**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

NGUYỄN THÀNH TÍN MSSV: N17DCCN153 CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM 2017-2022 Lớp: D17CQCP02-N

**Gáy**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*: “Xây dựng Game 2D Soldier's Avenger”**

**Người hướng dẫn: Ths. Huỳnh Trung Trụ**

**Sinh viên thực hiện: Nguyễn Thành Tín**

**Mã số sinh viên: N17DCCN153**

**Lớp: D17CQCP02-N**

**Hệ** **: Chính Quy**

TP. HCM

2021

**TP. HỒ CHÍ MINH, NĂM 2021**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*:**

**“Xây dựng Game 2D Soldier's Avenger”**

**Người hướng dẫn: Ths. Phan Thanh Hy**

**Sinh viên thực hiện: Lê Gia Bảo Phú**

**Mã số sinh viên: N17DCCN131**

**Lớp: D17CQCP02-N**

**Hệ: Chính Quy**

**TP. HỒ CHÍ MINH, NĂM 2021**

BM.HV-KT.01/06

|  |  |
| --- | --- |
| HỌC VIỆN | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG | **Độc lập – Tự do – Hạnh phúc** |
| CƠ SỞ TẠI TP. HỒ CHÍ MINH |  |
| **Khoa: Công nghệ thông tin 2** | *TP. Hồ Chí Minh, ngày 21 tháng 09 năm 2021* |

ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Căn cứ Quyết định số: 402/QĐ-HVCS, ngày 21 tháng 09 năm 2021 của Phó Giám đốc Học viện – Phụ trách Cơ sở tại TP. Hồ Chí Minh về việc “phê duyệt danh sách giáo viên hướng dẫn và giao đề tài đồ án tốt nghiệp cho sinh viên Đại học chính quy Khóa 2017-2022 Ngành Công nghệ thông tin, An toàn thông tin và Công nghệ đa phương tiện”;

Khoa Công nghệ thông tin 2 giao nhiệm vụ thực hiện Đồ án tốt nghiệp cho sinh viên:

1. **Họ và tên sv : Nguyễn Thành Tín Mã SV : N17DCCN153**

**Lớp : D17CQCP02-N Khóa : 2017-2022**

**Ngành đào tạo : Công nghệ thông tin Hệ đào tạo : Đại học Chính quy**

1. **Tên đề tài tốt nghiệp: Xây dựng Game 2D Soldier's Avenger**
2. **Nội dung chính của Đồ án:**

**Lý thuyết:**

Tìm hiểu nghiệp vụ, ngôn ngữ và công nghệ sử dụng:

-Tìm hiểu và xây dựng game trên Engine Unity3D trên thiết bị Mobile và PC

-Tìm hiểu ngôn ngữ C# , JavaScript

-Tìm hiểu NodeJS, WebSocket

-Database: MongoDb

**Thực hành:**

-Thiết kế Cơ sở dữ liệu hệ thống.

-Thiết kế và xây dựng hệ thống UI : Home, Inventory, Daily Reward, List Mission, Leader Board, GameSetting , GamePlay,.....

- Xử lí đăng nhập : login , register.

- Xây dựng hệ thống cốt truyện chính.

- Xây dựng và thiết kế hệ thống Player.

- Xây dựng và thiết kế hệ thống Enemy.

- Xây dựng và thiết kế Map, Trap.

- Xây dựng và thiết kế Bonus Items.

- Xây dựng hệ thống effect chiến đấu, nhạc nền, âm thanh.

- Xây dựng GamePlay :

+ Map : xây dựng Map cho mỗi màn chơi, hệ thống bẫy tương tác với Player.

+ Player : chiến đấu với Enemy, nhận vàng, kinh nghiệm nâng cấp sức mạnh, kĩ năng, vũ khí,... qua từng màn chơi.

+ Enemy: chiến đấu với Player , xử lí AI né tránh, giữ khoảng cách, tấn công Player.

- Đưa game lên CHPlay.

1. **Cơ sở dữ liệu ban đầu:**
2. **Giáo viên hướng dẫn: ThS. Huỳnh Trung Trụ**
3. **Ngày giao đề tài: 27/09/2021**
4. **Ngày nộp quyển: 07/12/2021**

**TRƯỞNG KHOA CNTT**

**Nơi nhận:**

* *Sinh viên có tên tại khoản 1;*
* *Lưu: VP Khoa.*

**TS. Nguyễn Hồng Sơn**

# MỞ ĐẦU

Hiện nay, công nghệ thông tin đã có những bước phát triển mạnh mẽ theo cả chiều rộng và sâu.Máy tính điện tử không còn là một thứ phương tiện quý hiếm mà đang ngày càng trở thành một công cụ làm việc và giải trí thông dụng của con người, không chỉ ở công sở mà còn ngay cả trong gia đình.

Đi cùng với sự tiến hóa công nghệ, nhu cầu giải trí của con người cũng được nâng tầm phát triển, trong đó Game điện tử là một khái niệm được sinh ra và phát triển theo quá trình phát triển mạnh mẽ của máy móc điện tử và Internet.

Game điện tử trên thế giới hiện nay đã đến những bước tiến lớn nhờ công nghệ hiện đại, hướng con người đến trải nghiệm tốt nhất về tất cả các mặt: âm thanh, đồ họa, gameplay, cốt truyện…. trên nhiều nền tảng như mobile, PC, gamepad, playstation,….Trên thế giới, game điên tử đã trở nên phổ biến toàn cầu, trở thành một nền công nghiệp phát triển mạnh mẽ là nơi để đầu tư chất xám hiệu quả. Mỗi giai đoạn phát triển và mỗi nền tảng đều có những game điện tử để lại dấu mốc quan trọng:Mario, HaftLife, GTA, …

Ở Việt Nam cũng có rất nhiều doanh nghiệp cá nhân đã và đang phát triển mạnh về lĩnh vực Game điện tử, nhiều studio cũng như cá nhân đạt nhiều thành tựu nổi tiếng toàn cầu : Emobi Game với game 7554 Chiến dịch Điện Biên Phủ gây tiếng vang lớn trên cộng đồng game thế giới về cốt truyện , hoặc gần đây với cú nổ mang tính lịch sử đối với thể loại game đơn giản và vô tận mang tên Flappy Bird của lập trình viên Nguyễn Hà Đông vang danh Việt Nam trên thị trường game quốc tế. Những phát triển trên đã tạo đà và cả động lực , ý tưởng cho thế hệ mới phát triển mạnh mẽ hơn, tiếp cận với công nghệ tiên tiến trên thế giới.

Để tiếp cận và góp phần đẩy mạnh sự phổ biến của game điên tử ở Việt Nam, em đã quyết định thực hiện đề tài “**Xây dựng Game 2D Soldier's Avenger**”. Việc xây dựng game 2D trên sẽ tạo bước đà đủ lớn để tiếp cận với game điện tử về mọi mặt, phát triển kĩ năng hoàn thiện để tiến tới các mục tiêu lớn hơn trong tương lai.Đồng thời, tạo ra một game 2D thân thiện với người chơi, đơn giản cuốn hút và đặc biệt là tính giải trí cho mọi người sau những khoảng thời gian làm việc và học tập mệt mỏi.

# LỜI CẢM ƠN

Đồ án tốt nghiệp đóng một vai trò rất quan trọng và mang lại nhiều hiệu quả tích cực đối với quá trình học tập và nghiên cứu của sinh viên khi còn ngồi trên ghế giảng đường Đại Học. Đồ án tốt nghiệp giúp cho sinh viên hệ thống hóa lại kiến thức mà sinh viên đã được học, tìm hiểu và nắm bắt được những kiến thức mới cũng như đi sâu vào những vấn đề và lĩnh vực mình yêu thích.

Bản báo cáo này trình bày về quá trình xây dựng và hoàn thiện game 2D cơ bản : xậy dựng bản đồ, nhân vật , kẻ thù cũng như gameplay và các chức năng cần thiết của một game điện tử trên nền tảng PC và mobile.

Trong quá trình thực hiện không thể tránh những thiếu sót, kính mong thầy và các bạn thông cảm.

Em xin chân thành cảm ơn sự giúp đỡ nhiệt tình của thầy **Huỳnh Trung Trụ** trong quá trình em thực hiện và hoàn thành đồ án này. Đồng thời, em xin gửi lời cảm ơn chân thành đến các thầy cô trong **Khoa Công nghệ Thông tin 2** đã tạo mọi điều kiện thuận lợi để em có thể thực hiện được bài báo cáo tốt nghiệp này

**TP.Hồ Chí Minh, ngày 7 tháng 10 năm 2021**

**Sinh viên**

**Nguyễn Thành Tín**

**MỤC LỤC**

[MỞ ĐẦU 4](#_Toc87045667)

[LỜI CẢM ƠN 5](#_Toc87045668)

[DANH SÁCH CÁC BẢNG, SƠ ĐỒ, HÌNH 8](#_Toc87045669)

[KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT 9](#_Toc87045670)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 10](#_Toc87045671)

[1.1 Lý do chọn đề tài: 10](#_Toc87045672)

[1.2 Mục đích, ý nghĩa chọn đề tài: 10](#_Toc87045673)

[1.3 Phương pháp nghiên cứu: 10](#_Toc87045674)

[1.4 Phạm vi, đối tượng đề tài: 11](#_Toc87045675)

[1.5 Ý nghĩa thực tiễn đề tài 11](#_Toc87045676)

[1.6 Cấu trúc báo cáo 11](#_Toc87045677)

[CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ NỀN TẢNG XÂY DỰNG HỆ THỐNG 13](#_Toc87045678)

[2.1. Tổng quan về Game điện tử 13](#_Toc87045679)

[2.1.1. Khái niệm Game điện tử 13](#_Toc87045680)

[2.1.2. Lợi ích của Game điện tử 13](#_Toc87045681)

[2.1.3. Các yêu cầu trong một game điện tử hay 13](#_Toc87045682)

[2.2. Xây dựng hệ thống 15](#_Toc87045683)

[2.2.1 Giới thiệu Unity3D 15](#_Toc87045684)

[2.2.2 Giới thiệu Visual Studio 2019 17](#_Toc87045685)

[2.2.3. Giới thiệu về MongoDB 19](#_Toc87045686)

[2.3. Tìm hiểu về .NET 20](#_Toc87045687)

[2.3.1 Ngôn ngữ C# 20](#_Toc87045688)

[2.3.2 Tìm hiểu về .NET Framework 21](#_Toc87045689)

[2.4. Tổng quan về RestFul API 21](#_Toc87045690)

[CHƯƠNG 3: Xây dựng game 2D Soldier's Avenger 23](#_Toc87045691)

[3.1 Khảo sát hệ thống Game 2D Soldier's Avenger 23](#_Toc87045692)

[3.2 Các chức năng của hệ thống 23](#_Toc87045693)

[3.2.1 Yêu cầu chức năng hệ thống 23](#_Toc87045694)

[3.2.2 Yêu cầu phi chức năng: 23](#_Toc87045695)

[3.5 Mô hình Usecase 23](#_Toc87045696)

[3.5.1 Danh sách các Actor 23](#_Toc87045697)

[3.5.2 Xác định các usecase 23](#_Toc87045698)

[3.5.3 Biểu đồ Usecase tổng quát của hệ thống 23](#_Toc87045699)

[3.5.4 Đặc tả các usecase 24](#_Toc87045700)

[3.6 Thiết lập mô hình thực thể 24](#_Toc87045701)

[3.6.1 Xác định các thực thể 24](#_Toc87045702)

[3.6.2 Xác định mối quan hệ giữa các thực thể 24](#_Toc87045703)

[3.6.3 Sơ đồ ERD (Entity Relationship Diagram) 24](#_Toc87045704)

[3.6.4 Mô hình dữ liệu quan hệ 24](#_Toc87045705)

[3.7 Thiết kế dữ liệu 27](#_Toc87045706)

[3.7.1 Từ điển dữ liệu 27](#_Toc87045707)

[3.7.2 Mô hình diagram 27](#_Toc87045708)

[3.8 Xây dựng dự án 27](#_Toc87045709)

[3.8.1 Xây dựng BackEnd 27](#_Toc87045710)

[CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 28](#_Toc87045711)

[4.1 Kết quả 28](#_Toc87045712)

# DANH SÁCH CÁC BẢNG, SƠ ĐỒ, HÌNH

**Hình 1 Phần mềm Unity3D**

**Hình 2 Phần mềm Visual Studio 2019**

**Hình 3 Database MongoDB**

**Hình 4 Lịch sử phát triển C#**

**Hình 5 Đặc trưng ngôn ngữ C#**

**Hình 6 Tổng quan ResFul API**

**Hình 7: Core**

**Hình 8: Animation Player**

**Hình 9: Ledge Climb State**

**Hình 10A: Blend Tree Player**

**Hình 10B: Pramester BlendTree Player**

**Hình 11: AbilityState và Movement**

**Hình 12: Chuyển đổi giữa In Air State và Grounded State**

**Hình 13: In Air State sang Touching Wall State**

**Hình 14: Ledge Climb State**

**Hình 15: Animation thực thể player dùng kiếm**

**Hình 16: Animation Weapon kiếm**

**Hình17: Animation thực thể Player dùng cung**

**Hình 18: Animation cung tên**

**Hình 19: Animation thực thể Player dùng dao**

**Hình 20: Animation dao**

**Hình 21: Animation thực thể Player dùng khiên**

**Hình 22: Animation khiên**

**Hình 23: Sơ đồ hoạt động Enemy 1 Pork**

**Hình 24 : Animation Enemy 1 Pork**

**Hình 25: Sơ đồ hoạt động Enemy 2 Archers**

**Hình 26: Animation Enemy 2 Archers**

**Hình 27: Sơ đồ hoạt động Enemy 3 Wizard**

**Hình 28: Animation Enemy 3 Wizard**

**Hình 29: Sơ đồ hoạt động Enemy 4 Goblin**

**Hình 30: Animation Enemy 4 Goblin**

**Hình 31: Sơ đồ hoạt động Enemy 5 Skeleton**

**Hình 32: Animation Enemy 5 Skeleton**

# KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
|  | |
| CSDL hoặc DB | Cơ sở dữ liệu: DataBase |
| NSD = User = Account | Quyền làm việc được cấp cho NGƯỜI SỬ DỤNG trên hệ thống App, quyền này thông thường được đảm bảo bởi tên đăng nhập (username) và mật khẩu (password); cũng có thể gọi là “tài khoản người dùng” [“tài khoản”] |
|  |  |

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài:

Hiện nay, Game điện tử đối với nhân loại là một món ăn tinh thần không thể thiếu với mọi lứa tuổi, mọi giới tính,…Game điện tử trở thành nơi giải trí hiệu quả sau những giờ học tập và làm việc căng thẳng, nơi để mọi người giao lưu, kết bạn, giải trí, rèn luyện và làm mọi việc mà trên thực tế chúng ta chưa đủ kinh tế hoặc thời gian dài để thực hiện.Đối với lập trình viên, đây là mảnh đất màu mỡ để đầu tư chất xám hiệu quả nhất, nơi ta có thể xây dựng mọi thứ bằng trí tưởng tượng của chúng ta và các dòng code.

