**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**Khoa Công Nghệ Phần Mềm**

--🙢🕮🙠---



**BÁO CÁO**

**KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI:**

**XÂY DỰNG GAME VR**

**SURVIVAL SHOOTER BẰNG UNITY TRÊN GOOGLE CARDBOARD**

Giảng viên hướng dẫn:

**ThS.** **NGUYỄN VĨNH KHA**

Sinh viên thực hiện:

**BÙI QUANG HÀ – 13520222**

**NGUYỄN ĐÌNH CHƯƠNG – 13520086**

***TP. Hồ Chí Minh, ngày 25 tháng 11 năm 2017***

# LỜI CẢM ƠN

Đầu tiên, nhóm tác giả xin gởi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Công Nghệ Phần Mềm đã giúp cho nhóm tác giả có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, nhóm tác giả xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy Nguyễn Vĩnh Kha, đã hướng dẫn tận tình, truyền đạt nhiều kiến thức bổ ích để giúp chúng em hoàn thành được sản phẩm.

Trong thời gian thực hiện đề tài, nhóm tác giả đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm tác giả vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm tác giả không tránh khỏi những thiếu sót. Chính vì vậy, nhóm tác giả rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm tác giả đã học tập và là hành trang để nhóm tác giả thực hiện tiếp các đề tài khác trong tương lai. Xin chân thành cảm ơn các quý Thầy Cô !

Nhóm sinh viên thực hiện.

# NHẬN XÉT CỦA GIẢNG VIÊN

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

[LỜI CẢM ƠN 2](#_Toc16804)

[NHẬN XÉT CỦA GIẢNG VIÊN 3](#_Toc26993)

[CHƯƠNG 1. TỔNG QUAN VỀ THỰC TẾ ẢO 6](#_Toc23613)

[1.1 Thực tế ảo là gì ? 6](#_Toc7410)

[1.2 Lịch sử phát triển của công nghệ thực tế ảo 7](#_Toc18550)

[1.3 Các đặc tính chính của VR 8](#_Toc4655)

[1.4 Các thành phần một hệ thống VR 9](#_Toc20645)

[1.4.1 Phần cứng (Hardware) 9](#_Toc275)

[1.4.2 Phần mềm (Software) 10](#_Toc12359)

[1.5 Các thiết bị cơ bản 10](#_Toc5471)

[**1.5.1 Thiết bị định hướng và chuyển động 10**](#_Toc12657)

[**1.5.2 Thiết bị tương tác và phản hồi 14**](#_Toc20701)

[1.6 Một số ứng dụng chính của VR 15](#_Toc22175)

[**1.6.1 Quân sự 16**](#_Toc7284)

[**1.6.2 Giáo dục 17**](#_Toc31765)

[**1.6.3 Xây dựng 18**](#_Toc26245)

[**1.6.4 Thể thao 18**](#_Toc29998)

[**1.6.5 Du hành không gian 19**](#_Toc24948)

[**1.6.6 Y học 19**](#_Toc3410)

[CHƯƠNG 2. GOOGLE CARDBOARD VÀ CÁC THƯ VIỆN HỖ TRỢ VIRTUAL REALITY 20](#_Toc19284)

[2.1 Giới thiệu Google Cardboard 20](#_Toc20842)

[**2.1.1 Tìm hiểu chung về Google Cardboard 20**](#_Toc12295)

[**2.1.2 Ứng dụng 22**](#_Toc28160)

[**2.1.3 Phần mềm phát triển Google Cardboard.................................... . 22**](#_Toc24919)

[2. 2 Các thư viện hỗ trợ virtual reality 23](#_Toc3962)

[**2.2.1 Google VR SDK 23**](#_Toc31950)

[**2.2.1.1 Tổng quan 23**](#_Toc23820)

[**2.2.1.2 Tích hợp Google VR SDK vào project. 25**](#_Toc21847)

[CHƯƠNG 3. TỔNG QUAN VỀ UNITY 3D 29](#_Toc11347)

[3.1 Giới thiệu Unity 29](#_Toc1781)

[**3.1.1 Tổng quan 29**](#_Toc15037)

[**3.1.2 Giới thiệu về giao diện 31**](#_Toc27915)

[3.2 Một số khái niệm cơ bản trong Unity 46](#_Toc25377)

[**3.2.1 Scene. 46**](#_Toc14348)

[**3.2.2 GameObject 47**](#_Toc14651)

[**3.2.3 Components. 48**](#_Toc1421)

[**3.2.4 Scripts 49**](#_Toc488)

[**3.2.5 Prefabs 49**](#_Toc13842)

[CHƯƠNG 4. THIẾT KẾ GAME 51](#_Toc30289)

[4.1. Thiết kế game 51](#_Toc22027)

[**4.1.1. Tổng quan 51**](#_Toc29035)

[**4.1.2. Mục tiêu 51**](#_Toc23507)

[**4.1.3. Tính năng 52**](#_Toc25202)

[**4.1.4. Cách chơi 52**](#_Toc19135)

[**4.1.5. Mindset 52**](#_Toc29773)

[**4.1.6. Đối tượng người chơi 52**](#_Toc2933)

[4.2. Công nghệ 53](#_Toc6394)

[**4.2.1. Luồng màn hình 53**](#_Toc3105)

[**4.2.2 Điều khiển 54**](#_Toc24950)

[**4.2.3 Cơ học (game mechanics) 55**](#_Toc3917)

[4.3 Các yếu tố Game Play 55](#_Toc3511)

[**4.3.1 Game Physics and Statistics 55**](#_Toc1627)

[**4.3.2 Trí tuệ nhân tạo 55**](#_Toc23992)

[4.4 Thiết kế màn chơi 56](#_Toc31991)

[**4.4.1 Các đối tượng 56**](#_Toc13224)

[**4.4.2 Chủ đề màn chơi 56**](#_Toc1277)

[**4.4.3 Luồng chạy game 56**](#_Toc22706)

[4.5 Development 56](#_Toc20133)

[**4.5.1 Thiết kế engine 56**](#_Toc260)

[**4.5.2 Các lớp / đối tượng 57**](#_Toc14066)

[4.6 Development schedule 58](#_Toc20725)

[4.7 Visual Design 58](#_Toc9911)

[4.8 User Interface 59](#_Toc29822)

[**4.8.1 Flow Chart 59**](#_Toc11955)

[4.9 Đồ họa 63](#_Toc9062)

[**4.9.1 Mục tiêu tổng quát 63**](#_Toc29068)

[**4.9.2 3D Art & Animation 63**](#_Toc32712)

[**4.9.3 GUI 64**](#_Toc12986)

[**4.9.4 Địa hình 64**](#_Toc27108)

[**4.9.5 Hiệu ứng 64**](#_Toc22936)

[4.10 Thiết kế nhạc nền / âm thanh 64](#_Toc31574)

[**4.10.1 Phong cách music / elements 64**](#_Toc29242)

[**4.10.2 Phong cách sound / elements 64**](#_Toc11562)

[**4.10.3 Game music 64**](#_Toc16823)

[**4.10.4 Game sound 64**](#_Toc1892)

# CHƯƠNG 1. TỔNG QUAN VỀ THỰC TẾ ẢO

## Thực tế ảo là gì ?

Thực tế ảo-Virtual Reality( VR ) là một hệ thống mô phỏng trong đó đồ họa máy tính được sử dụng để tạo ra một thế giới "như thật". Hơn nữa, thế giới "nhân tạo" này không tĩnh tại, mà lại phản ứng, thay đổi theo ý muốn (tín hiệu vào) của người sử dụng (nhờ hành động, lời nói,..). Điều này xác định một đặc tính chính của VR, đó là tương tác thời gian thực (real-time interactivity). Thời gian thực ở đây có nghĩa là máy tính có khả năng nhận biết được tín hiệu vào của người sử dụng và thay đổi ngay lập tức thế giới ảo. Người sử dụng nhìn thấy sự vật thay đổi trên màn hình ngay theo ý muốn của họ và bị thu hút bởi sự mô phỏng này.

Điều này chúng ta có thể nhận thấy ngay khi quan sát trẻ nhỏ chơi video game. Theo báo Bild (Đức), có hai trẻ nhỏ ở Anh bị thu hút và mải mê chơi Nintendo đến nỗi ngay cả khi nhà chúng đang bị cháy cũng không hề hay biết! Tương tác và khả năng thu hút của VR góp phần lớn vào cảm giác đắm chìm (immersion), cảm giác trở thành một phần của hành động trên màn hình mà người sử dụng đang trải nghiệm. Nhưng VR còn đẩy cảm giác này "thật" hơn nữa nhờ tác động lên tất cả các kênh cảm giác của con người. Trong thực tế, người dùng không những nhìn thấy đối tượng đồ họa 3D nổi (như hình nổi ở trang cuối báo Hoa học trò đã đăng trước kia), điều khiển (xoay, di chuyển,..) được đối tượng trên màn hình (như trong game), mà còn sờ và cảm thấy chúng như có thật. Ngoài khả năng nhìn (thị giác), nghe (thính giác), sờ (xúc giác), các nhà nghiên cứu cũng đã nghiên cứu để tạo các cảm giác khác như ngửi (khứu giác), nếm (vị giác). Tuy nhiên hiện nay trong VR các cảm giác này cũng ít được sử dụng đến.

Từ các phân tích trên, chúng ta có thể thấy định nghĩa sau đây của C. Burdea và P. Coiffet về VR là tương đối chính xác: VR- Thực Tế Ảo là một hệ thống giao diện cấp cao giữa Người sử dụng và Máy tính. Hệ thống này mô phỏng các sự vật và hiện tượng theo thời gian thực và tương tác với người sử dụng qua tổng hợp các kênh cảm giác. Đó là ngũ giác gồm: thị giác, thính giác, xúc giác, khứu giác, vị giác.

## Lịch sử phát triển của công nghệ thực tế ảo

Khái niệm thực tế ảo đã có trong nhiều thập niên nhưng nó chỉ thực sự được nhận thức vào đầu những năm 90. Vào giữa những năm 50 Morton Heilig (Mỹ) đã phát minh ra thiết bị mô phỏng SENSORAMA. Đó là 1 thiết bị điều khiển 1 người sử dụng gồm có : một màn hình thực thể kính, quạt, máy tạo mùi, loa âm thanh và 1 chiếc ghế có thể di chuyển được. Ông cũng phát minh ra màn hình truyền hình được gắn vào đầu để có thể xem phim 3D. Tuy là những sản phẩm phục vụ cho điện ảnh nhưng những khái niệm của Heilig đã trở thành tiền đề cho VR sau này.

Những kỹ sư của Công ty Philco là những người đầu tiên phát triển HMD vào 1961, gọi là Headsight. Cái mũ sắt bao gồm một màn ảnh và hệ thống theo dõi video đã những kỹ sư liên kết tới một hệ thống camêra mạch đóng. Họ dự định sử dụng HMD trong các tình huống nguy hiểm - một người có thể quan sát một môi trường thực sự từ xa, điều chỉnh góc quay camera bằng cách quay đầu. Bell Laboratories đã sử dụng HMD cho những phi công lai máy bay trực thăng. Họ liên kết HMD với những camera hồng ngoại gắn bên ngoài máy bay giúp phi công có thể nhìn rõ ngay cả trong môi trường thiếu ánh sáng.

Vào 1965, một nhà khoa học máy tính có tên Ivan Sutherland hình dung điều mà ông ta gọi là "Ultimate Display". Sử dụng hiển thị này, một người có thể thấy một thế giới ảo hiện ra như thế giới vật lý thật. Điều này đã định hướng toàn bộ tầm nhìn về VR. Khái niệm của Suntherland bao gồm :

* + - Một thế giới ảo mà ta có thể quan sát thông qua một HMD
    - Một máy tính để duy trì các mô hình trong thời gian thực
    - Các khả năng cho người sử dụng để thao tác những đối tượng thực tế một cách trực quan nhất.

