

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM**

**BÙI QUANG HÀ
NGUYỄN ĐÌNH CHƯƠNG**

**KHÓA LUẬN TỐT NGHIỆP
XÂY DỰNG GAME VR SURVIVAL SHOOTER BẰNG UNITY
TRÊN GOOGLE CARDBOARD**

Building a VR Survival Shooting game with Unity and Google Cardboard

KỸ SƯ NGÀNH KỸ THUẬT PHẦN MỀM

TP. HỒ CHÍ MINH, 2018

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

BÙI QUANG HÀ
NGUYỄN ĐÌNH CHUÔNG

KHÓA LUẬN TỐT NGHIỆP
XÂY DỰNG GAME VR SURVIVAL SHOOTER BẰNG UNITY
TRÊN GOOGLE CARDBOARD

Building a VR Survival Shooting game with Unity and Google Cardboard

KỸ SƯ NGÀNH KỸ THUẬT PHẦN MỀM

GIẢNG VIÊN HƯỚNG DẪN
Th.S NGUYỄN VĨNH KHA

TP. HỒ CHÍ MINH, 2018

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

..... – Chủ tịch.

..... – Thư ký.

..... – Ủy viên.

..... – Ủy viên.

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày.....tháng.....năm.....

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
(CỦA CÁN BỘ HƯỚNG DẪN)

Tên khóa luận:

**XÂY DỰNG GAME VR SURVIVAL SHOOTER BẰNG UNITY TRÊN GOOGLE
CARDBOARD**

Nhóm SV thực hiện:

Cán bộ hướng dẫn:

Bùi Quang Hà

13520222

Nguyễn Đình Chương

13520086

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang

Số chương

Số bảng số liệu

Số hình vẽ

Số tài liệu tham khảo

Sản phẩm

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung nghiên cứu:

3. Về chương trình ứng dụng:

4. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm từng sinh viên:

Bùi Quang Hà :...../10

Nguyễn Đình Chương:...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày.....tháng.....năm.....

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
(CỦA CÁN BỘ PHẢN BIỆN)

Tên khóa luận:

**XÂY DỰNG GAME VR SURVIVAL SHOOTER BẰNG UNITY TRÊN GOOGLE
CARDBOARD**

Nhóm SV thực hiện:

Cán bộ phản biện:

Bùi Quang Hà

13520222

Nguyễn Đình Chương

13520086

Đánh giá Khóa luận

5. Về cuốn báo cáo:

Số trang

Số chương

Số bảng số liệu

Số hình vẽ

Số tài liệu tham khảo

Sản phẩm

Một số nhận xét về hình thức cuốn báo cáo:

6. Về nội dung nghiên cứu:

7. Về chương trình ứng dụng:

8. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm từng sinh viên:

Bùi Quang Hà :...../10

Nguyễn Đình Chương:...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Đầu tiên, nhóm tác giả xin gửi 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 tác giả thực hiện.

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỀ TÀI: XÂY DỰNG GAME VR SURVIVAL SHOOTER BẰNG UNITY TRÊN GOOGLE CARDBOARD

Cán bộ hướng dẫn: ThS. Nguyễn Vĩnh Kha

Thời gian thực hiện: Từ ngày.....đến ngày.....

Sinh viên thực hiện:

Bùi Quang Hà - 13520222

Nguyễn Đình Chương - 13520086

Nội dung đề tài:

Mục tiêu:

- Áp dụng các kiến thức về thiết kế và phát triển Game trong quá trình học để xây dựng sản phẩm.
- Nghiên cứu, tìm hiểu game engine Unity3D và thư viện Google VR SDK nhằm xây dựng một game bắn súng triển khai được công nghệ thực tế ảo trên thiết bị di động và phần cứng giá rẻ Cardboard.
- Chức năng dịch chuyển để giúp người dùng tương tác với không gian ảo thông qua công nghệ VR. Ngoài ra, người chơi còn có thể kết nối với tay cầm thông qua bluetooth để có thể trải nghiệm game một cách dễ dàng hơn.
- Thiết kế được các màn chơi để tạo ra cảm giác thích thú, tránh cảm giác nhàm

chán khi chỉ chơi 1 màn.

- Tạo được cảm giác mới lạ, thú vị, có chút sợ hãi khi chơi game với những âm thanh và quái vật đáng sợ.
- Sản phẩm có thể chạy mượt trên các thiết bị android và iOS.

Đối tượng:

Game tập trung vào các đối tượng người chơi:

- Độ tuổi từ 16 tuổi trở lên.
- Mọi giới tính.
- Có sử dụng thiết bị android hoặc iOS.
- Yêu thích thể loại game bắn súng và hay khám phá cái mới

Phương pháp thực hiện:

Công nghệ sử dụng: Unity Engine, Google VR SDK.

Kết quả mong đợi:

- Giúp người chơi có thể di chuyển trong không gian ảo.
- Kết nối với tay cầm thông qua bluetooth để điều khiển nhân vật.
- Người chơi có thể nhập vai thành nhân vật, sống trong môi trường ảo để tiêu diệt các loại quái, bảo vệ bản thân và sinh tồn.
- Người chơi có thể sử dụng súng để bắn xa hoặc gậy để đánh gần khi có quái vật xuất hiện.
- Có vật phẩm tăng máu khi tiêu diệt được enemy.

Kế hoạch thực hiện:		
Thời gian	Công việc	Phân công
Tuần 1 (4/9 - 10/9)	Đề xuất đề tài + tên cụ thể của đề tài	Chương + Hà
Tuần 2 (11/9 - 17/9)	- Tìm hiểu về công nghệ Virtual Reality	Chương + Hà
Tuần 3 (18/9 - 24/9)	- Tìm hiểu Google VR SDK cho Unity - Các yếu tố tương tác với Google Cardboard - Tích hợp Google VR SDK vào project	
Tuần 4 (25/9 - 1/10)	Viết tài liệu thiết kế game chi tiết, các màn chơi, nhân vật và cách chơi.	Chương + Hà
Tuần 5 (2/10 - 8/10)	Cài đặt Environment và camera VR	Chương + Hà
Tuần 6 (9/10 - 15/10)	Xử lý nhân vật người chơi di chuyển, nhặt vũ khí	Chương + Hà
Tuần 7 (16/10 - 22/10)	Tạo các loại Enemy với các hành vi, animation, trạng thái và tự động tìm đường đi tới nhân vật	Chương + Hà

Tuần 8 (23/10 - 29/10)	Health HUD và xử lý thanh máu của nhân vật lên màn hình	Chương
Tuần 9 (30/10 - 5/11)	Thêm súng cho phép người chơi bắn enemy bằng cách sử dụng kỹ thuật raycasting, line Renderers,...	Chương
Tuần 10 (6/11 - 12/11)	<ul style="list-style-type: none"> - Thêm gậy, xử lí đánh gần cho player - Tính số enemy mà người chơi tiêu diệt được và hiển thị lên camera VR. 	Hà
Tuần 11 (13/11 - 19/11)	<ul style="list-style-type: none"> - Tạo màn hình game over lúc thua và thắng màn chơi - Thêm vật phẩm tăng máu. 	Hà
Tuần 12 (20/11 - 26/11)	Hỗ trợ điều khiển game bằng tay cầm kết nối thông qua bluetooth	Chương
Tuần 13 (27/11 - 3/12)	Gắn âm thanh và nhạc nền	Chương
Tuần 14 (4/12 - 10/12)	Kiểm thử, hoàn thiện báo cáo.	Chương + Hà
Tuần 15 (11/12 - 17/12)	Triển khai demo game, hoàn thiện báo cáo	Chương + Hà

Tuần 16 (18/12 - 24/12)	Hoàn thành game, hoàn thành báo cáo	Chương + Hà
Tuần 17 (25/12 - 31/12)	Hoàn thành luận văn chuẩn bị bảo vệ	Chương + Hà
<p align="center">Xác nhận của CBHD (Ký tên và ghi rõ họ tên)</p>		<p align="center">TP. HCM, ngày....thángnăm..... Sinh viên (Ký tên và ghi rõ họ tên)</p>

MỤC LỤC

CHƯƠNG 1. MỞ ĐẦU.....	2
1.1 Giới thiệu đề tài.....	2
1.2 Lý do chọn đề tài.....	2
1.3 Mục tiêu đề tài.....	3
1.4 Ý nghĩa đề tài.....	3
1.4.1 Đối với người chơi nói riêng và ngành công nghiệp game nói chung.....	3
1.4.2 Đối với nhóm.....	3
1.5 Nội dung thực hiện.....	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	5
2.1 Thực tế ảo là gì ?.....	5
2.1.1 Lịch sử phát triển của công nghệ thực tế ảo.....	6
2.1.2 Các đặc tính chính của VR.....	7
2.1.3 Các thành phần một hệ thống VR.....	7
2.1.3.1 Phần cứng (Hardware).....	8
2.1.3.2 Phần mềm (Software).....	8
2.1.4 Các thiết bị cơ bản.....	9
2.1.4.1 Thiết bị định hướng và chuyển động.....	9
2.1.4.2 Thiết bị tương tác và phản hồi.....	12
2.1.5 Một số ứng dụng chính của VR.....	13
2.1.5.1 Quân sự.....	14
2.1.5.2 Giáo dục.....	15
2.1.5.3 Xây dựng.....	15
2.1.5.4 Thể thao.....	16
2.1.5.5 Du hành không gian.....	16

2.1.5.6 Y học.....	17
2.2 Giới thiệu Google Cardboard.....	17
2.2.1 Tìm hiểu chung về Google Cardboard.....	17
2.2.2 Ứng dụng.....	20
2.3 Google VR SDK.....	21
2.4 Giới thiệu Unity.....	23
2.4.1 Tổng quan.....	23
2.4.2 Giới thiệu về giao diện.....	26
2.4.2.1 Cửa sổ project.....	26
2.4.2.2 Hierarchy.....	28
2.4.2.3 Parenting.....	29
2.4.2.4 Toolbar.....	29
2.4.2.5 Scene View.....	29
2.4.2.6 Game View.....	32
2.4.2.7 Play Mode.....	32
2.4.2.8 Inspector.....	33
2.4.2.9 Tùy biến không gian làm việc.....	35
2.4.3 Một số khái niệm cơ bản trong Unity.....	35
2.4.3.1 Scene.....	35
2.4.3.2 GameObject.....	36
2.4.3.3 Components.....	36
2.4.3.4 Scripts.....	37
2.4.4.5 Prefabs.....	38
CHƯƠNG 3. THIẾT KẾ GAME.....	39
3.1 Mô tả game.....	39

3.1.1 Tổng quan.....	39
3.1.2 Mục tiêu.....	39
3.1.3 Tính năng.....	40
3.1.4 Cách chơi.....	40
3.1.5 Mindset.....	41
3.1.6 Đối tượng người chơi.....	41
3.2 Công nghệ.....	41
3.2.1 Luồng màn hình.....	41
3.2.2 Điều khiển.....	42
3.2.3 Cơ học (game mechanics).....	43
3.3 Vật lý trong game.....	44
3.4 Thiết kế màn chơi.....	44
3.4.1 Các đối tượng.....	44
3.4.2 Chủ đề màn chơi.....	47
3.4.2.1 Các đối tượng.....	47
3.4.2.2 Thử thách.....	47
3.4.3 Luồng chạy game.....	47
3.5 Development.....	47
3.5.1 Thiết kế engine.....	47
3.5.2 Các lớp / đối tượng.....	47
3.6 Visual Design.....	49
3.7 Giao diện người dùng.....	50
3.7.1 Flow Chart.....	50
3.7.2 Thiết kế màn hình.....	51
3.7.2.1 Splash.....	51

3.7.2.2 Main menu.....	52
3.7.2.3 Chọn control.....	53
3.7.2.4 Chọn level.....	54
3.7.2.5 About.....	55
3.7.2.6 UI trong Gameplay.....	56
3.8 Đồ họa.....	57
3.8.1 Mục tiêu tổng quát.....	57
3.8.2 3D Art & Animation.....	57
3.8.3 GUI.....	57
3.8.4 Địa hình.....	57
3.8.5 Hiệu ứng.....	57
3.9 Thiết kế nhạc nền / âm thanh.....	57
3.9.1 Phong cách music / elements.....	58
3.9.2 Phong cách sound / elements.....	58
3.9.3 Game music.....	58
3.9.4 Game sound.....	58
CHƯƠNG 4. HIỆN THỰC GAME.....	59
4.1 Cài đặt environment và camera VR.....	59
4.1.1 Cài đặt SDK và camera VR.....	59
4.1.2 Environment.....	62
4.2 Điều khiển và nhật vũ khí.....	67
4.2.1 Điều khiển.....	67
4.2.2 Nhật vũ khí.....	69
4.3 Các loại enemy.....	72
4.3.1 Enemy.....	72

4.3.2 Animation.....	73
4.3.3 Trạng thái.....	74
4.3.4 Tìm đường đi tới nhân vật.....	74
4.4 Xử lý tấn công bằng súng và gậy.....	78
4.4.1 Tấn công bằng súng.....	78
4.4.2 Tấn công bằng gậy.....	82
4.5 Health HUD (head-up display).....	84
4.5.1 Health HUD, xử lý thanh máu.....	84
4.5.2 Hiệu ứng lúc người chơi mất máu.....	86
4.6 Tính và hiện thị số enemy tiêu diệt được.....	86
4.7 Tạo level 1 và level 2.....	87
4.7.1 Level 1.....	87
4.7.2 Level 2.....	88
4.8 Màn hình menu, chọn level, game over và win.....	89
4.8.1 Liên quan đến UI, có 3 tùy chọn hiển thị canvas.....	89
4.8.2 Màn hình Menu.....	90
4.8.3 Màn hình chọn màn chơi.....	91
4.8.4 Màn hình GameOver và Win.....	91
4.9 Nhạc nền, âm thanh.....	93
4.10 Màn hình chọn control.....	94
4.11 Màn hình loading cho game.....	95
4.12 Tối ưu game.....	96
CHƯƠNG 5. KẾT LUẬN, HƯỚNG PHÁT TRIỂN.....	99
5.1 Kết quả.....	99
5.1.1 Đối với đề tài.....	99

5.1.2 Đối với các thành viên trong nhóm.....	99
5.2 Hạn chế.....	99
5.3 Hướng phát triển.....	99
TÀI LIỆU THAM KHẢO.....	100

DANH MỤC HÌNH VẼ

Hình 2.1 Các thành phần của một hệ thống VR.....
Hình 2.2 DataGloves.....
Hình 2.3 3D Mouse và SpaceBall.....
Hình 2.4 Mouse.....
Hình 2.5 Shutter glasses.....
Hình 2.6 Head-Mounted Displays.....
Hình 2.7 Cave.....
Hình 2.8 CyberTouch.....
Hình 2.9 CyberGrasp.....
Hình 2.10 Binh lính học nhảy dù bằng thực tế ảo.....
Hình 2.11 Cảnh trong một lớp học ảo.....
Hình 2.12 VR giúp các vận động viên luyện tập phản ứng.....
Hình 2.13 Hình ảnh 1 chiếc Google Cardboard sau khi đã lắp ráp.....
Hình 2.14 Hình chưa lắp ráp và hình sau khi lắp ráp.....
Hình 2.15 Giao diện mặc định của Unity.....
Hình 2.16 Cửa sổ project Unity.....
Hình 2.17 Tab Hierarchy.....
Hình 2.18 Toolbar của Unity.....
Hình 2.19 Tab Scene View.....
Hình 2.20 Nhấp chuột và kéo thả để di chuyển góc nhìn.....
Hình 2.21 Giữ Alt và kéo thả chuột để xoay góc nhìn quanh điểm trục.....
Hình 2.22 Giữ Ctrl và kéo thả chuột để phóng to thu nhỏ góc nhìn.....
Hình 2.23 Tab Game View.....
Hình 2.24 Component của gameobject.....
Hình 2.25 Tab Inspector.....

Hình 2.26 Asset Labels.....	
Hình 2.27 Các Scenes Unity.....	
Hình 2.28 Các Game Object.....	
Hình 2.29 Các Components chứa trong một GameObject Camera.....	
Hình 2.30 Script C# trong Untiy.....	
Hình 2.31 Prefab trong Unity.....	
Hình 3.1 Tay cầm gamepad	
Hình 3.2 Nút bấm trên google cardboard.....	
Hình 3.3 Zombie bunny.....	
Hình 3.4 Zombie bear.....	
Hình 3.5 Zombie hellephant.....	
Hình 3.6 Súng.....	
Hình 3.7 Bình máu.....	
Hình 3.8 Sơ đồ màn hình.....	
Hình 3.9 Màn hình splash.....	
Hình 3.10 Màn hình menu.....	
Hình 3.11 Màn hình chọn control.....	
Hình 3.12 Màn hình chọn level.....	
Hình 3.13 Màn hình about.....	
Hình 3.14 Màn hình trong gameplay.....	
Hình 4.1 Màn hình import sdk.....	
Hình 4.2 Inspector của GvrViewerMain	
Hình 4.3 Giao diện camera trên VR.....	
Hình 4.4 Trang model environment trên asset store.....	
Hình 4.5 Bản đồ trong game.....	

Hình 4.6 Súng.....	
Hình 4.7 Thanh gỗ.....	
Hình 4.8 Bàn, sách vở và nhiều vật nhỏ khác.....	
Hình 4.9 Xuôn cứu hộ	
Hình 4.10 Đèn.....	
Hình 4.11 Cửa chính và cửa sổ.....	
Hình 4.12 Đường ống thoát nước.....	
Hình 4.13 Input manager.....	
Hình 4.14 Code xử lý tự động di chuyển.....	
Hình 4.15 Code xử lý nhặt vật.....	
Hình 4.16 Súng.....	
Hình 4.17 Khi nhặt súng lên.....	
Hình 4.18 Khi nhặt gậy lên.....	
Hình 4.19 Zombie bunny.....	
Hình 4.20 Zombie bear.....	
Hình 4.21 Zombie hellephant.....	
Hình 4.22 Animator của enemy.....	
Hình 4.23 Giao diện navigation tab object.....	
Hình 4.24 Giao diện navigation tab bake.....	
Hình 4.25 Giao diện trong map sau khi bake xong.....	
Hình 4.26 Nav mesh agent.....	
Hình 4.27 Code xử lý di chuyển của enemy.....	
Hình 4.28 NavMesh trong game.....	
Hình 4.29 Code xử lý tấn công.....	
Hình 4.30 Gắn collider cho object.....	

Hình 4.31 Code xử lý bắn.....	
Hình 4.32 Hiệu ứng khói bụi trên enemy.....	
Hình 4.33 Hiệu ứng phát sáng khi bắn.....	
Hình 4.34 Animation.....	
Hình 4.35 Trang tạo animation.....	
Hình 4.36 Hình chọn animation cần làm.....	
Hình 4.37 Animation gây.....	
Hình 4.38 Health HUD.....	
Hình 4.39 Code xử lý khi người chơi bị tấn công.....	
Hình 4.40 Hiệu ứng khi người chơi bị tấn công.....	
Hình 4.41 Số enemy bị tiêu diệt hiển thị trên màn hình.....	
Hình 4.42 Người chơi cầm súng.....	
Hình 4.43 Bản đồ level 1.....	
Hình 4.44 Người chơi cầm gây.....	
Hình 4.45 Bản đồ level 2.....	
Hình 4.46 Inspector của canvas.....	
Hình 4.47 Model nhà tại menu.....	
Hình 4.48 Canvas menu.....	
Hình 4.49 Canvas chọn màn chơi.....	
Hình 4.50 Canvas win.....	
Hình 4.51 Canvas game over.....	
Hình 4.52 Music manager.....	
Hình 4.53 Màn hình chọn control.....	
Hình 4.54 Màn hình loading.....	
Hình 4.55 Quality setting.....	

Hình 4.56 Giảm bớt model trong map.....	
Hình 4.57 Occulision culling.....	

DANH MỤC TỪ VIẾT TẮT

TỪ VIẾT TẮT	TỪ ĐẦY ĐỦ	ĐỊNH NGHĨA TIẾNG VIỆT
VR	Virtual Reality	Thực tế ảo
SDK	Software Development Kit	Bộ phát triển phần mềm
NDK	Native Development Kit	Bộ phát triển Native
2D	Two Dimensional	2 chiều
3D	Three Dimensional	3 chiều
API	Application Programming Interface	Giao diện lập trình ứng dụng
XML	Extensible Markup Language	Ngôn ngữ đánh dấu Mở rộng
BMP	Bitmap	Bitmap

TÓM TẮT KHÓA LUẬN

Survival shooter VR là một game thực tế ảo. Game là thể loại bắn súng sinh tồn, người chơi sẽ được trải nghiệm bối cảnh chiến đấu trong ngôi nhà bị bỏ hoang và cảm giác nhập vai vào nhân vật.