Việc chọn đề tài trên cũng là bước đầu tiên trên con đường tiếp cận các công nghệ làm game cũng như hướng tới việc xây dựng thế giới ảo trong Game.Việc xây dựng một game là xây dựng một thế giới ảo với những quy luật mà do chính chúng ta, game’s master đề ra, nói đơn giản, chúng ta xây dựng game chính là xây dựng 1 thế giới mơ ước của chính chúng ta.

## 1.2 Mục đích, ý nghĩa chọn đề tài:

* Xây dựng được một thế giới game 2D với những quy luật, nơi mà chúng ta vào vai một người lính tiến vào công cuộc trả thù những con quái vật và cả những kẻ thù ác.
* Xây dựng một gameplay thân thiện, người chơi sẽ được giải trí và giải đố, phát triển sự nhanh nhạy trong việc xử lí thông tin, phát triển IQ trong quá trình chơi game..
* Là bước đà để tiếp cận công nghệ thế giới.
* Giúp làm quen với việc xây dựng game hướng tới xây dựng một thế giới game rộng lớn.

## 1.3 Phương pháp nghiên cứu:

Tìm hiểu “**Xây dựng Game 2D Soldier's Avenger**”

* Khảo sát hệ thống

- Khảo sát các game có nội dung hoặc gameplay tương tự với phong cách phiêu lưu chiến đấu và cả giải đố ví dụ: Mario,…

* Phân tích chức năng hệ thống

- Từ quá trình khảo sát, phân tích và đưa ra bảng các chức năng chính của hệ thống và phân tích chi tiết từng chức năng

* Phân tích thiết kế với UML

- Vẽ và phân tích biểu đồ User Case

- Biểu đồ tuần tự

- Biểu đồ lớp

- Biểu đồ hoạt động

* Phân tích thiết kế CSDL
* Phân tích thiết kế chương trình

**a. Lý Thuyết:**

Tìm hiểu nghiệp vụ, ngôn ngữ và công nghệ sử dụng:

* Tìm hiểu và xây dựng game trên Engine Unity3D trên thiết bị Mobile và PC
* Tìm hiểu ngôn ngữ C# , JavaScript
* Tìm hiểu NodeJS, WebSocket
* Database: MongoDb

**b. Thực Hành:**

* Thiết kế Cơ sở dữ liệu hệ thống.
* Thiết kế và xây dựng hệ thống UI : Home, Inventory, Daily Reward, List Mission, Leader Board, GameSetting , GamePlay,.....
* Xử lí đăng nhập : login , register.
* Xây dựng hệ thống cốt truyện chính.
* Xây dựng và thiết kế hệ thống Player.
* Xây dựng và thiết kế hệ thống Enemy.
* Xây dựng và thiết kế Map, Trap.
* Xây dựng và thiết kế Bonus Items.
* Xây dựng hệ thống effect chiến đấu, nhạc nền, âm thanh.
* Xây dựng GamePlay :
* Map : xây dựng Map cho mỗi màn chơi, hệ thống bẫy tương tác với Player.
* Player : chiến đấu với Enemy, nhận vàng, kinh nghiệm nâng cấp sức mạnh, kĩ năng, vũ khí,... qua từng màn chơi.
* Enemy: chiến đấu với Player , xử lí AI né tránh, giữ khoảng cách, tấn công Player.
* Đưa game lên CHPlay.

**c. Công cụ hỗ trợ:**

* Visual Studio Code
* Unity3D

## 1.4 Phạm vi, đối tượng đề tài:

* Độ tuổi: 10+ (trên 10 tuổi, đủ nhận thức về trò chơi và giải đố trong trò chơi)
* Giới tính: phù hợp với tất cả giới tính.
* Độ phủ sóng: game sẽ được đưa lên các cửa hàng điện tử có thể tiếp cận dễ dàng với người chơi thông qua internet.
* Thiết bị: game đơn giản phù hợp với mọi loại thiết bị trên nhiều nền tảng.
* Nền tảng : PC và Mobile : điều khiển gamepad và keyboard
* Thể loại: game Phiêu lưu , giải đố phù hợp với nhu cầu giải trí.

## 1.5 Ý nghĩa thực tiễn đề tài

* Game giúp người chơi hóa thân vào nhân vật mô phỏng, tham gia các trận chiến và giải đố, chiến đấu giúp người chơi giải trí cũng như tăng cường khả năng xử lí tình huống, phát triển IQ.
* Việc xây dựng game sẽ giúp ta tiếp cận và học nhiều kĩ năng trong việc xây dựng thế giới game trong tương lai.

## 1.6 Cấu trúc báo cáo

Phần 1: Mở đầu

Phần 2: Nội dung

- Chương I: Tổng quan đề tài

- Chương II: Cơ sở lý luận và nền tảng xây dựng hệ thống

- Chương III: Phân tích và thiết kế hệ thống

- Chương IV: Xây dựng GamePlay

- Chương V: Đánh giá kết quả và hướng phát triển đề tài

Phần 3: Tài liệu tham khảo

# CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ NỀN TẢNG XÂY DỰNG HỆ THỐNG

## 2.1. Tổng quan về Game điện tử

### 2.1.1. Khái niệm Game điện tử

Trò chơi điện tử là trò chơi sử dụng thiết bị điện tử để tạo ra một hệ thống tương tác với người chơi nhằm mục đích chính là giải trí.

Game điện tử phải được hiểu đầy đủ là bao gồm: Các trò chơi dùng điện báo đánh chữ (Teletype games), Máy chơi game cầm tay (Electronic handhelds), Máy bắn bi và các thiết bị tương tự (Pinball machines and similar devices), Trò chơi đổi thưởng (Redemption games), Máy đánh bạc (Slot machines), Trò chơi âm thanh (Audio games) và Trò chơi video (Video games).

Đề tài trên đi vào xây dựng video game, thuộc Game điện tử cũng chính là xu hướng hiện đại phát triển nhất của Game điện tử.

### 2.1.2. Lợi ích của Game điện tử

**2.1.2.1. Ngành dịch vụ đang hot nhất trên thị trường**

Game điện tử nhất là video game hiện tại đang phát triển mạnh mẽ do sự phát triển của công nghệ cùng với nhu cầu giải trí càng cao của con người. Do đó đây luôn là môi trường màu mỡ cho việc đầu tư chất xám cũng như lợi nhuận đáng kể do ngành dịch vụ này mang lại do có thị trường người dùng trải dài trên mọi lứa tuổi, mọi giới tính, mọi nơi…

Người chơi còn có thể đầu tư vào game, buôn bán vật phẩm giá trị trong game để tạo thu nhập, từ đó đem lại lợi nhuận cho cả những người chơi giỏi.

**2.1.2.2. Tính giải trí**

Điều tiên quyết trong một game điện tử là tính giải trí. Một game điện tử phát triển được phải đảm bảo được nhu cầu giải trí của con người. Trong thời đại ngày nay, nhu cầu giải trí ngày càng cao cũng như chất lượng phải đảm bảo,do đó các game điện tử phải đảm bảo tính giải trí phù hợp với thời đại để tránh bị lạc hậu, lãng quên.

**2.1.2.3. Phát triển bản thân người chơi**

Game điện tử không chỉ đơn giản để giải trí, mà còn đảm bảo tính tò mò, thích thú và thử thách người chơi, và để hoàn thành game, người chơi phải hoàn thiện khả năng phản xạ, tính logic và khả năng giải đố tùy thuộc vào thể loại game. Do đó game nâng cao khả năng của người chơi, tăng cường IQ, luyện phản xạ cho tay, mắt…

Một số game ra đời nhằm giúp đỡ trong cả ngành y học, trong vật lí trị liệu giúp đỡ bệnh nhân đi đứng, lấy lại cảm giác từ các giác quan qua các game thực tế ảo hoặc các game phản xạ cơ bản.

**2.1.2.4. Tính xã hội**

Ngày nay với sự phát triển của internet đã tạo nên một cộng đồng game lớn mạnh trên toàn thế giới, các hội nhóm game hoặc các tổ chức ngày càng lón được tạo ra, xây dựng các giải đấu dành cho những game thủ chuyên nghiệp. Game điện tử trở thành môn thể thao điện tử không thể thiếu trong các kì vận hội trong tương lai.

Game online giúp các cá nhân tương tác với nhau như một xã hội thu nhỏ, giúp kết bạn mới hoặc kết hợp chơi với nhau tạo nên tính đoàn kết trong cộng đồng.

### 2.1.3. Các yêu cầu trong một game điện tử hay

Mỗi trò chơi đều có những đặc tính riêng tạo nên sự khác biệt so với những trò chơi khác. Nhưng đặc biệt một số game lại sở hữu các “quyền năng” vô hình nào đó khiến game thủ không thể nào dứt ra khỏi mà cứ chơi mãi ngày này qua ngày khác, trong khi có những game lại bị bỏ xó không thương tiếc, chờ ngày gỡ cài đặt.

Dưới đây là các yêu cầu của 1 game điện tử hay :

**2.1.3.1.Các nấc thang trình độ**

Đây có lẽ là khía cạnh quan trọng nhất làm nên sự hấp dẫn nhất của một trò chơi. Điểm cốt lõi của eSports và competitive games chính là sự cải thiện liên tục, thay đổi liên tục, đòi hỏi người chơi phải luôn luôn nâng cao kỹ năng của bản thân, tạo nên những thách thức để người chơi vượt qua, từ đó làm nên tính cạnh tranh và lôi cuốn của trò chơi.Dù vẫn có những tựa game cực kỳ khó như Mega man Classic hoặc game chiến thuật nhóm tuyệt hay như MOBA, nhưng tựu chung thì cách phổ biến nhất để lôi cuốn người chơi đó là cung cấp các cách thức rõ ràng giúp họ nâng cao trình độ. Nếu trong game không có tính năng đấu PvP (Player versus Player - người chơi đấu với người chơi) thì hãy sử dụng leaderboard (bảng xếp hạng), giúp người chơi có thể dễ dàng so sánh mình với những người chơi khác. Thêm vào đó, hãy nhớ bao gồm cả tính năng thử thách hàng ngày (daily challenges) để tạo thành thói quen cho người chơi.

**2.1.3.2.** **Cốt truyện hấp dẫn**

Khi được yêu cầu đưa ra ví dụ về một game có cốt truyện hấp dẫn, nhiều người sẽ nghĩ ngay đến những tựa game nặng tính kể chuyện như game JRPG (Japanese Role Playing Game - thể loại game nhập vai Nhật Bản) hay những game phương Tây có chế độ một người chơi (single player). Tuy nhiên, hiếm có khi nào mà một trò chơi lại bị “dở đi” bởi các yếu tố kể chuyện.Các yếu tố thuộc về kiến thức cũng khá cần thiết, thậm chí những game eSports như Liên minh huyền thoại(LMHT) cũng đã giải quyết được vấn đề này. Mỗi lượt chơi có thể không liên quan nhiều với nhau nhưng bằng việc cùng nhau giải quyết các vấn đề trong thế giới giả tưởng của game, LMHT đã tạo được sự liên kết và cộng hưởng giữa nhiều game thủ trong game. Xây dựng thế giới trong game và tạo nên tình huống truyện hấp dẫn không phải là những cách duy nhất để thỏa mãn yêu cầu của người chơi về mặt cốt truyện. Nhiều tình tiết trong game MMO(Massively Multiplayer Online) hiện nay còn hấp dẫn hơn cả câu chuyện gốc do những nhà văn chuyên nghiệp viết cho các nhà phát triển game. Các yếu tố trên được đưa vào trong mỗi trò chơi thông qua những hình ảnh trực quan, khơi gợi cảm xúc từ phía người chơi và trở thành câu chuyện riêng của mỗi người.

**2.1.3.3. Khả năng thành thục**

Hầu hết thời điểm chúng ta vui vẻ nhất khi chơi một trò chơi đó là khi chúng ta đã tương đối thành thạo nó và có thể chơi tự nhiên mà không cần phải suy nghĩ nhiều. Không cần phải “cày” cật lực để kiếm XP hay khám phá những hang động trong Spelunky đến cả nghìn lần, đôi khi niềm vui đơn giản đến từ những lúc CHỈ chơi game mà thôi.Điều này có thể áp dụng như nhau ở cả dòng game phổ thông và core game. Đôi khi sự mệt mỏi của những chuyến bay dài có thể bị đánh tan bởi những đường chạy gần như vô tận (endless run) của Banana kong hay Temple run. Những game này vẫn đòi hỏi người chơi có kỹ năng nhưng chúng gần như đem lại trạng thái thư giãn tuyệt đối, giúp chúng ta quên đi quãng thời gian dài dằng dặc và sự mệt mỏi của những chuyến bay.

