# LỜI CAM ĐOAN

Tôi xin cam đoan đồ án cơ sở “Xây dựng game hành động trên framework Unity” là công trình nghiên cứu của bản thân dưới sự hướng dẫn của GVHD Nguyễn Anh Tú. Các nội dung, kết quả nghiên cứu đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những thông tin được tham khảo của tác giả, cơ quan tổ chức trong quá trình thực hiện đồ án đã được nêu rõ trong tài liệu tham khảo.

Tôi hoàn toàn chịu trách nhiệm về đồ án của mình.

Thành phố Hồ Chí Minh, ngày 6 tháng 6 năm 2020

Sinh viên thực hiện

**Nguyễn Hoàng Quân**

# LỜI MỞ ĐẦU

Nhu cầu giải trí tăng cao cùng với sự phát triển của các nền tảng công nghệ khiến cho phát triển game trở thành một nghề cực kỳ có sức hút đối với giới trẻ. Trong đó một nghề sẽ khiến rất nhiều các bạn trẻ hứng thú: lập trình và phát triển game.

Lập trình game là một trong những lĩnh vực còn khá non trẻ tại Việt Nam. Nhưng không vì vậy mà thiếu đi nhiều công ty game làm nên tên tuổi của mình tại Việt Nam cũng như quốc tế, Có thể kể đến nhiều cái tên như Athena (chuyên sản xuất dòng game giải đố), Amanotes (game âm nhạc), TopeBox (game phổ thông), DivMob (game chiến đấu thời gian thực) hay WolfFun (game đấu trường trực tuyến)… đều được giới chuyên gia trong lĩnh vực game đánh giá là những công ty vẫn đang kiếm doanh thu khủng từ thị trường nước ngoài.

Em đã quyết định chọn đề tài: ”Xây dựng game hành động trên framework Unity” để thực hiện, nhằm thực hiên ước mơ, đam mê làm game cũng như mong muốn lĩnh vực game của Việt Nam phát triên mạnh và có nhiều game do người Việt phát hành và song song là việc tìm hiểu về Unity Framework.

**Mục lục**

[LỜI CAM ĐOAN 1](#_Toc42339692)

[LỜI MỞ ĐẦU 2](#_Toc42339693)

[DANH MỤC CÁC CHỮ VIẾT TẮT 4](#_Toc42339694)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 5](#_Toc42339695)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 6](#_Toc42339696)

[1.1 Nhiệm vụ đồ án. 6](#_Toc42339697)

[1.1.2 Lý do chọn đề tài: 6](#_Toc42339698)

[1.1.2 Ý nghĩa khoa học và thực tiển: 6](#_Toc42339699)

[1.1.3 Mục tiêu nghiên cứu và phạm vi giớ hạn: 7](#_Toc42339700)

[1.2 Cấu trúc đồ án. 8](#_Toc42339701)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc42339702)

[2.1 Unity là gì? 9](#_Toc42339703)

[2.2 Quá trình phát triển của Unity 9](#_Toc42339704)

[2.3 Tính năng của engine Unity 10](#_Toc42339705)

[2.4 Các khái niệm trong Unity 10](#_Toc42339706)

[2.4.1 Asset 10](#_Toc42339707)

[2.4.2 Scene 11](#_Toc42339708)

[2.4.3 Game Object 11](#_Toc42339709)

[2.4.4 Component 11](#_Toc42339710)

[2.4.5 Script 12](#_Toc42339711)

[2.4.6 Prefab 12](#_Toc42339712)

[2.4.7 Material và Shader 12](#_Toc42339713)

[2.4.8 UnityAPI 12](#_Toc42339714)

[2.4.9 Không gian 3D/2D và các vector trong không gian 13](#_Toc42339715)

[2.5 Cửa sổ làm việc 16](#_Toc42339716)

[2.5.5 Scene View 19](#_Toc42339717)

[2.5.6 Game View. 20](#_Toc42339718)

[2.5.7 Play Mode 20](#_Toc42339719)

[2.5.8 Inspector 20](#_Toc42339720)

[2.6 Lập trình C# trên Visual Studio 22](#_Toc42339721)

[2.7 kết luận 23](#_Toc42339722)

[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM: 24](#_Toc42339723)

[3.1 Giới thiệu game 24](#_Toc42339724)

[3.1.3 Bối cảnh 24](#_Toc42339725)

[3.1.2 Quy luật chơi 24](#_Toc42339726)

[3.2 Thiết kế nhân vật: 24](#_Toc42339727)

[3.2.1 Player: 24](#_Toc42339728)

[3.2.2 Enemy: 24](#_Toc42339729)

[3.3 Địa hình trông game: 25](#_Toc42339730)

[3.3.1 Dungeon 25](#_Toc42339731)

[3.3.2 Tự Động tạo Màng Chơi 25](#_Toc42339732)

[3.4 GUI 25](#_Toc42339733)

[3.5 Menu Game 25](#_Toc42339734)

[CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ 26](#_Toc42339735)

[4.1 Kết quả đạt được. 26](#_Toc42339736)

[4.2 Hướng phát triển & mở rộng đề tài. 26](#_Toc42339737)

[TÀI LIỆU THAM KHẢO 27](#_Toc42339738)

# DANH MỤC CÁC CHỮ VIẾT TẮT

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 2.0.1: các nền tảng được hỗ trợ của unity 10](#_Toc42341620)

[Hình 2.0.2: Vị trí lưu Scenes trong Asset của Project 12](#_Toc42341621)

[Hình 2.0.3: Một chiếc cầu thang trong không gian 3 chiều (nguồn internet) 14](#_Toc42341622)