Người chơi sẽ điều khiển nhân vật của mình di chuyển theo hướng của đầu (head) người chơi, cũng như nhặt súng và các vật dụng để đổi 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

Game có góc nhìn thực tế ảo mới lạ và thú vị. Hỗ trợ tay cầm giúp người chơi điều khiển dễ dàng hơn. Nhiều loại enemy cũng như súng và vật dụng. Âm thanh kịch tính, sống động.

Game được xây dựng và phát triển trên nền tảng Unity 3D kết hợp với Google VR SDK.

CHƯƠNG 1. MỞ ĐẦU

1.1 Giới thiệu đề tài

Survival shooter VR là một game thực tế ảo. Game là thể loại bắn súng sinh tồn, người chơi sẽ được trải nghiệm bối cảnh chiến đấu trong ngôi nhà bị bỏ hoang và cảm giác nhập vai vào nhân vật.

1.2 Lý do chọn đề tài

Sau hơn nửa thế kỷ phát triển, công nghệ thực tế ảo đã trở thành một ngành công nghiệp thực sự hái ra tiền tại các nước phát triển với nhiều ứng dụng trong đủ các lĩnh vực từ giải trí, du lịch, địa ốc, y học, kiến trúc, giáo dục cho đến cả khoa học kỹ thuật và quân sự.

Theo dự đoán của Gartner (tổ chức nghiên cứu thị trường toàn cầu) thì VR sẽ là một trong số 10 công nghệ chiến lược trong vòng 1 thập niên tới.

Tuy nhiên, đối với các thị trường đang phát triển, số lượng các nhà phát triển đi chuyên về mảng này còn rất ít vì sự mới mẻ của loại hình công nghệ này cũng như đầu ra của thị trường còn hạn chế. Giới lập trình tại các nước này vẫn chưa nhìn thấy tiềm năng vô hạn của thực tế ảo để thực sự đầu tư vào sáng tạo.

Vấn đề chung của các ứng dụng trên VR hiện nay là rất nhiều trong số chúng đang chạy theo phong trào, không phải thiết kế theo đúng tinh thần thực tế ảo mà chỉ là các game thông thường hỗ trợ thêm trải nghiệm kiểu thực tế ảo nên không thể giữ chân người dùng lâu dài.

Từ những điều trên, nhận thấy việc xây dựng game thực tế ảo ... với mong muốn đóng góp cho nền công nghiệp game cũng như ứng dụng các kiến thức đã học, nhóm đã quyết định thực hiện đề tài này.

1.3 Mục tiêu đề tài

Xây dựng game đa nền tảng có ứng dụng công nghệ mới dưới sự hỗ trợ của Unity và Google VR SDK để tạo nên một game VR (virtual reality) hoàn chỉnh.

1.4 Ý nghĩa đề tài

1.4.1 Đối với người chơi nói riêng và ngành công nghiệp game nói chung

Tạo ra một game mới giúp làm phong phú kho game hiện tại, giúp người chơi tiếp cận được những thứ mới mẻ, trải nghiệm được công nghệ thực tế ảo.

1.4.2 Đối với nhóm

- Giúp trau dồi đồng thời nâng cao kiến thức về lập trình.
- Ứng dụng những gì đã học được vào quá trình làm game.
- Có kinh nghiệm khi làm việc với Unity 3D và Google VR SDK.
- Học cách làm việc nhóm, quản lý thời gian, sử dụng các công cụ quản lý project và làm việc nhóm.

1.5 Nội dung thực hiện

Báo cáo được xây dựng trong 5 chương.

Chương 1: Mở đầu

Chương 2: Cơ sở lý thuyết

Trình bày tổng quan về thực tế ảo, các đặc tính của VR. Tổng quan về Google cardboard, các thư viện hỗ trợ và nền tảng phát triển trong đề tài.

Chương 3: Thiết kế Game

Mô tả về đề tài, những công nghệ sử dụng trong project, các yếu tố trong Game và trình bày về thiết kế màn chơi, giao diện người dùng, đồ họa, âm thanh, ...

Chương 4: Hiện thực Game

Hiện thực hóa ý tưởng GameDesign ở chương 3 với nhân vật, màn chơi cụ thể. Cài đặt SDK trong game.

Chương 5: Kết luận, hướng phát triển

Tổng kết kết quả đạt được, chưa đạt được, những tính năng cần phát triển trong tương lai.

Tài liệu tham khảo

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 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.

2.1.1 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 camera 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 lái 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 Sutherland 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.

2.1.2 Các đặc tính chính của VR

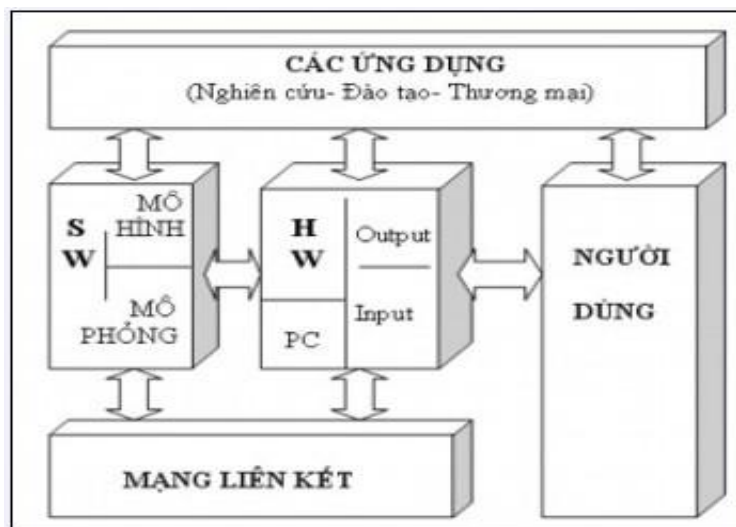
Như trên đã trình bày, 2 đặc tính chính của VR là Tương tác và Đắm chìm, đây là hai "I" (Interactive, Immersion) mà nhiều người đã biết. Tuy nhiên VR cần có 1 đặc tính thứ 3 mà ít người để ý tới.

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)

2.1.3 Các thành phần một hệ thống VR

Tổng quát một VR bao gồm những thành phần sau:



Hình 2.1 Các thành phần của một hệ thống VR

2.1.3.1 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,...

2.1.3.2 Phần mềm (Software)

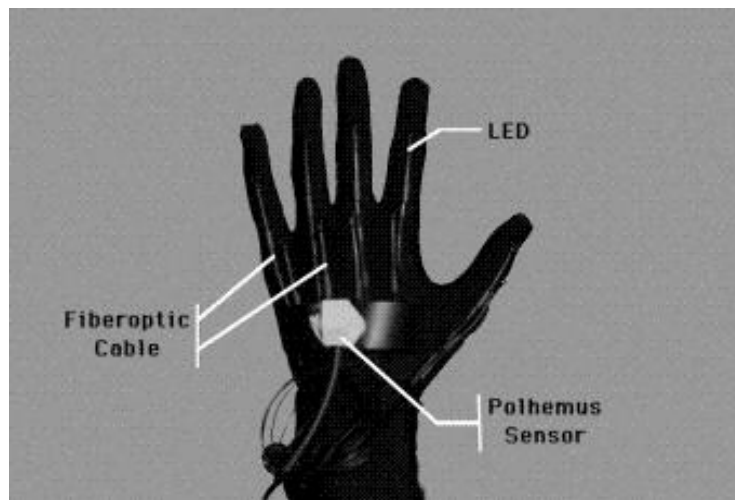
Phần mềm luôn là linh hồn của VR cũng như đối với bất cứ một hệ thống máy tính hiện đại nào. Về mặt nguyên tắc có thể dùng bất cứ ngôn ngữ lập trình hay phần mềm đồ họa nào để mô hình hóa (modelling) và mô phỏng (simulation) các đối tượng của VR.

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.

2.1.1 Các thiết bị cơ bản

2.1.4.1 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.



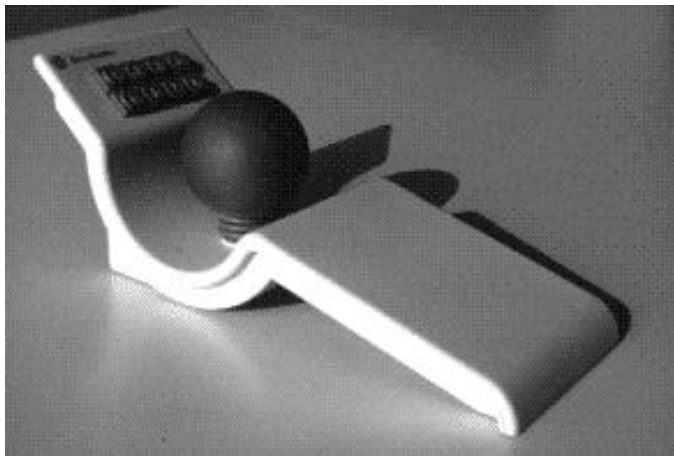
Hình 2.2 DataGloves

3D Mouse and SpaceBall



Hình 2.3 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 2.4 Mouse

Shutter glasses



Hình 2.5 Shutter glasses

Head-Mounted Displays



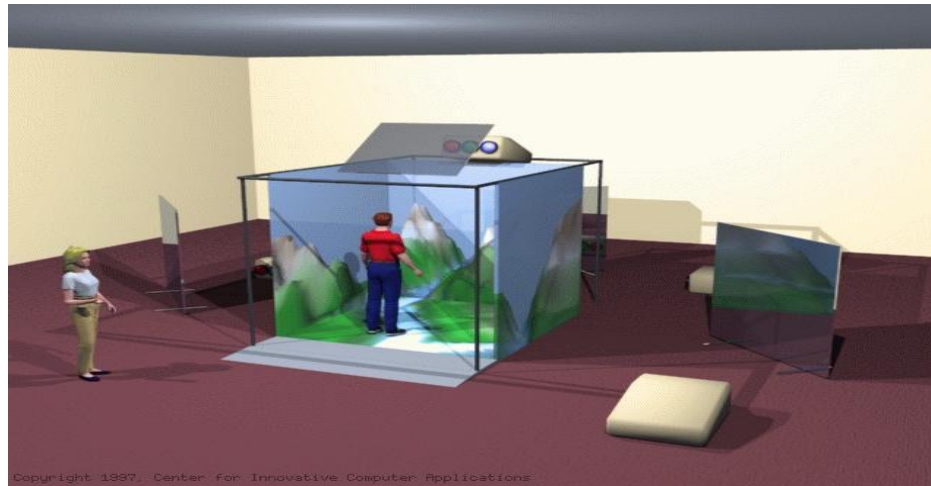
Hình 2.6 Head-Mounted Displays

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 những màn ảnh khác bằng các tấm gương phản chiếu. Người dùng sẽ đi vào bên trong CAVE và đeo 1 chiếc kính 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 là 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 tử. 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 2.7 Cave

2.1.4.2 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 2.8 CyberTouch

CyberGrasp



Hình 2.9 CyberGrasp

2.1.5 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.

2.1.5.1 Quân sự

Với việc phát triển của VR, các binh sĩ sẽ được huấn luyện 1 cách trực quan nhất các kỹ năng cần thiết như : lái máy bay, lái 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 khóa 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 tính 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 2.10 Binh lính học nhảy dù bằng thực tế ảo

2.1.5.2 Giáo dục



Hình 2.11 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 những thành viên khác trong lớp.

2.1.5.3 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 tính, bạn có thể đi lại khắp nơi trong nhà, xem xét từng ngõ ngách nhỏ nhất.

2.1.5.4 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.



Hình 2.12 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.

2.1.5.5 Du hành không gian

Các nhà khoa học của NASA đã thực hiện nhiều cuộc thám hiểm 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.

2.1.5.6 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 hiệu quả không cao mà chi phí tốn kém.

2.2 Giới thiệu Google Cardboard

2.2.1 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". 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.

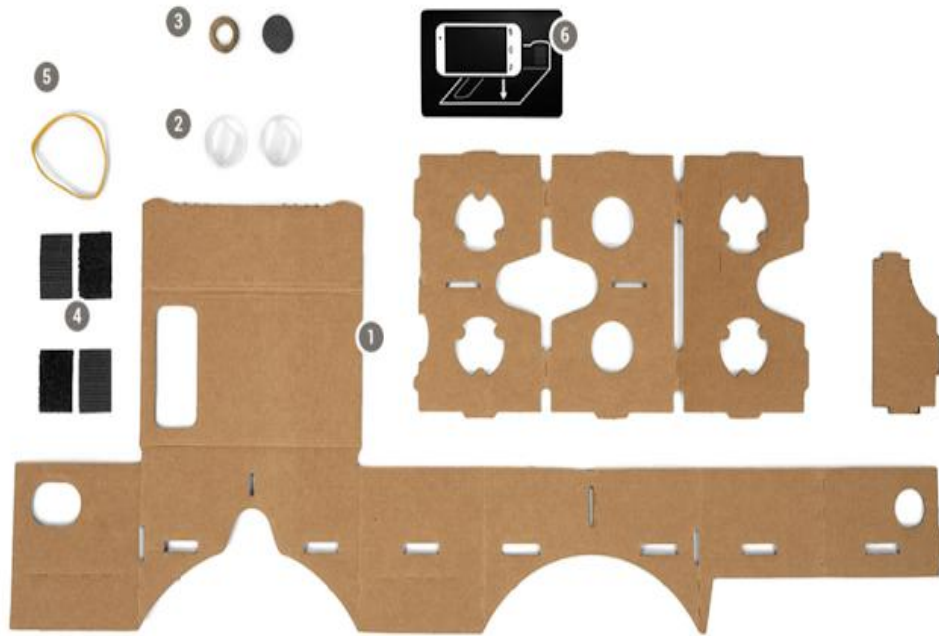


Hình 2.13 Hình ảnh 1 chiếc Google Cardboard sau khi đã lắp ráp

Cách lắp ráp 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 2.14: 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 đeo vào đầu, mấy miếng đệm cao su, và một tag giao tiếp NFC tùy 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 quả 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 từ 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 kiện đó.

2.2.2 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.

2.2.3 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.

2.3 Google VR SDK

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.

Cardboard SDK là bộ công cụ dành cho lập trình viên để họ có thể phát triển ứng dụng di động 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
- + Google VR SDK for Unity
- + Google VR SDK for Unreal
- + Google VR SDK for iOS
- + VR View

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

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.

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.

2.4 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.

2.4.1 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#. 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 độc quyền có cú pháp tương tự JavaScript. Phần mềm nhắm mục tiêu các đồ họa APIs sau: Direct3D trên Windows và Xbox One; 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. 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 đặc điểm kỹ 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, Android TV, Facebook Gameroom, Fire OS, Gear VR, Google Cardboard, Google Daydream, HTC Vive, iOS, Linux, macOS, Microsoft HoloLens, Nintendo 3DS family, Nintendo Switch, Oculus Rift, PlayStation 4, PlayStation Vita, PlayStation VR, Samsung Smart TV, Tizen, tvOS, WebGL, Wii U, Windows, Windows Phone, Windows Store, và 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 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 trong khi 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, occlusion 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.

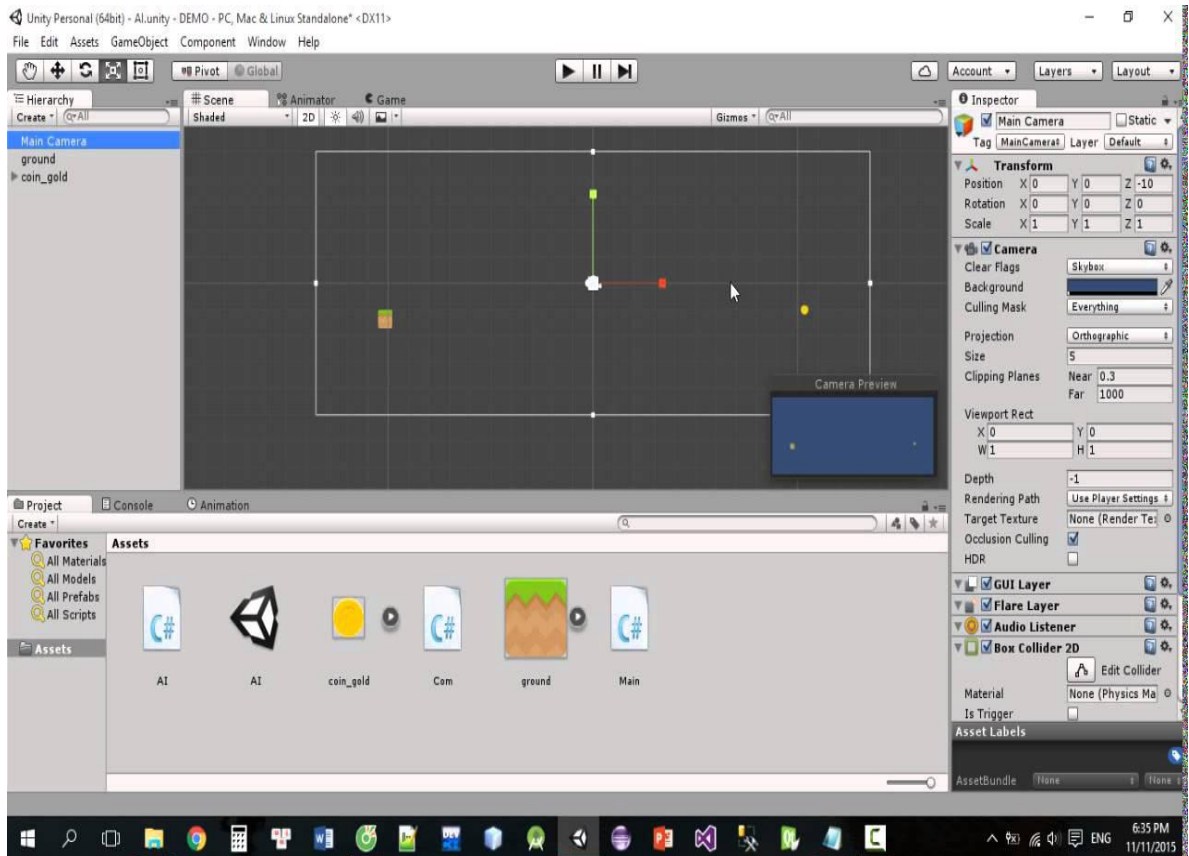
Cả Unity và Unity Pro đều bao gồm môi trường phát triển, hướng dẫn, các dự án mẫu và nội dung, hỗ trợ thông qua diễn đàn, và các cập nhật tương lai trong cùng một phiên bản chính (tức là mua Unity 3.0 được các cập nhật tất cả Unity Pro 3.x trong tương lai một cách miễn phí).

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.

2.4.2 Giới thiệu về giao diện



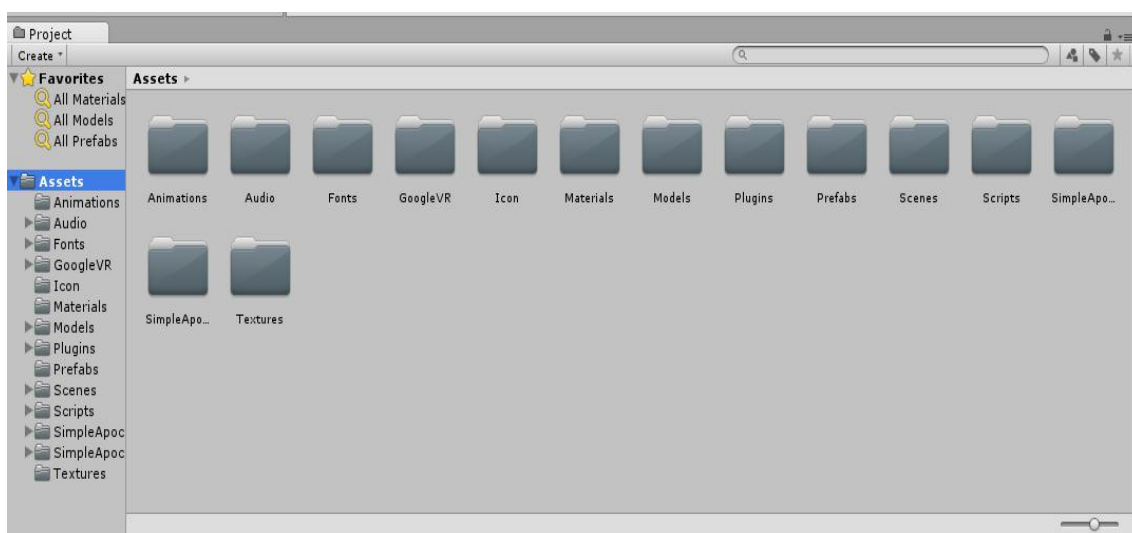
Hình 2.15 Giao diện mặc định của Unity

2.4.2.1 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 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.

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



Hình 2.16 Cửa sổ project Unity

Phân 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 phân 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 cỡ bằng cách sử dụng thanh trượt nằm ở góc dưới bên phải. Nếu con trượt chạy về phía bên trái thì các biểu tượng 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.

2.4.2.2 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 2.17 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 được thêm hay xóa trong scene, nó sẽ xuất hiện hoặc biến mất trong

thẻ Hierarchy.