**2.1.3.4. Sự bí ẩn**

Mong muốn tìm được câu trả lời cho những câu hỏi kéo dài không có lời giải đáp hoặc biết cho bằng được kết thúc của một chuyện gì đó hẳn là tác nhân mạnh mẽ nhất khiến một người kiên nhẫn theo dõi câu chuyện cho đến cuối cùng. Đôi khi chúng ta cứ nấn ná xem cho bằng hết một bộ phim chỉ vì muốn biết kết thúc của nó ra sao dù cho phim dở tệ. Game cũng vậy, dù cho rất hay chăng nữa nhưng nếu trong game không có những tình tiết bí mật nhằm kích thích trí tò mò của người chơi thì cũng khó mà giữ chân họ được lâu.

**2.1.3.5. Giảm bớt/Tối thiểu hóa hình phạt khi người chơi mất mạng trong game**

Trong nhiều game, sau khi bị “chết”, người chơi đã quen với việc có những hình đồ họa sặc sỡ màu mè thể hiện những câu kiểu như “Hết mạng. Đi lối này!” như thể không có chúng thì chúng ta không biết đường để... thoát game vậy. Thực ra không nhất thiết phải buộc người chơi rời khỏi game mỗi khi mất mạng như thế. Thay vì như vậy các nhà phát triển game nên áp dụng những cách thức nhẹ nhàng và tế nhị hơn để giảm thiểu hình phạt cho việc “chết” trong game, có thể chỉ đơn giản là một menu được thiết kế hợp lý, hoặc đến cách thức phức tạp hơn như là hệ thống tính năng autosave. Game Buff knight của studio Hàn Quốc Buff đã thực hiện điều này rất tốt khi dẫn dắt người chơi đến cửa hàng nhằm upgrade nhân vật của mình sau khi vừa bị mất mạng.

**2.1.3.6. Tính mới lạ**

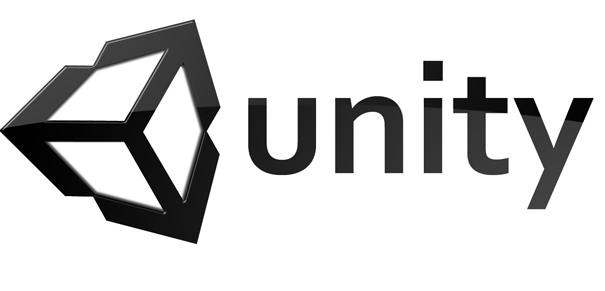
Chắc chắn là người chơi nào cũng sẽ có cảm giác thích thú phấn khích khi được chơi một trò chơi "không giống ai". Khi Flappy bird mới ra lò, rất nhiều người chơi nhanh chóng trở nên hào hứng khi chưa bao giờ gặp trò chơi nào khó như vậy!Đồ họa đẹp mắt, thế giới trong game sinh động mới lạ, nhân vật kì quặc hay cốt truyện bất ngờ v.v. hãy cố gắng tạo ra bất cứ thứ gì có thể khiến cho trò chơi của bạn khác lạ so với những game đã xuất hiện nhan nhản trên thị trường, chỉ có thế thì mới khiến người chơi có hứng thú được. Và tất nhiên sự mới mẻ này cũng chỉ tồn tại được trong một thời gian ngắn mà thôi, vì chỉ cần có game nào trở nên thành công một tẹo thì sẽ nhanh chóng có một lô một lốc những game ăn theo nó ngay, đơn cử như Flappy bird!

**2.1.3.7. Thử thách trí tuệ**

Người chơi luôn thích được thử thách, có thể chỉ đơn giản là một câu đố, hoặc một thử thách phức tạp cần phải “động não” nhiều hơn. Các thể loại thử thách trí tuệ trải dài từ chọn cánh cửa nào để mở trong các survival horror game cho đến việc hoàn thành một cuộc bố ráp như trong các game MMO. Cảm giác giải mã được câu đố hay tìm ra được những bước thích hợp để hoàn thành nhiệm vụ quả thật khiến người chơi thấy thích thú và thỏa mãn. Những nhà phát triển game Nhật Bản từ lâu đã khai thác yếu tố này một cách rất hiệu quả, với series Level 5’s Professor Layton và nhiều game khác trong dòng game survival horror.

## 2.2. Xây dựng hệ thống

### 2.2.1 Giới thiệu Unity3D

****

***Hình 1 Phần mềm Unity3D***

(Nguồn: *https://teky.edu.vn/blog/unity-3d-la-gi*/)

Hiện nay, với 27 nền tảng phát triển liên tục 15 năm (từ năm 2005), Unity đang được đánh giá là một trong những phần mềm đa nền tảng mạnh mẽ nhất cho những lập trình viên game.

Unity hỗ trợ được hầu hết trên các nền tảng từ IOS, Android, hay Windowa, MacOs,…, sử dụng các ngôn ngữ lập trình C# và Javascript. Điều đó đã tạo nên thế mạnh để Unity trở thành chìa khóa để phát triển phần mềm game tốt nhất hiện nay.

Ngoài ra, việc cho phép lập trình game theo thời gian thực của Unity đã cho phép các Designer có thêm nhiều lợi thế:

* Không cần Render.
* Thiết kế được Map, Character, Interface Graphic,..từ những phần mềm khác (như Modo, Lightwave,…)
* Tối ưu hóa định dạng file theo dạng “.FBX” hay “.exe”. Vì vậy, thiết kế có thể dễ dàng chia sẻ trên các thiết bị khác.

Bên cạnh đó, Unity có khả năng cung cấp các tính năng tích hợp quan trọng. Lập trình với Unity ngày càng trở nên đơn giản với các công cụ:

* Cho phép lập trình viên có thể chia sẻ các bản thiết kế.
* Cho phép sử dụng các tài nguyên hiệu ứng có sẵn trên Unity. Hoặc bạn có thể mua hoặc bán sản phẩm cho story của Unity.
* Cung cấp hệ thống toàn diện, dễ sử dụng cho lập trình viên.

Được coi là chìa khóa để phát triển phần mềm game, Unity 3D sở hữu nhiều ưu điểm.

* Về quy mô sử dụng phần mềm.

Theo một số thống kê, hiện nay có tới 47% số các nhà phát triển game sử dụng Unity 3D. Nó đang được coi là nền tảng thiết kế game phổ biến nhất trên toàn thế giới.

Cộng đồng Unity rất lớn mạnh. Với quy mô trên toàn thế giới, cộng đồng designer được thành lập ở nhiều nơi. Trong đó chia sẻ các kiến thức và kinh nghiệm về Unity 3D. Nhờ đó, kho sưu tập tài nguyên hiệu ứng có thể dử dụng trong Unity 3D ngày càng trở nên phong phú, đa dạng,…

* Chi phí sử dụng phần mềm.

Chính sách về chi phí của Unity được nới lỏng đến mức tối đa. Phần mềm được sử dụng miễn phí cho các cá nhân và doanh nghiệp có doanh thu thấp. Nếu muốn nâng cấp và sử dụng bản Pro, chi phí phải bỏ ra cũng rất nhỏ. Chỉ từ 1.500 USD cho một năm sử dụng Unity 3D.

* Cách sử dụng phần mềm đơn giản.

Là một phần mềm lập trình đa nền tảng, có thể dử dụng trên nhiều thiết bị khác nhau. Unity 3D cung cấp một hệ thống toàn diện cho cả nhiệm vụ soạn thảo và sửa lỗi. Điều này giúp phần mềm lập trình này có thể tối ưu hóa mọi công việc cho lập trình viên. Vì vậy, Unity 3D đảm bảo dễ sử dụng cho cả newbie.

* Nền tảng được phát triển liên tục, không ngừng nâng cấp phần mềm.

Phát triển Unity trong vòng 15 năm (từ năm 2015 đến nay). Unity Technologies đã liên tục cho ra đời các phiên bản khác nhau của Unity. Các phiên bản không ngừng được chú trọng phát triển tạo được những tính năng ấn tượng.

### 2.2.2 Giới thiệu Visual Studio 2019



**Hình 2 Phần mềm Visual Studio 2019**

(Nguồn: *https://lapcamerahanoi.com/key-visual-studio-2019/)*

* Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft.
* Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu.
* Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI , VB.NET , C# và F#. Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS
* Một số tính năng Visual Studio Code:
* Cải thiện trải nghiệm tìm kiếm nâng cao

Một trong những tính năng hữu ích trong Visual Studio 2019 là hộp tìm kiếm được cập nhật. Trong một phiên bản trước, nó được gọi là Khởi động nhanh. Trải nghiệm tìm kiếm mới trong Visual Studio 2019 nhanh hơn và hoạt động hiệu quả để có kết quả tốt nhất.

Bây giờ kết quả tìm kiếm sẽ hiển thị các đề xuất khi bạn nhập vào hộp tìm kiếm và nó sẽ tự động trình bày kết quả. Bạn cũng có thể sử dụng các phím tắt trong khi tìm kiếm và điều này sẽ giúp bạn nhớ các truy vấn tìm kiếm để sử dụng trong tương lai và nhớ lại bất cứ khi nào bạn xây dựng các ứng dụng web mới.

* Tái cấu trúc được cải thiện

Tái cấu trúc C # mới giúp tổ chức mã của bạn dễ dàng hơn. Chỉ cần gọi các phép tái cấu trúc bằng cách nhấn Ctrl +. và chọn hành động bạn muốn thực hiện.

* IntelliCode tốt hơn

Visual Studio IntelliCode là một phần mở rộng giúp tăng cường nỗ lực phát triển phần mềm của bạn bằng cách sử dụng trí tuệ nhân tạo (AI).

Nó cung cấp các đề xuất và các đối tượng mã hóa trong quá trình phát triển. Intellicode sẽ đạt được sức mạnh khi sử dụng 2000 dự án nguồn mở khác Quy trình làm việc trên nền tảng đám mây

Cửa sổ bắt đầu trình bày cho bạn một số tùy chọn để giúp bạn mã hóa nhanh chóng. Trước tiên, chúng tôi đã đặt tùy chọn để sao chép hoặc kiểm tra mã từ repo.

* Chia sẻ trực tiếp

Visual Studio Live Share là một dịch vụ dành cho nhà phát triển cho phép bạn chia sẻ một cơ sở mã và bối cảnh của nó với đồng đội và nhận được sự cộng tác hai chiều ngay lập tức từ bên trong Visual Studio. Với Live Share, đồng đội có thể đọc, điều hướng, chỉnh sửa và gỡ lỗi một dự án mà bạn đã chia sẻ với họ và thực hiện một cách liền mạch và an toàn.

* Đánh giá mã tích hợp

Chúng tôi đang giới thiệu một tiện ích mở rộng mới mà bạn có thể tải xuống để sử dụng với Visual Studio 2019. Với tiện ích mở rộng mới này, bạn có thể xem lại, chạy và thậm chí gỡ lỗi các yêu cầu kéo từ nhóm của mình mà không cần rời khỏi Visual Studio. Chúng tôi hỗ trợ mã trong cả kho GitHub và Azure DevOps.

* Tính năng tìm kiếm trong khi gỡ lỗi

Bạn đã biết rằng gỡ lỗi là quá trình xác định lỗi và loại bỏ các lỗi đó trong phần mềm. Tìm kiếm các đối tượng và các giá trị phù hợp trong quá trình gỡ lỗi rất hữu ích cho các nhà phát triển.

Trong Visual Studio 2019 tìm kiếm trong khi gỡ lỗi được thêm Địa phương, Ô tô và cửa sổ Xem được sử dụng để tìm các giá trị và đối tượng. Dưới đây hoạt hình sẽ giải thích quá trình thực tế gỡ lỗi cửa sổ tìm kiếm trong VS2019.

* Hiệu suất tốt hơn

Cho dù bạn đang mã hóa trong C ++ hay .NET Core, các điểm dừng dữ liệu có thể là một lựa chọn tốt để chỉ đặt các điểm dừng thông thường. Điểm dừng dữ liệu cũng tuyệt vời cho các tình huống như tìm kiếm nơi một đối tượng toàn cầu đang được sửa đổi hoặc được thêm hoặc xóa khỏi danh sách.

* Dọn mã trong một lần nhấp

Lệnh dọn mã mới trong Visual Studio 2019 là phát hiện các cảnh báo và đề xuất bằng nút một lần bấm. Lệnh này sẽ giúp định dạng mã và thực hiện các thay đổi thành định dạng mã hóa được đề xuất bởi các tệp và cài đặt .editorconfig. Nó cũng cho phép lưu các bộ sưu tập các bản sửa lỗi làm hồ sơ mặc định

* Hỗ trợ Xamarin

Visual Studio 2019 hỗ trợ Xamarin.Forms 4.0 cùng với các tính năng mới bao gồm các mẫu dự án Shell mới và các điều khiển CollectionView mới. Các mẫu Shell giúp người dùng thiết kế các ứng dụng dành cho thiết bị di động với giao diện được cải tiến và được liên kết chặt chẽ với Material Design. Xamarin CollectionView là một giải pháp thay thế tốt hơn và linh hoạt hơn cho các điều khiển ListView.

### 2.2.3. Giới thiệu về MongoDB



**Hình 3 Database MongoDB**

(Nguồn*: https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N.html*)

* MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.
* MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
* Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng
* So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.
* Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.
* Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

**2.2.3.1. Ưu điểm của MongoDB**

* Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.
* Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.
* MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:
* Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

**2.2.3.2. Nhược điểm của PostgreSQL:**

* Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.
* Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.
* Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện..

