TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**MÔN PHÁT TRIỂN TRÒ CHƠI**

**NINJA X ZOMBIE**

*Giảng viên hướng dẫn*: **ThS. VŨ ĐÌNH HỒNG**

*Người thực hiện*: **TRƯƠNG BẢO HOÀNG – 51603121**

**PHẠM SƠN THỊNH - 51603313**

Khoá  **: 20**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**MÔN PHÁT TRIỂN TRÒ CHƠI**

**NINJA X ZOMBIE**

*Giảng viên hướng dẫn*: **ThS. VŨ ĐÌNH HỒNG**

*Người thực hiện*: **TRƯƠNG BẢO HOÀNG – 51603121**

**PHẠM SƠN THỊNH - 51603313**

Khoá  **: 20**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cám ơn chân thành đến với Thầy Vũ Đình Hồng đã cung cấp kiến thức cho chúng em cũng như hỗ trợ chúng em trong quá trình tìm hiểu đồ án này. Nhờ có sự hướng dẫn nhiệt tình của Thầy mà chúng em mới có thể hoàn thiện được đồ án được báo cáo dưới đây.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

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

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

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

*TP. Hồ Chí Minh, ngày 6 tháng 12 năm 2019*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trương Bảo Hoàng*

*Phạm Sơn Thịnh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Ngày nay, điện thoại di động đã trở thành phương tiện không thể thiếu trong cuộc sống hằng ngày của chúng ta. Điện thoại không còn đơn thuần là dành cho những cuộc gọi hay nhắn tin nữa, nhu cầu sử dụng điện thoại bây giờ rất đa dạng và phong phú, trong đó không thể bỏ qua nhu cầu giải trí. Game trên trên điện thoại di động đã trở thành thú vui số 1 của giới trẻ.

Tuy game trên điện thoại di động chơi không sướng như trên máy tính nhưng tính tiện lợi thì rất rõ, có thể chơi mọi lúc mọi nơi. Trước kia các ứng dụng game trên di động hầu hết là đồ họa 2D, nhưng với sự bùng nổ công nghệ hiện nay dẫn đến các thiết bị di động có đủ sức mạnh để chạy được các ứng dụng đòi hỏi cấu hình cao, các ứng dụng đồ họa phức tạp trong không gian 3 chiều.

Chính vì vậy mà nhu cầu chơi game 3D trên di động trở nên tăng cao, các hãng sản xuất liên tục cho ra đời những thể loại game 3D hình ảnh đẹp mắt, âm thanh sống động chạy trên các thiết bị di động. Song song đó, các game Engine 3D cũng không ngừng phát triển và hỗ trợ làm game cho các thiết bị di động với nhiều hệ điều hành khác nhau như Android, iOS.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc26629351)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc26629352)

[TÓM TẮT iv](#_Toc26629353)

[MỤC LỤC 1](#_Toc26629354)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc26629355)

[CHƯƠNG 1 – TỔNG QUAN VỀ ENGINE UNITY 6](#_Toc26629356)

[1. Unity là gì? 6](#_Toc26629357)

[2. Sơ lược lịch sử hình thành và phát triển của Unity 8](#_Toc26629358)

[2.1 Hình thành 8](#_Toc26629359)

[2.2 Giải thưởng 8](#_Toc26629360)

[2.3 Khách hàng 8](#_Toc26629361)

[3. Tính năng của Engine Unity 9](#_Toc26629362)

[4. Các khái niệm cơ bản trong Unity 9](#_Toc26629363)

[4.1 Asset 9](#_Toc26629364)

[4.2 Scene 10](#_Toc26629366)

[4.3 Game Object 10](#_Toc26629367)

[4.4 Component 11](#_Toc26629368)

[4.5 Script 12](#_Toc26629369)

[4.6 Prefab 14](#_Toc26629370)

[4.7 Material và Shader 14](#_Toc26629371)

[5. Sơ nét về giao diện của Unity 15](#_Toc26629372)

[5.1 Cửa sổ Scene và Hierarchy 15](#_Toc26629373)

[5.2 Inspector 16](#_Toc26629374)

[5.4 Cửa sổ Project 17](#_Toc26629375)

[6. Tổng quan kiến trúc engine Unity trên Android 17](#_Toc26629376)

[6.1 Kiến trúc tổng quan 17](#_Toc26629377)

[6.2 Chu kỳ sống của thành phần script gắn trên đối tượng game 19](#_Toc26629378)

[7. Kết luận 20](#_Toc26629379)

[CHƯƠNG 2 – ỨNG DỤNG GAME PHÁT TRIỂN TRÊN UNITY 20](#_Toc26629380)

[1. Giới thiệu game 20](#_Toc26629381)

[2. Các quy luật chơi chính 21](#_Toc26629382)

[2.1 Di chuyển 21](#_Toc26629383)

[2.2 Tấn công 21](#_Toc26629384)

[2.3 Chiến thắng 22](#_Toc26629385)

[2.4 Tạm dừng 22](#_Toc26629386)

[3. Xử lý di chuyển, va chạm, điểm số, kết thúc màn chơi 22](#_Toc26629387)

[CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 25](#_Toc26629388)

[3.1 Kết luận 25](#_Toc26629389)

[3.2 Định hướng và phát triển 25](#_Toc26629390)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

LP Liferay Portal

FIS FPT Information System

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1: Hình minh họa đa nền 6](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629447)

[Hình 2: Asset trong Unity 9](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629448)

[Hình 3: Các scene của Unity 10](#_Toc26629449)

