

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG MÔ PHỎNG
CĂN HỘ CAO TẦNG HOẶC KHU BIỆT
THỰC BẰNG THỰC TẾ ẢO**

Người hướng dẫn: ThS VŨ ĐÌNH HỒNG

Người thực hiện: TRẦN GIA BẢO - 51303235

TRẦN ĐỨC TRỌNG - 51303196

Lớp: 13050301

Khoa: 17

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

**XÂY DỰNG ỦNG DỤNG MÔ PHỎNG
CĂN HỘ CAO TẦNG HOẶC KHU BIỆT
THỰ BẰNG THỰC TẾ ẢO**

Người hướng dẫn: ThS VŨ ĐÌNH HỒNG

Người thực hiện: TRẦN GIA BẢO - 51303235

TRẦN ĐỨC TRỌNG - 51303196

Lớp: 13050301

Khoa: 17

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017

LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn Quý Khoa và Quý Nhà Trưởng đã đồng ý duyệt đề tài luận văn của chúng em, tạo cơ hội cho chúng em có thể tìm hiểu và nghiên cứu đề tài. Đồng thời chúng em cũng xin gửi lời cảm ơn tới thầy Vũ Đình Hồng, là giảng viên hướng dẫn của chúng em, đã luôn theo dõi và luôn đưa ra chúng em những lời khuyên, hướng giải quyết các vấn đề.

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

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

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

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

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Trần Gia Bảo

Trần Đức Trọng

TÓM TẮT

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

LỜI CẢM ƠN	iii
TÓM TẮT	v
MỤC LỤC.....	1
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	5
DANH MỤC CÁC BẢNG BIÊU, HÌNH VẼ, ĐỒ THỊ	6
CHƯƠNG 1 – TÌM HIỂU CÔNG NGHỆ THỰC TẾ ẢO (VIRTUAL REALITY)....	11
1.1 Chức năng của VR	11
1.1.1 Tương tác thời gian thật.....	11
1.1.2 Tương tác ngũ giác	12
1.1.3 Nâng cao trí tưởng tượng con người.....	12
1.2 Giới thiệu về VR	12
1.2.1 Khái niệm.....	12
1.2.2 Đặc tính.....	12
1.3 Lịch sử phát triển	13
1.3.1 Trước năm 1950.....	13
1.3.2 Giai đoạn 1950 – 1970.....	14
1.3.3 Giai đoạn 1970 – 1990.....	15
1.3.4 Giai đoạn 1990 – 2000.....	18
1.3.5 Giai đoạn 2000 – hiện tại.....	21
1.4 Thiết bị thực tế ảo	24
1.4.1 Cấu tạo	24
1.4.1.1 Phần cứng.....	24
1.4.1.2 Phần mềm	27
1.4.2 Nguyên lý hoạt động	27
1.4.2.1 Phần cứng.....	27
1.4.2.2 Phần mềm	30

1.4.3 Các cách tương tác trên VR	35
1.4.3.1 Hệ thống tự động	36
1.4.3.2 Tay cầm và hệ thống cảm biến	39
CHƯƠNG 2 – MÔ PHỎNG ĐẠI HỌC TÔN ĐỨC THẮNG BẰNG PHẦN MỀM BLENDER	43
2.1 Blender là gì	44
2.2 Lịch sử phát triển	45
2.2.1 Khởi đầu không thuận lợi	45
2.2.2 Sự cố gắng không ngừng	45
2.2.3 Sự hồi sinh của Blender	45
2.3 Tính năng	46
2.4 Giao diện người dùng.....	47
2.4.1 Hai chế độ chính	47
2.4.1.1. Object Mode	47
2.4.1.2. Edit Mode	47
2.4.2 Phím tắt	48
2.4.2.1 Phím tắt chung	48
2.4.2.2 Phím tắt dành riêng cho edit mode	51
2.4.2.3 Phím tắt dành riêng cho object mode	51
2.4.3 Chuyển động camera.....	52
2.4.4 Modifiers.....	52
2.4.4.1 Array	53
2.4.4.2 Boolean	54
2.4.4.3 Edge Split.....	55
2.4.4.4 Mirror	56
2.4.4.5 Subdivision Surface	57
2.4.5 Material và Texture.....	58

2.4.5.1 Material	58
2.4.5.2 Texture	62
2.5 Quá trình xây dựng đồ án.....	67
2.5.1 Đo đạc số liệu.....	67
2.5.2 Những đối tượng đầu tiên	67
2.5.3 Quy trình mô phỏng toà nhà	68
2.5.4 Những khó khăn gặp phải và cách khắc phục	68
2.6 Một vài ví dụ	73
CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG BẰNG UNITY	76
3.1 Giới thiệu về Unity.....	76
3.1.1 Tổng quan	76
3.1.2 Lịch sử phát triển	77
3.1.3 So sánh Unity với các game engine khác	78
3.1.4 Các phiên bản của Unity	79
3.2 Làm quen với Unity	80
3.2.1 Giao diện người dùng	80
3.2.2 Một số khái niệm	83
3.2.3 Các thao tác cơ bản	84
3.3 VR trên Unity	94
3.3.1 Oculus	94
3.3.2 Thư viện Google VR SDK.....	96
3.3.2.1 DayDream	96
3.3.2.2 Cardboard.....	98
3.3.2.3 Sử dụng thư viện Google SDK trên Unity.....	100
3.4 Áp dụng mô hình trường TĐT vào Unity	105
3.4.1 Thiết kế chức năng.....	105
3.4.2 Xây dựng ứng dụng.....	107

3.4.2.1 AutoWalk - Chức năng di chuyển	108
3.4.2.2 OpenDoor – Chức năng mở cửa	109
3.4.2.3 Reticle	111
3.4.2.4 ShowMenu, Slider và ChangeScene	112
3.4.3 Một số khó khăn gặp phải	113
CHƯƠNG 4 – KẾT QUẢ ĐẠT ĐƯỢC	114
4.1 Hình ảnh ứng dụng.....	114

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

f *Tần số của dòng điện và điện áp (Hz)*

p *Mật độ điện tích khói (C/m³)*

CÁC CHỮ VIẾT TẮT

VR Vitural Reality

HMD Head-mounted display

AR Augmented reality

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1 – Đặc tính 3I của VR.....	13
Hình 2 - Cỗ máy chiếu phim Sensorama	14
Hình 3 – Hệ thống thật tế ảo HMD đầu tiên	14
Hình 4 - Hệ thống thực tế ảo Aspen Movie Map.....	15
Hình 5 – Thiết bị DataGlove và EyePhone.....	16
Hình 6 – Hệ thống VR tạo cú hit năm 1990.....	17
Hình 7 - Thiết bị HMD Sega VR	18
Hình 8 - Trò chơi thật tế ảo đầu tiên có chế độ multiplayer	18
Hình 9 – Căn phòng hình khối thật tế ảo	19
Hình 10 – Trò chơi thật tế ảo Second Life.....	20
Hình 11 – Thiết bị hỗ trợ VR – Virtuix Omni	21
Hình 12 – Kính thật tế ảo của Google – Cardboard.....	22
Hình 13 – Trò chơi Minecraft phiên bản thật tế ảo.....	23
Hình 14 – Cấu tạo kính VR.....	24
Hình 15 - Kính thật tế ảo của SamSung – Gear VR	26
Hình 16 – Con quay hồi chuyển.....	26
Hình 17 - Nguyên lý 3D side by side.....	28
Hình 18 – Điều chỉnh tiêu cự kính	28
Hình 19 - Hai luồng video được render lệch nhau.....	29
Hình 20 – Video 360 độ có thể xem dưới nhiều góc độ	30
Hình 21 – Camera 360 độ	31
Hình 22 – Hình ảnh được chụp từ Camera 360 độ	31
Hình 23 – Các điểm đầu và cuối của ảnh khớp nhau.....	32
Hình 24 – Thiết bị nhận một phần hình ảnh	32
Hình 25 - Kéo bức ảnh để xem nhiều góc độ khác	33

Hình 26 – Hình ảnh được biểu diễn dưới dạng hình cầu	33
Hình 27 – Hình được chụp bằng Camera với 6 ống kính	34
Hình 28 – Các điểm khớp nhau trên bức ảnh.....	34
Hình 29 – Kết quả thu được khi nối các điểm với nhau	35
Hình 30 - Ứng dụng VR hiển thị Menu khi nhân vật nhìn xuống đất	36
Hình 31 – Mô phỏng về Reticle của VR	37
Hình 32 – Ví dụ về Reticle trên VR.....	37
Hình 33 – Game Zombie Shooter sử dụng chức năng AutoShoot.....	38
Hình 34 – Game MyRollBall sử dụng Reticle để tương tác	38
Hình 35 – Tay cầm Console.....	39
Hình 36 – Tay cầm bluetooth một tay.....	39
Hình 37 - Sử dụng Reticle kết hợp thao tác Click để chọn Menu	40
Hình 38 – Thiết bị tay cầm cảm biến có tên Oculus Touch.....	40
Hình 39 – Thiết bị Leap Motion	41
Hình 40 – Thiết bị VR Gloves	41
Hình 41 – Mô phỏng thiết bị Horus VR Suit	42
Hình 42 – Thiết bị Horus VR Suit	42
Hình 43 - Một số phần mềm vẽ 3D.....	43
Hình 44 – Icon phần mềm Blender	44
Hình 45 – Modifier trong Blender	52
Hình 46 – Sử dụng Modifier Array (P1)	53
Hình 47 – Sử dụng Modifier Array (P2)	53
Hình 48 – Sử dụng modifier Boolean	54
Hình 49 – Sử dụng modifier Edge Split.....	55
Hình 50 – Sử dụng modifier Mirror.....	56
Hình 51 – Sử dụng modifier Subdivision Surface	57
Hình 52 – Sử dụng material (P1)	58

Hình 53 – Sử dụng material (P2)	59
Hình 54 – Sử dụng material (P3)	59
Hình 55 – Sử dụng material (P4)	60
Hình 56 – Sử dụng material (P5)	60
Hình 57 – Sử dụng material (P6)	61
Hình 58 – Sử dụng material (P7)	61
Hình 59 – Sử dụng texture (P1)	62
Hình 60 – Sử dụng texture (P2)	63
Hình 61 – Sử dụng texture (P3)	63
Hình 62 – Sử dụng texture (P4)	64
Hình 63 – Sử dụng texture (P5)	64
Hình 64 – Sử dụng texture (P6)	65
Hình 65 – Sử dụng texture (P7)	65
Hình 66 – Sử dụng texture (P8)	66
Hình 67 – Sử dụng texture (P9)	66
Hình 68 – Những đối tượng đầu tiên để đặt tiêu chuẩn kích thước	67
Hình 69 – Hình ảnh tượng đá nghệ thuật thật tế và mô phỏng	73
Hình 70 – Quá trình mô phỏng tượng đá nghệ thuật	73
Hình 71 – Hình ảnh nhà mái vòm tròn thật tế và mô phỏng	74
Hình 72 – Quá trình mô phỏng nhà mái vòm tròn	74
Hình 73 – Hình ảnh ghế ngồi thật tế và mô phỏng	75
Hình 74 – Quá trình mô phỏng ghế ngồi	75
Hình 75 – Giao diện khởi động	80
Hình 76 – Project được lưu trữ trên web	80
Hình 77 – Các layout cơ bản	81
Hình 78 – Các GameObject trên Scene	84
Hình 79 – Tùy chỉnh thuộc tính của GameObject	85

Hình 80 – Quản lý thư mục Project	85
Hình 81 – Tùy chỉnh màu sắc cho GameObject	86
Hình 82 – Tùy chỉnh Texture cho GameObject.....	87
Hình 83 – GameObject đã được tùy chỉnh màu sắc và texture.....	87
Hình 84 – Một số loại Texture	88
Hình 85 – Các Component của GameObject	89
Hình 86 – GameObject chịu ảnh hưởng vật lý sau khi gắn Component	89
Hình 87 – Xây dựng Script cho GameObject	90
Hình 88 – Trình Editor và IDE Visual Studio để viết Script	90
Hình 89 – Hai Object được vẽ bằng công cụ Blender	91
Hình 90 – Sử dụng GameObject từ Blender vào Unity	91
Hình 91 – Các nền tảng có thể build trên Unity.....	92
Hình 92 - Ứng chạy chạy trên nền tảng Web OpenGL.....	92
Hình 93 - Ứng dụng chạy trên nền tảng PC Window	93
Hình 94 - Ứng dụng chạy trên nền tảng Android	93
Hình 95 – Tính năng VR được tích hợp trên Unity	94
Hình 96 – Game VR xây dựng bằng Unity.....	95
Hình 97 – Chế độ DayDream trên Android N	97
Hình 98 – Chế độ DayDream kết hợp với tay cầm điều khiển	97
Hình 99 – Demo Cardboard trên Unity.....	98
Hình 100 – Thiết bị và thư viện Cardboard của Google	99
Hình 101 – Trang tải thư viện SDK	100
Hình 102 – Cấu trúc thư viện Google SDK	101
Hình 103 – Hai Camera đã được xây dựng sẵn	102
Hình 104 – Cấu trúc Prefab của GvrMain	102
Hình 105 – Camera được thiết kế bằng GvrViewerMain	103
Hình 106 – Cấu trúc Prefab được sinh tự động khi Play	103

Hình 107 – Mô phỏng thao tác xoay người	104
Hình 108 – Mô phỏng thao tác nghiên đầu	105
Hình 109 – Mô hình trường TDT trên Unity	107
Hình 110 – Thuộc tính Character Controller	108
Hình 111 – Đánh dấu Tag cho nhân vật là Player	109
Hình 112 – Mỗi cánh cửa sẽ có thông số góc đóng/mở cửa khác nhau.....	109
Hình 113 – Thư viện SDK cung cấp hỗ trợ Reticle	111
Hình 114 – Reticle trong ứng dụng.....	111
Hình 115 – ShowMenu, ClickButton, Slider và ChangeScene	112

DANH MỤC BẢNG

Bảng 1 – Các phím tắt chung của Blender.....	50
Bảng 2 – Các phím tắt trong chế độ edit mode của Blender.....	51
Bảng 3 – Các phím tắt trong chế độ object mode của Blender.....	51
Bảng 4 – Các phương thức di chuyển camera trong Blender	52
Bảng 5 – Những khó khăn và cách khắc phục	72
Bảng 6 – Những khó khăn trong Unity	113

CHƯƠNG 1 – TÌM HIỂU CÔNG NGHỆ THỰC TẾ ẢO (VIRTUAL REALITY)

Thực Tế Ảo (Virtual Reality - VR) là một công nghệ mạnh mẽ với khả năng ứng dụng rộng rãi trong mọi lĩnh vực (nghiên cứu, công nghiệp, giáo dục, đào tạo, thương mại, giải trí,...), VR hứa hẹn sẽ mang đến thay đổi đời sống của chúng ta. VR có thể được xem như là chiếc thang đưa ta đến những giấc mơ, nơi ta thức dậy có thể sử dụng ma thuật, bay vào vũ trụ hoặc có những khả năng đặc biệt khác mà thế giới thật tại không có.

1.1 Chức năng của VR

Vì sao không bắt đầu với khái niệm trước ?

Để nói về VR, không có một khái niệm nào là hoàn toàn chính xác, chính vì sự ứng dụng rỗng rãi của VR mà từ đó cũng xuất hiện rất nhiều khái niệm về nó thông qua các phương tiện truyền thông, các buổi hội thảo, các bài báo, gây ngộ nhận cho người muốn tiếp cận với VR. Vậy ta sẽ tiếp cận với VR thông qua khía cạnh chức năng trước.

VR là một hệ thống mô phỏng, trong đó đồ họa máy tính bằng cách kích thích các giác quan nhân tạo sẽ đánh lừa cơ thể của chúng ta để tạo ra một thế giới ảo “như thật” và buộc tâm trí ta phải chấp nhận một phiên bản khác ở thế giới đó.

1.1.1 Tương tác thời gian thực

Vì là “như thật” nên VR trang bị cho mình khả năng tương tác thời gian thực (Real-time Interactivity), tức là nhận biết được tín hiệu của người sử dụng và phản ứng lại, tạo sự thay đổi trong 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, khiến họ đắm chìm (Immersion) vào nó như thể họ đang sống trong đó.

1.1.2 *Tương tác ngũ giác*

Ở thế giới thật, ta không những nhìn thấy (thị giác) đối tượng đồ hoạ ở dạng 3D nổi mà còn có thể tiếp xúc và cảm nhận (xúc giác), nghe (thính giác) cũng như ngửi (khứu giác) và nếm (vị giác) của nó. Để mang lại cảm giác “như thật”, VR cũng đã và đang mô phỏng lại các giác quan này để mang lại cảm xúc chân thật nhất.

1.1.3 *Nâng cao trí tưởng tượng con người*

VR còn giúp con người phá bỏ giới hạn để đến với những điều mới mẻ, điên rồ hơn mà cuộc sống thường ngày không thể thực hiện được. Từ những trí tưởng tượng (Imagination) đó sẽ nảy sinh các sáng kiến, phát minh trong đời sống thường ngày.

1.2 Giới thiệu về VR

1.2.1 *Khái niệm*

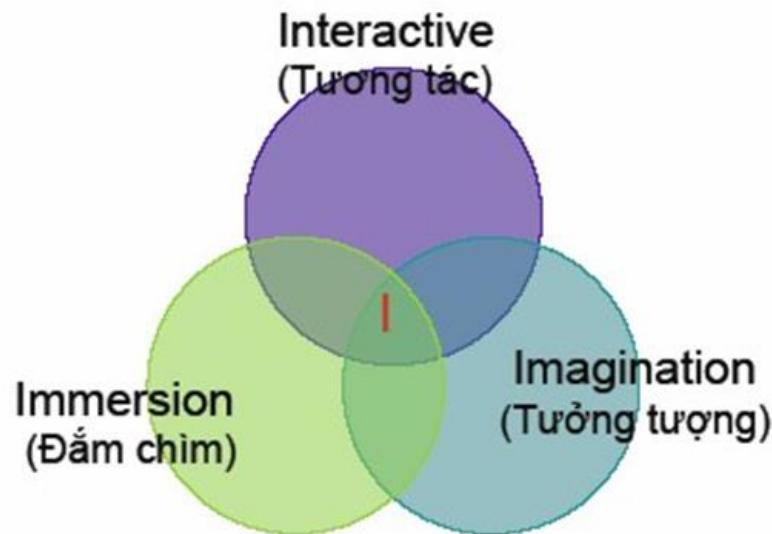
Vậy, từ các phân tích trên về chức năng của VR, ta có thể xem định nghĩa của C. Burdea và P. Coiffet về VR là tương đối chính xác: “VR - Thực tế ảo là một hệ thống giao diện cấp cao giữa người sử dụng và máy tính. Hệ thống này mô phỏng các sự vật và hiện tượng theo thời gian thực và tương tác với người sử dụng qua tổng hợp các kênh cảm giác. Đó là ngũ giác gồm: thị giác, thính giác, xúc giác, khứu giác, vị giác”.

1.2.2 *Đặc tính*

Ta còn có thể định nghĩa VR thông qua các đặc tính của nó. Đặc tính của VR dựa vào 3 yếu tố “I” bắt nguồn từ các từ trong tiếng anh:

- Immersion: sự đắm chìm, cho phép ta đứng trong một thế giới thật trong màn hình máy vi tính, cảm nhận bằng giác quan, di chuyển theo mệnh lệnh chính mình chứ không như tương tác thông thường từ chuột và bàn phím máy tính
- Interactive: sự tương tác, đặc tính quan trọng nhất của VR, yếu tố khiến người sử dụng cảm thấy mình đang được sống trong thế giới ảo.

- Imagination: sự tưởng tượng, là đặc tính ít được người biết đến. Đưa ra các quyết định hoặc giải pháp cho vấn đề, khiến sự mô phỏng trở nên hữu thích hơn, tất cả điều đó đều phụ thuộc vào sự tưởng tượng.



Hình 1 – Đặc tính 3I của VR

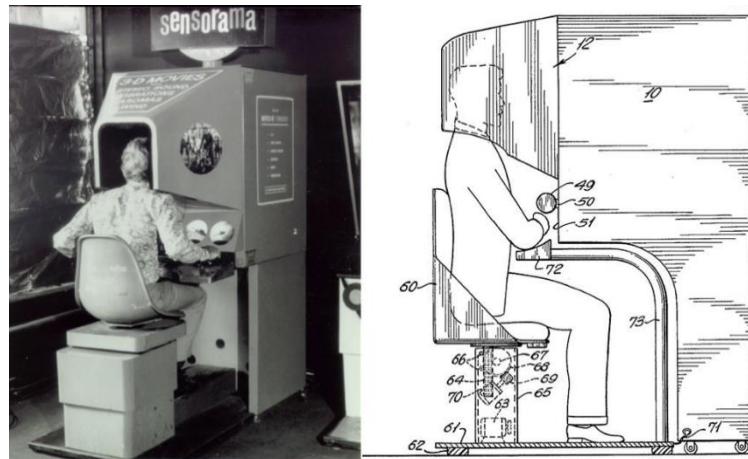
1.3 Lịch sử phát triển

1.3.1 Trước năm 1950

Trang đầu tiên của ngành công nghệ thực tế ảo bắt nguồn từ một tập truyện ngắn khoa học viễn tưởng của Stanley G. Weinbaum, xuất bản năm 1935 với tiêu đề “Pygmalion's Spectacles”. Nội dung của câu chuyện xoay quanh một hệ thống thực tế ảo hình dáng chiếc kính (goggle) với việc ghi nhận các trải nghiệm giả tưởng ba chiều, bao gồm khứu giác và xúc giác.

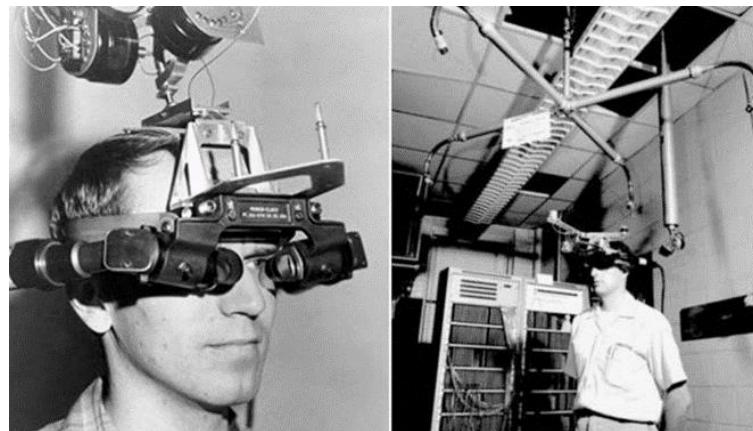
1.3.2 Giai đoạn 1950 – 1970

Vào những năm 1950, nhà quay phim Morton Heilig đã cho ra mắt ý tưởng về một cỗ máy chiếu phim có tên Sensorama, giúp người xem có thể trải nghiệm 5 thước phim ngắn bằng cả 5 giác quan thông qua các thiết bị trong cỗ máy gồm: một màn hình thực thể kính, quạt, máy tạo mùi, loa âm thanh và 1 chiếc ghế có thể di chuyển được.



Hình 2 - Cỗ máy chiếu phim Sensorama

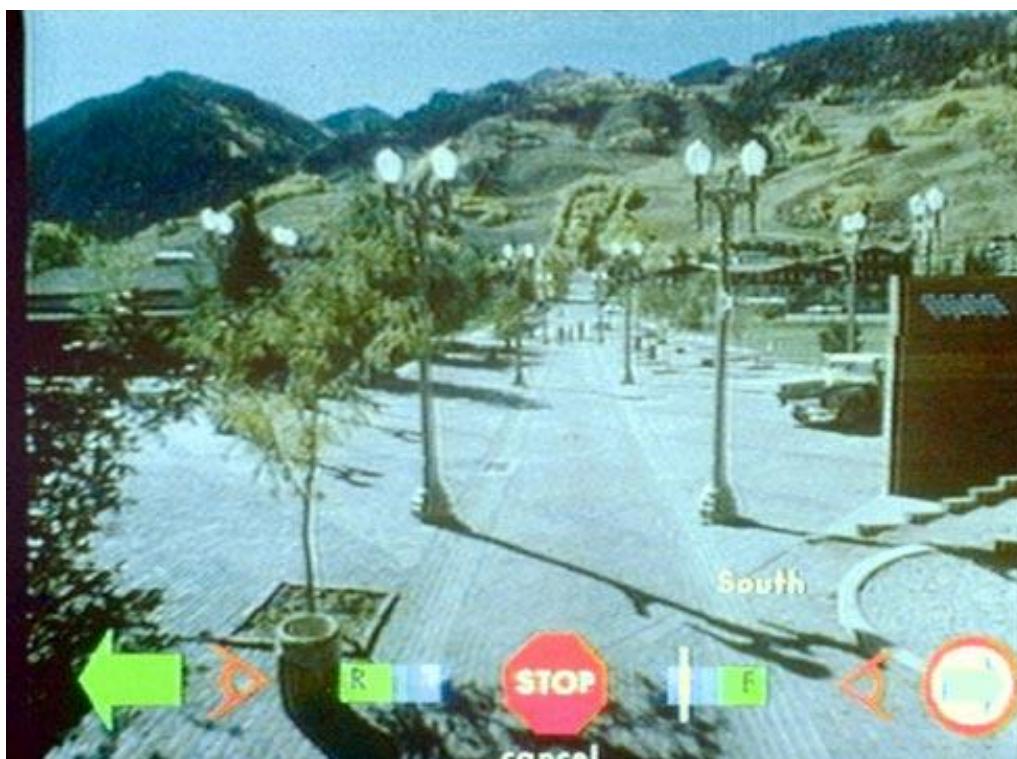
Năm 1968, Ivan Sutherland, với sự hỗ trợ của người học trò Bob Sproull đã chế tạo ra một hệ thống hiển thị gắn trên đầu (HMD) thực tế ảo và thực tế tăng cường (AR) đầu tiên, The Sword of Damocles, tuy nhiên nó quá nặng đên mức phải được treo lên trần nhà. Đồ họa bên trong các môi trường ảo là mô hình khung lưới đơn giản.



Hình 3 – Hệ thống thật tế ảo HMD đầu tiên

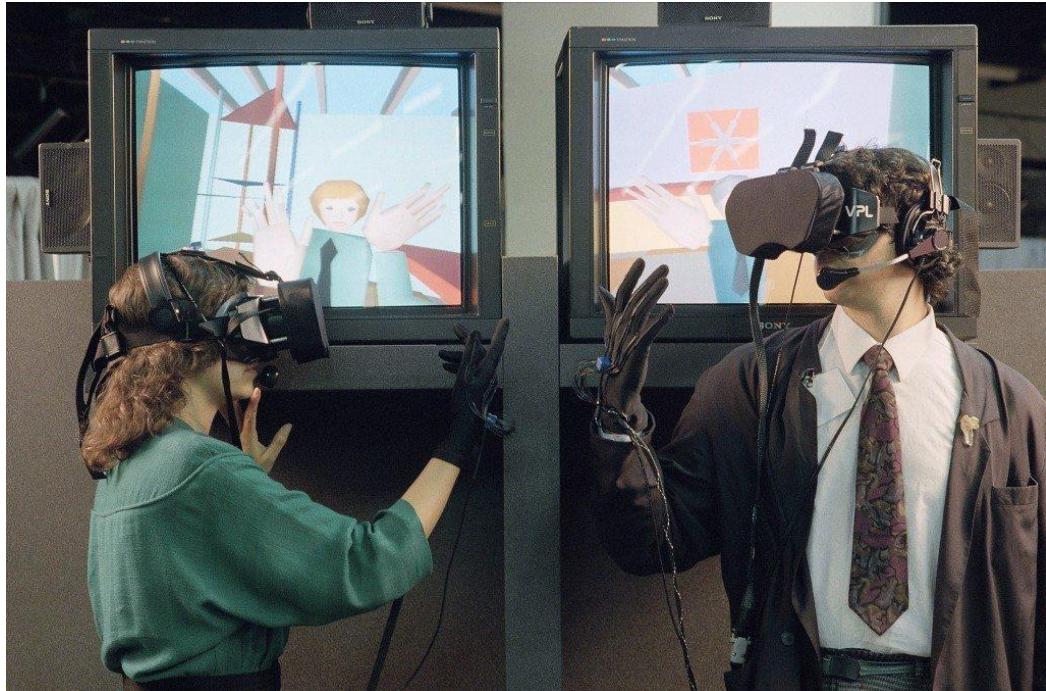
1.3.3 Giai đoạn 1970 – 1990

Cũng đáng chú ý vào những buổi đầu của hệ thống siêu truyền và hệ thống thực tế ảo là Aspen Movie Map, tạo bởi MIT vào năm 1978. Đây là chương trình mô phỏng lại thành phố Aspen ở Colorado, trong đó người sử dụng có thể đi dạo đường phố trong 3 chế độ: mùa hè, mùa đông và đa giác. Mùa hè và mùa đông thật chất chỉ là những hình ảnh được chụp lại từ những khoảnh khắc các mùa trong năm rồi ghép vào những khung hình của người xem. Và chế độ đa giác là một mô hình 3D cơ bản của thành phố Aspen.



Hình 4 - Hệ thống thực tế ảo Aspen Movie Map

Vào những năm 1980, thuật ngữ “Virtual Reality – thật tế ảo” đã được phổ biến bởi Jason Lanier, một trong những nhà tiên phong hiện đại của lĩnh vực này. Thành lập công ty nghiên cứu VPL vào năm 1985, VPL đã phát triển một số thiết bị VR như DataGlove, EyePhone và AudioSphere (lược dịch: găng tay dữ liệu, điện thoại Eye và âm thanh vòm).



Hình 5 – Thiết bị DataGlove và EyePhone

Trong những năm 1980, thực tế ảo vẫn chưa thật sự nổi tiếng. Hầu hết sự phổ biến của VR đến từ các nền văn hóa bên lề, như Cyberpunks (một nhánh nhỏ của khoa học viễn tưởng về những điều xảy ra trong tương lai), xem công nghệ này như một phương tiện tiềm năng cho sự thay đổi của xã hội, và nhóm văn hóa sử dụng chất kích thích, những người ca ngợi thực tế ảo không chỉ là một hình thức nghệ thuật mới, mà còn là một chân trời mới. Một vài người sử dụng chất kích thích trong khi sử dụng công nghệ thực tế ảo. Các khái niệm về thực tế ảo được phổ biến rộng rãi trong phương tiện truyền thông đại chúng bởi những bộ phim như Brainstorm (1983) và The Lawnmower Man.

Các nghiên cứu về VR thật sự bùng nổ vào những năm 1990 khi cuốn sách phi tiểu thuyết với nhan đề Virtual Reality được cho ra mắt vào năm 1991 bởi Howard Rheingold. Cuốn sách phục vụ việc làm sáng tỏ thực tế ảo là gì, giúp các nhà nghiên cứu hoặc những người đam mê khoa học dễ tiếp cận hơn với thực tế ảo.

Từ khi nhận được sự thu hút từ phía phương tiện truyền thông, thậm chí đã có một vài so sánh giữa sự đổi mới của VR với phát minh tiên phong của Wright Brothers trong lĩnh vực hàng không. Vào năm 1990, Jonathan Waldern, tiến sĩ chuyên nghiên cứu về công nghệ thực tế ảo, đã phát triển đầy đủ một hệ thống tích hợp VR và cho ra mắt tại cuộc triển lãm ở cung điện Alexandra thuộc Luân Đôn. Hệ thống này thông qua thiết bị đeo đầu, làm đắm chìm người sử dụng vào thế giới ảo.



Hình 6 – Hệ thống VR tạo cú hit năm 1990

CyberEdge và PCVR, hai tạp chí công nghệ về thực tế ảo, bắt đầu xuất bản vào đầu những năm 1990. Tuy nhiên, hầu hết mọi ý tưởng về VR vẫn còn mang nặng tính lý thuyết do sự giới hạn sức mạnh máy tính tại thời điểm này. Chi phí quá đắt đỏ của ngành công nghệ này đã khiến cho người tiêu dùng khó đón nhận được nó. Khi Internet trở nên phổ biến rộng rãi, VR trở thành công nghệ mà đối tượng sử dụng có thể là bất cứ ai, chủ yếu tập trung vào y tế, mô phỏng các chuyến bay, công nghiệp thiết kế ô tô và mục đích huấn luyện quân sự trong giai đoạn 1970 đến 1990.

1.3.4 Giai đoạn 1990 – 2000

Vào năm 1991, Sega (công ty trò chơi điện tử dịch vụ Nhật Bản) đã cho ra mắt thiết bị đeo đầu Sega VR và bộ thiết bị chơi game Mega Drive. Thiết bị này sử dụng màn hình LCD với một tấm che, tai nghe stereo và cảm biến quán tính cho phép hệ thống theo dõi và phản ứng với chuyển động phần đầu của người dùng.



Hình 7 - Thiết bị HMD Sega VR

Cũng vào năm 1991, trò chơi điện tử Virtuality được tung ra và trở thành một hệ thống thực tế ảo giải trí đầu tiên được sản xuất hàng loạt cho phép kết nối mạng và nhiều người chơi. Virtuality được phát hành tại nhiều quốc gia với giá lên đến \$73,000 cho mỗi bộ thiết bị của Virtuality, bao gồm thiết bị đeo đầu và găng tay khung xương ngoài đưa đến trải nghiệm nhập vai đầu tiên cho người dùng.