[Hình 2.0.4: Tương ứng hình 2.3 với Rotation X, Y, Z (nguồn internet) 15](#_Toc42341623)

[Hình 2.0.5: Một GameObject trong không gian 2 chiều 15](#_Toc42341624)

[Hình 2.0.6: Tương ứng hình 2.5 với Rotation X, Y, Z 16](#_Toc42341625)

[Hình 2.0.7: Giao diện chính của Unity Editor 16](#_Toc42341626)

[Hình 2.0.8: Cửa sổ Project 17](#_Toc42341627)

[Hình 2.0.9: Của sổ Hierachy 17](#_Toc42341628)

[Hình 2.0.10: Parenting trong Hierachy 18](#_Toc42341629)

[Hình 2.0.11: Toolbar 19](#_Toc42341630)

[Hình 2.0.12: Các góc nhìn từ các hướng khác nhau (Nguồn Internet) 20](#_Toc42341631)

[Hình 2.0.13: Bản Inspector của một GameObject Player 21](#_Toc42341632)

[Hình 0.1 Animation di chuyễn của Player 24](#_Toc42341633)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

# 1.1 Nhiệm vụ đồ án.

Đề tải ”Xây dựng game hành động trên framework Unity” được thực hiện trên framework Unity và các kiến thức tổng hợp từ các môn Lập trình hướng đối tượng, Kỹ thuật lập trình, Trí tuệ nhân tạo, C#, … Trong đó kiến thức Unity là tự học và tìm hiểu

#### 1.1.2 Lý do chọn đề tài:

Do xã hội ngày càng phát triển, nhu cầu giải trí của con người ngày càng tăng cao. Cùng với sự phát triển của công nghệ thông tin, điện tử, các thiết bị chơi game ngày càng phổ biến. Trong những năm gần đây ngành công nghệ game nổi lên và đem lại doanh thu khổng lồ. Có thể nói game không chỉ là một phương tiện giải trí cơ bản, nó còn là một lĩnh vực đáng quan tâm và theo đuổi. Thiết kế game là một công việc không dễ dàng đòi hỏi rất nhiều thời gian và công sức, tuy nhiên quá đó ta có thể biến những dòng code khô khan thành những trải nghiệm thú vị cho người chơi, đồng thời có thể học hỏi những kiến thức mới về nền tảng Unity Framework và củng cố ngôn ngữ lập trình C#.

Một phần là do đam mê lập trình game, muốn thực hiện hóa ý tưởng của bản thân và đã có tìm hiều trước về lĩnh vực này.

#### 1.1.2 Ý nghĩa khoa học và thực tiển:

Không chỉ hội họa hay sáng tác nhạc hay viết lách, lập trình game cũng được vào tốp ngành yêu cầu cao về yếu tố thẩm mỹ và sáng tạo vượt trội. Thao tác đầu tiên cho mọi ván game đó chính là lên ý tưởng cho game, cho nhân vật của bạn như thế nào, sau đó đến bối cảnh, màu sắc, hành động nhân vật và các chuyển động và hiệu ứng trên thế giới ảo. Bạn phải thảo những ý tưởng đó trên giấy và ghi chép lại thật cẩn thận đến khi nào hoàn thiện nó. Đây chính là tài liệu quan trọng để bạn triển khai xây dựng hình ảnh thông qua các Game Developer. Bộ óc sáng tạo cho cả quá trình sản xuất ra ấn phẩm game phục vụ công động game thủ chính thuộc về Game Artist và Game Designer. [1]

Theo thống kê của Pearl Research - công ty nghiên cứu về Internet và công nghệ của Hoa Kỳ - Việt Nam, thời điểm 2015, tại Việt Nam có đến 12 triệu game thủ. Một năm sau đó, 2016, theo kết quả từ cuộc điều tra của công ty chuyên tư về vấn công nghệ Digi- Capital, nguồn lợi nhuận mà các dự án Game công nghệ thế giới mang lại cho là trên 85 tỷ USD trong tốp ngành thuộc về công nghệ. Thông tin mới nhất từ New Zoom năm 2018 cho hay, toàn cầu có khoảng 2,3 tỷ game thủ và số tiền rót vào game và các trò chơi trực tuyến lên đến con số khoảng 137,9 tỷ USD. Ở Mỹ, tổng thu nhập của một lập trình viên game ước đạt con số 40.000 USD/năm. Tại Việt Nam, ngành Game bao gồm lập trình Game là ngành dẫn đầu các ngành công nghệ mang lại doanh thu lên đến 150 triệu USD chiếm 70% tổng nguồn thu của ngành công nghệ số. [1]

#### 1.1.3 Mục tiêu nghiên cứu và phạm vi giớ hạn:

Mục tiêu nghiên cứu: nền tảng Unity 2D trên PC.

Cách thiết kế nhân vật, tạo bản đồ, khung cảnh, character animator, xử lý va chạm, healthy player và enemy, điểm số ,màn chơi,tạo AI cho enemy, xử lý hiệu ứng hình ảnh, âm thanh

Hạn chế: do lần đầu thực thiện về lĩnh vực lập trình game nên vẫn còn nhiều sai xót và một phần kiến thức tìm hiều chưa hoàn thiện nên hiện tại chỉ có thể làm game đơn giản 2D.

#### 1.2 Cấu trúc đồ án.

Đồ án gồm bốn chương:

Chương 1: Tổng quan về đồ án, hướng để xây dựng đồ án, các thống kế thông tin về việc quản lý sinh viên ngoại trú,các khó khăn hiện hữu phát sinh khi xây dựng đề tài,tầm nhìn khi thực hiện đề tài, cũng như hạn chế nhận thấy khi thực hiện đề tài.