2.4.2.3 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

2.4.2.4 Toolbar

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

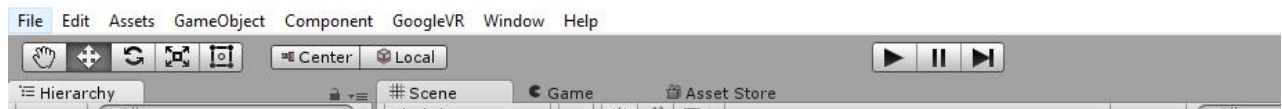
A: Transform Tool: được dùng với Scene view, như quay trái , phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.

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

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

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

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



Hình 2.18 Toolbar của Unity

2.4.2.5 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 sử dụng các công cụ Transform Tools ở trong Toolbar để di chuyển, xoay, phóng to thu 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.



Hình 2.19 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.



Hình 2.20 Nhấp chuột và kéo thả để di chuyển góc nhìn.



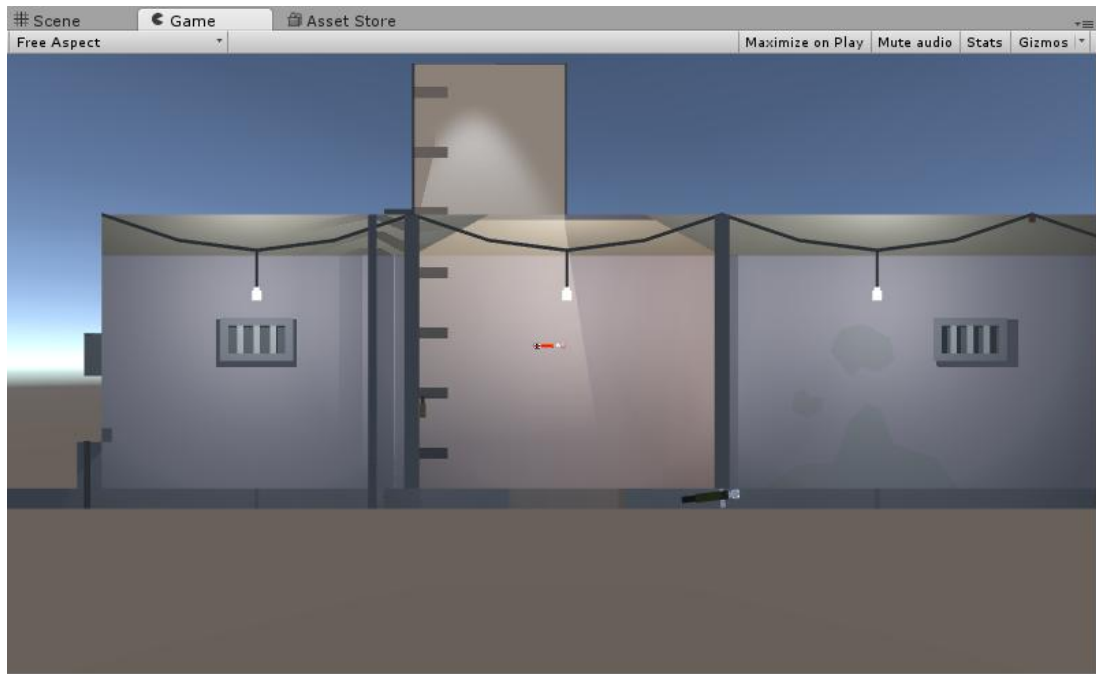
Hình 2.21 Giữ Alt và kéo thả chuột để xoay góc nhìn quanh điểm trục.



Hình 2.22 Giữ Ctrl và kéo thả chuột để phóng to thu nhỏ góc nhìn.

2.4.2.6 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 quét định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game.



Hình 2.23 Tab Game View

2.4.2.7 Play Mode

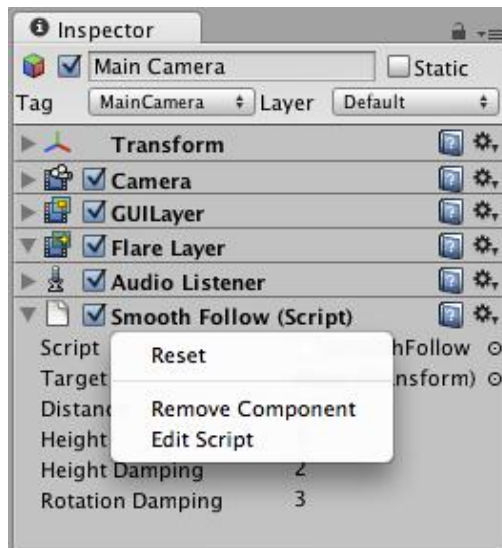
Sử dụng những nút trên Toolbar để điều khiển Editor Play Mode, và xem trước Game sẽ như thế nào khi chơi. Trong chế Play, mọi giá trị thay đổi sẽ được lưu tạm, và bị xóa khi thoát khỏi chế độ play.

2.4.2.8 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.

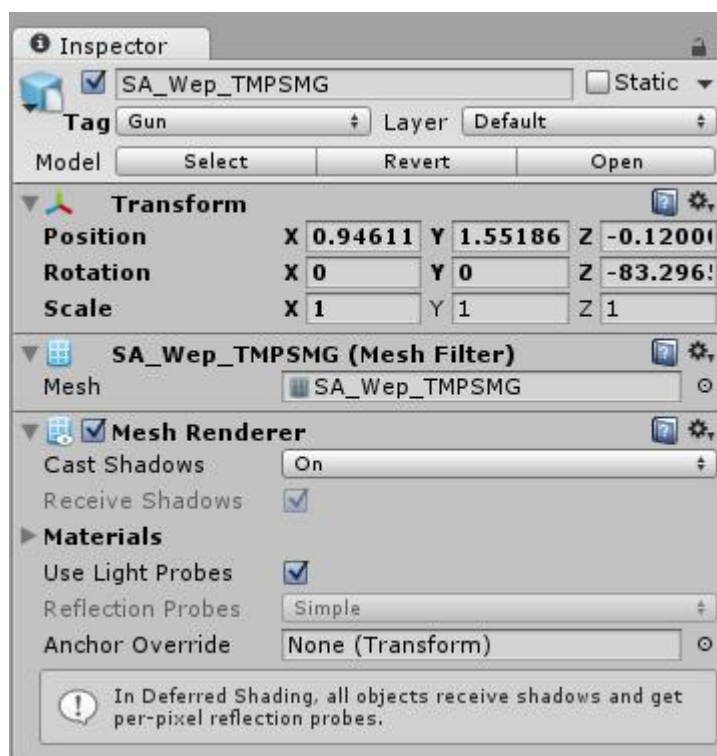
Mọi thuộc tính thể hiện trong Inspector đều có thể được tùy biến một cách trực tiếp. Ngay cả với những biến trong script cũng có thể được hiệu chỉnh mà không cần xem mã. Trong script, nếu định nghĩa một giá trị là public cho một kiểu đối tượng (như GameObject hay Transform), thì có thể drag-drop một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



Hình 2.24 Component của gameobject

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



Hình 2.25 Tab Inspector

Sử dụng thanh số Layer để sắp xếp các lớp sẽ được hiển thị..Thanh số 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.



2.26 Asset Labels

2.4.2.9 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.

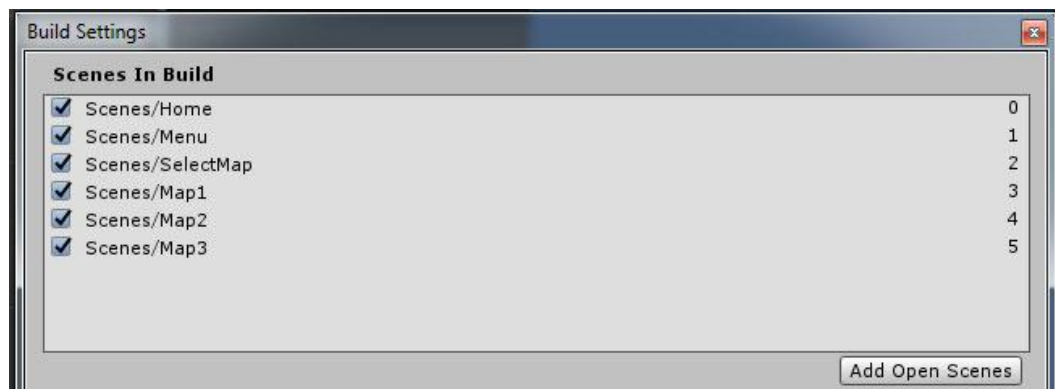
Chúng ta cũng có thể lưu lại những thiết lập về cấu trúc của các Views, bằng cách click vào menu Dropdown Layout ở thanh Toolbar, chọn Save Layout... Đặt tên cho nó và sau đó bạn dễ dàng gọi nó ra bằng cách chọn nó trong Dropdown Menu Layout.

2.4.3 Một số khái niệm cơ bản trong Unity

2.4.3.1 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 2.27 Các Scenes Unity

2.4.3.2 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(vị 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 2.28 Các Game Object

2.4.3.3 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 2.29 Các Components chứa trong một GameObject camera

2.4.3.4 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#.

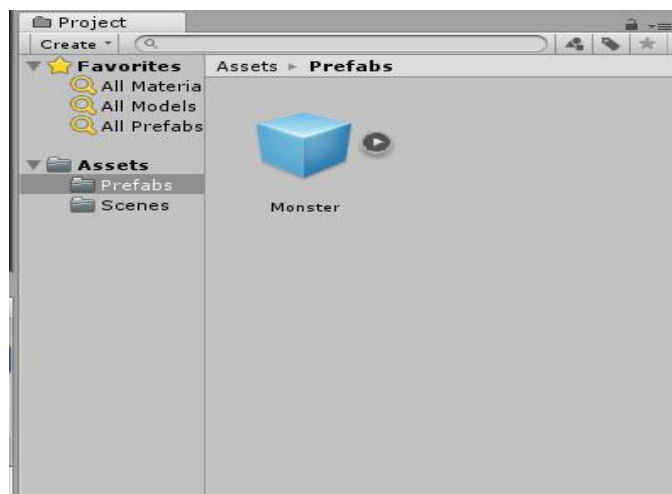
Để viết Script, chúng ta có thể biên tập Script trên Mono Developer được tích hợp mặc định vào Unity hoặc bạn có thể cài một trình biên tập riêng như Visual Studio, Visual Code.



Hình 2.30 Script C# trong Unity

2.4.4.5 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 một 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 2.31 Prefab trong Unity

CHƯƠNG 3. THIẾT KẾ GAME

3.1 Mô tả game

3.1.1 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 (head) người chơi, cũng như nhặt súng và các vật dụng để đổi 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.

3.1.2 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.

3.1.3 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.

Nhiều loại enemy cũng như súng, gậy và vật phẩm.

Âm thanh kịch tính, sống động.

3.1.4 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.

Người chơi sẽ đối đầu với enemy được sinh ra từ các hướng 1 cách ngẫu nhiên, người chơi có thể nhặt được súng, gậy dưới mặt 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.

Sử dụng tay cầm gamepad ở chế độ gamepad hoặc auto nếu ở chế độ tự động để điều khiển nhân vật di chuyển theo hướng của đầu (head) người chơi trong bản đồ 3D. Ở chế độ gamepad người chơi có thể sử dụng điều khiển bằng nút bấm để qua trái, phải, tới, lui, tăng tốc, nhặt đồ vật và tấn công. Còn ở chế độ bình thường thì sẽ sử dụng nút của cardboard để nhặt đồ vật và tấn công.

Các enemy sẽ được sinh ra trong map ở các vị trí khác nhau được định sẵn, tùy theo loại map sẽ có các loại enemy khác nhau, các enemy này sẽ tự động tìm đường đi đến player và tìm cách tiêu diệt player.

Khi tiêu diệt được enemy sẽ ngẫu nhiên sinh ra bình máu cho người chơi. Game sẽ kết thúc nếu như người chơi tiêu diệt được đủ số lượng enemy trên bản đồ mà màn chơi yêu cầu hoặc người chơi bị enemy tiêu diệt. Nếu toàn bộ enemy bị tiêu diệt thì người chơi sẽ được đi tiếp qua màn sau, và ngược lại nếu người chơi bị enemy tiêu diệt thì người chơi sẽ phải chơi lại để có thể đi đến màn tiếp theo.

3.1.5 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 theo level, người chơi sẽ thích ứng dễ dàng hơn.

3.1.6 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

3.2 Công nghệ

3.2.1 Luồng màn hình

- Splash Screen
 - Main Menu
 - Play Game
 - Level Select
 - Level
 - Level Complete
 - Next Level
 - Main menu
 - About
 - Control
 - Gamepad

- Auto

3.2.2 Điều khiển

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



Hình 3.1: Tay cầm gamepad

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

Đối với nút của google cardboard:



Hình 3.2: Nút bấm trên google cardboard

Sử dụng nút của google cardboard để điều khiển nhặt đồ vật và tấn công.

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

3.2.3 Cơ học (game mechanics)

Nhận vật, thành phần trong game:

- Enemy khi tấn công vào người chơi sẽ bị trừ đi 5 máu.
- Người chơi có 100 máu, khi dùng súng bắn vào trúng enemy thì enemy bị mất 15 máu, còn khi dùng gậy đánh trúng enemy thì enemy bị mất 20 máu đến khi enemy hết máu thì enemy sẽ bị tiêu diệt.
- Các công trình, vật cản sẽ không bị phá hủy.
- Enemy và player sẽ không có khả năng đi xuyên qua các công trình, vật cản.
- Enemy khi hết máu sẽ chết, biến mất và ngẫu nhiên sinh ra bình máu có khả năng tăng 15 máu cho người chơi.
- Player khi chết thì game sẽ được coi là thua và kết thúc game.
- Khi bắn hoặc dùng gậy đập trúng enemy sẽ tạo effect khói bụi trên enemy.

3.3 Vật lý trong game

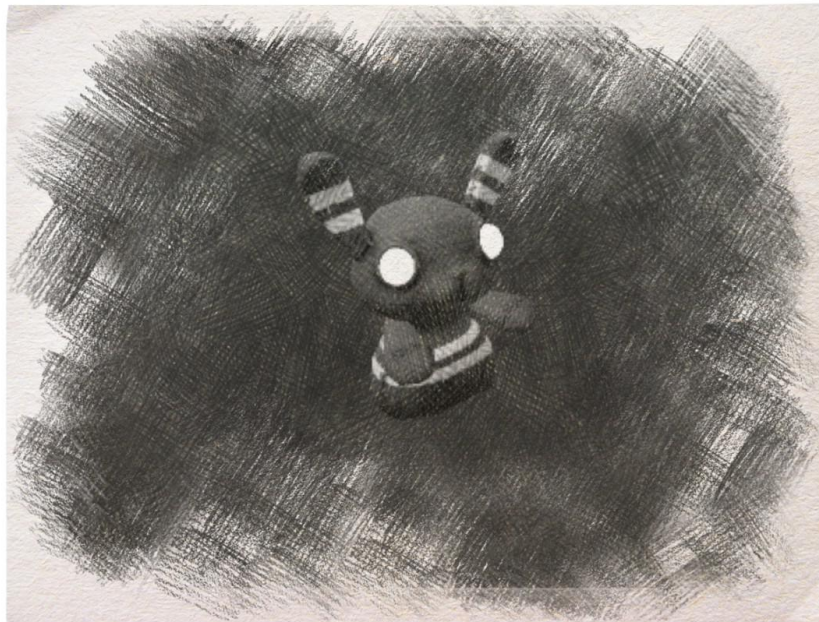
Bản đồ 3D là nơi nhân vật có thể di chuyển trên đó, bản đồ có các căn phòng và bức tường được gắn collider và physic để giới hạn các đường biên xung quanh, ngăn cho nhân vật cũng như các enemy không thể di chuyển qua được.

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 enemy 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.

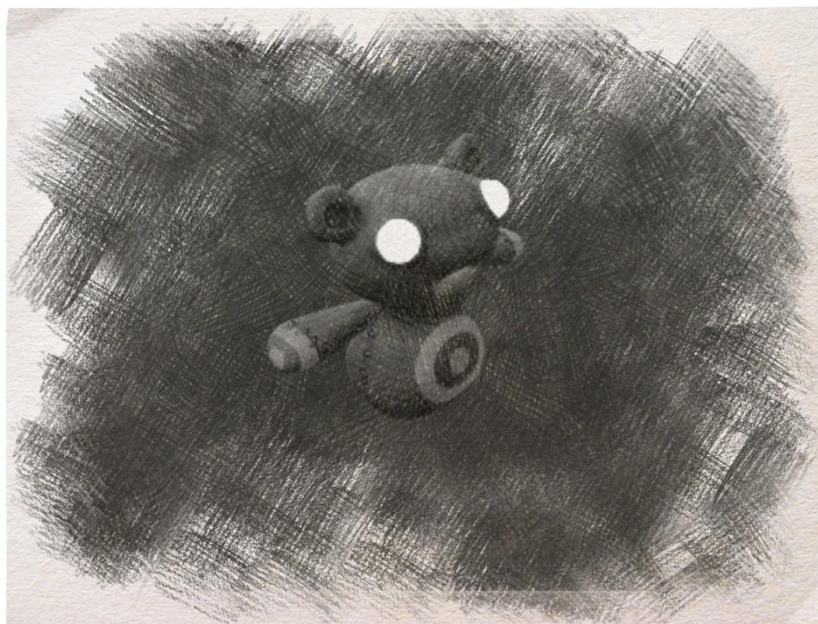
3.4 Thiết kế màn chơi

3.4.1 Các đối tượng

- Static level pieces
 - Môi trường xung quanh.
- Interactive/dynamic level pieces
 - Enemy



Hình 3.3: Zombie bunny



Hình 3.4: Zombie bear



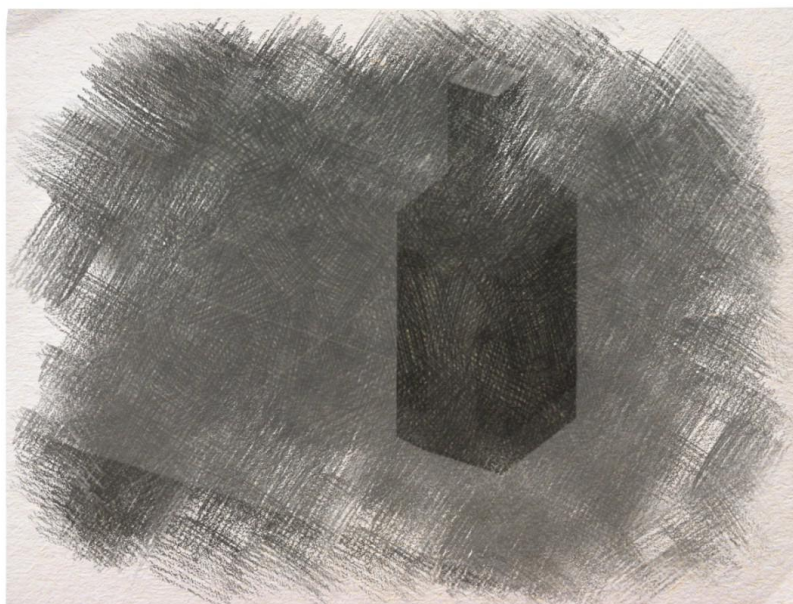
Hình 3.5: Zombie hellephant

- Nhân vật
- Súng và gậy



Hình 3.6: Súng

- Vật phẩm tăng máu



Hình 3.7: Bình máu

3.4.2 Chủ đề màn chơi

Ống cống thoát nước

3.4.2.1 Các đối tượng

Ambience

- Nhà cửa.
- Ống thoát nước.
- Các vật nhỏ khác.

Interactive

- Súng.
- Gậy.
- Bình máu.

3.4.2.2 Thử thách

Có các loại enemy xanh và hồng và boss vàng, số lượng tăng dần theo thời gian.

3.4.3 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 hoặc nút của cardboard.
- Tiêu diệt các enemy xuất hiện từ mọi hướng.
- Khi tiêu diệt được đủ số lượng enemy thì sẽ được qua màn tiếp theo.

3.5 Development

3.5.1 Thiết kế engine

- Sử dụng sự hỗ trợ của engine Unity3D.
- Google VR SDK giúp hỗ trợ tạo ra game thực tế ảo.

3.5.2 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 hiệu ứng.
- Music Controller:
 - Xử lý các sự kiện liên quan tới âm thanh nhạc nền.
- Player Controller
 - Xử lý di chuyển theo các hướng bằng tay cầm và nút của cardboard.
- 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.