Hình 8 - Trò chơi thật tế ảo đầu tiên có chế độ multiplayer

Antonio Medina, tốt nghiệp viện công nghệ MIT và đồng thời là nhà khoa học của NASA, đã thiết kế một hệ thống thực tế ảo mang tên Computer-Simulated Teleoperation (lược dịch: hệ thống điều khiển từ xa thông qua mô phỏng máy tính) để “lái xe” trên bề mặt sao Hỏa từ trái đất với thời gian thực (mặc dù có một sự chậm trễ đáng kể do đường truyền tín hiệu từ sao Hỏa đến trái đất).

Đặc biệt là sự ra đời của căn phòng nhập vai (immersive room) hình khói đầu tiên của Carolina Cruz-Neira, Daniel J. Sandin và Thomas A. DeFanti để mang lại sự mới mẻ, thay thế những cặp kính VR trước đó.



Hình 9 – Căn phòng hình khói thật tế ảo

Vào năm 1994, Apple cho ra mắt QuickTime VR, mặc dù sử dụng thuật ngữ “VR” nhưng nó hè là VR, cho phép hiển thị hình ảnh toàn cảnh 360 độ.

Một năm sau đó, nghệ sĩ Maurice Benayoun đã tạo ra nghệ thuật thực tế ảo đầu tiên kết nối thời gian thực của 2 châu lục: “Tunnel under the Atlantic - đường hầm dưới Đại Tây Dương” giữa trung tâm Pompidou ở Paris và bảo tàng nghệ thuật đương đại ở Montreal. Cũng năm 1995, một nhóm ở Seattle đã tạo ra một căn phòng 270 độ có tên Virtual Environment Theater, có cấu trúc gần giống với căn phòng hình khói đã đề cập trước đó. Đến rồi vào năm 1996, một hệ thống tương tự được Jim Barksdale trình diễn trong cuộc triển lãm thương mại tài trợ bởi Netscape Communications, lần đầu tiên một hệ thống thực tế ảo có kết nối Internet với nội dung của mạng lưới World Wide Web được nhúng trong mô hình thế giới ảo VRML 3D.

Năm 1999, doanh nhân Philip Rosedale cùng với công ty Linden Lab, đặt mục tiêu ban đầu tập trung vào phát triển phần cứng cho phép người sử dụng máy tính có thể trải nghiệm cảm giác đắm tröm trong một môi trường thật tế ảo 360 độ. Mục tiêu đó đã dẫn đến sự ra đời của trò chơi thế giới ảo trực tuyến nổi tiếng Second Life vào năm 2003.



Hình 10 – Trò chơi thật tế ảo Second Life

1.3.5 Giai đoạn 2000 – hiện tại

Năm 2007, Google ra mắt Street View, dịch vụ hiển thị tầm nhìn toàn cảnh sự già tăng các con số của các vị trí trên toàn thế giới như đường xá, công trình kiến trúc và các khu vực nông thôn. Đến năm 2010 nó được cho ra mắt chế độ 3D lập thể.

Cùng năm 2010, Palmer Luckey, một trong những nhà sáng lập của Oculus VR, thiết kế nguyên mẫu đầu tiên của Oculus Rift. Nguyên mẫu này được xây dựng trên vỏ của một tai nghe thật tế ảo, chỉ có khả năng theo dõi luân phiên. Tuy nhiên, nó đã thúc đẩy một lĩnh vực 90 độ mà trước đây chưa từng thấy ở các thị trường tiêu dùng tại thời điểm đó. Và thiết kế ban đầu này là bước đệm hình thành cho các thiết kế sau này.

Vào tháng 7/2013, Vendetta Online, trò chơi điện tử thẻ loại MMORPG đầu tiên của hãng Guild Software được hỗ trợ trên Oculus Rift, làm cho nó có khả năng trở thành thế giới trực tuyến kéo dài đầu tiên với sự hỗ trợ của một thiết bị thực tế ảo đeo đầu. Kể từ 2013, đã có một vài thiết bị thật tế ảo tìm cách thâm nhập vào thị trường để bổ sung cho Oculus Rift để nâng cao chất lượng trải nghiệm trò chơi. Trong số đó có Virtuix Omni, dựa trên khả năng di chuyển trong môi trường ba chiều thông qua một máy chạy bộ đa hướng.



Hình 11 – Thiết bị hỗ trợ VR – Virtuix Omni

Ngày 25/3/2014, Facebook mua lại Oculus VR với giá 2 tỷ dollar. Cùng trong tháng đó, Sony công bố dự án Morpheus, một thiết bị thật tế ảo đeo đầu cho thiết bị chơi trò chơi điện tử PlayStation 4. Và Google cũng đồng thời công bố Cardboard, một thiết bị xem thật tế ảo cho điện thoại thông minh mà người dùng có thể dễ dàng tự lắp ráp. Người dùng đặt điện thoại thông minh vào trong giá đỡ của Cardboard và đội Cardboard lên đầu như bao thiết bị VR khác.



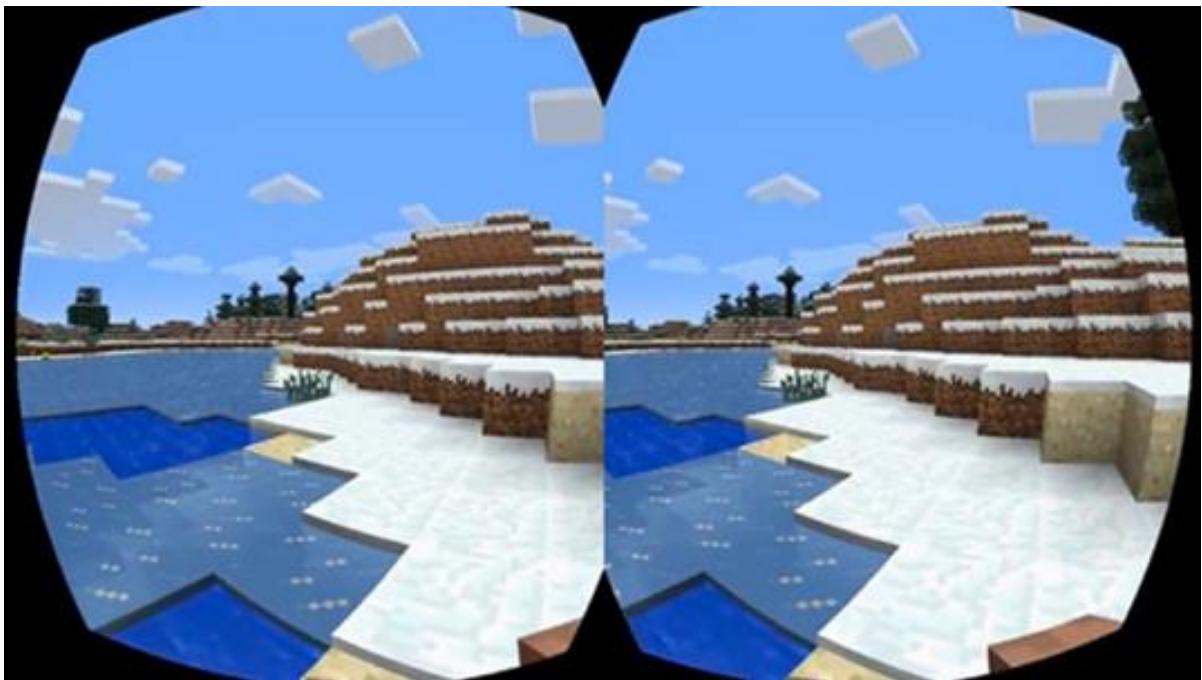
Hình 12 – Kính thật tế ảo của Google – Cardboard

Năm 2015, các chiến dịch Kickstarter cho găng tay Gloveone, cung cấp sự theo dõi chuyển động và phản hồi xúc giác đã nhận được sự tài trợ thành công lên tới hơn \$150.000. Giai đoạn tháng 2-3/2015, HTC hợp tác với công ty Valve và công bố thiết bị thật tế ảo đeo đầu của họ, HTC Vive, và các thiết bị điều khiển, bên cạnh công nghệ theo dõi chuyển động Lighthouse, giúp điều khiển “các trạm cơ sở” được gắn vào tường phía trên đầu người sử dụng trong các góc của căn phòng để theo dõi vị trí của thiết bị đeo đầu Vive và bộ điều khiển chuyển động bằng ánh sáng hồng ngoại. Vive đã được phát hành vào tháng 8/2016.

Vào tháng 7/2015, OnePlus trở thành công ty đầu tiên khởi chạy sản phẩm sử dụng công nghệ thật tế ảo. Họ sử dụng VR như một nền tảng để khởi chạy thiết bị chủ lực thứ hai của họ, OnePlus 2, đầu tiên cho phép sử dụng ứng dụng trên Google Play Store, sau đó là trên Youtube. Việc ra mắt đó có thể xem thông qua thiết bị OnePlus

Cardboard, được thiết kế dựa trên nền tảng Cardboard của Google. Toàn bộ việc ra mắt đó chạy trên công nghệ thật tế ảo với thời lượng 33 phút trên toàn thế giới.

Tháng 27/8/2016, Mojang (một nhóm lập trình viên game Minecraft) công bố rằng trò chơi Minecraft hiện có thể được chơi trên thiết bị Gear VR (thiết bị đeo đầu thật tế ảo, dành riêng cho điện thoại thông minh của SamSung). Minecraft hiện vẫn đang được phát triển cho thiết bị Oculus Rift nhưng một phiên bản riêng biệt đã được phát hành cho Oculus Store (giống Google Play Store hay App Store) để sử dụng với Gear VR. Phiên bản này cũng tương tự như bản Pocket Edition của Minecraft.



Hình 13 – Trò chơi Minecraft phiên bản thật tế ảo

1.4 Thiết bị thực tế ảo

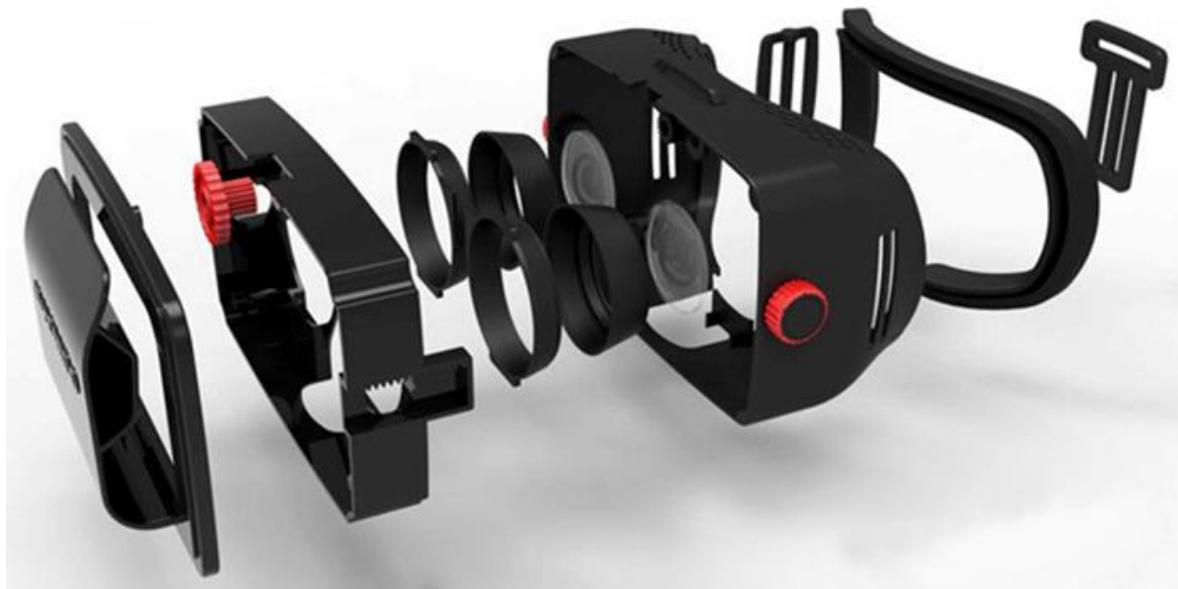
Đi kèm với sự phát triển của VR đến nay, đã xuất hiện rất nhiều sản phẩm hỗ trợ tạo cho con người cảm giác đắm chìm vào trong thế giới ảo như: găng tay dữ liệu, thiết bị tay cầm, các căn phòng hình khói, ... Nhưng sản phẩm được quan tâm nhất vẫn chính là thiết bị đeo đầu (HMD). Bỏ qua những chiếc kính của quá khứ, ta hãy cùng đi sâu vào mổ xé một chiếc kính thật tế ảo ở thời điểm hiện tại (2017).

1.4.1 Cấu tạo

Cấu tạo của một chiếc kính thật tế ảo bao gồm 2 thành phần chính là phần cứng và phần mềm.

1.4.1.1 Phần cứng

Các thiết bị đầu vào giúp tăng cường khả năng kích thích các giác quan của người dùng để mô phỏng môi trường thực tại ảo, ví dụ tai nghe âm thanh nổi, thiết bị cảm biến vị trí G-sensor, điều khiển Bluetooth... Các thiết bị đầu ra gồm màn hình hiển thị, bộ phản hồi cảm giác, xung lực... Đó chính là toàn bộ đầy đủ nhất của một chiếc kính thật tế ảo.



Hình 14 – Cấu tạo kính VR

Một chiếc kính đầy đủ các thành phần như thế sẽ mang trong mình một hệ thống thực tại ảo. Nghĩa là chỉ cần một chiếc kính là người dùng có thể đắm mình vào môi trường thật tế ảo và hoàn toàn chất lượng, đương nhiên chất lượng cũng tỷ lệ thuận với giá thành. Ví dụ như chiếc kính HoloLens của Microsoft với giá trị khoảng \$3,000.

Nhưng trên thị trường hiện nay, không phải chiếc kính nào cũng có đầy đủ tất cả các bộ phận như thế. Để đáp ứng khát vọng được trải nghiệm thực tế ảo đối với mọi tầng lớp trong xã hội, những chiếc kính giá “rẻ như bèo” đã lũ lượt ra đời. Cấu tạo của chúng chỉ đơn giản bao gồm:

- **Hệ thống giá đỡ có thể điều chỉnh cự ly:** để điều chỉnh tiêu cự thấu kính, tiêu cự giữa hai mắt mỗi người khác nhau, độ rộng màn hình điện thoại hiển thị cũng khác nhau nên hệ thống điều chỉnh tiêu cự là rất cần thiết.
- **Hệ thống thấu kính:** với các kính thực tế ảo thông thường bao gồm 2 thấu kính, đây là thành phần quan trọng nhất. Hình ảnh có rõ nét hay không, không gian mô phỏng có sống động không phụ thuộc đến 95% vào thấu kính. Do đó có thể nói chất lượng của kính thực tế ảo nằm ở thấu kính, các kính loại rẻ tiền có thấu kính làm không tốt dẫn đến ảnh sau thấu kính quá giàn và không trung thực làm hoa mắt và mỏi mắt. Một thấu kính chuẩn thường có kích thước đường kính từ 25mm tới 35mm và tiêu cự 45mm. Chất liệu của thấu kính thường bằng pha lê, nhựa, hay thủy tinh.
- **Hệ thống ốp mặt** và đầu hệ thống này gắn kính thực tế ảo ôm vào đầu, các kính thực tại ảo có trọng lượng nhẹ và hệ thống ốp thiết kế tốt sẽ giúp dùng kính lâu mà không mỏi cổ, đau đầu, gây nóng da mặt...

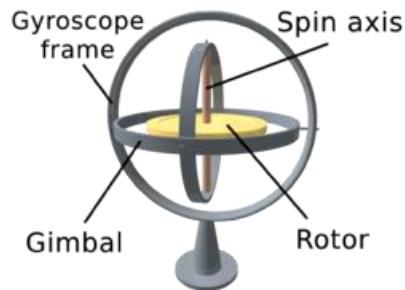
Hoặc đôi lúc những phiên bản biến thể cũng có thể đi kèm tai nghe. Nhưng điều đó cũng không quan trọng, vì các loại kính VR thông thường này luôn phải sử dụng đi kèm với các điện thoại thông minh có hỗ trợ con quay hồi chuyển, mượn các tính năng của điện thoại thông minh như tính năng hiển thị hình ảnh 3D side by side, tính năng cảm biến vị trí G-sensor, tính năng điều khiển... Điều đó cũng có nghĩa chất lượng hình

ảnh, âm thanh, độ mềm mượt đều phụ thuộc vào điện thoại, điện thoại càng mạnh càng tốt. Một vài ví dụ cho loại kính này đơn giản nhất chính là Cardboard của Google, hay mắc nhất hiện nay là Gear VR của SamSung.



Hình 15 - Kính thật tế ảo của SamSung – Gear VR

Nói thêm về con quay hồi chuyển trên các thiết bị. Con quay hồi chuyển là một thiết bị dùng để đo đặc hoặc duy trì phương hướng, dựa trên các nguyên tắc bảo toàn mô men động lượng. Thực chất, con quay cơ học là một bánh xe hay đĩa quay với các trục quay tự do theo mọi hướng.



Hình 16 – Con quay hồi chuyển

Con quay hồi chuyển được dùng kết hợp với gia tốc kế trong các thiết bị di động hiện đại ngày nay, đặc biệt là điện thoại thông minh và máy tính bảng. Gia tốc kế hỗ trợ việc tính toán gia tốc tuyến tính tương đối so với khung tham chiếu – hệ qui chiếu (frame of reference). Nó dùng để nhận biết thiết bị đang nằm ngang hay đang đứng, từ đó điều chỉnh khung hình thành chế độ portrait hoặc landscape và áp dụng vào các trò chơi cần cảm biến chuyển động như đua xe hoặc các trò chơi tương tác ảo (Wii).

1.4.1.2 Phần mềm

Nếu ta ví phần cứng của hệ thống thật té ảo giống như bộ phận cơ thể con người thì phần mềm chính là phần linh hồn, đóng vai trò quan trọng không kém. Nó chịu trách nhiệm cho việc quản lý các thiết bị nhập xuất của phần cứng, phân tích dữ liệu và tạo ra phản hồi thích hợp.

So sánh giữa các thiết bị thông thường như máy tính để bàn với hệ thống VR thì VR hoàn toàn phức tạp hơn nhiều, chúng yêu cầu phải được xử lý cực kỳ chính xác và gửi một lưu lượng lớn dữ liệu vào hệ thống. Hơn nữa, toàn bộ ứng dụng đều dựa trên thời gian thực, nên phần mềm cần phải quản lý nó: dữ liệu đầu vào phải được xử lý kịp thời và đáp ứng hệ thống để gửi tới màn hình đầu ra, không làm phá hủy cảm giác đắm chìm của người dùng.

Các ứng dụng này có thể được lập trình từ rất nhiều các ngôn ngữ khác nhau để mô hình hóa và mô phỏng các đối tượng. Hiện tại có rất nhiều các framework như AutoCAD, Blender, 3D Max, ... hay các engine cho game nổi tiếng như Unity, Unreal... phục vụ cho việc mô phỏng 3D.

1.4.2 Nguyên lý hoạt động

1.4.2.1 Phần cứng

Các loại kính thực tế ảo hiện có trên thị trường đều hoạt động theo nguyên lý 3D side by side: chia màn hình thành 2 khung hình, mỗi khung hình đáp ứng hình ảnh cho mỗi mắt. Khi đeo kính thực tế ảo, hai khung hình sẽ được hội tụ qua hệ thống thấu kính giúp hình ảnh chập lại và tạo ra độ nét khác nhau.



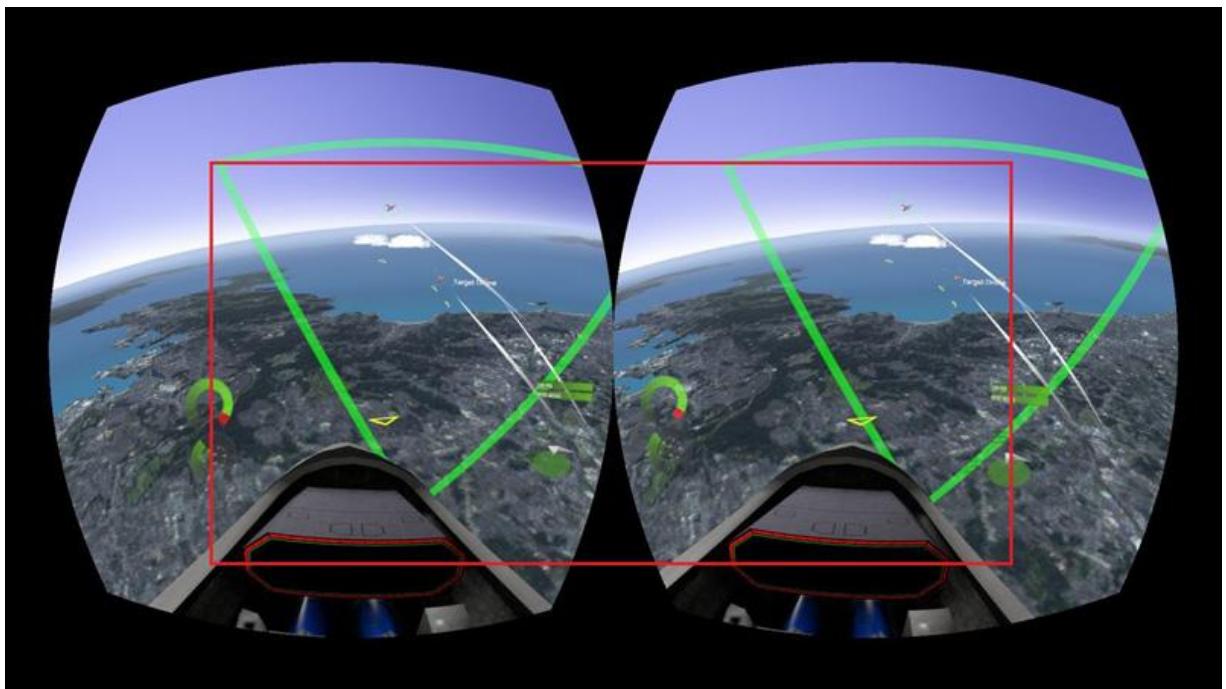
Hình 17 - Nguyên lý 3D side by side

Các màn hình chiếu hình ảnh thường rất sát với mắt, thường từ 10-15 cm tuy nhiên khi đeo kính vào ta lại thấy hình ảnh ở rất xa, giống như xem phim chiếu trên các phòng chiếu có kích thước lên đến cả trăm inch. Đây cũng là do hình ảnh sau thấu kính đã hội tụ ở điểm rất xa. Để ảnh ảo của vật sau khi qua thấu kính càng xa thì vật phải ở càng gần tiêu cự, tức là màn hình điện thoại phải ở vị trí cách mắt xấp xỉ gần 4cm. Khi đó ảnh ảo do thấu kính tạo ra sẽ ở xa vô cực và mắt sẽ nhìn thấy ảnh đó mà không cần điều tiết. Không gây nhức hay mỏi mắt.



Hình 18 – Điều chỉnh tiêu cự kính

Một điểm đặc biệt là mỗi luồng video hay hình ảnh của hai khung hình được render ở góc hơi lệch nhau. Bằng cách này, não sẽ bị đánh lừa và nghĩ rằng ảnh đang hiện trước mặt là ảnh 3D. Vì thực tế cả hai mắt sẽ không nhìn thấy một hình ảnh giống nhau mà mỗi mắt sẽ nhìn thấy hình ảnh khác nhau. Có thể thử ngay hiệu ứng này bằng cách nhìn vào một vật thể nào đó, sau đó lần lượt nhắm mắt trái rồi đến mắt phải để thấy sự thay đổi về góc độ ra sao. Nếu nhắm mở liên tục thì sẽ thấy vật đó như nổi ra hơn.



Hình 19 - Hai luồng video được render lệch nhau

Để có được các tương tác giữa người dùng với môi trường mô phỏng thực tế ảo, các màn hình hiển thị có thể là điện thoại, tivi hoặc ngay trên kính có các bộ cảm biến vị trí (modun G-sensor) giúp xử lý thông tin khi quay sang trái, sang phải hay nhìn lên xuống. Các điện thoại thông minh hiện nay đều có modun G-sensor do đó rất thích hợp khi sử dụng với kính thực tại ảo.

1.4.2.2 Phần mềm

Hiện nay có rất nhiều ứng dụng VR đa dạng khác nhau như: hình tĩnh, hình động, video, game... Tuy nhiên, nếu xét trên mặt khái quát nhất hình động, video và game về bản chất là tập hợp của nhiều frame hình chạy liên tục. Chính vì thế trong phần này sẽ chủ yếu phân tích về hình ảnh.

Trên một số trang web như Facebook, Youtube đã trình chiếu một số hình ảnh, video 360 độ. Người dùng có thể dùng chuột kéo xoay để xem hình, video dưới nhiều góc độ khác nhau tại cùng một thời điểm.



Hình 20 – Video 360 độ có thể xem dưới nhiều góc độ

Để thực hiện được một bức ảnh hay video như thế này, cần có một chiếc Camera 360 độ. Đó là một chiếc Camera đặc biệt được thiết với nhiều ống kính ở nhiều hướng khác nhau. Ví dụ như Camera có tên Jump của Google được ghép từ Camera thành một vòng tròn. Hay một chỉ duy nhất một chiếc Camera nhưng có nhiều ống kính và tạo thành hình cầu



Hình 21 – Camera 360 độ

Mỗi ống kính Camera sẽ thu hay chụp một phần hình ảnh, cuối cùng xuất ra một file duy nhất đã được ghép toàn bộ các phần nhỏ lại với nhau tạo thành một chuỗi hình liền mạch nhau.



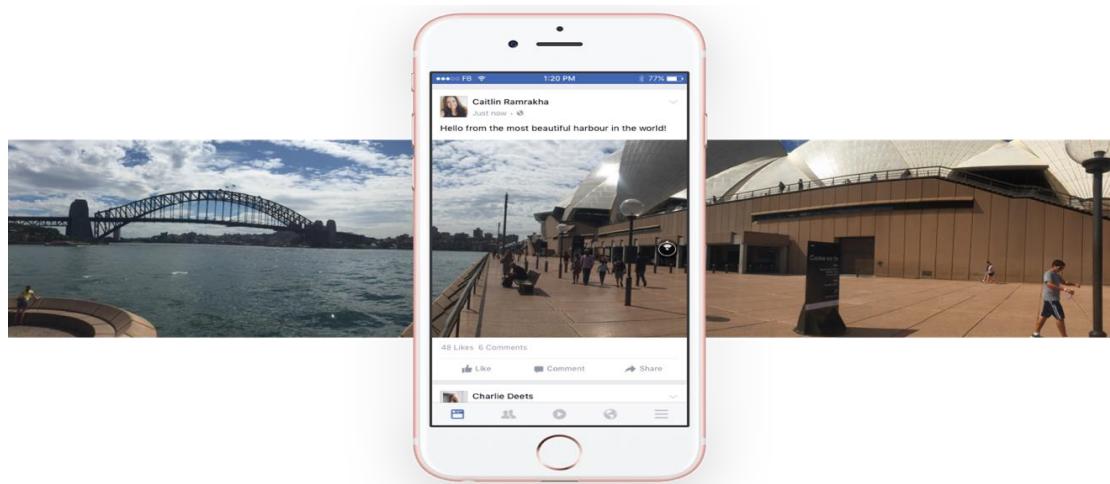
Hình 22 – Hình ảnh được chụp từ Camera 360 độ

Một hình ảnh 360 sẽ dễ dàng nhận thấy điểm đầu của bức ảnh sẽ khớp với điểm cuối của bức ảnh đó.



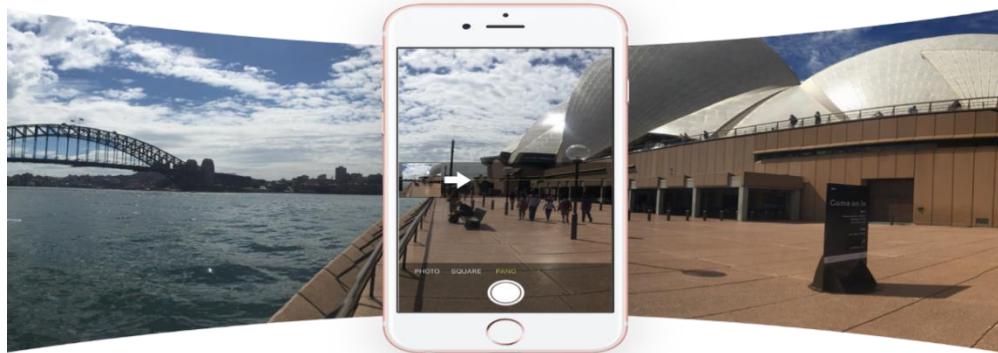
Hình 23 – Các điểm đầu và cuối của ảnh khớp nhau

Như vậy, nếu ghép toàn bộ các điểm khớp nhau tạo thành một chuỗi liền mạch với nhau và biểu diễn trong không gian 3 chiều (3D) ta có thể hình dung bức ảnh này có hình dáng là một hình cầu. Khi đưa bức ảnh đó vào một thiết bị có xem 360 độ hoặc VR. Hãy tưởng tượng khi biểu diễn bức ảnh đó trong không gian 2 chiều (2D), bức ảnh giống như một “thuốc phim”, và thiết bị xem chỉ nhận một phần hình ảnh trong “thuốc phim” đó.



Hình 24 – Thiết bị nhận một phần hình ảnh

Việc kéo xoay bức ảnh như đang thực hiện thao tác “kéo cuộn phim” để thiết bị có thể thu được một phần hình ảnh khác trong “thước phim” đó. Đối với VR, thao tác “kéo cuộn phim” đó hoàn toàn tự động nhờ vào con quay hồi chuyển trên thiết bị có thể cảm biến được thao tác khi ta xoay người.



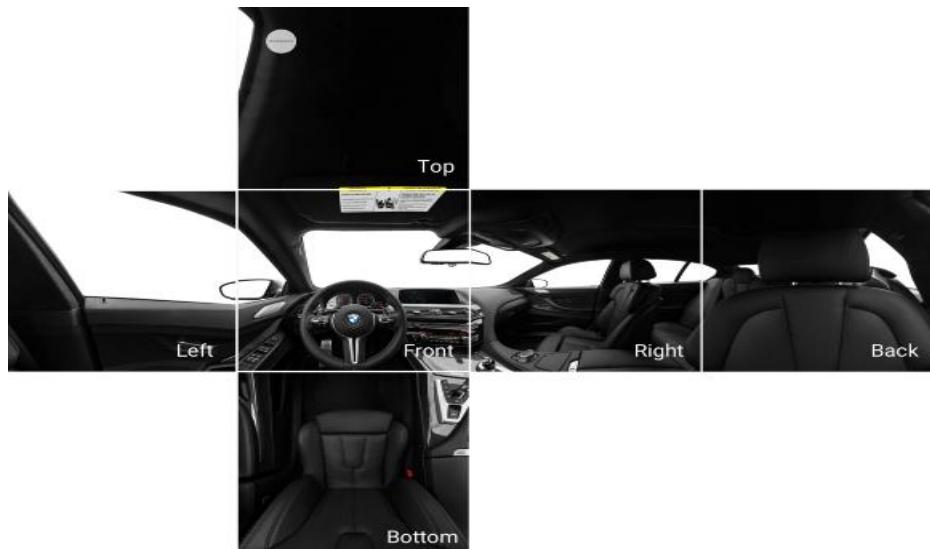
Hình 25 - Kéo bức ảnh để xem nhiều góc độ khác

Trong trường hợp khi kéo đến điểm cuối của bức ảnh, thiết bị sẽ tiếp tục nhận điểm đầu của bức ảnh, tạo thành một vòng tròn liên tục và không có điểm kết thúc. Tương tự như mọi điểm khác trên ảnh đã tạo thành một hình cầu. Việc ta có thể quan sát gần như vô hạn trong “hình cầu” đã tạo cảm giác như chính bản thân đang đứng trong khung cảnh đó. Đó chính là nguyên lý “đắm chìm” trong khái niệm “3I”.



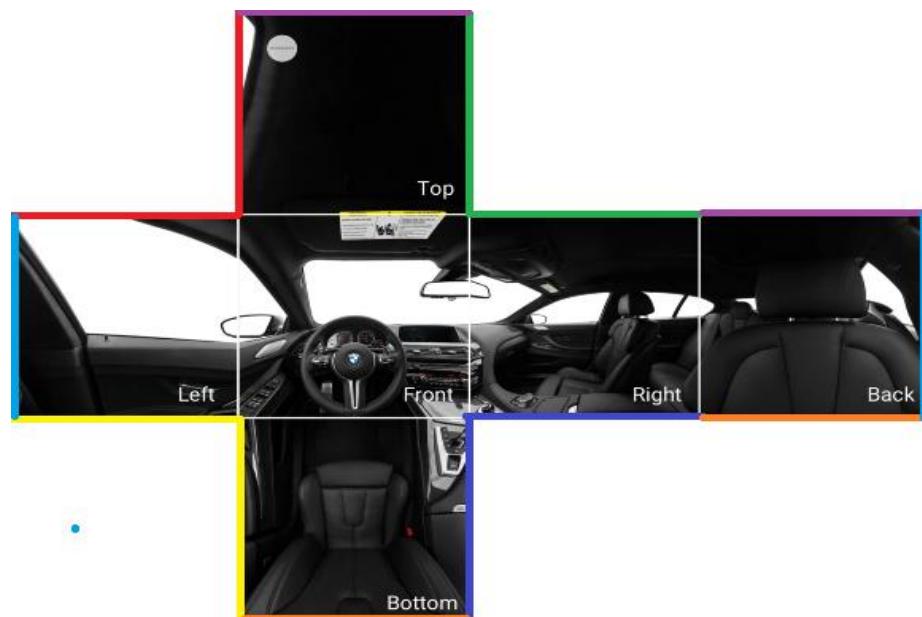
Hình 26 – Hình ảnh được biểu diễn dưới dạng hình cầu

Một trường hợp khác, khi xây dựng nên một bức ảnh chỉ từ Camera có 6 ống kính. Ta sẽ thu được một bức ảnh có dạng hình hộp lập phương (Cube). Khi biểu diễn trong không gian 2 chiều sẽ bao gồm 6 mặt phẳng.