Chương 2: Cơ sở lý thuyết trình bày về các khái niệm liên quan đến Unity, giới thiệu tổng quan về các tính năng của Unity sử dụng trong đề tài và giới thiệu sơ bộ về C# ngôn ngữ lập trình chính trên Unity

Chương 3: Mô tả về game, các kết quả nghiên cứu lý thuyết, thiết kế nhân vật, giao diện, bản đồ, cách thức vận hành của game.

Chương 4: Kết luận và phương hướng phát triển đưa ra các kết quả đạt được cũng như phương hướng mở rộng, phát triển trong tương lai cho đồ án.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Unity là gì?

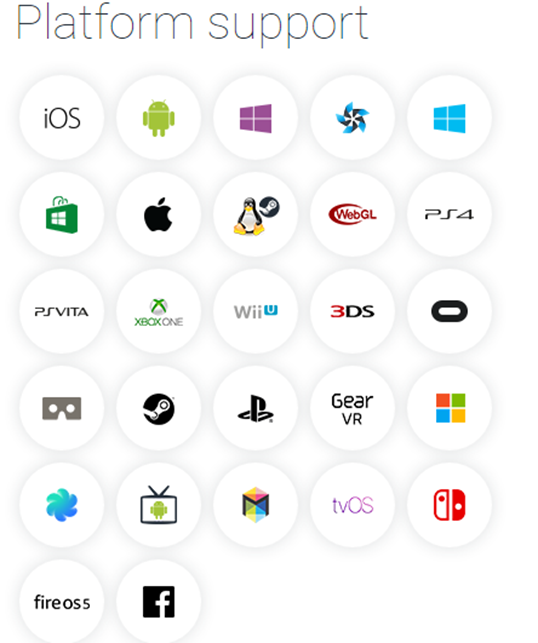
Unity là một “cross- flatform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát trển game trên PC, consoles, thiết bị di động và trên websites. [2]

## 2.2 Quá trình phát triển của Unity

Ra mắt đầu tiên vào năm 2005 tại sự kiện Apple’s Worldwide Developer Conference bởi nhà sáng lập David Helgason, trải qua hơn 12 năm phát triển, nay Unity đã có version 5.5 hoàn thiện hơn về rất nhiều mặt. Tháng 5-2012 theo cuộc khảo sát Game Developer Megazine được công nhận là Game engine tốt nhất cho mobile. Năm 2014 Unity thắng giải “Best Engine” tại giải UK’s annual Develop Industry Exellence.

Tác động của Unity Trong 12 tháng qua:

* + 100 triệu game thủ đã liên lạc trên các nền tảng bằng dịch vụ thoại và văn bản Unity vào năm 2019.
  + Tỷ ứng dụng được phát triển bởi Studio sử dụng Unity được tải xuống mỗi tháng 54% trong số 1.000 trò chơi trên di động có doanh thu cao nhất do Unity vận hành trên toàn cầu.
  + Trung bình 50% trò chơi vận hành bởi Unity trên các nền tảng (PC/console/mobile).
  + Trong đó có hơn 20 nền tảng được hỗ trợ và còn hơn thế nữa.



Hình 2.0.1: các nền tảng được hỗ trợ của unity

## 2.3 Tính năng của engine 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….

Dễ sử dụng: Unity 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 Unity 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 Unity 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.

**2.4 Các khái niệm trong Unity**

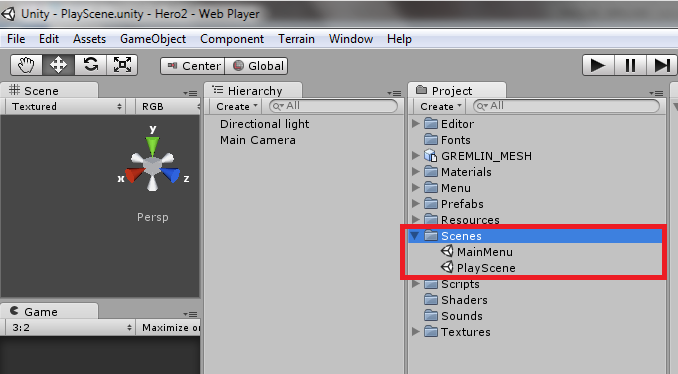
### **2.4.1 Asset**

Đây là kho tài nguyên cho việc xây dựng game trong một project của Unity. Các tài nguyên này có thể là hình ảnh, âm thanh, hoặc một mô hình 3D có sẵn. Unity sẽ tham chiếu đến các tập tin chúng ta sẽ sử dụng để tạo ra các tài nguyên cho trò chơi. Đây là lý do tại sao trong bất kỳ thư mục chứa project sử dụng Unity thì tất cả các tập tin tài nguyên phải được lưu trữ trong một thư mục con tên là Assets.

### **2.4.2 Scene**

Trong Unity, chúng ta có thể xem Scenes là các màn chơi, cấp độ chơi riêng lẻ, hoặc các vùng của nội dung trò chơi. Ví dụ như Main menu, Options, About …

Bằng cách xây dựng trò chơi với nhiều cảnh, chúng ta sẽ có thể phân phối thời gian tải và thử nghiệm các phần khác nhau của trò chơi riêng lẻ một cách nhanh chóng và chính xác.



Hình 2.0.2: Vị trí lưu Scenes trong Asset của Project

### **2.4.3 Game Object**

Khi một tài nguyên được sử dụng trong một scene, khi đó chúng ta có thể coi tài nguyên này là một “Game Object” mới. Mỗi GameObject phải chứa ít nhất một thành phần, đó là thành phần “Transform”. Transform chứa các phép để biến đổi góc quay, tỷ lệ hay tịnh tiến của đối tượng.