## 2.3. Tìm hiểu về .NET

### 2.3.1 Ngôn ngữ C#

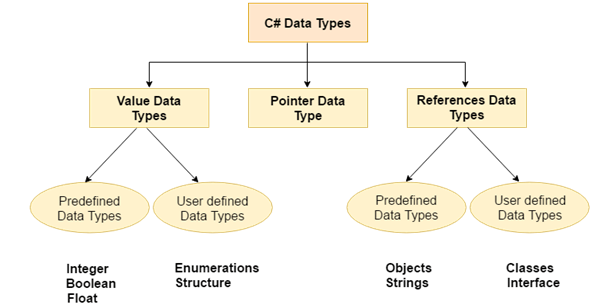
**2.3.1.1 Lịch sử phát triển của C#**



**Hình 4 Lịch sử phát triển C#**

(Nguồn: [https://kcntt.duytan.edu.vn/Home/ArticleDetail/vn/168/3633/lich-su-phat-trien-cua-c)](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh))

**2.3.1.2 Những đặc trưng của ngôn ngữ C#**



**Hình 5 Đặc trưng ngôn ngữ C#**

- Ngôn ngữ lập trình C# có những đặc trưng cơ bản sau:

* Đơn giản(Simple)
* Ngôn ngữ lập trình hiện đại(Modern programming language)
* Lập trình hướng đối tượng(Object oriented)
* Kiểu an toàn(Type safe)
* Khả năng tương tác(Interoperability)
* Có thể mở rộng và cập nhật(Scalable and Updateable)
* Hướng thành phần(Component oriented)
* Ngôn ngữ lập trình có cấu trúc(Structured programming language)
* Nhiều thư viện(Rich Library)
* Tốc độ nhanh(Fast speed)

### 2.3.2 Tìm hiểu về .NET Framework

**2.3.2.1 .NET Framework**

* .NET Framework được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.
* Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin…
* Cho đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)

## 2.4. Tổng quan về RestFul API



**Hình 6 Tổng quan ResFul API**

(Nguồn: <https://movan.vn/rest-api-gioi-thieu-rest-api/>)

- REST (REpresentational State Transfer) được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP).Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến ​​trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

- API (Application Programming Interface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau.

- REST API là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức để kết nối với các thư viện và ứng dụng khác. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),…

- Các ưu điểm của REST:

* Giúp cho ứng dụng trở nên rõ ràng hơn.
* REST URL đại diện cho resource chứ không phải là hành động.
* Dữ liệu được trả về với nhiều định dạng khác nhau như: xml, html, rss, json …
* Code đơn giản và ngắn gọn.
* REST chú trọng vào tài nguyên hệ thống.

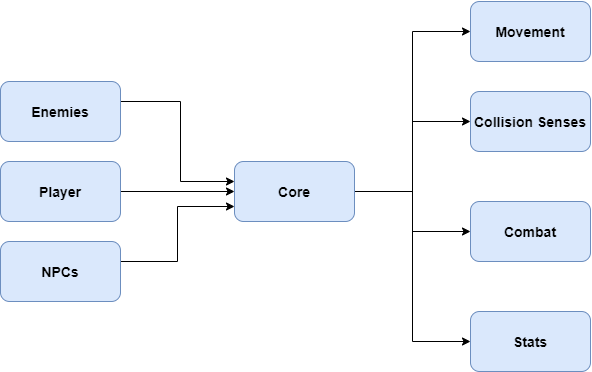
# CHƯƠNG 3: Xây dựng game 2D Soldier's Avenger

## 3.1 Khảo sát hệ thống Game 2D Soldier's Avenger

* Xây dựng Core cho tất cả đối tượng
* Xây dựng hệ thống Player dựa trên Core
* Xây dựng hệ thống Enemy dựa trên Core
* Xây dựng hệ thống Tile Map
* Xây dựng hệ thống bẫy.
* Xây dựng hiệu ứng chiến đấu, hiệu ứng chết.
* Xây dựng hệ thống audio clip và audio nền.
* Xây dựng Tile Map:
* Flatform: đất, mỏm đá tác động trực tiếp với player và enemy.
* Background: nền sau, tạo không khí địa điểm chiến đấu.
* Decorate front: nền trang trí ở phía trước layer Player và Enemy: đèn, cây, cỏ, nấm…
* Decorate back: nền trang trí ở phía sau layer Player và Enemy: đèn, cây, cỏ, nấm, bụi cây…
* Xây dựng hệ thống bẫy.
* Thiết kế Animation: mỗi loại bẫy có cách hoạt động và tác động khác nhau.
* Tác động Player và gây sát thương.
* Xây dựng hệ thống item.
* Item mặc định: coin và hạt kinh nghiệm.
* Item Inventory: Posion, rương vàng.
* Xây dựng hiệu ứng chiến đấu, hiệu ứng chết.
* Xây dựng hiệu ứng bị đánh hitPartical cho player, enemy và các thực thể không có máu như người rơm.
* Xây dựng hiệu ứng chết với các thực thể xác định: máu và mảnh xác.
* Xây dựng hiệu ứng lướt cho player.

## 3.2 Các chức năng của hệ thống

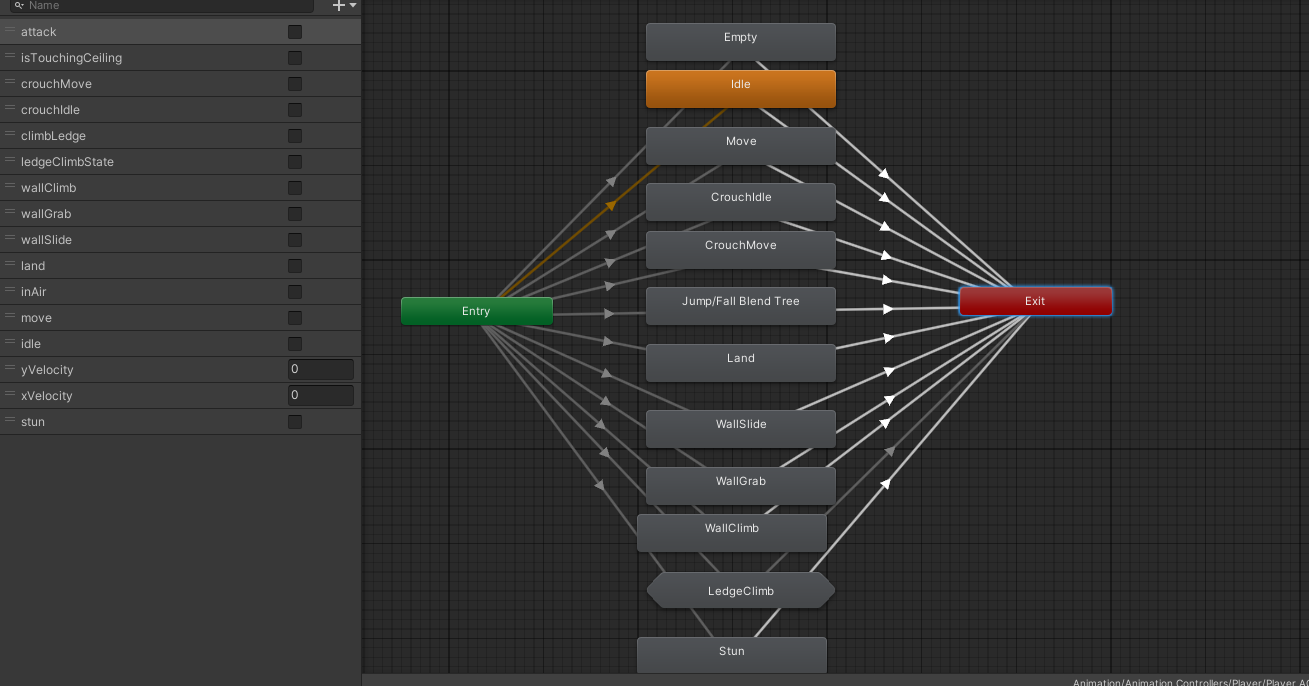
### 3.2.1 Xây dựng Core cho tất cả đối tượng

* 

**Hình 7: Core**

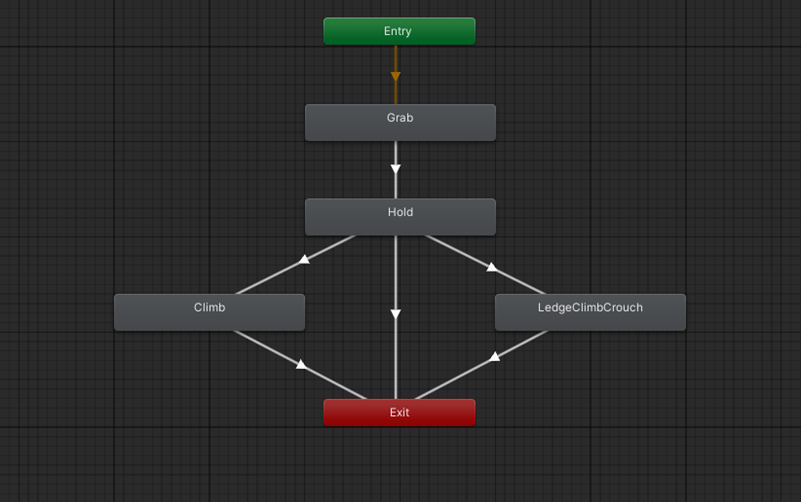
* Hệ thống di chuyển: đi, đứng.
* Hệ thống tương tác: Tương tác với vật phẩm và đối tượng khác.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm đất, chạm tường, chạm trần.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

### 3.2.2 Xây dựng hệ thống Player dựa trên Core

* Animation của Player : 

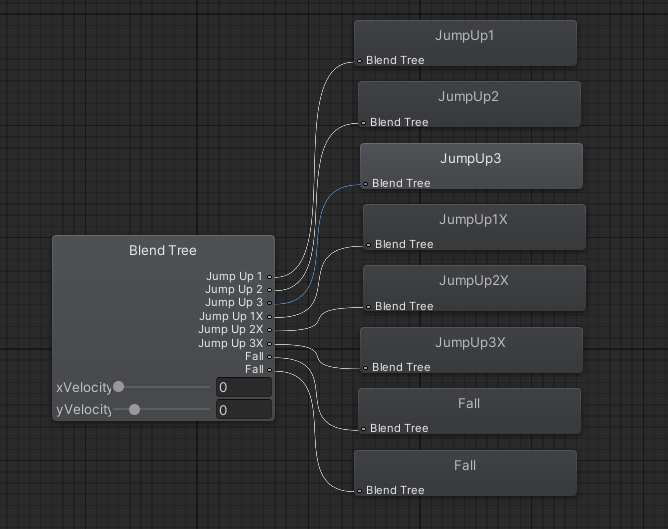
**Hình 8: Animation Player**

* Ledge Climb State: Trạng thái leo mỏm đá, đu mỏm đá.



**Hình 9: Ledge Climb State**

* BlendTree Ledge Climb State : Animation cho sự kiện nhảy lên và rơi xuống khi không tác động vector lực (vector x =0) và khi tác động lên vật thể 1 vector lực(vector x khác 0)



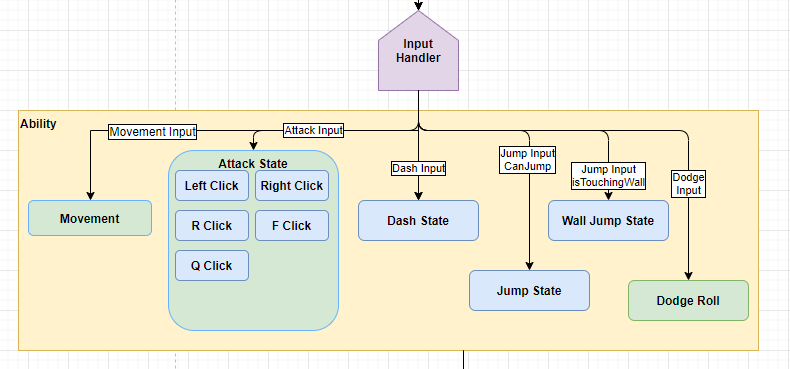
**Hình 10A: Blend Tree Player**

* Pramester BlendTree Player: các giá trị vector lực x, y để xác định animation của Player.

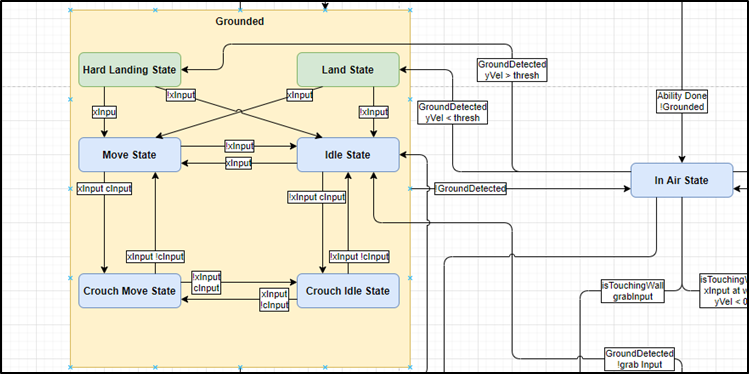


**Hình 10B: Pramester BlendTree Player**

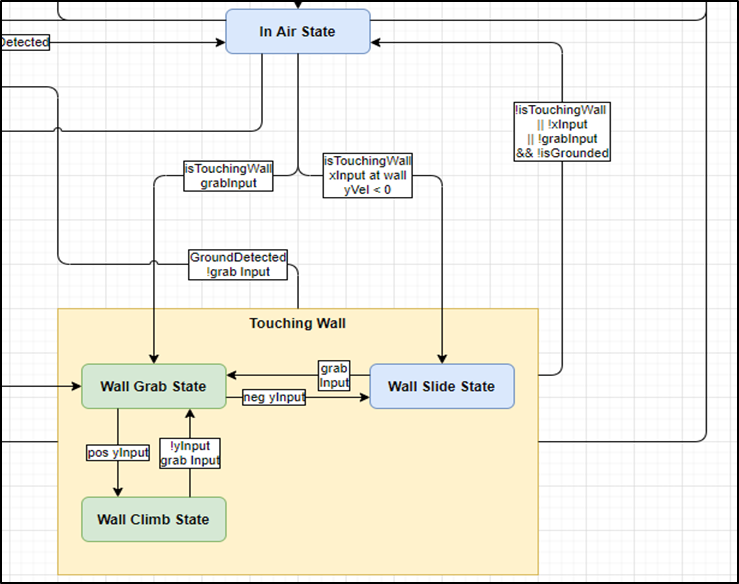
* Các Trạng thái của Player :
* AbilityState bao gồm :
* Attack State: trạng thái tấn công.
* Dash State: trạng thái lướt.
* Jump State: trạng thái nhảy thường.
* Wall Jump State: trạng thái nhảy tường.
* DodgeRoll: trạng thái cuộn tròn.
* In Air State: trạng thái trên không trung không chạm đất, tường.
* Touching Wall State: trạng thái khi chạm tường bao gồm :
* Wall Grab State: trạng thái bám tường.
* Wall Slide State: trạng thái trượt xuống tường.
* Wall Climb State: trạng thái trèo tường.
* Grounded State : trạng thái khi chạm đất bao gồm :
* Land State: trạng thái chạm đất khi vận tốc y bằng 0 khi đang trên không trung.
* Idle State: trạng thái đứng yên.
* Move State: trạng thái di chuyển.
* Crouch Move State: trạng thái di chuyển khom lưng.
* Crouch Idle State: trạng thái khom lưng.
* Ledge Climb: trạng thái trèo lên mỏm đá.
* Hệ thống Input chuyển đổi các trạng thái của AbilityState: di chuyển, tấn công, lướt, nhảy, nhảy tường.

