

**BỘ LAO ĐỘNG – THƯƠNG BINH VÀ XÃ HỘI**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG**

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN CƠ SỞ NGÀNH

**Lập trình game bắn súng 3D với Unity engine**

**và ngôn ngữ C#**



**Giảng viên hướng dẫn: ThS. Trần Thị Cẩm Tú**

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

**MSSV:16004074**

**Huỳnh Văn Thông**

**MSSV:16004076**

**Lớp: ĐH.CNTT2016**

**HỌC KÌ I**

**Năm Học 2018- 2019**

*Đồ án cơ sở ngành - Trường đại học sư phạm kỹ thuật Vĩnh Long*

**TRƯỜNG ĐHSPKT VĨNH LONG CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Công nghệ thông tin Độc lập - Tự do -H ạnh phúc**



PHIẾU GIAO ĐỀ TÀI

ĐỒ ÁN CƠ SỞ NGÀNH

**1. Họ và tên sinh viên/ nhóm sinh viên được giao đề tài:**

- Huỳnh Văn Thông MSSV: 16004076 Lớp ĐH - CNTT2016

- Nguyễn Văn Thành MSSV: 16004074 Lớp ĐH - CNTT2016

1. **Tên đề tài:**

Lập trình game bắn súng 3D với Unity engine và ngôn ngữ C#

1. **Nội dung nhiệm vụ:**

- Tìm hiểu về ứng dụng làm game Unity3D, các phần mềm hỗ trợ để lập trình một game hoàn chỉnh

- Tìm hiểu sâu hơn về ngôn ngữ C# trên nền tảng Unity engine trong công nghệ làm game.

- Thiết kế ra game bắn súng 3D góc nhìn thứ nhất

1. **Kết quả tối thiểu phải có:**

- Thiết kế một sản phẩm game 3D hoàn chỉnh

1. **Dự kiến kinh phí (Nếu có):**

….………………………………………………….……………………………………

….……………………………………………………………………………………….

….………………………………………………………………………………………

….………………………………………………………………………………………

**Ngày giao đề**…………………………..

**Ngày nộp** ...……………………………

**LỜI NHÂN XÉT**

Nhận xét và đánh giá điểm của người hướng dẩn:

**-Ý thức thực hiện:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Nội dung thực hiện:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Hình thức trình bày:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Tổng hợp kết quả:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

Vĩnh Long, ngày…tháng…… năm 2019

Người hướng dẫn

*(Kí và ghi rõ họ tên)*

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại học Sư phạm kỹ thuật Vĩnh Long đã tạo điều kiện tốt cho chúng em thực hiện đề tài này!

Cùng với sự giúp đỡ hướng dẫn tận tính của Thạc sĩ – Giảng viên – cô Trần Thị Cẩm Tú chúng em đã phát triển nên Game bắn súng 3D phiên bản Halloween lấy bối cảnh (MAP) ngay khuôn viên trường đại học của chúng ta. Hứa hẹn sẽ mang đến trải nghiệm mới lạ, giúp người chơi thư giãn, giải trí sau những giờ học tập lao động căng thẳng. Game còn là công cụ rèn luyện kỹ năng phản xạ, tính tập trung, khả năng tính toán suy nghĩ để có thể hoàn thành thử thách khó nhất của trò chơi.

Do thời gian còn hạn hẹp cũng như chưa có kinh nghiệm làm một game lớn, nền tảng Framework mới xuất hiện trong những năm gần đây, chương trình thành phẩm của chúng em có thể còn tồn tại lỗi không mong muốn. Rất mong nhận được góp ý của cô để chúng em hoàn thiện hơn và rút kinh nghiệm cho những đồ án, dự án lớn hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn cô và chúc cô nhiều sức khỏe!

**MỤC LỤC**

[CHƯƠNG 1 TỔNG QUAN 1](#_Toc533551257)

[1.1. Lý do chọn đề tài 1](#_Toc533551258)

[1.2. Đối tượng nghiên cứu 1](#_Toc533551259)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 2](#_Toc533551260)

[2.1. Tổng quan về Unity engine 2](#_Toc533551261)

[2.2. Các khái niệm trong Unity 4](#_Toc533551265)

[2.3. Cửa sổ làm việc 8](#_Toc533551270)

[2.4. Các phần mềm hỗ trợ khác 12](#_Toc533551271)

[2.5. Game và hướng phát triển 13](#_Toc533551274)

[CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME 15](#_Toc533551275)

[3.1. Ý tưởng 15](#_Toc533551276)

[3.2. Tạo bản đồ và nhân vật 16](#_Toc533551277)

[3.3. Xử lý di chuyển, va chạm, điểm số, kết thúc màn chơi 19](#_Toc533551280)

[3.4. Tiến trình của Game 20](#_Toc533551281)

[3.5. Hoạt họa và cảm nhận 21](#_Toc533551282)

[CHƯƠNG 4 CHƯƠNG TRÌNH DEMO 22](#_Toc533551283)

[4.1. Gameplay 22](#_Toc533551285)

[4.2. Các chức năng cơ bản 22](#_Toc533551286)

[CHƯƠNG 5 KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 24](#_Toc533551288)

[5.1 Kết luận 24](#_Toc533551289)

[5.2. Định hướng phát triển 24](#_Toc533551290)

[CHƯƠNG 6 TÀI LIỆU THAM KHẢO – PHỤ LỤC 25](#_Toc533551291)

[6.1. Tài liệu tham khảo 25](#_Toc533551292)

[6.2. Phụ lục 25](#_Toc533551293)

# CHƯƠNG 1 TỔNG QUAN

* 1. **Lý do chọn đề tài**

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 C#

* 1. **Đối tượng nghiên cứu**

Nền tảng unity engine

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

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

* 1. **Tổng quan về Unity engine**
     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 (Android, IOS ) và trên websites.



Unity3D là phần mềm làm games trực tiếp theo thời gian thực, mà không cần render, cho phép người  design game có thể thiết kế InterfaceGraphic, map hay character,… từ một phần mềm thứ 2 (thường là các phần mềm thiết kế đồ họa chuyên nghiệp như 3Dsmax, Blender, Maya, XSL,Cinema4D,Cheetah3D, Modo, Autodesk FBX, LightWave…) sau đó chỉ việc import nó vào trong Unity với định dạng của tập tin là \*.FBX hay \*.dae, \*.3DS, \*.dxf và \*.obj

### **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 nhiều năm phát triển, 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. Phiên bản mới nhất hiện nay là 2018.2 với nhiều tính năng nổi trội

### **Một số thống kê về Unity:**

Tính đến quý 3 năm 2016 đã có 5 tỉ lượt download game và ứng dụng được phát triển bởi Unity

2,4 tỉ thiết bị di động đã từng tải ít nhất 1 ứng dụng bởi unity.

Trong top 1000 game Mobile miễn phí thì số lượng game tạo ra bởi Unity chiếm tới 34%

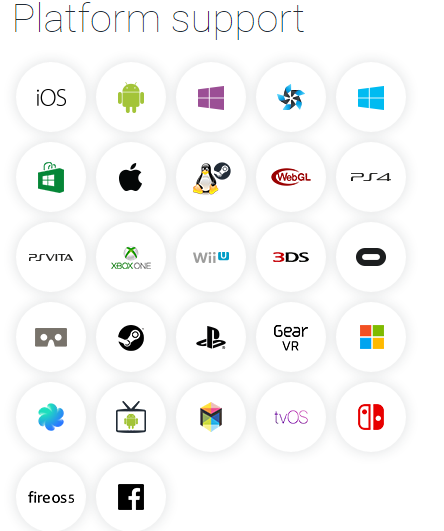
Số lượng người dùng (gamer) của Unity đạt tới con số 770 triệu, trong khi đó số người thường xuyên sử dụng Twitter là 310 triệu người.

*(nguồn: thống kê từ internet)*

### **Ưu điểm của Unity:**

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

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



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

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

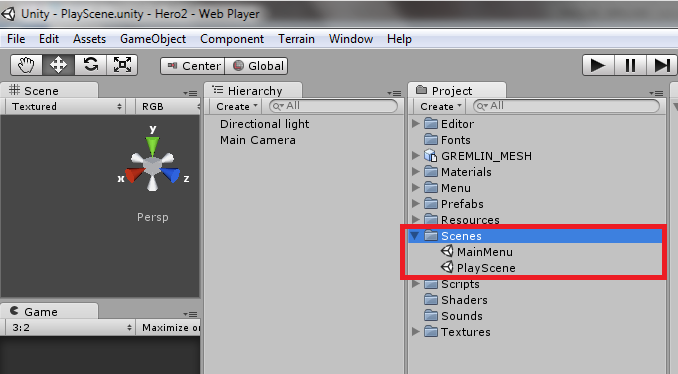
* 1. **Các khái niệm trong Unity**

### **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.   
**2.2.2. 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.