[Hình 4: Kéo tài nguyên vào Scene để sử dụng 11](#_Toc26629450)

[Hình 5: Các thành phần trong đối tượng Camera 12](#_Toc26629451)

[Hình 6: Cách tạo file script mới 13](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629452)

[Hình 7: Lập trình Unity bằng C# trên Visual Studio 13](#_Toc26629453)

[Hình 8: Một file script đang gắn vào đối tượng 14](#_Toc26629454)

[Hình 9: Material và Shader 14](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629455)

[Hình 10: Giao diện Editor của Unity 15](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629456)

[Hình 11: Chọn đối tượng trong Scene và Hierrarchy 16](#_Toc26629457)

[Hình 12: Cửa sổ Inspector hiển thị thông tin một đối tượng 17](file:///D:\Hoang\Năm%203\ThucTap\BaocaoPTTC.docx#_Toc26629458)

[Hình 13: Tổng quan kiến trúc Unity 18](#_Toc26629459)

[Hình 14: Chu kỳ sống của thành phần script 19](#_Toc26629460)

[Hình 15: Giao diện màn hình chính 21](#_Toc26629461)

[Hình 16: Các động tác của nhân vật 21](#_Toc26629462)

[Hình 17: Các hành động của nhân vật 22](#_Toc26629463)

[Hình 18: Màn hình người chơi chiến thắng 22](#_Toc26629464)

[Hình 19: Màn hình tạm dừng 22](#_Toc26629465)

[Hình 20: Script tiền thưởng 23](#_Toc26629466)

[Hình 21: Script load lại scene 23](#_Toc26629467)

[Hình 22: Script tính máu 24](#_Toc26629468)

[Hình 23: Script chuyển động nhân vật 24](#_Toc26629469)

[Hình 24: Script điều khiển nhân vật 24](#_Toc26629470)

[Hình 25: Script chuyển động tự động của enemy 24](#_Toc26629471)

[Hình 26: Enemy từ di chuyển sang tấn công 25](#_Toc26629472)

[Hình 27: Script hiển thị đồng xu 25](#_Toc26629473)

**DANH MỤC BẢNG**

**No table of figures entries found.**

CHƯƠNG 1 – TỔNG QUAN VỀ ENGINE UNITY

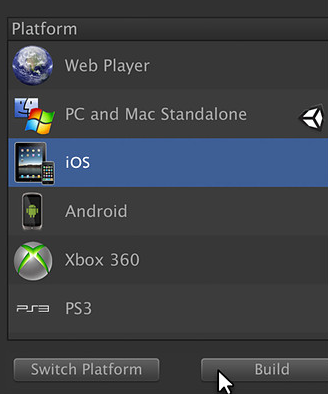
* Nội dung chương này giới thiệu chung về engine Unity và nêu lên các khái niệm cơ bản trong Unity, đặc biệt trong chương này còn trình bày kiến trúc tổng quan của Unity mà nhóm đã xác định được từ những thành phần mà Unity cung cấp.

1. Unity là gì?

Đã qua rồi thời kỳ làm game trên nền Flash căn bản và buồn chán với những chuyển động thật cứng nhắc. Unity 3D mang lại sức mạnh kỳ diệu cho nhân vật mà chúng ta muốn thể hiện sống động hơn trong không gian 3 chiều đầy huyền ảo. Công nghệ cao này tạo ra một bước đột phá mới về sự khác biệt trong công nghệ làm game 3D hiện nay, mang đến cho người chơi 1 cảm giác rất khác lạ và hào hứng trong từng chuyển động, tương lai công nghệ này được áp dụng vào game Việt Nam sẽ mở ra một trang mới trong thế giới game 3D huyền ảo.

Unity 3D được dùng để làm video game 3D, hoặc những nội dung có tính tương tác như thể hiện kiến trúc, hoạt hình 3D thời gian thực. Unity hao hao với Director, Blender game engine, Virtools hay Torque Game Builder trong khía cạnh dùng môi trường đồ họa tích hợp ở quá trình phát triển game là chính.

Unity là một trong những engine được giới làm game không chuyên cực kỳ ưa chuộng bởi khả năng tuyệt vời của nó là phát triển trò chơi đa nền. Trình biên tập có thể chạy trên Windows và Mac OS, và có thể xuất ra game cho Windows, Mac, Wii, iOS, Android. Game cũng có thể chơi trên trình duyệt web thông qua plugin Unity Web Player. Unity mới bổ sung khả năng xuất ra game trên widget cho Mac, và cả Xbox 360, PlayStation 3.



Hình 1: Hình minh họa đa nền

(Nguồn: [http://unity3d.com/](http://unity3d.com/unity/publishing/))

Chỉ với khoản tiền bỏ ra khá khiêm tốn (1.500 USD) là phiên bản pro đã nằm trong tay của chúng ta, dĩ nhiên tại Việt Nam số tiền này vẫn là quá lớn nhưng thật may là đã có phiên bản Unity Free. Tuy nhiên, nhiều tính năng quan trọng (Network) bị cắt giảm nhưng đó không phải là vấn đề quá lớn nếu muốn phát triển một tựa game tầm trung.