 **Hình 11: AbilityState và Movement**

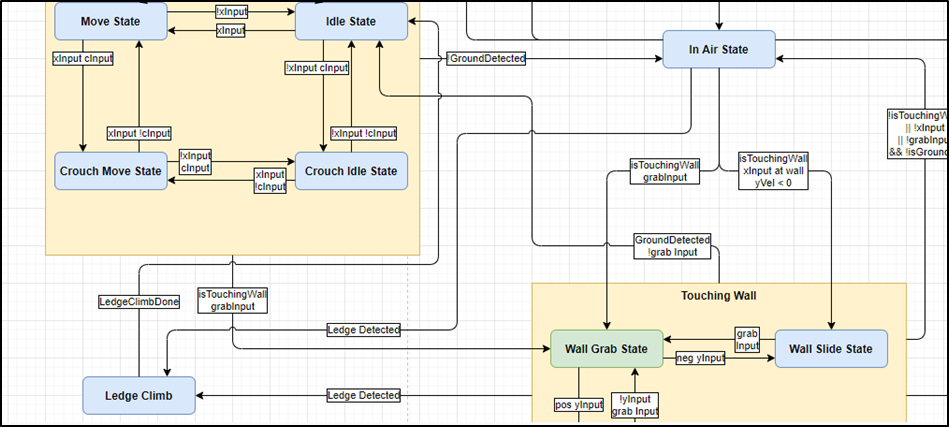
* Chuyển đổi các trạng thái Player :
* In Air State và Grounded State

**Hình 12: Chuyển đổi giữa In Air State và Grounded State**

* In Air State và Touching Wall State

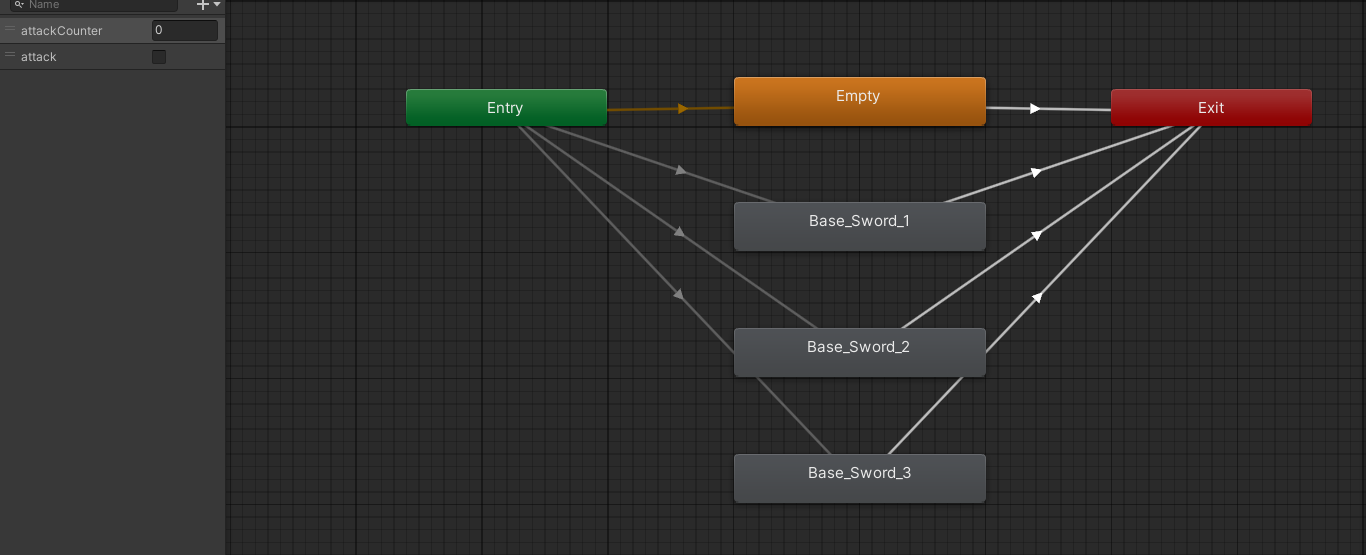


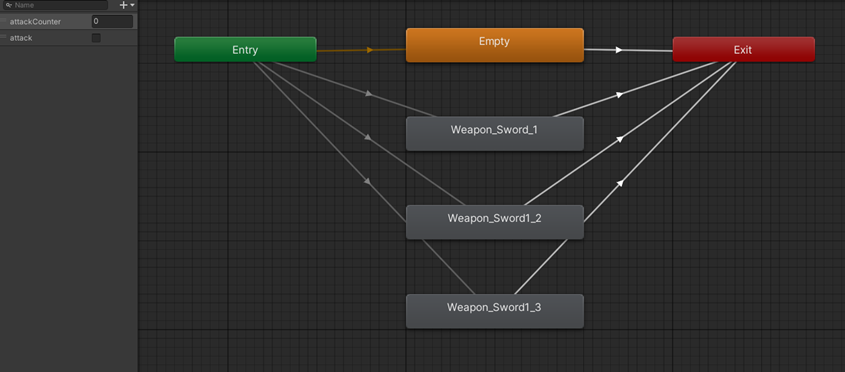
**Hình 13: In Air State sang Touching Wall State**

* Chuyển đổi giữa Ledge Climb State với các trạng thái khác:

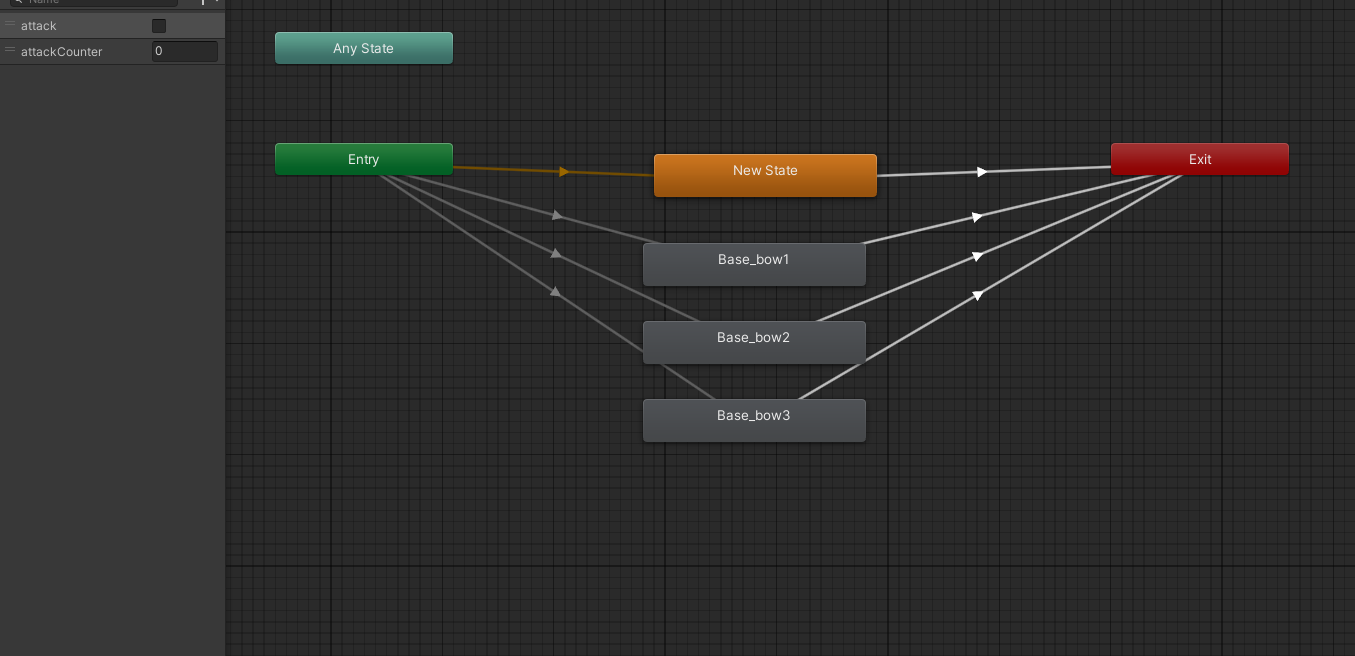
**Hình 14: Ledge Climb State**

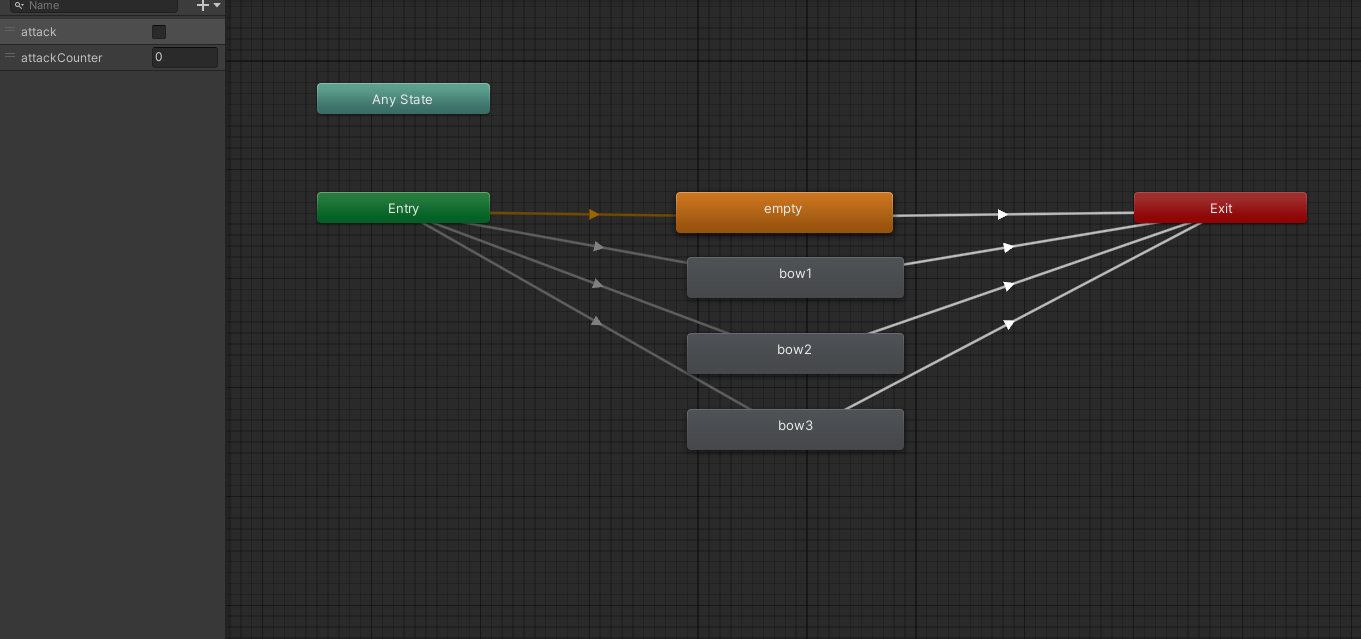
* Hệ thống tương tác: Chiến đấu với quái vật, kẻ thù, nhặt vật phẩm, sử dụng vật phẩm.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra Player chạm đất, chạm tường, chạm trần.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại, vàng, kinh nghiệm, cấp độ.
* Hệ thống vũ khí: kiếm, dao, khiên, cung tên.
* Kiếm : thiết lập animation đồng bộ giữa Player và kiếm.

