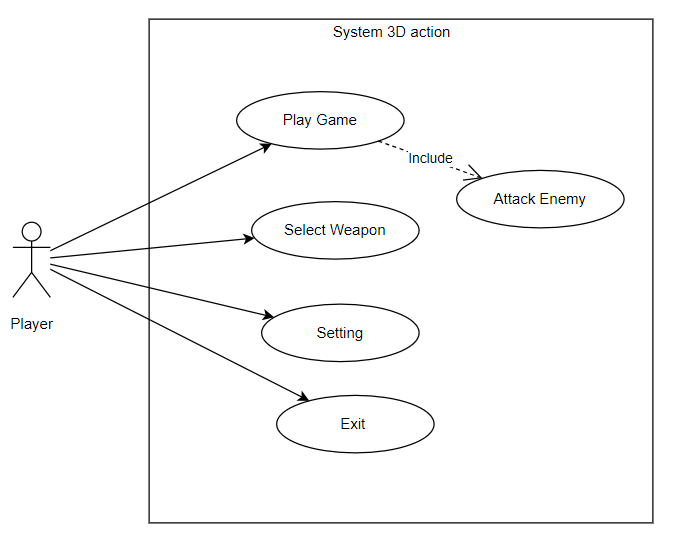
* Bảo trì sửa chữa: Là quá trình cập nhật game để sửa chữa các lỗi đã được phát hiện trong quá trình kiểm thử hoặc phản hồi từ người chơi, bao gồm việc sửa lỗi (bugs), và giải quyết các vấn đề liên quan đến hiệu suất.
* Bảo trì hoàn thiện: Đây là quá trình cải thiện và nâng cấp game dựa trên phản hồi của người chơi và những tiến bộ công nghệ mới, bao gồm việc thêm tính năng mới, cải thiện đồ họa, và tối ưu hóa gameplay.
* Bảo trì phòng ngừa: Đây là quá trình kiểm tra và bảo trì định kỳ để ngăn ngừa các vấn đề trước khi chúng xảy ra., bao gồm việc kiểm tra tối ưu hóa mã nguồn, và cập nhật hệ thống để đảm bảo game luôn chạy mượt mà.

# : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## . Xác định tác nhân

Player: Điều khiển nhân vật và tiêu diệt Enemy

## . Xây dựng biểu đồ Usecase



Hình 4.1: Mô hình UseCase tổng quát

## . Danh sách các Usecase

**Play Game:** Người chơi bắt đầu chơi game

**Select Weapon:** Chọn vũ khí đi kèm

**Setting**: Điều chỉnh hiệu ứng game (âm thanh, ...)

**Exit:** Thoát khỏi game

## . Đặc tả Use-case

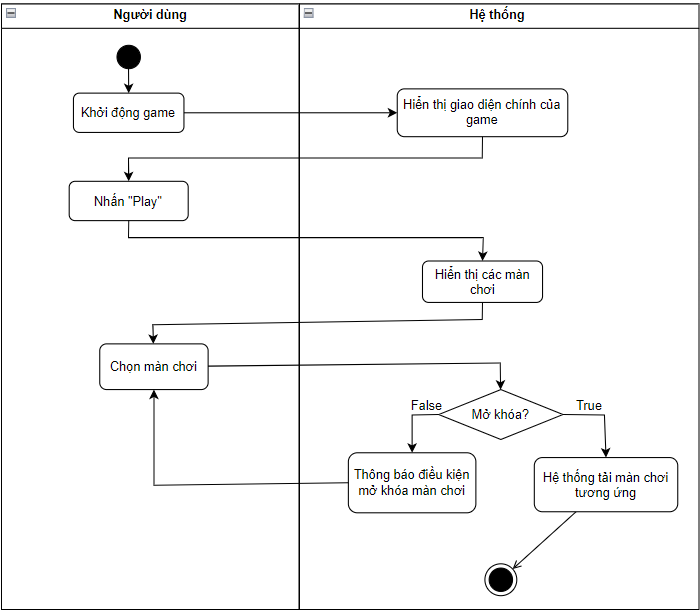
### . Use-case “Play Game”