3.6 Visual Design

3.6.1 Color pallete

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

3.6.2 Các đối tượng trong game

- Các đối tượng cơ bản cụ thể
 - Player
 - Enemy
 - Súng
 - Gậy
 - Bình máu
- Bản đồ
 - Môi trường xung quang tùy vào map như ống cống thoát nước,...
- Các loại enemy
 - Enemy Base Class
 - Zombie đánh gần
 - Zombie hồng
 - Zombie xanh
 - Zombie vàng

3.6.3 Game scenery

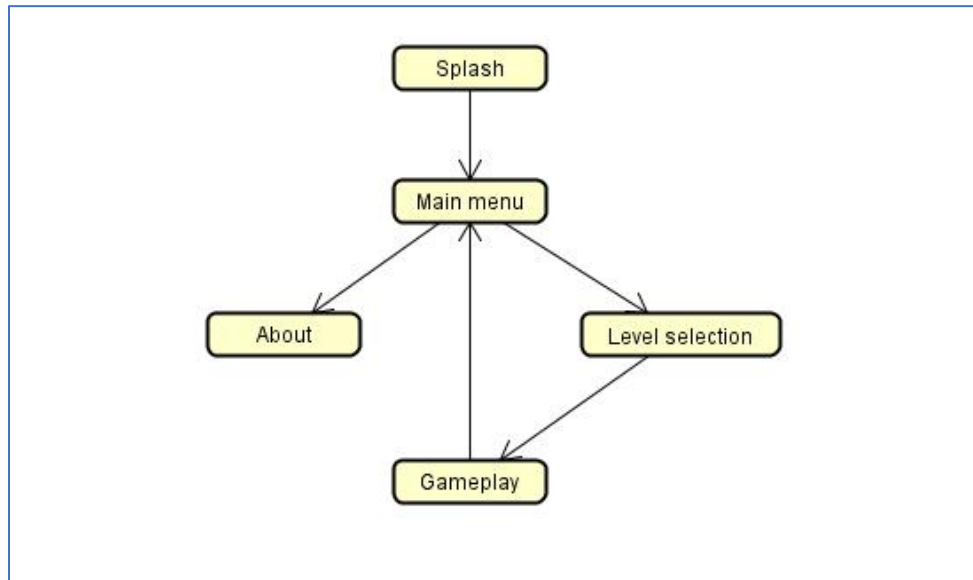
Asset dùng để thiết kế map cho game lấy từ nguồn trên mạng, có nhiều model theo nhiều chủ đề phù hợp với game zombie.

3.6.4 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,...

3.7 Giao diện người dùng

3.7.1 Flow Chart



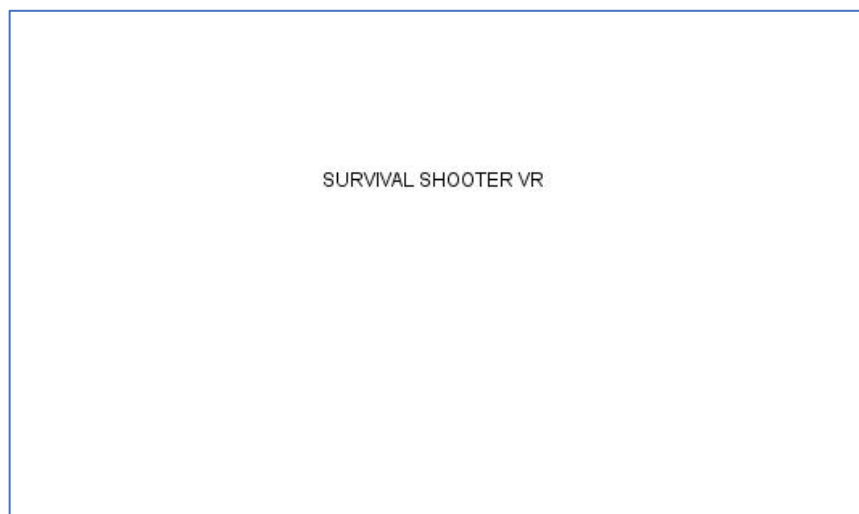
Hình 3.8: Sơ đồ màn hình

STT	Tên màn hình	Chức năng
1	Splash	Màn hình mở đầu của game dùng để 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 chọn màn chơi, 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	Control	Cho phép chọn 1 trong 2 cách điều khiển
6	About	Hiển thị thông tin của nhóm và những thông tin khác.
7	Game Play	Nơi game diễn ra và sự tương tác 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.

3.7.2 Thiết kế màn hình

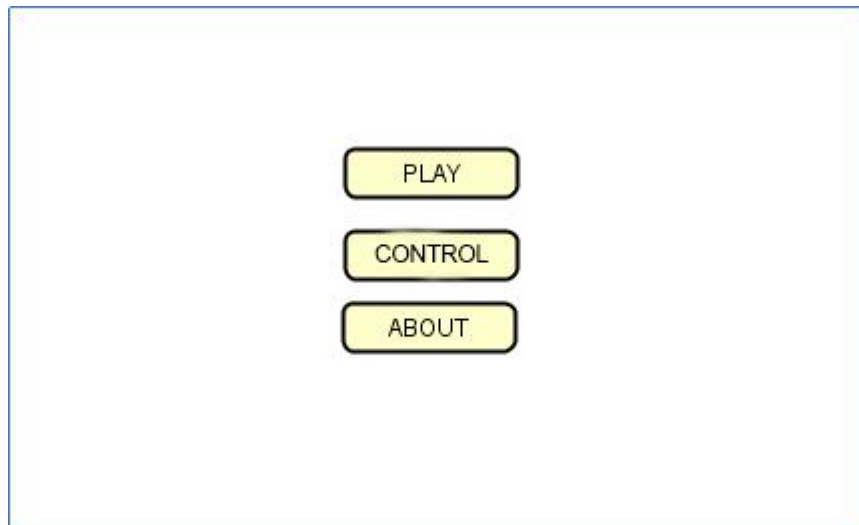
3.7.2.1 Splash



Hình 3.9: Màn hình splash

STT	Tên	Kiểu	Chức năng
1	splash	Image	Hiển thị splash của game

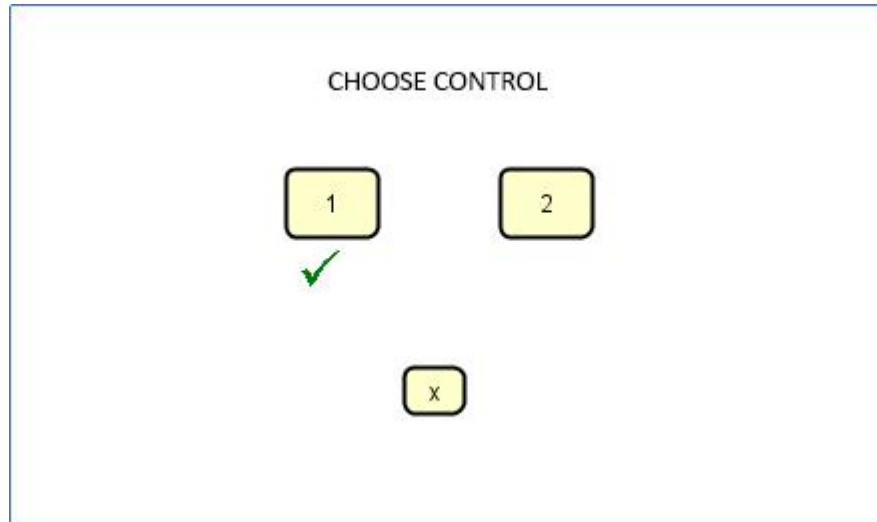
3.7.2.2 Main menu



Hình 3.10: Màn hình menu

STT	Tên	Kiểu	Chức năng
1	btnPlay	Button	Vào màn hình chọn màn chơi
2	btnAbout	Button	Vào màn hình about
3	btnControl	Button	Vào màn hình chọn control

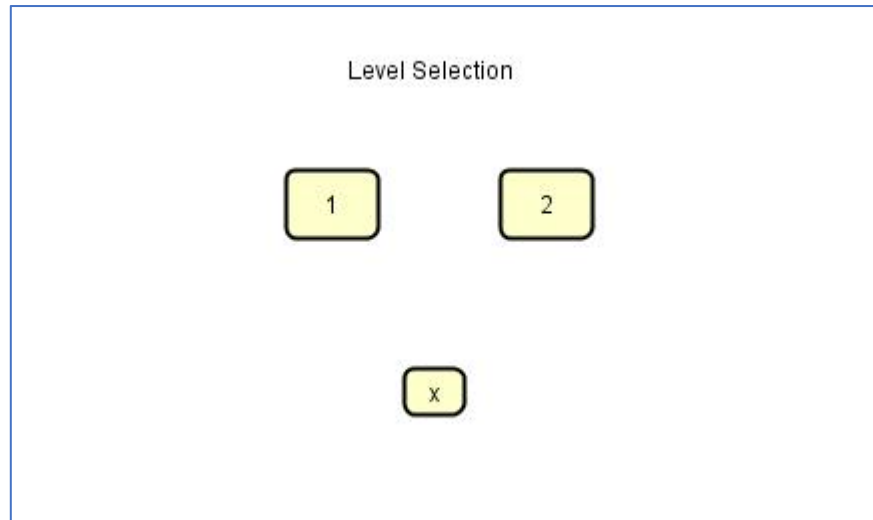
3.7.2.3 Chọn control



Hình 3.11: Màn hình chọn control

STT	Tên	Kiểu	Chức năng
1	txtChooseControl	Text	Hiển thị choose control
2	btnAuto	Button	Chọn auto
3	btnGamePad	Button	Chọn gamepad
4	btnClose	Button	Trở về menu
5	imgTick	Image	Hiện dấu tick cho control đang được chọn

3.7.2.4 Chọn level



Hình 3.12: Màn hình chọn level

STT	Tên	Kiểu	Chức năng
1	txtLevelSelect	Text	Hiển thị level select
2	btnLv1	Button	Vào level 1
3	btnLv2	Button	Vào level 2
4	btnClose	Button	Trở về menu

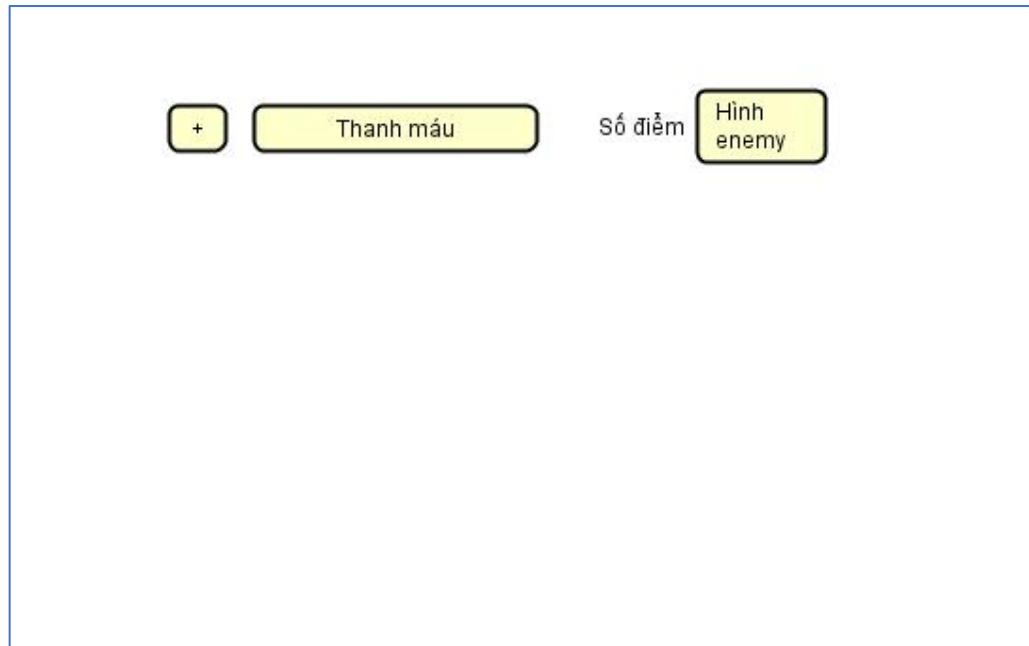
3.7.2.5 About



Hình 3.13: Màn hình about

STT	Tên	Kiểu	Chức năng
1	txtAbout	Text	Hiện about
2	txt1	Text	Tên thành viên 1
3	txt2	Text	Tên thành viên 2

3.7.2.6 UI trong Gameplay



Hình 3.14: Màn hình trong gameplay

STT	Tên	Kiểu	Chức năng
1	splashImg	Image	Hiển thị splash của game
2	healthSlider	Slider	Thanh máu
3	txtScore	Text	Hiển thị số enemy diệt được
4	enemyImg	Image	Hình đại diện enemy

3.8 Đồ họa

3.8.1 Mục tiêu tổng quát

Thể hiện được khung cảnh của một nhà trong công thoát nước, với những căn phòng, ...

Màu sắc tông màu tối, nhiều không gian góc cạnh, phù hợp với style low poly của game. Vì vậy nên thiết kế enemy có ánh sáng tự phát ra.

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

3.8.2 3D Art & Animation

- 3D Art:

+ Các model nhà cửa.

+ Model súng và các vật dụng.

- Animation:

+ Các enemy.

+ Súng và gậy.

3.8.3 GUI

- Thiết kế giao diện người dùng:

+ Giao diện đơn giản với màu sắc tối phù hợp với game thể loại zombie.

3.8.4 Địa hình

- Map trong game là trong ống cống thoát nước.

- Địa hình bằng phẳng.

- Trong căn nhà nhà nhiều phòng ngăn biệt.

3.8.5 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,...

3.9 Thiết kế nhạc nền / âm thanh

3.9.1 Phong cách music / elements

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

3.9.2 Phong cách sound / elements

- Âm thanh kịch tính,...

3.9.3 Game music

- Nhặt nền kịch tính.

3.9.4 Game sound

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

CHƯƠNG 4. HIỆN THỰC GAME

4.1 Cài đặt environment và camera VR

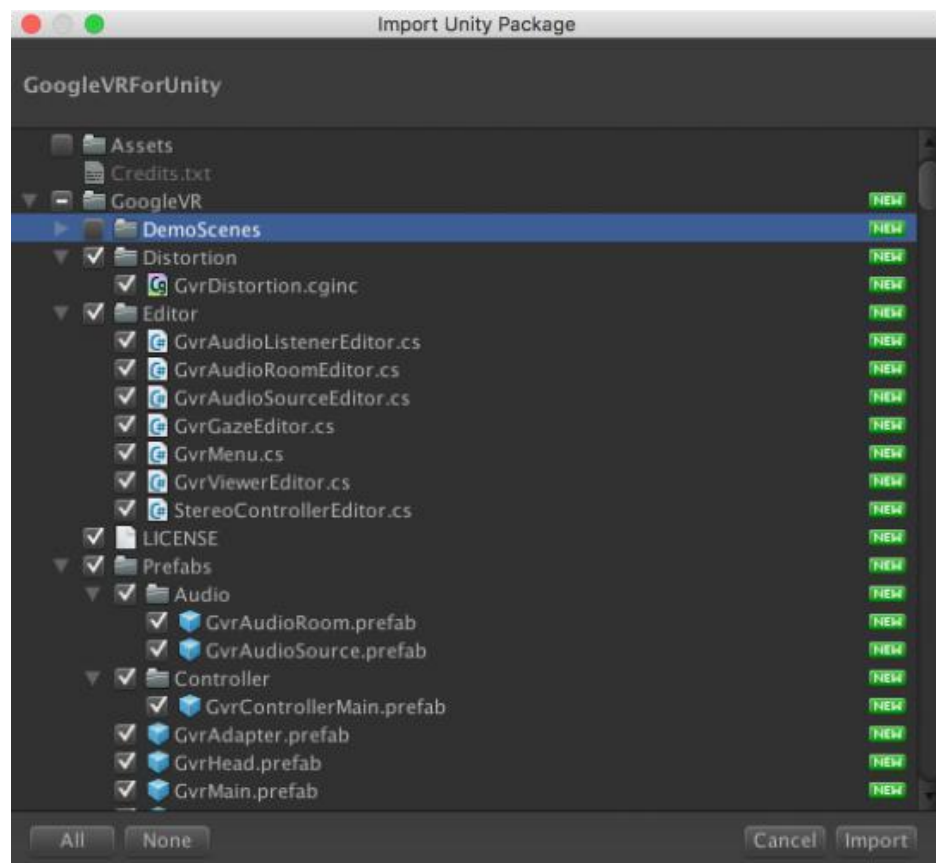
4.1.1 Cài đặt SDK và camera VR

Sử dụng Unity 5.3.4f1 và Google VR SDK 1.4

Download Unity: <https://unity3d.com/get-unity/download>

Download Google VR SDK 1.4 for Unity: <https://github.com/googlevr/gvr-unity-sdk/releases/tag/v1.40.0>

Sau khi download SDK, giải nén và import vào project Unity đã tạo sẵn.
(Import file GoogleVRForUnity.unitypackage).

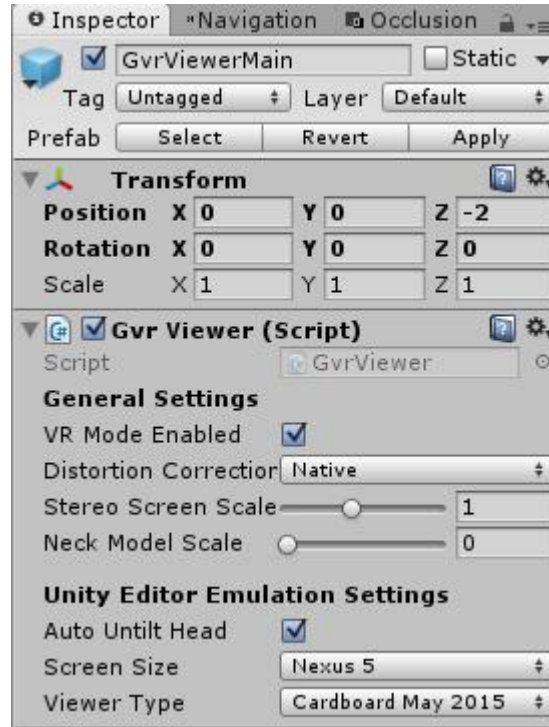


Hình 4.1: Màn hình import sdk

- Camera VR:

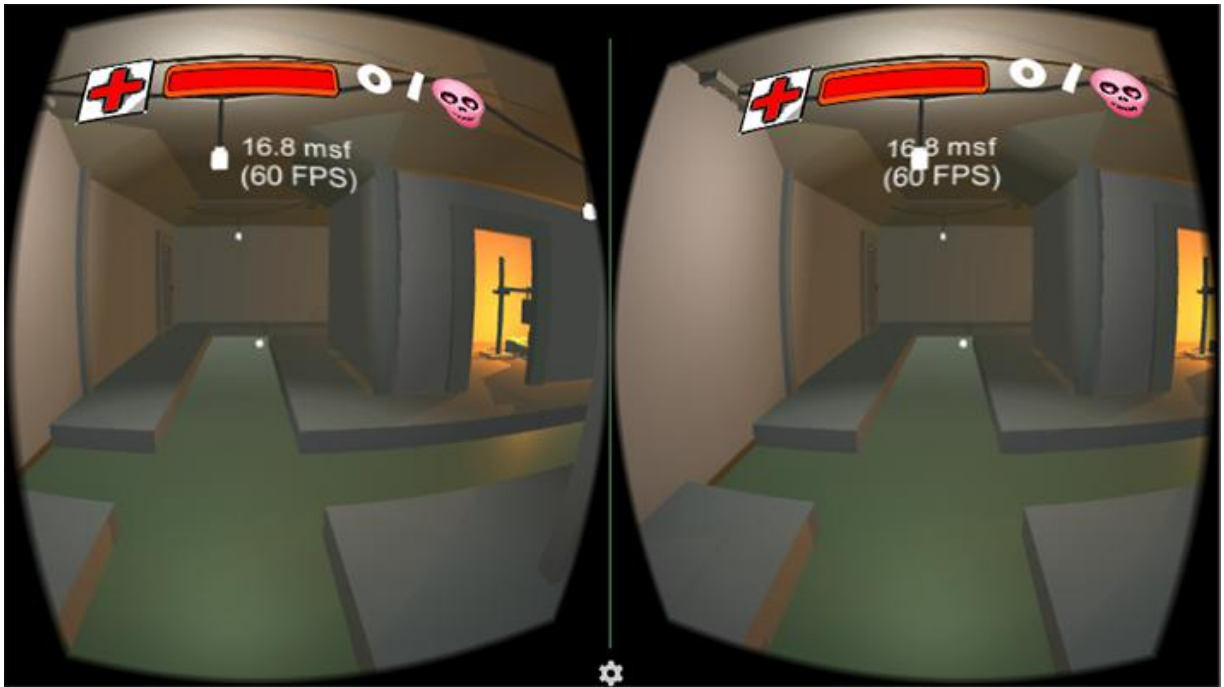
Để tạo không gian ảo, màn hình của game cần phải chia ra thành 2 phần, rồi qua lens của chiếc kính, mỗi mắt sẽ có một màn hình riêng tạo nên hiệu ứng 3D.

Chính vì vậy, trong SDK có prefab GvrViewerMain , kéo prefabs này vào scene.