Hình 27 – Hình được chụp bằng Camera với 6 ống kính

Tương tự như bức ảnh “hình cầu” trên, bức ảnh này cũng có những điểm khớp nhau khi nối lại, tạo thành một khối Cube trong không gian 3 chiều.



Hình 28 – Các điểm khớp nhau trên bức ảnh



Hình 29 – Kết quả thu được khi nối các điểm với nhau

1.4.3 Các cách tương tác trên VR

Tương tác trong VR là yếu tố quan trọng không thể thiếu trong đặc tính “3I” đã nêu. Ngoài việc người dùng đã được đắm chìm trong thế giới thực tế ảo thì việc tương tác sẽ bổ sung tính chân thật cho hệ thống nhiều hơn, đặc biệt là các ứng dụng game.

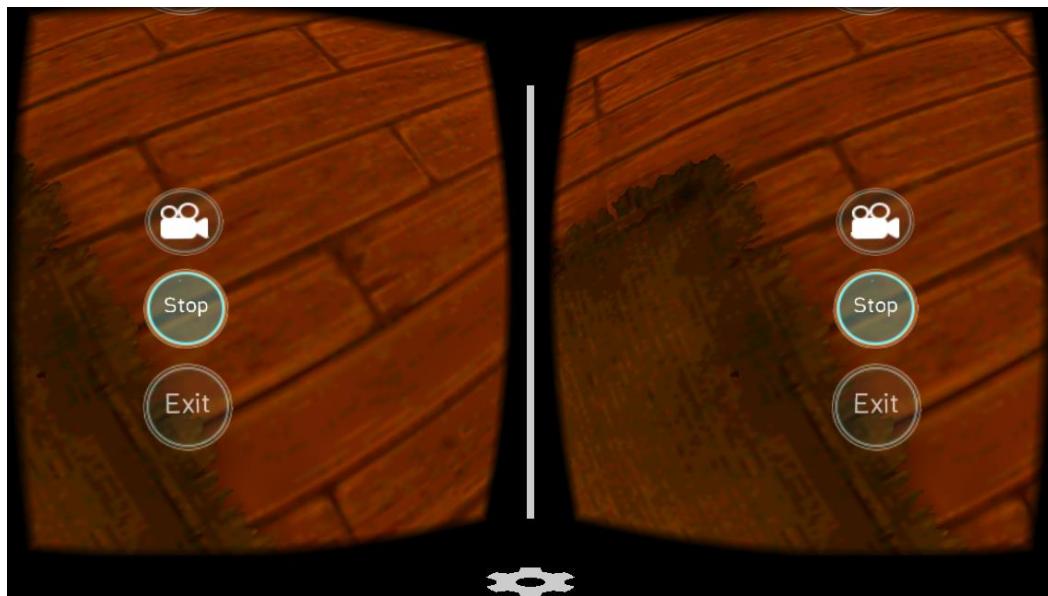
Hiện tại có rất nhiều cách tương tác khác nhau cho nhiều loại thiết bị. Tuy nhiên đa dạng và phức tạp nhất là trên các thiết bị PC, Console sử dụng hệ thống Oculus. Riêng với nền tảng di động vẫn còn nhiều hạn chế.

1.4.3.1 Hệ thống tự động

Đây là cách đơn giản nhất và hiệu quả nhất trên các thiết bị di động và hầu như chẳng tốn kém thêm chi phí nào về thiết bị. Vì cơ bản nó được thiết lập trong quá trình xây dựng ứng dụng bằng các trình Studio hay Engine.

Về mặt này đơn giản chỉ là xây dựng những đoạn mã nhận diện thao tác cơ bản của nhân vật trong ứng dụng như mọi Game trước đó. Ví dụ, hệ thống tự động mở cửa khi nhân vật tiến lại gần.

Đối với VR, một số chức năng được xây dựng dựa vào hệ thống tự động thường thấy như: AutoWalk (tự động đi tới), Look Around (Nhìn xung quanh), CheckDistance (Kiểm tra khoảng cách)... Hệ thống có thể hiện được sự đa dạng, phong phú hay không là tùy thuộc vào nhà phát triển ứng dụng đó.

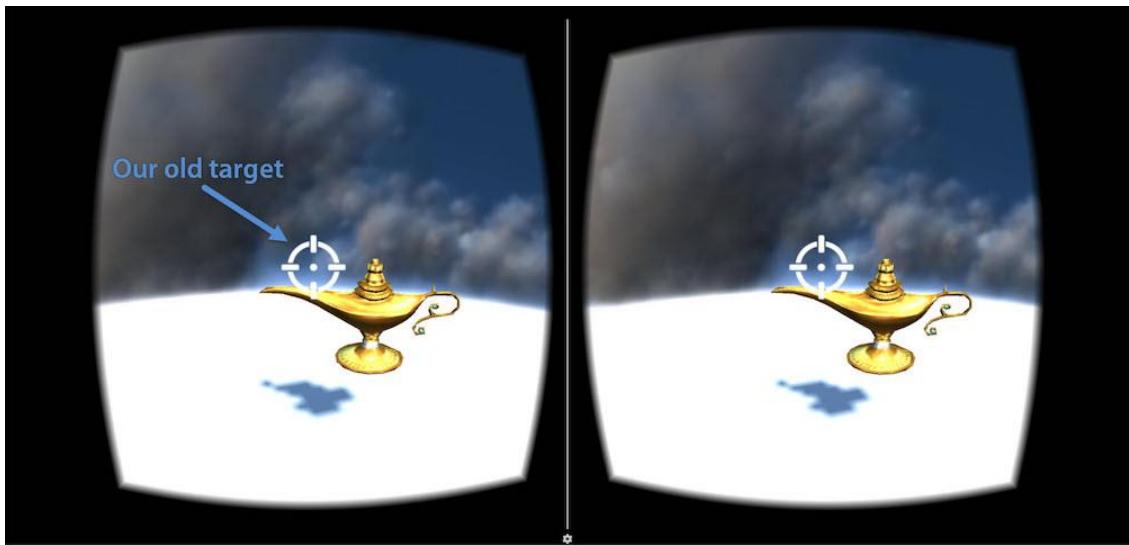


Hình 30 - Ứng dụng VR hiển thị Menu khi nhân vật nhìn xuống đất

Ngoài ra, đặc biệt trên VR còn xuất hiện ý tưởng chức năng khá độc đáo gọi là Gaze (Nhìn chằm chằm). Người dùng có thể hạn chế những thao tác rườm rà như chọn Menu, click vào Button nhờ vào chức năng Gaze. Cụ thể hơn, trên ứng dụng VR sẽ xuất hiện một hồng tâm (Reticle) ở trung tâm mỗi màn hình. Khi các luồng video, hình ảnh được xoay, hồng tâm này cũng sẽ di chuyển theo và người dùng có thể sử dụng như một con trỏ chuột khi sử dụng PC.

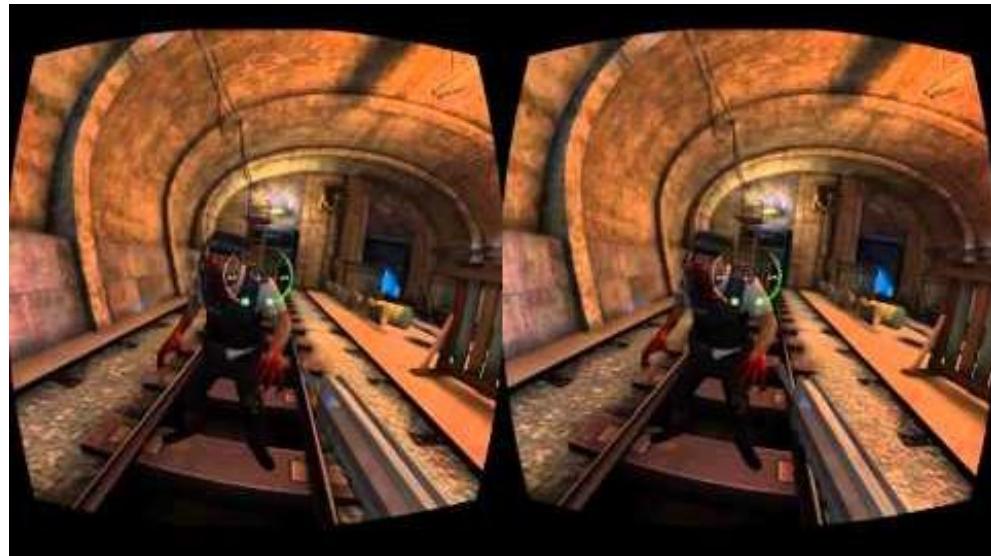


Hình 31 – Mô phỏng về Reticle của VR

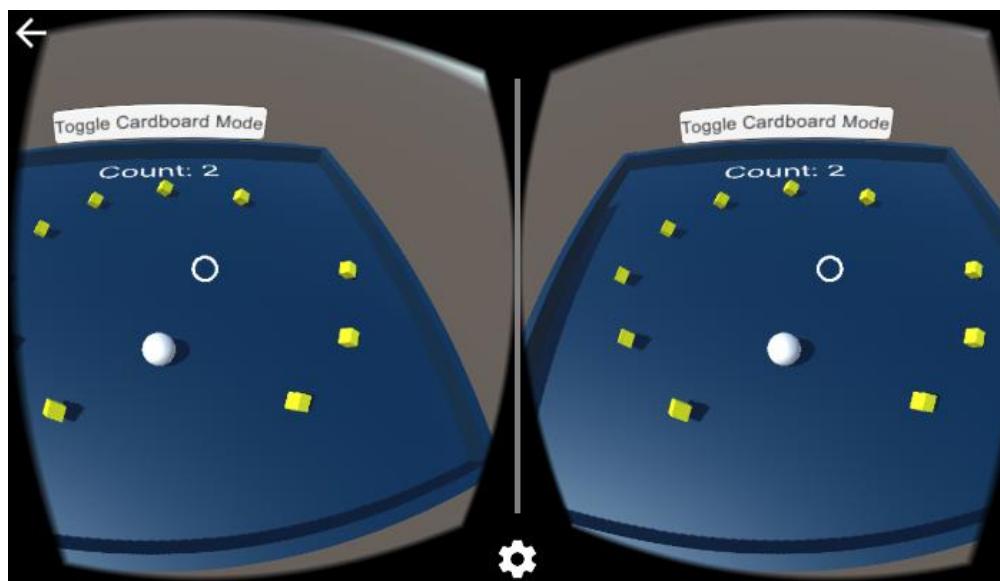


Hình 32 – Ví dụ về Reticle trên VR

Như vậy với Reticle, nhà phát triển có thể xây dựng thêm nhiều tính năng khác nhau cho ứng dụng. Ví dụ như Auto Shoot (tự động bắn) cho những game bắn súng, hoặc thực hiện thao tác Click Button. Khi người dùng đưa hồng tâm vào mục tiêu cần tương tác, hệ thống sẽ cho một khoảng thời gian chờ, sau đó hệ thống sẽ thực hiện chức năng cho mục tiêu đó.



Hình 33 – Game Zombie Shooter sử dụng chức năng AutoShoot



Hình 34 – Game MyRollBall sử dụng Reticle để tương tác

1.4.3.2 Tay cầm và hệ thống cảm biến

Khi sử dụng kính VR, người dùng hầu như không thể tương tác bằng thao tác chạm trên điện thoại hay sử dụng thiết bị ngoại vi như chuột, bàn phím. Một cách khác để tăng tính tương tác là sử dụng thiết bị tay cầm không dây được kết nối bằng bluetooth và bộ cảm biến chuyển động

Đối với tay cầm (Gamepad) thường thấy trên hệ máy Console như (PSx, xBox...) không thể sử dụng như một công cụ tương tác được. Vì nó sẽ bó buộc cả hai tay, thiếu thoải mái và mất đi tính chân thật.



Hình 35 – Tay cầm Console

Như vậy các thiết bị tay cầm sử dụng một tay (One-Handed Gamepad) được ra đời. Khi chỉ cần thao tác bằng một tay, người dùng sẽ không còn cảm thấy sự bó buộc so với tay cầm cũ.



Hình 36 – Tay cầm bluetooth một tay

Đặc điểm chung của các loại tay cầm được thiết kế bao gồm một cần Analog, các nút nhấn L/R như trên console, và một số nút cứng khác. Tuy nhiên, các tay cầm hiện nay được chia thành 2 loại chính: tay cầm truyền thống và tay cầm cảm biến.

Tay cầm truyền thống: là loại căn bản và ít thao tác nhất. Các phím nhấn tương tác thường bao gồm: Click/Touch (chạm), Back (Quay về màn hình trước), Volume up/down (tăng giảm âm lượng)... Thường được sử dụng cho những ứng dụng ít chức năng thao tác, hay chỉ đơn giản là thao tác OnClick



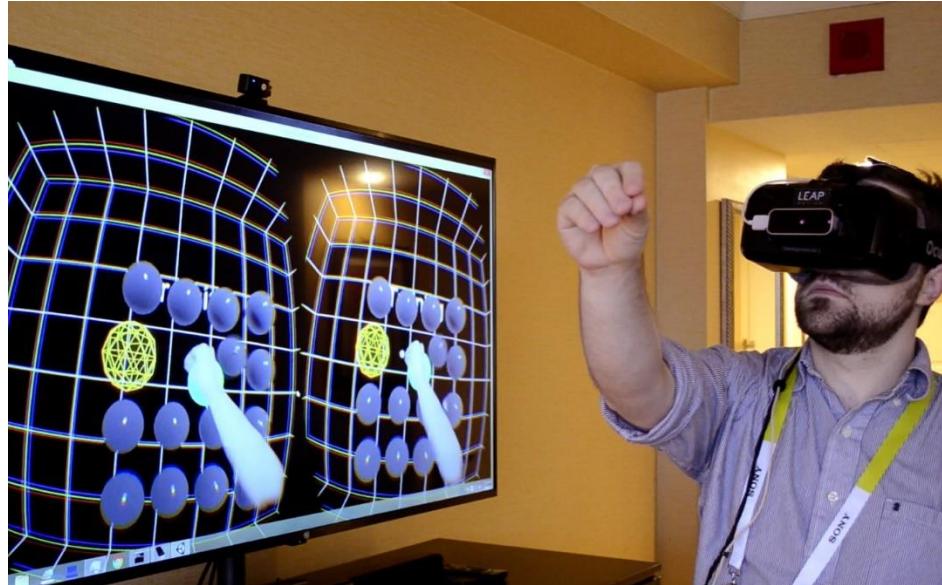
Hình 37 - Sử dụng Reticle kết hợp thao tác Click để chọn Menu

Tay cầm cảm biến: là loại tay cầm cao cấp hơn được trang bị hàng loạt các bộ phận cảm biến, nhận dạng cử chỉ tay của người dùng. Thiết bị này có thể dễ dàng nhận biết được khi người dùng cầm vào, cử động ngón tay, vẫy tay... Tất cả hành vi này sẽ được máy tính xử lý và chuyển hóa thành chuyển động tương ứng cho nhân vật trong game hoặc một thao tác nào đó.



Hình 38 – Thiết bị tay cầm cảm biến có tên Oculus Touch

Một số phiên bản khác tương tự như tay cầm cảm biến có thể giúp nâng cao khả năng tương tác bằng cách nhận diện được cử chỉ tay như thiết bị Leap Motion (Thiết bị cảm biến cử chỉ) hoặc VR Gloves (Găng tay dành cho VR)

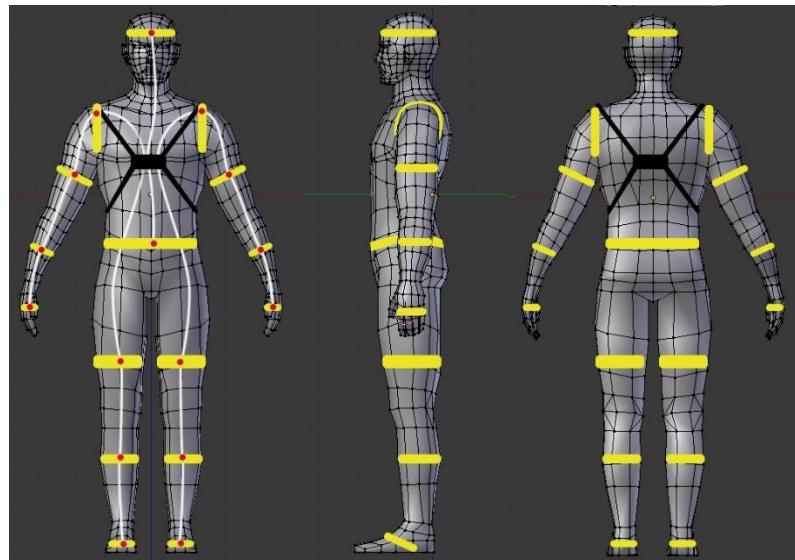


Hình 39 – Thiết bị Leap Motion



Hình 40 – Thiết bị VR Gloves

Bên cạnh tay cầm cảm biến, hãng VR lớn nhất hiện nay là Oculus còn cung cấp một thiết bị cảm biến chuyển động toàn thân gọi là Horus VR Suit. Horus VR Suit là bộ cảm biến chuyển động toàn thân giúp mô phỏng toàn bộ chuyển động của cơ thể người ở ngoài đời thực vào thế giới ảo. Bộ cảm biến được phát triển từ công nghệ cảm biến không gian. Khi sử dụng bộ cảm biến này, mọi chuyển động của cơ thể người dùng sẽ được mô phỏng lại và có thể tương tác với mọi đối tượng trong thế giới ảo.



Hình 41 – Mô phỏng thiết bị Horus VR Suit

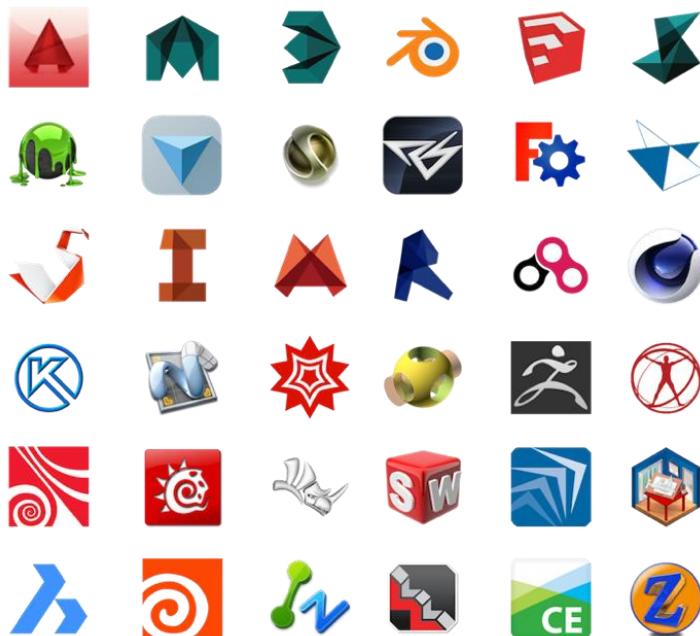


Hình 42 – Thiết bị Horus VR Suit

CHƯƠNG 2 –MÔ PHỎNG ĐẠI HỌC TÔN ĐỨC THẮNG BẰNG PHẦN MỀM BLENDER

Sau khi tìm hiểu về khái niệm thực tế ảo. Nhóm quyết định xây dựng ứng dụng mô phỏng trường Đại Học Tôn Đức Thắng. Quá trình xây dựng sẽ trải qua hai giai đoạn: Giai đoạn dựng mô hình bằng phần mềm vẽ 3D và Giai đoạn xây dựng chức năng thực tế ảo bằng Unity.

Để mô phỏng bất cứ hình thể gì ở dạng 3D, ta cần sử dụng các phần mềm chuyên hỗ trợ vẽ 3D. Hiện nay có rất nhiều phần mềm hỗ trợ việc đó như: 3Ds Max, Google SketchUp, AutoCAD, Maya, ...



Hình 43 - Một số phần mềm vẽ 3D

Nhóm quyết định sử dụng phần mềm Blender để mô phỏng trường đại học Tôn Đức Thắng vì một vài lý do phụ như:

- + Dung lượng để tải Blender về tương đối nhỏ gọn và cấu hình yêu cầu để sử dụng cũng không quá cao.
- + Một số phần mềm vẽ 3D chuyên nghiệp cần trả phí để sử dụng như 3Ds Max và Maya.
- + Blender hỗ trợ mã nguồn mở.

Còn lý do chính thì có thể nói là do chọn bừa, sau quá trình tìm kiếm trên Google và đọc được các đề xuất từ những người đi trước, chứ không có tính toán khoa học nào ở đây cả. Vì nhóm quan niệm rằng chất lượng của sản phẩm phụ thuộc vào người sử dụng chứ không phải phụ thuộc vào phần mềm.

Trước khi đi vào chi tiết quá trình mô hình trường Tôn Đức Thắng sẽ lượt qua một số nội dung giới thiệu về phần mềm Blender

2.1 Blender là gì

Ban đầu Blender được thành lập chỉ với mục đích làm phim hoạt hình 3D cho xưởng phim của hãng Neo Geo. Ngày nay Blender là một phần mềm đồ họa máy tính chuyên nghiệp, miễn phí và mã nguồn mở, được sử dụng để tạo ra các thước phim hoạt hình, hiệu ứng hình ảnh, tác phẩm nghệ thuật, mô hình in 3D, các ứng dụng tương tác 3D và các trò chơi điện tử.



Hình 44 – Icon phần mềm Blender

2.2 Lịch sử phát triển

Tháng 1/1995, cha đẻ của Blender, Ton Roosendaal, đã xây dựng một công cụ dành riêng cho Neo Geo (hãng chuyên phát triển phần cứng dành cho trò chơi điện tử thuộc tập đoàn SNK – Nhật Bản) nhằm mục đích thiết kế ba chiều và làm phim ở studio của Neo Geo.

2.2.1 *Khởi đầu không thuận lợi*

Khi Neo Geo được mua lại bởi công ty khác, Ton Roosendaal và Frank van Beek đã sáng lập đã sáng lập ra một tổ chức riêng là Not a Number Technologies (NaN) vào tháng 6/1998 để tiếp tục phát triển Blender. Blender khi đó được phát hành dưới dạng shareware (dùng thử miễn phí, hết hạn trả tiền) cho đến khi tổ chức NaN phá sản vào năm 2002.

2.2.2 *Sự cố gắng không ngừng*

Không dừng lại ở đó, tháng 7/2002 Roosendaal tiếp tục vẫn động chiến dịch “Free Blender” nhằm huy động tài chính, kêu gọi sự tài trợ và góp vốn từ cộng đồng để các chủ nợ đồng ý phát hành Blender với hình thức mã nguồn mở.

2.2.3 *Sự hồi sinh của Blender*

Không lâu sau đó, ngày 7/9/2002, số tiền 100.000 Euro được thu thập đủ và mã nguồn của Blender được công khai. Đến thời điểm hiện tại, Blender là một phần mềm tự do và đang phát triển nhanh chóng dưới sự chỉ đạo của công ty độc lập phi lợi nhuận Blender Foundation.

2.3 Tính năng

Blender được trang bị nhiều tính năng mang đặc điểm của một phần mềm công nghệ cao 3D gồm:

- Hỗ trợ một loạt các khái niệm học nguyên thuỷ bao gồm lưới đa giác, phân chia bề mặt của mô hình một cách nhanh chóng, đường cong Bézier, các bề mặt NURBS (mô hình toán học sử dụng trong kỹ thuật đồ họa máy tính để vẽ đường cong và bề mặt), metaballs, icospheres, kỹ thuật điêu khắc kỹ thuật số (bao gồm dynamic topology, maps baking, remeshing, resymetrize, decimation), font chữ outline (được tạo thành từ những công thức toán học) và một hệ thống mô hình n-gon mới là B-mesh.
- Công nghệ render Cycle giúp tận dụng lợi thế của GPU để phục vụ cho việc render.
- Tích hợp với một số các công nghệ render bên ngoài thông qua hệ thống plugin.
- Công cụ diễn hoạt Keyframed bao gồm chuyển động đảo ngược, các phụ kiện (bộ khung xương), hook, đường cong, biến dạng lưới (lattice-based deformat), diễn hoạt động, diễn hoạt phi tuyến tính, các ràng buộc và trọng số đỉnh (vertex weighting).
- Công cụ mô phỏng động lực cho phần thân mềm (soft body) bao gồm lưới phát hiện va chạm, động lực học chất lưu LBM, mô phỏng khói, Bullet rigid body dynamics, hiệu ứng sóng biển.
- Hệ thống hạt bao gồm việc hỗ trợ cho particle-based hair.
- Hệ thống modifiers để tạo các hiệu ứng không gây biến đổi cấu trúc.
- Mã Python để tạo công cụ và tạo mẫu, logic trò chơi, nhập và xuất từ các định dạng khác, tự động hóa công việc và tùy chỉnh công cụ.
- Chỉnh sửa cơ bản video/âm thanh phi tuyến tính.

- Blender Game Engine, một dự án phụ, cung cấp các tính năng tương tác như phát hiện va chạm, động cơ động và logic lập trình. Nó cũng cho phép tạo ra các ứng dụng độc lập, thời gian thực khác nhau từ mô hình kiến trúc đến các trò chơi điện tử.
- Kép cầu hoạ tiết gồm texture painting, projective painting, vertex painting, weight painting và dynamic painting.
- Kiểm soát thời gian thực trong quá trình mô phỏng vật lý và render.
- Hệ thống camera và theo dõi đối tượng.

2.4 Giao diện người dùng

2.4.1 Hai chế độ chính

Mặc dù còn có các chế độ khác như VertexPaint, WeightPaint, SculptMode, cơ chế hoạt động của Blender dựa trên 2 chế độ chính là Object mode và Edit mode.

2.4.1.1. Object Mode

Được sử dụng để điều khiển toàn bộ đối tượng. Các hoạt động như di dời (grab), xoay (rotate) và phóng thu (scale) đều ảnh hưởng đến toàn bộ một hay nhiều đối tượng.

2.4.1.2. Edit Mode

Được sử dụng để thao tác trên dữ liệu thực tế của đối tượng. Trên thật tế, một đối tượng được tạo thành từ nhiều mặt phẳng nhỏ, các mặt phẳng nhỏ đó cấu thành từ nhiều cạnh hay nói sâu hơn là từ nhiều đỉnh nhỏ mà ra. Dựa trên nguyên lý đó, edit mode cho phép ta chỉnh sửa các mặt phẳng/cạnh/đỉnh của đối tượng để thay đổi hình dáng đối tượng.

2.4.2 Phím tắt

Cũng như bao phần mềm thiết kế, lập trình, soạn thảo văn bản, ... khác. Blender cũng trang bị cho mình rất nhiều phím tắt hữu ích, hỗ trợ cho việc thao tác, thiết kế một cách nhanh gọn nhất. Lưu ý: Bên dưới đây là một số phím tắt được rút ra trong quá trình học tập và làm việc cùng Blender, chưa phải là phiên bản đầy đủ nhất. Và chỉ liệt kê những phím tắt liên quan đến thiết kế, không liệt kê những phím tắt khác (lưu file, mở file mới, mở tuỳ chỉnh, ...)

2.4.2.1 Phím tắt chung

Phím tắt chung là phím tắt được sử dụng ở cả chế độ object mode và edit mode.

Phím tắt	Công dụng
‘ctrl’ + ‘alt’ + Q	Bật chế độ 4 màn hình, quan sát đối tượng ở mọi góc cạnh
‘ctrl’ + ‘phím up’	Bật chế độ full màn hình Blender
‘Phím số 1’	Di chuyển màn hình về mặt trước đối tượng
‘Phím số 2’	Di chuyển màn hình hướng xuống so với màn hình hiện tại
‘Phím số 3’	Di chuyển màn hình về mặt bên hông đối tượng
‘Phím số 4’	Di chuyển màn hình hướng về bên trái so với màn hình hiện tại
‘Phím số 5’	Di chuyển màn hình về mặt bên dưới đối tượng (mặc định là quan sát đối tượng từ trên xuống)
‘Phím số 6’	Di chuyển màn hình hướng về bên phải so với màn hình hiện tại
‘Phím số 7’	Di chuyển màn hình về phần đỉnh đầu của đối tượng
‘Phím số 8’	Di chuyển màn hình hướng lên so với màn hình hiện tại
‘Phím số 9’	Xoay ngang 180 độ màn hình hiện tại để quan sát vị trí đối diện của đối tượng 3D
‘shift’ + ‘c’	Công dụng 1: xem tất cả đối tượng (ví dụ chọn 5 đối tượng nhưng màn hình hiện tại chỉ thấy 1 đối tượng, sau khi bấm màn hình sẽ tự thu nhỏ để thấy cả 5 đối tượng)

	Công dụng 2: đưa con trỏ 3D cursor (con trỏ click chuột phải trên màn hình để phục vụ cho việc định vị trực của đối tượng) về vị trí chính giữa màn hình
‘a’	Chọn tất cả đối tượng (trong object mode) hoặc chọn tất cả các mặt/đỉnh/cạnh (trong edit mode)
‘b’	Chọn đối tượng (trong object mode) hoặc chọn mặt/đỉnh/cạnh (trong edit mode) theo một phạm vi ô vuông/chữ nhật nhất định
‘c’	Chọn đối tượng (trong object mode) hoặc chọn mặt/đỉnh/cạnh (trong edit mode) theo một phạm vi hình tròn nhất định
‘ctrl’ + ‘chuột trái’	Chọn đối tượng (trong object mode) hoặc chọn mặt/đỉnh/cạnh (trong edit mode) theo một phạm vi được vẽ bằng chuột máy tính
‘+’ hoặc ‘-’	Zoom vào hoặc ra
‘ctrl’ + ‘i’	Bỏ chọn (các) đối tượng (object mode) hoặc đỉnh/cạnh/mặt (edit mode) đang chọn hiện tại và chọn tất cả các đối tượng hoặc đỉnh/cạnh/mặt còn lại
‘g’ (+ ‘x/y/z’)	g là viết tắt của grab, dùng để di chuyển đối tượng (object mode) hoặc đỉnh/cạnh/mặt (edit mode) đã chọn. Có thể chọn x/y/z để di chuyển theo chiều ngang/dọc/cao
‘r’ (+ ‘x/y/z’)	r là viết tắt của rotate, dùng để xoay đổi hướng đối tượng (object mode) hoặc đỉnh/cạnh/mặt (edit mode) đã chọn. Có thể chọn x/y/z để xoay theo chiều ngang/dọc/cao
‘s’ (+ ‘x/y/z’)	g là viết tắt của scale, dùng để thay đổi kích thước đối tượng (object mode) hoặc đỉnh/cạnh/mặt (edit mode) đã chọn. Có thể chọn x/y/z để phóng thu theo chiều ngang/dọc/cao
‘shift’ + ‘s’	Các hình thức di chuyển đặc biệt cho con trỏ 3D cursor và cho các đối tượng đã chọn

‘shift’ + ‘a’	Khởi tạo các giao diện có sẵn của blender (cube, cylinder, ...)
‘tab’	Chuyển đổi giữa object mode và edit mode
‘shift’ + ‘tab’	Bật/tắt chế độ snap, dùng để kết dính các vật thể/cạnh/góc/đỉnh dựa vào cạnh/góc/đỉnh của vật thể khác hoặc của chính nó
‘z’	Bật/tắt chế độ wire frame (chế độ loại bỏ mặt phẳng, chỉ nhìn thấy các đỉnh và cạnh của thấu đốï tượng)
‘h’	Ẩn đối tượng (object mode) hoặc đỉnh/cạnh/mặt (edit mode) đi
‘alt’ + ‘h’	Hiện đối tượng (object mode) hoặc /đỉnh/cạnh/mặt (edit mode) đã ẩn
‘m’ + ‘số bất kỳ’	Di chuyển đối tượng đến trang khác nhưng vẫn nằm cùng project
‘chọn đối tượng /đỉnh/cạnh/mặt bất kỳ’ + ‘shift’ + ‘d’	Nhân đối đối tượng (object mode) hoặc /đỉnh/cạnh/mặt (edit mode) thành một đối tượng hoặc đỉnh/mặt/cạnh mới
‘shift’ + ‘f1’	Import đối tượng từ một tập tin blender khác.