* Kịch bản Use-case

Bảng 4.1: Kịch bản Play Game

|  |  |
| --- | --- |
| **Mô tả** | Bắt đầu chơi game |
| **Tác nhân** | Player |
| **Tiền điều kiện** | Người chơi đã khởi động game thành công và đã vào màn hình chính. |
| **Hậu điều kiện** | Người chơi đã vào được môi trường game chính, có thể di chuyển , tấn công kẻ địch ... |
| **Luồng sự kiện** | Người chơi khởi động game  Hệ thống hiển thị giao diện chính của game  Người chơi nhấn nút "Play"  Màn hình hiển thị các màn chơi  Người chơi chọn màn chơi  Hệ thống tải màn chơi tương ứng  Người chơi có thể điều khiển nhân vật và tương tác với môi trường |
| **Luồng ngoại lệ** | 1a. Lỗi không thể khởi động game  4a. Người chơi chọn màn chơi chưa được mở khóa  4a1. Hệ thống thông báo điều kiện mở khóa của màn  6a. Không thể di chuyển nhân vật và tương tác với môi trường |

* Sơ đồ Activity



Hình 4.2: Sơ đồ Activity Use Case Play game

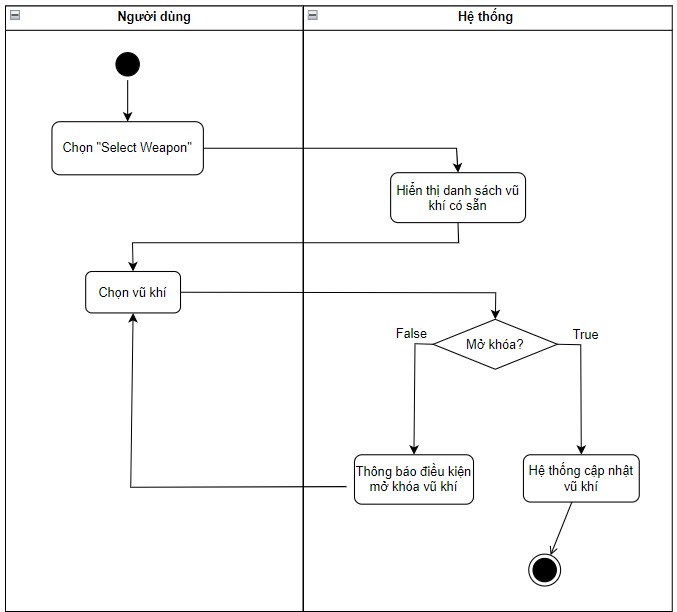
### . Use-case “Select Weapon”

* Kịch bản Use-case

Bảng 4.2: Kịch bản Select Weapon

|  |  |
| --- | --- |
| **Mô tả** | Chọn vũ khí đi kèm |
| **Tác nhân** | Player |
| **Tiền điều kiện** | Người chơi đã vào kho vũ khí  Danh sách vũ khí có sẵn được hiển thị |
| **Hậu điều kiện** | Người chơi đã chọn một vũ khí đi kèm |
| **Luồng sự kiện** | Người chơi chọn “Select Weapon” ngoài giao diện  Danh sách vũ khí có sẵn được hiển thị  Người chơi chọn một vũ khí đi kèm  Hệ thống xác nhận lựa chọn và cập nhật vũ khí cho nhân vật. |
| **Luồng ngoại lệ** | 2a. Lỗi không hiển thị danh sách vũ khí  3a. Người chơi chọn vũ khí chưa được mở khóa  3a1.Hệ thống thông báo điều kiện để mở khóa vũ khí |