Vào năm 2009, Unity nằm trong top 5 game engine tốt nhất cho việc sản xuất game với chỉ sau 4 năm phát triển. Unity đứng thứ 4, xếp sau Unreal Engine 3, Gamebryo Engine (được VTC Studio mua về phát triển SQUAD) và Cry Engine 2. Lượng tài liệu hướng dẫn Unity rất phong phú. Hơn thế nữa nó còn có sẵn một cộng đồng cực lớn với diễn đàn riêng. Bất cứ điều gì không hiểu chúng ta đều có thể thoải mái hỏi và nhận được câu trả lời nhanh chóng, tận tâm.

Quá trình tạo địa hình cũng như truy xuất từ các phần mềm 3DSMax, Maya, Cinema4D... rất nhanh chóng. Sức mạnh và sự tiện lợi của Unity là vô cùng lớn.

* Sức mạnh: Unity có thể tạo ra được nhiều loại game 3D đa dạng, dễ sử dụng với người làm game chưa chuyên nghiệp, chất lượng cao, chạy hầu hết trên các hệ điều hành.
* Sự tiện lợi: nếu chúng ta là một người chuyên dùng 3Dmax, hay Maya hoặc phần mềm mã nguồn mở Blender thì quả là thật tuyệt, chúng ta sẽ có một lợi thế lớn khi viết game trên Unity này, bởi công việc tạo các mô hình 3D sẽ trở lên dễ dàng hơn rất nhiều, việc kết hợp giữa người lập trình và người thiết kế các mô hình sẽ nhanh và hiệu quả hơn. Trong Unity chúng ta có thể import trực tiếp các file mô hình đang thiết kế và sẽ thiết kế hoàn thiện tiếp nếu chưa xong trong khi đó công việc import chỉ diễn ra một lần. Không như việc phải dùng các công cụ khác để thực hiện viết game chúng ta sẽ phải xuất chúng ra một dạng nào đó và mỗi lần sửa lại phần mô hình chúng ta lại phải import lại, và như thế là quá mất thời gian trong việc tạo và chỉnh sửa các mô hình theo ý muốn. Ngoài ra Unity còn cho chúng ta trực tiếp tạo các mô hình nếu muốn. Việc đặt các thuộc tính vật lý trong Unity cũng cực kỳ dễ dàng và hỗ trợ sẵn nhiều chức năng.

1. Sơ lược lịch sử hình thành và phát triển của Unity
   1. Hình thành

Phần lõi của Unity ban đầu được viết bởi Joachim Ante vào năm 2001. Sau đó công ty được hình thành vào năm 2005 và bắt đầu với phiên bản 1.0. Đến năm 2007, Unity được nâng lên phiên bản 2.0. Unity bắt đầu hỗ trợ iPhone vào năm 2008. Vào tháng 6/2010, Unity chính thức hỗ trợ Android và cho ra đời phiên bản 3.0 có hỗ trợ Android vào tháng 9/2010. Có thể thấy tốc độ phát triển của Unity khá nhanh.

* 1. Giải thưởng

Unity đã đoạt được nhiều giải lớn với những giải chính sau:

* Năm 2006, Unity đạt "Best Use of Mac OS X Graphics" tại Apple's WWDC. Đây là lần đầu tiên một công cụ phát triển game đạt được chất lượng do giải thưởng uy tín này đưa ra.
* Năm 2009, Unity Technologies có tên trong "Top 5 công ty game của năm" do Gamasutra tổ chức.
* Năm 2010, Unity đoạt giải Best Engine Finalist do Develop Magazine bình chọn, giải Technology Innovation Award của Wall Street Journal ở thể loại phần mềm
  1. Khách hàng

Unity được trên 250.000 người đăng ký sử dụng gồm Bigpoint, Cartoon Network, Coca-Cola, Disney, Electronic Arts, LEGO, Microsoft, NASA, Ubisoft, Warner Bros, các hãng phim lớn nhỏ, các chuyên gia độc lập, sinh viên và những người đam mê.

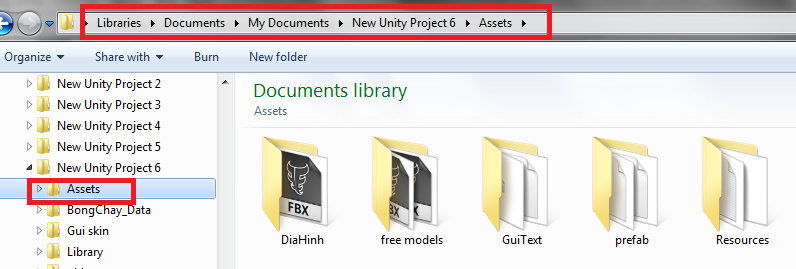
1. Tính năng của Engine Unity

Môi trường phát triển được tích hợp với tính năng kế thừa, khả năng chỉnh sửa đồ họa, chức năng kiểm tra chi tiết, và đặc biệt tính năng xem trước game ngay trong lúc xây dựng (live game preview).