### **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.

* + 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… 

### **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 .

* + 1. **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. 

### **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.

* + 1. **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.

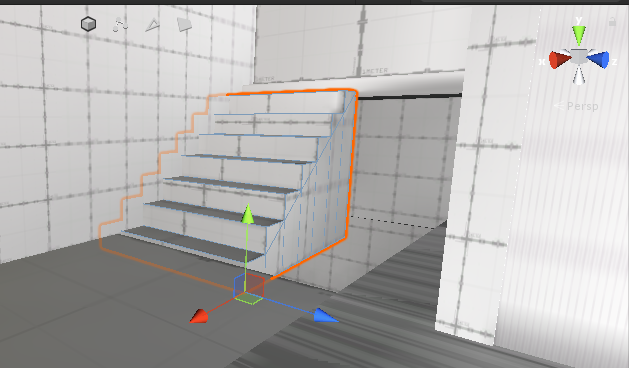
**Collider**: hỗ trợ lập trình va chạm giữa các vật thể.

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

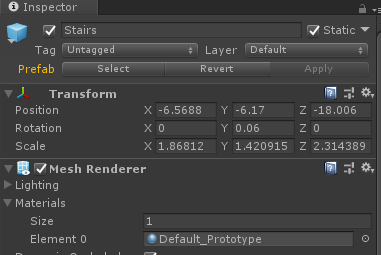
* + 1. **Không gian 3D và vector trong không gian**

**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.



*Một chiếc cầu thang trong không gian 3 chiều*



*Tương ứng với Rotation X, Y, Z*

* 1. **Cửa sổ làm việc**



**2.3.1.**     **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

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.

**2.3.2. 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 .

**2.3.3. 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.

**2.3.4. 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: dung cho view game, chỵ game ngay trong Editor để kiểm tra.

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

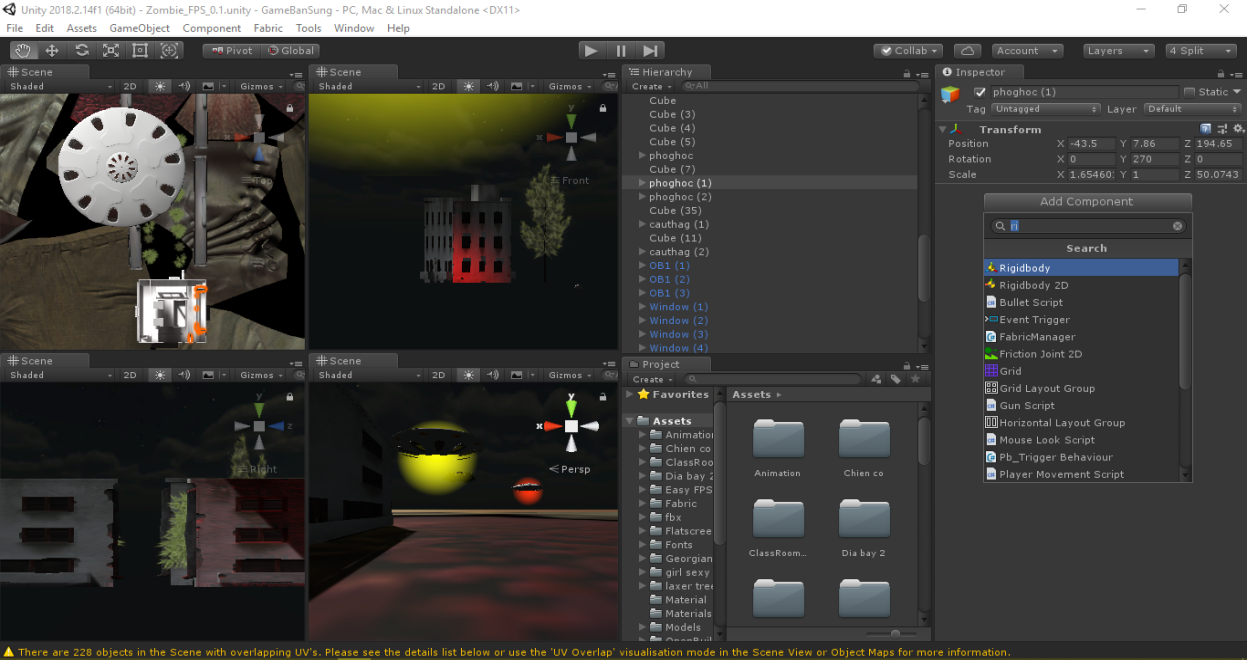
- Layout Drop-down kiểm soát sự sắp xếp cảu các Views.