### **2.4.4 Component**

Component là các thành phần trong một Game Object của Unity. Bằng cách đính kèm các thành phần vào cho một đối tượng, chúng ta có thể áp dụng ngay các phần mới của game engine vào đối tượng. Thông thường các thành phần này được Unity xây dựng sẵn như ánh sáng, camera, particle, hiệu ứng vật lý… 

### **2.4.5 Script**

Script là thành phần quan trọng nhất trong Unity, có thể xem scripts như là linh hồn của game. Chúng ta có thể viết kịch bản cho game bằng C#, Java Scripts hoặc Boo (một dẫn xuất của ngôn ngữ Python). Theo nhiều người đã sử dụng Unity thì code bằng C# sẽ giúp game chạy nhanh hơn và giúp kiểm soát code tốt hơn do tất cả các biến phải được khai báo rõ ràng. Mặt khác ngôn ngữ C# rất tiện dụng để lập trình, nên trong thành phẩm Game bắn súng 3D ở chương 4, chúng em dùng ngôn ngữ C# để viết kịch bản cho game. Mỗi file script C# là một class bắt buộc kế thừa từ lớp MonoBehaviour, có tên class phải trùng với tên file script.

Một đoạn script muốn thực thi được thì nó phải được gắn vào một đối tượng.

### **2.4.6 Prefab**

Prefab là một bản sao lưu các vật thể chúng ta đã tạo, bao gồm các kịch bản cho hành động (khởi tạo, di chuyển, hay hủy đối tượng). Ta có thể sử dụng đối tượng này nhiều lần trong trò chơi, và cũng có thể sử dụng lại cho project khác. Prefab cho phép chúng ta lưu trữ các đối tượng, toàn bộ thành phần bên trong và cấu hình hiện tại. 

### **2.4.7** **Material và Shader**

Shader là đoạn script qui định cách thức render của chất liệu trên bề mặt vật thể. Material sử dụng shader để làm chất liệu cho mô hình. Giữa materials và shaders có mối liên hệ với nhau. Shaders qui định các thuộc tính cần để shader làm việc. Còn material cho phép gán hình ảnh vào các thuộc tính đó từ Asset.

### **2.4.8 UnityAPI**

UnityAPI chứa rất nhiều lớp hỗ trợ lập trình game, trong đó có một số lớp quan trọng như :

**MonoBehaviour**: tất cả các script muốn gắn vào một đối tượng game bắt buộc phải kế thừa từ lớp này.

**GameObject**: lớp cha của tất cả các thực thể trong scene.

**Component**: lớp cha của tất cả các thành phần có thể gắn vào đối tượng.

**Transform**: giúp thay đổi vị trí, xoay, biến đổi tỉ lệ mô hình.

**Input**: hỗ trợ lập trình với chuột, cảm ứng đa điểm, cảm biến gia tốc.

**Camera**: giúp lập trình camera.

**Light**: giúp tạo ánh sáng trong game.

**Projector**: giúp chiếu texture lên bề mặt vật thể.

**ParticleEmitter**: hỗ trợ tạo các hiệu ứng particle đẹp mắt.

**Audio**:hỗ trợ lập trình với âm thanh.   
**Animation**: chạy chuyển động của mô hình nhân vật.

**Rigidbody**: giúp tạo hiệu ứng vật lý liên quan đến trọng lực như bóng nẩy, lăn, …

**CharacterController**: giúp điều khiển nhân vật di chuyển theo độ cao địa hình.

**Collider**: hỗ trợ lập trình va chạm giữa các vật thể.

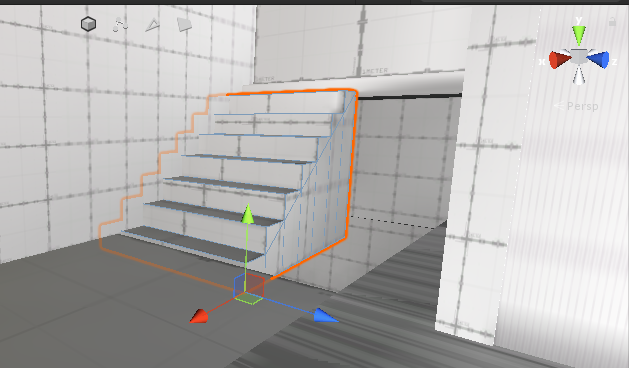
**GUI**: giúp lập trình giao diện người dùng trên Unity.

### **2.4.9 Không gian 3D/2D và các vector trong không gian**

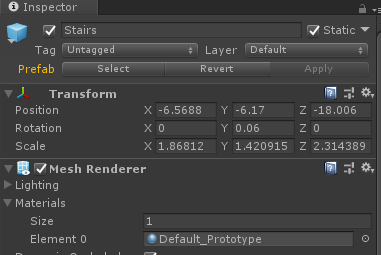
Nhắc về hệ tọa độ không gian 3 chiều chắc hẳn chúng ta đều nhớ môn hình học không gian hoặc đã từng sử dụng một công cụ dựng hình 3D nào đó. Hệ tọa độ không gian 3 chiều bao gồm 3 trục X, Y, Z hay hiểu đơn giản là chiều ngang (X) chiều cao (Y) và chiều sâu (Z). Chúng được kí hiệu theo cú pháp: (X, Y, Z)

Trong không gian 3 chiều, tồn tại 1 điểm gọi là Origin hoặc Word Zero. Đây là điểm có tọa độ (0,0,0). Tất cả tọa độ của các đối tượng tồn tại trong không gian đều có mối liên hệ với điểm này, cách tính này là theo Word Space. Tuy nhiên, để việc tính toán đơn giản, người ta sử dụng thêm một khái niệm gọi là Local Space nhằm xác định tọa độ của đối tượng này so với đối tượng khác. Mối quan hệ này gọi là parent – child.