## Các đặc tính chính của VR

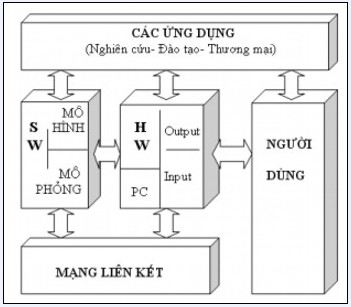
Như trên đã trình bày, 2 đặc tính chính của VR là Tương tác và Đắm chìm, đây là hai "I" (Interactive, Immersion) mà nhiều người đã biết. Tuy nhiên VR cần có 1 đặc tính thứ 3 mà ít người để ý tới.

VR không chỉ là một hệ thống tương tác Người- Máy tính, mà các ứng dụng của nó còn liên quan tới việc giải quyết các vấn đề thật trong kỹ thuật, y học, quân sự,…

Các ứng dụng này do các nhà phát triển VR thiết kế, điều này phụ thuộc rất nhiều vào khả năng Tưởng tượng của con người, đó chính là đặc tính "I" (Imagination) thứ 3 của VR. Do đó có thể coi VR là tổng hợp của 3 yếu tố: Tương tác- Đắm chìm- Tưởng tượng, (3 I trong tiếng Anh: Interactive- Immersion- Imagination)

## Các thành phần một hệ thống VR

Tổng quát một VR bao gồm những thành phần sau



Hình 1.1 Các thành phần của một hệ thông VR

## Phần cứng (Hardware)

Phần cứng của một VR bao gồm:

* + - Máy tính (PC hay Workstation với cấu hình đồ họa mạnh).
    - Các thiết bị đầu vào (Input devices): Bộ dò vị trí (position tracking) để xác định vị trí quan sát. Bộ giao diện định vị (Navigation interfaces) để di chuyển vị trí người sử dụng. Bộ giao diện cử chỉ (Gesture interfaces) như găng tay dữ liệu (data glove) để người sử dụng có thể điều khiển đối tượng.
    - Các thiết bị đầu ra (Output devices): gồm hiển thị đồ họa (như màn hình, HDM,..) để nhìn được đối tượng 3D nổi. Thiết bị âm thanh (loa) để nghe được âm thanh vòm (như Hi-Fi, Surround,..). Bộ phản hồi cảm giác (Haptic feedback như găng tay,..) để tạo xúc giác khi sờ, nắm đối tượng. Bộ phản hồi xung lực (Force Feedback) để tạo lực tác động như khi đạp xe, đi đường xóc,…

## Phần mềm (Software)

Phần mềm luôn là linh hồn của VR cũng như đối với bất cứ một hệ thống máy tính hiện đại nào. Về mặt nguyên tắc có thể dùng bất cứ ngôn ngữ lập trình hay phần mềm đồ họa nào để mô hình hóa (modelling) và mô phỏng (simulation) các đối tượng của VR.

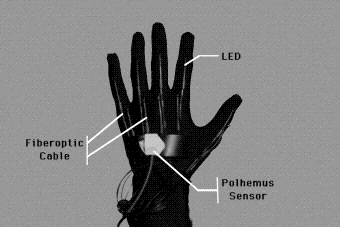
Ví dụ như các ngôn ngữ (có thể tìm miễn phí) OpenGL, C++, Java3D, VRML, X3D,. ..hay các phần mềm thương mại như WorldToolKit, PeopleShop,... Phần mềm của bất kỳ VR nào cũng phải bảo đảm 2 công dụng chính: Tạo hình vào Mô phỏng. Các đối tượng của VR được mô hình hóa nhờ chính phần mềm này hay chuyển sang từ các mô hình 3D (thiết kế nhờ các phần mềm CAD khác như AutoCAD, 3D Studio,..). Sau đó phần mềm VR phải có khả năng mô phỏng động học, động lực học, và mô phỏng ứng xử của đối tượng.

## Các thiết bị cơ bản

* + 1. **Thiết bị định hướng và chuyển động**

**DataGloves**

Thiết bị đo lường bàn tay phải cảm nhận được cả độ cong của các ngón tay và vị trí, sự định hướng của cổ tay trong thời gian thực. Thiết bị thương mại đầu tiên là DataGloves từ viện nghiên cứu VPL. DataGloves bao gồm 1 găng tay nylon nhẹ có các cảm biến quang học được gắn ở các ngón tay.



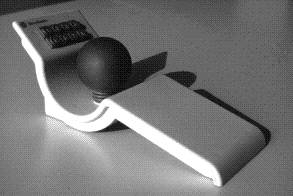
Hình 1.2 DataGloves

**3D Mouse and SpaceBall**



Hình 1.3 3D Mouse và SpaceBall

Chuột Logitech 3D dựa trên một mảng các vị trí siêu âm tham chiếu, đó là 1 cái kiềng gồm 3 loa siêu âm đặt ở 3 góc tam giác phát ra tín hiệu siêu thanh. Nó được sử dụng để theo dõi thiết bị thu, định hướng và chuyển động. Nó qui định thành phần của tỷ lệ gửi ra trong tất cả 6. mức tự do: X, Y, Z, Pitch, Yaw, và Roll.



Hình 1.4 Mouse

**Shutter glasses**

Hình 1.5 Shutter glasses

**Head-Mounted Displays**

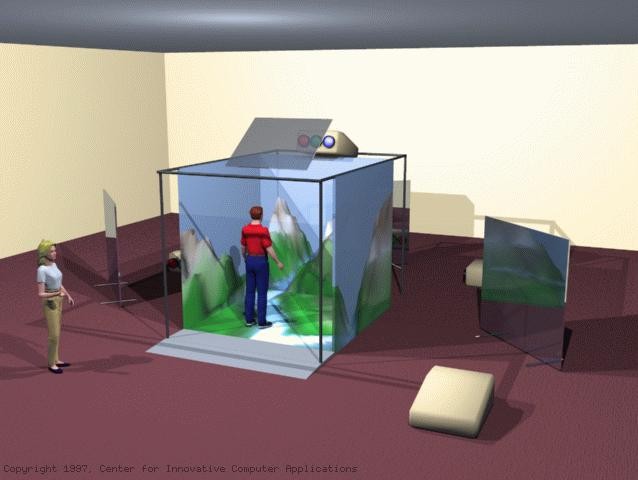


Hình 1.6 Head-Mounted Displays

**Cave**

CAVE là 1 nhà hát có kích thước 10 X 10 X 9 được đặt bên trong 1 phòng lớn hơn có kích thước 35 X 25 X 13. Phòng bên ngoài phải được chiếu sáng trong suốt quá trình sử dụng CAVE. Các bức tường của CAVE được tạo bởi các màn chiếu và sàn nhà cũng là một màn chiếu thẳng đứng. Máy chiếu độ phân giải cao hiển thị hình ảnh lên toàn bộ nhứng màn ảnh khác bằng các tấm gương phản chiếu. Nguwoif dùng sẽ đi vào bên trong CAVE và đeo 1 chiêc kinh đặc biệt để có thể nhìn thấy những hình ảnh 3 chiều mà CAVE hiển thị.

Với những chiếc kính này người dùng có thể thấy các đối tượng thực sự nổi trong không khí và có thể đi lại xung quanh chúng. Điều này la hoàn toàn khả dĩ với các cảm biến điện tử. Khung của CAVE được làm từ i-nox không từ tính để có thể can thiệp một cách tốt nhất vào các cảm biến điện từ. Khi một người đi lại trong CAVE, chuyển động của họ được theo dõi bởi các cảm biến này và video sẽ điều chỉnh cho phù hợp. Máy tính sẽ kiểm soát việc này của CAVE cũng như cả khía cạnh âm thanh nữa. Có rất nhiều loa được đặt trong CAVE dưới nhiều góc độ giúp cho không chỉ có hình ảnh 3 chiều mà có cả âm thanh 3 chiều nữa.



Hình 1.7 Cave

### **Thiết bị tương tác và phản hồi**

Các thiết bị này cảm nhận một số nhân tố sau của thiết bị khác gây ra: nhiệt độ, vận tốc di chuyển, sự chuyển động, áp lực và các ngoại lực khác.

**CyberTouch**

Hình 1.8 CyberTouch

**CyberGrasp**



Hình 1.9 CyberGrasp

## Một số ứng dụng chính của VR

Tại các nước phát triển, chúng ta có thể nhận thấy VR được ứng dụng trong mọi lĩnh vực: Khoa học kỹ thuật, kiến trúc, quân sự, giải trí,... và đáp ứng mọi nhu cầu: Nghiên cứu- Giáo dục- Thương mại. Y học là lĩnh vực ứng dụng truyền thống của VR.

Bên cạnh đó VR cũng được ứng dụng trong giáo dục, nghệ thuật, giải trí. Trong lĩnh vực quân sự, VR cũng được ứng dụng rất nhiều ở các nước phát triển. Bên cạnh các ứng dụng truyền thống ở trên, cũng có một số ứng dụng mới nổi lên trong thời gian gần đây của VR như: VR ứng dụng trong sản xuất, VR ứng dụng trong ngành rôbốt, VR ứng dụng trong hiển thị thông tin (thăm dò dầu mỏ, hiển thị thông tin khối,....) VR có tiềm năng ứng dụng vô cùng lớn. Có thể nói tóm lại một điều: Mọi lĩnh vực "có thật " trong cuộc sống đều có thể ứng dụng "thực tế ảo" để nghiên cứu và phát triển hoàn thiện hơn.

### **Quân sự**

Với việc phát triển của VR, các binh sĩ sẽ được huấn luyện 1 cách trực quan nhất các kĩ năng cân thiết như : lái máy bay, lai xe tăng, . .. trước khi tham gia công việc thực tế. Điều này vừa bảo đảm an toàn cho binh sĩ, vừa tiết kiệm được chi phí cho các khoa huấn luyện thực tế.

Lầu Năm Góc vừa đưa ra quyết định sẽ đầu tư 36 triệu USD cho quân đội Mỹ để phát triển một game đặc biệt nhằm huấn luyện binh sĩ chống lại khủng bố dưới dạng chiến thuật thực tế ảo**.**Với hệ thống trò chơi đặc biệt này, những binh sĩ có thể tập luyện những bài tập của mình ngay tại nhà nhằm chống lại những tình huống có thể phát sinh ra trong thực tế. Đây sẽ là một game rất sống động, có tình hành động cao với môi trường và bối cảnh bám sát với thực tế. Những người lính sẽ phải vận dụng tất cả những kỹ năng đã được rèn giũa trong quân đội.



Hình 1.10 Binh lính học nhảy dù bằng thực tế ảo

### **Giáo dục**



Hình 1.11 Cảnh trong một lớp học ảo

Ở các nước phương Tây việc ở nhà học qua Internet không còn là điều mới mẻ nữa. Và công nghệ VR sẽ làm cho việc này trở nên thú vị hơn rất nhiều. Giống như một game MMORPG bạn điều khiển 1 nhân vật đại diện cho bạn đi lại trong 1 trường học ảo được xây dựng trên máy tính. Bạn có thể tham gia vào bất cứ lớp học ảo nào mà bạn thích, nói chuyện với nhưng thành viên khác trong lớp.

### **Xây dựng**

Bạn muốn xây nhà. Bạn thuê một kiến trúc sư thiết kế cho ngôi nhà tương lai của bạn. Anh ta hoàn thành nó trên bản vẽ và liệu bạn có thể tưởng tượng ra nó thế nào không ? Có thể nhưng chắc là không thể chính xác được. Và khi hoàn thành thì chưa chắc nó đã đúng ý của bạn. Giờ đây ngôi nhà đó được xây dựng trên máy tinh, bạn có thể đi lại khắp nơi trong nhà, xem xét tưng ngõ ngách nhỏ nhất.