**Hình 15: Animation thực thể player dùng kiếm**

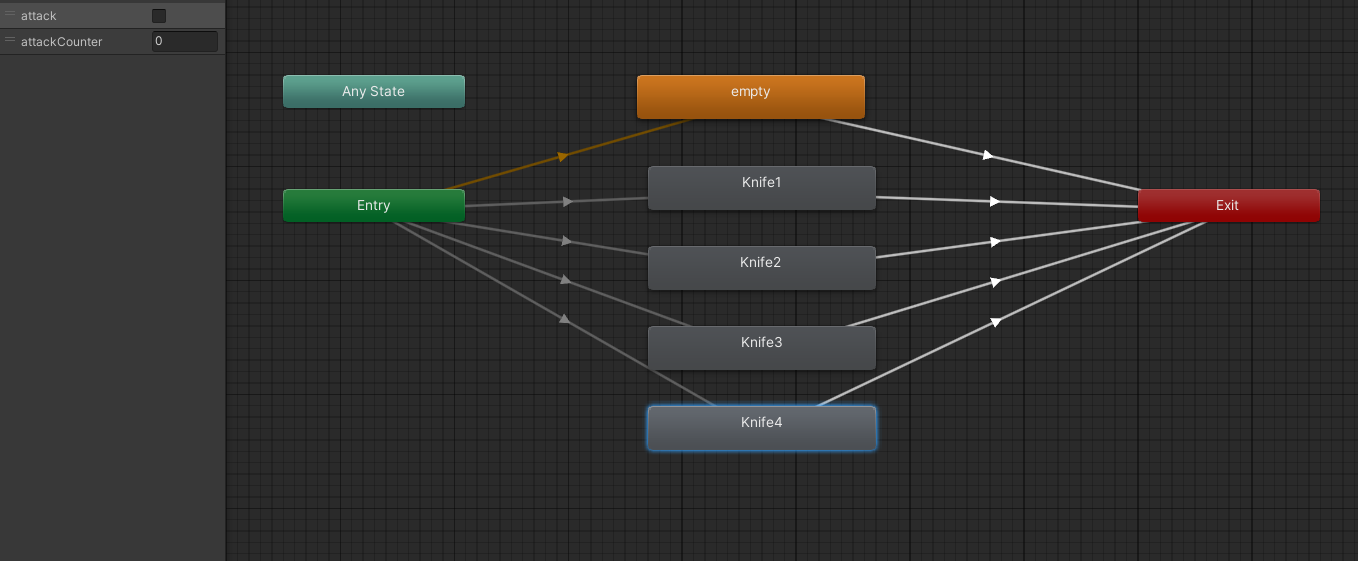
**Hình 16: Animation Weapon kiếm**

* Cung tên : thiết lập animation đồng bộ giữa Player và cung tên

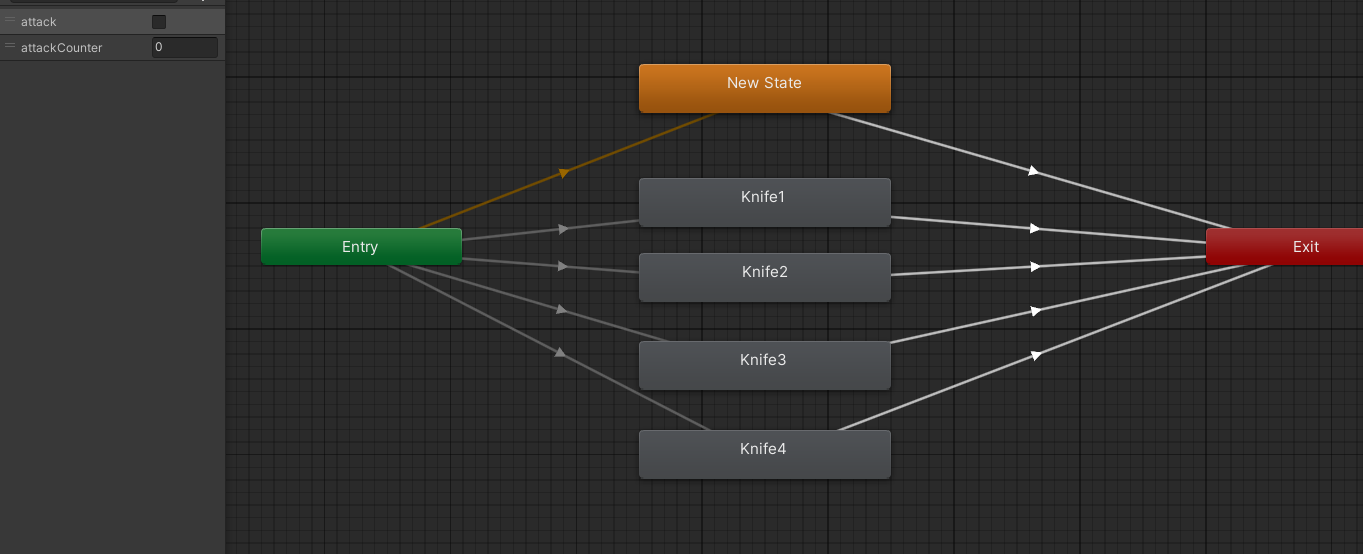
**Hình17: Animation thực thể Player dùng cung**



**Hình 18: Animation cung tên**

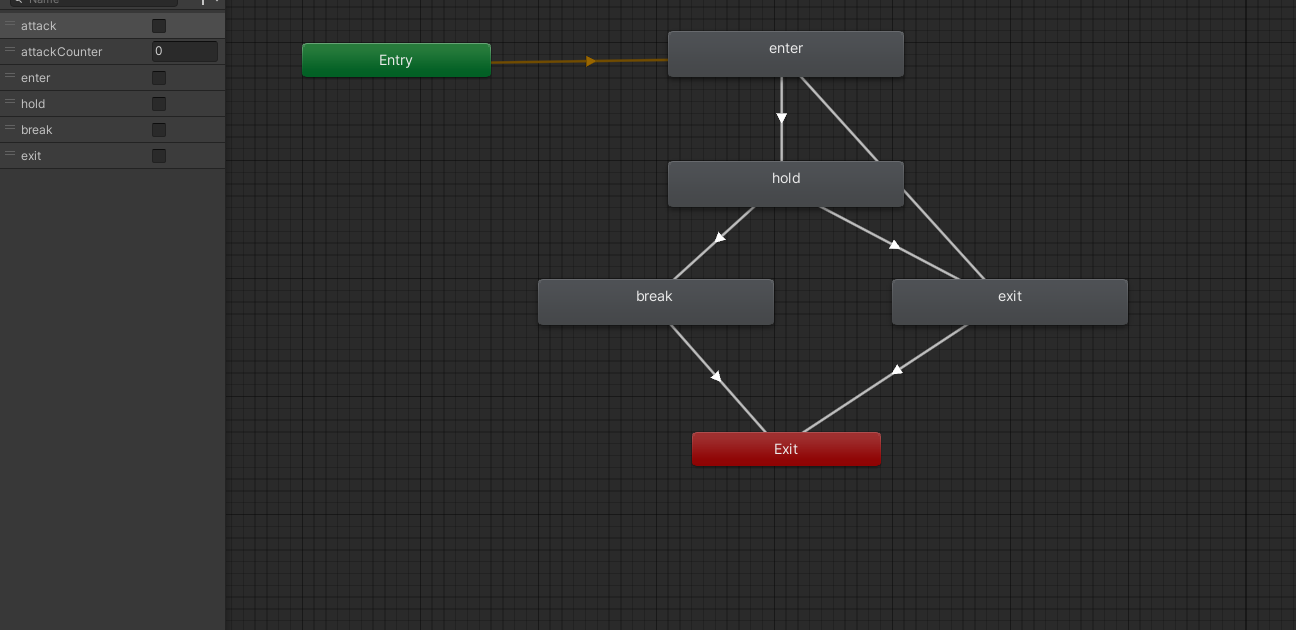
* Dao : thiết lập animation đồng bộ giữa Player và dao

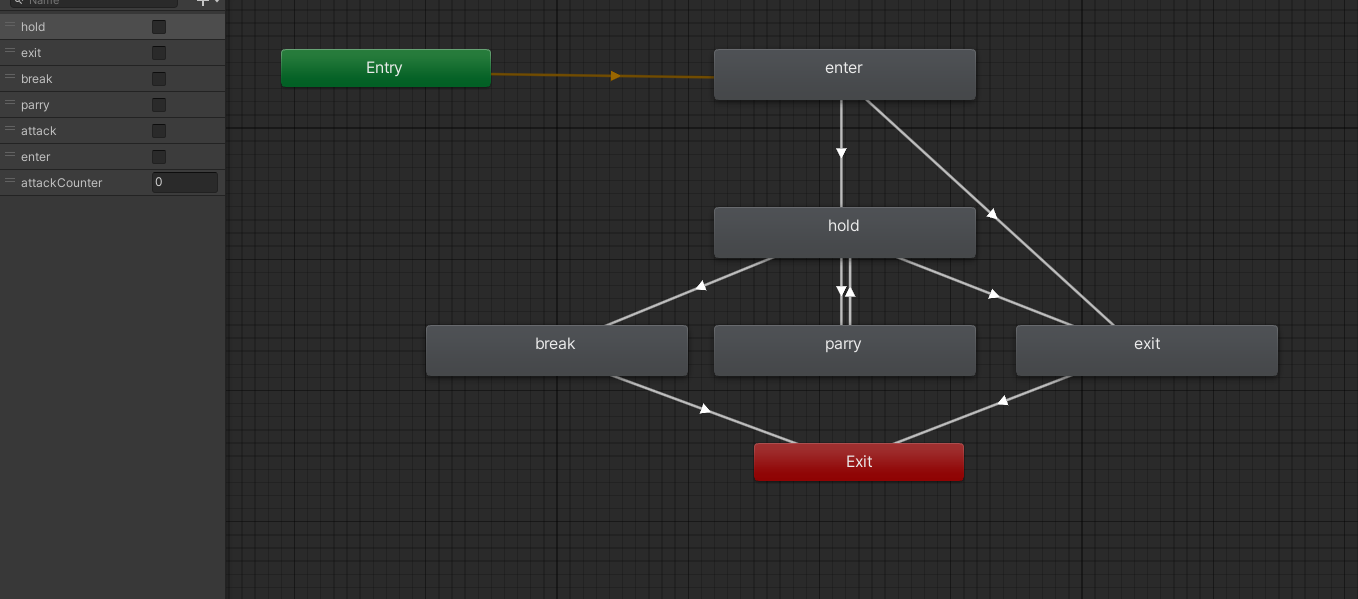
**Hình 19: Animation thực thể Player dùng dao**

****

**Hình 20: Animation dao**

* khiên : thiết lập animation đồng bộ giữa Player và khiên

**Hình 21: Animation thực thể Player dùng khiên**

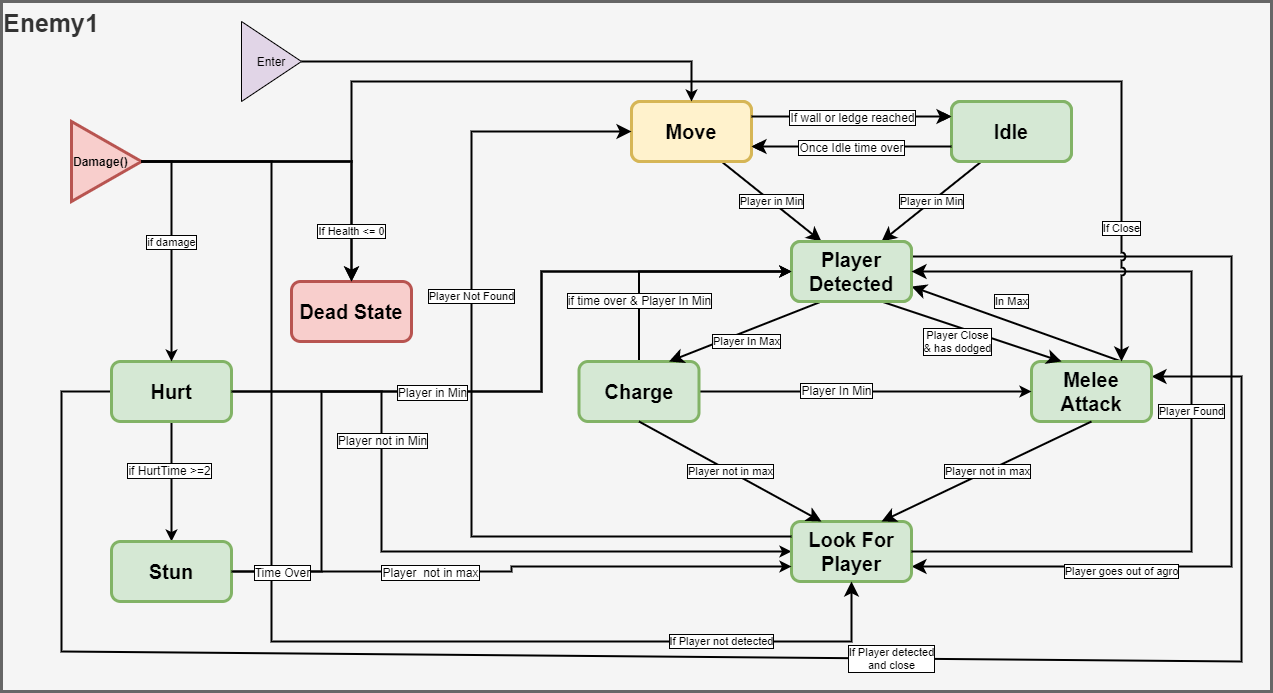
****

**Hình 22: Animation khiên**

### 3.2.3 Xây dựng hệ thống Enemy dựa trên Core

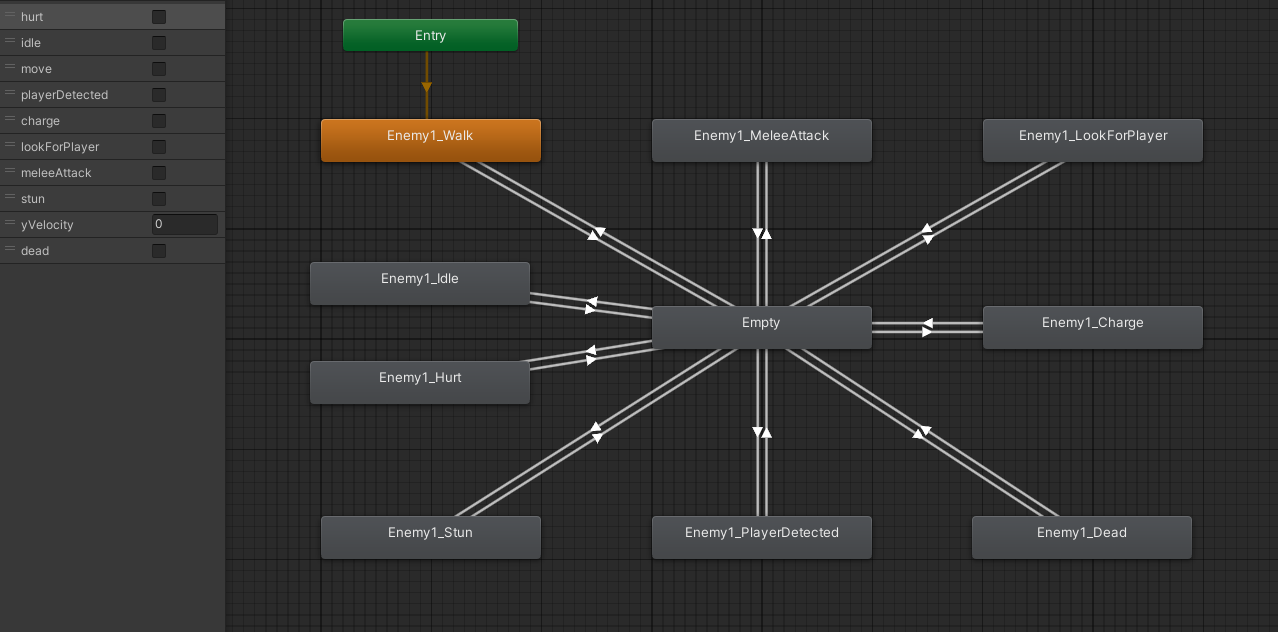
* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