Cũng giống với không gian 3 chiều, không gian 2 chiều chỉ có gồm 2 trục X, Y còn Z thì mặc định nó sẽ bằng 0.



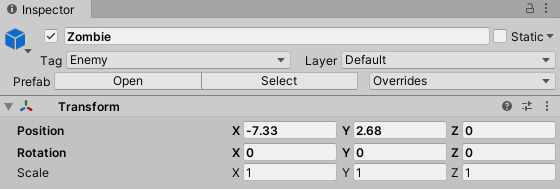
Hình 2.0.3: Một chiếc cầu thang trong không gian 3 chiều (nguồn internet)



Hình 2.0.4: Tương ứng hình 2.3 với Rotation X, Y, Z (nguồn internet)

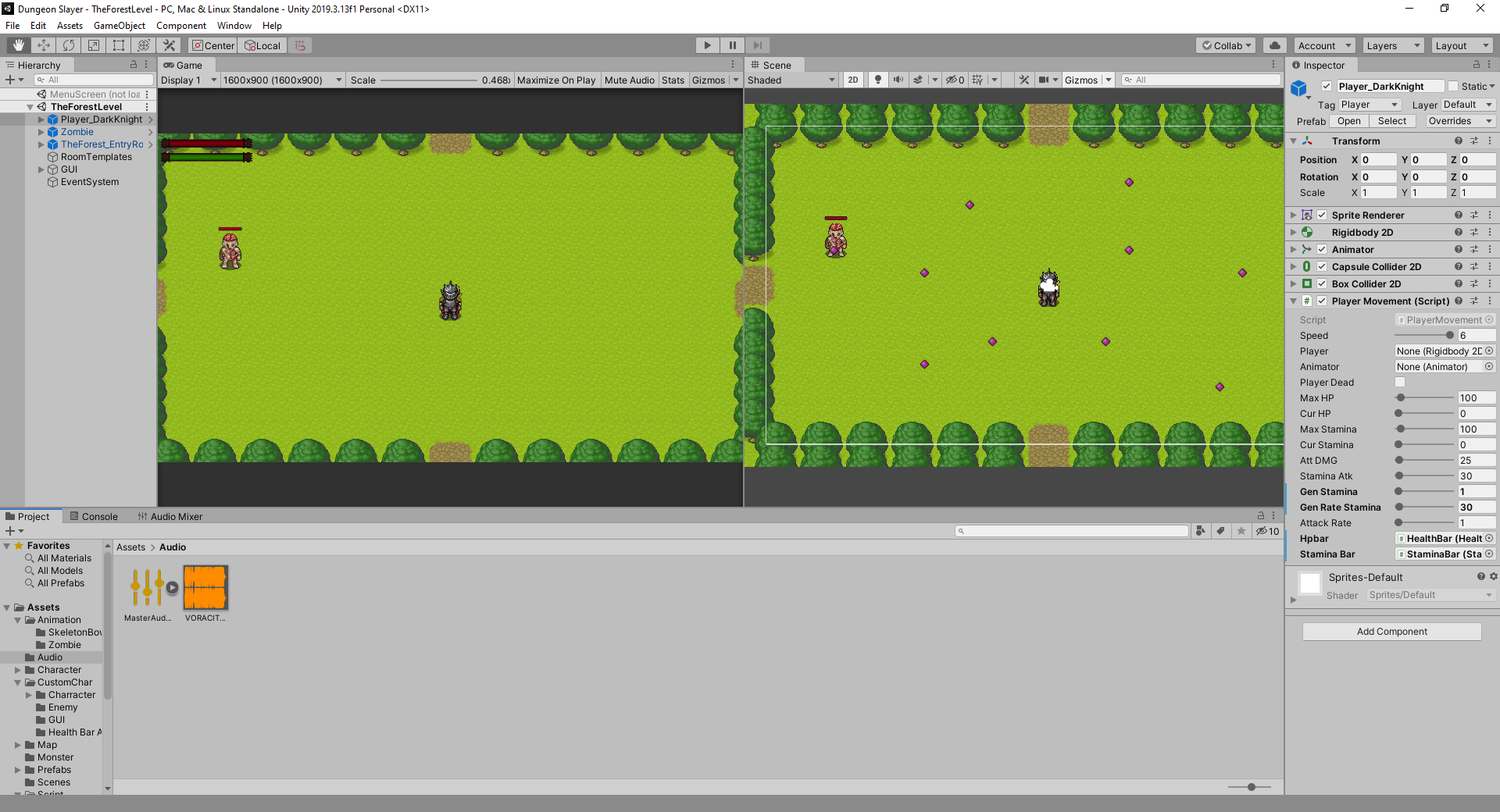


Hình 2.0.5: Một GameObject trong không gian 2 chiều



Hình 2.0.6: Tương ứng hình 2.5 với Rotation X, Y, Z

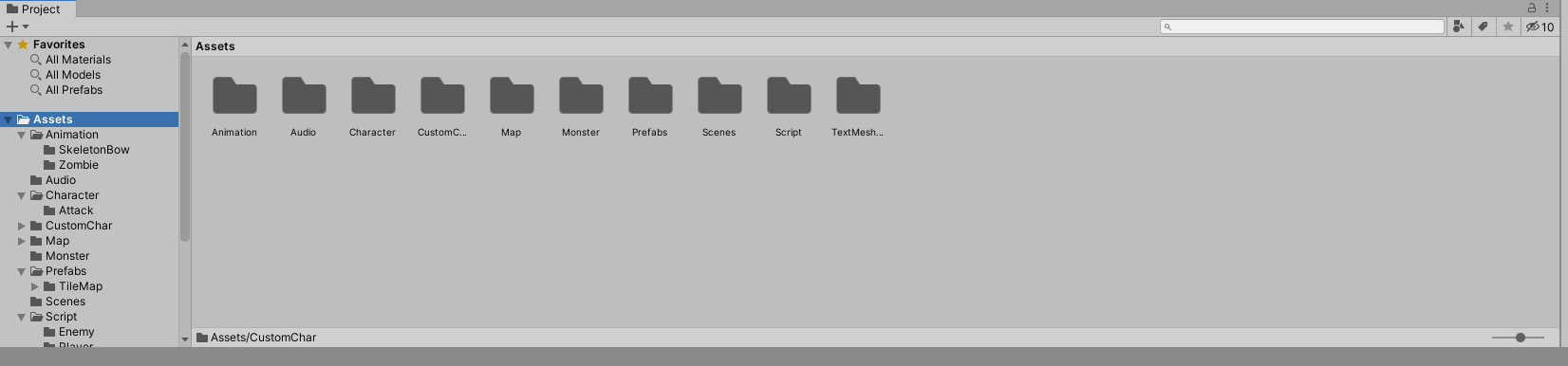
**2.5 Cửa sổ làm việc**



Hình 2.0.7: Giao diện chính của Unity Editor

**2.5.1.**     **Project:**

Mỗi Project của Unity đều chứa một thư mục Assets. Nội dung của thư mục này được hiển thị trong Project View. Đây là nơi chứa tất cả các assets để tạo Game như Scenes, Script, 3D models, Textures, Audio, Prefabs. Chúng ta không nên di chuyển assets bằng cách sử dụng  Window Explorer (hay Finder trong MAC), vì nó sẽ phá vỡ cấu trúc siêu dữ liệu của assets với nhau, và luôn sử dụng Project View để cấu trúc assets

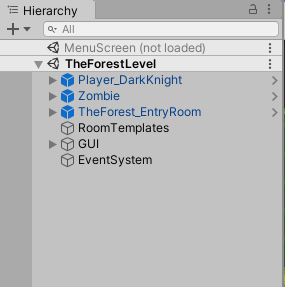


Hình 2.0.8: Cửa sổ Project

Để thêm assets vào project, chúng ta có thể kéo thả bất kỳ file nào vào trong project view hoặc vào công cụ Assets – Import new Assets(Click chuột phải vào Project View). Scenes cũng được lưu trữ trong Project view, và đây là một level độc lập mang tính cá nhân. Chúng ta dễ dàng tạo một assets game trong Unity bằng cách chọn hình tam giác nhỏ nằm bên phải Create trong cửa sổ Project hoặc click chuột phải trong Project View, sau đó chọn assets tương ứng.