### **Thể thao**

Cũng giống như trong quân sự, kính thực tế ảo chính là công cụ đắc lực giúp các vận động viên luyện tập phản ứng trước những tình huống thực với chi phí thấp và không ngại bị chấn thương.

Hơn thế nữa, sự phát triển của thực tế ảo cũng là một tin vui cho rất nhiều người hâm mộ thể thao. Bởi giờ đây họ có thể quan sát thần tượng của mình ở cự ly gần và ngay giữa sân thi đấu thay vì theo dõi trên ghế khán đài hoặc qua màn hình vô tuyến.

|  |
| --- |
| 5 ung dung huu ich cua cong nghe thuc te ao hinh anh 2 |
| Hình 1.11 VR giúp các vận động viên luyện tập phản ứng. |

Năm 2015, bóng rổ Mỹ đã trở thành môn thể thao đầu tiên cho phép khán giả dùng kính VR để xem trận đấu. Năm 2016 có thêm sự tham gia của Microsoft khi hãng này cung cấp thiết bị đeo Hololens, cho phép người xem thưởng thức trận đấu ngay trước mắt với các cầu thủ dường như xuất hiện trong chính căn phòng.

### **Du hành không gian**

Các nhà khoa học của NASA đã thực hiện nhiều cuộc thám hiểu sao hỏa bằng cách ứng dụng công nghệ thực tế ảo. Đặc biệt, tại triển lãm điện tử tiêu dùng CES 2016 vừa qua, NASA đã trình diễn công nghệ thực tế ảo, cho phép người xem có trải nghiệm sống trong các con tàu du hành không gian.

### **Y học**

Thực tại ảo giải quyết được rất nhiều vấn đề trong y học: cung cấp môi trường thực hành cho nghiên cứu và học tập, rất hữu ích trong việc mô phỏng các ca phẫu thuật tránh gây rủi ro trong thực tế

Như vậy thực tại ảo có ứng dụng trong hầu hết các lĩnh vực của cuộc sống. Qua đó cũng nhận thấy được ý nghĩa to lớn của việc ứng dụng thực tại ảo, bởi những vấn đề khó khăn mà nếu không có thực tại ảo thì rất khó giải quyết hoặc hiểu quả không cao mà chi phí tốn kém.

# CHƯƠNG 2. GOOGLE CARDBOARD VÀ CÁC THƯ VIỆN HỖ TRỢ VIRTUAL REALITY

## 2.1 Giới thiệu Google Cardboard

### **2.1.1 Tìm hiểu chung về Google Cardboard**

Google Cardboard là nền tảng thực tế ảo (VR) và tương tác thực tế (AR) được phát triển bởi Google sử dụng cho thiết bị đeo lên đầu dùng với điện thoại di động. Tên của nó là Cardboard viewer, nền tảng này như là một hệ thống giá rẻ để khuyến khích sự yêu thích và phát triển các ứng dụng thực tế ảo và tương tác thực tế (AR).

Người dùng có thể tự làm kính thực tế ảo vô cùng đơn giản, giá rẻ sử dụng bản mô tả cách làm của Google, hay là mua các kính thực tế ảo *Google Cardboard* của các nhà cung cấp thứ 3. Nền tảng được tạo bởi David Coz và Damien Henry, các kỹ sư của Viện nghiên cứu văn hoá ở Paris, trong thời gian sáng tạo theo cơ chế 20% ["Innovation Time Off"](http://lanhdaotoancau.blogspot.com/2016/02/sang-tao-moi-tu-google-innovation-time-off.html). Nó được giới thiệu tại sự kiện Google I/O 2014 dành cho các nhà phát triển Android, và phát hành phiên bản cho iOS vào năm vừa rồi 2015. Qua tháng 01, năm 2016, đã có 5 triệu GoogleCardboard được bán ra trên toàn thế giới và hơn 1000 ứng dụng tương thích đã công bố trên Google Play.

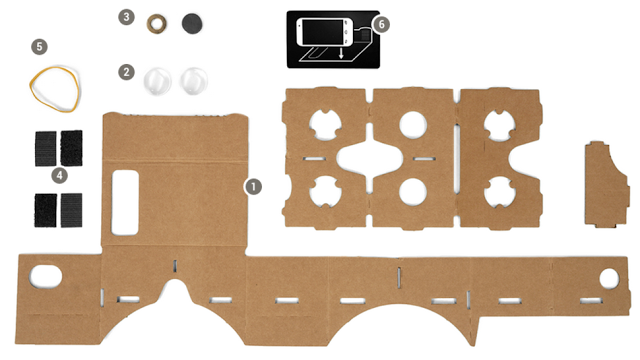
****

Hình 2.1 Hình ảnh 1 chiếc Google Cardboard sau khi đã lắp ráp

**Cách lắp rắp và hoạt động của Google Cardboard:**

Theo Google, Google Cardboard sẽ hoạt động dựa vào cảm biến gia tốc của smartphone để khi bạn xoay, di chuyển hoặc thực hiện chuyển động thì màn hình hiển thị sẽ thay đổi theo giúp bạn thưởng thức một không gian như ngoài đời thực.

Bên cạnh đó, camera của smartphone có một vài nhiệm vụ điều khiển các hoạt động trong ứng dụng, còn cặp nam châm trên Google Cardboard sẽ tác động đến từ kế để hỗ trợ chúng ta thực hiện thao tác khi sử dụng thay thế cho việc chạm vào màn hình cảm ứng. Tuy nhiên trên thực tế, có một số mẫu máy vẫn không tương thích lắm với chiếc Google Cardboard này.



Hình 2.2: Hình chưa lắp ráp và hình sau khi lắp ráp

Kính Google Cardboard được tạo ra rất đơn giản từ những phụ tùng giá rẻ. Bảng mô tả kính đeo được thiết kế bởi Google, nhưng không có bất kỳ nhà sản xuất hay nhà cung cấp chính quy nào. Thay vào đó Google làm danh sách các bộ phận, hình thiết kế, và hướng dẫn lắp ráp miễn phí ở trên website của họ, từ đó cho phép mọi người tự lắp ráp Cardboard từ các bộ phận đó.

Các bộ phận đó là cardboard được cắt theo các hình chính xác, thấu kính với focus 45mm, nam châm magnets hay điện dung, một móc, và dây deo vào đầu,  mấy miếng đệm cao su, và một tag giao tiếp NFC tuỳ chọn ko yêu cầu bắt buộc. Google cũng công bố các tài liệu hỗ trợ các nhà sản xuất số lượng lớn và các phụ kiện để lắp ráp từ các nhà cung cấp khác nhau, và có nhiều biến thể của Google Cardboard từ các hãng cung cấp thứ 3.

Khi bộ kit được lắp ráp xong, một điện thoại di động được gắn vào phía sau của thiết bị. Google Cardboard tương thích với các Apps chia đôi màn hình điện thoại trên di động tương ứng với mỗi mắt của chúng ta, có áp dụng chống méo hình từ các ống kính. Kết là là hình ảnh 3D với độ rộng của trường ảnh. .   
  
 Phiên bản đầu tiên của Cardboard chỉ dành cho các điện thoại màn hình tối đa là 5.7 inch và sử dụng nút nhấn nam châm, và yêu cầu điện thoại có cảm biến là bàn. Một nâng cấp thiết kế được công bố tại sự kiện Google I/O 2015 tương thích với màn hình điện thoại tờ 6 inch và thay thế nút nhấn nam châm bằng công tắc điện dung để tương thích với nhiều thiết bị điện thoại hơn. Và Google cũng đưa ra phiên bản Apps hỗ trợ iOS trong sự kiên đó.

### **2.1.2 Ứng dụng**

Ứng dụng Cardboard của Google đi kèm với những demo tương tự Google Earth, khiến người dùng có thể du lịch đến bất cứ nơi nào họ muốn. Ứng dụng còn có khả năng chuyển đổi video để tạo cảm giác 3D bằng cách chia đôi màn hình.

Với những người muốn trải nghiệm game trên thiết bị tuyệt vời này thì Lamper - một game được thiết kế để chạy trong môi trường thực tế ảo chắc chắn sẽ đem lại cho người chơi những trải nghiệm chưa từng có. Mặc dù không phải ứng dụng nào người dùng cũng có thể sử dụng được Google Cardboard nhưng chúng ta cũng đừng vội lo lắng. Google đã kịp thời cho ra mắt bộ SDK hỗ trợ cho việc lập trình cho thiết bị của mình. Điều này hứa hẹn một lượng lớn những ứng dụng sẽ được phát triển trong tương lai.

**2.1.3 Phần mềm phát triển Google Cardboard**  
 Google cung cấp hai kits phát triển để tạo các ứng dụng cho Cardboard, cả hai đều sử dụng OpenGL: một cho thiết bị Android với nền tảng Java, một là Game engine Unity sử dụng C#. Lúc đầu chỉ hỗ trợ Android, sau đó Google công bố hỗ trợ cả iOS với Unity plugin từ tháng 05-2015 tại Google I/O. Các Apps ứng dụng của nhà cung cấp cho Google Cardboard hiện diện trên Android Google Play và App Store cho iOS. Bên cạnh các ứng dụng native, Google Chrome cũng hỗ trợ trải nghiệm thực tế ảo sử dụng WebGL, điện thoại, hỗ trợ cả Apple. Tháng 01-2016, Google thông báo kit phát triển ứng dụng có hỗ trợ âm thanh đa chiều, như hiệu ứng thực tế ảo với Doppler.

## 2. 2 Các thư viện hỗ trợ virtual reality

### **2.2.1 Google VR SDK**

### **2.2.1.1 Tổng quan**

Tạo ra trải nghiệm thực tế ảo với SDK VR của Google sẽ mang lại tính giải trí và giáo dục cho người dùng, cho dù họ đang sử dụng điện thoại thông minh hỗ trợ Daydream hay Google Cardboard.

**Carboad SDK** là bộ công cụ dành cho lập trình viên để họ có thể phát triển ứng dụng di động cho cho công nghệ thực tế ảo trên nền tảng Android và Unity. Được biết bộ công cụ SDK nhằm dành cho Android, các ứng dụng trong tương lai sẽ hỗ trợ điều chỉnh tiêu cự của kính, theo dõi cử chỉ đầu người đeo và chế độ render hình ảnh 3D Side-by-side. Bên cạnh đó, Google còn cung cấp thêm bộ SDK dành cho nền tảng Unity, qua đó các ứng dụng tạo ra sẽ làm việc tốt với Metal của iOS. Đây là hướng đi đúng đắn nhằm phổ cập công nghệ thực tế ảo đang ở những bước đầu chập chững.

SDK VR của Google dành cho Android và Unity hoặc Unreal Engine 4 cho phép bạn nhanh chóng bắt đầu tạo các ứng dụng và trò chơi VR hoặc điều chỉnh ứng dụng hiện có của mình cho VR. Những công cụ này nhằm mục đích đơn giản hóa các tác vụ phát triển VR thông thường để bạn có thể tập trung vào việc xây dựng trải nghiệm phong phú mới của mình.