**2.3.5. 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.



*Các góc nhìn từ các hướng khác nhau*

**2.3.6. 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.

**2.3.7. 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.

**2.3.8. 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.

        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.

* 1. **Các phần mềm hỗ trợ khác**

### **2.4.1. Photoshop**

Adobe Photoshop là phần mềm đồ họa có tác dụng chính là chỉnh sửa ảnh. Được phát triển và phát hành bởi Adobe năm 1988. Hiện nay Ps được xem là phần mềm đồ họa bitmap mạnh nhất trên thị trường, và được hầu hết những người làm thiết kế đồ họa cũng như các photography sử dụng. Photoshop cũng được đưa vào giáo trình giảng dạy chính thức của các trường đào tạo nghề cũng như các trường cao đẳng, đại học.

Photoshop giúp ta tạo ra các bản vẽ nhân vật, cắt ghép các chi tiết, tạo chữ, tạo nền trong suốt

**2.4.2**. **After Effects**

**Adobe After Effects** là phần mềm đồ họa tổng hợp được phát triển bởi hãng Adobe Systems. Phần mềm này được ứng dụng nhiều trong  đồ họa động và kỹ xảo chuyên nghiệp. Chức năng chính của After Effects là tạo ra được những chuyển động đồ họa ấn tượng, đẹp mắt. Đây là một công cụ tuyệt vời cho ngành thiết kế đồ họa, kỹ xảo trong quảng cáo hay game hoặc truyền hình. After Effects tương thích ngược với nhiều phần mềm khác của Adobe vì vậy ta có thể lấy sản phẩm tạo ra từ Photoshopđể xử lý tiếp, hoàn thiện hiệu ứng nhân vật trong game

### **2.4.3. Autodesk 3ds Max**

**3D Max** là một phần mềm diễn họa các vật thể 3 chiều.  Phần mềm tuyệt vời này còn bao gồm thiết kế mô hình 3D, hoạt hình và các trình diễn mô hình 3Ds. Các mô hình nhân vật, quái thú, đĩa bay UFOs trong game được hỗ trợ đắc lực bởi phần mềm này

**2.4.4. Visual Studio**

**Visual Studio** là (IDE**– *Integrated Development Environment***) một bộ công cụ phát triển phần mềm do Microsoft phát triển. Visual Studio cũng là một tool được sử dụng bởi các lập trình viên để xây dựng nên các sản phẩm phần mềm.



* 1. **Game và hướng phát triển**

**2.5.1. Các thể loại game nổi bật trong những năm gần đây:**

* **Game nhập vai**

MMORPG là chữ viết tắt của Massively Multiplayer Online Role - Playing Game (game nhập vai trực tuyến nhiều người chơi). Thực tế, MMORPG là một dạng trò chơi RPG (nhập vai) nhưng có nhiều người chơi giao tiếp với nhau (thông qua người điều khiển) thay vì chỉ giao tiếp với các nhân vật NPC của game - thông qua một máy chủ (server) của nhà cung cấp quản lý.

Đối với nhiều người, việc được hoà nhập vào cuộc sống trong game là một yếu tố quan trọng quyết định độ hay dở của trò chơi đó. Trong tất cả thể loại game thì MMORPG có thể nói là làm được điều này tốt nhất. Không chỉ xây dựng được đủ những yếu tố của một game nhập vai đơn thuần, những trò chơi như thế này còn tạo ra được một xã hội ảo thu nhỏ để người chơi giao lưu trong đó. Một bom tấn trên thị trường hiện nay có thể kể đến sản phẩm Blade & Soul. Không chỉ sở hữu đồ họa vượt trội, game còn khiến người chơi ngất ngây bởi tính chất thần thoại và kỳ ảo hiếm có trong cốt truyện hấp dẫn, từ đó xây dựng hệ thống gameplay có chiều sâu và luôn luôn ẩn chứa vô vàn bí mật thú vị.