**2.5.2. Hierarchy**

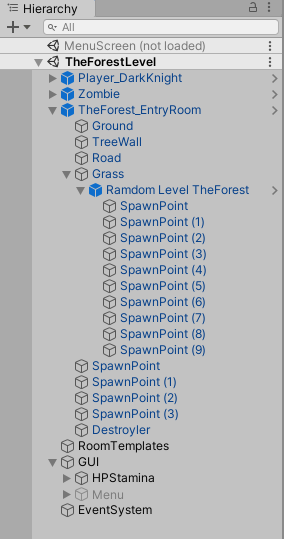
Trong Hierarchy  chứa các GameObject hiện thời, một số có thể trỏ trực tiếp tới những file assets như 3D models, một số khác đại diện cho Prefabs – những đối tượng đã được tùy biến, dùng làm các công việc khác nhau sau này trong Game. Chúng ta có thể chọn và parenting Object trong Hierarchy . Một Object có thể được thêm vào hay loại bỏ trong scene và có thể thấy nó mất đi hay xuất hiện trong Hierarchy.



Hình 2.0.9: Của sổ Hierachy

**2.5.3. Parenting**

          Tức là thư mục chứa hay thư mục gốc, bất kỳ một game object nào muốn là đối tượng con(child) thì ta chỉ việc kéo thả đối tượng đó vào trong đối tượng dự tính làm Parenting trong Hierarchy  và nó sẽ kế thừa chuyển động và quay của parenting.



Hình 2.0.10: Parenting trong Hierachy

**2.5.4 Toolbar**

Toolbar chứa 5 loại điều khiển cơ bản, mỗi loại giữ một vai trò quan trọng trong Editor.

-  Transform Tool: được dung với Scene view, như quay trái , phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.

- Transform Gizmo Toggles: dung cho việc thể hiện Scene view.

- Play/Pause/Step Buttons: dùng cho view game, chạy game ngay trong Editor để kiểm tra.

- Layer Drop-down kiểm soát đối tượng nào đang được thục hiện trong Scene view

- Layout Drop-down kiểm soát sự sắp xếp cảu các Views.