Bảng 1 – Các phím tắt chung của Blender

2.4.2.2 Phím tắt dành riêng cho edit mode

Phím tắt	Công dụng
‘ctrl’ + ‘tab’	Chuyển đổi kiểu tương tác giữa đỉnh, cạnh và mặt phẳng
‘Chọn (các) đỉnh/cạnh/mặt bất kỳ’ + ‘ctrl’ + ‘+/-’	Chọn các mặt phẳng xung quanh (các) đỉnh/cạnh/mặt đã chọn
‘alt’ + ‘chuột phải chọn đỉnh/cạnh /mặt phẳng bất kỳ’	Chọn các đỉnh/cạnh/mặt xung quanh đỉnh/cạnh/mặt phẳng đã chọn theo một đường tròn
‘ctrl’ + ‘v/f/e’	Mở thanh menu tùy chỉnh đối với các đỉnh/mặt/cạnh
‘w’	Mở thanh tùy chỉnh đặc biệt
‘alt’ + ‘r’	Tạo một đường cắt nằm giữa đường hoặc mặt phẳng cần cắt
‘chọn 2 đỉnh/cạnh trở lên’ + ‘f’	Hợp 2 đỉnh thành một cạnh, nhiều đỉnh thành một mặt, 2 cạnh hoặc nhiều hơn thành một mặt
‘chọn một đỉnh/cạnh/mặt bất kỳ’ + ‘e’	Tạo thêm một đỉnh/cạnh/mặt tương tự từ đỉnh/cạnh mặt đã chọn, đồng thời đỉnh/cạnh/mặt mới cũng kết nối dính liền với đỉnh/cạnh mặt đó.
‘chọn một đỉnh/cạnh/mặt bất kỳ’ + ‘p’	Cắt một đỉnh/cạnh/mặt từ một đối tượng ra thành một đối tượng nằm riêng biệt. Ví dụ cắt một cạnh từ đối tượng A ra thì cạnh đó sẽ trở thành đối tượng B

Bảng 2 – Các phím tắt trong chế độ edit mode của Blender

2.4.2.3 Phím tắt dành riêng cho object mode

Phím tắt	Công dụng
‘ctrl’ + ‘j’	Hợp các đối tượng đã chọn lại thành một đối tượng duy nhất
‘alt’ + ‘ctrl’ + ‘shift’ + c	Điều chỉnh trực cho đối tượng

Bảng 3 – Các phím tắt trong chế độ object mode của Blender

2.4.3 Chuyển động camera

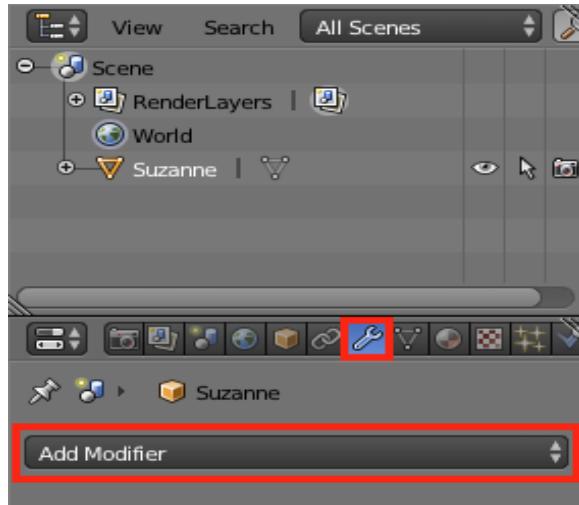
Ngoài các phím tắt giúp chuyển động camera như ở bên trên, ta còn có thể sử dụng chuột giữa (chuột lăn) để xoay camera phục vụ dễ dàng cho việc thiết kế.

Hình thức	Công dụng
‘đè chuột giữa’	Xoay camera trong phạm vi 360 độ
‘shift’ + ‘đè chuột giữa’	Xoay camera trong phạm vi 180 độ
‘alt’ + ‘đè chuột giữa’	Đưa camera tiến lại gần hoặc ra xa
‘lăn chuột giữa’	Đưa camera tiến lại gần hoặc ra xa
‘ctrl’ + ‘lăn chuột giữa’	Di chuyển camera sang trái hoặc phải
‘shift’ + ‘lăn chuột giữa’	Di chuyển camera lên hoặc xuống

Bảng 4 – Các phương thức di chuyển camera trong Blender

2.4.4 Modifiers

Là những chức năng quan trọng giúp tiết kiệm thời gian, hỗ trợ thiết kế các đối tượng đặc thù một cách tối ưu nhất.



Hình 45 – Modifier trong Blender

Sau đây là một vài modifier phổ biến được sử dụng để xây dựng nên đồ án này.

2.4.4.1 Array

Giúp sao lưu đối tượng một cách nhanh chóng, có thể sử dụng một lúc nhiều array theo chiều x, y, z để tạo một chuỗi các đối tượng liền kề có khoảng cách bằng nhau.



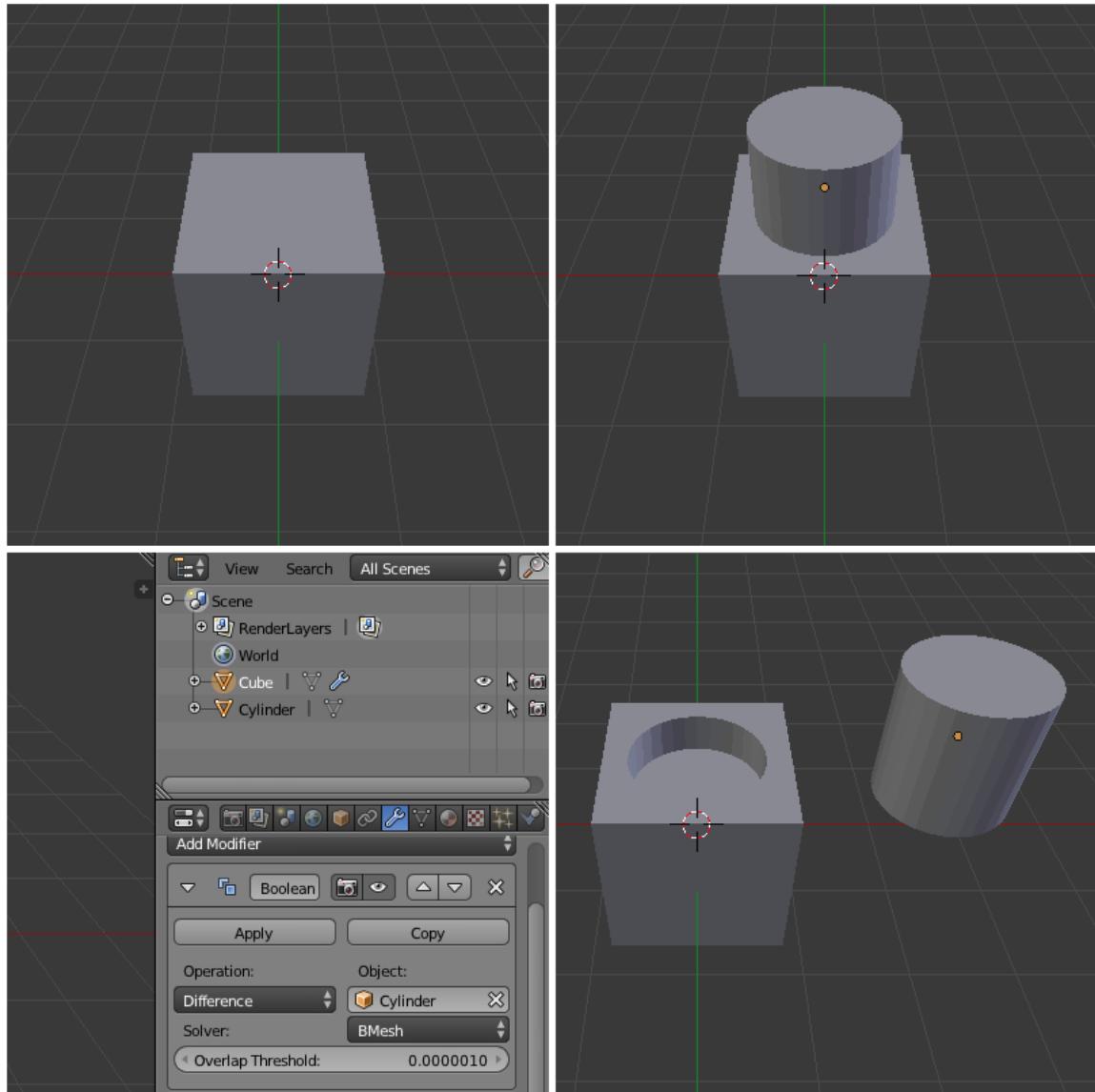
Hình 46 – Sử dụng Modifier Array (P1)



Hình 47 – Sử dụng Modifier Array (P2)

2.4.4.2 Boolean

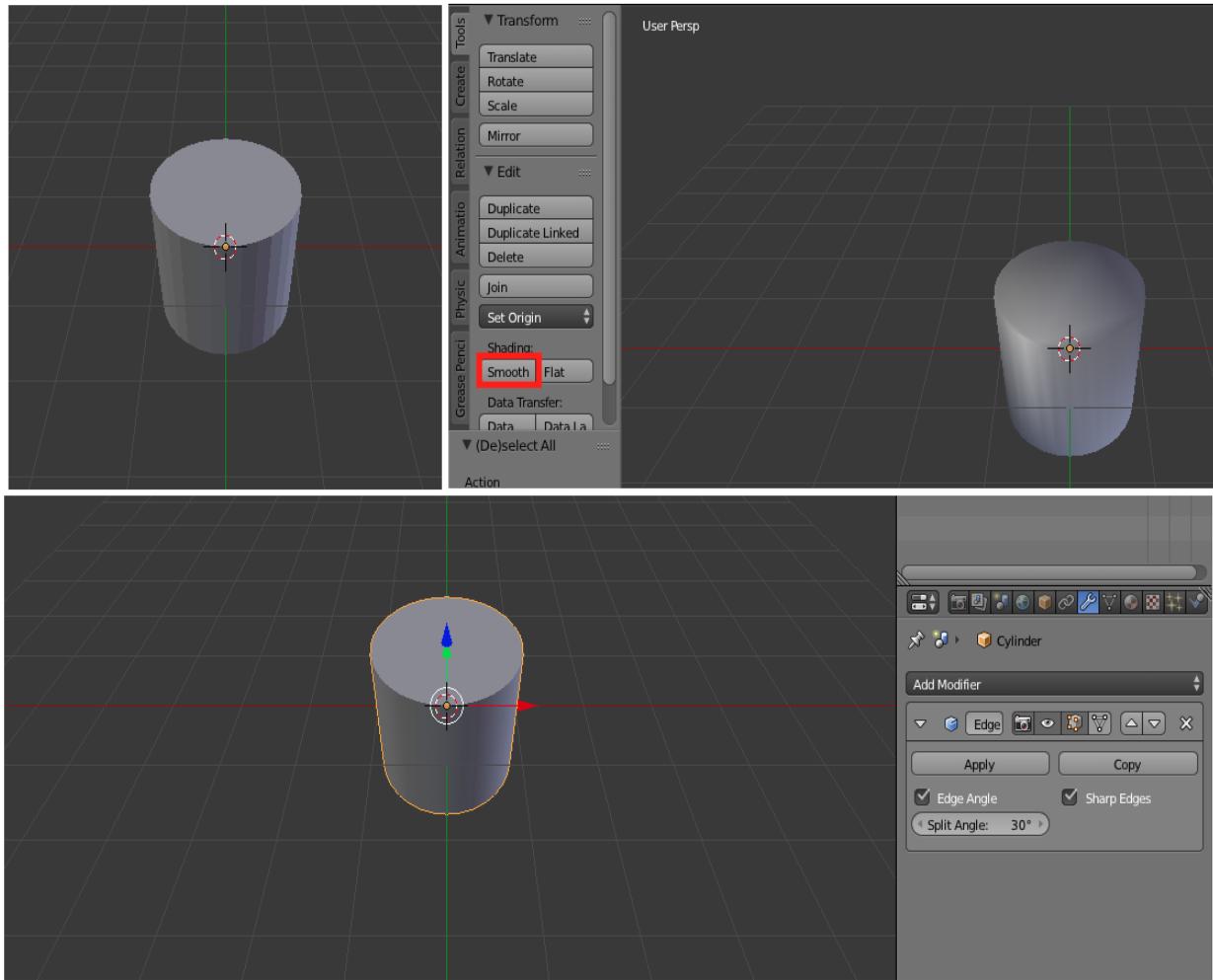
Chức năng chính là được sử dụng để đục lỗ một đối tượng. Bên cạnh đó còn một vài chức năng phụ khác như kết hợp 2 đối tượng lại với nhau hoặc giúp lấy những phần chung của 2 đối tượng.



Hình 48 – Sử dụng modifier Boolean

2.4.4.3 Edge Split

Giúp các đối tượng phức tạp, được tạo nên từ nhiều mặt phẳng khác nhau, có được độ mềm mại nhất định.

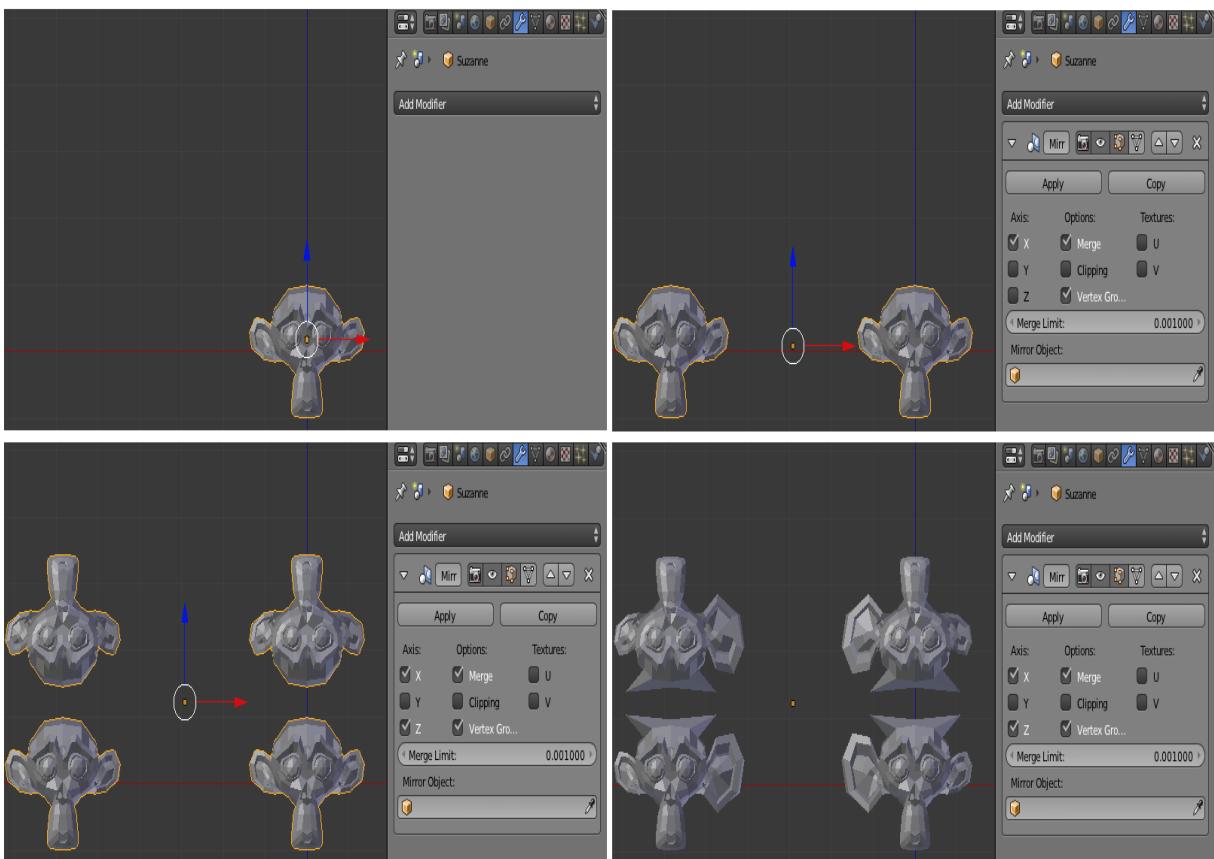


Hình 49 – Sử dụng modifier Edge Split

2.4.4.4 Mirror

Cũng giống như khi ta soi gương, hình ảnh trong một chiếc gương (thông thường) sẽ phản chiếu tất cả hình ảnh chi tiết của ta thì cách hoạt động của mirror cũng giống hệt thế.

Modifier Mirror cũng mang một phần giống với modifier array, nhưng khác biệt lớn nhất đó chính là việc đặt chiếc gương ở đâu để nó có thể phản chiếu hình ảnh (đặt trực cho đối tượng với trực là điểm vàng trên màn hình)

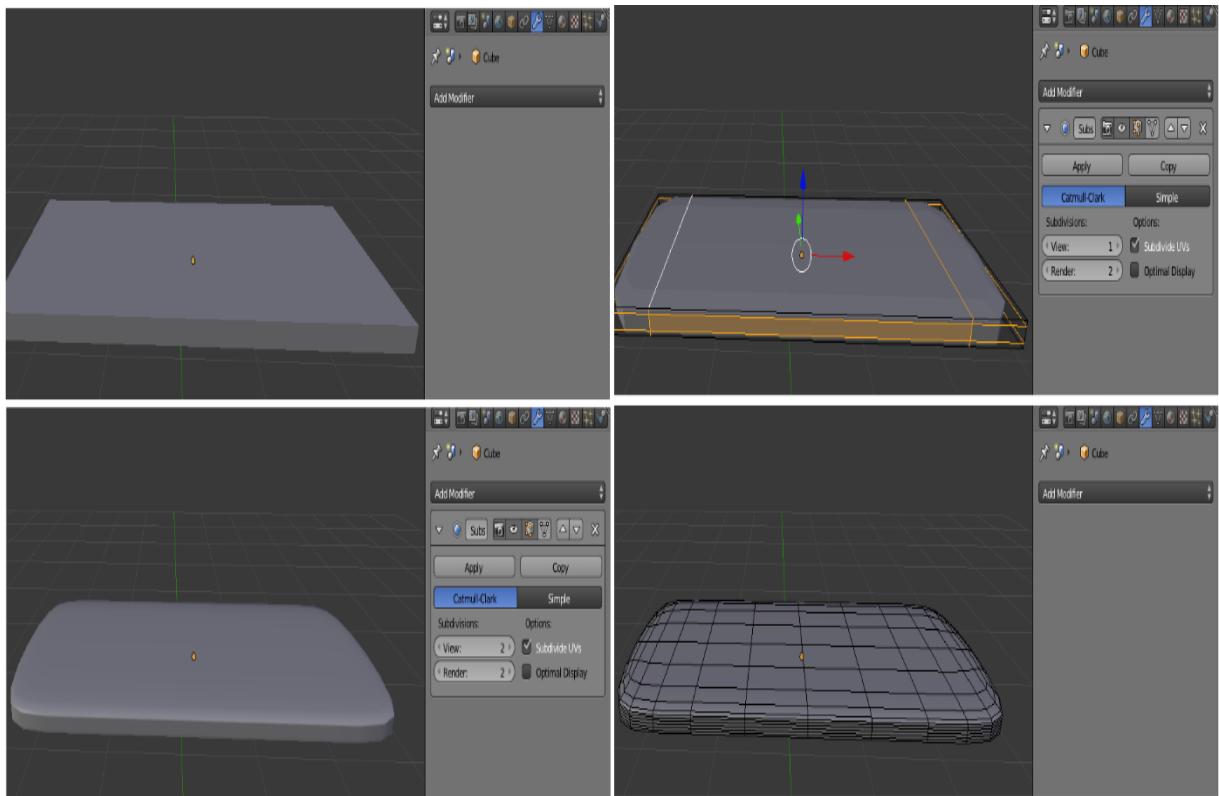


Hình 50 – Sử dụng modifier Mirror

2.4.4.5 Subdivision Surface

Có tác dụng chia một mặt phẳng ra thành nhiều mặt phẳng nhỏ, giúp bo tròn các đường nét vật thể và tạo thành một vật thể đẹp mắt, rõ nét tùy theo thông số điều chỉnh.

Tuy có thể tùy chỉnh thông số để làm rõ nét vật thể, nhưng đó cũng là yếu điểm duy nhất của modifier này, vì càng chia ra nhiều mặt phẳng nhỏ, càng làm vật thể thêm phức tạp và dung lượng vật thể càng cao hơn.



Hình 51 – Sử dụng modifier Subdivision Surface

2.4.5 Material và Texture

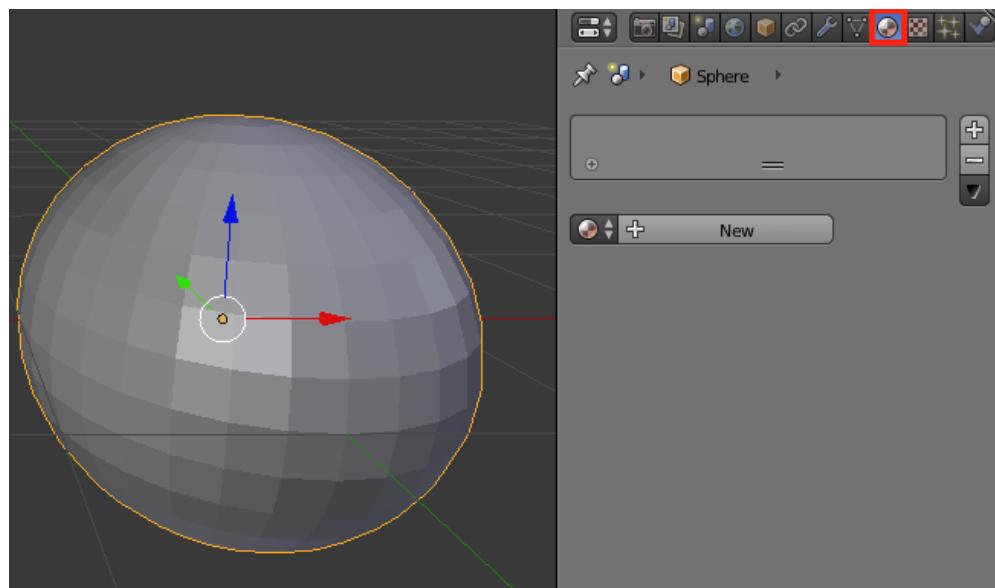
Để một đối tượng trở nên sinh động và bắt mắt, việc phôi màu sắc cho vật thể là cực kỳ quan trọng. Một thiết kế đẹp được trang bị thêm material hoặc texture cũng có thể ví như một người phụ nữ đẹp khoác lên mình một lớp áo cùng một chút son phấn trang điểm, không khác gì hổ mọc thêm cánh.

2.4.5.1 Material

Sử dụng material, ta có thể thay đổi màu sắc của vật thể hoặc mặt phẳng của vật thể, cùng với một số hiệu ứng khác như trong suốt, độ sáng tối, độ phản chiếu như một tấm kính, độ bóng.

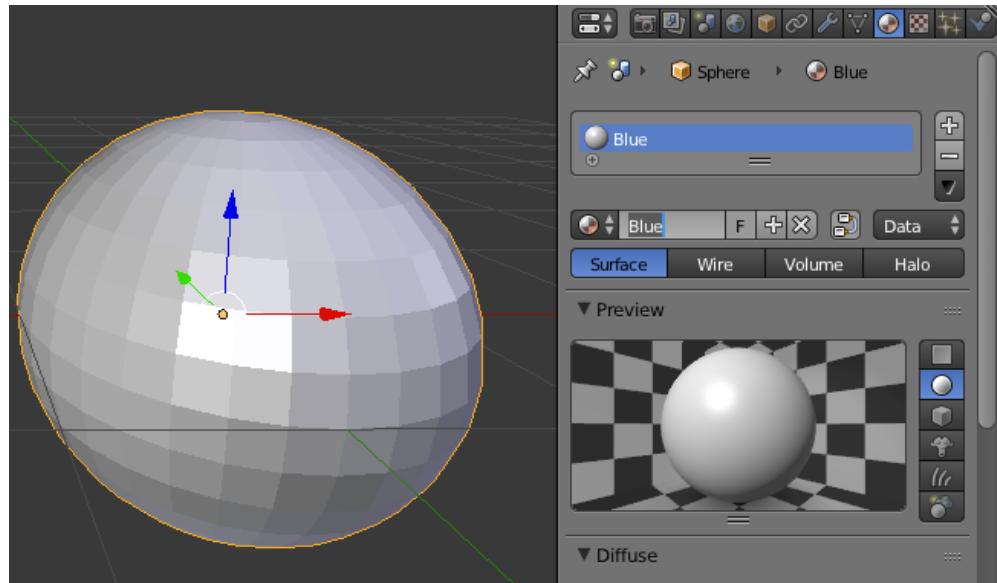
Để tô màu cho toàn bộ object ta thực hiện theo quy trình sau:

Bước 1: Chọn đối tượng và click vào giao diện tùy chỉnh material



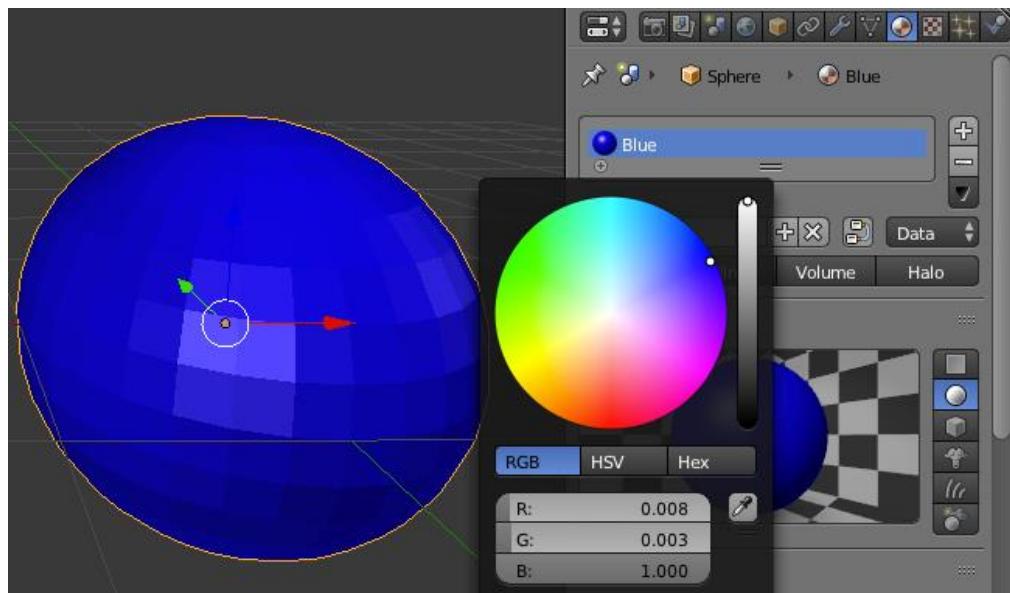
Hình 52 – Sử dụng material (P1)

Bước 2: Chọn nút New, một material mới sẽ được khởi tạo. Có thể đặt tên cho material đó để có thể quản lý và tái sử dụng. Vì material cũng như áo, quần, son môi, ... có thể áp dụng đối với nhiều đối tượng khác nhau



Hình 53 – Sử dụng material (P2)

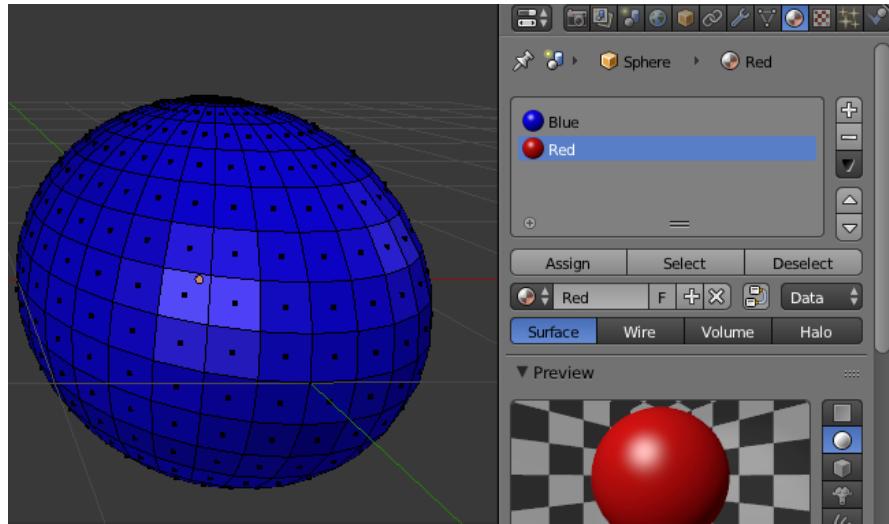
Bước 3: Click vào thanh màu ở phần Diffuse để chọn màu cần thiết



Hình 54 – Sử dụng material (P3)

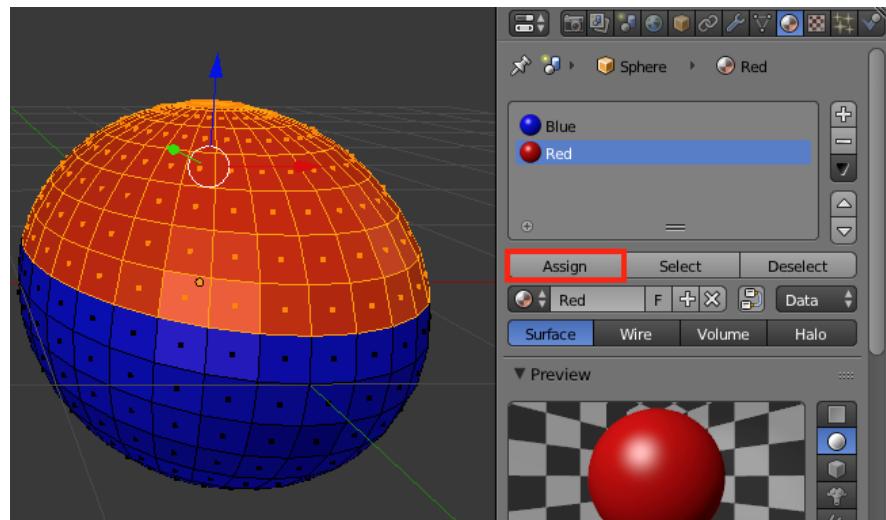
Một đối tượng không chỉ có một mà có hai hoặc rất nhiều màu, để thêm các màu khác cho đối tượng, ta cần thực hiện tiếp các bước sau:

Bước 4: Chọn dấu '+' để thêm một material mới (tương đương với bước 1) và tiếp tục thực hiện bước 2 và 3



Hình 55 – Sử dụng material (P4)

Bước 5: Chọn các mặt phẳng cần tô màu thứ hai của đối tượng và bấm Assign để gán màu đó cho mặt phẳng đã chọn

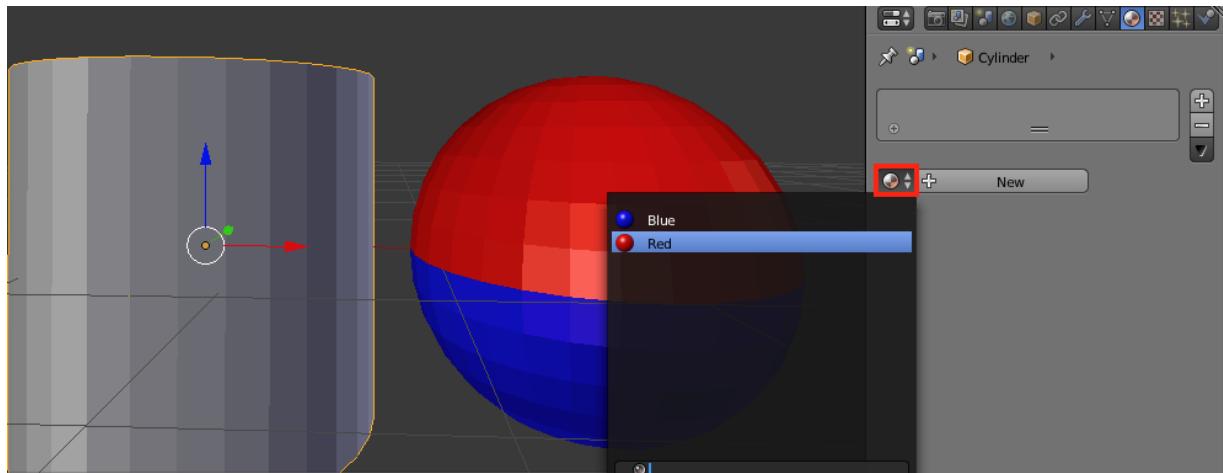


Hình 56 – Sử dụng material (P5)

Tiếp tục lặp lại bước 4 và 5 nếu đối tượng có nhiều màu khác nhau.