Hình 4.2: Inspector của GvrViewerMain

Nó sẽ thực hiện việc chia màn hình thành 2 phần giống như hình bên dưới.



Hình 4.3: Giao diện camera trên VR

Ngoài ra , nếu muốn camera trong game follow theo nhân vật thì phải để camera là child của Object đó.

- Tương tác trong game:

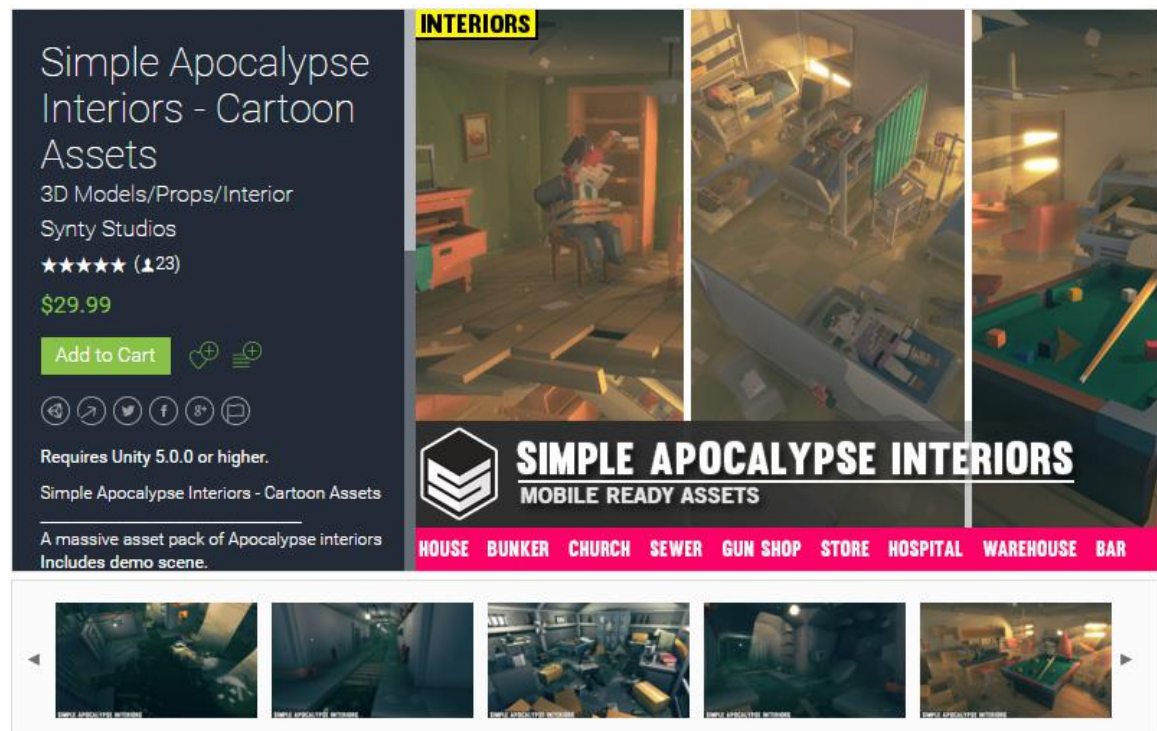
Cardboard chỉ có duy nhất 1 button phía bên phải → Trigger. Nhận biết trong SDK:

`GvrViewer.Instance.Triggered`

Ngoài ra, để tương tác trong game, chúng ta phải thông qua 1 đốm tròn ở giữa (giống như trong hình) để biết rằng chúng ta tương tác với đối tượng nào. Trong Unity có 1 kỹ thuật: Raycast dùng để xử lý việc này.

Raycast là 1 tia bắn từ 1 vị trí nhất định theo 1 vector trong khoảng cách nào đó sẽ bắn trúng 1 Collider tạo 1 điểm hit. Khi điểm hit được xác định, chúng ta sẽ xử lý sự kiện tại đây.

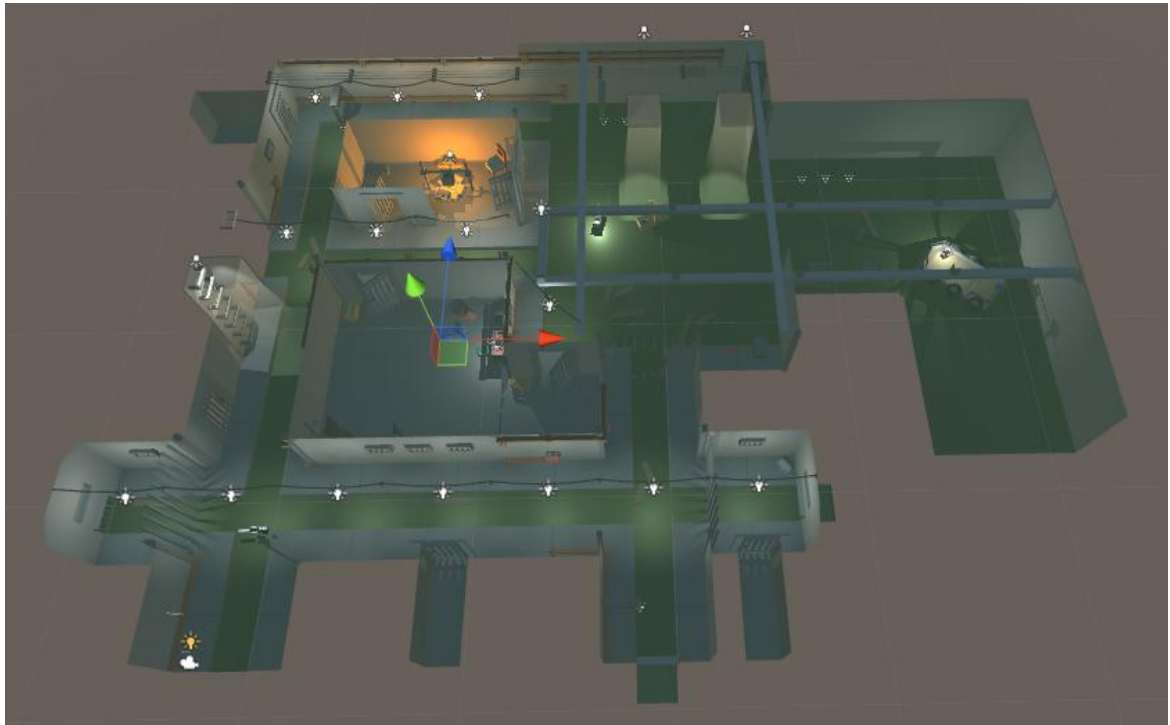
4.1.2 Environment



Hình 4.4 Trang model environment trên asset store

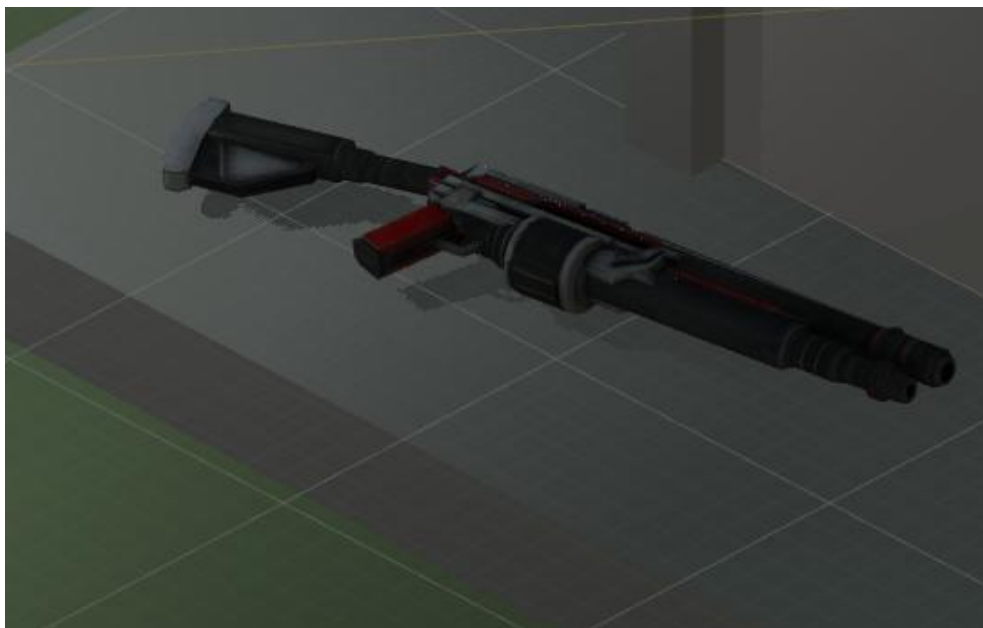
Map Sewer Environment

- Nhân vật sẽ di chuyển trong map này để tiêu diệt hết tất cả các enemy trong đây để vượt qua màn chơi mới.



Hình 4.5: Bản đồ trong game

- Map mô phỏng một ngôi nhà nhưng bị ngập nước, có nhiều căn phòng với các đồ vật hỗn độn.
- Các đồ vật trong map:
 - + Súng: Vũ khí chính của Player, dùng để tiêu diệt enemy



Hình 4.6 Súng

+ Gỗ nổi lềnh bềnh



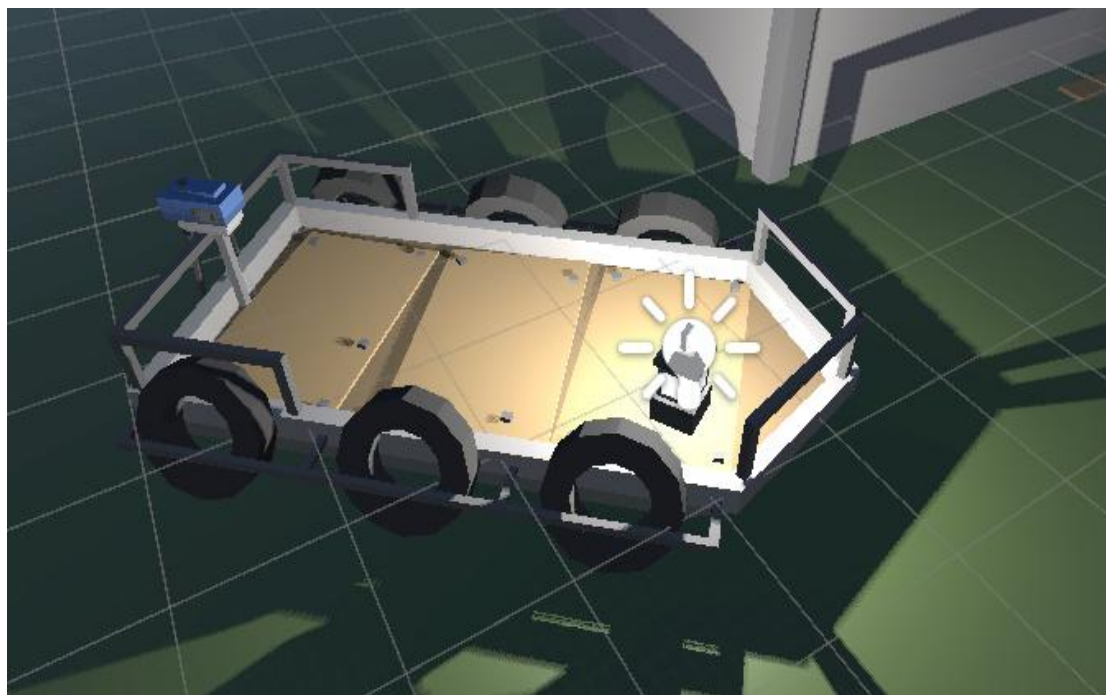
Hình 4.7 Thanh gỗ

- Bàn ghế, sách vở, các đồ vật trong phòng bị xáo trộn, không ngăn nắp



Hình 4.8 Bàn, sách vở và nhiều vật nhỏ khác

+ Xuồng cứu hộ:



Hình 4.9 Xuồng cứu hộ

+ Các bóng đèn: tạo ánh sáng cho ngôi nhà



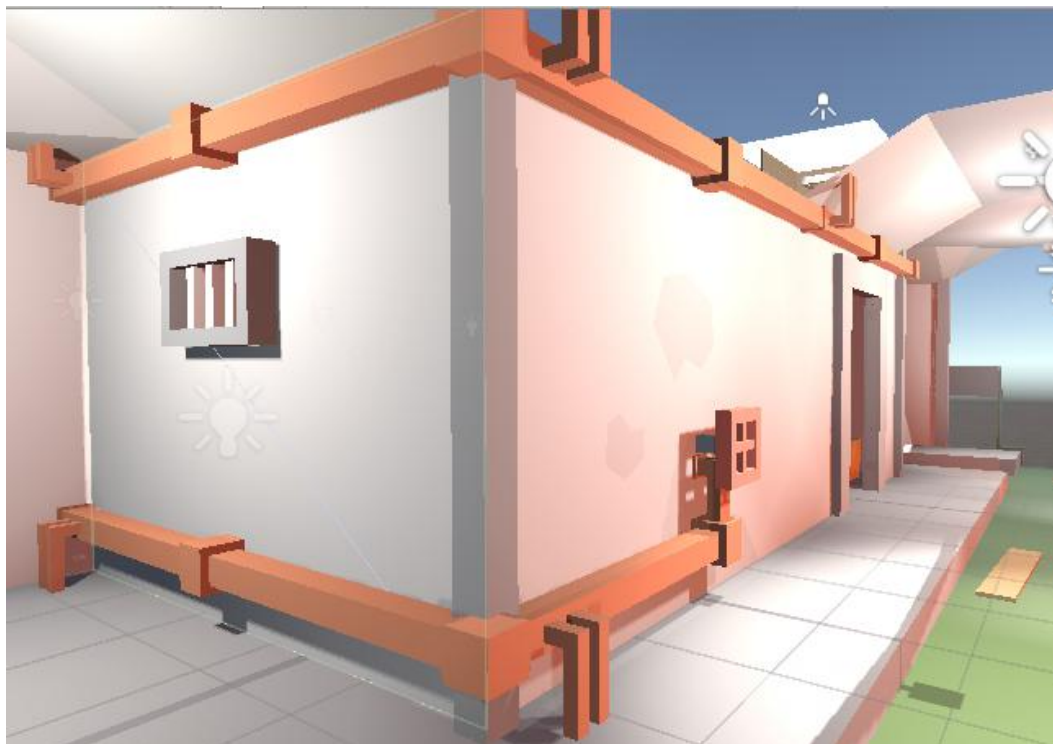
Hình 4.10 Đèn

+ Các căn phòng có cửa chính và cửa sổ:



Hình 4.11 Cửa chính và cửa sổ

+ Các đường ống thoát nước:



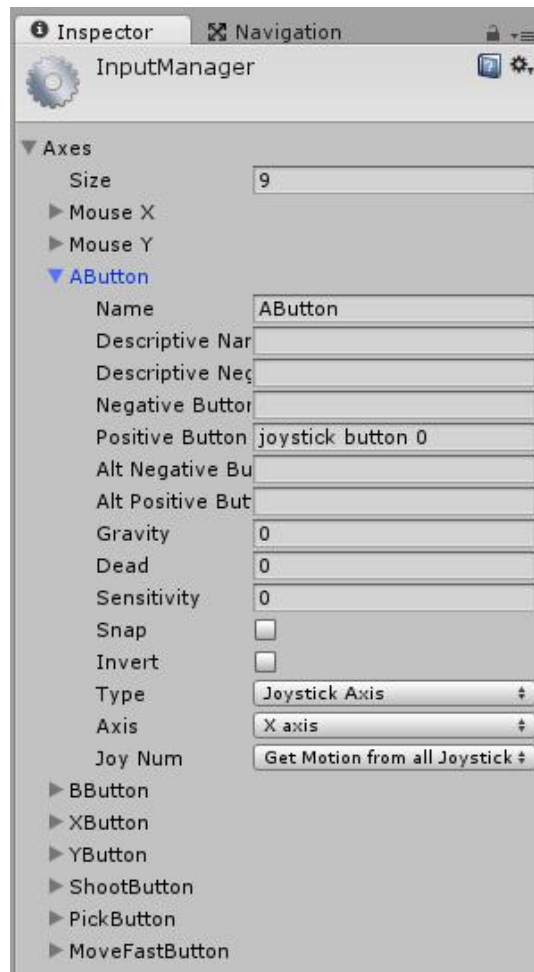
Hình 4.12 Đường ống thoát nước

4.2 Điều khiển và nhật vũ khí

4.2.1 Điều khiển

- Gamepad

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



Hình 4.13 Input manager

Điều khiển di chuyển:

- + Đi tới: joystick button 4 (Y)
- + Đi lùi: joystick button 0 (A)
- + Qua trái: joystick button 3 (X)
- + Qua phải: joystick button 1 (B)

Điều khiển nhặt vật, súng: joystick button 7

Điều khiển bắn: joystick button 6

Điều khiển di chuyển nhanh: joystick button 9

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

Nút của cardboard:

- Sử dụng trigger của google cardboard để điều khiển nhân vật đi hoặc đứng im. Khi di chuyển, nhân vật sẽ di chuyển theo hướng đầu người chơi (Camera trong game follow theo nhân vật) → Camera là child của Object Player.
- Player:
 - + Speed : Tốc độ di chuyển
 - + Move: để nhận biết nhân vật di chuyển hay đứng yên thông qua Trigger.

Cách sử dụng trong Unity: GvrViewer.Instance.Triggered

+ Head: Hướng di chuyển của nhân vật

```
if(GvrViewer.Instance.Triggered)
{
    m_move = !m_move;
}

if (m_move)
{
    m_forward = m_head.transform.forward;
    m_forward.y = m_y;

    m_playerRigid.velocity = m_forward * m_speed;
}
else
{
    m_playerRigid.velocity = Vector3.zero;
}
```

Hình 4.14 Code xử lý tự động di chuyển

4.2.2 Nhặt vũ khí

Nhặt súng và gậy

Bằng gamepad: sử dụng pick button để nhặt

Nút của cardboard:

- Khi nhặt súng cũng cần phải tương tác thông qua Trigger. Sử dụng kỹ thuật raycast trong unity để thực hiện. Nhưng người chơi chỉ nhặt được súng ở một khoảng cách nhất định khi đến gần súng.

```
Physics.Raycast(recticle.transform.position, recticle.transform.forward, out hit);
if(hit.collider!=null)
{
    if (hit.collider.tag == "Gun" && GvrViewer.Instance.Triggered)
    {
        GameObject gun = hit.collider.gameObject;
        float distance = Vector3.Distance(gun.transform.position, m_player.transform.position);
        if (distance < 3.0f)
        {
            gun.SetActive(false);
            m_gun.SetActive(true);
            m_isGun = true;
        }
    }
}
```

Hình 4.15 Code xử lý nhặt vật

Hình ảnh súng trong game:



Hình 4.16 Súng

Và sau khi nhặt súng lên:



Hình 4.17 Khi nhặt súng lên



Hình 4.18 Khi nhặt gậy lên

4.3 Các loại enemy

4.3.1 Enemy

- Loại enemy đánh gần:
- + Zombie bunny (xanh):



Hình 4.19 Zombie bunny

- + Zombie bear (hồng):



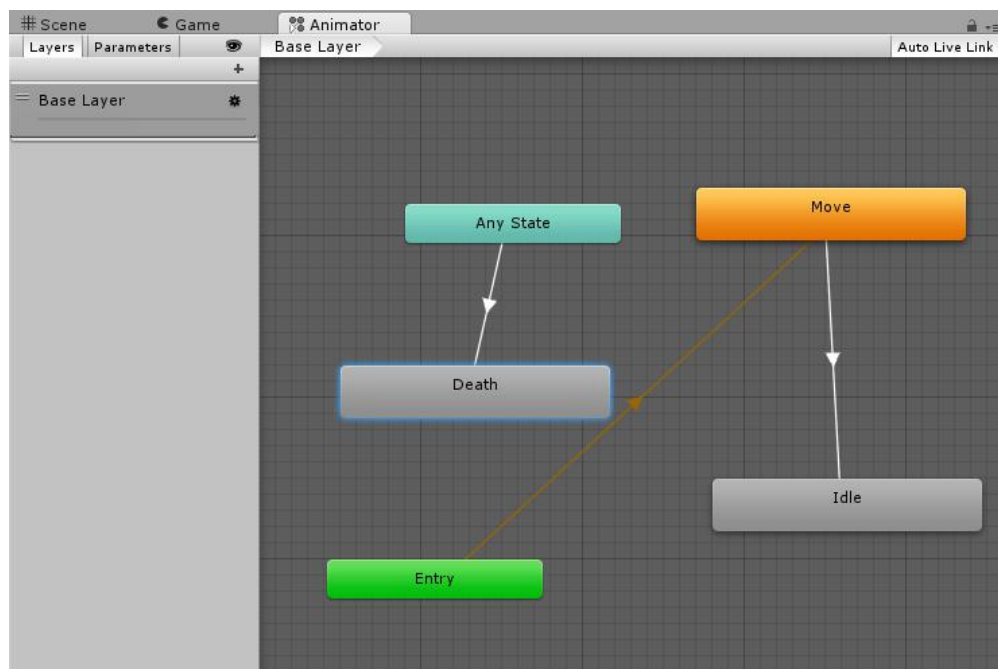
Hình 4.20 Zombie bear

+ Zombie hellephant (vàng):



Hình 4.21 Zombie hellephant

4.3.2 Animation



Hình 4.22 Animator của enemy

- Idle: animation lúc enemy đứng im.
- Move: animation lúc enemy di chuyển
- Death: animation lúc enemy chết.