* Triển khai được trên nhiều nền tảng:
  + Chương trình độc lập trên Windows và Mac OS.
  + Trên web, thông qua Unity Web Player plugin cho Internet Explorer, Firefox, Safari, Opera, Chrome, cho cả Windows và Mac OS.
  + Trên Mac OS Dashboard widget.
  + Cho Nintendo Wii (cần mua license thêm.)
  + Cho iPhone, iPad application (cần mua license thêm.)
  + Cho Google Android (cần mua license thêm.)
  + Cho Microsoft Xbox 360 (cần mua license thêm.)
  + Cho Sony PlayStation 3 (cần mua license thêm.)
* Tài nguyên (model, âm thanh, hình ảnh, ...) được tải vào trong Unity và tự động cập nhật nếu tài nguyên có sự thay đổi. Unity hỗ trợ các kiểu định dạng từ 3DS Max, Maya, Blender, Cinema 4D và Cheetah3D.
* Graphics engine sử dụng Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iPhone OS), và các API khác trên Wii.
* Hỗ trợ bump mapping, reflection mapping, parallax mapping, Screen Space Ambient Occlusion v...v...
* Unity Asset Server: Đây là một tính năng khá mới của Unity, theo đó Unity sẽ cung cấp một hệ thống quản lý theo dạng phiên bản cho tất cả asset và cả script. Đây là một kho chứa các tài nguyên cần thiết cho việc làm game. Khi import cũng như sửa chữa, trạng thái của asset ngay lập tức được cập nhật. Server chạy trên database opensource PostgreSQL và có thể truy cập trên cả Mac lẫn Windows, Linux. Asset Server đòi hỏi một khoản phí phụ trội là $499 cho mỗi bản copy Unity, và một license Unity Pro.

1. Các khái niệm cơ bản trong Unity
   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.

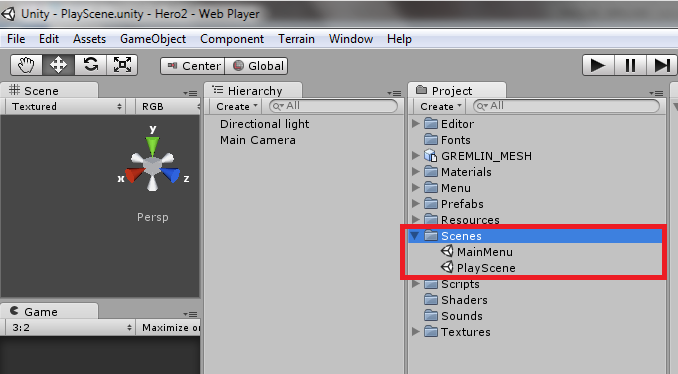


Hình 2: Asset trong Unity

* 1. 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 …

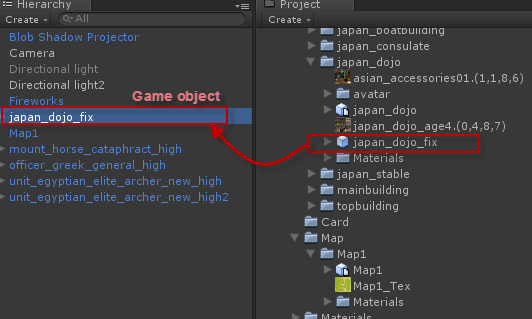
Bằng cách xây dựng trò chơi với nhiều cảnh, chúng ta sẽ có thể phân phối thời gian tải và thử nghiệm các phần khác nhau của trò chơi riêng lẻ một cách nhanh chóng và chính xác.



Hình 3: Các scene của Unity

* 1. Game Object

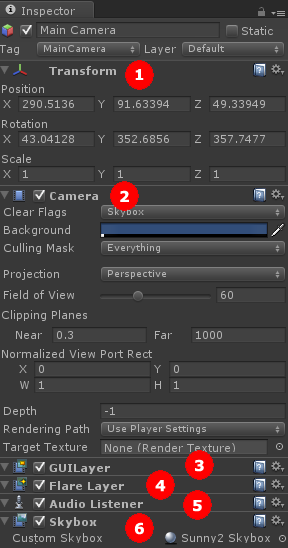
Khi một tài nguyên được sử dụng trong một scene, khi đó chúng ta có thể coi tài nguyên này là một “Game Object” mới. Mỗi GameObject phải chứa ít nhất một thành phần, đó là thành phần “Transform”. Transform chứa các phép để biến đổi góc quay, tỷ lệ hay tịnh tiến của đối tượng. Từ đây trong báo cáo này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



Hình 4: Kéo tài nguyên vào Scene để sử dụng

* 1. Component

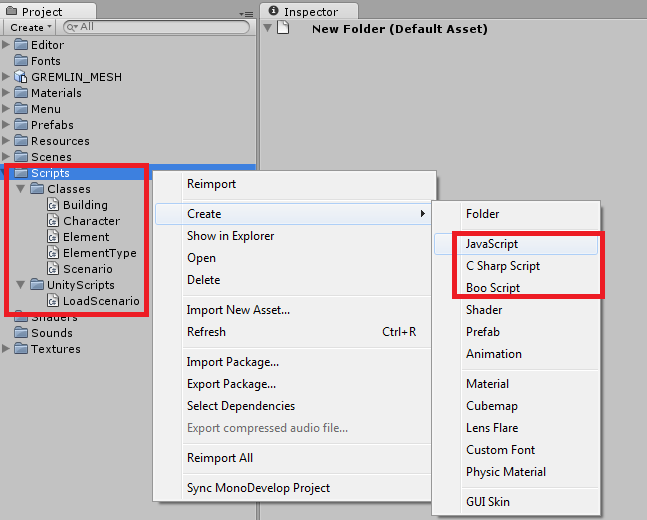
Component là các thành phần trong một Game Object của Unity. Bằng cách đính kèm các thành phần vào cho một đối tượng, chúng ta có thể áp dụng ngay các phần mới của game engine vào đối tượng. Thông thường các thành phần này được Unity xây dựng sẵn như ánh sáng, camera, particle, hiệu ứng vật lý…