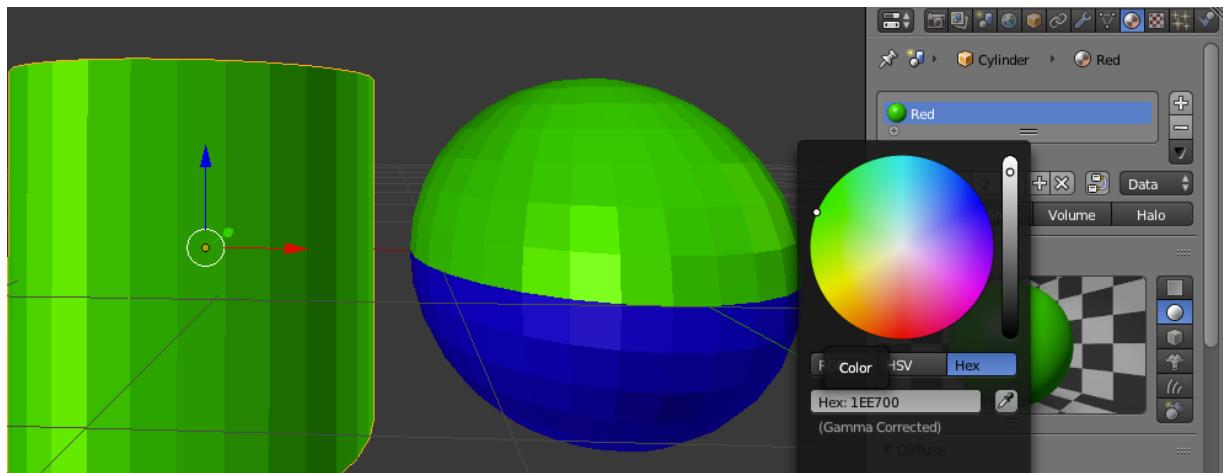
Để tái sử dụng các material đã tạo, ta thực hiện như sau:

- Tạo một đối tượng mới
- Chọn biểu tượng material ở bên dưới. Một danh sách các material đã tạo trước đó sẽ hiện ra. Ta chỉ việc chọn vào material muốn tái sử dụng.



Hình 57 – Sử dụng material (P6)

Nhưng do tính tái sử dụng, cả 2 đối tượng đều dùng chung một material, nên việc thay đổi material sẽ làm thay đổi màu sắc của cả 2 đối tượng. Vì vậy nên cẩn thận khi quyết định cho 2 đối tượng dùng chung một material.



Hình 58 – Sử dụng material (P7)

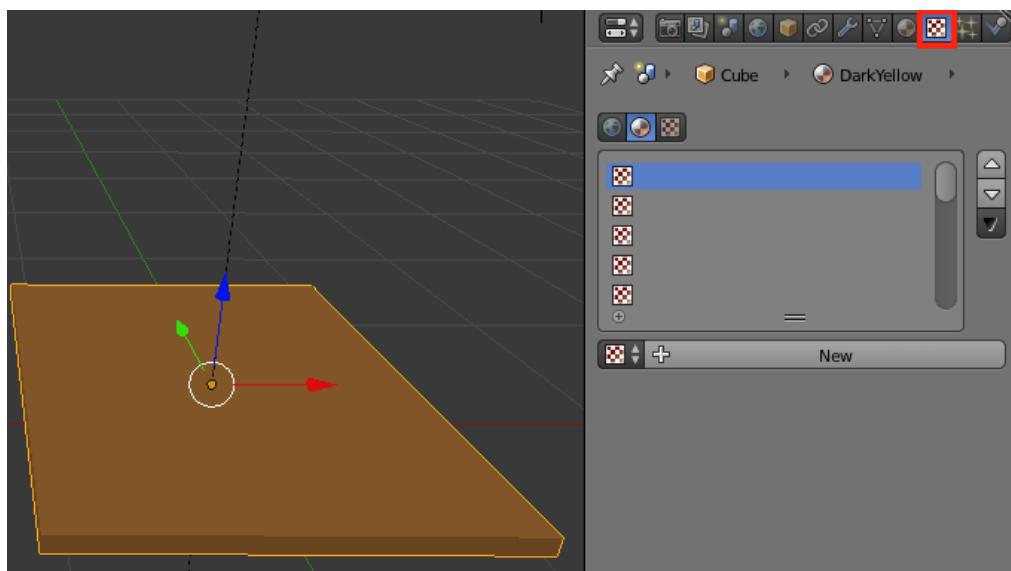
2.4.5.2 Texture

Nếu như material là một chiếc áo trơn, chỉ toàn màu rắn, rất ít hiệu ứng hay họa tiết trang trí thì texture là các đường nét, hoa văn, họa tiết khác nhau giúp chiếc áo thêm bắt mắt người dùng.

Để thêm texture, ta thực hiện các bước sau:

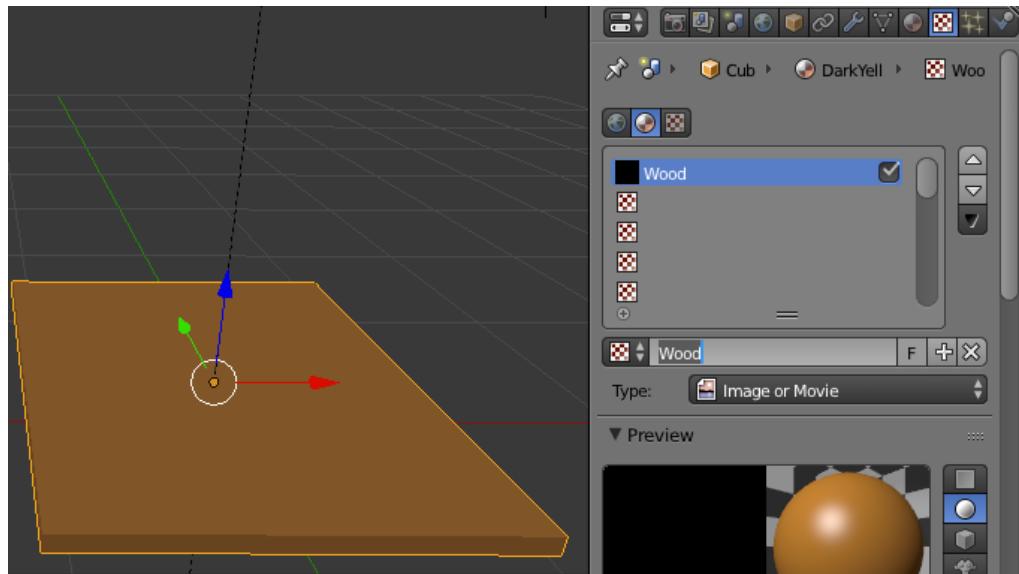
Bước 1: Chọn đối tượng và chọn material cho đối tượng

Bước 2: Chọn giao diện tùy chỉnh texture



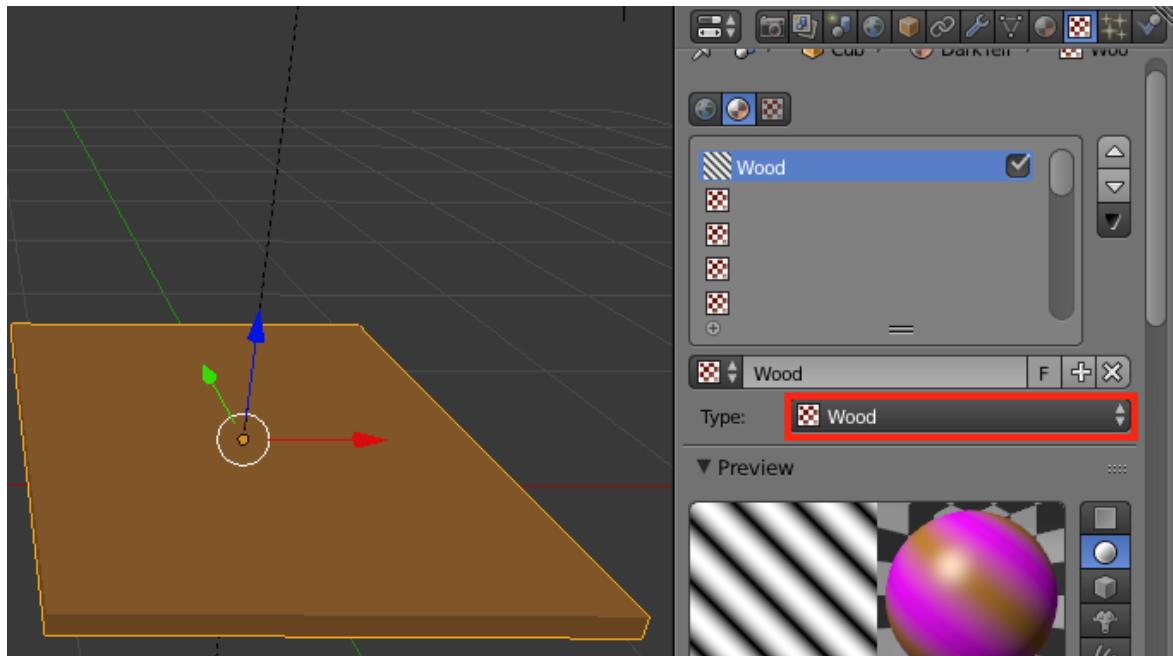
Hình 59 – Sử dụng texture (P1)

Bước 3: Chọn nút New để tạo một texture mới và đặt tên cho texture đó để dễ quản lý và tái sử dụng



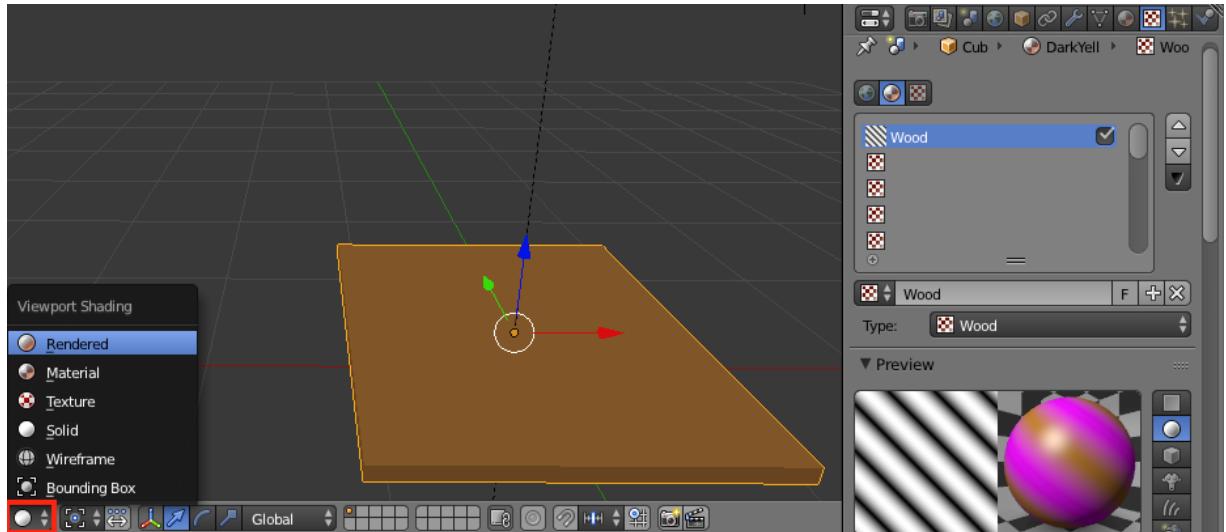
Hình 60 – Sử dụng texture (P2)

Bước 4: Chọn kiểu hoa văn của texture ở mục Type. Mỗi kiểu hoa văn sẽ có những đặc điểm riêng.



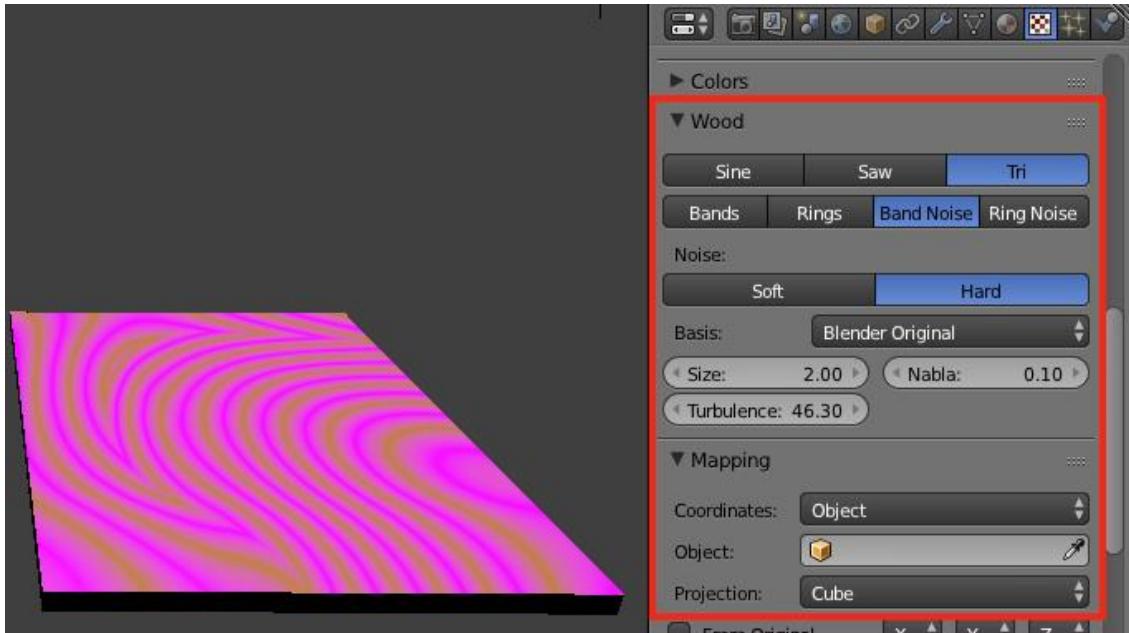
Hình 61 – Sử dụng texture (P3)

Bước 5: Để có thể thấy được sự thay đổi của texture, ta phải bật chế độ hiển thị render (mặc định là solid)



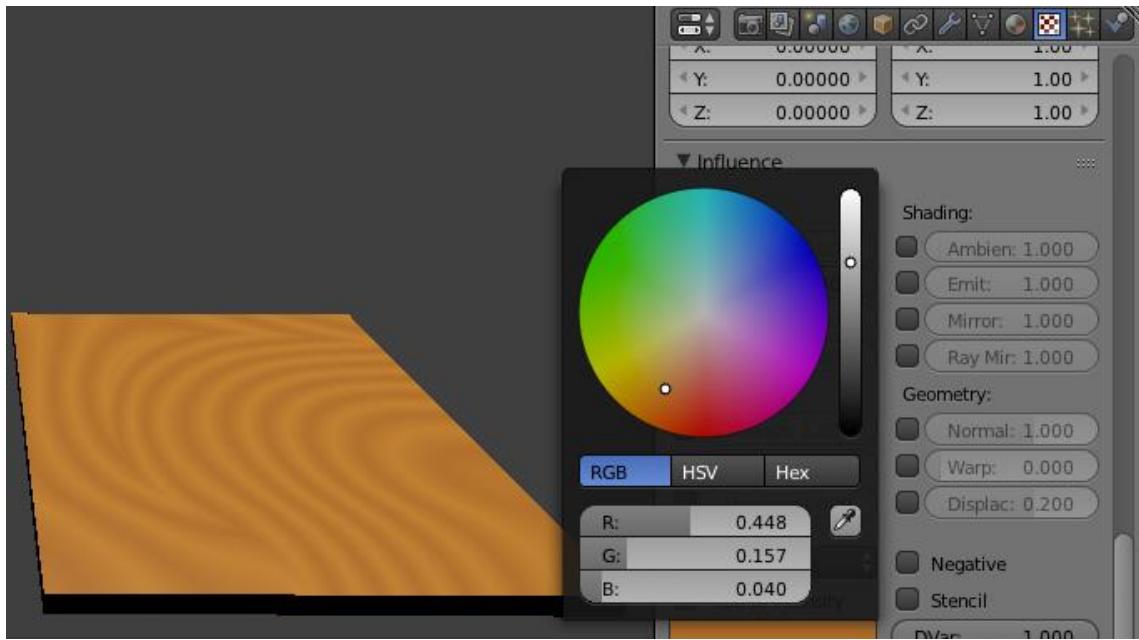
Hình 62 – Sử dụng texture (P4)

Bước 6: Thay đổi thuộc tính của texture để tìm được hoạ tiết vừa ý nhất



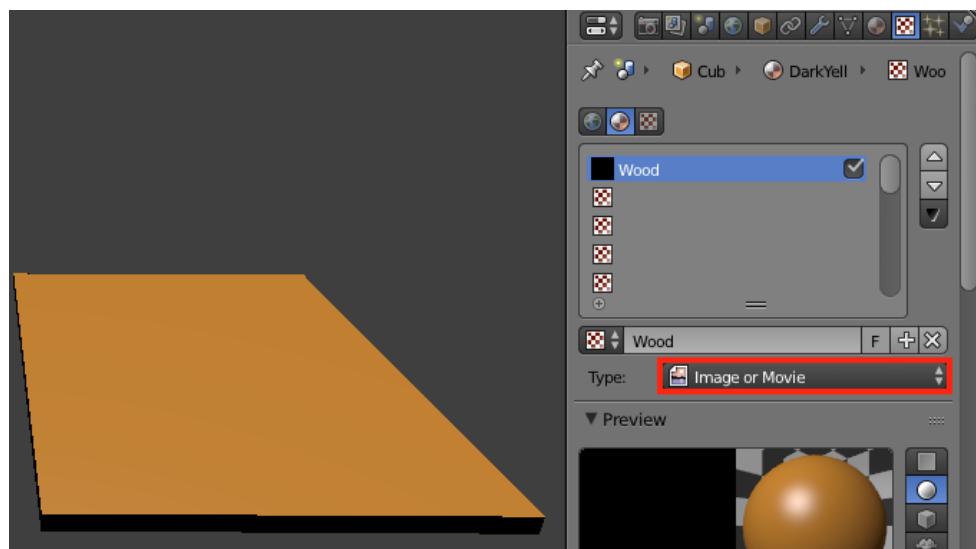
Hình 63 – Sử dụng texture (P5)

Bước 7: Thay đổi màu của phần còn lại ở phần Influence



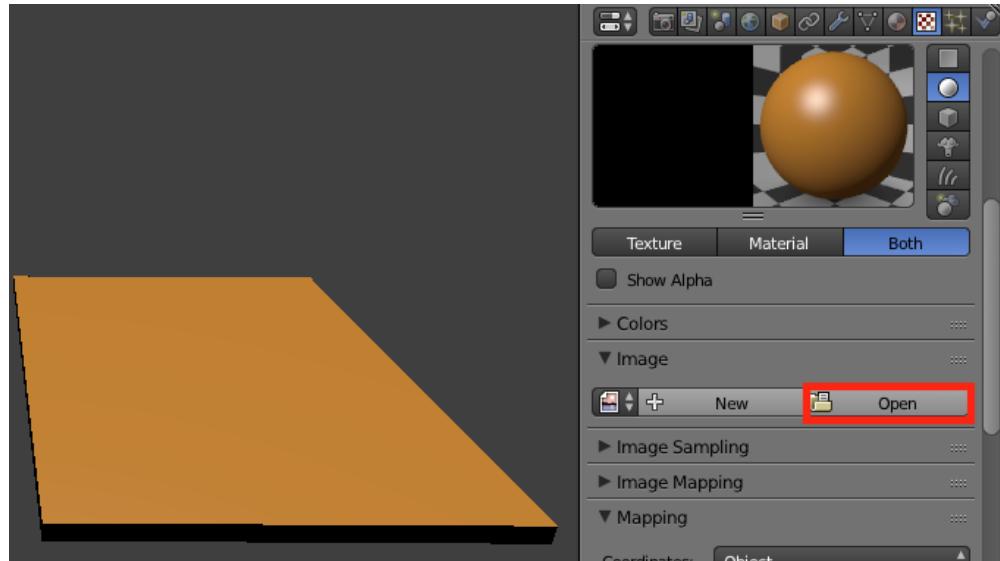
Hình 64 – Sử dụng texture (P6)

Khi chọn loại texture, ta có 2 loại chính. Loại thứ nhất là hoa văn do chính người dùng tự tạo, đó chính là ví dụ ở trên. Loại còn lại là phủ một lớp hình ảnh có sẵn lên đối tượng để tạo hiệu ứng hoa văn. Trở lại bước 4 bên trên, thay vì chọn Wood, ta sẽ chọn Image and Movie



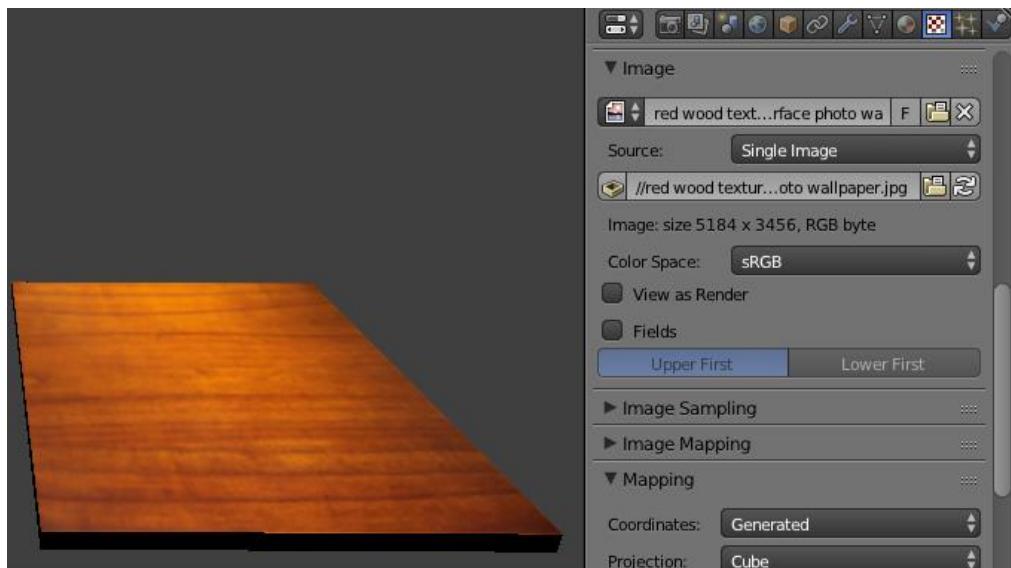
Hình 65 – Sử dụng texture (P7)

Bước 5’: Tải một tập tin hình ảnh về máy. Có thể dùng google hình ảnh để tìm kiếm từ khoá “texture”. Ví dụ: “Wood texture”. Sau đó ở mục Image chọn Open



Hình 66 – Sử dụng texture (P8)

Bước 6’: Chọn hình ảnh đã tải về và hướng thụ kết quả



Hình 67 – Sử dụng texture (P9)

2.5 Quá trình xây dựng đồ án

2.5.1 Đo đạc số liệu

Việc đo đạc số liệu rất quan trọng, là nền tảng cho một kiến trúc chuẩn, đẹp mắt, đồng thời cũng giúp người thiết kế đỡ nhọc công, không phải thay đổi nhiều trong quá trình thiết kế.

Để đo đạc số liệu nhóm đã kết hợp 3 phương án sau: đo bằng số ô gạch đối với kích thước toà nhà, đo bằng số liệu thật tế (theo đơn vị cm) đối với các thiết bị có thể tiếp cận được và sử dụng phương pháp ước lượng để có thể vẽ vật thể một cách chuẩn xác nhất.

2.5.2 Những đối tượng đầu tiên

Đối tượng đầu tiên mà nhóm vẽ là viên gạch và cánh cửa. Vì viên gạch là vật thể hình vuông và cả viên gạch và cánh cửa đều có thể lấy số liệu dễ dàng, kết hợp hai điều này, ta có thể quyết định cấu trúc của toà nhà một cách dễ dàng.



Hình 68 – Những đối tượng đầu tiên để đặt tiêu chuẩn kích thước

Ví dụ: một ô gạch có kích thước 40x40cm, vậy ch่อง 11 viên gạch theo chiều cao sẽ có được kích thước của một cánh cửa theo thực tế (440cm).

Ví dụ khác: để vẽ một số máy móc trong phòng, nhóm sẽ so sánh nó với nhiều cao của cánh cửa, cao bằng 2/3 hoặc phân nửa, hoặc cao hơn 10cm, ...

2.5.3 Quy trình mô phỏng tòa nhà

Để có thể hoàn thành mô phỏng được tòa nhà C trường đại học Tôn Đức Thắng, nhóm đã quyết định sẽ bắt đầu từ những gì dễ nhìn thấy và nhỏ nhặt nhất. Cụ thể như sau: viên gạch -> cánh cửa -> tường -> thiết bị máy móc bên trong -> thiết bị máy móc bên ngoài hành lang. Cứ thế áp dụng từ tầng trệt đến tầng 6 của tòa nhà C. Và cuối cùng chính là khuôn viên trường đại học Tôn Đức Thắng.

Đó là quy trình mà nhóm rút ra được sau khi hoàn thành tòa nhà. Việc sai thứ tự có thể dẫn đến việc phải sửa đổi rất hao tốn thời gian trong tương lai.

Ví dụ: vẽ khuôn viên trước, sau đó bắt đầu vẽ tòa nhà, việc đó sẽ gây ra không đủ không gian, các thiết bị rất nhỏ => phải sửa lại khuôn viên

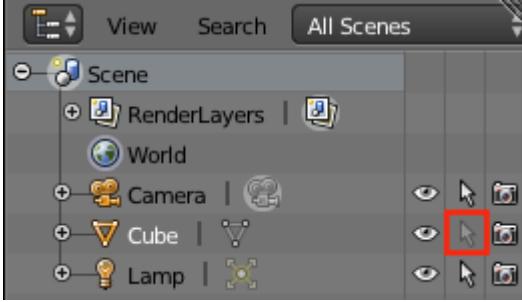
Ví dụ 2: vẽ máy móc trước: khi đặt vào căn phòng có thể phóng to, thu nhỏ, nhưng điều đó có thể gây ra việc thiết bị hoặc quá to, hoặc quá nhỏ, nhìn không chính xác so với thật tế.

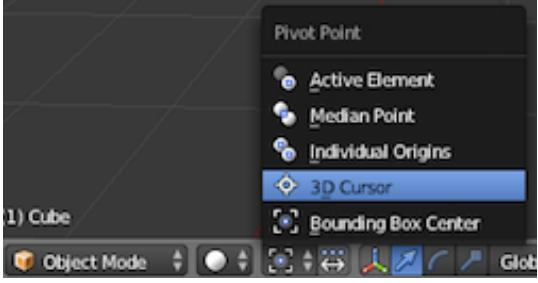
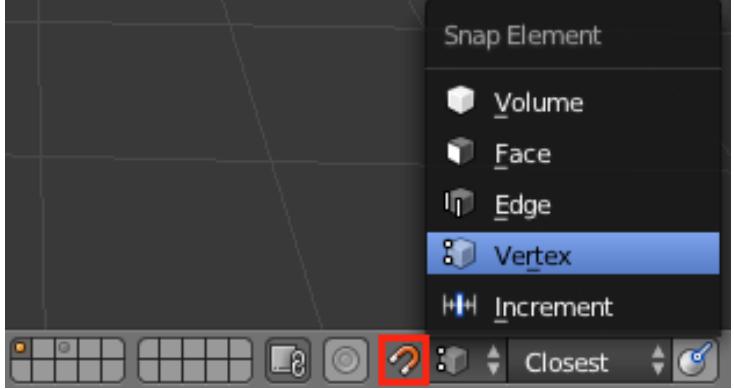
2.5.4 Những khó khăn gặp phải và cách khắc phục

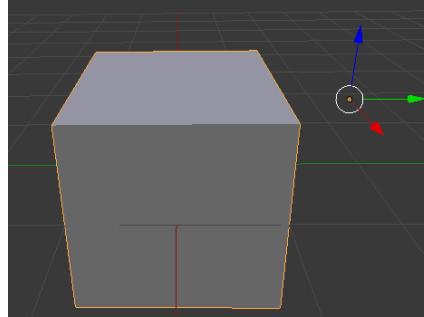
Trong quá trình thao tác với Blender cũng như thu thập dữ liệu để vẽ thực tế, nhóm đã gặp không ít khó khăn.

Khó khăn	Kinh nghiệm
Vấn đề di chuyển camera trong blender	Thật tế không những phức tạp mà còn rất dễ, kết hợp các phím số và các nút ctrl, alt, shift như đã nêu bên trên có thể dễ dàng di chuyển camera đến bất kỳ vị trí nào.
Không thể Scale hay rotate đối tượng	Có thể trong quá trình thao tác, ta đã vô tình chọn vào biểu tượng dưới đây, chỉ cần bấm lại là được.



Không thể di chuyển hay chọn vào đối tượng	<p>Có thể trong quá trình thao tác, ta đã vô tình chọn vào biểu tượng dưới dây, nhằm mục đích khoá đối tượng, giúp hạn chế tuyệt đối sự thay đổi hay chỉnh sửa.</p> 
Không xuất hiện 3 trục để thay đổi vị trí đối tượng	<p>Trong trường hợp khi chọn vào đối tượng mà không thấy biểu tượng này:  nghĩa là đã vô tình tắt nó đi, cần bật lại bằng cách chọn vào biểu tượng dưới đây:</p> 
Project quá lag, không thể thoát khỏi thiết kế	<p>Có 3 cách giải quyết điều này:</p> <ul style="list-style-type: none"> - Cách 1: thiết kế ở một project khác rồi sử dụng phím tắt 'shift' + 'f1' để import đối tượng vào lại. - Cách 2: ẩn các đối tượng không cần thiết, sử dụng phím tắt 'h', sau khi thiết kế xong thì hiện tất cả lên lại bằng cách sử dụng phím tắt 'alt' + 'h' - Cách 3: di chuyển đối tượng đang thiết kế xong một ô khác bằng chọn đối tượng và sử dụng phím tắt 'm' cùng một con số bất kỳ. Hình bên dưới, đối tượng đã được di chuyển sang ô thứ 3, ô số 1 và 3 xuất hiện biểu tượng chấm tròn để báo hiệu có đối tượng tồn tại trong đó.

	
Đối tượng nằm quá xa nơi cần kéo đến	<p>Có 3 cách giải quyết vấn đề này:</p> <ul style="list-style-type: none"> - Cách 1: Chọn vị trí trực kéo theo vị trí của 3D Cursor (click chuột trái) bằng cách thay đổi ở đây  <ul style="list-style-type: none"> - Cách 2: sử dụng nam châm để kéo vật thể cần di chuyển lại sát đỉnh/cạnh/mặt của vật thể đích bằng cách bật cách thay đổi ở đây  <ul style="list-style-type: none"> - Cách 3: di chuyển vật thể đến vị trí của 3D cursor (click chuột trái) bằng cách sử dụng phím tắt ‘shift’ + ‘s’ và chọn “Selection to Cursor (offset)”
Không biết đến sự tồn tại của trực	Việc không biết sử dụng trực gây ra việc mất rất nhiều thời gian cho việc thiết kế. Để có thể chỉnh trực cho đối tượng, ta chọn đối tượng rồi sử dụng phím tắt ‘ctrl’ +

	<p>‘shift’ + ‘alt’ + ‘c’ và chọn vị trí của trục, thông thường là Origin to 3D Cursor và Origin to Center of Mass.</p> <p>Vậy trục gây ra những khó khăn gì ?</p> <p>Khó khăn 1: khi xoay một đối tượng, đối tượng sẽ xoay theo trục, áp dụng vào cánh cửa, nếu trục của cánh cửa không nằm ngay mép cửa thì cánh cửa sẽ xoay hoàn toàn sai</p> <p>Khó khăn 2: nếu trục không nằm giữa đối tượng, khi scale sẽ không đồng đều, một bên dài hơn bên còn lại, hoặc khiến đối tượng vừa di chuyển vừa scale</p>  <p>Trục là biểu tượng chấm nhỏ màu vàng, khi bấm vào đối tượng sẽ xuất hiện.</p>
Vị trí khởi tạo đối tượng không đúng vị trí cần thiết	<p>Cần biết vị trí đối tượng được khởi tạo ra sẽ nằm ở vị trí của 3D Cursor, việc đặt 3D Cursor ở đâu cũng tiết kiệm được khá nhiều thời gian trong việc thiết kế. Cụ thể:</p> <p>Ví dụ 1: đặt sai vị trí 3D Cursor làm cho vật thể tạo ra nằm cách xa vị trí cần thiết</p> <p>Ví dụ 2: Nếu muốn tạo một vật thể ở đúng vị trí chính giữa của một vật thể nào đó ta thực hiện như sau: chọn vật thể gốc -> chỉnh trục của vật thể về chính giữa Origin to Center of Mass -> ‘shift’ + ‘s’ chọn “cursor to selected” để di chuyển 3D cursor về trục của vật thể đã chọn -> tạo vật thể mới.</p>

Một số hiệu ứng không hoạt động khi chuyển từ blender sang unity	Cụ thể: hiệu ứng trong suốt (transparent), kính đối chiếu (mirror) không hoạt động. Texture tự tạo không sử dụng được ở unity và texture hình ảnh phải được xử lý ở unity một lần nữa mới có thể hiển thị được ở unity.
Dung lượng cao	Hạn chế sử dụng subdivision surface hết mức có thể. Vì có thể tạo ra những đôi tượng sắc nét, đẹp đẽ nhưng tăng dung lượng đáng kể. Một project lớn lên tới 1 Gb sẽ gây trở ngại cho quá trình thiết kế (lag). Một phương pháp nữa để giảm dung lượng đó là xoá các đỉnh trùng lặp với nhau (remove double). Khi thiết kế, có đôi lúc ta để các đỉnh đè lên nhau nhưng không hay biết, gây tăng dung lượng. Ta có thể xoá bằng cách chọn object, bấm ‘tab’ để bật chế độ edit mode, bấm tổ hợp ‘ctrl’ + ‘v’ và bấm tiếp ‘d’ hoặc chọn “Remove Double”.
Nhà trường không cho phép chụp ảnh	Vì ván đề chụp ảnh là một ván đề nhạy cảm, gây ảnh hưởng đến sự riêng tư, cũng như để đề phòng các bức ảnh bôi bát, gây ảnh hưởng đến danh tiếng nhà trường, khi chụp ảnh cần phải biết những điều sau: - Không chụp ảnh khi chưa xin phép nhà trường - Xin phép giám thị trước khi chụp - Nếu tình hình quá căng thẳng cần liên hệ văn phòng khoa để làm đơn xin phép và cam kết

Bảng 5 – Những khó khăn và cách khắc phục

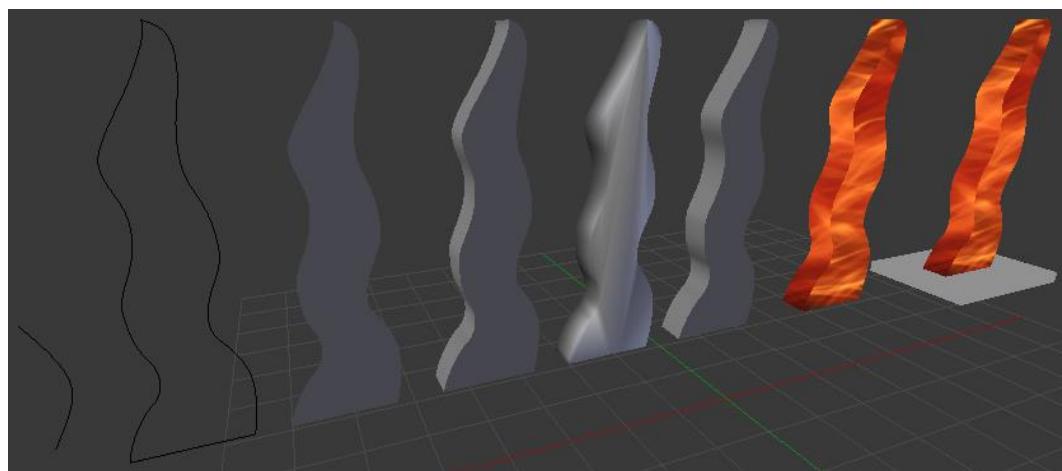
2.6 Một vài ví dụ

Dưới đây là một vài ví dụ trong số những đối tượng được vẽ để mô phỏng trường đại học Tôn Đức Thắng.