* **FPS**

FPS ám chỉ thể loại game bắn súng góc nhìn thứ nhất: First-Person Shooter, cùng với góc nhìn thứ 3 Third-Person Shooter cũng là 1 trong những thể loại game có từ rất lâu trên thị trường. Người chơi sẽ hòa mình vào các trận chiến đầy cam go, một là chết hai là sống, chiến đấu như những chiến binh thực thụ, đánh thức bản năng sinh tồn mạnh mẽ trong mỗi người chơi. Ngoài ra, 1 số tựa game FPS còn cho phép người chơi trang bị các loại vũ khí, phương tiện chiến đấu hoành tráng như xe tăng, máy bay hay robot. Seri game Modern Combat là cái tên tiêu biểu của thể loại này, ngoài ra còn có Call Of Duty hay Dead Trigger.

* **Strategy**

Game chiến thuật là một loại hình khá phổ biến, dựa vào việc sắp xếp các chiến thuật để thực hiện những mục tiêu theo thứ tự. Game chiến thuật chú trọng cách quản lý nguồn tài nguyên và quân đội trước những đối thủ do máy hoặc người chơi khác điều khiển và hầu hết đều “dính dáng” đến chiến tranh. Nhìn chung, có 2 dạng phân biệt được là real-time strategy (chiến thuật thời gian thực, RTS) và turn-based strategy (chiến thuật theo lượt, TBS).

Có thể thấy, game chiến thuật được đánh giá là kén người chơi nhất trong tất cả các thể loại game đã được phát hành ở Việt Nam. So với những thể loại khác, người chơi dễ dàng nắm bắt được cách chơi và trở nên giỏi hơn nhờ tập luyện hay đầu tư tiền của, nhưng với game chiến thuật thì tập luyện, đầu tư tiền bạc thôi là chưa đủ, mà nó đòi hỏi người chơi phải có những tố chất nhất định.

* **MOBA**

MOBA là cụm từ viết tắt của Multiplayer Online Battle Arena (tạm dịch là Đấu trường người chơi trực tuyến). Các trò chơi thuộc thể loại game MOBA còn được gọi là game hành động chiến lược thời gian thực (Action Real Time Strategy - ARTS) vì nó có nguồn gốc từ thể loại game chiến lược thời gian thực (Real Time Strategy RTS), trong đó thì người chơi điều khiển một nhân vật duy nhất (còn được gọi là Hero hay Anh Hùng). Game MOBA sở hữu lối chơi trên 2 phe đối lập, giao tranh ở bất cứ đâu trên bản đồ rộng lớn và nhiệm vụ chung của 2 bên là phải phá hủy căn cứ đối phương, farm, giết quái, đối thủ. LMHT, DOTA2 là ví dụ điển hình của thể loại game này.

**2.5.2. Hướng phát triển:**

Mỗi thể loại game đều có những điểm hấp dẫn riêng và phù hợp với sở thích của từng nhóm đối tượng. Để mở rộng phạm vi người chơi chúng em phát triển game bắn súng 3D nhập vai FPS góc nhình thứ nhất, có yếu tố kinh dị kết hợp chiến thuật, sử dụng MAP (bản đồ) thế giới mở. Hiện tại chúng em xây dựng phiên bản game cho người dùng chạy hệ điệu hành Windows dưới dạng file thực thi (.exe). Nhớ tính đa nền tảng của Unity, trong thời gian sắp tới game sẽ được xây dựng bản (.apk) hoặc phát hành trên CH Play cho người dùng android, có mặt trên Appstore cho người dùng IOS

# CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME

- Phần mềm: Unity

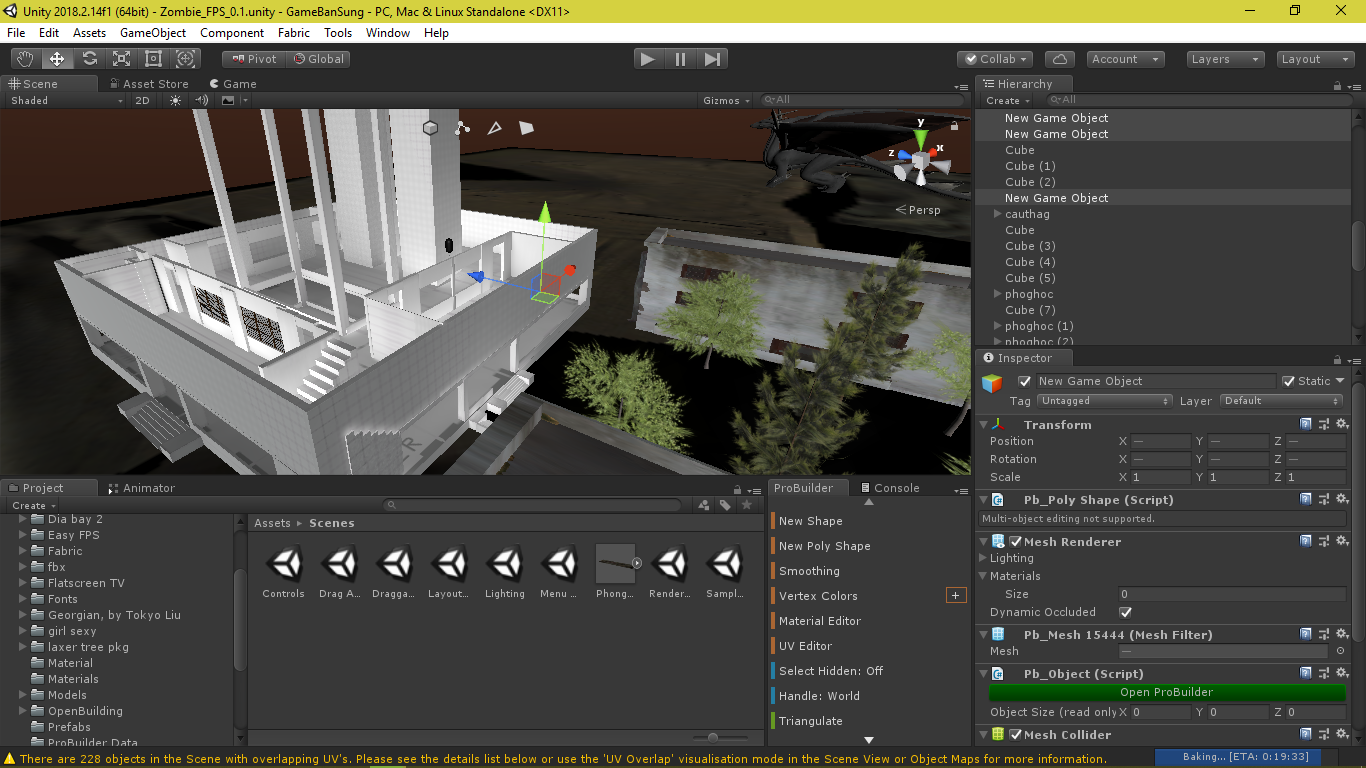
- Nền tảng: Windows

* Ngôn ngữ lập trình : C#
* Công cụ hỗ trợ Visual Studio 2013

## 3.1 Ý tưởng

Vào ngày Halloween các thế lực huyền bí kéo đến, chúng tạo ra vô số thây ma đi khắp nơi trên thành phố, tìm và hút máu những sinh vật chúng nhìn thấy. Người chơi sẽ nhập vai vào một dũng sĩ tiêu diệt những thây ma đáng sợ và phá bỏ UFOs bay lơ lửng phía trên với các máy tạo zombie để chúng không tạo ra thêm những thây ma mới nữa.

Dưới góc nhìn thứ nhất chân thực, người chơi được hòa mình vào không khí đậm chất ma quái, kích tích và thú vị, rèn luện kỹ năng phản xạ. Thiết kế game bắn súng góc nhìn thứ nhất



*Tạo dãy nhà C bằng công cụ probuilder \**

## 3.2. Tạo bản đồ và nhân vật

**3.2.1. Bản đồ (Map):**

Tạo dựng bằng cách kết hợp nhiều môi nhà cây cỏ và những thứ giống như một khu phố ngoài đời thực.



*Xây dựng lớp học, cửa sổ bục giảng, thang máy từ những khối cube*

**3.2.2. Nhân vật**

* **Player**



* Di chuyển và chạy nhảy (chạy nhanh) để tránh chướng ngại vật và thoát khỏi kẻ địch
* Sử dụng vũ khí để tiêu diệt kẻ địch, ghi điểm tìm cách sống sót tìm “key” phá bỏ UFOs để chúng không thể sản sinh thêm zombie nữa

* Enemy (kẻ địch):