4.3.3 Trạng thái

- Sống (alive):
- Chết (died): khi enemy hết máu nó sẽ chết
- Tấn công (attack): tấn công player khi đủ tầm

4.3.4 Tìm đường đi tới nhân vật

- Sử dụng NavMesh của Unity.

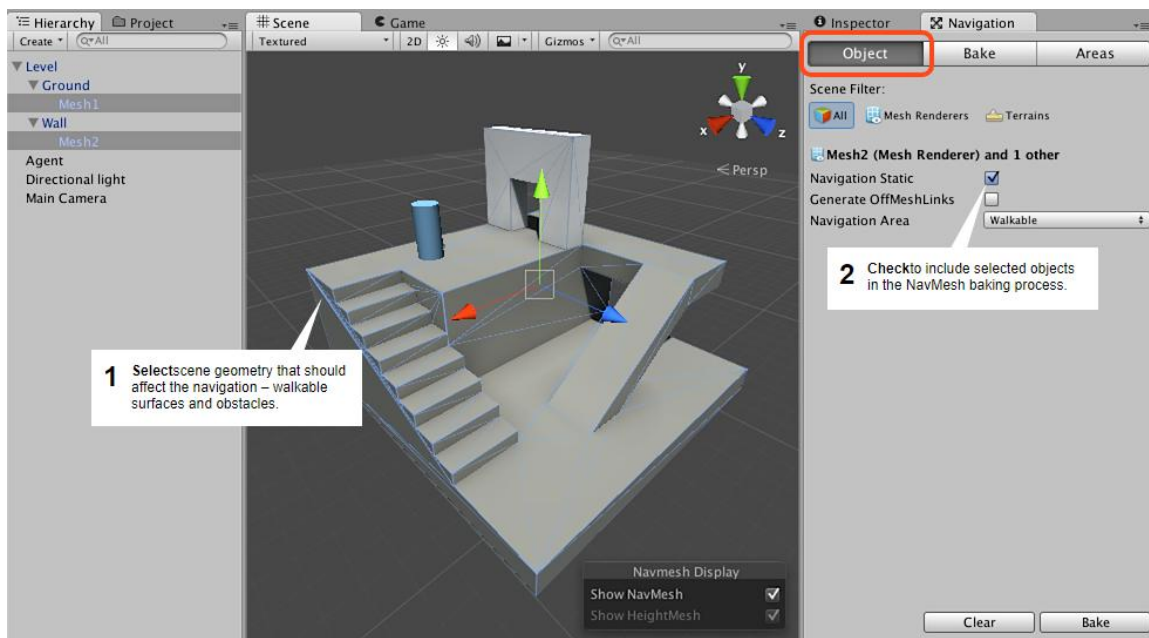
NavMesh

class của UnityEngine.AI

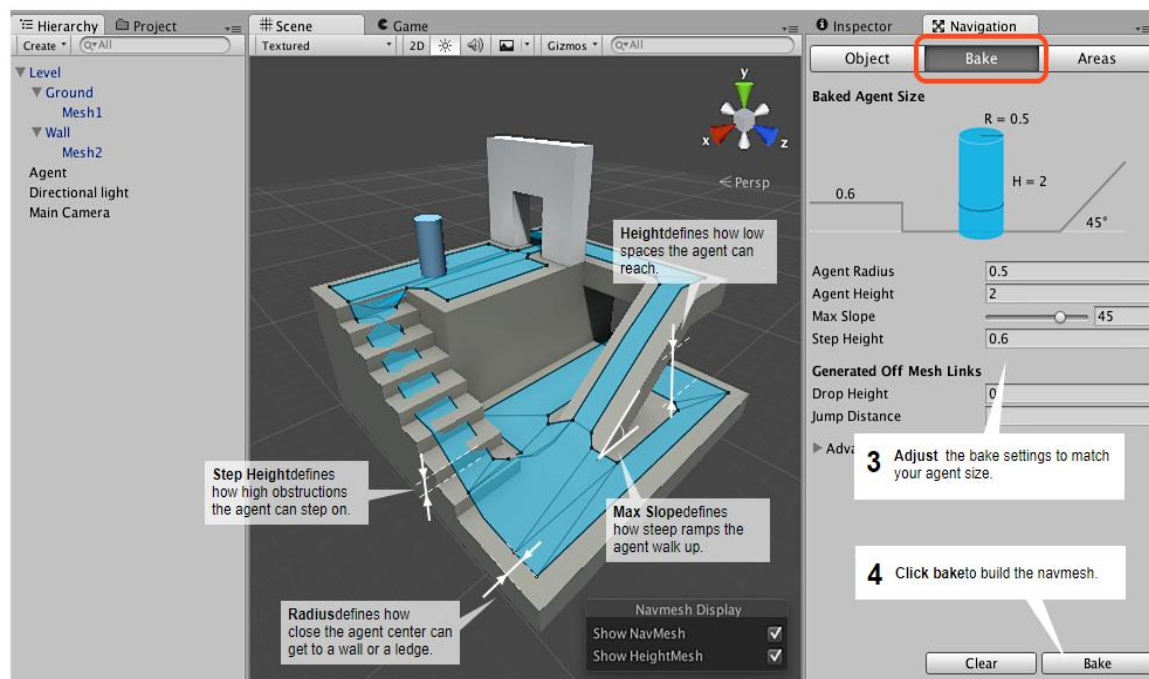
NavMesh là một lớp có thể được sử dụng để làm các truy vấn không gian, giống như tìm đường đi (pathfinding) và khả năng di chuyển (walkability), thiết lập các chi phí tìm đường với nhiều loại diện tích cụ thể, và để tinh chỉnh hành vi toàn cầu của tìm đường đi và tránh né.

Để sử dụng được thì ta cần tiến hành bake NavMesh cho Scene:

+ Building a NavMesh

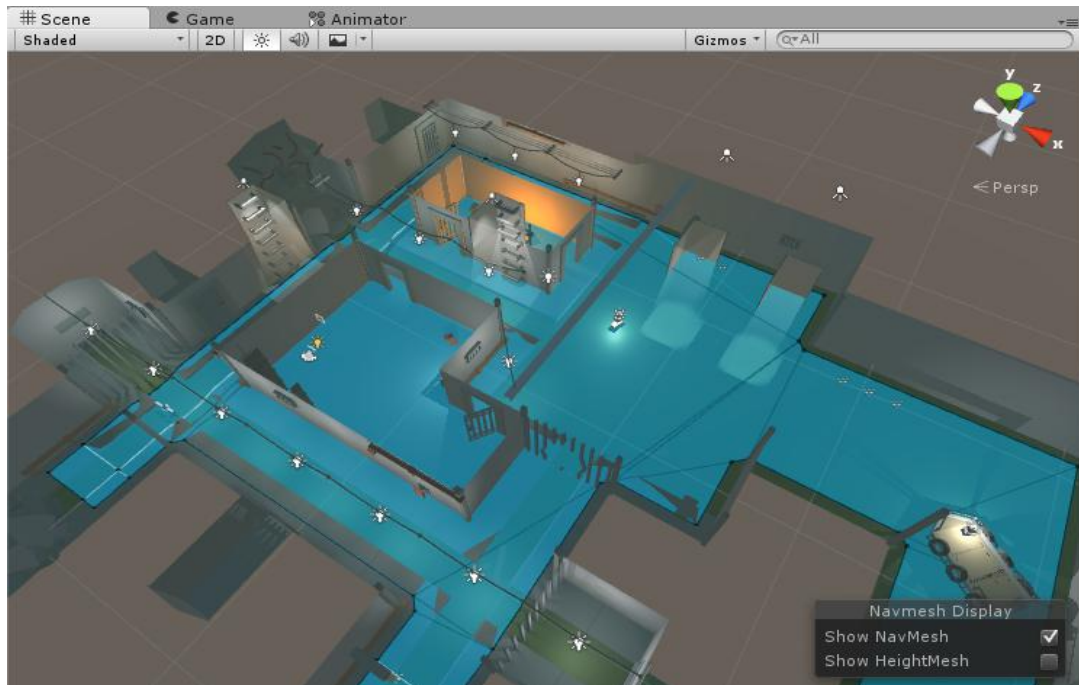


Hình 4.23 Giao diện navigation tab object



Hình 4.24 Giao diện navigation tab bake

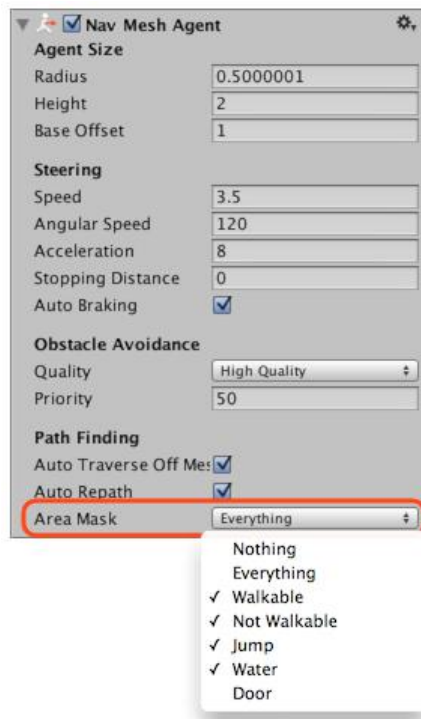
+ Đây là NavMesh trong game:



Hình 4.25 Giao diện trong map sau khi bake xong

+ NavMeshAgent

Là component được gắn vào nhân vật trong game cho phép nó navigate scene sử dụng NavMesh.



Hình 4.26 Nav mesh agent

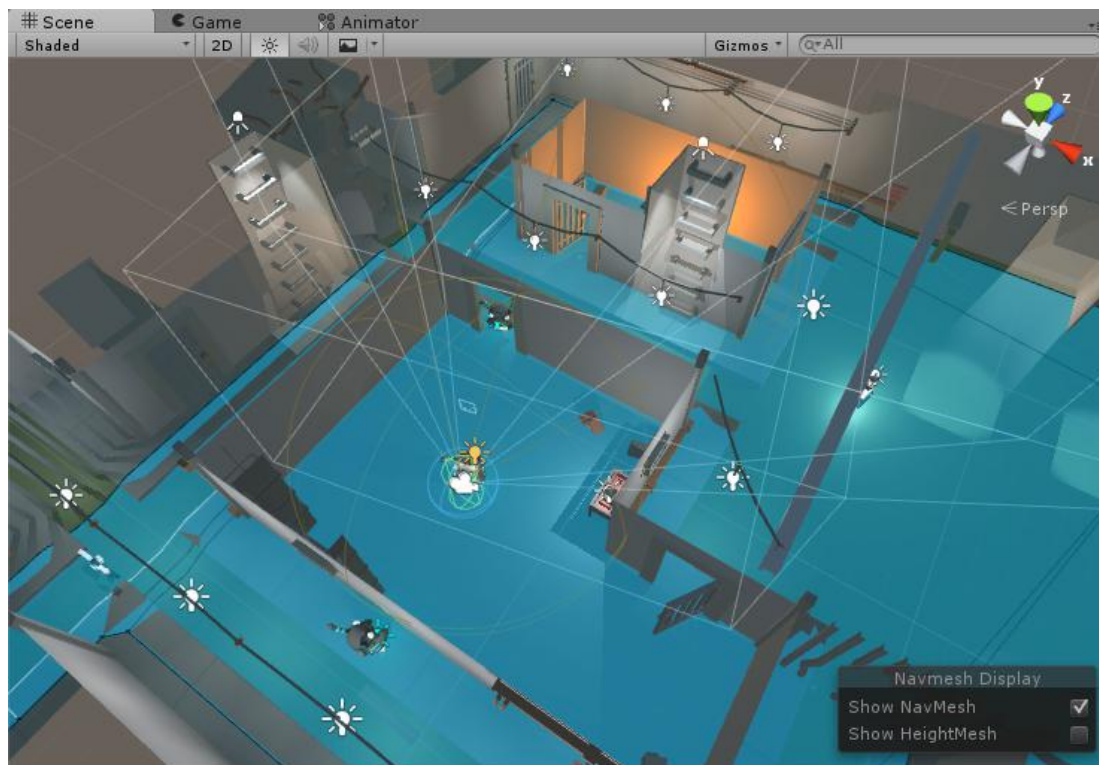
+ EnemyMovement.cs

```
public class EnemyMovement : MonoBehaviour
{
    Transform player;
    //PlayerHealth playerHealth;
    EnemyHealth enemyHealth;
    NavMeshAgent nav;

    0 references | 0 changes | 0 authors, 0 changes
    void Awake ()
    {
        player = GameObject.FindGameObjectWithTag ("Player").transform;
        //playerHealth = player.GetComponent <PlayerHealth> ();
        enemyHealth = GetComponent <EnemyHealth> ();
        nav = GetComponent <NavMeshAgent> ();
    }

    0 references | 0 changes | 0 authors, 0 changes
    void Update ()
    {
        if (enemyHealth.currentHealth > 0) // && playerHealth.currentHealth > 0
        {
            nav.SetDestination(player.position);
        }
        else
        {
            nav.enabled = false;
        }
    }
}
```

Hình 4.27 Code xử lý di chuyển của enemy



Hình 4.28 NavMesh trong game

4.4 Xử lý tấn công bằng súng và gậy

4.4.1 Tấn công bằng súng

Line Renderer:

Được dùng để tạo hiệu ứng 1 đường thẳng của viên đạn khi bắn từ khẩu súng đến enemy (Vẽ tia sáng lúc bắn)

Để thêm component này vào Object hoặc là bạn có thể chọn object trên Hierachy
-> Inspector -> Add Component-> Effects -> LineRenderer

Raycast:

Raycast là 1 tia bắn từ 1 vị trí nhất định theo 1 vector trong khoảng cách nào đó sẽ bắn trúng 1 Collider tạo 1 điểm hit. Khi điểm hit được xác định, chúng ta sẽ xử lý sự kiện tại đây.

Trong game được sử dụng trong script GameController.cs:

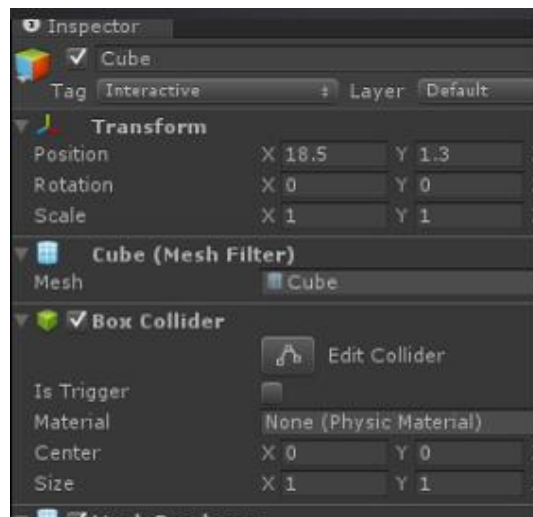
```
Physics.Raycast(recticle.transform.position, recticle.transform.forward, out hit);
if(hit.collider!=null)
{
    if (hit.collider.tag == "Gun" && GvrViewer.Instance.Triggered)
    {
        GameObject gun = hit.collider.gameObject;
        float distance = Vector3.Distance(gun.transform.position, m_player.transform.position);
        if (distance < 3.0f)
        {
            gun.SetActive(false);
            m_gun.SetActive(true);
            m_isGun = true;
        }
    }

    if (hit.collider.tag == "Enemy" && m_isGun)
    {
        Shoot();
        enemy = hit.collider.gameObject;
    }

    if (hit.collider.tag == "Untagged")
    {
        m_isEffect = false;
        Effect(m_isEffect);
    }
}
```

Hình 4.29 Code xử lý tấn công

- Cách sử dụng raycast :
 - + Sử dụng tag để xác định được đối tượng chúng ta cần tìm kiếm, gắn collider cho đối tượng đó.



Hình 4.30 Gắn collider cho object

+ Tạo vị trí bắn đi, trong game sử dụng `reticle.transform.position` : vị trí của hình tròn giữa camera → đảm bảo là tất cả các vị trí khi player nhìn đều có thể bắn trúng.

+ Thông số hit (điểm va chạm): `hit.collider.tag = " ... "`; xác định được đối tượng mà tia raycast đã bắn trúng và xử lý đối tượng.

Trong game, raycast được gọi trong hàm `Update()`, tức là raycast luôn luôn được bắn ra liên tục trong một khoảng thời gian nhất định.

Function `Shoot()`: Được gọi theo

```
void Shoot()
{
    SoundController.Sound.Shoot();

    m_timer += Time.deltaTime;

    if (m_timer > m_speedShoot)
    {
        m_timer = 0;
        m_isEffect = !m_isEffect;
        //Show light effect of gun
        Effect(m_isEffect);

        //Take damage to enemy
        if (enemy != null)
        {
            enemy.GetComponent<EnemyHealth>().TakeDamage(15, hit.point);
        }
    }
}
```

Hình 4.31 Code xử lý bắn

+ `m_timer` : để xác định chu kì của 1 lần bắn. Trong 1 khoảng thời gian sẽ gọi lại.

+ `hit.point` : Điểm va chạm để play effect khi bắn trúng enemy. Hiệu ứng khói trên người enemy.



Hình 4.32 Hiệu ứng khói bụi trên enemy

+ Hàm Effect:

Enable light khi bắn. (bóng đèn được gắn sẵn ở gần cây súng và luôn đi theo nhân vật, khi va chạm sẽ bật tắt tạo nên hiệu ứng)



Hình 4.33 Hiệu ứng phát sáng khi bắn

4.4.2 Tấn công bằng gậy

Một animation là một hình ảnh động mô tả một đối tượng nào đó trong game.

Ví dụ: có thể là một chiếc xe đang chạy, hay một nhân vật đang đi....

Một animation trong Unity có thể bao gồm nhiều hành động, một hành động như vậy gọi là một clip.

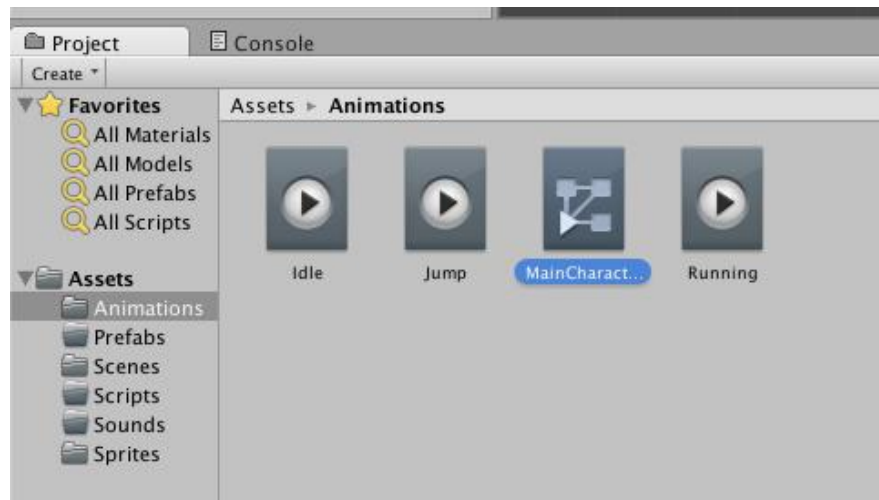
Ví dụ: một nhân vật có thể có các hành động đi, đứng, nhảy....

Có hai kỹ thuật để tạo animation (cả 2D và 3D): đó là kỹ thuật Mã frame và kỹ thuật skeletal hay spine.

Tạo animation bằng Unity:

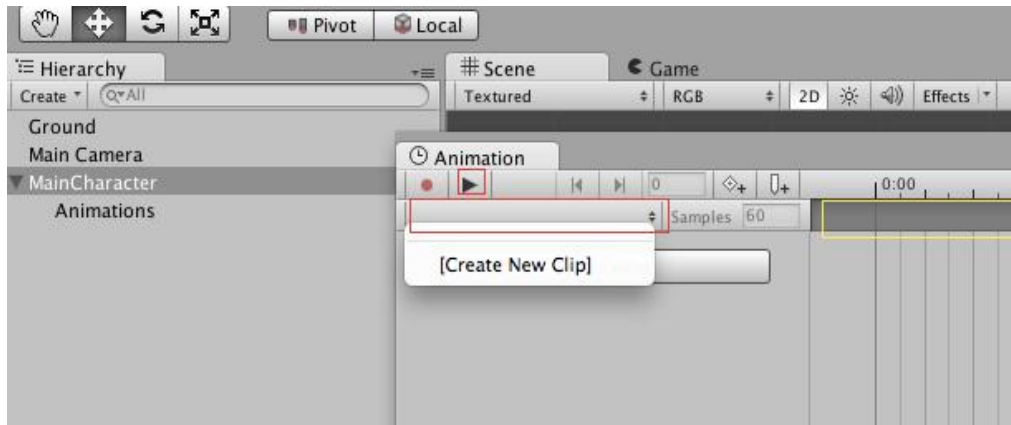
Bước 1: Chọn 1 GameObject cần tạo animation

Bước 2: Add một component là Animator. Animator này để quản lí các hành động (Animaton clip) của GameObject đó.