Các loại VR SDK:

+ [Google VR SDK for Android](https://developers.google.com/vr/android)

+ [Google VR SDK for Unity](https://developers.google.com/vr/unity)

+ [Google VR SDK for Unreal](https://developers.google.com/vr/unreal)

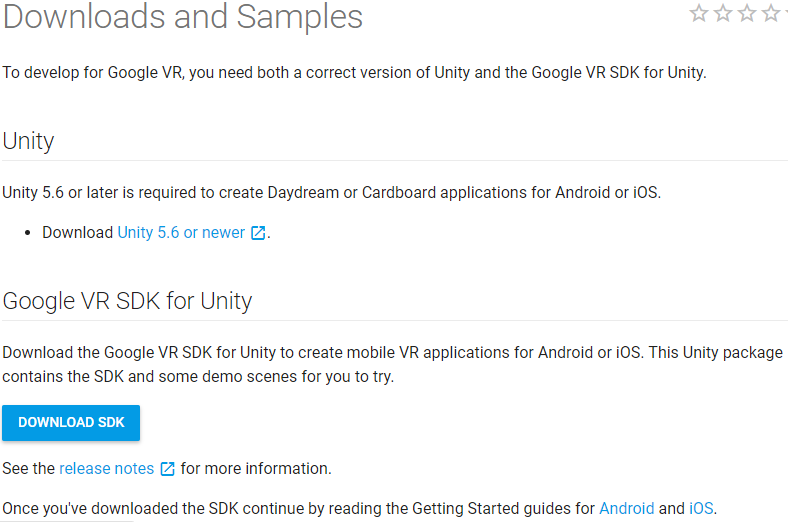
+ [Google VR SDK for iOS](https://developers.google.com/vr/ios)

+ [VR View](https://developers.google.com/vr/concepts/vrview)

**Giới thiệu về Google VR SDK sử dụng cho Unity:**

Google VR SDK - Unity cho phép bạn dễ dàng phát triển một ứng dụng Unity 3D với công nghệ thực tế ảo hoặc xây dựng trải nghiệm VR theo cách riêng của bạn.

Từ bản Unity 5.6 trở lên, VR SDK đã được tích hợp luôn vào unity. Đối với các bản khác thì cần tải sdk riêng từ trang chủ của google.



Hình 2.3: Trang web tải Google VR SDK cho Unity

Tính năng:

* Có thể phát triển một ứng dụng mới hoàn toàn.
* Có thể chuyển đổi ứng dụng Unity 3D có sẵn sang mode VR.

Google VR cung cấp:

* Theo dõi người dùng.
* Điều khiển âm thanh stereo.
* Phát hiện sự tương tác của người dùng với hệ thống (Thông qua trigger hoặc controller).
* Tự động cấu hình âm thanh stereo cho 1 trình xem VR.
* Điều khiển chế độ xem cho phù hợp với mắt của người dùng.(chỉnh lại tiêu cự của kính).
* Điều chỉnh điện thoại cho phù hợp với kính.
* Tự động điều chỉnh cảm biến gyro

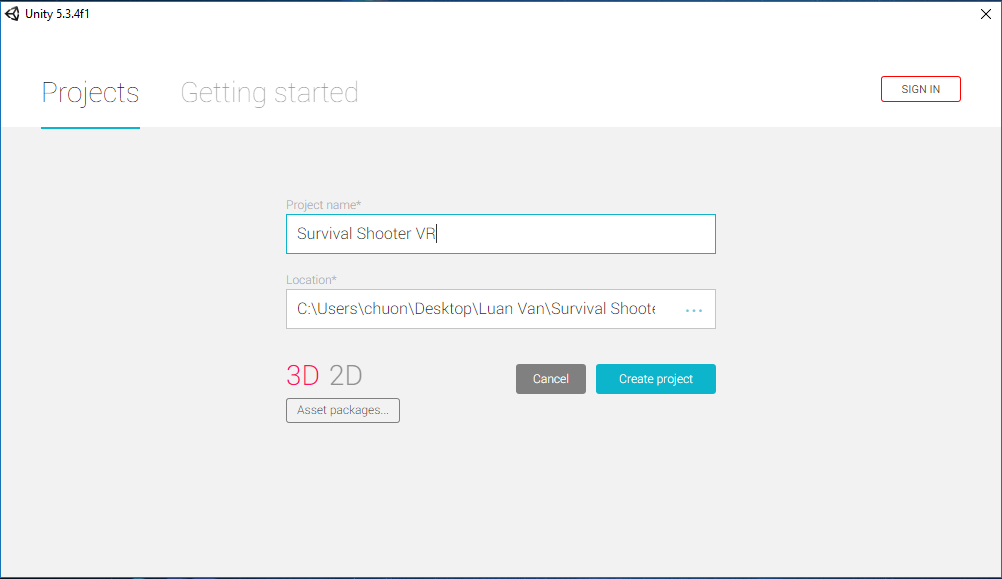
Tính năng Google VR SDK trong Unity:

* Hỗ trợ Daydreams controller.
* Âm thanh trong không gian.
* Một prefab thể hiện sự tương tác của người dùng.
* Có thể sử dụng chuột, các phím alt/control để điều khiển trong chế độ VR Emulation của Unity.
* Một prefab hiển thị FPS để hiển thị hiệu năng của ứng dụng

### **2.2.1.2 Tích hợp Google VR SDK vào project.**

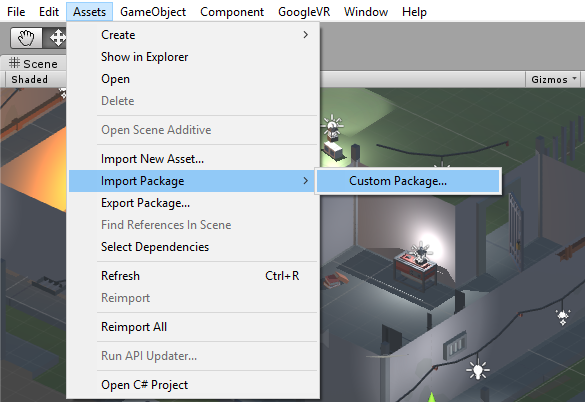
Tạo 1 project Unity và import SDK

* + Mở unity và tạo 3D project mới.



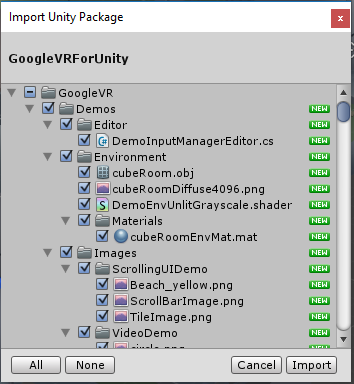
Hình 2.4: Giao diên tạo project mới

* + Chọn Assets > Import Package > Custom Package.



Hình 2.5: Hình …

* + Chọn GoogleVRForUnity\_\*.unitypackage file đã download.
  + Trên Importing Package dialog, chọn Import. Chấp nhận nâng cấp API nếu yêu cầu.



Hình 2.6: Cửa sổ import

Configure build settings

* + Chọn File > Build Settings.
  + Chọn Android và click Switch Platform.
  + Trên Build Settings window, chọn Player Settings. Thiết lập các cài đặt sau đây:

|  |  |
| --- | --- |
| Cài đặt | Giá trị |
| Player Settings > Other Settings > Virtual Reality Supported | Enabled |
| Player Settings > Other Settings > SDK | Click + và chọn Daydream hoặc Cardboard. |
| Player Settings > Other Settings > Minimum API Level | Daydream: Android 7.0 'Nougat' (API level 24) hoặc cao hơn.  Cardboard: Android 4.4 'Kit Kat' (API level 19) hoặc cao hơn. |

# CHƯƠNG 3. TỔNG QUAN VỀ UNITY 3D

## 3.1 Giới thiệu Unity

Unity là một phần mềm làm game đa nền tảng được phát triển bởi Unity Technologies, mà chủ yếu để phát triển video game cho máy tính, consoles và điện thoại. Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng.

### **3.1.1 Tổng quan**

Môi trường phát triển của Unity là trên Windows và Mac OS X, và các trò chơi mà nó tạo ra có thể chạy trên Windows, Mac, Xbox 360, Playstation 3, iPad, iPhone, cũng như nền tảng Android. Nó cũng có thể sản xuất trò chơi trên trình duyệt web với thành phần bổ sung Unity web player, đã hỗ trợ trên Mac và Windows nhưng không hỗ trợ trên Linux. Web player cũng được sử dụng để triển khai các Mac widget. Ngoài ra, Unity còn có khả năng xuất các trò chơi đến chức năng Stage 3D của Adobe trong Flash, nhưng một số tính năng mà web player không sử dụng được do hạn chế trong Flash.

Unity bao gồm một trình biên tập cho phát triển, thiết kế nội dung và một game engine để thực thi các sản phẩm cuối cùng. Unity tương tự như Director, Blender game engine, Virtools, Torque Game Builder, và Gamestudio, cũng sử dụng môi trường đồ họa tích hợp như phương pháp chính của sự phát triển.

Unity đã giành được giải thưởng Đổi mới công nghệ Wall Street Journal 2010 trong các loại phần mềm. Trong năm 2009, Công nghệ Unity được danh hiệu là một trong “Top 5 Các công ty Game của năm 2009” của Gamasutra cho Unity. Unity là một á quân cho việc sử dụng đồ họa tốt nhất trên hệ điều hành Mac OS X trong Giải thưởng thiết kế của Apple năm 2006.

Unity hỗ trợ đồ họa 2D và 3D, các chức năng được viết chủ yếu qua ngôn ngữ [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng). Hai ngôn ngữ lập trình khác cũng được hỗ trợ: Boo, đã bị loại cùng với việc phát triển Unity 5 và UnityScript bị loại vào tháng 8 năm 2017 sau khi phát hành Unity 2017.1. UnityScript là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh_k%E1%BB%8Bch_b%E1%BA%A3n) độc quyền có cú pháp tương tự [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Phần mềm nhắm mục tiêu các đồ họa APIs sau: Direct3D trên Windows và Xbox One; [OpenGL](https://vi.wikipedia.org/wiki/OpenGL) trên Linux, macOS, và Windows; OpenGL ES trên Android và iOS; WebGL trên web; và APIs độc quyền trên các [máy chơi video game](https://vi.wikipedia.org/wiki/Video_game_console). Ngoài ra, Unity hỗ trợ APIs cấp thấp như Metal trên iOS và macOS và Vulkan trên Android, Linux, và Windows, cũng như Direct3D 12 trên Windows và Xbox One. Trong 2D games, Unity cho phép nhập sprites và một renderer thế giới 2D tiên tiến. Đối với 3D games, Unity cho phép thiết lập các đập điểm kĩ thuật của các kết cấu và độ phân giải mà công cụ trò chơi hỗ trợ, cung cấp các hỗ trợ cho bump mapping, reflection mapping, parallax mapping, cảnh không gian ambient occlusion (SSAO), hiệu ứng bóng đổ bằng cách sử dụng shadow maps, render thiết lập toàn cảnh đến hiệu ứng. Unity cũng cung cấp các dịch vụ cho nhà phát triển, bao gồm: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity IAP, Unity Multiplayer, Unity Performance Reporting and Unity Collaborate.

Unity nổi bật với khả năng xây dựng trò chơi chạy trên nhiều nền tảng. Các nền tảng được hỗ trợ hiện nay là [Android](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)), Android TV, Facebook Gameroom, Fire OS, [Gear VR](https://vi.wikipedia.org/wiki/Samsung_Gear_VR), Google Cardboard, Google Daydream, HTC Vive, [iOS](https://vi.wikipedia.org/wiki/IOS), [Linux](https://vi.wikipedia.org/wiki/Linux), [macOS](https://vi.wikipedia.org/wiki/MacOS), [Microsoft HoloLens](https://vi.wikipedia.org/wiki/Microsoft_HoloLens), Nintendo 3DS family, Nintendo Switch, Oculus Rift, [PlayStation 4](https://vi.wikipedia.org/wiki/PlayStation_4), [PlayStation Vita](https://vi.wikipedia.org/wiki/PlayStation_Vita), PlayStation VR, [Samsung Smart TV](https://vi.wikipedia.org/wiki/Samsung_Electronics), [Tizen](https://vi.wikipedia.org/wiki/Tizen), tvOS, WebGL, [Wii U](https://vi.wikipedia.org/wiki/Wii_U), [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Windows Phone](https://vi.wikipedia.org/wiki/Windows_Phone), [Windows Store](https://vi.wikipedia.org/wiki/Windows_Store), và [Xbox One](https://vi.wikipedia.org/wiki/Xbox_One). Unity trước đây cũng đã hỗ trợ 7 nền tảng khác chạy trên Unity Web Player. Unity Web Player là một plugin của trình duyệt chạy trên Windows và OS X, đã bị gỡ bỏ vì lợi ích của WebGL.