Ví dụ 1: tác phẩm tượng đá nghệ thuật của trường Tôn Đức Thắng



Hình 69 – Hình ảnh tượng đá nghệ thuật thật tế và mô phỏng

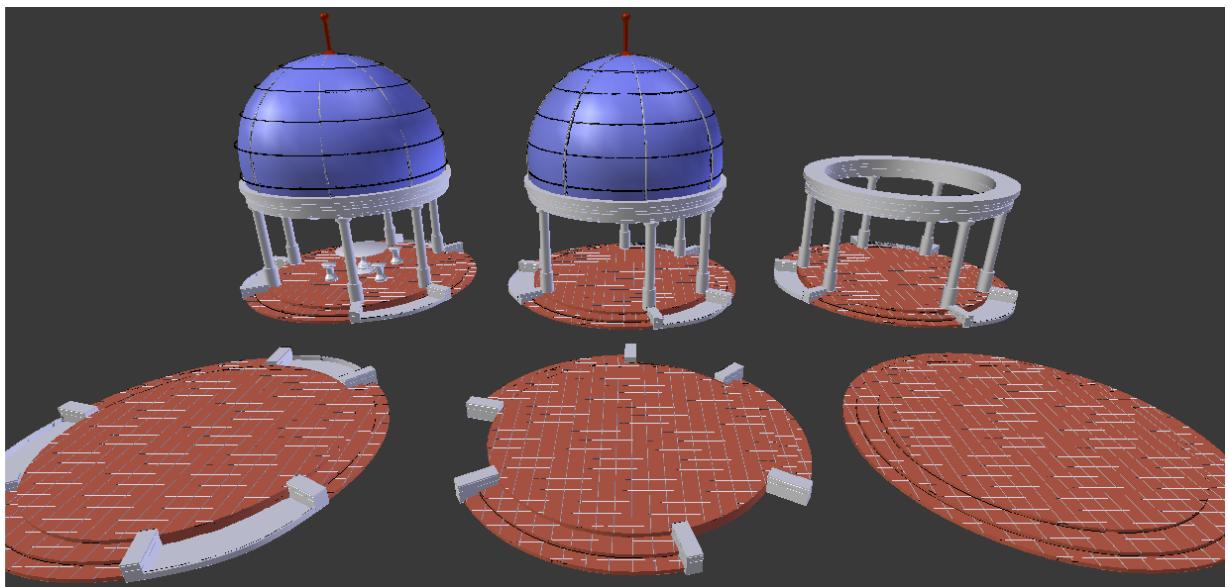


Hình 70 – Quá trình mô phỏng tượng đá nghệ thuật

Ví dụ 2: Nhà mái vòm tròn để sinh viên nghỉ chân



Hình 71 – Hình ảnh nhà mái vòm tròn thật tế và mô phỏng

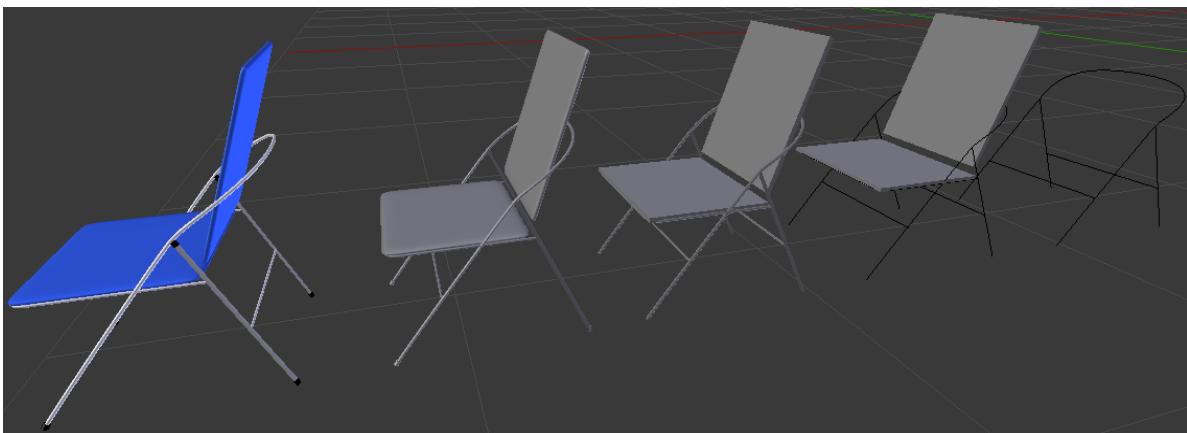


Hình 72 – Quá trình mô phỏng nhà mái vòm tròn

Ví dụ 3: ghế ngồi ở văn phòng khoa



Hình 73 – Hình ảnh ghế ngồi thật tế và mô phỏng



Hình 74 – Quá trình mô phỏng ghế ngồi

CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG BẰNG UNITY

Sau khi hoàn tất công việc dựng mô hình trường Tôn Đức Thắng, phần tiếp theo sẽ trình bày áp dụng mô hình vào công cụ Unity Engine và xây dựng thành một ứng dụng VR.

3.1 Giới thiệu về Unity

3.1.1 Tổng quan

Unity hay Unity3D là một game engine đa nền tảng được phát triển bởi Unity Technologies cho phép sử dụng để phát triển game cho máy tính (PC), consoles (PSx, xBox...), thiết bị di động (iOS, Android, WindowPhone...) và Web. Engine này được phát triển bằng C/C++ và có khả năng hỗ trợ mã viết bằng C#, JavaScript hoặc Boo. Đây là một trong 4 engine game phổ biến nhất thế giới, hiện đã phát hành bản Unity3D 5.5

Unity lần đầu được công bố chỉ sử dụng trên OS X, tại Hội Nghị Phát Triển Toàn Thế Giới của Apple vào năm 2005. Nó đã được mở rộng để nhắm tới mục tiêu trên 27 nền tảng khác nhau, mới nhất gần đây là các nền tảng thực tế ảo (Virtual Reality). Nintendo đã cung cấp giấy phép Unity 5 cho tất cả nhà phát triển Nintendo cùng với bộ công cụ phát triển phần mềm của họ (SDKs) cho hệ máy Wii U và Nintendo 3DS.

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

Điều đáng chú ý ở Unity là nhắm tới xây dựng các game cho nhiều nền tảng. Trong một dự án, nhà phát triển có thể kiểm soát và phân phối đến các thiết bị di động, trình duyệt web, máy tính và consoles.

3.1.2 Lịch sử phát triển

Vào đầu những năm 2000, ba lập trình viên trẻ với nguồn kinh tế eo hẹp đã tập trung tại một tầng hầm và bắt đầu lập trình ra thứ mà sau này trở thành một trong những phần mềm được ứng dụng rộng rãi nhất trong ngành công nghiệp video game.

Một thập kỷ sau, Unity đã trở nên phổ biến hơn khi được các nhà phát triển game sử dụng một cách rộng rãi để làm hàng ngàn game cho các thiết bị di động, máy tính, web...

Năm 2008, các nhà phát triển gấp rút xử lý để hỗ trợ cho iPhone và trở thành game engine cho iOS đầu tiên trên thế giới. Cùng thời gian này, Cartoon Network đã sử dụng Unity để tạo ra FusionFall, một MMORPG cho trẻ em đã có được 8 triệu người chơi. Electronic Arts sử dụng Unity3D trong năm 2009 để làm Tiger Woods PGA Tour Online, và thậm chí cả Microsoft và Ubisoft đã trở thành khách hàng của Unity3D.

Trong năm 2011, Unity đã mua một công ty hoạt hình có tên Mecanim, thúc đẩy công nghệ cơ bản của game engine.

Ngày nay, Unity và 285 nhân viên của mình trên khắp thế giới hỗ trợ cho phát triển iOS, Android, Windows, Mac, Linux, trình duyệt Web, PS3, Xbox 360, Wii U, Sony PlayStation Vita, Windows Phone, BlackBerry. Trên 1,8 triệu nhà phát triển sử dụng Unity, Plugin trình duyệt của phần mềm đã được cài đặt hơn 200 triệu lần. Dead Trigger, Dead trigger 2 và Song Of Knight... là những trò chơi có đồ họa phức tạp nhất cho iOS và Android, được dựa trên Unity3D.

3.1.3 So sánh Unity với các game engine khác

Unity chắc chắn không phải engine đỉnh cao nhất về mặt đồ họa. Về mặt này, Cry Engine vẫn đang dẫn đầu với ưu thế đồ họa 3D cực kỳ chân thực. Có thể cảm nhận rõ ràng điều này qua chất lượng hình ảnh các tựa game gần đây sử dụng Cry Engine như Far Cry hay Crysis 3. Hoặc engine phổ biến nhất cho PC Window là DirectX với các tựa game đình đám như Assassin's Creed, Batman Arkham...

Tuy nhiên, những engine khủng như Cry rất kén chọn và yêu cầu cấu hình cũng khủng không kém. Hơn nữa cái giá để được cấp phép sử dụng Cry Engine chắc chắn cũng không dễ chịu chút nào. Hay như DirectX lại yêu cầu một kỹ năng lập trình đồ họa khá phức tạp. Unity vẫn được các nhà phát triển game nhỏ ưa thích hơn nhờ vào các đặc điểm sau:

- Hỗ trợ đa nền tảng: bao gồm Android, Apple TV, BlackBerry 10, iOS, Linux, Nintendo 3DS line, macOS, PlayStation 4, PlayStation Vita, Unity Web Player (bao gồm Facebook), Wii, Wii U, Windows Phone 8, Windows, Xbox 360, and Xbox One.
- Dễ sử dụng: Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ.
- Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại.

- Làm game thời gian thực: Nhà phát triển không cần phải trang bị các kiến thức về lập trình đồ họa, cũng như không cần phải có hoa tay để phác thảo, vẽ vời trên các bản thiết kế. Unity cho phép thiết kế game theo kiểu Interface Graphic trực tiếp và có thể chạy thử demo trong quá trình xây dựng. Ngoài ra, Unity có thể sử dụng hình ảnh, đồ họa từ một phần mềm thứ 2 là các công cụ vẽ 3D như 3Dsmax, Blender, Maya, XSL, Cinema4D, Cheetah3D, Modo, Autodesk FBX, LightWave... Sau đó chỉ cần import vào Unity với định dạng tương ứng của phần mềm.

Có thể thấy, tuy không phải engine “khủng” nhất, nhưng nếu xét toàn diện, Unity3D đích thực là một trong những engine game đa nền tảng tốt nhất hiện nay.

3.1.4 Các phiên bản của Unity

Tính tới thời điểm đã có 5 phiên bản Unity được ra mắt. Các phiên bản đầu tiên như 1.x - 2.x, Unity còn rất đơn giản và gần như chỉ hướng đến các nhà làm game không chuyên với những khả năng đơn giản.

Với phiên bản Unity 3.5 trở đi, rất nhiều hiệu ứng hữu dụng được tạo ra bởi Particle System (Hệ thống tạo hạt). Nhiều hiệu ứng cần phải viết script đều được cung cấp trong hệ thống Shuriken particle.

Unity 4.0 lần đầu tiên được nhà sản xuất cam kết có thể dùng để phát triển một game hạng AAA với những tính năng mạnh mẽ không thua kém một engine nào trên thị trường.

Phiên bản mạnh nhất hiện nay là 5.5 với các tính năng được cải tiến như tạo các hoạt chuyển (Animation) đơn giản hơn nhờ vào hộp công cụ của Window, hỗ trợ âm thanh cho Window 10 và hoạt động tốt cho ứng dụng VR/AR, quản lý hiệu suất của game hiệu quả hơn, hỗ trợ nhiều định dạng đồ họa, cải tiến hệ thống tạo hạt (Particles), hỗ trợ viết script trên Visual Studio...

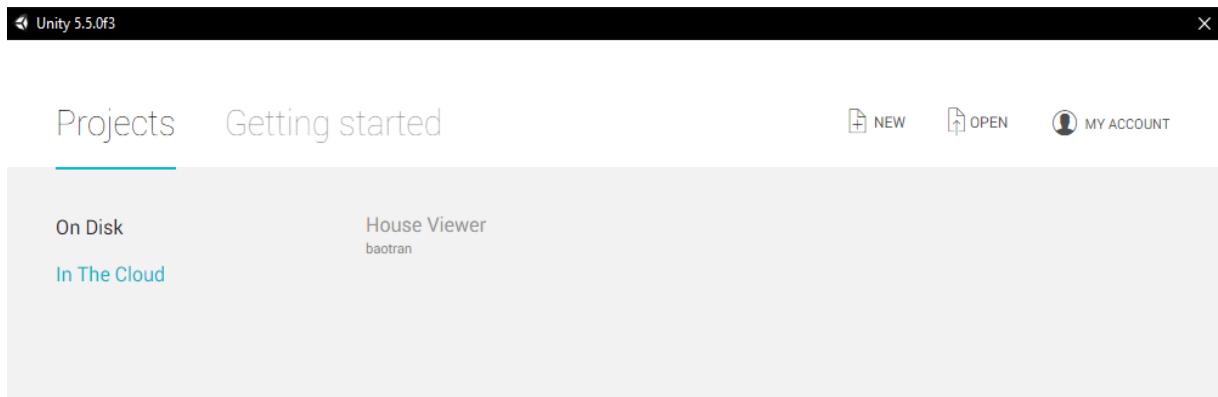
Ngoài ra, hiện tại cũng đang có phiên bản 5.6 nhưng vẫn đang trong giai đoạn beta (chưa chính thức).

3.2 Làm quen với Unity

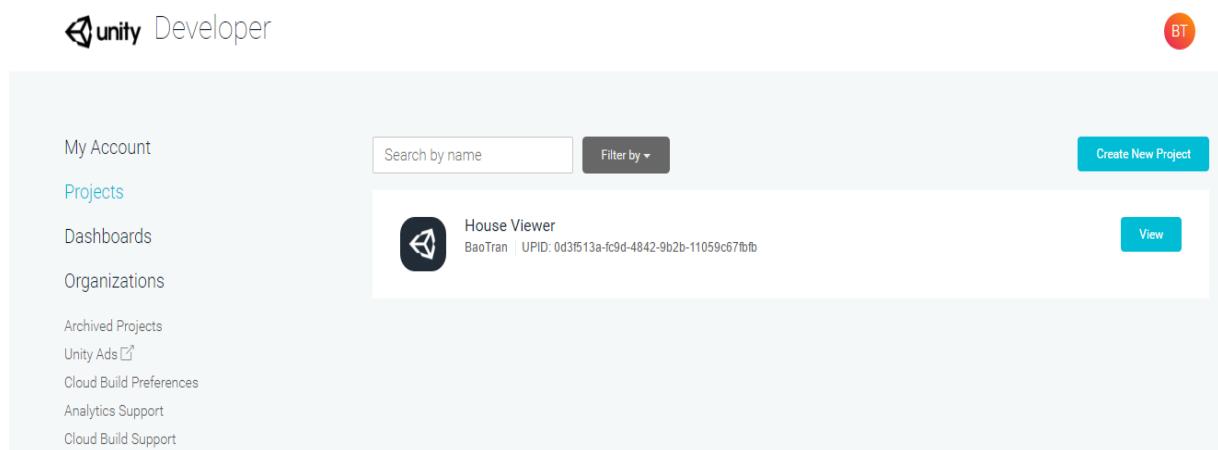
3.2.1 Giao diện người dùng

1. Khởi động chương trình

Khi khởi động chương trình sẽ hiện thị màn hình khởi động cho phép đăng nhập. Từ bản 5.5 trở lên, việc đăng nhập còn cho phép ta có thể đồng bộ và lưu trữ project lên server.



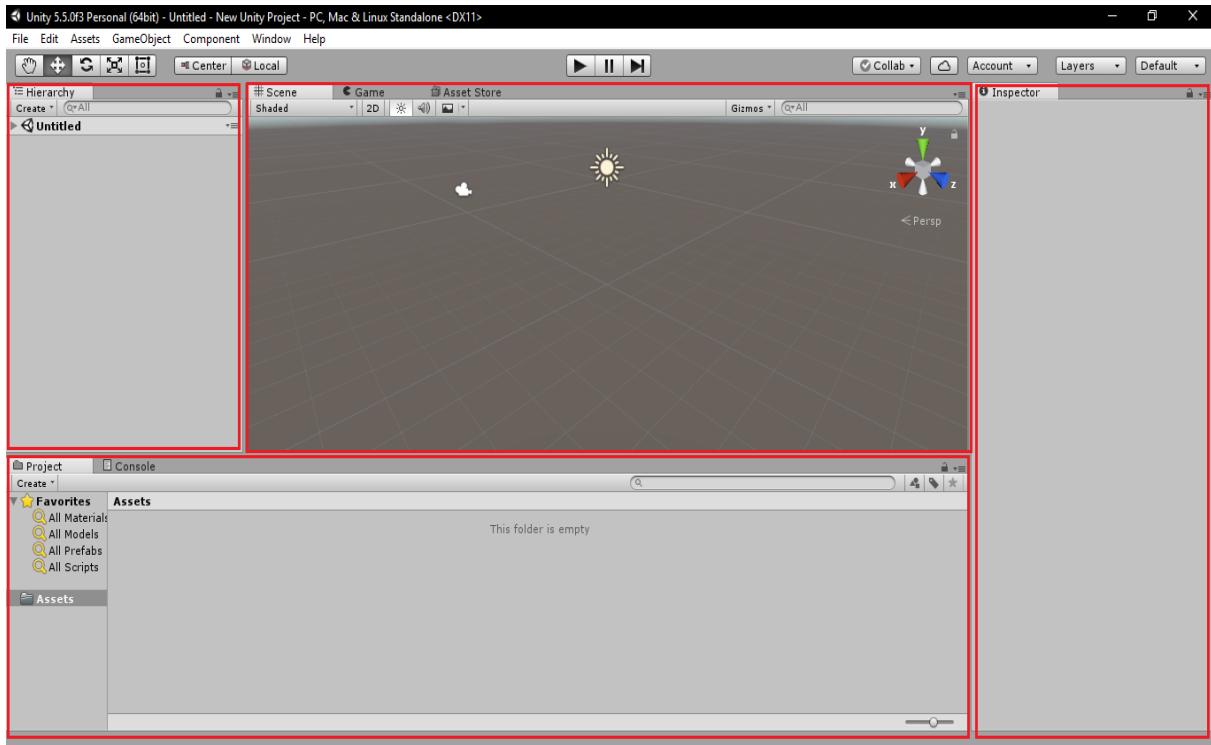
Hình 75 – Giao diện khởi động



Hình 76 – Project được lưu trữ trên web

2. Giao diện chính

Trên màn hình chính được chia thành nhiều Layout, mỗi Layout mang một chức năng khác nhau.



Hình 77 – Các layout cơ bản

- **Scene:** Là màn hình thiết kế, tại đây có thể kéo thả các vật thể game (GameObject) tùy ý tự do trong không gian 3 chiều để sắp xếp, bố cục các vật thể lại với nhau.
- **Game:** Màn hình chơi thử. Trong quá trình thiết kế, nhà phát triển có thể chơi thử game trực tiếp mà không cần phải xuất ra file thực thi. Mọi thao tác chơi game tùy thuộc vào cách xây dựng các kịch bản game (Script)
- **Hierarchy:** Là hệ thống phân cấp, tại đây chúc danh sách các GameObject đang được sử dụng trong màn hình thiết kế (Scene). Ta có thể nhóm các GameObject đó lại với nhau tạo thành cây thư mục, giúp dễ dàng trong việc quản lý.

- Inspector: Bản chất của Unity được xây dựng theo mô hình hướng đối tượng, vì thế mỗi GameObject sẽ có những thuộc tính nhất định (Mesh, Script, Âm thanh...) và được hiển thị cụ thể trong Inspector. Ta có thể chỉnh sửa các thuộc tính theo ý muốn thiết kế.
- Project: Cấu trúc cây thư mục dùng để quản lý hệ thống file của project. Trong hệ thống thư mục của một project Unity bất kỳ sẽ có nhiều thư mục khác nhau, tuy nhiên một số thư mục sẽ do Unity tự sinh ra, và riêng thư mục Asset sẽ do chính ta quản lý. Mọi thư mục và tập tin được lưu trong Asset sẽ hiển thị trên màn hình Project. Có thể thêm, xóa, sửa file trực tiếp trên đây. Các tập tin được chứa thường là các màn hình (scene), gameObject, hình ảnh, âm thanh, script...
- Console: Bảng điều khiển sẽ hiển thị các lệnh mà ta Debug hoặc sẽ hiện lỗi, cảnh báo nếu có phát sinh trong lúc chơi thử. Các lệnh console sẽ dẫn ta đến vị trí chính xác phát sinh lỗi để có thể khắc phục kịp thời.
- Ngoài các màn hình cần thiết kể trên còn có nhiều màn hình khác ta cũng quan tâm như Asset Store, Lighting, Service...

Ta có thể tùy biến giao diện bằng cách trên thanh công cụ toolbar, bấm chọn Button Layout và chọn kiểu tùy biến thích hợp. Hoặc đơn giản là kéo thả các màn hình layout đến vị trí mong muốn và sắp xếp theo ý mình sao cho thuận tiện nhất.

3.2.2 Một số khái niệm

Để làm việc dễ dàng với Unity ta cần nắm rõ một số khái niệm sau:

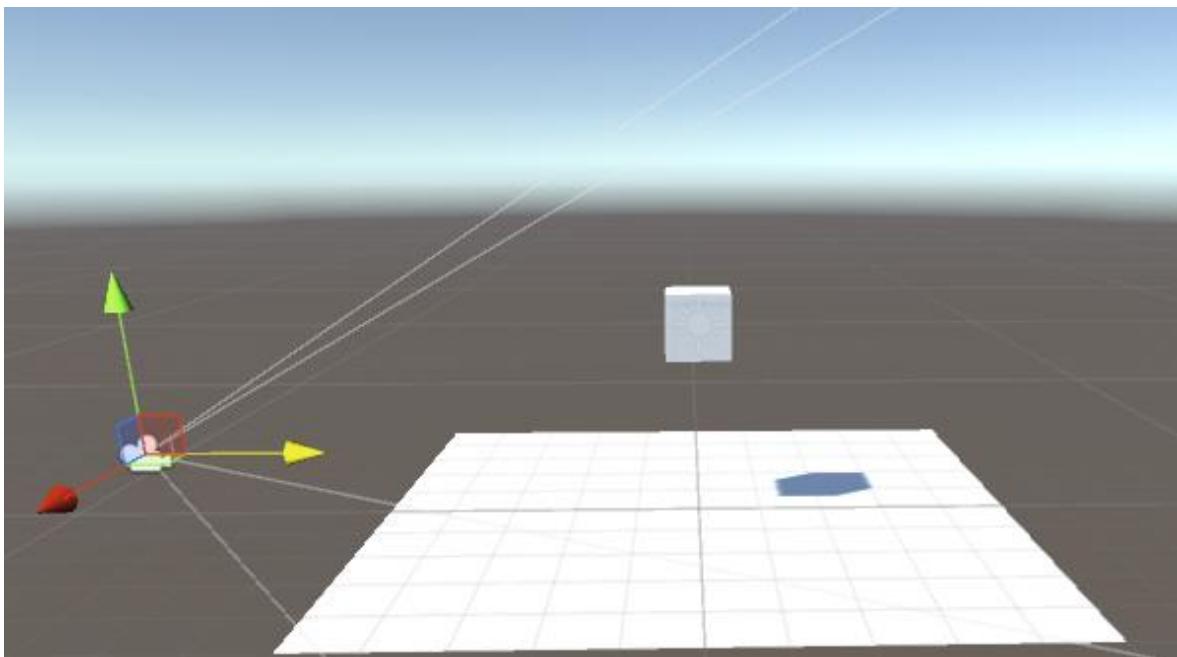
- Scene: Khác với Scene ở mục 3.2.1, Scene được hiểu như một cảnh chơi (màn chơi - level). Khi ta đã thiết kế game tại màn hình Scene, toàn bộ GameObject trong Project sẽ được lưu lại thành 1 file duy nhất có định dạng *.unity. Việc này cho phép ta có thể tạo ra nhiều cảnh, màn chơi, phân phối các thành phần trong game.
- GameObject: Là các đối tượng, vật thể trong game, ví dụ như các khối lập phương (Cube), khối trụ (Cylinder)... Hoặc có những GameObject phức tạp hơn được thiết kế bằng công cụ vẽ 3D và đưa vào Unity.
- Prefab: Giống như GameObject nhưng đặc tính khác là có thể tái sử dụng và thường được tạo bằng các công cụ thứ 2. Prefab có các thiết lập thuộc tính cố định vì thế ta có thể dễ dàng tái sử dụng nếu cần.
- Material: Tương tự như ở Blender, material sẽ cho phép phủ các vật liệu như màu sắc, hình ảnh, hoa văn vào GameObject, để giúp game trở nên sống động, chân thật hơn.
- Texture: Là những file hình ảnh hoa văn chi tiết được gắn vào Material
- Script: Là những đoạn mã được viết bằng ngôn ngữ C#, JavaScript hoặc Boo. Dùng để xử lý các sự kiện giữa các GameObject như bay, nhảy, va chạm...

3.2.3 Các thao tác cơ bản

Bắt đầu từ việc sau khi tạo Project, bố cục các layout theo ý muốn. Tiếp theo sẽ demo một ứng dụng cơ bản và build ra một số nền tảng.

1. Tạo GameObject và thiết kế Scene

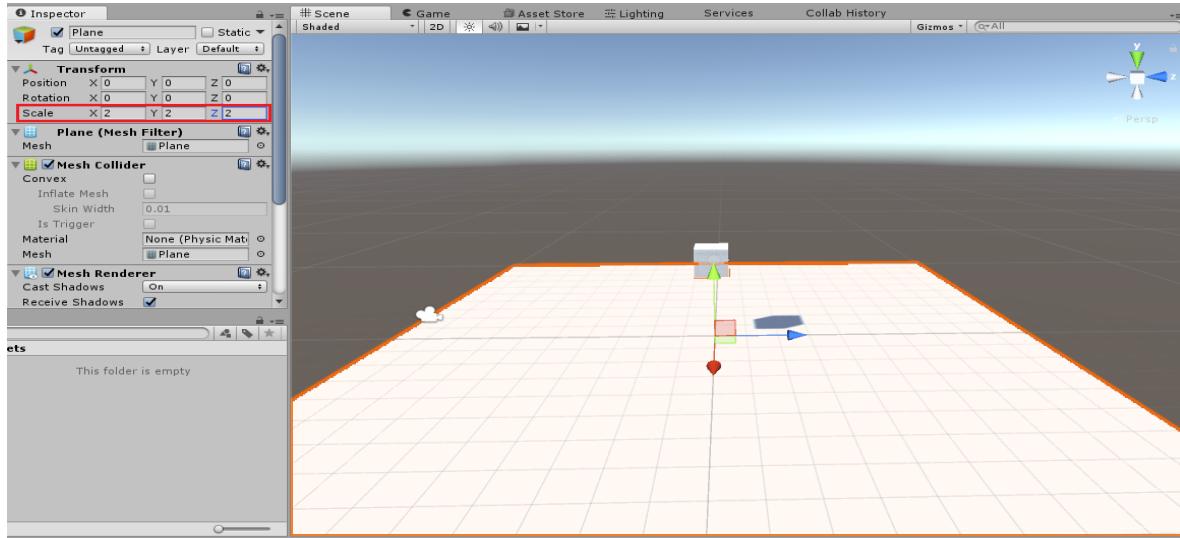
Trên thanh công cụ GameObject cho phép ta ra các GameObject nhiều loại như 3D (Cube, Plane, Cylinder...), 2D (Sprite), UI (Button, Image, Slider...)... Các GameObject sẽ được hiển thị trên màn hình Scene và danh sách các GameObject sẽ hiển thị ở Hierarchy. Trên màn hình Scene có thể sử dụng chuột kéo thả sắp xếp, bố cục theo ý tưởng của mỗi người.



Hình 78 – Các GameObject trên Scene

2. Thao tác với Inspector

Tại màn hình Inspector ta có thể tùy biến các thông số thuộc tính của từng loại GameObject như Position (vị trí), Rotation (hướng), Scale (tỉ lệ)...

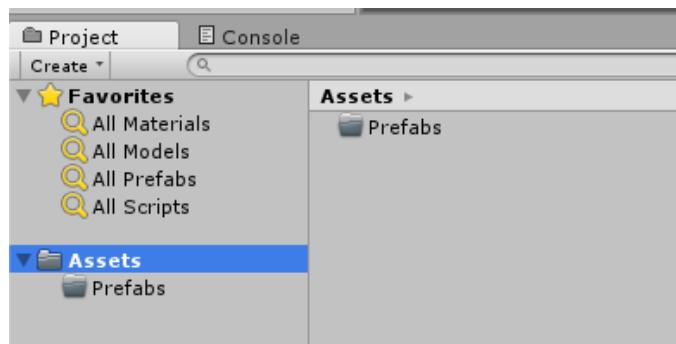


Hình 79 – Tùy chỉnh thuộc tính của GameObject

Ở đây thay đổi thông số Scale theo 3 chiều X,Y,Z để phóng to GameObject theo tỉ lệ.

3. Quản lý Project

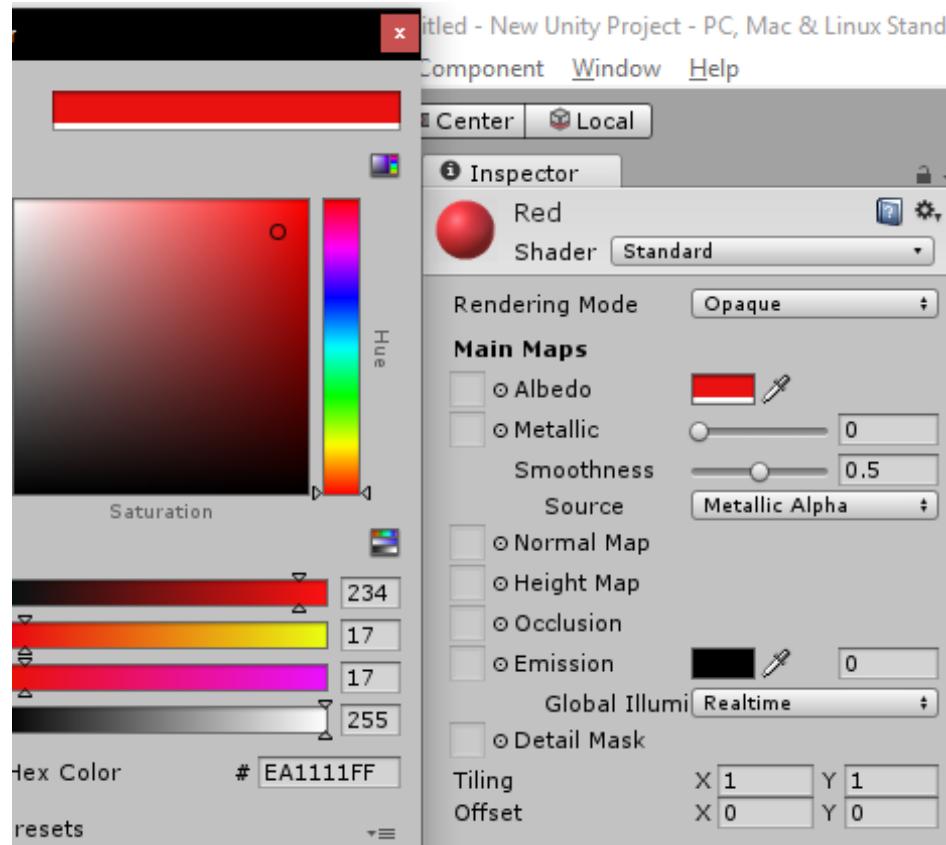
Toàn bộ cấu trúc thư mục và các file tài nguyên sẽ đều nằm trong thư mục Asset. Ta có thể thêm, xóa, sửa, sắp xếp bố cục các thư mục, file giúp dễ dàng trong việc quản lý tài nguyên. Một gợi ý trong việc quản lý Project hiệu quả là các tài nguyên nào thì nên đặt trong thư mục đó (thường là Prefabs, Materials, Audios, Textures, Script...)