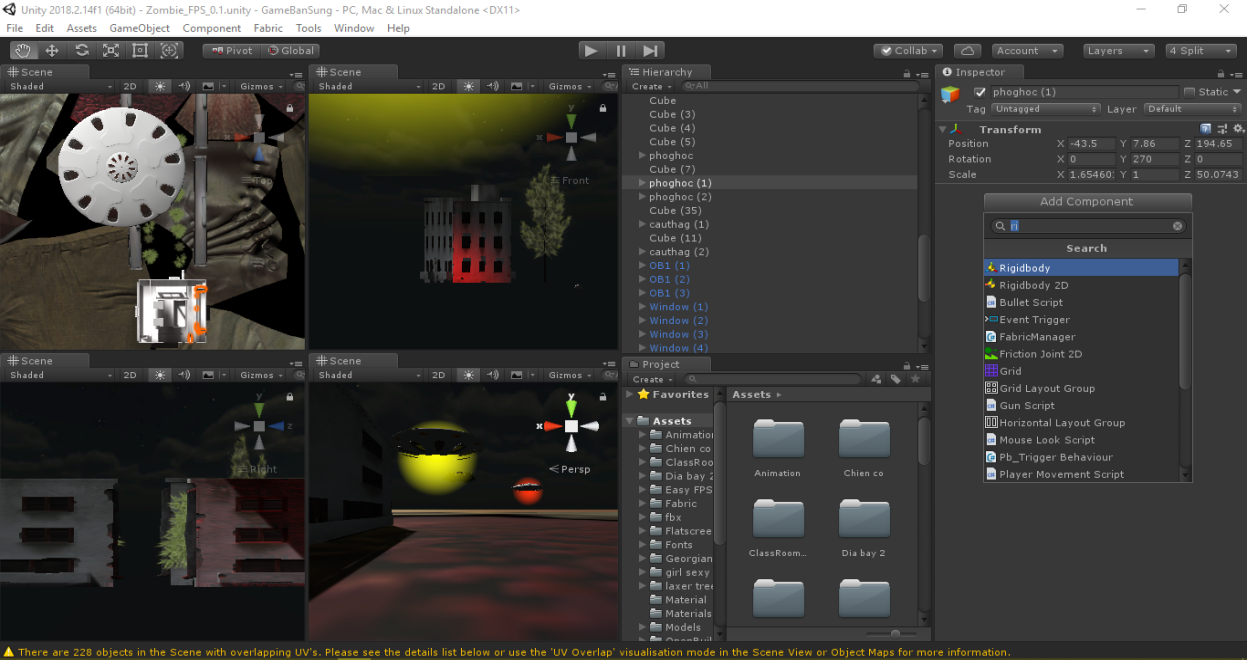
Hình 2.0.11: Toolbar

### **2.5.5 Scene View**

Là nơi Design Game , đối tượng Maneuvering và Importanting trong Scene view (chuyển động và điều khiển) là hai trong số các chức năng quan trọng của Unity, ở góc bên phải của Scene là Scene Gizmo, nó thể hiện hướng nhìn trong không gian của camera trong Scene View hiện thời, cho phép thay đổi góc nhìn trực quan và nhanh chóng.

Click lên các nhánh hình nón để chuyển qua các góc nhìn khác nhau có thể xem ở chế độ Isometric Model(tức ở dạng mặt cắt hai chiều), để chuyển qua chế độ 3D, click vào hình vuông ở giữa hay giữ phím Shift + Click để chuyển đổi chế độ nhìn.

Khi xây dựng một Game, chúng ta sẽ đặt rất nhiều đối tượng vào trong Game, khi đó ta có thể dử dụng các công cụ Transform Tools ở trong Toolbar để di chuyển, xoay, phóng to tu nhỏ từng đối tượng. Nếu họn một đối tượng trong Scene View, xung quanh đối tượng được chọn sẽ có những thay đổi tương ứng với từng chế độ trong Transform Tools.Ta có thể thay đổi đối tượng tùy ý, nếu muốn chính xác, có thể chỉnh chi tiết ở bảng Inspector.



Hình 2.0.12: Các góc nhìn từ các hướng khác nhau (Nguồn Internet)

### **2.5.6 Game View.**

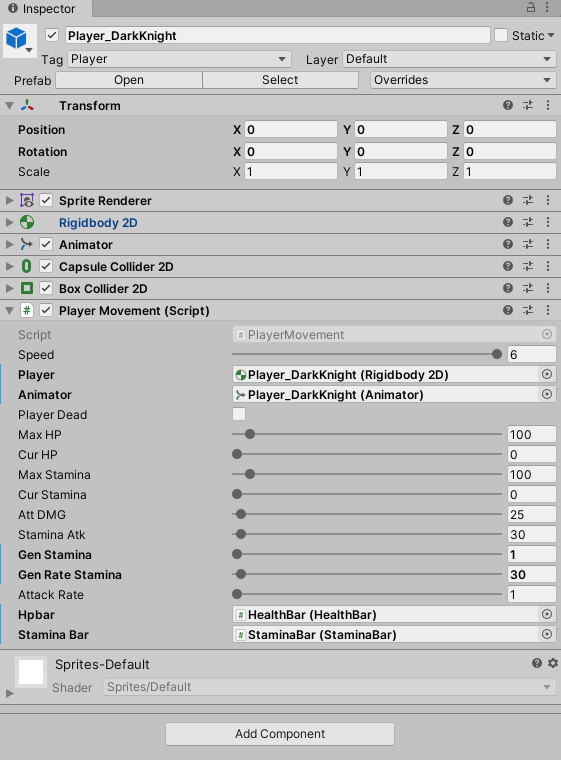
Game View được rendered từ những Camera trong Game. Đó là những gì được nhìn thấy khi hoàn tất, khi Game được xuất bản. Chúng ta sẽ cần ít nhất là một hoặc nhiều hơn số lượng các Camera để quết định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game.

### **2.5.7 Play Mode**

Sử dụng những nút trên Toolbar để điều khiển Editor Play Mode, và xem trước Game sẽ như thế nào khi chơi. Trong chế Play, mọi giá trị thay đổi sẽ được lưu tạm, và bị xóa khi thoát khỏi chế độ play.