Unity là bộ công cụ phát triển phần mềm mặc định (SDK) cho [máy chơi game video game](https://vi.wikipedia.org/wiki/Video_game_console) [Wii U](https://vi.wikipedia.org/wiki/Wii_U) của Nintendo, kèm theo bản miễn phí của Nintendo với mỗi giấy phép phát triển Wii U. Unity Technologies gọi việc kết hợp sản xuất SDK với một bên thứ ba là "industry first".

Ngoài ra khi design được một game thì bạn có thể xuất được ra một file.exe và có thể chạy và chơi được trên PC khác.

Một thế mạnh nữa của Unity là có thể chạy demo game ngay trongkhi design, nó có hỗ trợ hai chế độ là Scene và Game, rất thuận tiện cho việc test thử các modulGame.

Các loại giấy phép:

Có hai loại giấy phép chính: Unity và Unity Pro, với phiên bản Pro hiện có sẵn với một mức giá và phiên bản không Pro thì được miễn phí. Phiên bản Pro có nhiều tính năng bổ sung chẳng hạn như, render-to-texture, occlution culling, global lighting và các hiệu ứng post-processing. Phiên bản miễn phí, mặt khác, hiển thị một màn hình splash (trong các trò chơi độc lập) và một watermark (trong các trò chơi web) mà không thể tùy chỉnh hoặc vô hiệu hóa được.

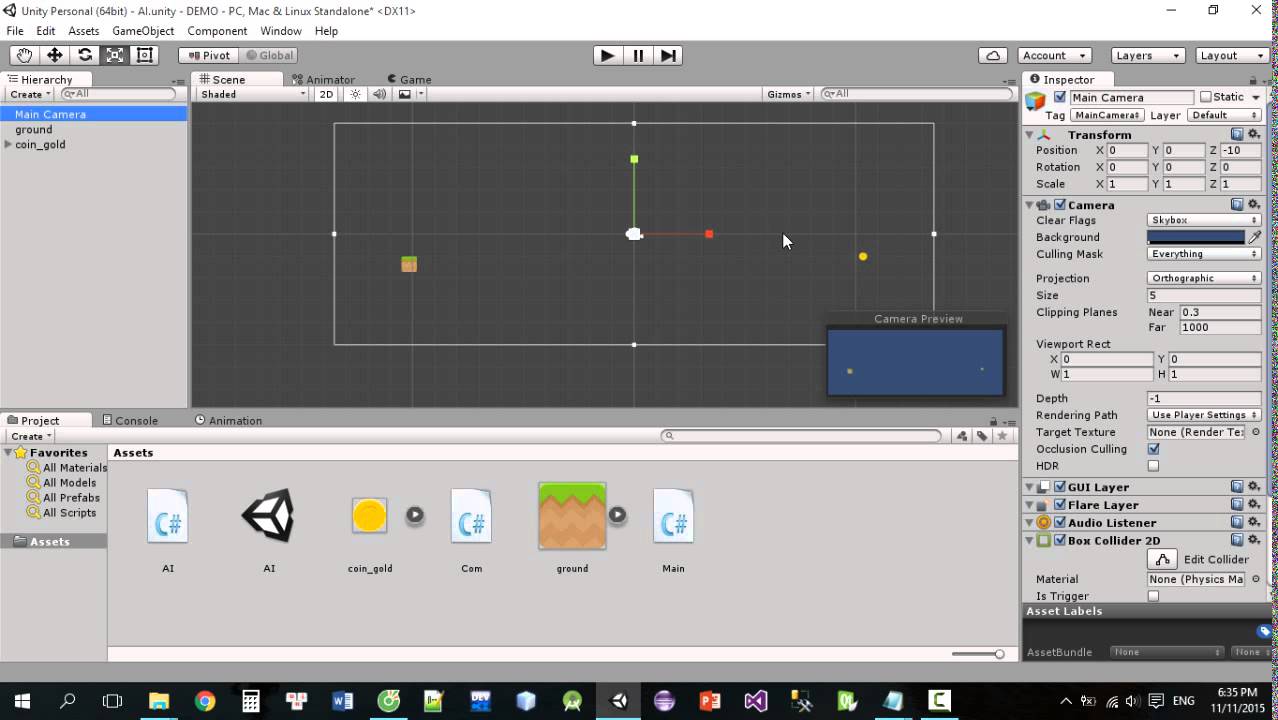
Cả Unity và Unity Pro đều bao gồm môi trường phát triển, hướng dẫn, các dự án mẫu và nội dung, hỗ trợ thông qua diễn đàn, và các cập nhật tương lai trong cùng một phiên bản chính (tức là mua Unity 3.0 được các cập nhật tất cả Unity Pro 3.x trong tương lai một cách miễn phí).

Unity cho iOS và Unity cho Android là tiện ích có thể mua thêm từ Unity. Một giấy phép Unity Pro được yêu cầu để mua một giấy phép iOS Pro hay Android Pro. Các giấy phép Android và iOS thông thường có thể được sử dụng với phiên bản miễn phí của Unity.

Các giấy phép Mã nguồn, PS3, Xbox 360, và Wii thì được thỏa thuận trên cơ sở từng trường hợp.

Giấy phép giáo dục được cung cấp bởi Studica với quy định rằng nó cho phép mua và sử dụng bởi các trường học, hoàn toàn cho nền giáo dục.

### **3.1.2** **Giới thiệu về giao diện**



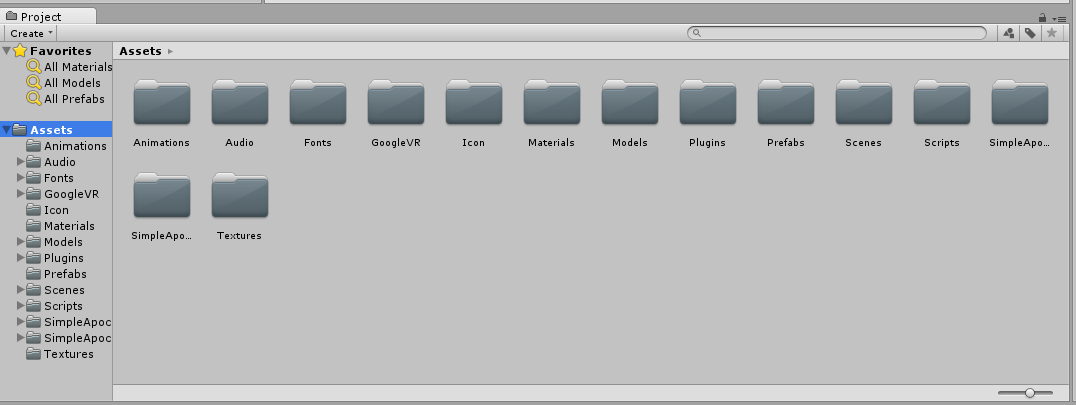
Hình 3.1 Giao diện mặc định của Unity

**3.1.2.1 Cửa sổ 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 asseets 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.

Ngoài ra cũng có thể thêm các assets và đổi tên các assets một cách dễ dàng.



Hình 3.2 Cửa sổ project Unity

Phần bảng bên trái là nơi hiển thị các thư mục theo cấu trúc cây thư mục. Khi nhấp chọn một thư mục ở bên cây thư mục thì các dữ liệu sẽ được hiển thị ở phần bảng bên phải. Các tài nguyên khác nhau sẽ hiển thị các biểu tượng khác nhau tùy thuộc vào định dạng của chúng (script, vật liệu, thư mục cha,...). Các biểu tượng có thể thay đổi kích cở bằng cách sử dụng thanh trượt nằm ở góc dưới bên phải. Nếu con trượt chạy về phía bên trái thì các biểu tưởng sẽ được hiển thị dưới dạng cây thư mục. Phần khoảng trống phía bên trái con trượt giúp hiển thị đường dẫn đầy đủ của file đang được chọn.

Phía trên danh sách cây thư mục là mục Favorite, đây là nơi bạn có thể truy cập nhanh đến các dữ liệu bạn thường xuyên sử dụng. Bạn có thể kéo thả dữ liệu từ trong danh sách cây thư mục vào mục Favourite và cũng có thể lưu trữ các tệp được tìm kiếm ở đó.

Phía trên mục bảng chọn hiển thị các tệp chính là đường dẫn hiển thị đường dẫn đến thư mục chứa tệp đang xem. Các thư mục được phân chia rỏ rệt để dễ dàng truy cập. Khi tìm kiếm, thanh này sẽ được thay thế để hiển thị khu vực đang tìm kiếm (thư mục chủ của tài nguyên, thư mục đang được chọn hoặc Kho tài nguyên) tùy thuộc vào tài nguyên trả phí hay miễn phí trong kho tài nguyên. Ở mục General trong Preferences của Unity có thể tắt hiển thị số lượt đếm của Kho tài nguyên nếu nó không cần thiết đối với bạn.



Hình 3.3 Thanh hiển thị đường dẫn



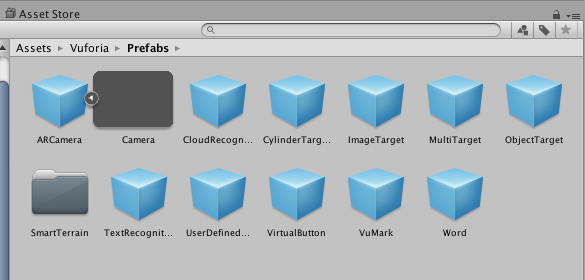
Hình 3.4 Thanh tìm kiếm

Phía bên trái thanh công cụ, mục Create giúp bạn thêm các tài nguyên mới và thư mục mới vào thư mục hiện tại. Nói tóm lại, nó giúp bạn tìm kiếm tài nguyên cho project của bạn.

Danh sách các mục chọn trong menu Window giúp bạn chuyển đổi dạng hiển thị về dạng bảng một cột, chỉ hiển thị cây thư mục mà không có các biểu tượng định dạng tệp. Biểu tượng khóa tệp nằm kế bên danh sách giúp bạn khóa tệp được chọn lại tránh trường hợp nó bị thay đổi khi làm việc.

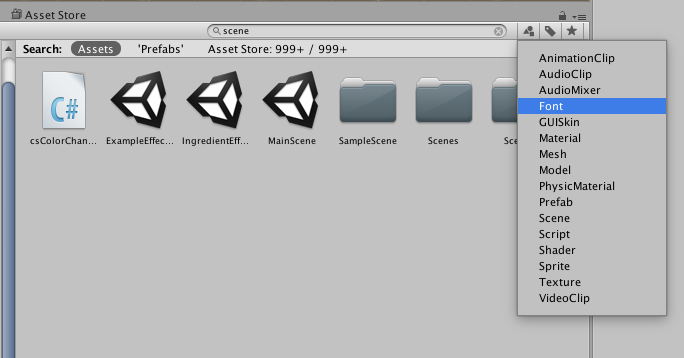
Tìm kiếm:

Chức năng tìm kiếm mạnh mẽ ngay cả khi các tài nguyên nằm trong các khu vực lớn hay các project không phù hợp. Thanh tìm kiếm sẽ lọc ra các tài nguyên phụ thuộc vào các kí tự trong ô tìm kiếm.



Hình 3.5 Khung tìm kiếm

Nếu bạn chỉ gõ 1 từ thì khung tìm kiếm sẽ hiển thị tất cả tệp liên quan, nếu bạn gõ coastal scene thì nó sẽ hiển thị những tài nguyên có tên liên quan đến từ "coastal" và "scene". Bên phải khung tìm kiếm có 3 biểu tượng nút. Cái đầu tiên giúp bạn lọc ra các tài nguyên phụ thuộc vào định dạng của chúng.