* Sơ đồ Activity



Hình 4.3: Sơ đồ Activity Use Case Select Weapon

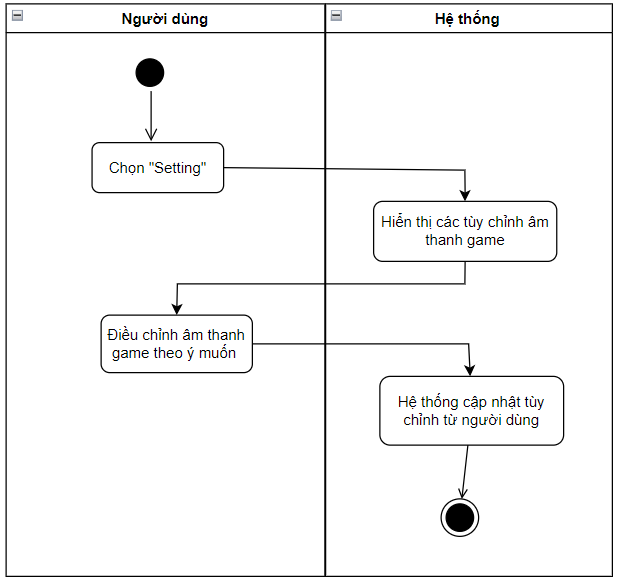
### . Use-case “Setting”

* Kịch bản Use-case

Bảng 4.4: Kịch bản Setting

|  |  |
| --- | --- |
| **Mô tả** | Cài đặt âm thanh game |
| **Tác nhân** | Player |
| **Tiền điều kiện** | Người chơi đã vào môi trường game hoặc ngoài giao diện chính của game |
| **Hậu điều kiện** | Người chơi đã điều chỉnh các cài đặt âm thanh theo ý muốn |
| **Luồng sự kiện** | Người chơi vào môi trường game hoặc ngoài giao diện chính của game  Người chơi chọn “Setting” hoặc biểu tượng hình răng cưa  Hệ thống hiển thị các tùy chọn tùy chỉnh âm thanh game  Người chơi điều chỉnh theo ý muốn  Hệ thống lưu lại các cài đặt âm thanh mới và áp dụng chúng trong game. |
| **Luồng ngoại lệ** | 5a.Lỗi tùy chọn âm thanh không áp dụng như điều chỉnh của người chơi |

* Sơ đồ Activity



Hình 4.4: Sơ đồ Activity Use Case Setting

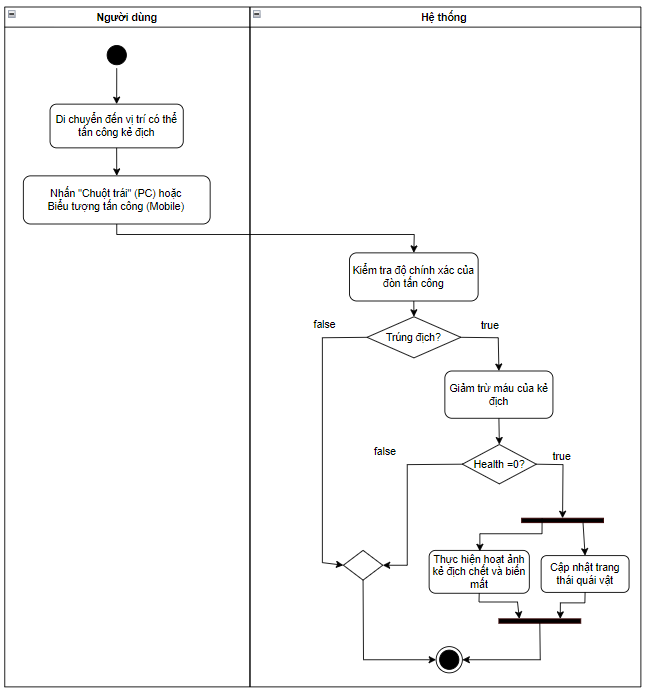
### . Use-case “Attack Enemy”