Hình 5: Các thành phần trong đối tượng Camera

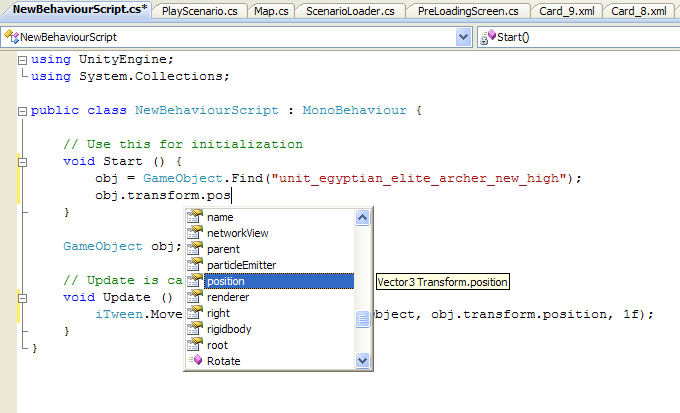
* 1. Script

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 luận văn này, 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.



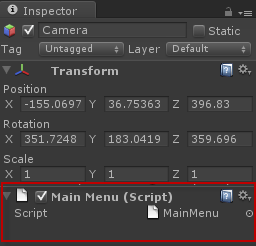
Hình 6: Cách tạo file script mới

Giao diện code C# rất quen thuộc khi edit trên Visual Studio 2015



Hình 7: Lập trình Unity bằng C# trên Visual Studio

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



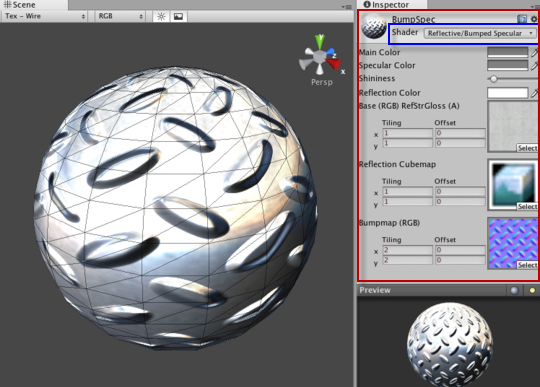
Hình 8: Một file script đang gắn vào đối tượng

* 1. Prefab

Hãy tưởng tượng Prefab là một cái thùng rỗng, mà bên trong nó chúng ta có thể chứa đựng các thành phần hay đối tượng khác nhau, chúng ta có thể viết kịch bản cho hành động của Prefab (khởi tạo, di chuyển, hay hủy đối tượng). Chúng 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.

* 1. Material và Shader

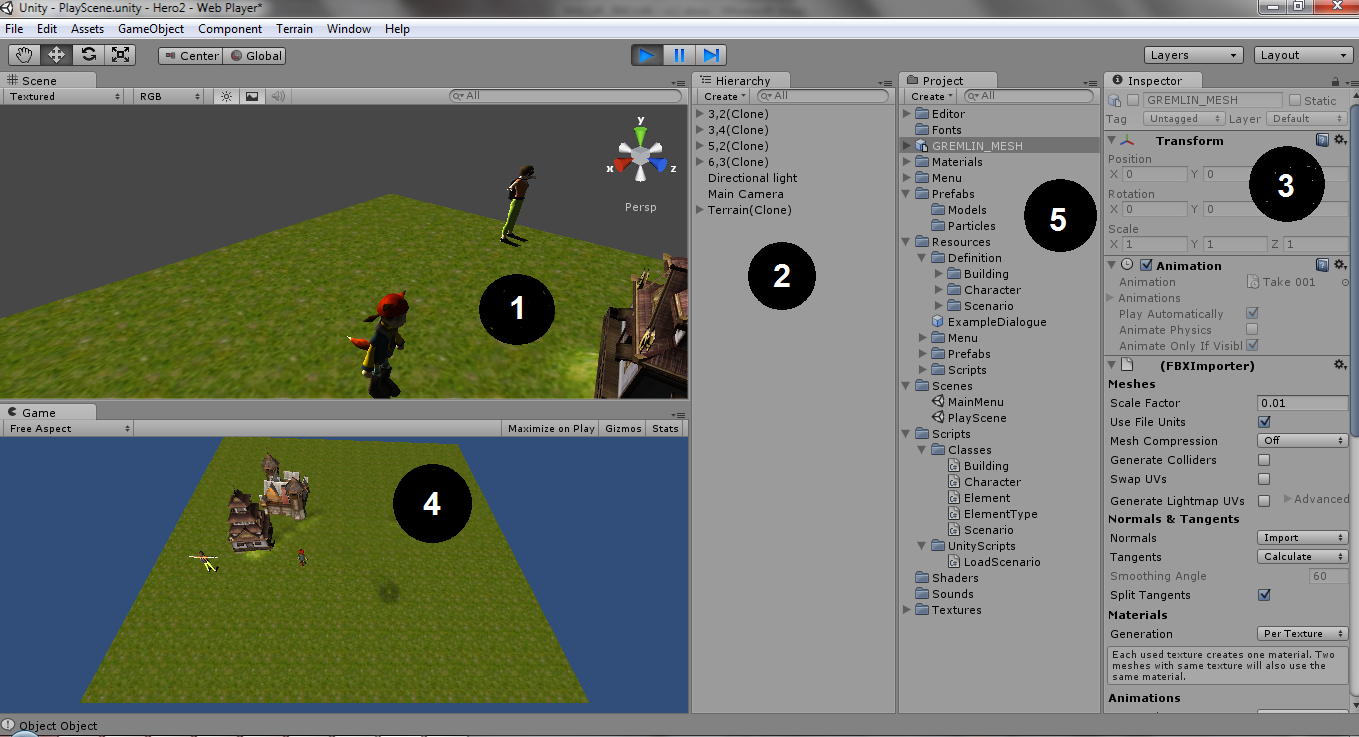
Shader là đoạn script qui định cách thức render của chất liệu trên bề mặt vật thể. Material sử dụng shader để làm chất liệu cho mô hình. Giữa materials và shaders có mối liên hệ với nhau. Shaders qui định các thuộc tính cần để shader làm việc. Còn material cho phép gán hình ảnh vào các thuộc tính đó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