Hình 4.34 Animation

Bước 3: Chọn đối tượng MainCharacter ở cửa sổ Hierarchy, rồi chọn Menu -> Window -> Animation. Một cửa sổ Animation editor hiện ra như sau:

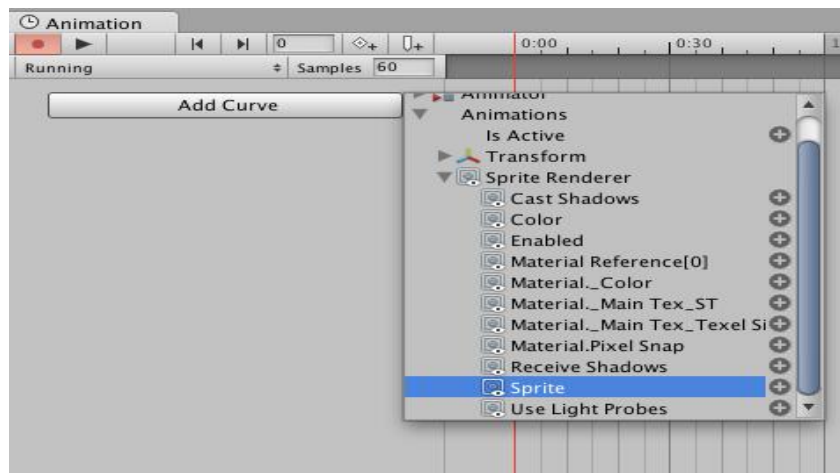


Hình 4.35 Trang tạo animation

- * Hình chữ nhật đỏ nhỏ: nút play để xem trước animation
- * Hình chữ nhật đỏ lớn: danh sách các clip hiện thời của animation
- * Hình chữ nhật vàng: thanh Mã frame

Đầu tiên ta sẽ click vào danh sách clip rồi chọn Create New Clip, ta đặt tên clip là Running, rồi save lại ở thư mục Animations của Assets.

Bước 4: Ở cửa sổ Animation Editor, chọn Add Curve, chọn Animations cần làm cho gameobject.



Hình 4.36 Hình chọn animation cần làm

Thực hiện đánh gần cho player:

Player sẽ có thêm 1 trạng thái là đánh gần, trong game nhân vật sẽ cầm 1 khúc gỗ khi chưa nhặt súng. Giúp cho người chơi tiêu diệt quái ở khoảng cách gần.



Hình 4.37 Animation gậy

Ngoài ra sẽ có các animation cho vũ khí khi nhân vật di chuyển và khi tiêu diệt enemy.

4.5 Health HUD (head-up display)

4.5.1 Health HUD, xử lý thanh máu

Trong trò chơi điện tử, màn hình hiển thị HUD hoặc thanh trạng thái là phương pháp thông tin được chuyển tiếp trực tiếp tới người chơi như là một phần của giao diện người dùng của trò chơi. Nó lấy tên của nó từ màn hình head-up được sử dụng trong máy bay hiện đại.

HUD thường được sử dụng để hiển thị đồng thời một số thông tin bao gồm máu của nhân vật chính, các mục và chỉ số về tiến trình trò chơi (như điểm số hoặc cấp độ).

Health HUD:



Hình 4.38 Health HUD

- + Biểu tượng heal.
- + Hiển thị thanh máu màu đỏ.
- + Lượng máu của người chơi là 100.

Xử lý thanh máu:

Scripts PlayerHealth.cs

```
public void TakeDamage (int amount)
{
    blood.Play("Blood", -1, 0);
    damaged = true;

    currentHealth -= amount;

    healthSlider.value = currentHealth;

    //playerAudio.Play ();

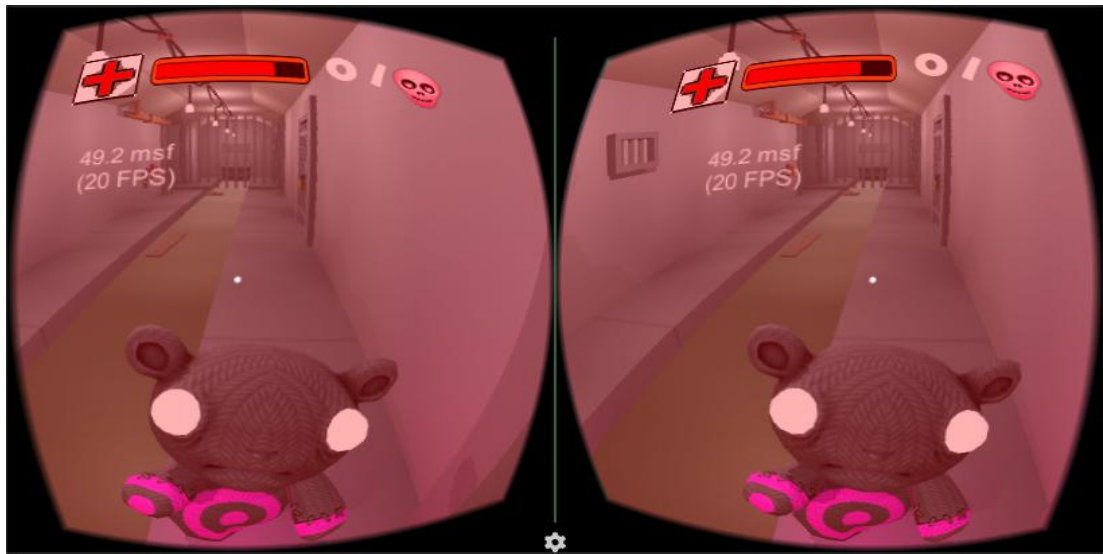
    if(currentHealth <= 0 && !isDead)
    {
        Death ();
    }
}
```

Hình 4.39 Code xử lý khi người chơi bị tấn công

Về đoạn code xử lý thanh máu:

- + Người chơi có 100 máu, khi enemy tấn công (attack) vào người chơi thì sẽ trừ đi 1 lượng máu (amount).

4.5.2 Hiệu ứng lúc người chơi mất máu

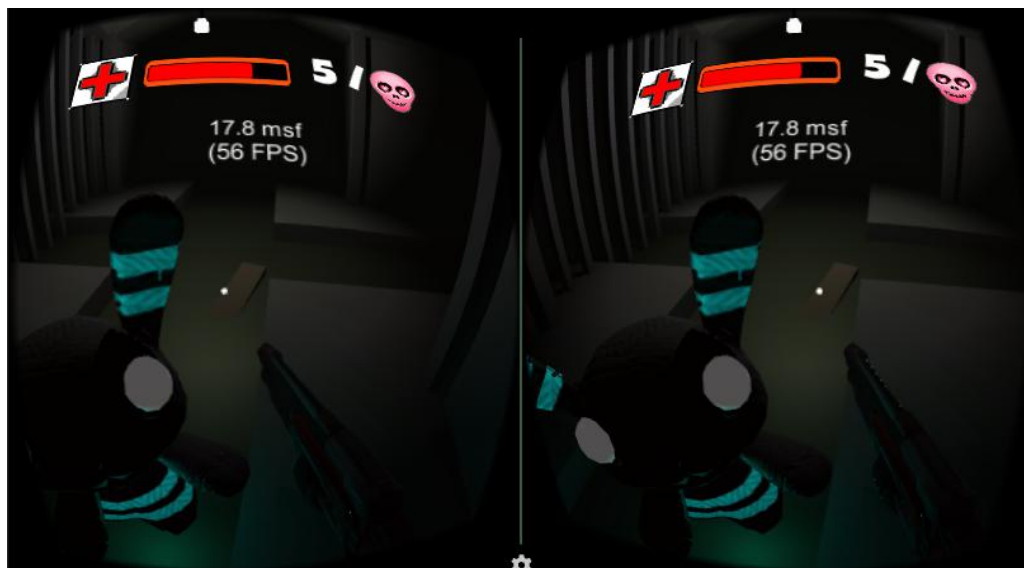


Hình 4.40 Hiệu ứng khi người chơi bị tấn công

- Hiện thị hiệu ứng animation màu đỏ đậm chuyển đến nhạt toàn màn hình.

4.6 Tính và hiện thị số enemy tiêu diệt được

Tính số enemy mà người chơi tiêu diệt được và hiển thị lên camera VR:



Hình 4.41 Số enemy bị tiêu diệt hiển thị trên màn hình

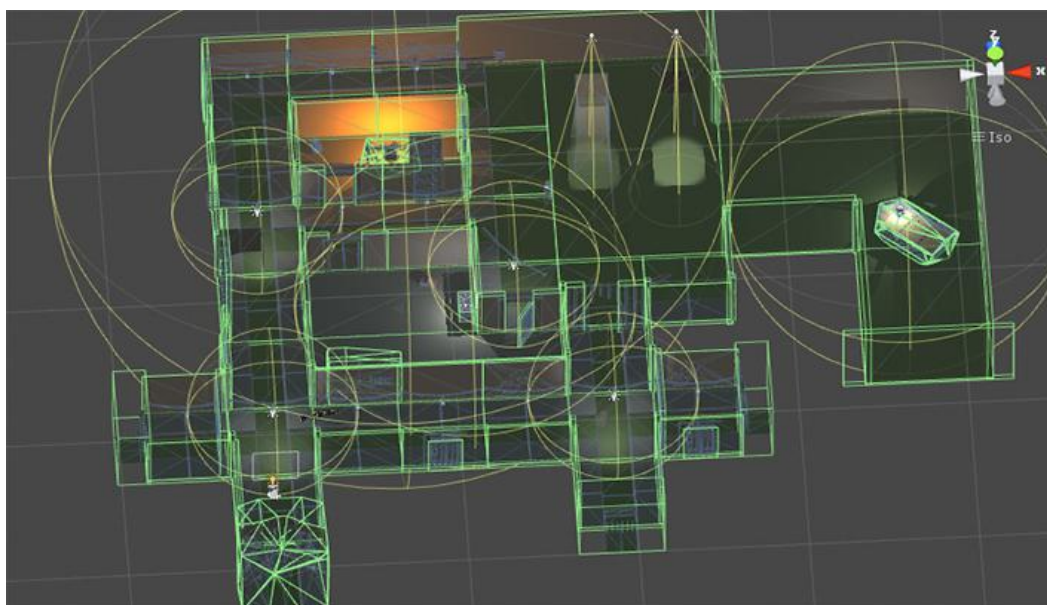
4.7 Tạo level 1 và level 2

4.7.1 Level 1

Trong level 1 người chơi chỉ có thể sử dụng súng.



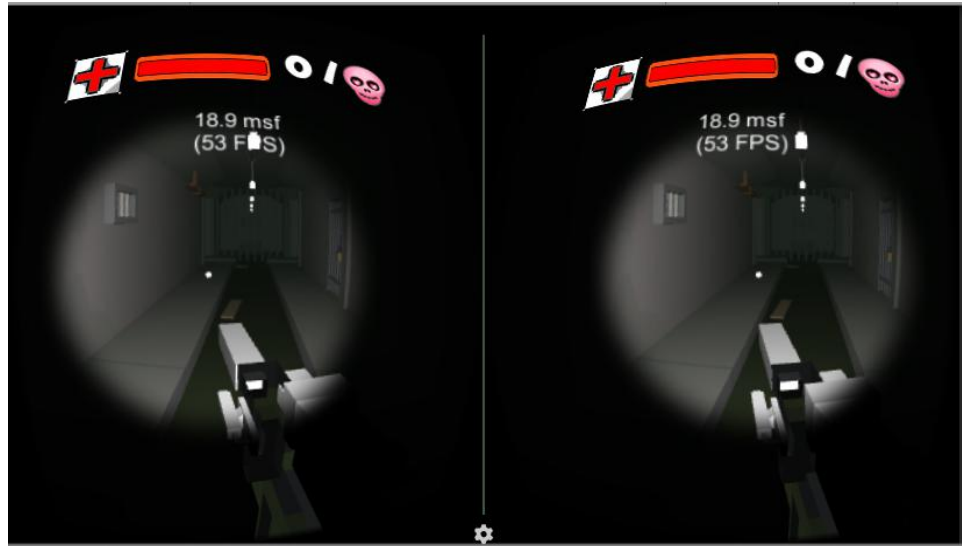
Hình 4.42 Người chơi cầm súng



Hình 4.43 Bản đồ level 1

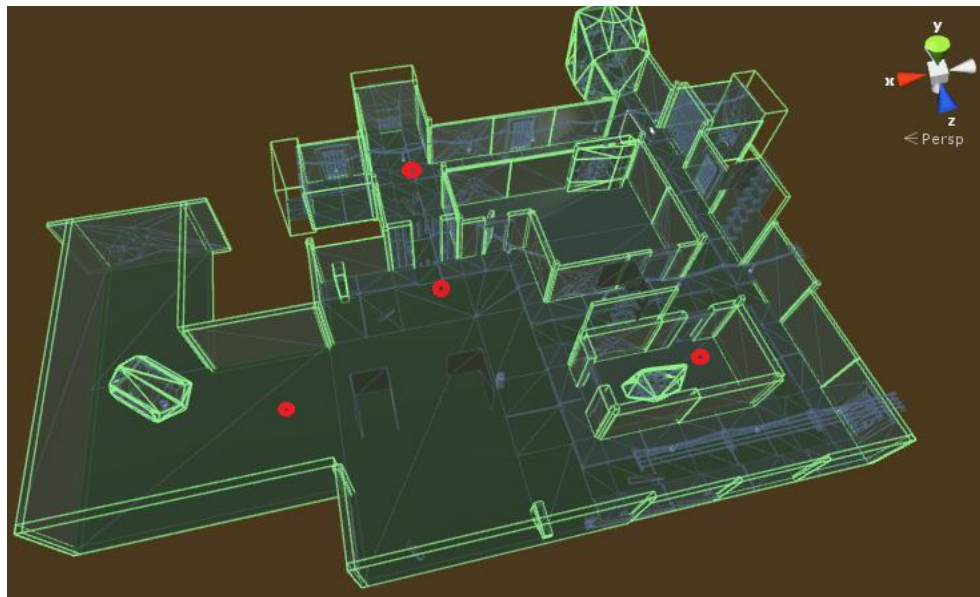
4.7.2 Level 2

- Sử dụng environment của Level 1 nhưng đã tắt hết bóng điện. Ở map này, người chơi sẽ dựa vào đèn pin để di chuyển và nhìn thấy enemy.
- Trong level 2 có súng và gậy



Hình 4.44 Người chơi cầm gậy

- Quái vật sẽ được sinh ra tại các điểm đỏ như trong hình:



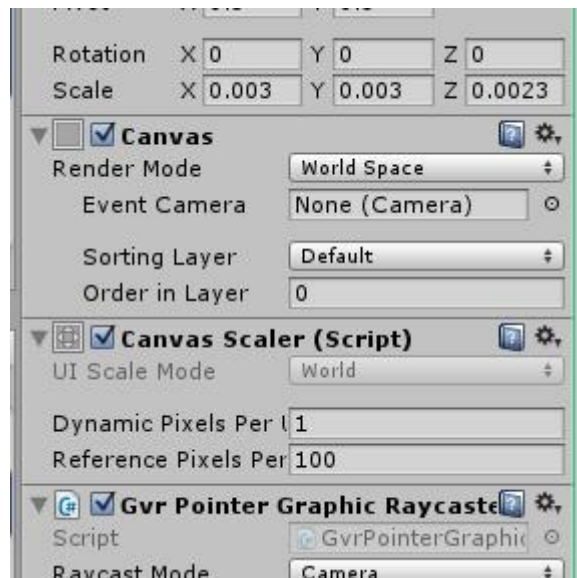
Hình 4.45 Bản đồ level 2

4.8 Màn hình menu, chọn level, game over và win

4.8.1 Liên quan đến UI, có 3 tùy chọn hiển thị canvas

- Screen Space – Overlay. Canvas sẽ được vẽ lên layer cao nhất của màn hình và nằm trên mọi game object khác. Canvas với render mode này hoàn toàn không phụ thuộc vào camera.
- Screen Space – Camera. Đối với mode này, ta cần chỉ định một camera cho canvas, nó sẽ được render theo camera. Nếu như không có camera được chỉ định thì canvas và các thành phần bên trong sẽ không được render.
- World Space. Với tùy chọn này, đối tượng canvas sẽ được xem như một game object thông thường. Tùy chọn này sử dụng event camera thay vì render camera. Ngoài các chức năng như render camera, event camera còn có thêm chức năng bắt sự kiện, dựa trên thứ tự render, toạ độ z, ... của các đối tượng UI.

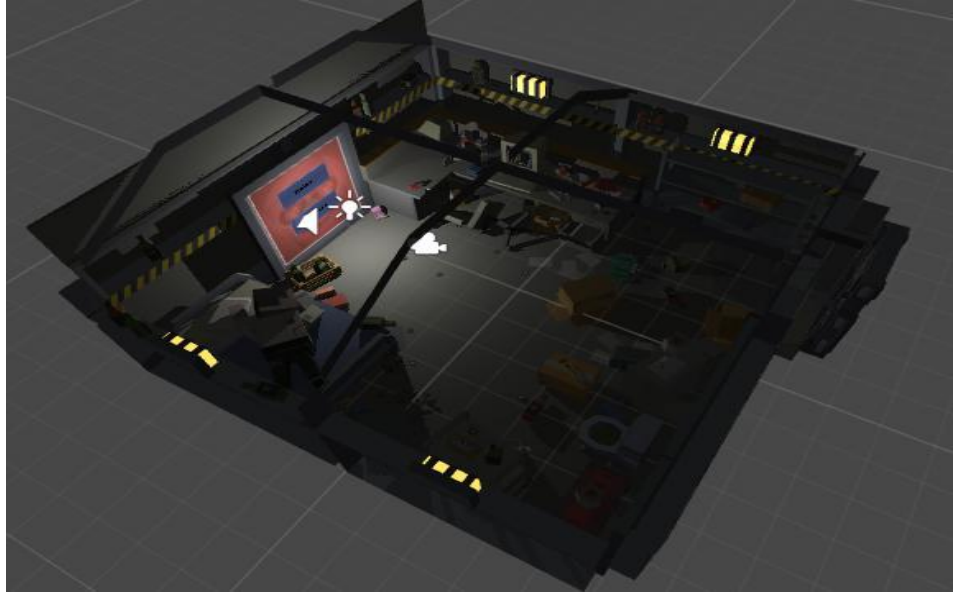
Trong VR, Canvas thường được sử dụng ở chế độ World Space:



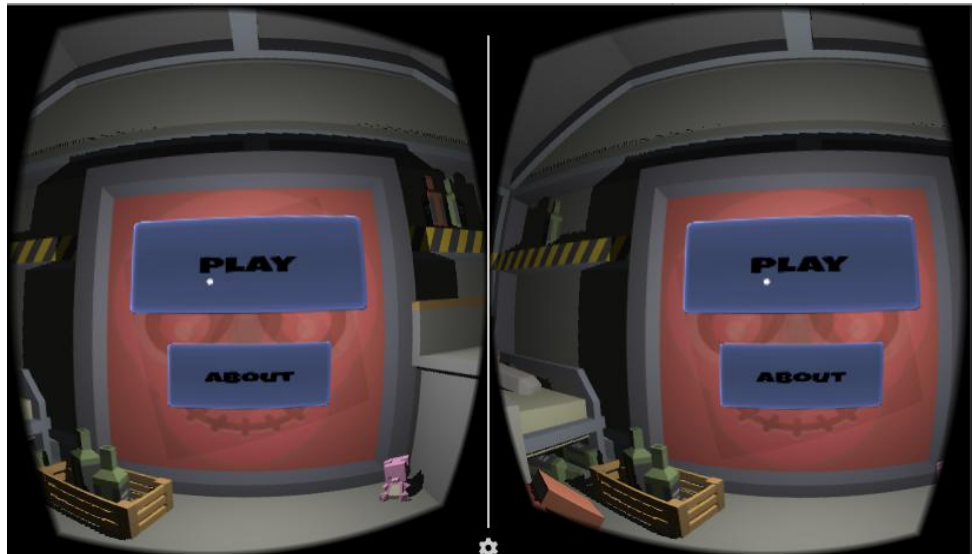
Hình 4.46 Inspector của canvas

4.8.2 Màn hình Menu

Nằm trong environment là một ngôi nhà, có một model 3D là màn hình dùng để gắn canvas và button vào.