### **2.5.8 Inspector**

Games trong Unity được tạo ra bởi tập hợp rất nhiều GameObject, trong đó bao gồm meshes, scripts, âm thanh, hay những đối tượng Graphic như nguồn sáng v..v... Inspector sẽ hiển thị mọi thông tin về đối tượng đang làm việc một cách chi tiết, kể cả những Components được đính kèm và những thuộc tính của nó. Tại đây ta có thể điều chỉnh, thiết lập mọi thông số chức năng của những mối liên kết GameObject-Component.



Hình 2.0.13: Bản Inspector của một GameObject Player

Mọi thuộc tính thể hiện trong Inspector đều có thể được tùy biến một cách trực tiếp. Ngay cả với những biến trong script cũng có thể được hiệu chỉnh mà không cần xem mã. Trong script, nếu chúng ta định nghĩa một giá trị là public cho một kiểu đối tượng (như GameObject hay Transform), ta có thể drag-drop một GameObject hay một Prefab vào trong Inspector để gán giá trị cho nó.

Chúng ta có thể click lên icon hình bánh răng nhỏ bên phải hay click chuột phải lên tên cảu Component để xuất hện context menu dành cho những thiết lập của Component.

Inspector cũng sẽ thể hiện mọi thông số Import Setting của assets đang làm việc.

## 2.6 Lập trình C# trên Visual Studio

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

C# là một ngôn ngữ lập trình thuần hướng đối tượng:

Lập trình hướng đối tượng(tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên. Và để hiểu rõ hơn thì chúng ta sẽ có một chương trình bày về phần này.

C# là một ngôn ngữ ít từ khóa:

C được sử dụng để mô tả thôn# là ngôn ngữ sử dụng giới hạn những từ khóa (gồm khoảng 80 từ khóa và mười mấy kiểu dữ liệu xây dựng sẵn). Nếu bạn nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, lấy ví dụ ngôn ngữ C# làm điển hình nhé. Nếu bạn học sâu về C# bạn sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

Ngoài những đặc điểm trên thì còn một số ưu điểm nổi bật của C#:

* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

C# còn là ngôn ngữ lập trình chính trên Unity được nhiều người sử dụng.

## 2.7 kết luận

Ta đã tìm hiểu được các khái niệm ban đầu, đặc điểm và tầm quan trọng của Ngôn ngữ Lập trình C# trong Unity và hiểu thêm về cách chức năng chính của Framework unity.

# CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM:

## 3.1 Giới thiệu game

### **3.1.3 Bối cảnh**

Nhân vật chính là một mạo hiểm giả bị dịch chuyễn đến một Mê Cung Rừng Rậm Không xác định đầy rảy các quái vật Undead với một nhiệm vụ tiêu diệt toàn bộ Undead trong khu Rừng và sống sót trong khu rừng nguy hiểm này.

Góc nhìn của game sẽ là từ trên xuống (Top-Down) với đồ họa 2D pixel

### **3.1.2 Quy luật chơi**

Người chơi sẻ di chuyễn bằng các phím WASD và tấn công bằng phím J

Người chơi phải luôn luôn giữ máu của mình trên 0 tức là ko để mất hết máu vì sẽ khiến GAME OVER

Nhiễm Vụ chính là càng quét tiêu diệt tất cả quái vật trên bản đồ đề Qua màng sau khi qua 3 màng chơi người chơi hoàn thành game

## 3.2 Thiết kế nhân vật:

### **3.2.1 Player:**



Hình 3.2.1 Animation di chuyễn của Player

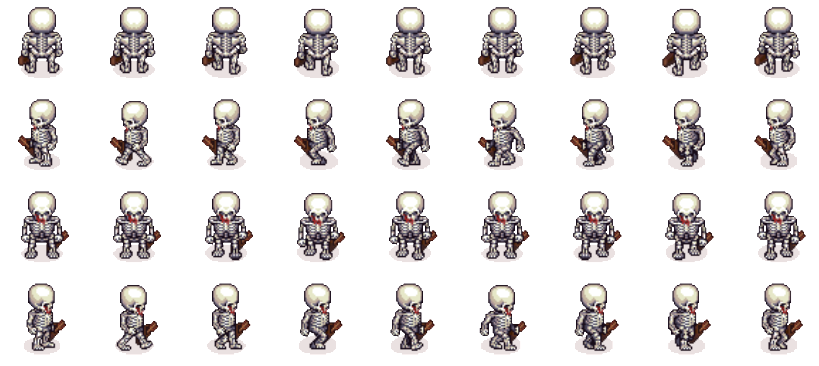
### **3.2.2 Enemy:**

**Zombie:**

****

Hình 3.2.2.1 Animation di chuyển Của Zombie

**Skeleton:**



Hình 3.2.2.2 Animation di chuyển Của Skeleton

## 3.3 Địa hình trông game:

### **3.3.1 Dungeon**

### **3.3.2 Tự Động tạo ngẩu nhiên Màng Chơi**

## 3.4 GUI

## 3.5 Menu Game

# CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ

# 4.1 Kết quả đạt được.

# 4.2 Hướng phát triển & mở rộng đề tài.

# TÀI LIỆU THAM KHẢO

[1]: <https://timviec365.vn/blog/lap-trinh-game-la-gi-new7371.html#bo-oc-sang-tao>

[2]: <https://unity.com/our-company>

[3]: <https://techblog.vn/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay>

[4]: <https://codelearn.io/sharing/lap-trinh-game-voi-unity-nhu-the-nao-phan-1>

[4]:https://freetuts.net/c-sharp-la-gi-tong-quan-ve-c-sharp-1045.html#:~:text=C%23%20(hay%20C%20sharp)%20l%C3%A0,nh%E1%BA%A5t%20l%C3%A0%20C%2B%2B%20v%C3%A0%20Java.

[5]: <https://codelearn.io/sharing/lap-trinh-game-voi-unity-nhu-the-nao-phan-1>