Hình 9: Material và Shader

1. Sơ nét về giao diện của Unity

Giao diện Unity, giống như nhiều môi trường làm việc khác, layout có thể tùy chỉnh. Layout của Unity bao gồm nhiều tab khác nhau và có thể bật tắt. Chúng ta hãy xem xét một cách bố trí giao diện Unity điển hình:



Hình 10: Giao diện Editor của Unity

Như hình trên chúng ta thấy có 5 khung khác nhau:

* + Scene [1] – nơi xây dựng trò chơi
  + Hierarchy [2] – danh sách các GameObject trong một cảnh game
  + Inspector [3] – màn hình cài đặt cho tài nguyên/đối tượng đang được chọn
  + Game [4] – cửa sổ xem trước game, chỉ hoạt động ở chế độ chơi (khi nhấn Play)
  + Project [5] – danh sách các tài nguyên trong project, đóng vai trò như một thư viện.
  1. Cửa sổ Scene và Hierarchy

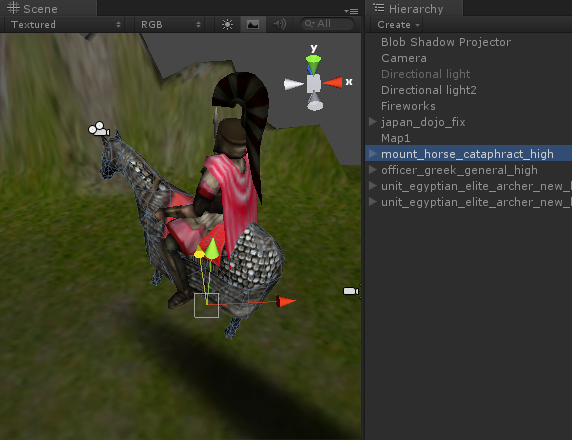
Cửa sổ Scene là nơi chúng ta sẽ xây dựng toàn bộ các đối tượng trong game. Cửa sổ cung cấp nhiều góc nhìn khác nhau, có thể nhìn dạng phối cảnh hoặc dạng song song. Chúng ta có thể kéo thả đối tượng trên cửa sổ này, di chuyển, xoay…



Cửa sổ Scene cũng kèm theo bốn nút điều khiển tiện lợi như hình trên. Truy cập từ bàn phím bằng cách sử dụng các phím Q, W, E, và R. Các phím thực hiện các hoạt động sau đây:

* + Công cụ bàn tay [Q]: công cụ này cho phép di chuyển trong cửa sổ Scene, xoay góc nhìn, phóng to, thu nhỏ góc nhìn.
  + Công cụ di chuyển [W]: Công cụ này dùng để di chuyển một đối tượng.
  + Công cụ xoay [E]: Công cụ này cho phép chúng ta xoay nhân vật theo một trục nào đó trong không gian.
  + Công cụ tỷ lệ [R]: Công cụ này cho phép chúng ta tăng giảm tỷ lệ kích thước của đối tượng.

Khi chọn một đối tượng trong cửa sổ Scene, đối tượng này sẽ được tự động chọn trong cửa sổ Hierarchy và ngược lại.



Hình 11: Chọn đối tượng trong Scene và Hierrarchy

* 1. Inspector

Inspector sẽ hiển thị tất cả thông tin, các thành phần trong đối tượng game đang chọn, và cho phép điều chỉnh các biến của các thành phần này. Có thể xem cửa sổ này như cửa sổ Properties khi design giao diện Winform trên Visual Studio

* 1. Cửa sổ game

Hình 12: Cửa sổ Inspector hiển thị thông tin một đối tượng

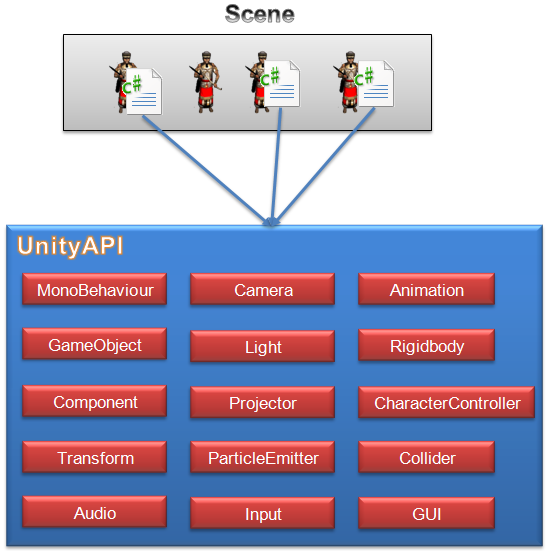
Cửa sổ này sẽ hiển thị những gì có trong cửa sổ Scene và sẽ hoạt động khi nhấn nút Play. Trong cửa sổ này chúng ta có thể chọn các kích cỡ hiển thị khác nhau để build cho các loại máy khác nhau. Chúng ta có thể chơi thử game trên cửa sổ này khi đã nhấn nút Play. Lưu ý rằng khi cửa sổ này hoạt động rồi thì mọi chỉnh sửa trên cửa sổ Scene và cài đặt cho các đối tượng chỉ là tạm thời và khi nhấn nút Stop, cửa sổ này về lại trạng thái tĩnh thì mọi chỉnh sửa trước đó là không còn.