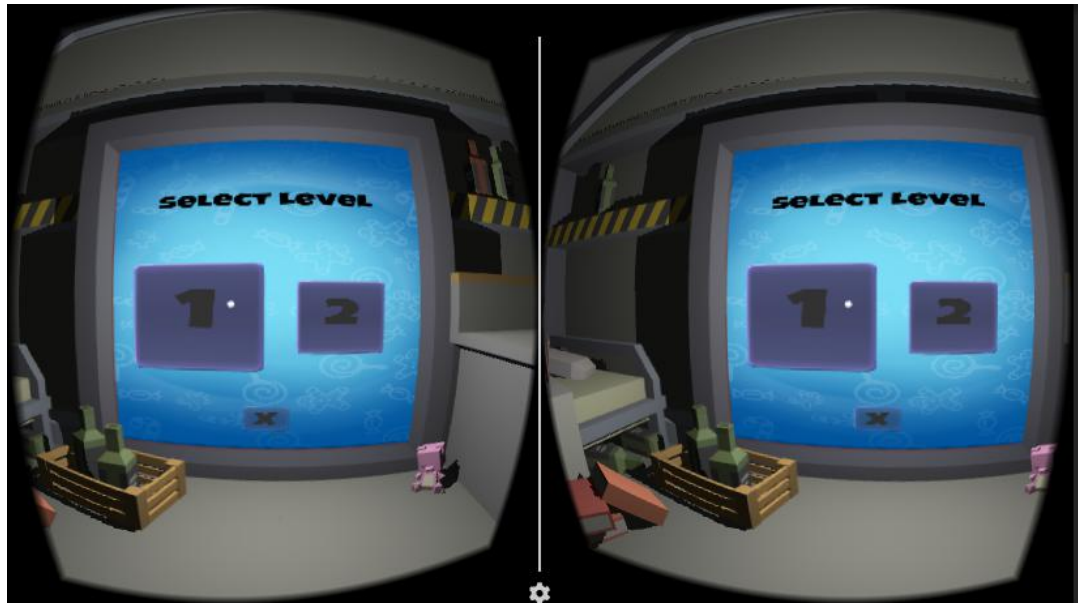


Hình 4.47 Model nhà tại menu



Hình 4.48 Canvas menu

4.8.3 Màn hình chọn màn chơi



Hình 4.49 Canvas chọn màn chơi

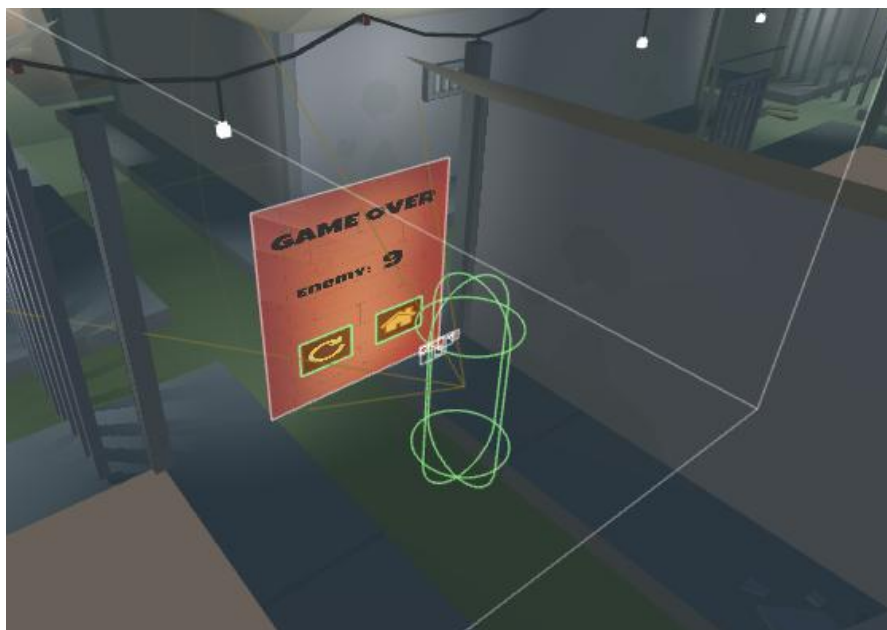
Game sẽ có 2 màn chơi. Màn hình chọn màn chơi được hiển thị dưới dạng pop-up. Animation cho button được sử dụng iTween để scale to nhỏ khi tương tác.

4.8.4 Màn hình GameOver và Win

Được xử lý trực tiếp trong scene Game Play. Có các Animation FadeIn, FadeOut, Scale khi tương tác.



Hình 4.50 Canvas win



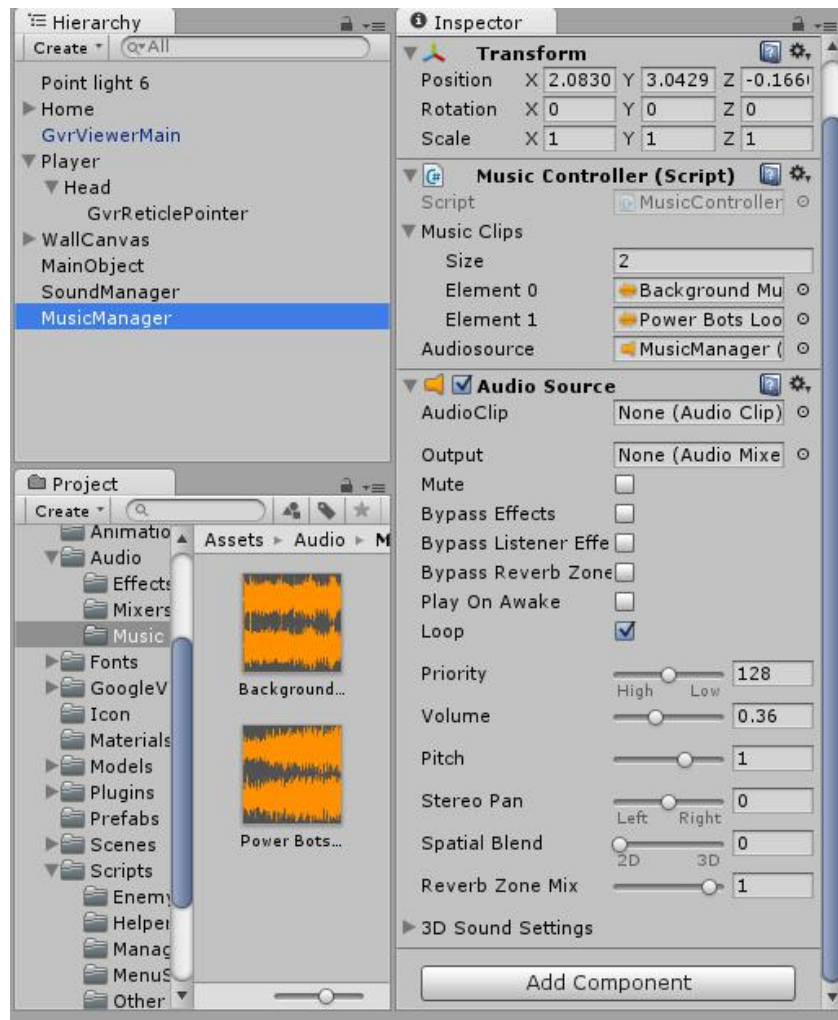
Hình 4.51 Canvas game over

Tương tác với Canvas trong Script:

- Sử dụng tag để xác định đối tượng.
- Dùng raycast để tương tác và xử lý sự kiện cho đối tượng thông qua tag đó.

4.9 Nhạc nền, âm thanh

- Thêm âm thanh background music và sound cho game ở menu scene và play scene



Hình 4.52 Music manager

4.10 Màn hình chọn control

- Game sẽ có 2 chế độ để điều khiển :
 - + Điều khiển theo cách thông thường khi không có Gamepad, nhân vật sẽ tự động di chuyển theo đầu (head) của người chơi, tương tác trong game thông qua nút của google cardboard.
 - + Kết nối qua Bluetooth và điều khiển nhân vật bằng gamepad tay cầm.



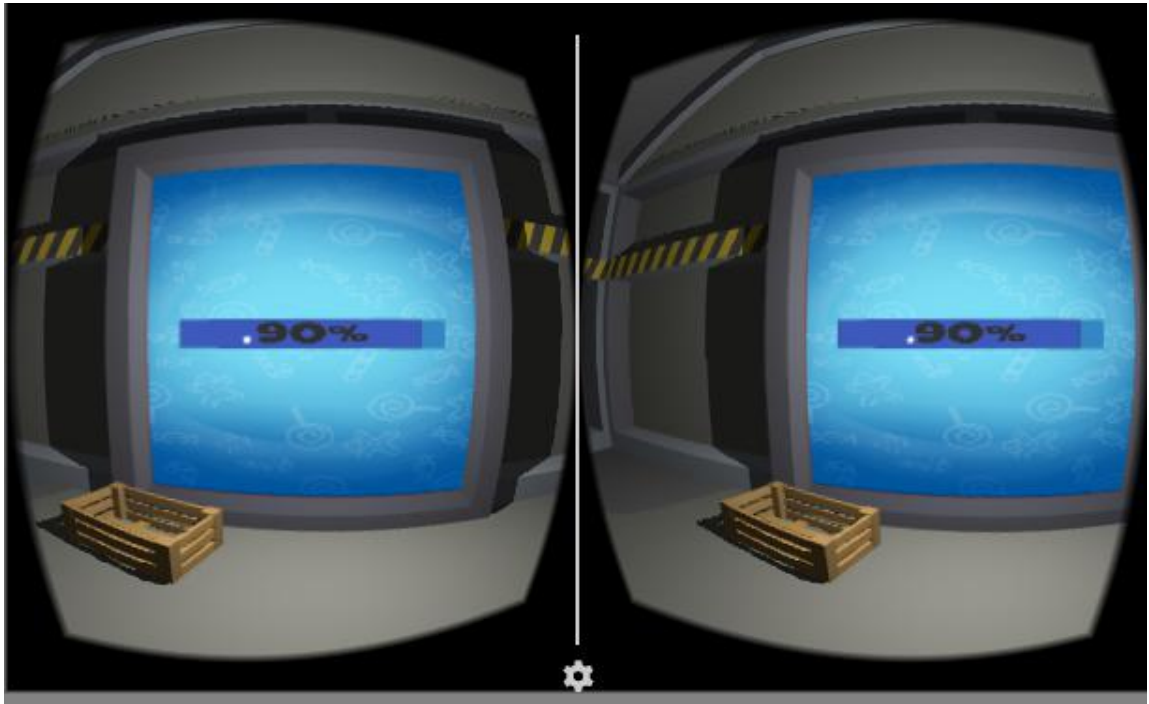
Hình 4.53 Màn hình chọn control

- Điều khiển mặc định khi mở game là chế độ điều khiển không có gamepad.

4.11 Màn hình loading cho game

Thêm loading cho game khi chuyển scene.

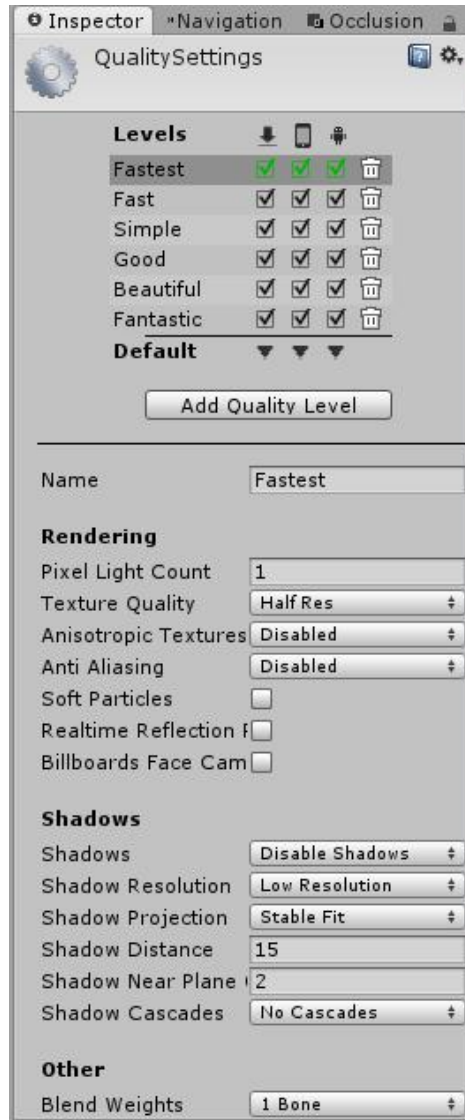
Sử dụng `Application.LoadLevelAsync()` và hiện thị số phần trăm lên màn hình.



Hình 4.54 Màn hình loading

4.12 Tối ưu game

Quality settings:

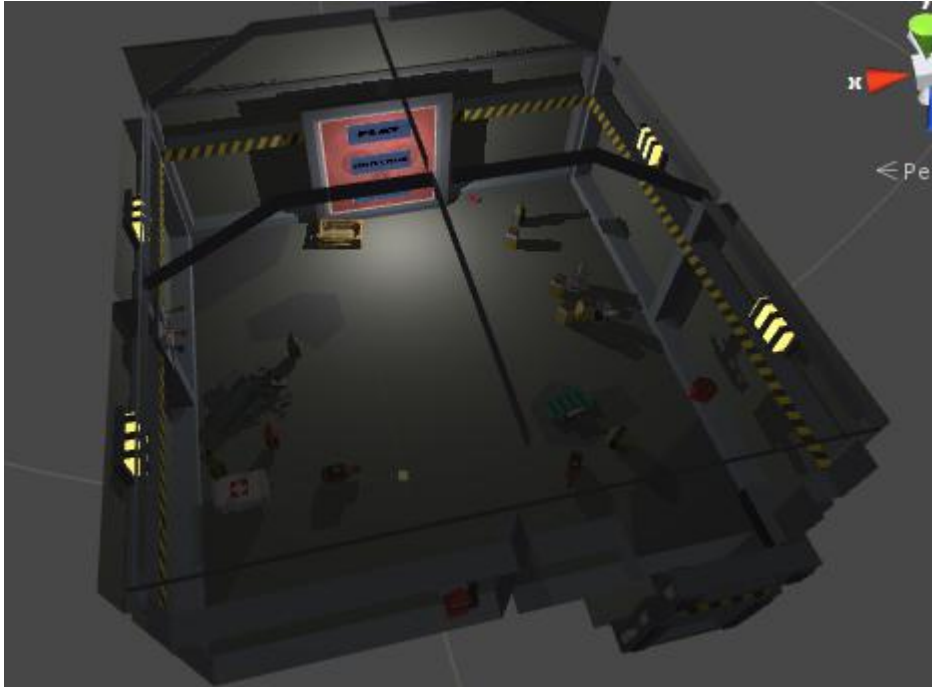


Hình 4.55 Quality setting

Trong quality settings, chúng ta có thể chọn được chất lượng game xuất ra theo các levels có sẵn hoặc tùy chỉnh theo ý mình. Trong đó có các tùy chọn về shadows, rendering,...

Hiện tại game được chọn mức quality setting thấp nhất là Fastest để đảm bảo được hiệu suất cũng như ưu khi chạy trên mobile.

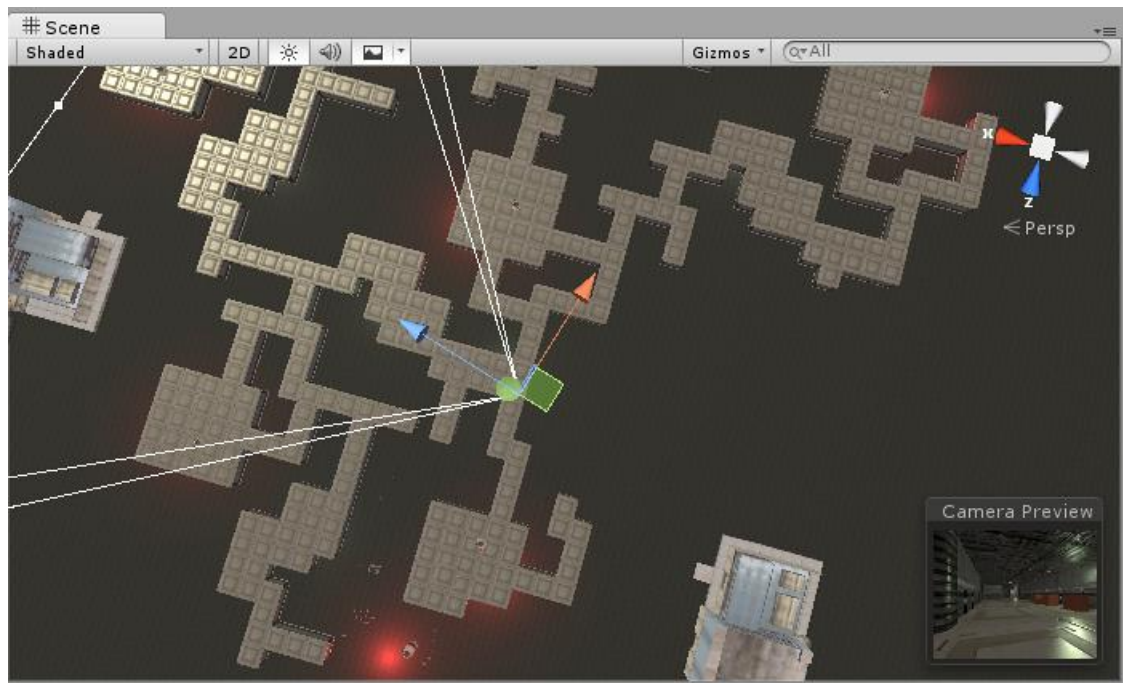
- Xóa một số model 3D và light trong scene để giảm dung lượng.



Hình 4.56 Giảm bớt model và light trong map

- Gắn Occlusion culling:

Occlusion Culling là tính năng vô hiệu hóa rendering các đối tượng hiện tại khi chúng không được nhìn thấy bởi camera vì chúng bị che khuất bởi đối tượng khác. Điều này không xảy ra tự động trong đồ họa máy tính 3D vì hầu hết các vật thể xa nhất từ máy ảnh được vẽ trước và các đối tượng gần hơn được vẽ trên đầu của chúng (điều này được gọi là "overdraw"). Occlusion Culling khác với Frustum Culling. Frustum Culling chỉ vô hiệu hóa hiển thị các đối tượng mà chúng nằm ngoài tầm nhìn camera nhưng không vô hiệu hóa anything ẩn thì tầm nhìn bởi overdraw. Lưu ý rằng khi bạn sử dụng Occlusion Culling bạn sẽ vẫn được hưởng lợi từ Frustum Culling.



Hình 4.57 Occulision culling

Một cấp độ trong nhà giống như mê cung. Chế độ xem cảnh bình thường này hiển thị tất cả các đối tượng có thể nhìn thấy được.

CHƯƠNG 5. KẾT LUẬN, HƯỚNG PHÁT TRIỂN

5.1 Kết quả

5.1.1 Đối với đề tài

- Đã hoàn thiện gameplay cho một game bắn súng thực tế ảo.
- Người chơi có thể sử dụng súng hoặc gậy để tấn công enemy. Ngoài ra còn có các vật phẩm như tăng máu khi tiêu diệt được enemy.
- Thiết kế được 2 level chơi cho game.
- Kết nối thành công với tay cầm thông qua bluetooth giúp điều khiển game thú vị và dễ dàng hơn.

5.1.2 Đối với các thành viên trong nhóm

- Hiểu được quy trình phát triển game.
- Hiểu được cấu trúc Unity Engine và Google VR SDK.
- Biết cách xây dựng game 3D và thực tế ảo bằng Unity3d.
- Kỹ năng quản lý dự án, quản lý thời gian được nâng cao.
- Ứng dụng được các kiến thức đã tìm hiểu được từ Unity, game design vào việc thực hiện đồ án.
- Nâng cao khả năng làm việc nhóm.

5.2 Hạn chế

- Chỉ có 1 vật phẩm tăng máu.
- Ít level chơi, chỉ có 2 level.

5.3 Hướng phát triển

- Thiết kế thêm level.
- Tăng số lượng enemy, boss.
- Tạo thêm các vật phẩm nhặt được khi tiêu diệt được enemy.
- Tạo khả năng đánh từ cho cho enemy.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng anh

- [1] Jeremy Gibson Bond, Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#, 1st Edition (2014)
- [2] Ray Barrera, Unity AI Game Programming - Second Edition, 2nd Edition (2015)
- [3] Francesco Sapio, Unity UI Cookbook (2015)
- [4] Y. Antonio Goncalves Vilas Boas, “Overview of Virtual Reality Technologies.” .

Tài liệu tham khảo từ internet

- [1] Game Programming Patterns < <https://gameprogrammingpatterns.com> >
- [3] Unity Document <<https://docs.unity3d.com/Manual/index.html>>
- [4] Unity Community < <https://unity3d.com/community>>
- [5] Unity Knowledge Base < <https://support.unity3d.com/hc/en-us>>
- [6] Wikipedia, “Unity (game engine).” [Online]. Available: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)).