#### **3.2.3.1 Enemy1\_Pork**



**Hình 23: Sơ đồ hoạt động Enemy 1 Pork**

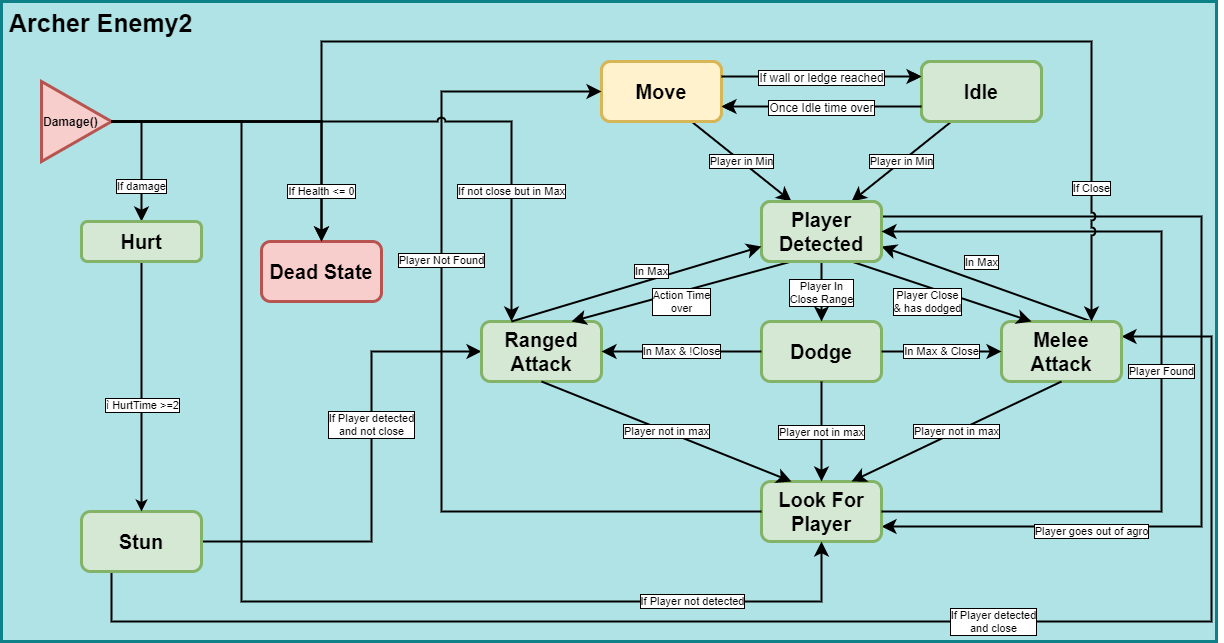
* Các Trạng thái của Enemy1\_Pork:
* Idle State: trạng thái đứng.
* Moving State: trạng thái mặc định di chuyển.
* Player Detected State: trạng thái xác định Player bước vào tầm xử lí.
* Charge Player State: trạng thái chuyển, tăng tốc tiến về phía Player.
* Attack Player State: trạng thái tấn công Player.
* Look For Player State: trạng thái tìm Player khi Player ra khỏi tầm xử lí.
* Stun State: Trạng thái choáng khi máu hiện tại dưới 20%.
* Dead State: Trạng thái chết.

**Hình 24 : Animation Enemy 1 Pork**

* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

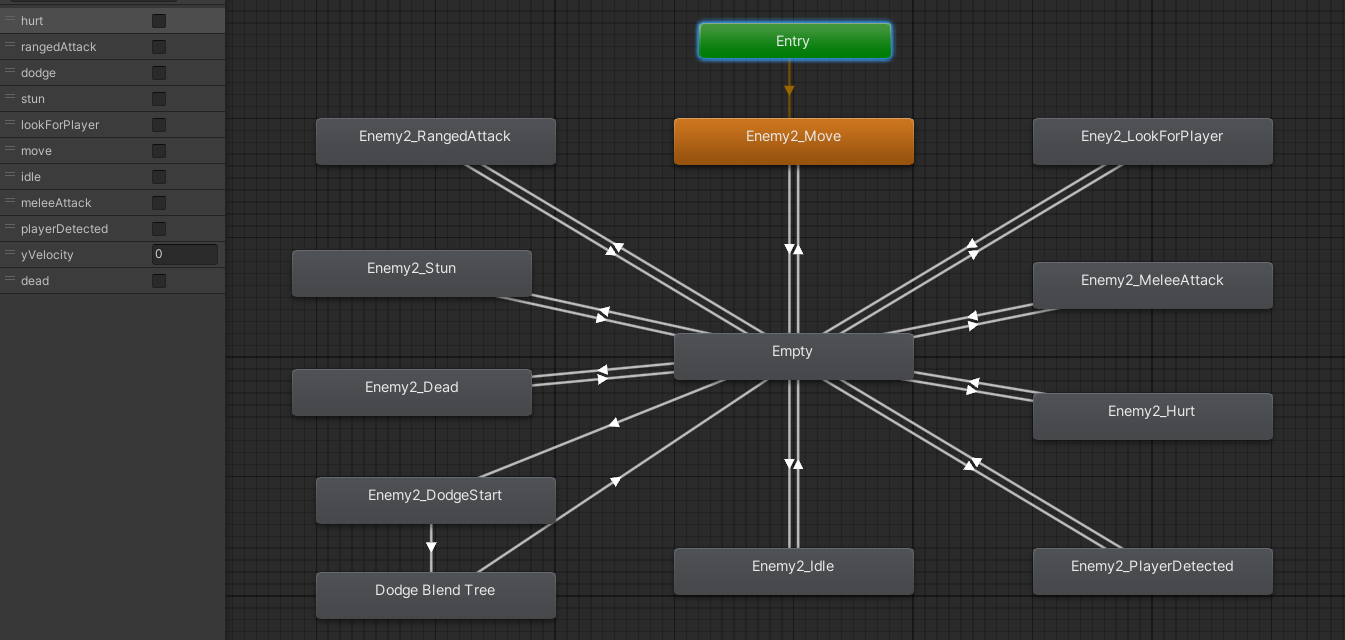
#### **3.2.3.2 Enemy2\_Archers**

* Các Trạng thái của Enemy2\_Archers:



**Hình 25: Sơ đồ hoạt động Enemy 2 Archers**

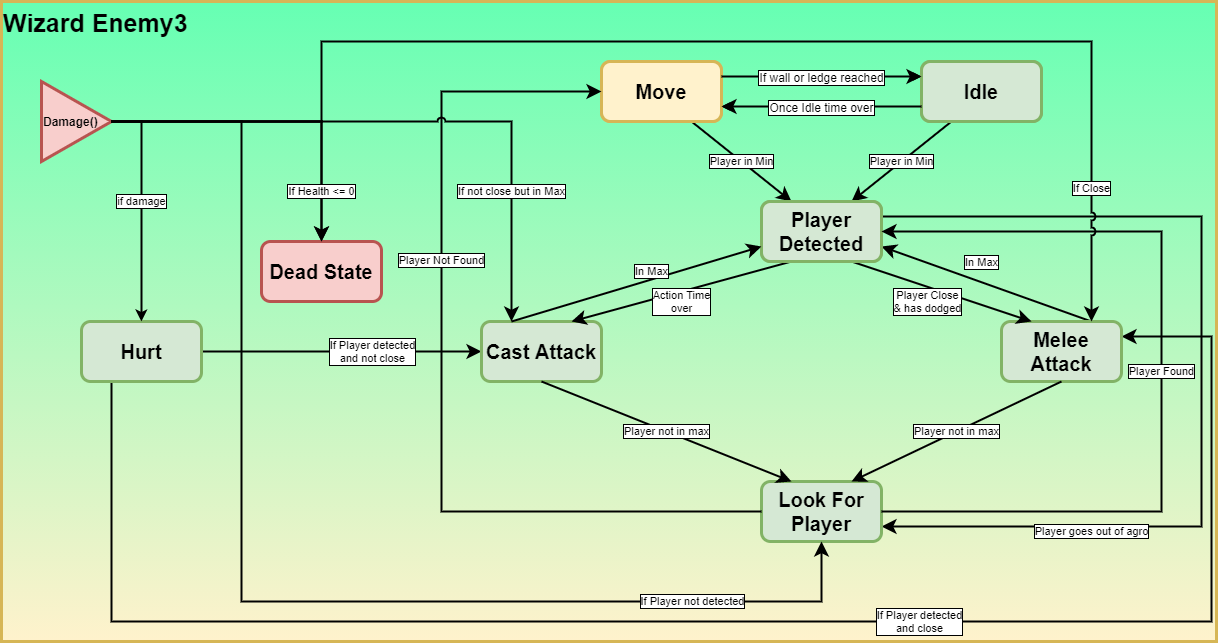
* Idle State: trạng thái đứng.
* Moving State: trạng thái mặc định di chuyển.
* Player Detected State: trạng thái xác định Player bước vào tầm xử lí.
* Dodge State: trạng thái nhảy lùi né tránh sự áp sát của Player.
* Melee Attack Player State: trạng thái tấn công cận chiến khi Player ở trong tầm đánh gần.
* Range Attack Player State: trạng thái bắn cung khi Player ở trong tầm đánh xa.
* Look For Player State: trạng thái tìm Player khi Player ra khỏi tầm xử lí.
* Stun State: Trạng thái choáng khi máu hiện tại dưới 20%.
* Hurt State: Trạng thái tổn thương khi bị đánh.
* Dead State: Trạng thái chết.

**Hình 26: Animation Enemy 2 Archers**

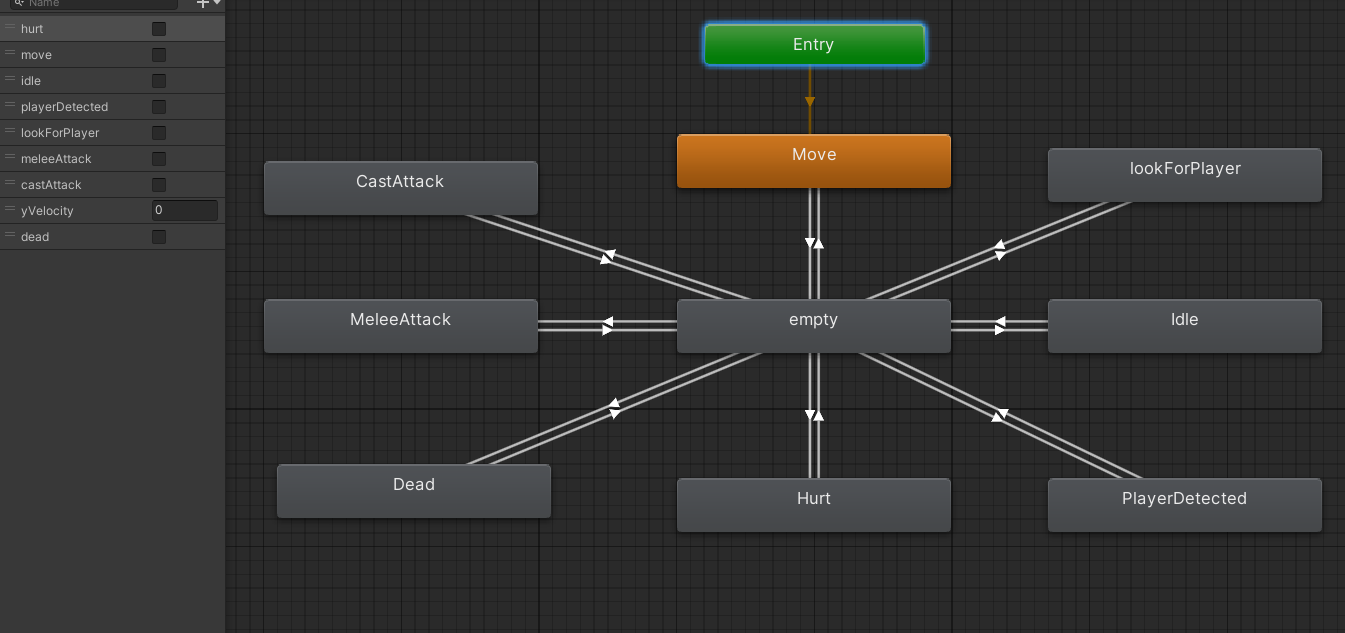
* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