* Kịch bản Use-case

Bảng 4.5: Kịch bản Attack Enemy

|  |  |
| --- | --- |
| **Mô tả** | Tấn công kẻ thù |
| **Tác nhân** | Player |
| **Tiền điều kiện** | Người chơi đã vào môi trường game chính và nhân vật của người chơi đang ở trong tình trạng có thể tấn công (không bị kẹt, không trong trạng thái vô hiệu, v.v.). |
| **Hậu điều kiện** | Kẻ địch bị tấn công và nhận sát thương (hoặc bị tiêu diệt). |
| **Luồng sự kiện** | Người chơi di chuyển đến vị trí trong tầm tấn công  Người chơi nhấn “chuột trái” đối với máy tính hoặc biểu tượng tấn công đối với điện thoại  Hệ thống tính toán sát thương gây ra, giảm trừ lượng máu của kẻ địch |
| **Luồng ngoại lệ** | 3a. Người chơi tấn công không trúng mục tiêu  3a1. Kẻ địch không bị mất máu, hệ thống tiếp tục hoạt động bình thường  3b. Kẻ địch hết máu (bị tiêu diệt)  3b1. Hệ thống hiển thị hoạt ảnh bị tiêu diệt  3b2. Hệ thống cập nhật trạng thái kẻ địch (loại bỏ kẻ địch khỏi game) |

* Sơ đồ Activity



Hình 4.5: Sơ đồ Activity Use Case Attack Enemy

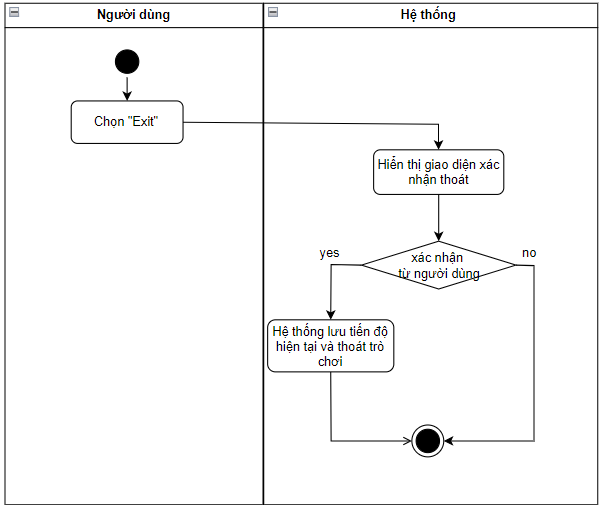
### . Use-case “Exit”

* Kịch bản Use-case

Bảng 4.6: Kịch bản Exit

|  |  |
| --- | --- |
| **Mô tả** | Thoát game |
| **Tác nhân** | Player |
| **Tiền điều kiện** | Người chơi đã vào môi trường game hoặc ngoài giao diện chính của game |
| **Hậu điều kiện** | Game được thoát ra |
| **Luồng sự kiện** | Người chơi nhấn nút “Exit” ngoài giao diện chính hoặc trong giao diện setting trong môi trường game  Trong menu, người chơi chọn tùy chọn "Thoát game"  Hệ thống hiển thị thông báo xác nhận để đảm bảo người chơi thực sự muốn thoát game.  Người chơi xác nhận lựa chọn.  Game được đóng và hệ điều hành trở về màn hình chính. |
| **Luồng ngoại lệ** | 2a. Người chơi chọn “Trở lại” thay vì thoát ra  2a1. Game không đóng, hệ thống trở về giao diện setting |

* Sơ đồ Activity



Hình 4.5: Sơ đồ Activity Use Case Exit

## . Cơ sở dữ liệu

Trong game, cơ sở dữ liệu đóng vai trò quan trọng trong việc lưu trữ và quản lý thông tin về tiến trình của người chơi. Để đơn giản hóa việc lưu trữ và truy xuất dữ liệu, game sử dụng PlayerPrefs, một công cụ tích hợp sẵn trong Unity. PlayerPrefs cho phép lưu trữ các thông tin cần thiết một cách dễ dàng và hiệu quả.

PHỤ LỤC