* 1. Cửa sổ Project

Cửa sổ Project thể hiện nội dung bên trong thư mục Assets của project chúng ta. Khi thêm tài nguyên vào thư mục Assets ngay lập tức chúng sẽ tự động được cập nhập vào project Unity của chúng ta.

1. Tổng quan kiến trúc engine Unity trên Android
   1. Kiến trúc tổng quan

* Engine Unity hỗ trợ cho chúng ta UnityAPI để viết script game. UnityAPI là API lập trình game trong Unity rất mạnh. UnityAPI chứa các đối tượng và phương thức hỗ trợ hầu hết các đối tượng và các loại thành phần trong Unity.
* Trong một scene thường có nhiều đối tượng game. Mỗi đối tượng này có thể có hoặc không có đoạn script nào gắn lên đó. Nếu muốn gắn script vào đối tượng, ta bắt buộc phải kế thừa class đó từ lớp **MonoBehaviour** của UnityAPI và tên class phải trùng với tên file script. Mỗi script khi gắn lên đối tượng game đều được đối tượng game xem như một thành phần bên trong và được cấp phát vùng nhớ khi chạy game.

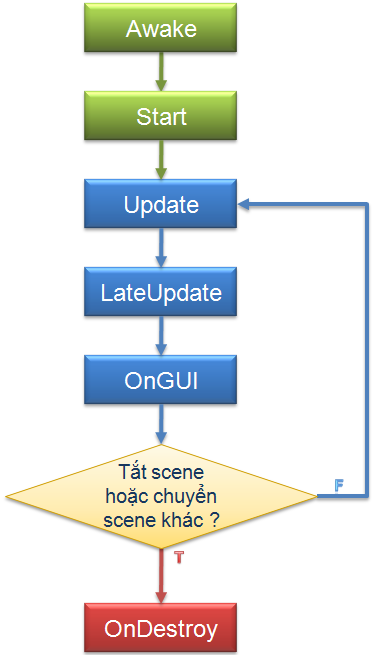


Hình 13: Tổng quan kiến trúc Unity

Bên trong 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.
  1. Chu kỳ sống của thành phần script gắn trên đối tượng game

Chu kỳ sống của một thành phần script được tính kể từ khi scene được chạy cho đến lúc scene bị tắt hoặc chuyển sang scene khác.



Hình 14: Chu kỳ sống của thành phần script

Tùy theo trạng thái của scene mà sự kiện tương ứng sẽ được gọi.

* + **Awake**: được gọi khi script được load xong
  + **Start**: được gọi khi script được load xong nếu thành phần script đó không bị disable.
  + **Update**: đây là sự kiện thường sử dụng nhất và được gọi liên tục từng frame để vẽ lại màn hình.
  + **LateUpdate**: sự kiện này chạy sau sự kiện Update và được gọi liên tục từng frame.
  + **OnGUI**: sự kiện này dùng để vẽ GUI và được gọi liên tục từng frame, chỉ trong sự kiện này ta mới có thể sử dụng các lớp hỗ trợ tạo giao diện người dùng của UnityAPI.
  + **OnDestroy**: được gọi khi thành phần script bị hủy khỏi bộ nhớ.

1. Kết luận

Trong chương này chúng ta đã tìm hiểu tổng quan về Unity và nắm rõ các khái niệm cơ bản trong Unity. Chương tiếp theo ta sẽ tìm hiểu rõ hơn về cách lập trình game trên Unity.

CHƯƠNG 2 – ỨNG DỤNG GAME PHÁT TRIỂN TRÊN UNITY

* Nội dung chương này mô tả các khái niệm, quy luật chơi trong dòng game theo lượt mà chúng em xây dựng, đồng thời nêu lên kiến trúc của game.

1. Giới thiệu game

Bạn là fan của Ninja, một nhân vật anh hùng nửa chính nửa tà, có nhiều vũ khí độc mà lạ. Bên cạnh đó, Ninja còn có nhiều khả năng phi thường với các chiến thuật biến thân, kĩ năng thần bí, tinh thần bất khuất nhưng lại hành tung bí ẩn. Vậy thì Ninja X Zombie là một tựa game mà bạn không thể bỏ qua.

Ninja X Zombie là một game nhập vai rất hay đang được rất nhiều các bạn trẻ quan tâm với rất nhiều lượt chơi và lượt tải về. Đến với game Ninja X Zombie bạn không chỉ có được những phút giây giải trí tuyệt vời mà còn có khả năng phát triển trí tuệ, được  tham gia trải nghiệm mình trong những thử thách mang đầy tính nhân văn cũng như tính tư duy chiến lược cao.Các quy luật chơi chính



Hình 15: Giao diện màn hình chính

1. Các quy luật chơi chính
   1. Di chuyển

Người chơi sẽ sử dụng các nút trái, phải để điều khiển nhân vật di chuyển theo ý muốn của mình, nút space để nhảy và nút ctrl để lướt trên cỏ hoặc né các đạn từ zombie. Trong quá trình chơi, người chơi có thể bị các zombie tấn công.