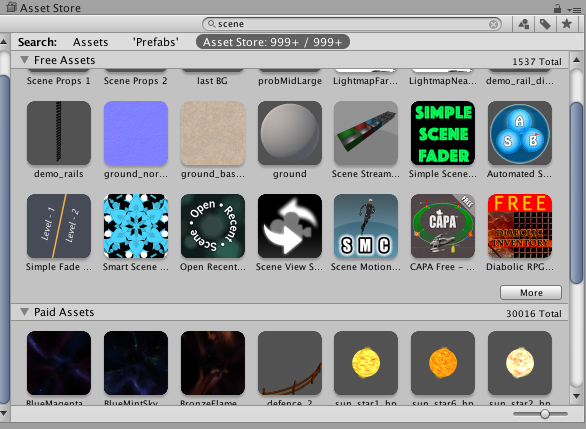


Hình 3.6 …

Nút biểu tượng kế tiếp giúp lọc theo nhãn của các tài nguyên (Nhãn được đặt cho các tài nguyên ở Thẻ Inspector). Số lượng lọc theo nhãn là rất lớn so với khung nhìn rất hạn hẹp. Có thể thêm vào các phần mở rộng như "t:" để lọc theo định dạng, hay "l:" để lọc theo nhãn. Có thể gõ trực tiếp lên khung tìm kiếm nếu bạn biết rõ dữ liệu đang tìm. Có thể tìm kiếm một lượt nhiều định dạng hay nhãn. Khi gõ thêm nhiều định dạng, tìm kiếm sẽ giao các kết quả tìm kiếm lại. Khi gõ thêm nhiều nhãn, bộ lọc sẽ hợp các kết quả tìm kiếm theo nhãn lại.

Tìm kiếm trong Kho tài nguyên

Thẻ Project còn cho phép tìm kiếm các tài nguyên đang hiện có trên Kho tài nguyên của Unity, các tài nguyên trả phí và miễn phí đều được hiển thị ở đây. Việc tìm kiếm tương tự như tìm tài nguyên trong một project của Unity. Công cụ tìm kiếm sẽ kiểm tra tên tài nguyên trước rồi đến gói tên, gói nhãn và gói mô tả theo tuần tự (Vì thế các tài nguyên có tên gần nhất so với từ trong ô tìm kiếm sẽ được xếp trước rồi mới đến các tài nguyên được tìm thấy trong phần gói mô tả bộ sản phẩm).

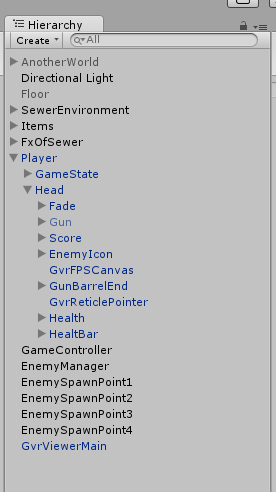


Hình 3.7 Asset Store của Unity

Nếu lựa chọn một tài nguyên từ danh sách tìm kiếm này, chi tiết mục trả phí hay tải về sẽ được hiển thị kèm theo bên dưới. Một vài tài nguyên còn có những phần mô tả để có thể tham khảo trước khi mua. Ngoài ra còn có các mục để xem các chi tiết phụ trong Kho tài nguyên.

**3.1.2.2 Hierarchy**

Tức là hệ thống phân cấp. 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 của bạn. Bạn 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.

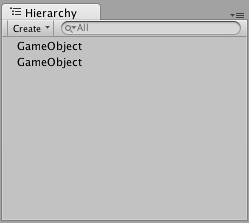


Hình 3.8 Tab Hierarchy

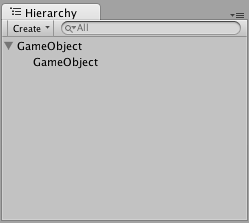
Thẻ Hierarchy bao gồm các GameObject (Các đối tượng làm nên một game hoàn chỉnh) trong Scene (một cảnh / một phân đoạn / một màn chơi riêng...) hiện hành. Nó bao gồm các tài nguyên như model 3D, prefabs (các gói tài nguyên), các đối tượng tự tạo góp phần tạo nên một game hoàn chỉnh. Có thể lựa chọn các đối tượng trong thẻ Hierarchy và kéo nó vào một đối tượng khác để tạo mối liên hệ cha con cho đối tượng. Khi một đối tượng đượt thêm hay xóa trong scene, nó sẽ xuất hiện hoặc biến mất trong thẻ Hierarchy.

Quan hệ cha con:

Nếu muốn bất kỳ GameObject thành con của một cái khác, ta có thể kéo thả nó vào đối tượng khác trong thẻ Hierarchy. Một đối tượng con sẽ di chuyển và xoay tròn giống như đối tượng cha. Có thể sử dụng tam giác nhỏ phía trước đối tượng cha để hiển thị và ẩn các đối tượng con nếu cần thiết.



Hình 3.9 Hai đối tượng GameObject bình thường



Hình 3.10 Đối tượng cha và đối tượng con

**3.1.2.3 Parenting.**

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.

Ví dụ parenting là Player như hình 3.3

**3.1.2.4 Toolbar.**

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

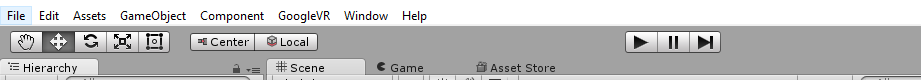
A:  Transform Tool: được dung với Scene view, như quay trái , phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.

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

C:  Play/Pause/Step Buttons: dung cho view game, chỵ game ngay trong Editor để kiểm tra.

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

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



Hình 3.4 Toolbar của Unity

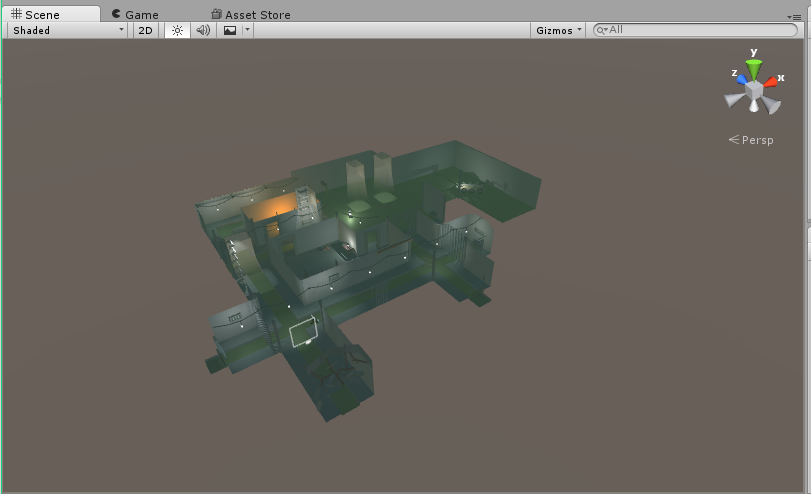
**3.2.1.5 Scene View.**

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 bạn Click vào hình vuông ở giữa hay giữ phím Shift + Click để chuyển đổi chế độ nhìn.

Khi xây dựng một Game, sẽ đặt rất nhiều đối tượng vào trong Game . Do đó có thể dử dụng các công cụ Transform Tools ở trong Toolbar để di chuyển, xoay, phóng to tu nhỏ từng đối tượng.Khi bạn chọn 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.Sau đó thay đổi đối tượng tùy ý, nếu muốn chính xác có thể chỉnh chi tiết ở bảng Inspector.

Scene View Control Bar. Thanh điều khiển Scene view cho phép xem screen Game với nhiều chế độ như đã phủ vật liệu (material), khung lưới(wireframe), RGB, Overdraw, … Cũng có thể xem và cả nghe những hiệu ứng ánh sáng, âm thanh, những thành phần động trong Game ngay tại Scene.



Hình 3.5 Tab Scene View

Khung nhìn Scene là nơi để chọn và bố trí các đối tượng như cảnh quan, người chơi, camera, kẻ địch và mọi đối tượng khác trong game. Sự bố trí hoạt cảnh với khung nhìn Scene là một chức năng quan trọng nhất của Unity, vì thế chúng ta có thể thực hiện việc này một cách nhanh chóng. Unity cung cấp các phím tắt cho hầu hết các chức năng thông dụng.

Định hướng khung nhìn Scene

Giữ chuột phải trong phần khung nhìn Scene để kích hoạt chế độ điều chỉnh góc nhìn. Sau khi giữ chuột phải, bạn hãy kết hợp với di chuyển chuột hoặc các phím WASD để di chuyển góc nhìn, phím Q và E để di chuyển góc nhìn lên và xuống.

Chọn bất kỳ một GameObject nào và ấn phím F để chuyển góc nhìn về đối tượng đã được chọn.

* Sử dụng các phím mũi tên để di chuyển trên mặt phẳng tọa độ X/Z.
* Giữ Alt và kéo thả chuột trái để xoay góc nhìn theo điểm trục.
* Giữ Alt và kéo thả bằng nút giữa chuột để di chuyển góc nhìn.
* Giữ Alt và kéo thả chuột phải để phóng to, thu nhỏ khung nhìn tương tự như bạn lăn chuột giữa.
* Bạn có thể sử dụng Công cụ thủ công (phím tắt: Q) nếu như bạn sử dụng chuột một nút.

 Nhấp chuột và kéo thả để di chuyển góc nhìn.

 Giữ Alt và kéo thả chuột để xoay góc nhìn quanh điểm trục.

 Giữ Ctrl và kéo thả chuột để phóng to thu nhỏ góc nhìn.

Phía trên bên phải khung nhìn Scene là biểu tượng Gizmo (trục tọa độ XYZ). Đây là nơi hiển thị góc nhìn định hướng hiển tại trong khung nhìn Scene, nó giúp Ta nhanh chóng thay đổi tọa độ góc nhìn.

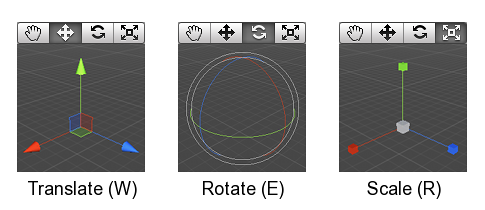


Hình 3.7 Trục tọa độ

Mỗi màu là một trục tọa độ khác nhau. Có thể nhấp vào bất kỳ màu nào để xoay góc nhìn về phía trục tọa độ tương ứng.

**Định hướng cho các GameObject:**

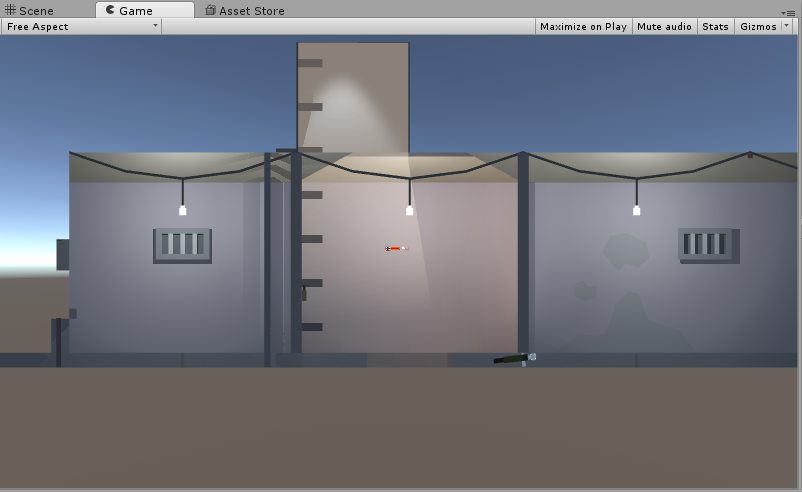
Khi xây dựng game, ta sẽ phải đặt rất nhiều đối tượng vào thế giới trong game. Để làm điều này ta phải sử dụng các công cụ biến đổi trên thanh công cụ như dịch chuyển, xoay và kéo dãn đối tượng. Mỗi công cụ đều có trục tọa độ được hiển thị xung quanh đối tượng được chọn. Ta có thể dùng chuột và thao tác bất kỳ trục tọa độ để biến đổi đối tượng, hoặc ta có thể gõ trực tiếp giá trị vào phần ô nhập giá trị trong thẻ Inspector.