Hình 80 – Quản lý thư mục Project

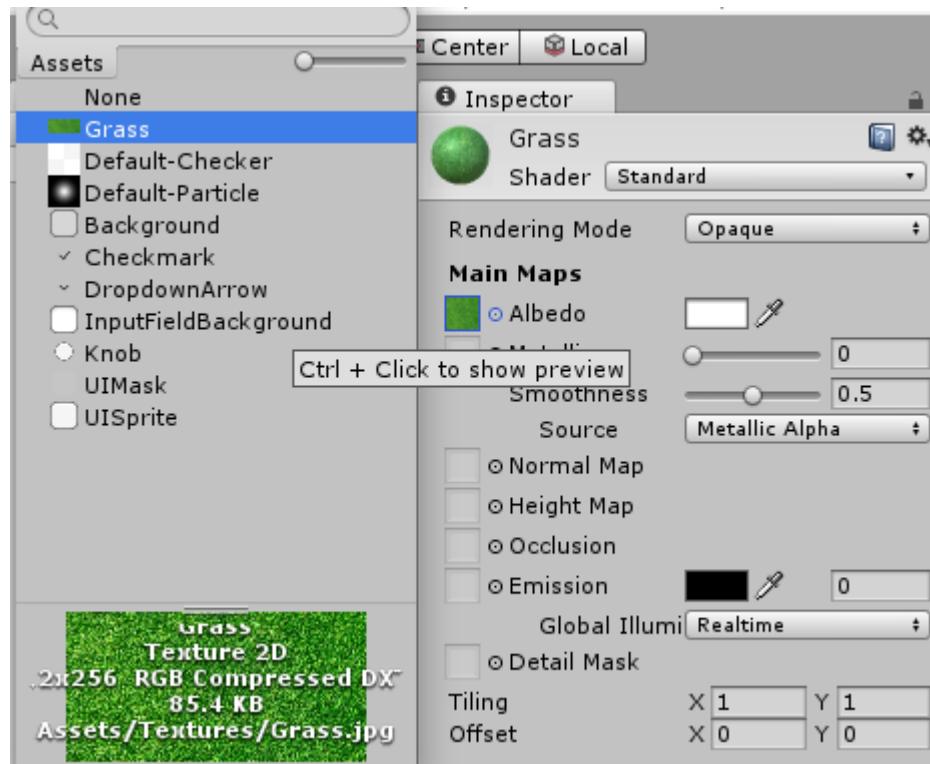
4. Materials và Textures

Việc tô màu cho các đối tượng là không thể thiếu nếu không muốn game trở nên nhảm chán bởi hình ảnh không có màu sắc, thiếu sức sống. Ta có thể tùy biến thêm bớt các màu sắc khác nhau với hệ thống bảng màu đa dạng, phong phú, làm nổi bật nên các đối tượng trong game.

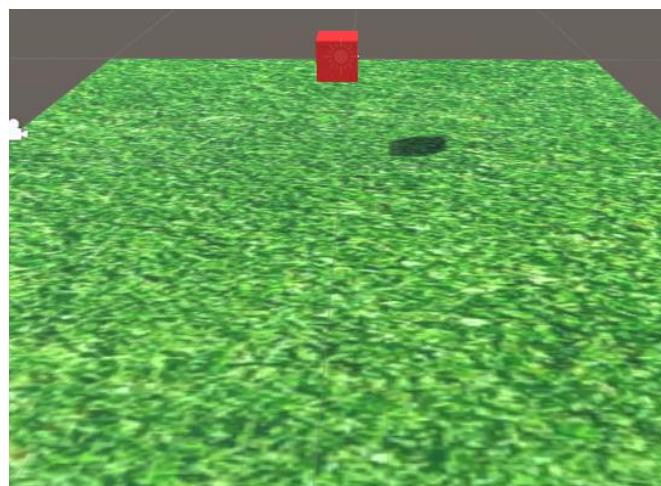


Hình 81 – Tùy chỉnh màu sắc cho GameObject

Tuy nhiên nếu chỉ đơn giản là các màu sắc như xanh, đỏ, tím, vàng... cũng chỉ như mặc một chiếc áo tron và trông khá đơn điệu. Chúng cần một chiếc áo với các hoa văn nổi bật nhằm xóa tan đi sự đơn điệu của chính nó, đó là chính là Texture.

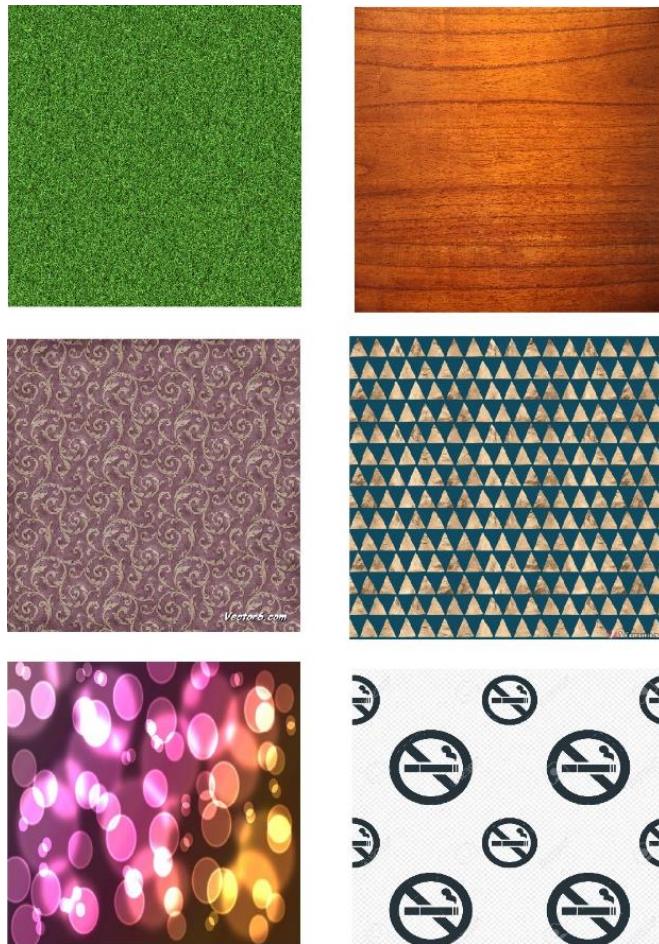


Hình 82 – Tùy chỉnh Texture cho GameObject



Hình 83 – GameObject đã được tùy chỉnh màu sắc và texture

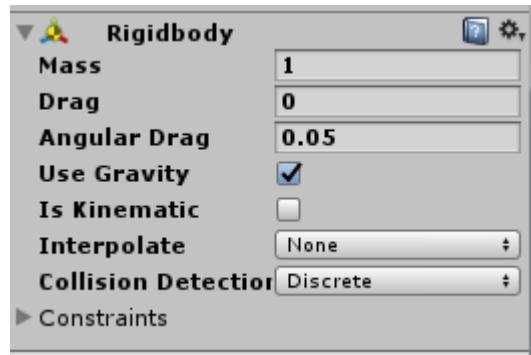
Texuter thật chất chính là file hình ảnh về một loại chất liệu vào đó (như cỏ, tường, gạch, gỗ...) hay một loại hoa văn cách kiều, hoặc cũng có thể là một loại hình ảnh đã được thiết kế trước đó.



Hình 84 – Một số loại Texture

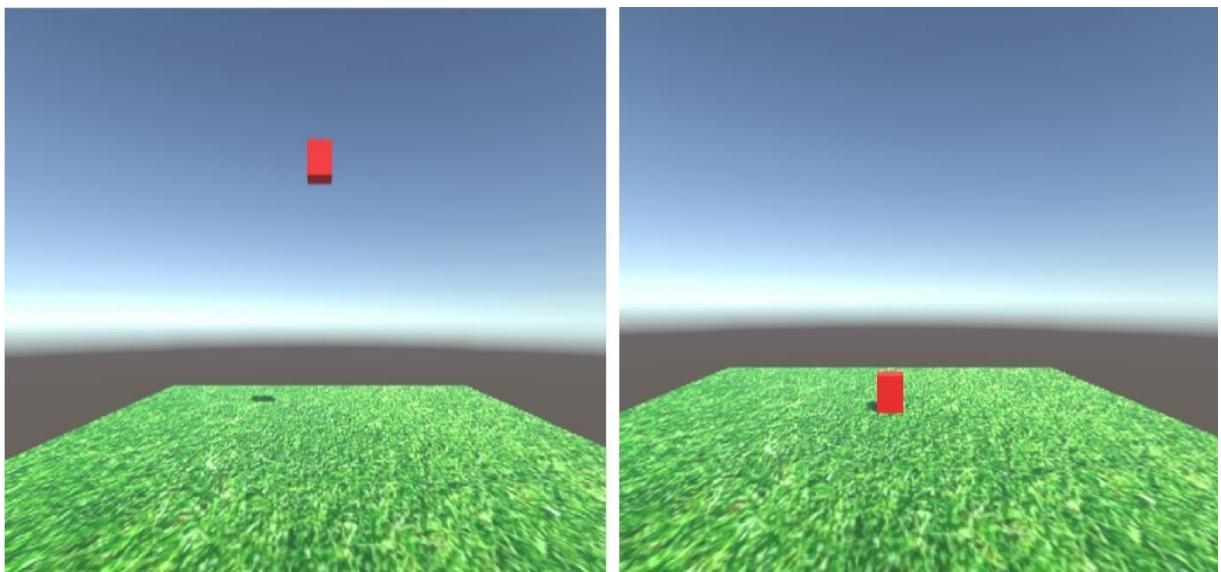
5. Component của GameObject

Ngoài các thuộc tính cơ bản của GameObject như Position, Rotation, Scale; ngoài ra ta còn có thể thêm các thuộc tính khác như Script, các thuộc tính vật lý....



Hình 85 – Các Component của GameObject

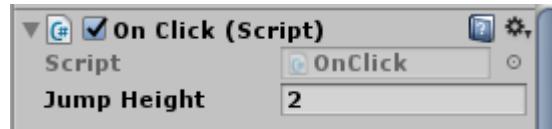
Đối với các thuộc tính vật lý đại diện điển hình như Rigidbody để xử lý trọng lực của vật, Collider để xử lý va chạm giữa các vật...



Hình 86 – GameObject chịu ảnh hưởng vật lý sau khi gắn Component

6. Viết Script cho GameObject

Script chính là kịch bản được biểu diễn dưới dạng các đoạn mã C# hoặc Javascript. Các đoạn mã có công việc sẽ xử lý các hành động của các GameObject mà nguyên thủy các GameObject đó chưa thể tự xử lý được. Các hành động có mang tính logic hay không tùy thuộc khả năng viết mã của nhà phát triển. Mỗi một Script sẽ được gắn cho một hoặc nhiều GameObject. Trình viết mã và biên dịch của Unity được tích hợp với công cụ Monodevelop hoặc Visual Studio, từ phiên bản 5.5 trở lên Unity có thể kế hợp tốt Visual Studio và gần như không cần dùng đến Mono nữa.



Hình 87 – Xây dựng Script cho GameObject

Tất cả mọi Script luôn được thừa kế từ lớp MonoBehaviour. Đây là một lớp cơ bản chứa các Interface là các phương thức hành vi của tất cả GameObject.

```

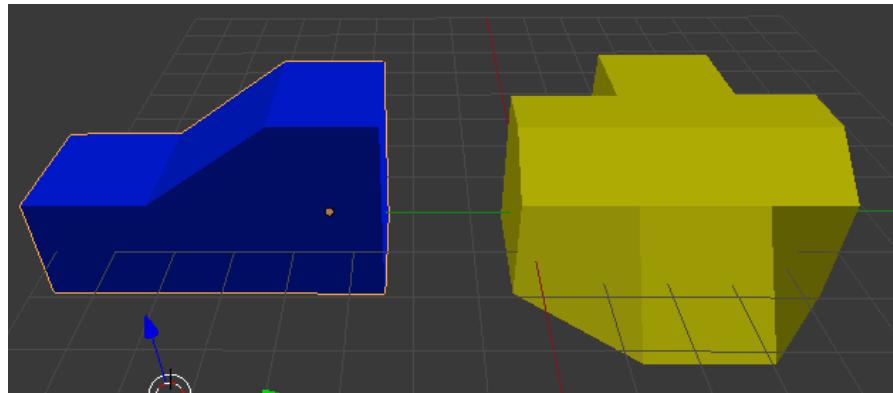
OnClick.cs ✘ ×
New Unity Project | OnClick | Update()
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class OnClick : MonoBehaviour
6  {
7      public float JumpHeight;
8
9      void Update()
10     {
11         if (Input.GetButtonDown("Fire1"))
12         {
13             GetComponent<Rigidbody>().AddForce(transform.up * JumpHeight, ForceMode.Impulse);
14         }
15     }
16 }
17

```

Hình 88 – Trình Editor và IDE Visual Studio để viết Script

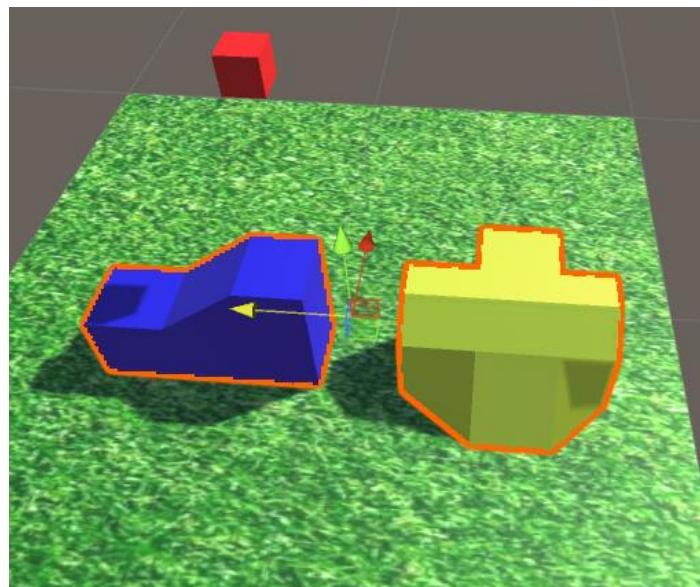
7. Tạo GameObject từ công cụ thứ 2

Bản thân Unity chỉ có thể tạo ra các GameObject cơ bản, và rất hạn chế trong mặt thiết kế. Vì Unity vốn không phải là công cụ xử lý đồ họa. Muốn xây dựng nên những hình ảnh cầu kỳ, phức tạp để hình ảnh game trở nên đa dạng và phong phú. Nhà phát triển có thể sử dụng một công cụ thứ 2 dùng để thiết kế nên các Object phức tạp.



Hình 89 – Hai Object được vẽ bằng công cụ Blender

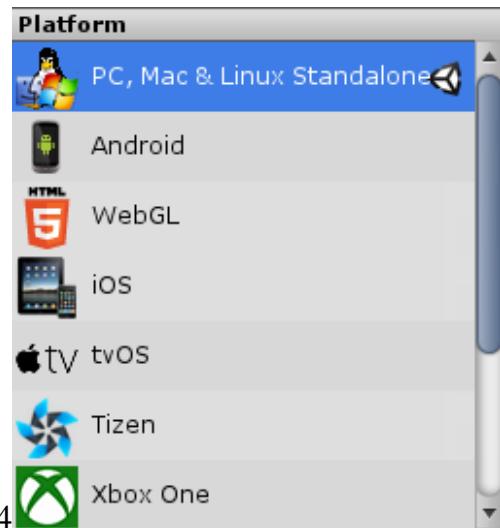
Unity sẽ hỗ trợ cho phép import các file xuất từ công cụ vẽ 3D nổi tiếng như Maya, Blender, 3DS Max... Sau khi import có thể sử dụng, tương tác như một GameObject nguyên thủy của Unity



Hình 90 – Sử dụng GameObject từ Blender vào Unity

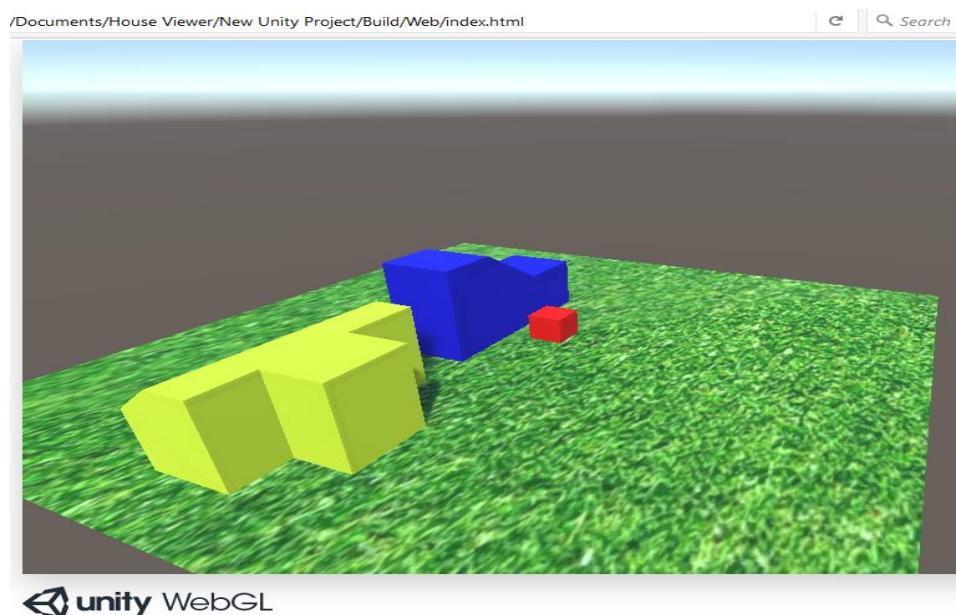
8. Cấu hình và Build đa nền tảng

Unity cho phép xuất project thành các file thực thi trên nhiều nền tảng khác nhau. Ta có thể chọn lựa, thay đổi, bổ sung các nền tảng khác nhau mà Unity cung cấp.

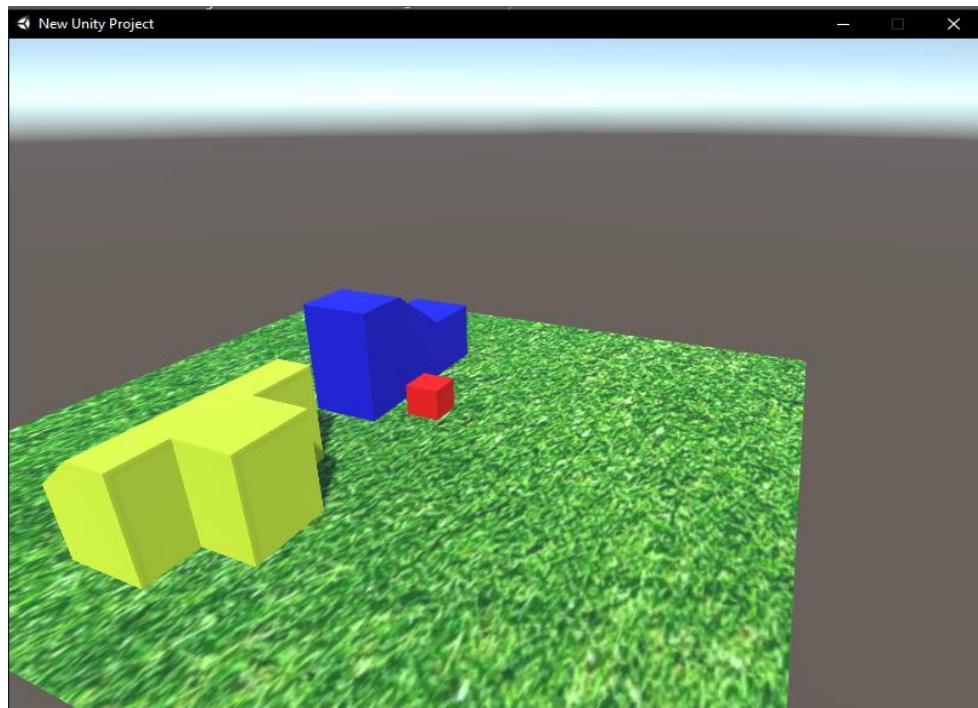


Hình 91 – Các nền tảng có thể build trên Unity

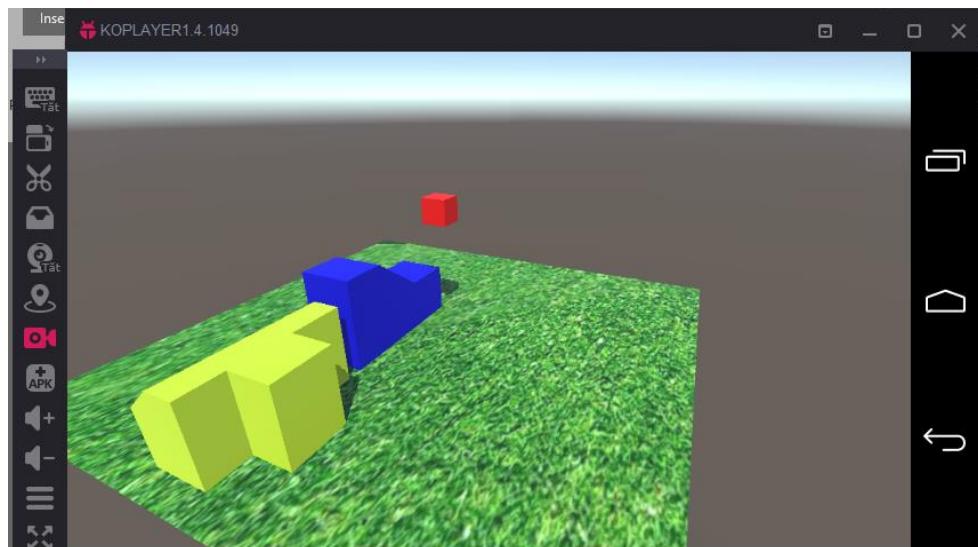
Kết quả sau khi Build ta có được các file thực thi cho từng nền tảng



Hình 92 - Ứng dụng chạy trên nền tảng Web OpenGL



Hình 93 - Ứng dụng chạy trên nền tảng PC Window



Hình 94 - Ứng dụng chạy trên nền tảng Android

3.3 VR trên Unity

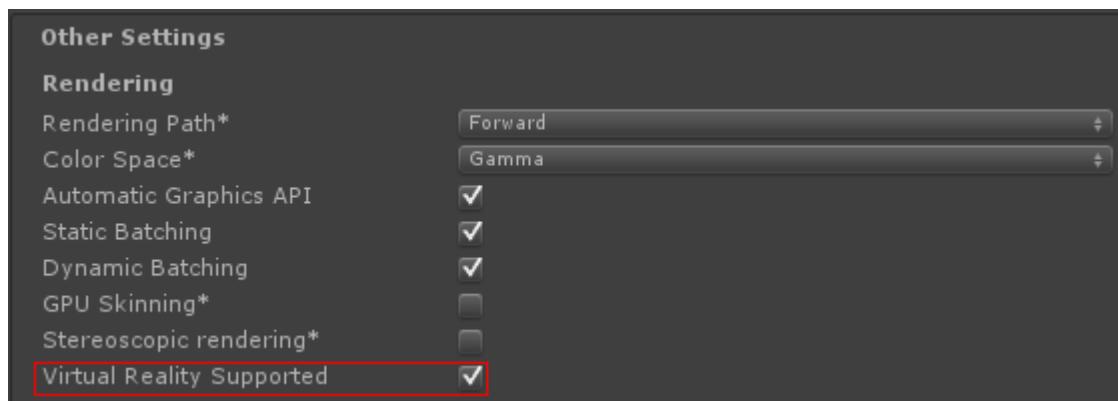
3.3.1 Oculus

Unity Technology đã lên kế hoạch từ rất lâu để làm cho Unity Engine của họ hỗ trợ công nghệ thực tế ảo. Công ty đã tung ra phiên bản Unity 5.1 với một số tính năng quan trọng dành cho các nhà phát triển VR.

Unity 5.1 cung cấp một loạt các tính năng mới cho phép các nhà phát triển tạo ra các trò chơi đa nền tảng tốt hơn, họ có thể chuyển từ iOS sang PC (hoặc ngược lại) dễ dàng hơn rất nhiều. Nó cũng bao gồm Unity Multiplayer Service, cho phép các nhà phát triển thực hiện một số thứ như matchmaking, hiện tại nó vẫn còn trong bản preview, nhưng bạn vẫn có thể sử dụng dịch vụ này miễn phí với tối đa là 100 người chơi cùng lúc. Unity 5.1 cũng được mở rộng hỗ trợ cho VR, đặc biệt là với Oculus Rift. Unity Technology cũng hứa sẽ sớm hỗ trợ cho Microsoft Hololens và Gear VR.

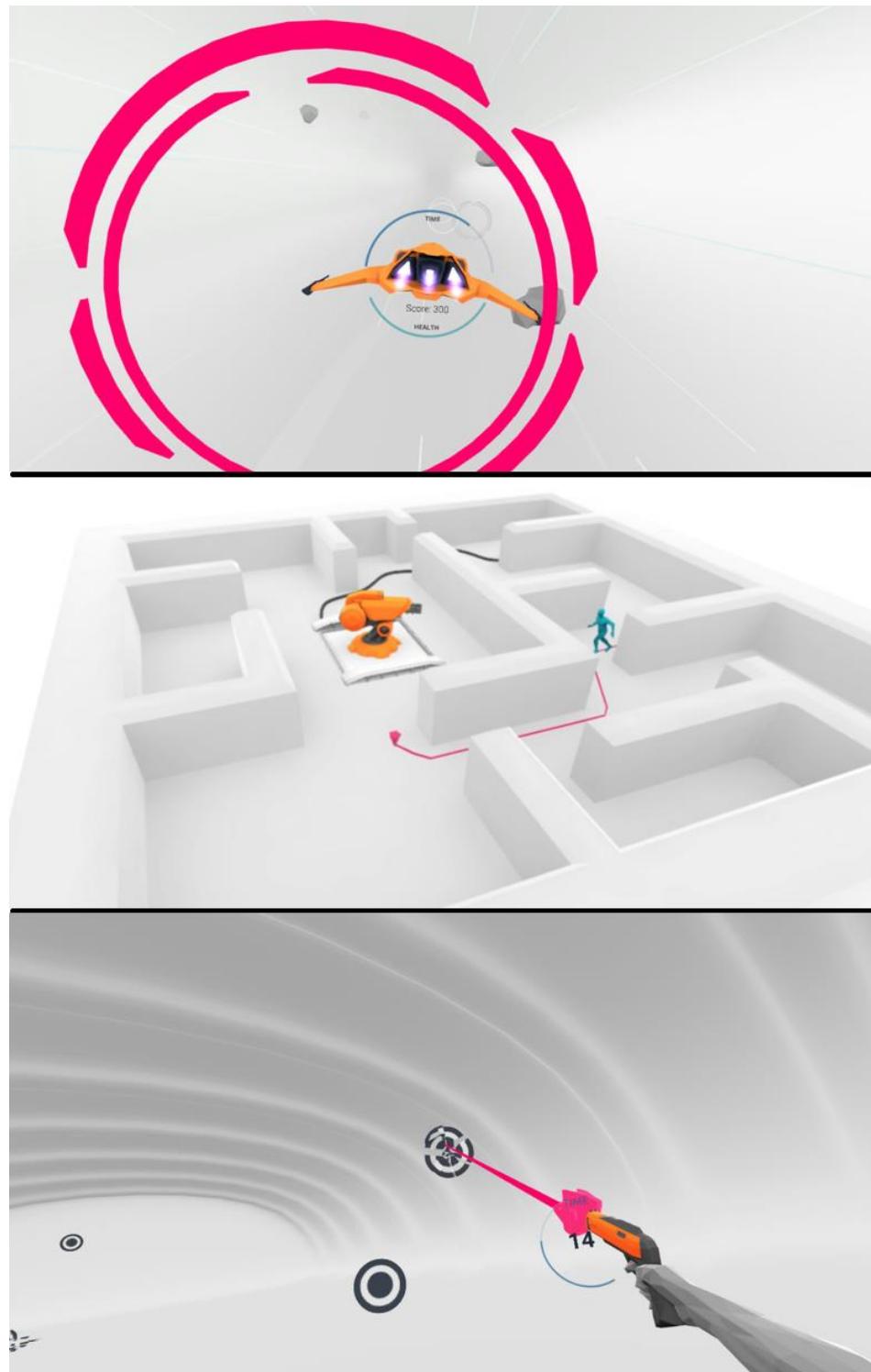
Unity đã giới thiệu tích hợp được hỗ trợ cho các thiết bị VR nhất định. Tập trung vào các thiết bị VR của Oculus, đặc biệt là Oculus Rift Development Kit 2 (DK2) và phiên bản của Gear VR (Samsung Galaxy S6, S6 Edge, S6 Edge+, hoặc Note 5). Các thiết bị VR Head Mounted Displays (HMDs) cũng sẽ hoạt động với Unity, chẳng hạn như HTC Vive. Và trong tương lai sẽ được cập nhật các nền tảng VR khác.

Để xây dựng ứng dụng VR trên nền tảng Oculus bằng Unity, chỉ cần kích hoạt tính năng VR được tích hợp trên Unity.



Hình 95 – Tính năng VR được tích hợp trên Unity

Một số ứng dụng VR Oculus được xây dựng trên Unity



Hình 96 – Game VR xây dựng bằng Unity

3.3.2 Thư viện Google VR SDK

Google VR SDK là một thư viện bên thứ 3 của Google phát hành cho phép hỗ trợ xây dựng nên các ứng dụng thực tế ảo. Google cung cấp cho các nhà phát triển hai nền tảng thực tế ảo:

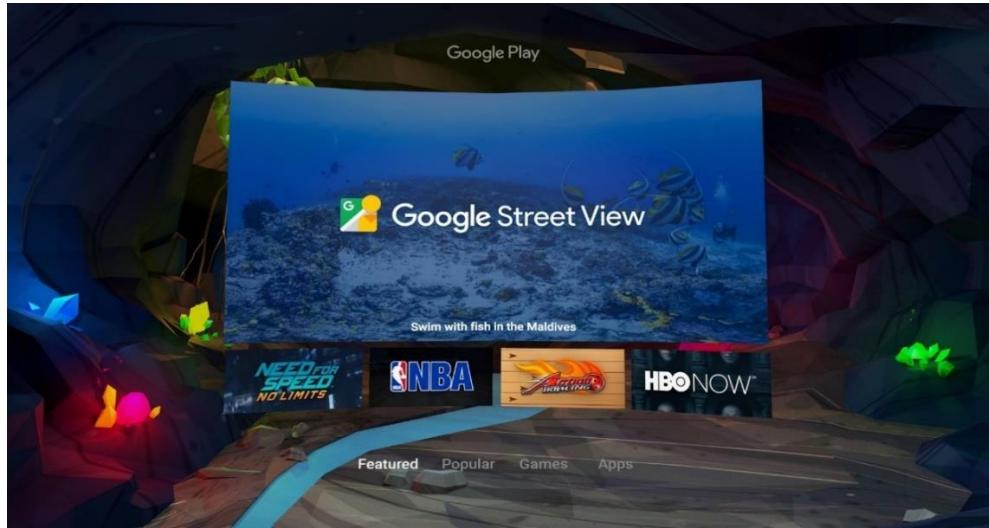
- **Google Cardboard:** là nền tảng VR cho di động phổ biến nhất và dễ tiếp cận nhất. Cardboard cho phép trải nghiệm VR một cách đơn giản, hợp lý. Cardboard được xây dựng cho những trải nghiệm VR nhỏ và làm việc với hầu hết bất kỳ điện thoại thông minh trên Android hay iOS.
- **Daydream:** một nền tảng mới cho độ trễ thấp, nhập vai và tương tác di động. Daydream là một nền tảng chất lượng cao, thực tế ảo mang tính di động. Daydream cung cấp những trải nghiệm phong phú, trải nghiệm nhập vai với phần cứng và phần mềm được xây dựng cho VR.

Google VR SDK bao gồm mọi thứ cần thiết để phát triển cho các nền tảng, bao gồm thư viện, tài liệu API, mẫu nhà phát triển, và hướng dẫn thiết kế.

3.3.2.1 DayDream

Ở sự kiện Google I/O 2016, Google đã giới thiệu một nền tảng mới được tối ưu hóa cho thực tế ảo với tên gọi Daydream. Nền tảng này được xây dựng trên nền Android N nhằm mang lại trải nghiệm VR tốt hơn so với hiện nay.