Hình 16: Các động tác của nhân vật

* 1. Tấn công

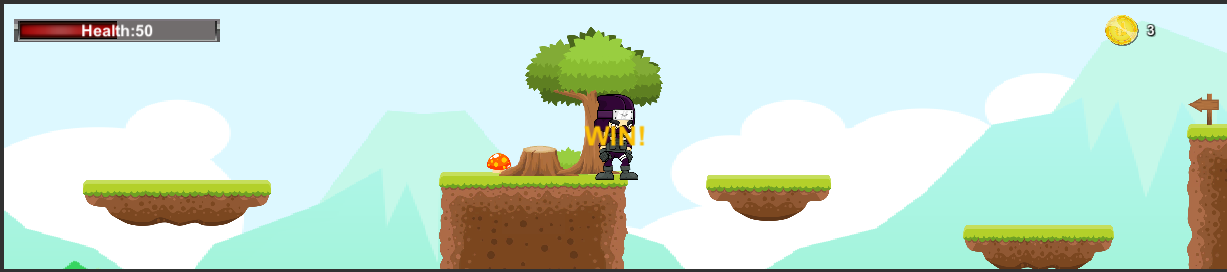
Người chơi sẽ sử dụng nút Z để tấn công zombie bằng kiếm tầm gần và nút X để tấn công tầm xa với Kunai.



Hình 17: Các hành động của nhân vật

* 1. Chiến thắng

Người chơi phải đánh bại hết tất cả zombie, nhặt các đồng tiền rớt ra từ chúng để chiến thắng.

Hình 18: Màn hình người chơi chiến thắng

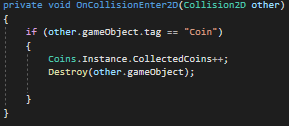
* 1. Tạm dừng



Hình 19: Màn hình tạm dừng

1. Xử lý di chuyển, va chạm, điểm số, kết thúc màn chơi

* Khi chạm đồng tiền, số tiền thưởng tăng lên và biến mất



Hình 20: Script tiền thưởng

* Khi player chết sẽ load lại scene chơi từ đầu



Hình 21: Script load lại scene

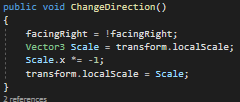
* Khi bị đánh, người chơi bị trừ 10 máu. Nếu hết máu, đợi vài giây để được hồi sinh.





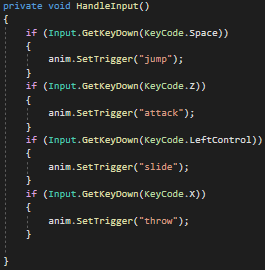
Hình 22: Script tính máu

* Các hàm giúp hình ảnh của nhân vật quay trái quay phải được



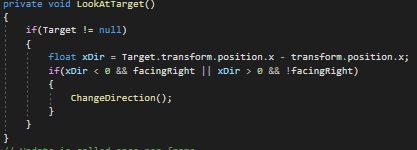
Hình 23: Script chuyển động nhân vật

* Các nút dung để điều khiển nhân vật



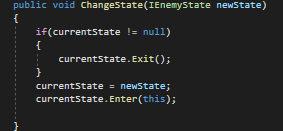
Hình 24: Script điều khiển nhân vật

* Giúp cho enemy có thể quay trái quay phải tự động khi chưa bị tác động bởi player



Hình 25: Script chuyển động tự động của enemy

* Chuyển trạng thái của enemy



Hình 26: Enemy từ di chuyển sang tấn công

* Khi enemy bị đánh sẽ bị giảm máu. Nếu enemy chết sẽ rớt đồng xu cho người chơi



Hình 27: Script hiển thị đồng xu

CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

* 1. Kết luận

Qua trò chơi demo chúng em đã từng bước tiếp cận được với công nghệ làm game, cách thức cũng như phương pháp, nền tảng tối ưu, biết cách kết hợp các phần mềm để tạo ra một mô hình vật thể, sản phẩm. Trong quá trình xây dựng game chúng em nhận ra còn nhiều vấn đề phát sinh, nhiều khó khăn hơn mình suy nghĩ, rất nhiều bug trong code C# cần fix, đôi khi 1 lỗi nhỏ làm hỏng cả hệ thống, vì vậy khi thiết kế game chúng ta nên chia nhỏ từng công việc và module riêng ra để dễ xây dựng và gắn kết khi các thành phần đã chạy ổn định. Viết game rất tốn thơi gian cho các công việc như design các nhân vật, bản đồ, đội ngũ xây dựng phải đông đảo hơn, cần sự kiên trì và sáng tạo hơn.

* 1. Định hướng và phát triển
* Thiết kế thêm và hoàn thiện về player (vũ khí và các nâng cấp).
* Thiết kế thêm và thêm chức năng cho Enemy (phù hợp với bản đồ).
* Thiết kế thêm các màn chơi. - Chức năng unlock vũ khí, điều chỉnh độ khó của màn chơi. - Pause và resume game.
* Các chức năng điều chỉnh options (âm thanh, ánh sáng).

**TÀI LIỆU THAM KHẢO**

1. Khóa học các công cụ của Adobe – DayNhauHoc, HowKteam, KhoaPham, CodeDao, …
2. <https://www.youtube.com/playlist?list=PLXuZVK_0K_6VXcSajfFbXDXndb6AdBLO>
3. https://www.youtube.com/playlist?list=PL2614K6kEvHC37KsgRtqnydn1vQd8uCPy