Bao gồm zombie (thây ma), quái vật khổng lồ, và vật thể bay UFOs



### zombie





### Vật thể bay

Thây ma có khả năng tìm và tiêu diêt bất kể sinh vật nào nó nhìn thấy, chúng sẽ lại gần cào và hút máu người chơi nếu như khoảng cách giữa người chơi và chúng gần, tuy nhiên tốc độ chạy của người chơi cao hơn tốc độ của “thây ma” nên khi bị tấn công người chơi có thể trốn chạy. Tầm nhìn của zombie hạn chế, nên khi chạy xa zombie, người chơi “tạm” an toàn

Áp dụng AI để “zombie” có thể di chuyển một cách tự nhiên, tránh chướng ngại vật, giới hạn tầm nhìn của zombie chỉ 20m để giảm bớt độ khó cho người chơi, ở những level sau độ khó sẽ tăng dần

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

- Di chuyển: Người chơi sử dụng bàn phím để di chuyển với tổ hợp phím WASD (tương ứng với tiến lên, sáng trái, lùi xuống và sang phải), ngoài ra giữ nút Shift để chạy. Thanh Space để giúp người chơi nhảy , leo trèo trên các chướng ngại vật. - Tấn công : Mouse 1 (chuột trái) để bắn súng đang cầm trên tay, Mouse 2 (chuột phải) để ngắm bắn (aim down sight) – tăng cường độ chính xác khi bắn những kẻ đích ở xa. Mouse Wheel (con lăn) để chuyển vũ khí (khi đã unlock). Người ra người chơi có nút bắn phụ là Left Ctrl.

- Zombie được gắn code AIEnemy để chúng tự động đi săn lùng mục tiêu, khi thấy người chơi nó sẽ tiến lại gần và tấn công, khi gặp các chướng ngạy vậy chúng sẽ tìm hướng đi khác. Zombie có 100 đơn vị máu, với mỗi phát bắn từ súng của người chơi chúng giảm 10% máu. Zombie chết khi máu <=0 và gọi tới animator nằm xuống, sau một khoảng thời gian xác zombie tự biến mất

-Thanh máu: Máu là một giá trị quan trọng với người chơi, luôn luôn phải giữ cho máu >=0 nếu không thì trò chơi sẽ kết thúc (Game Over). Nó sẽ được hiển thị ở góc trái phía bên dưới của màn hình.

- Nhu yếu phẩm (ammunition) hiển thị ở góc phải dưới của màn hình, là nơi người nhận biết số lượng đạn của mình hiện đang có (đạn hiện có trong 1 băng đạn / tổng số đạn còn lại)

- Điểm hiển thị ở góc trên của màn hình, nơi người chơi nhận biết được mình đạt được bao nhiêu điểm từ tiêu diệt kẻ địch

* Khi người chơi hết máu thì trò chơi sẽ kết thúc (game Over).

## 3.4. Tiến trình của Game

* Người chơi sẽ phải khéo léo tiêu diệt những thây ma di chuyển từ UFOs xuống măt đất và những máy tạo ra thây ma
* Trên tay khẩu súng được trang bị số lượng đạn khổng lồ, người chơi phải “tiêu diệt” những “thây ma” khát máu đang truy tìm và tiêu diệt người chơi
* Càng ngày số lượng thây ma càng nhiều, người chơi phải cố gắng sống sót, tìm được “Key” là chiều khóa ở một vị trí ngẫu nhiên nào đó để kích hoạt tàu chiến khổng lồ. Khi tàu chiến được kích hoạt thì tàu chiến khởi hành từ một vị trí xa xôi ngoài trái đất tiến đến một căn cứ nào đó tiêu hủy nó thì các UFOs sẽ biến mất.
* Trò chơi kết thúc khi người chơi bị “thây ma” đánh bại hoặc bị Boss chùm tiêu diệt
* Từ giao diện chính người chơi có những lựa chọn sau:
  + - New Game – Bắt đầu trò chơi
    - Options – Điều chỉnh thông số của các chức năng
    - Quit – Thoát game.

## 3.5. Hoạt họa và cảm nhận

* UFOs kỳ lạ và quái vật xuất hiện trên nền trời rơi xuống đất đi lung tung trên mặt đất tạo cảm giác ghê gơn.
* Màn chơi là khung cảnh thân thuộc của sinh viên: Khu phố bị zombie tấn công và đã di chuyển nơi ở khác tạo nên một thành phố hoang
* Hiệu ứng vật lý, hiệu ứng ánh sáng kết hợp nhạc nền dồn dập tạo cảm giác thôi thúc, kịch tính, thú vị

# CHƯƠNG 4 CHƯƠNG TRÌNH DEMO

# *(Kèm file chương trình .exe)*

## 4.1. Gameplay

- Bắt đầu người chơi được hồi sinh tại cổng hồi sinh.