Hình 3.8 Định hướng game Object

**3.2.1.6 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.Có thể 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.



Hình 3.9 Tab Game View

**3.2.1.7 Play Mode.**

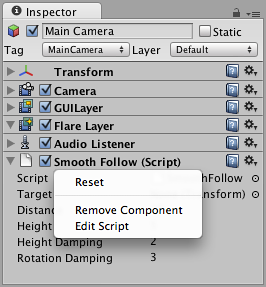
Sử dụng những nút trên Toolbar để điều khiển Editor Play Mode, và xem trước Game sẽ như thế nào khi chơi. Trong chế Play, mọi giá trị thay đổi sẽ được lưu tạm, và bị xóa khi thoát khỏi chế độ play.

**3.2.1.8 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 bạn có thể điều chỉnh, thiết lập mọi thông số chức năng của những mối liên kết GameObject-Component.

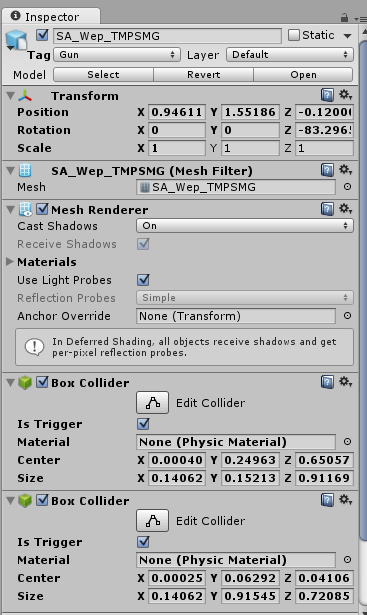
Mọi thuộc tính thể hiện trong Inspector đều có thể được tùy biến một cách trực tiếp. Ngay cả với những biến trong script cũng có thể được hiệu chỉnh mà không cần xem mã. Trong script, nếu định nghĩa một giá trị là public cho một kiểu đối tượng (như GameObject hay Transform), thì có thể drag-drop một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



Hình 3.10 Component của gameobject

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



Hình 3.11 Tab Inspector

Sử dụng thanh sổ Layer để sắp xếp các lớp sẽ được hiển thị..Thanh sổ của mục Tag giúp bạn đặt tag cho đối tượng được chọn.

**Labels (Nhãn)**

Unity cho phép các tài nguyên được gắn Label để dễ dàng truy cập và phân loại. Mục nằm dưới cuối cùng của  thẻ Inspector là bảng lựa chọn nhãn cho tài nguyên.



*3.12 Asset Labels*

**3.2.1.9 Tùy biến không gian làm việc.**

Chúng ta có thể tùy ý thay đổi cấu trúc của các Views bằng cách click-dragging những tab tới những nơi chúng ta muốn. Đặt cạnh một View nào đó, nó sẽ tự phân chia windows ra, đặt ra ngoài nó sẽ tự động tạo thành một windows riêng

Các tab cũng có thể kéo ra ngoài cửa sổ chính và được sắp xếp theo ý thích người dùng. Nếu cần thiết, chúng ta cũng có thể cho nó nằm ngang hàng với các Views khác trong cửa sổ chính.

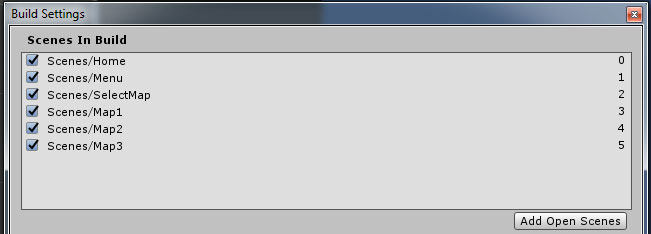
Chúng ta cũng có thể lưu lại những thiết lập về cấu trúc của các Views, bằng cách click vào menu Dropdown Layout ở thanh Toolbar, chọn Save Layout... Đặt tên cho nó và sau đó bạn dễ dàng gọi nó ra bằng cách chọn nó trong Dropdown Menu Layout.

## 3.2 Một số khái niệm cơ bản trong Unity

**3.2.1 Scene.**

Trong Unity, một cảnh chơi (hoặc một phân đoạn) là những màn chơi riêng biệt, một khu vực trong game hoặc thành phần có trong nội dung của trò chơi (các menu). Các thành phần này được gọi là Scene. Bằng cách tạo ra nhiều Scene, chúng ta có thể phân phối thời gian và tối ưu tài nguyên, kiểm tra các phân đoạn trong game một cách độc lập.

Ví dụ ở game này đã phân các phân cảnh Home, Menu, SelectMap, Map1, Map2, Map3. Bạn thấy đấy, nếu phân như vậy thì mình sẽ dễ quản lý hơn, ta gọi đây là các Scene.



Hình 3.13 Các Scenes Unity

**3.2.2 GameObject**

Để có thể sử dụng các tài nguyên chúng ta cần phải tạo ra các GameObject (đối tượng game). Luôn luôn có thành phần là Transform xác định Position(ví trí), Rotate(độ xoay) và Scale (độ phóng đại)

Ví dụ trong game đá banh thì ta sẽ có Player, Goal, Ball, Stadium,.. và bạn lưu ý rằng luôn phải có Camera, Camera quay cái gì thì người nhìn thấy cái đó.

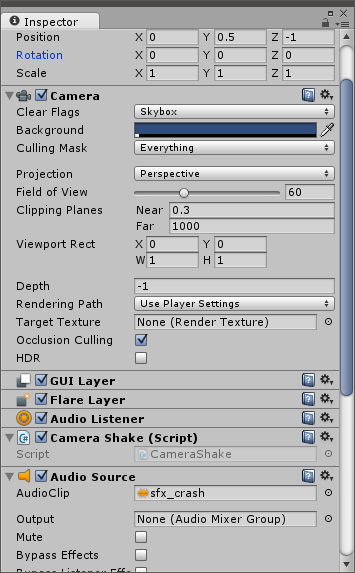


Hình 3.14 Các Game Object

**3.2.3 Components.**

Các GameObject đóng vai trò là các đối tượng trong game, mỗi GameObject sẽ chứa các thành phần thể hiện thuộc tính của nó như vị trí, góc xoay, tỷ lệ, xử lý chức năng… Các thành phần chứa trong GameObject chính là các Component.

Ví dụ trong GameObject Camera có Component Transform lưu thông tin về Position, Rotation, Scale, có Component Camera xử lý các chức năng của Camera,  có Component Audio Listener xử lý lắng nghe âm thanh,…

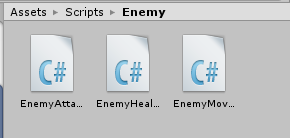


Hình 3.15 Các Components chứa trong một GameObjectCamera

**3.2.4 Scripts**

Script được Unity xem như một Component. Đây là thành phần thiết yếu trong quá trình phát triển game. Bất kỳ một game nào, dù đơn giản nhất đều cần đến Scripts để tương tác với các thao tác của người chơi, hoặc quản lý các sự kiện để thay đổi chiều hướng của game tương ứng với kịch bản game. Unity cung cấp cho lập trình viên khả năng viết Script bằng các ngôn ngữ: JavaScript, C#.

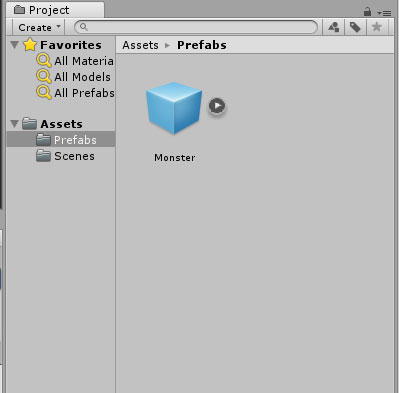
Để viết Script, chúng ta có thể biên tập Script trên Mono Developer được tích hợp mặc định vào Unity hoặc bạn có thể cài một trình biên tập riêng như Visual Studio, Visual Code.



Hình 3.16 Script C# trong Untiy

**3.2.5 Prefabs**

Prefabs thực chất là Game Object được lưu trữ lại để tái sử dụng, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay từ môt đối tượng ban đầu. Các Game Object được nhân bản từ một prefab sẽ giống nhau hoàn toàn. Để tạo ra một prefab, ta đơn giản chỉ cần kéo một Game Object vào cửa sổ Project.



Hình 3.17 Prefab trong Unity

**CHƯƠNG 4. THIẾT KẾ GAME**

* 1. Thiết kế game
     1. Tổng quan

- Survival shooter VR là một game thực tế ảo.

- Game là thể loại bắn súng sinh tồn, người chơi sẽ được trải nghiệm khung cảnh chiến trường và cảm giác chiến đấu khi nhập vai vào nhân vật.

- Người chơi sẽ điều khiển nhân vật của mình di chuyển theo hướng của đầu người head) người chơi, cũng như nhặt súng và các vật dụng để đối đầu và tiêu diệt lũ quái enemy.

- Vận dụng hết khả năng của mình để chiến đấu đến cùng nhằm sống sót và tiêu diệt đủ số lượng enemy để trở thành người chiến thắng cuộc chơi.

* + 1. Mục tiêu

- Ứng dụng công nghệ thực tế ảo để tạo ra game có trải nghiệm tốt hơn và chân thật hơn bao giờ hết.

- Sử dụng tối đa sự hỗ trợ của Unity và Google VR Sdk để làm game VR, từ đó

đưa ra nhận xét, cảm nhận, ưu và khuyết điểm của Unity Engine và VR Sdk của Google.

- Tạo ra game cho lứa tuổi và đối tượng phù hợp chơi, từ đó có thể cảm nhận

và thích thú với game VR hơn. Giúp người chơi giải trí sau những giờ học tập, làm

việc mệt mỏi, từ đó cũng giúp mình hiểu được nhu cầu, thị hiếu của người chơi

hơn.

- Các nhà phát triển, các hãng điện thoại, các cửa hàng đang tập trung lớn vào

mảng game VR nên việc tạo ra các sản phẩm VR sẽ được hỗ trợ nhiệt tình hơn

cũng như lượng người dùng tiếp cận cũng được các store ưu đãi giúp tiếp cận

người dùng và có thể giúp có được tập user lớn cũng như góp phần tăng lợi nhuận

sau này.

* + 1. Tính năng

- Game có góc nhìn thực tế ảo mới lạ và thú vị.

- Hỗ trợ tay cầm giúp người chơi điều khiển dễ dàng hơn.

- Nhìu loại enemy cũng như súng và vật dụng.

- Âm thanh kịch tính, sống động.

* + 1. Cách chơi

- Nhân vật có lượng máu nhất định và sẽ chết nếu máu về không.

- Người chơi sẽ vào vai nhân vật trong game dưới góc nhìn camera của game VR.

- Sử dụng tay cầm để điều khiển nhân vật di chuyển cũng như bắn và nhặt đồ vật.

- Người chơi sẽ đối đầu với enemy được sinh ra từ các hướng 1 cách ngẫu

nhiên, người chơi có thể nhặt được súng,… dưới măt đất lên và bắn tiêu diệt

các con enemy.

- Mỗi khi tiêu diệt được enemy thì sẽ hiển thị lên camera số enemy đã tiêu diệt được.

- Sau khi tiêu diệt đủ số lượng zombie người chơi có thể qua màn tiếp theo.