#### **3.2.3.3 Enemy3\_Wizard**

**Hình 27: Sơ đồ hoạt động Enemy 3 Wizard**



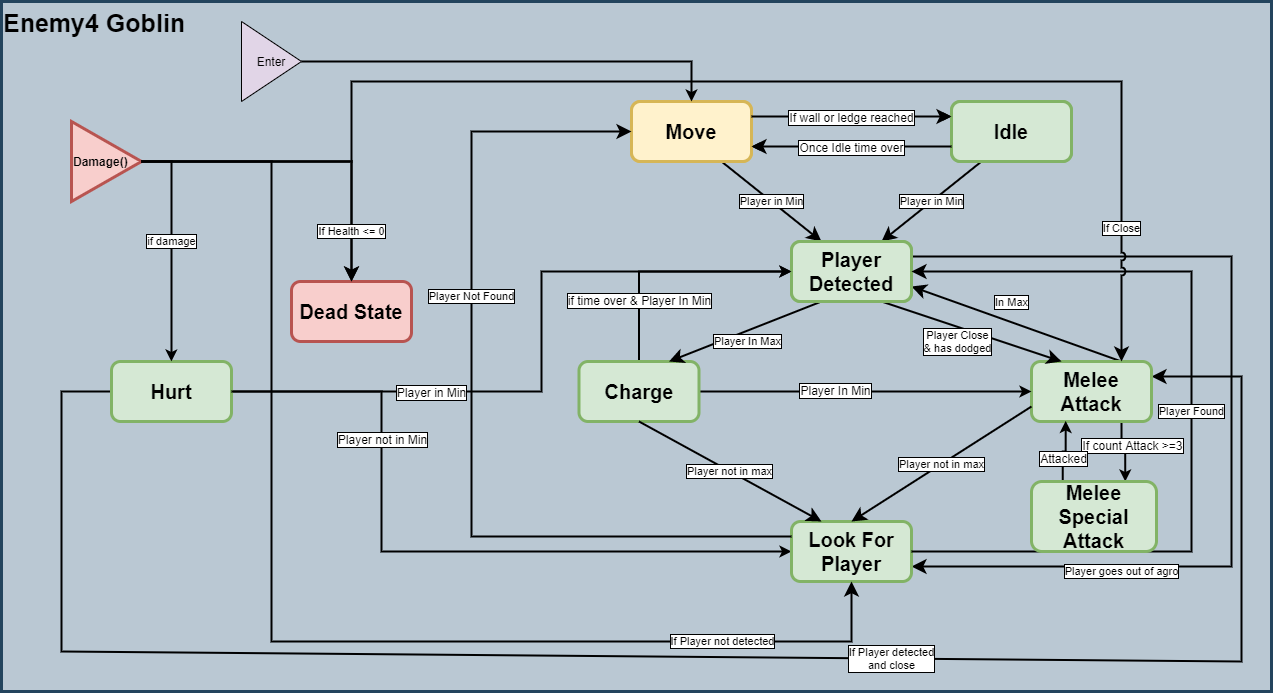
* Các Trạng thái của Enemy3\_Wizard:
* Idle State: trạng thái đứng.
* Moving State: trạng thái mặc định di chuyển.
* Player Detected State: trạng thái xác định Player bước vào tầm xử lí.
* Melee Attack Player State: trạng thái tấn công cận chiến khi Player ở trong tầm đánh gần.
* Cast Attack Player State: trạng thái niệm phép khi Player ở trong tầm đánh xa.
* Look For Player State: trạng thái tìm Player khi Player ra khỏi tầm xử lí.
* Hurt State: Trạng thái tổn thương khi bị đánh.
* Dead State: Trạng thái chết.

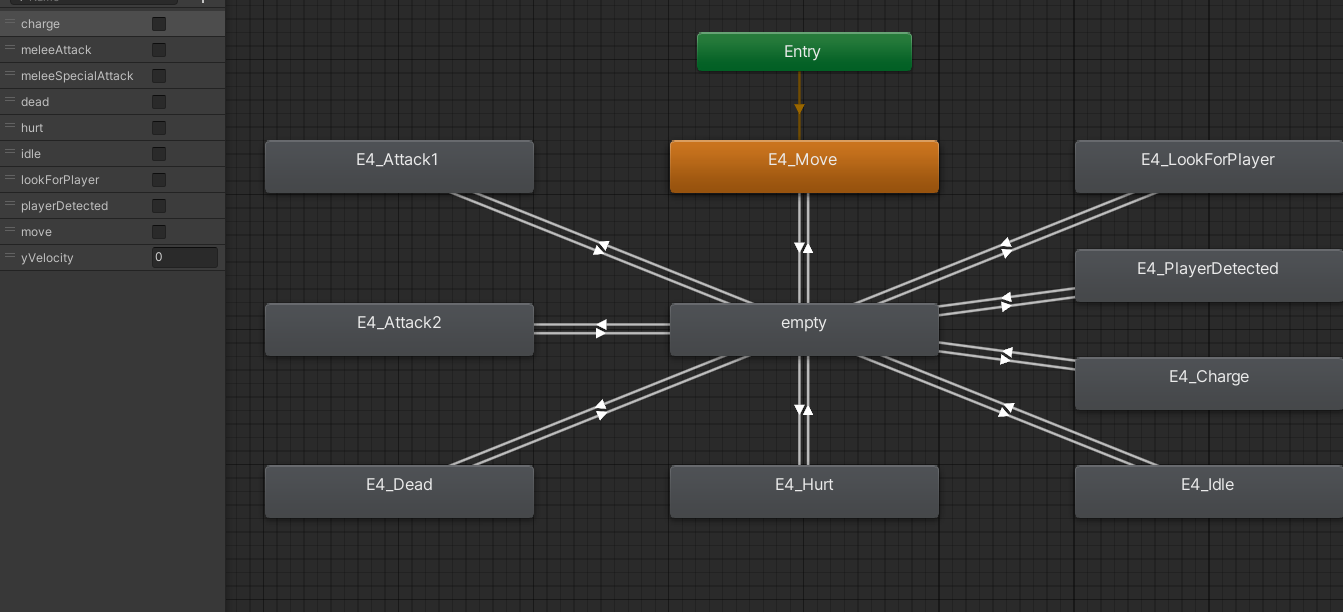
**Hình 28: Animation Enemy 3 Wizard**

* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

#### **3.2.3.4 Enemy4\_Goblin**

**Hình 29: Sơ đồ hoạt động Enemy 4 Goblin**

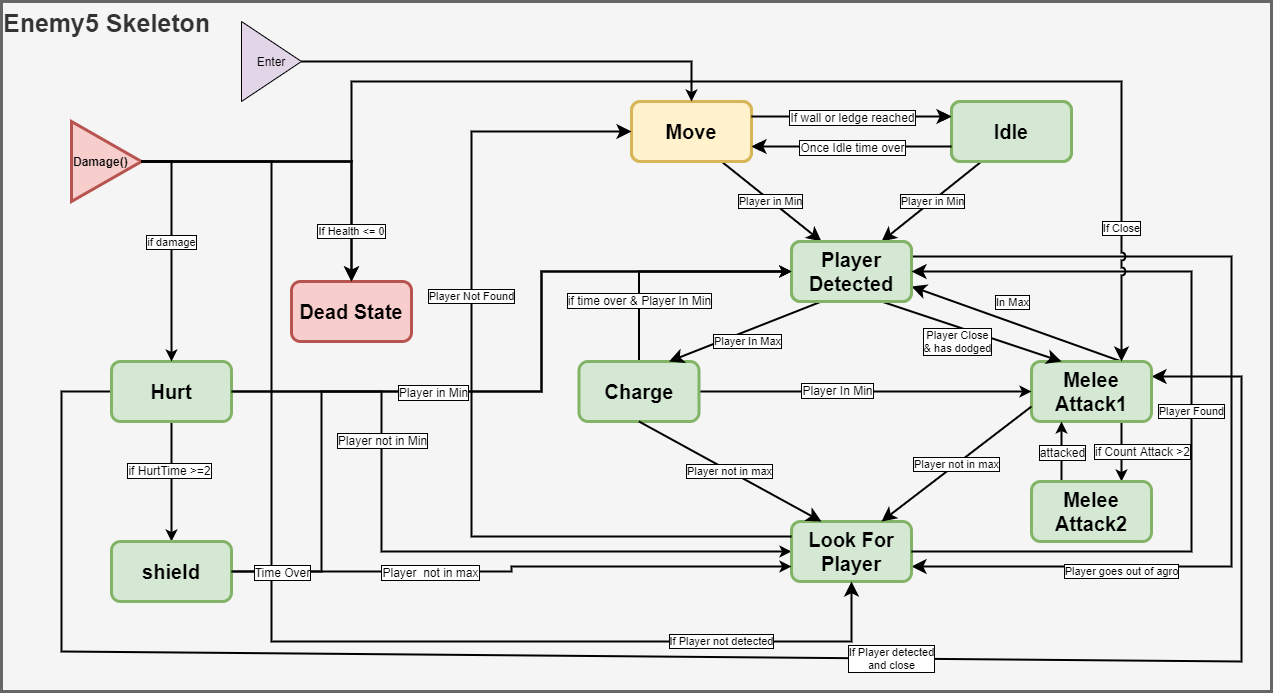


* Các Trạng thái của Enemy4\_Goblin:
* Idle State: trạng thái đứng.
* Moving State: trạng thái mặc định di chuyển.
* Player Detected State: trạng thái xác định Player bước vào tầm xử lí.
* Melee Attack State: trạng thái tấn công cận chiến khi Player ở trong tầm đánh gần.
* Melee Special Attack State: trạng thái tấn công cận chiến khi đã đánh thường hai lần.
* Charge Player State: trạng thái chuyển, tăng tốc tiến về phía Player.
* Look For Player State: trạng thái tìm Player khi Player ra khỏi tầm xử lí.
* Hurt State: Trạng thái tổn thương khi bị đánh.
* Dead State: Trạng thái chết.

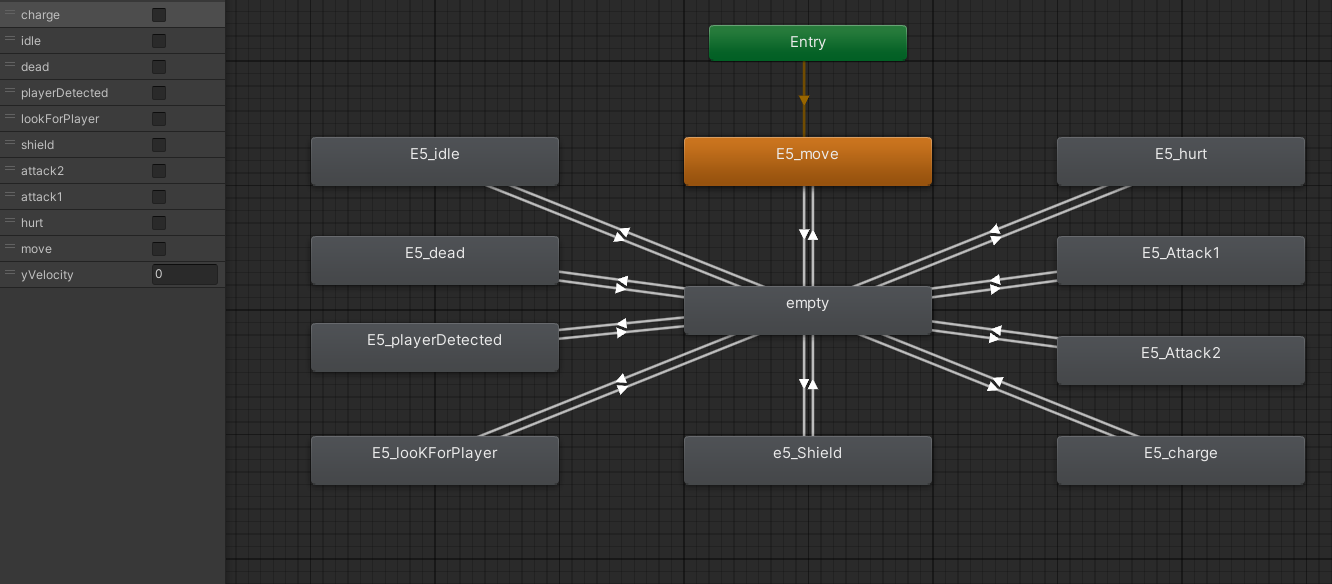
**Hình 30: Animation Enemy 4 Goblin**

* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

#### **3.2.3.5 Enemy5\_Skeleton**



**Hình 31: Sơ đồ hoạt động Enemy 5 Skeleton**

* Các Trạng thái của Enemy5\_Skeleton:
* Idle State: trạng thái đứng.
* Moving State: trạng thái mặc định di chuyển.
* Player Detected State: trạng thái xác định Player bước vào tầm xử lí.
* Melee Attack 1 Player State: trạng thái tấn công cận chiến khi Player ở trong tầm đánh gần.
* Melee Attack 2 State: trạng thái tấn công cận chiến khi đã đánh thường hai lần.
* Charge Player State: trạng thái chuyển, tăng tốc tiến về phía Player.
* Look For Player State: trạng thái tìm Player khi Player ra khỏi tầm xử lí.
* Hurt State: Trạng thái tổn thương khi bị đánh.
* Shield State: Trạng thái mở khiên chắn khi bị tổn thương hai lần.
* Dead State: Trạng thái chết.

**Hình 32: Animation Enemy 5 Skeleton**

* Hệ thống di chuyển: đi, chạy.
* Hệ thống tương tác: Chiến đấu với Player.
* Hệ thống kiểm tra tương quan môi trường: Kiểm tra chạm tường, vượt ra khỏi địa hình, kiểm tra Player ở trong tầm xử lí.
* Hệ thống trạng thái: Máu tối đa, máu hiện tại.

## 3.5 Mô hình Usecase

### 3.5.1 Danh sách các Actor

### 3.5.2 Xác định các usecase

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Actor** | **Tên use case** | **Ý nghĩa** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.5.3 Biểu đồ Usecase tổng quát của hệ thống

### 3.5.4 Đặc tả các usecase

## 3.6 Thiết lập mô hình thực thể

### 3.6.1 Xác định các thực thể

### 3.6.2 Xác định mối quan hệ giữa các thực thể

### 3.6.3 Sơ đồ ERD (Entity Relationship Diagram)

**Sơ đồ 3.1. Sơ đồ ERD**

### 3.6.4 Mô hình dữ liệu quan hệ

## 3.7 Thiết kế dữ liệu

### 3.7.1 Từ điển dữ liệu

-

### 3.7.2 Mô hình diagram

**Hình 3.5. Mô hình Diagram**

## 3.8 Xây dựng dự án

### 3.8.1 Xây dựng BackEnd

# CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## 4.1 Kết quả