- Mục tiêu chính: Sống sót và đạt điểm cao nhất có thể. "Phá đảo" khi tìm được "key" và kích hoạt phi cơ tiêu diệt tận gốc UFOs

- Người chơi dùng chuột để định hướng camera. Dùng phím W,A,S,D để đi chuyển, phím Shift để chạy nhanh, phím cách để nhảy, lăn bánh xe chuột để đổi vũ khí

- Khi ngắm đúng mũi súng vào cơ thể zombie và nổ súng zombie sẽ bị mất 10% máu. Khi bắn trúng đầu, zombie sẽ giảm 20% máu và rơi đầu ra, tuy nhiên chúng vẫn tấn công người chơi được

- Càng ngày số lượng zombie từ vật thể bay UFOs xuống càng nhiều, nếu người chơi không nhanh chóng, sẽ bị chúng bao vây là tấn công

- Với mỗi lần zombie cào, người chơi sẽ mất 10% máu, zombie xung quanh càng đông thì cơ hội sống của người chơi càng thấp

- Khi người chơi tìm được “key” , phi cơ riêng sẽ được kích hoạt và phá bỏ UFOs, zombie sẽ không sản sinh ra thêm được nữa, người tìm chiến đấu BOSS là một rồng ngoài vũ trụ thật to lớn. Tiêu diệt nó người chơi sẽ chiến thắng. Nếu bị tiêu diệt người chơi sẽ thua và quay trở lại lúc hòi sinh ban đấu.

## 4.2. Các chức năng cơ bản

* Các tính năng chính được đưa ra và test trong demo bao gồm:
  + Chức năng cơ bản của player (bắn, ngắm bắn, thay đạn, di chuyển …)
  + Chức năng kẻ địch (di chuyển, tìm đường đến player, tấn công player khi vào tầm với, bị hạ gục …)
  + Chức năng hiển thị của map (ánh sáng, vật dụng và quang cảnh)
  + Chức năng nhạc nền, tính điểm và GameOver.
  + Một số các cơ chế phụ trợ khác: tần suất xuất hiện kẻ địch, vị trí xuất hiện kẻ địch

* Từ góc nhìn thứ nhất (First Person Shooter) của người chơi thấy tay và vũ khí hiện tại mình đang sử dụng và tận dụng tối đa không gian màn hình để quan sát kẻ dịch, tìm đường đi, vượt chướng ngại vật, tìm key. Ngoài ra trên screen của người chơi còn hiện số đạn dược hiện tại, số máu hiện tại và số điểm đạt được.

* Hiện tại trong trương trình demo, vũ khí đã được trang bị sẵn và không cần phải mua
* Mỗi khẩu súng có những setting riêng biệt, âm thanh, hình ảnh, hoạt họa khác nhau.

# CHƯƠNG 5

# KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## 5.1 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.

## 5.2. Định hướng 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).

# CHƯƠNG 6 TÀI LIỆU THAM KHẢO – PHỤ LỤC

## 6.1. Tài liệu tham khảo

* Lập trình game 2D - Thầy Nguyễn Minh Tuấn
* Lập trình Unity căn bản – HowKteam
* Lập trình Unity nâng cao – Kim Long
* After effect cơ bản – Nhiều nguồn
* Learning C# Programming with Unity 3D - Alex Okita
* Tài liệu lập trình C# - Đại học KHTN TP HCM , J2team
* Lập trình hướng đối tượng – Cô Phan Xuân Hồng
* Unity Learn Tutorials
* Lịch sử ra đời Unity – Nhiều nguồn
* Thống kê sử dụng unity – Nhiều nguồn
* Khóa học các công cụ của Adobe – DayNhauHoc, HowKteam,KhoaPham,CodeDao,…

## 6.2 Phụ lục

Probuilder:  là 1 plugin được viết cho Unity - bộ công cụ giúp các ta tao ra khối/đối tượng/môi trường 3D từ những hình khối cơ bản nhất

Zombie: thây ma, xác sống, những nhân vật tưởng tượng mang tính chất kinh dị, chúng đi cào cấu, hút máu

Phá đảo: đạt được thử thách khó nhất của trò chơi, vượt qua các level cao nhất