* + 1. Mindset

- Với góc nhìn thực tế ảo, tạo nên một cảm giác chân thật cho người chơi hơn bao giờ hết thông qua kính Google Cardboard.

- Hỗ trợ thêm tay cầm giúp việc điều khiển trong game một cách dễ dàng thay vì chỉ một nút bấm của kính Google Cardboard.

- Độ khó game tăng dần, người chơi sẽ thích ứng dễ dàng hơn.

* + 1. Đối tượng người chơi

Game tập trung vào các đối tượng người chơi:

* Mọi giới tính.
* Độ tuổi từ 10 tuổi trở lên
* Có sử dụng android hoặc iOS
* Yêu thích thể loại game bắn súng và hay khám phá cái mới
  1. Công nghệ
     1. Luồng màn hình
* Loading Screen
  + Title Screen
    - Play Game
      * Level Select
        + Level

Level Complete

Next Level

Level Select

* + - Load Game
    - Options
      * Controls
      * Graphics
      * Sounds

4.2.2 Điều khiển

Sử dụng controller gamepad tay cầm kết nối thông qua bluetooth để điều khiển.



Hình : Tay cầm gamepad bluetooth

Điều khiển di chuyển:

+ Đi tới: joystick button 3

+ Đi lùi: joystick button 0

+ Qua trái: joystick button 2

+ Qua phải: joystick button 1

Điều khiển nhặt vật, súng: joystick button 5

Điều khiển bắn: joystick button 4

Hướng di chuyển của nhân vật phụ thuộc vào hướng xoay của đầu (head) của người chơi.

4.2.3 Cơ học (game mechanics)

Nhận vật, thành phần trong game:

- Enemy khi tấn công vào người chơi sẽ bị trừ đi một lượng máu.

- Player bắn vào enemy đến khi enemy hết máu thì enemy sẽ bị tiêu diệt.

- Các công trình, vật cản sẽ không bị phá huỷ.

- Zombie và Player sẽ không có khả năng đi xuyên qua các công trình,

vật cản.

- Enemy khi hết máu sẽ chết và biến mất.

- Player khi chết thì game sẽ được coi là thua và kết thúc game.

- Khi bắn trung enemy sẽ tạo effect khói bụi trên enemy.

4.3 Các yếu tố Game Play

4.3.1 Game Physics and Statistics

- Bản đồ 3D là nơi nhân vật có thể di chuyển trên đó, bản đồ có các căn phòng và bức tường được gắn collider và physic để giới hạn các đường biên xung quanh, ngăn cho nhân vật cũng như các Zombie không thể di chuyển qua được.

- Các Static Physic là các đối tượng trên bản đồ như nhà, tường, công trình,...

những đối tượng này là đối tượng tĩnh, không thể di chuyển được và nhân vật

người chơi cũng như các Zombie không có khả năng đi chuyên hay phá huỷ các

đối tượng này. Vì vậy di chuyển gặp những đối tượng này thì nhân vật người chơi

cũng như các enemy sẽ phải tìm đường đi mới phù hợp.

-

4.3.2 Trí tuệ nhân tạo

-

4.4 Thiết kế màn chơi

4.4.1 Các đối tượng

* Static level pieces
  + Môi trường xung quanh.
* Interactive/dynamic level pieces
  + Zombie
  + Nhân vật
  + Súng và vật phẩm

4.4.2 Chủ đề màn chơi

Ống cống thoát nước

* 1. Objects
     1. Ambience
        1. Nhà cửa
        2. Ống thoát nước
        3. Các vật nhỏ khác
     2. Interactive
        1. Súng
        2. Vật phẩm
        3. Gậy
  2. Challenges
     1. Có các loại enemy xanh và hồng và boss vàng, số lượng tăng dần theo thời gian.

4.4.3 Luồng chạy game

Những hành động mà nhân vật có thể thực hiện:

- Di chuyển theo tất các hướng trong map 3D.

- Bắn và nhặt đồ vật bằng việc sử dụng tay cầm.

- Tiêu điệt các enemy xuất hiện từ mọi hướng.

- Khi tiêu diệt được đủ số lượng enemy thì sẽ được qua màn tiếp theo

- Nâng cấp súng.

4.5 Development

4.5.1 Thiết kế engine

- Sử dụng sự hỗ trợ của engine Unity3D.

- Google VR Sdk giúp hỗ trợ tạo ra game thực tế ảo.

4.5.2 Các lớp / đối tượng

* UIControl:
  + Xử lý các sự kiện liên quan tới sự kiện canvas.
* Sound Controller:
  + Xử lý các sự kiện liên quan tới âm thanh.
* Player Controller
  + Xử lý ii chuyển theo các hướng bằng tay cầm.
* Player Health
  + Xử lý thanh máu của người chơi.
* Enemy Health
  + Xử lý máu enemy/
* Enemy Movement
  + Enemy tự động tìm đường di tới vị trí của nhân vật thông qua sự hỗ trợ của NavMesh trong Unity/
* Enemy Attack
  + Xử lý việc tấn công của enemy vào nhân vật/
* Enemy Manager
  + Quản lý việc sinh tự động ngẫu nhiên loại enemy ra từ các hướng và vị trí khác nhau.
* Score Manager
  + Quản lý điểm.
* Game over Manager
  + Quản lý hiển thị gameover lúc thua.
* Menu Manager
  + Quản lý các sự kiện ở menu.
* Game Controller
  + Xử lý bắn cho nhân vật, các effect.
  + Xử lý nhặt đồ vật.
  + Tính điểm.

4.6 Development schedule

* Basic Game-specific Objects
  + Player
  + Enemy
  + Súng
  + Gậy
* Deadly Obstacles
  + Môi trường xung quang tùy vào map như ống cống thoát nước,…
* Enemies
  + Enemy Base Class
  + Zombie đánh gần
  + Zombie hồng
  + Zombie xanh
  + Zombie vàng

4.7 Visual Design

4.7.1 Color pallete

- Sử dụng gam màu tối thiêng hướng game kinh dị.

4.7.2 Style attributes / elements

-

4.7.3 Game objects

-

4.7.4 Game scenery

Asset dùng để thiết kế map cho game lấy từ nguồn trên mạng, có nhiều model theo nhiều chủ đề phù hợp với game zombie.

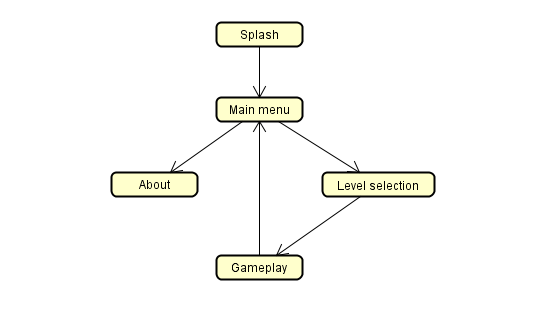
4.7.5 Hiệu ứng

- Hiệu ứng cho game sử dụng particle system, line renderer của unity.

- Các loại hiệu ứng lúc bắn súng, khói, bụi, ánh sáng,…

4.8 User Interface

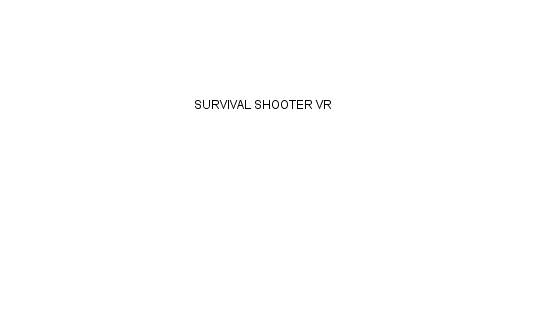
4.8.1 Flow Chart



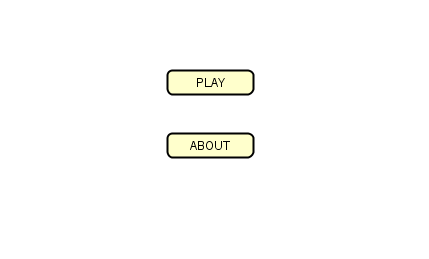
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Chức năng** |
| 1 | Splash | Màn hình mở đầu của game dùng để giới thiệu đồng thời load data, hiển thị tên game và logo sau đó sẽ chuyển sang màn hình Main Menu. |
| 2 | Main Menu | Hiển thị tất cả các lựa chọn chơi game, cài đặt, chọn nhân vật, thông tin nhóm, chơi game và cho người dùng lựa chọn để đến các màn hình mong muốn. |
| 3 | Level Selection | Hiển thị 2 màn chơi cho người dùng chọn. |
| 4 | Play | Hiển thị các chế độ chơi và khi nhấp vào nút play thì sẽ chuyển sang màn hình chọn map để bắt đầu trò chơi. |
| 5 | Setting | Cho phép điều chỉnh âm thanh, nhạc nền trong game. |
| 6 | About | Hiển thị thông tin của nhóm và những thông tin khác. |
| 7 | Game Play | Nơi game diễn ra và sự tương giác giữa người chơi với nhân vật chính và những đối tượng trong map.  Nó hiển thị thanh máu, nhân vật, enemy, map với góc nhìn VR. |

4.8.2 Thiết kế màn hình

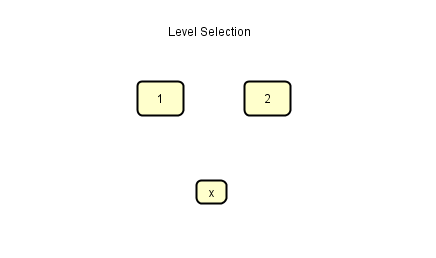
#### Splash



#### Main menu



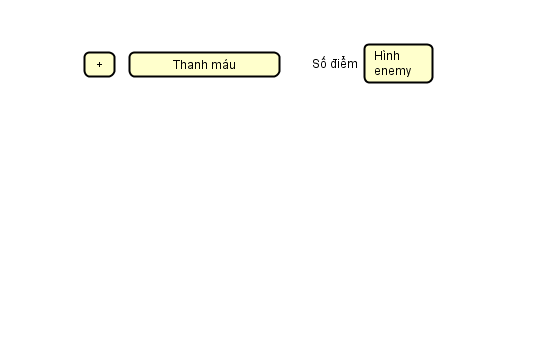
#### Level selection



#### About



#### UI trong Gameplay



4.9 Đồ họa

4.9.1 Mục tiêu tổng quát

- Thể hiện được khung cảnh của một nhà trong cống thoát nước, với những căn phòng, …

- Màu sắc tông màu tối, nhiều không gian góc cạnh, phù hợp với style low poly của game. Cho nên sẽ chọn những gam màu tối cho enemy và màu sáng cho người chơi.

- Những hình ảnh như nhà, ống cống, … sẽ có màu phù hợp như những màu ở khu nhà ở cống khoát nước để tạo khung cảnh hơi đáng sợ và một chút rùng rợn.

4.9.2 3D Art & Animation

- 3D Art:

+ Các model nhà cửa.

+ Model súng và các vật dụng.

+

- Animation:

+

+

+

4.9.3 GUI

- Thiết kế giao diện người dùng:

+

+

+

+

4.9.4 Địa hình

- Map trong game là trong ống cống thoát nước.

- Địa hình bằng phảng.

- Trong căn nhà nhà nhiều phòng ngăn biệt.

4.9.5 Hiệu ứng

- Hiệu ứng cho game sử dụng particle system, line renderer của unity.

- Các loại hiệu ứng lúc bắn súng, khói, bụi, ánh sáng,…

4.10 Thiết kế nhạc nền / âm thanh

4.10.1 Phong cách music / elements

- Âm thanh theo chủ đề hồi hộp, rung rợn.

4.10.2 Phong cách sound / elements

- Âm thanh kịch tính,…

4.10.3 Game music

- Nhạt nền kịch tính.

4.10.4 Game sound

- Âm thanh zombie gào thét, lúc bị bắn chết và tiếng bắn của súng.