Daydream sẽ bao gồm một chế độ mới tên là Android VR Mode. Khi kích hoạt nó lên, giao diện, ứng dụng sẽ được chuyển sang tối ưu cho không gian thực tế ảo. Màn hình chính của Android VR Mode sẽ cho phép truy cập chỉ những ứng dụng hay nội dung nào được thiết kế cho thực tế ảo mà thôi. Trong chế độ này còn có phiên bản đặc biệt của YouTube, Street View, Play Store, Play Movies và Google Photos, tất nhiên là các ứng dụng này đều sẽ được phát triển riêng cho VR.



Hình 97 – Chế độ DayDream trên Android N

Thiết kế này bao gồm một chiếc kính VR và tay cầm, trong đó chiếc kính sẽ có một khay gắn điện thoại. Về tay cầm điều khiển, bên cạnh các nút nhấn truyền thống, "motion controller" này còn có thể dùng cử chỉ để điều khiển trong không gian thực tế ảo. Thao tác bằng cử chỉ này sẽ trực quan hơn và mang lại cảm giác thật hơn so với việc chỉ nhấn các nút tĩnh trên tay cầm.

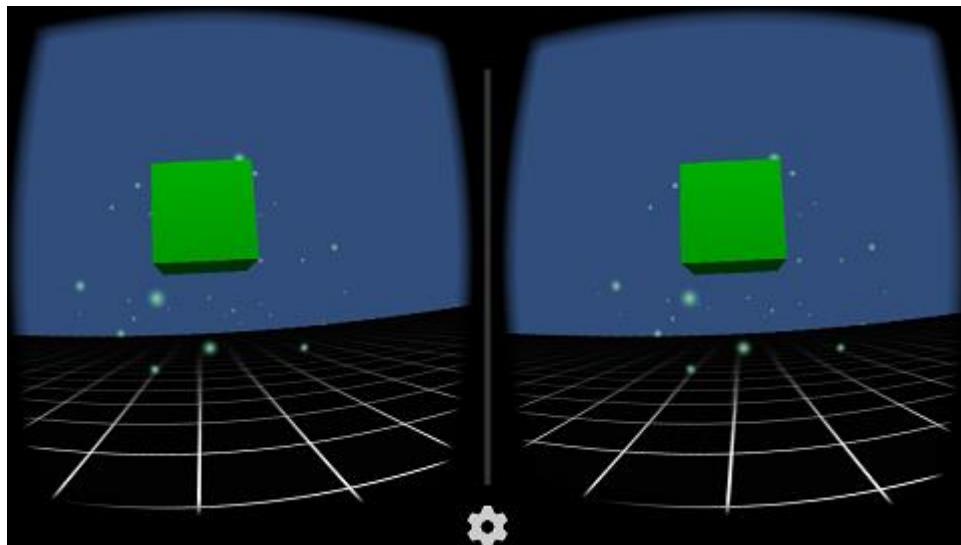


Hình 98 – Chế độ DayDream kết hợp với tay cầm điều khiển

Về phần cứng yêu cầu một thiết bị VR có tai nghe và tay điều khiển. Kết hợp với một chiếc điện thoại có thể chạy DayDream, nhà phát triển có thể tạo ra một trải nghiệm VR chất lượng và có khả năng tương tác cao. Tuy nhiên, DayDream chỉ có thể được xây dựng trên các thiết bị điện thoại chạy hệ hành Android N (API 24).

3.3.2.2 Cardboard

Carboad 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.



Hình 99 – Demo Cardboard trên Unity

Đầu năm 2014, Google đã đề xuất ý tưởng dùng giấy bìa các tông và một số vật liệu giá rẻ để người dùng có thể tự chế chiếc kính thực tế ảo Google Cardboard. Đến hôm nay, Google đã khẳng định mục đích nghiêm túc của mình qua việc nâng cấp một số ứng dụng trên smartphone cho Cardboard, đồng thời phát hành bộ công cụ SDK cho lập trình viên.

Google mới cho biết ứng dụng Google Cardboard đã cho phép tải về cho cả iOS và Android. Với ứng dụng này, người dùng có thể sẽ xem được những ứng dụng và Game hỗ trợ công nghệ thực tế ảo thông qua Google Cardboard.



Hình 100 – Thiết bị và thư viện Cardboard của Google

Ngoài ra, họ còn cập nhật bộ công cụ phát triển phần mềm để loại bỏ tình trạng hình ảnh vẫn tiếp tục được hiển thị khi bạn đã ngừng quay đầu (drift). Bằng cách sử dụng các thuật toán mới, kiểu soát, hợp nhất tốt hơn dữ liệu của con quay hồi chuyển và gia tốc kế thì Google đã giải được bài toán khó này.

Google Cardboard sẽ hỗ trợ cả ứng dụng Street View. Giờ đây bạn chỉ cần ngồi một chỗ là có thể du lịch vòng quanh thế giới một cách miễn phí. Bạn cũng có thể chụp ảnh 360 độ để chia sẻ lên Street View như một cách quảng bá du lịch khá hiệu quả.

Với công nghệ thực tế ảo đang mạnh nha phát triển và chi phí để sở hữu một chiếc kính thực tế ảo quá cao so với nhu nhập của người dùng, Google đã giúp công nghệ này phổ cập đến cả với những người dùng bình dân thông qua bộ kính thực tế ảo Google Cardboard giá rẻ và bộ công cụ Cardboard SDK nhằm phát triển hơn nữa công nghệ này trong tương lai.

3.3.2.3 Sử dụng thư viện Google SDK trên Unity

Cả hai nền tảng VR Cardboard và DayDream đều có hướng xây dựng tương tự nhau, điểm khác nhau duy nhất là là nền tảng thiết bị sẽ chơi ứng dụng đó. Với Cardboard có thể chơi được hầu hết các dòng điện thoại thông minh Android và iOS có hỗ trợ con quay hồi chuyển. Còn DayDream chỉ có thể xây dựng trên Android N.

Trước khi tiến hành xây dựng ứng dụng, ta cần chuẩn bị các thứ sau:

- Thư viện SDK chính là Asset sẽ được import vào Unity có thể tải trên trang chủ Google VR
- Unity phiên bản 5.1 trở lên
- Bộ cài đặt Android SDK
- Thiết bị điện thoại và kính VR

Để tải bộ thư viện SDK ta vào trang chủ của Google VR

Google VR SDK for Unity

Download the Google VR SDK for Unity to create mobile VR applications for Android or iOS. This Unity package contains the SDK and some demo scenes for you to try.

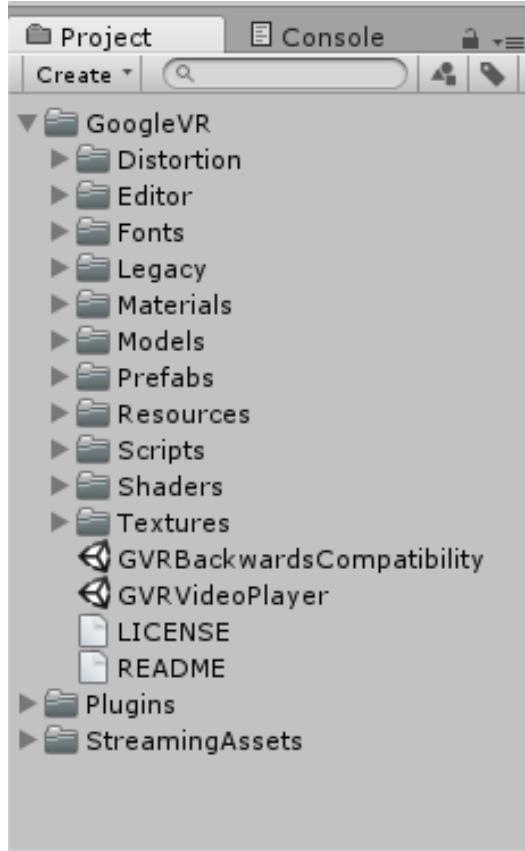
[DOWNLOAD SDK](#)

See the [release notes](#) for more information.

Once you've downloaded the SDK continue by reading the Getting Started guides for [Android](#) and [iOS](#).

Hình 101 – Trang tải thư viện SDK

Sau khi Import thư viện vào, ta có thể bắt đầu xây dựng ứng dụng



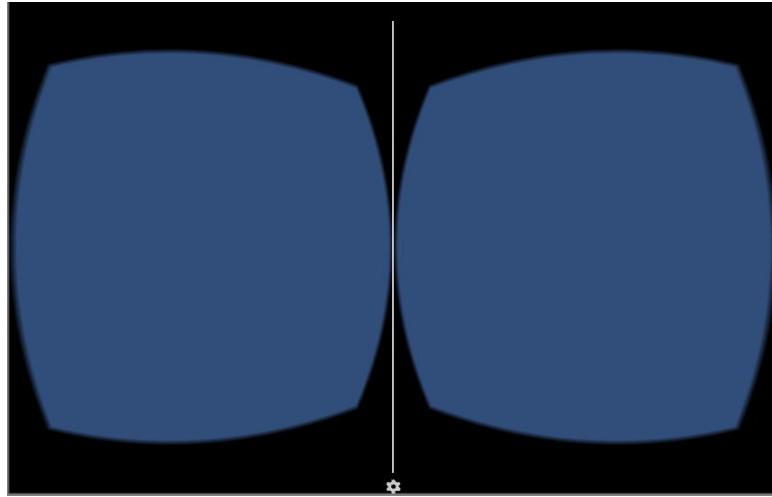
Hình 102 – Cấu trúc thư viện Google SDK

Cấu trúc thư viện khá phức tạp, tuy nhiên ta sẽ chỉ quan tâm một số thao tác cơ bản. Ở phần trên đã mô tả một ứng dụng VR sẽ được chia thành 2 màn hình. Như vậy có nghĩa ta sẽ phải tạo ra 2 Camera và sẽ hoạt động đồng bộ với nhau.

Có hai cách để thiết kế một Camera VR là: **GvrMain** và **GvrViewerMain**

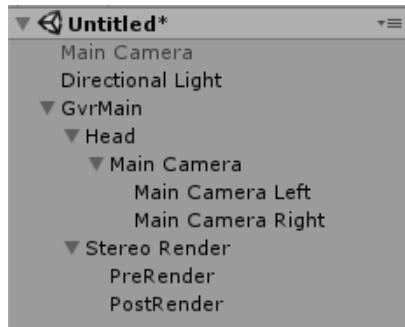
Cách 1: GvrMain

GvrMain là một Prefabs đã được xây dựng sẵn với nhiệm vụ thiết kế hai Camera dành cho mắt trái và mắt phải. Ngoài ra còn giúp render thu nhận hình ảnh và xử lý hiển thị trên Camera, đồng thời còn thiết lập sẵn một số đặc tính quan trọng của VR như bo tròn Camera, render hai luồng hình ảnh lệch nhau.



Hình 103 – Hai Camera đã được xây dựng sẵn

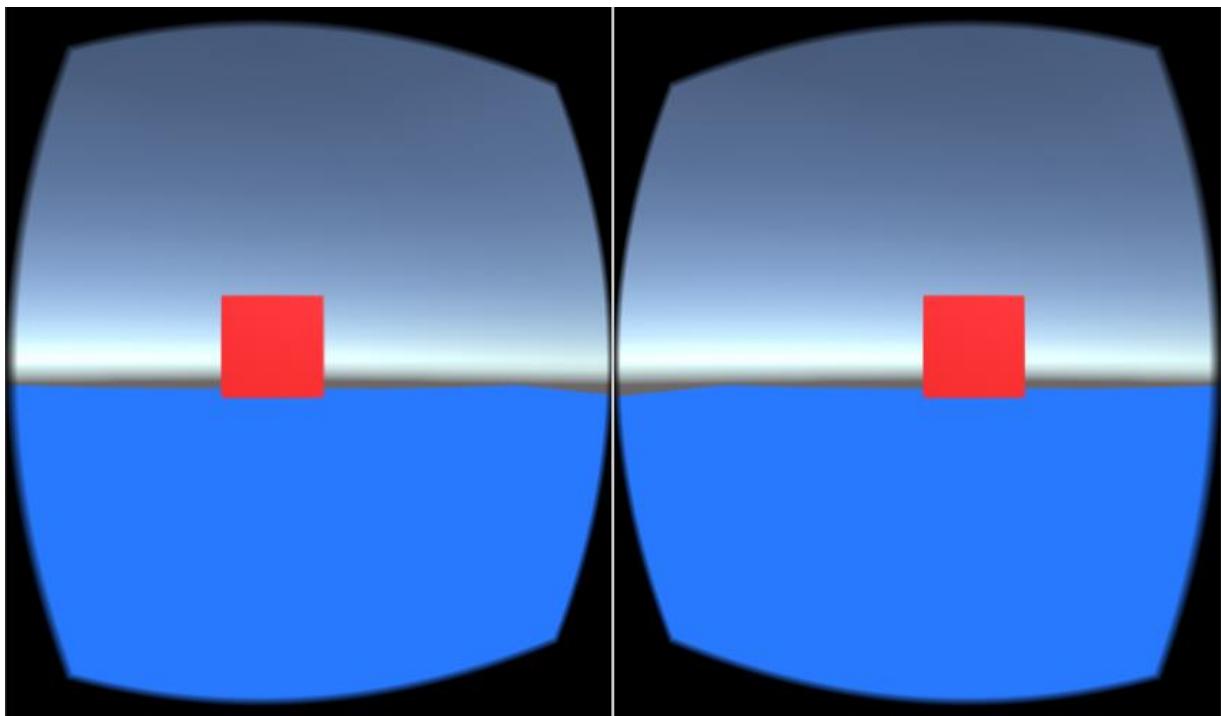
Như vậy, GvrMain đã giúp ta làm công việc chia đôi màn hình thành 2 và mỗi màn hình được tạo sẵn độ cong giúp cho thấu kính trên kính VR có thể dễ dàng thu nhận hình ảnh từ màn hình thiết bị. Ta có thể quan sát thấy, GvrMain đã bao gồm trong đó 2 Camera trái phải và 2 Object Render hình ảnh. Trong mỗi Object có chạy các Script hỗ trợ cho việc thu hình ảnh thời gian thực và xử lý Camera tiếp nhận hình ảnh đồng bộ với nhau.



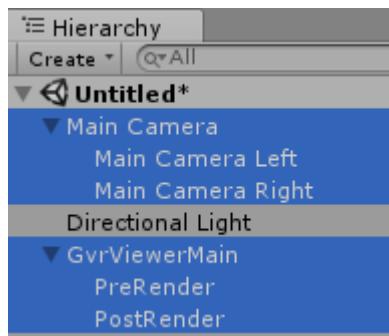
Hình 104 – Cấu trúc Prefab của GvrMain

Cách 2: GvrViewerMain

Nếu nhu cầu cấu trúc GvrMain rườm rà và phức tạp thì vẫn còn một cách thứ hai để thiết kế Camera VR chỉ với một Prefabs đơn giản là GvrViewerMain và một Camera mặc định của Unity. Tuy chỉ có một Camera, nhưng GvrViewerMain sẽ giúp ta tự sinh ra thêm một Camera nữa và cũng tự động làm công việc thu nhận hình ảnh thời gian thực, chia đôi màn hình và tạo camera cong lại.



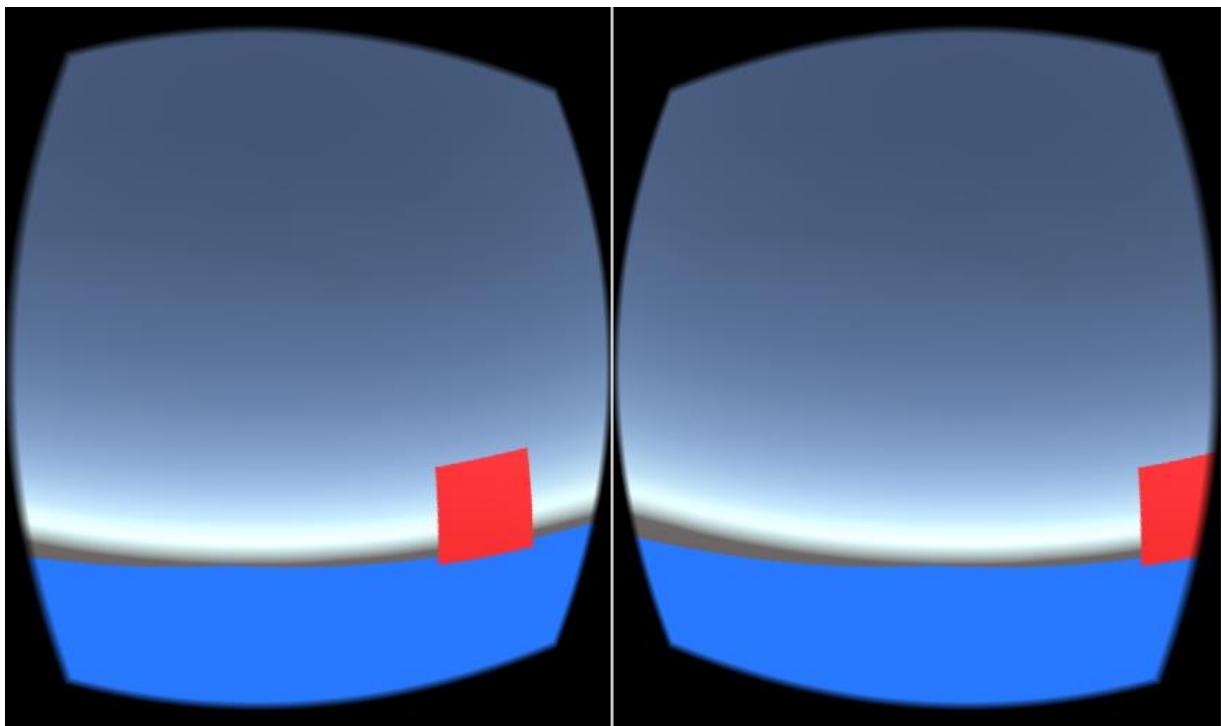
Hình 105 – Camera được thiết kế bằng GvrViewerMain



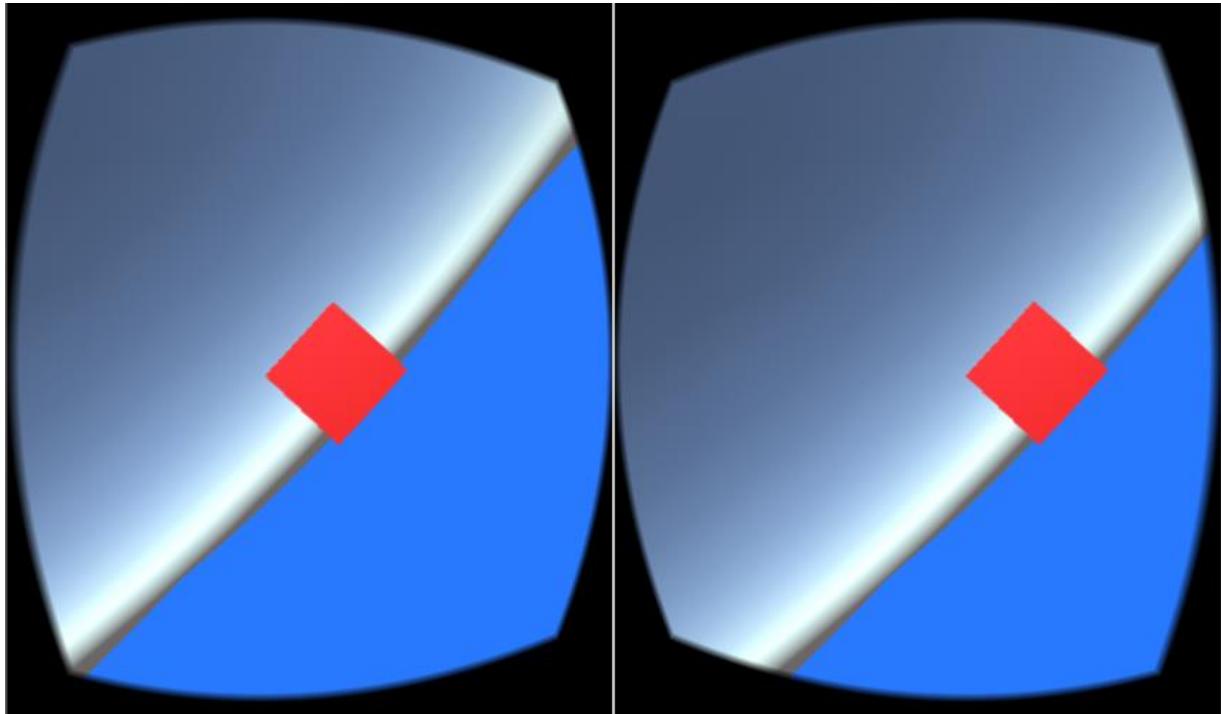
Hình 106 – Cấu trúc Prefab được sinh tự động khi Play

Để mô phỏng một số thao tác khi sử dụng thiết VR ta có thể sử dụng phím tắt như:

- Alt: Để mô phỏng thao tác xoay người sang trái, sang phải, nhìn lên, nhìn xuống – thao tác chỉ có thể thực hiện trên thiết bị có hỗ trợ con quay hồi chuyển
- Ctrl: Để mô phỏng thao tác nghiên đầu sang trái hoặc phải tương tự như phím Alt



Hình 107 – Mô phỏng thao tác xoay người



Hình 108 – Mô phỏng thao tác nghiên đầu

Như vậy bước quan trọng nhất trong quá trình xây dựng ứng dụng VR đã xong. Công việc còn lại chỉ là xây dựng mô hình, chức năng và kịch bản Script cho ứng dụng như mọi ứng dụng Unity khác.

3.4 Áp dụng mô hình trường TĐT vào Unity

Trước khi tiến hành áp dụng mô hình vào Unity ta sẽ điểm qua về tài liệu thiết kế chức năng ứng dụng

3.4.1 Thiết kế chức năng

1. Mô tả chức năng

Mô hình khu C trường Tôn Đức Thắng bao gồm 6 tầng lầu và tầng trệt. Ở mỗi tầng sẽ có các sảnh, phòng (phòng học, phòng giảng viên...), một số thiết bị, cầu thang, cửa ra vào... Ứng dụng sẽ thiết kế cho phép người dùng tham quan từng tầng của khu C, người sử dụng có thể tự do đi lại trong phạm vi của tầng đó. Đối với các phòng có cửa ra vào, khi người dùng tiến lại gần cửa một khoảng cách xác định sẽ tự động mở cửa

và người dùng có thể đi vào bên trong phòng tham gian. Mỗi tầng sẽ được chia thành nhiều cảnh. Khi người dùng muốn lên hoặc xuống lầu tiếp theo thì tiến lại cầu thang sẽ xuất hiện một menu cho phép chọn lựa chọn tầng muốn lên hoặc xuống. Do điện thoại được lắp vào kính VR nên không theo thực hiện thao tác chọn được nên trong ứng dụng giữa Camera sẽ xuất hiện một hồng tâm (hoặc chữ thập – Reticle). Hồng tâm này luôn nằm ở giữa Camera và sẽ di chuyển theo khi Camera di chuyển. Khi người dùng đưa hồng tâm vào menu tầng lầu, người dùng sẽ có một khoảng thời gian chờ trước khi ứng chuyển cảnh. Nếu người dùng muốn từ chối việc lên tầng đã chọn mà muốn chọn tầng khác hay thực hiện thao tác nào khác chỉ việc đưa hồng tâm ra chỗ khác trước khi thời gian chờ đã kết thúc. Sau khi thực hiện chuyển cảnh lên tầng khác, vị trí của người dùng sẽ đứng ngay tại cầu thang mà người đó đã chọn trước đó.

2 . Các mô hình chức năng

3.4.2 Xây dựng ứng dụng

Nội dung tiếp theo sẽ tiến hành xây dựng ứng dụng hoàn chỉnh.

Trước khi bắt đầu ta cần Import các mô hình đã thiết kế bằng Blender trước đó. Sau khi Import mô hình, ta tiến hành tạo các Scene, Camera VR, sắp xếp và bố cục các tầng theo từng Scene tương ứng. Tiếp theo áp dụng các phương pháp tương tác trên VR để xây dựng các chức năng

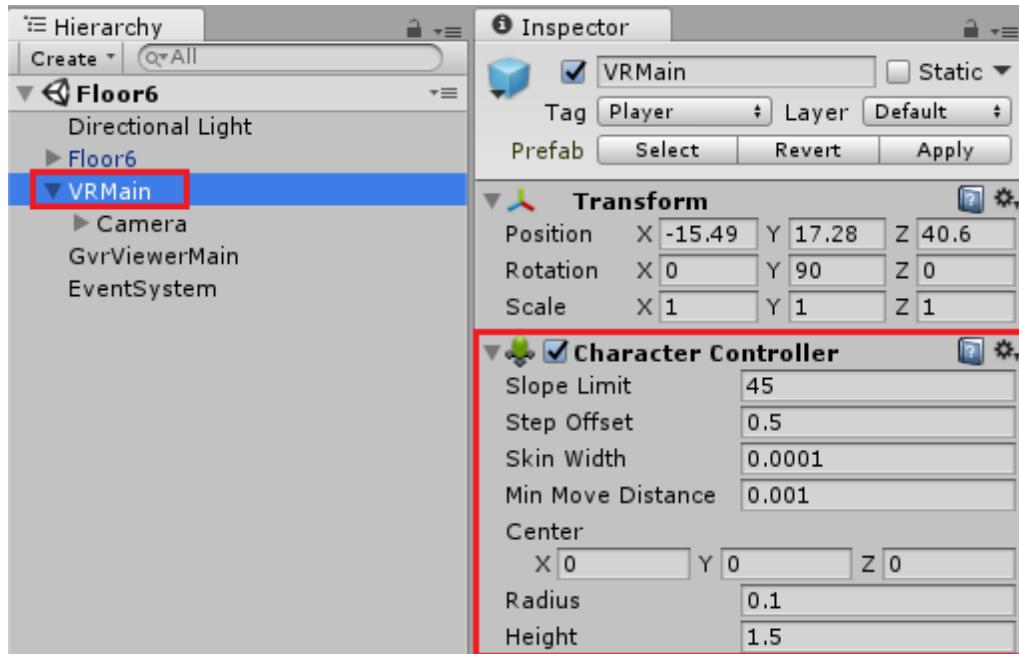


Hình 109 – Mô hình trường TDT trên Unity

3.4.2.1 AutoWalk - Chức năng di chuyển

Chức năng này có thể tương tác bằng cách sử dụng thiết bị tay cầm truyền thống, với thiết kế đơn giản chỉ cần Click và giữ nút nhán vật sẽ di chuyển tiến lên và khi không click nữa nhán vật sẽ dừng lại.

Để Camera VR có thể di chuyển như nhân vật game thể loại góc nhìn thứ nhất (First Person Shooter – FPS), ta cần gắn thuộc tính Character Controller cho nó.

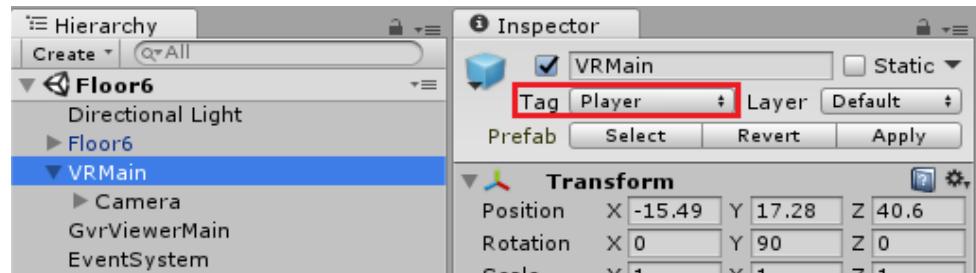


Hình 110 – Thuộc tính Character Controller

Thuộc tính Character Controller được thư viện Unity cung cấp phương thức SimpleMove giúp cho Object có thể di chuyển được. Ta chỉ cần lấy ra thông tin hướng tọa độ của Camera, và truyền vào phương thức SimpleMove tham số tốc độ, Object sẽ tiến về phía trước một đoạn tương ứng.

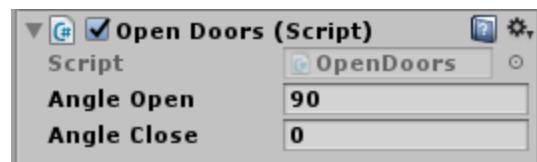
3.4.2.2 OpenDoor – Chức năng mở cửa

Chức năng mở cửa sử dụng hệ thống tự động. Cửa sẽ tự động được mở khi nhân vật đến gần và tự động đóng khi nhân vật ra xa. Hệ thống xử lý bằng cách kiểm tra khoảng cách từ vị trí cửa tới vị trí nhân vật (Player).



Hình 111 – Đánh dấu Tag cho nhân vật là Player

Với mỗi cánh cửa sẽ có một chiều xoay khác nhau theo tọa độ Z. Cứ mỗi frame hình ảnh, mỗi cánh cửa sẽ tự xử lý tính toán vị trí của chính nó theo hướng của Player, dù cho Player có di chuyển đi đâu. Nếu Player di chuyển tới gần cửa một khoảng xác định. Object cửa đó sẽ thực hiện phương thức Quaternion để xoay theo góc chiều Z được định trước đó. Và khi Player đi ra xa, Object sẽ thực hiện lại phương thức Quaternion theo góc chiều Z lúc ban đầu.



Hình 112 – Mỗi cánh cửa sẽ có thông số góc đóng/mở cửa khác nhau

Một lưu ý quan trọng, trong Unity tồn tại 2 phương thức xoay khác nhau là Rotation và Quaternion, trong đó:

- Rotation: xoay Object theo trọng tâm của vật. Trọng tâm là điểm cân bằng của một vật dựa theo kiến thức toán học, ví dụ trọng tâm của hình vuông là điểm giao nhau của hai đường chéo, trọng tâm của hình tròn là tâm của chính nó)
- Quaternion: xoay Object theo điểm trục của vật. Điểm trục là một điểm nằm bất kỳ trên Object và Object sẽ xoay quanh theo bán kính quỹ đạo của điểm trục đó. Có thể hiểu cách khác điểm trục chính là bản lề của một số vật như cửa, bản lề được cố định và chỉ có thể xoay cửa theo hướng của điểm trục.

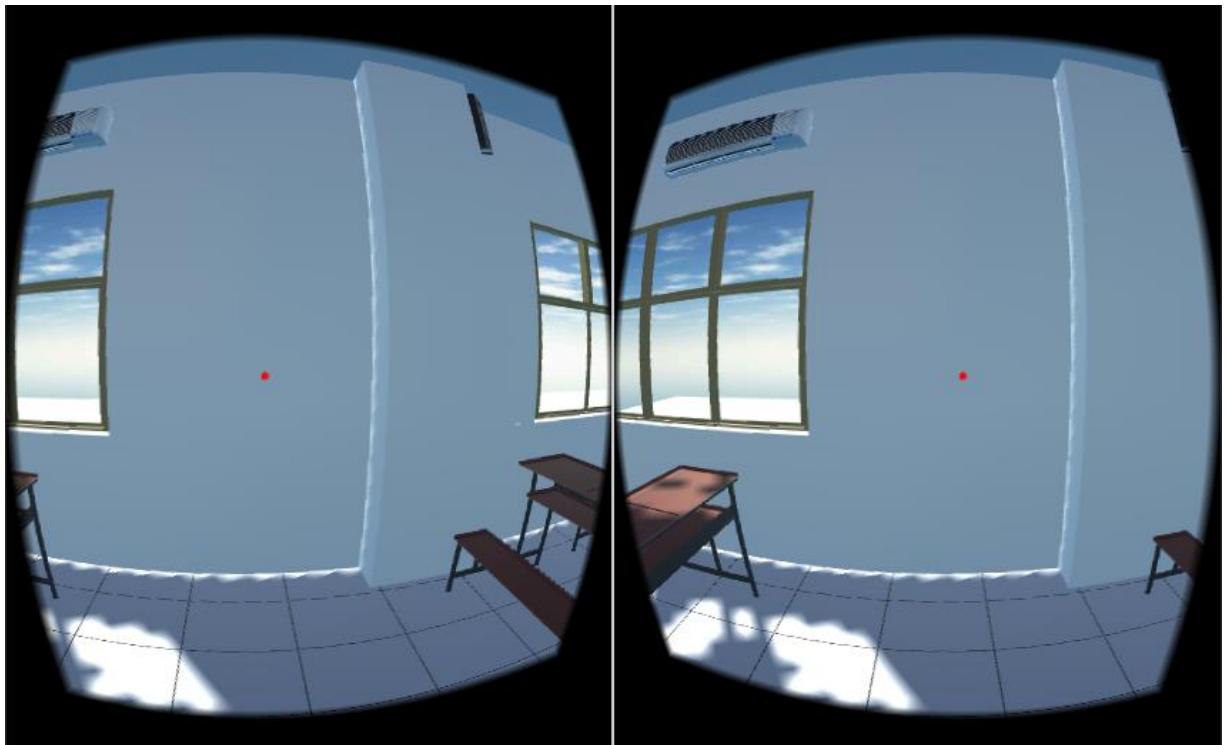
3.4.2.3 Reticle

Để thực hiện các chức năng liên quan đến Button và thao tác Click Button ta cần áp dụng phương pháp dùng hồng tâm (Reticle). Không thể sử dụng tay cầm cho thao tác Click Button nữa vì đã sử dụng cho AutoWalk bên trên. Vì thế ta sử dụng Reticle để tạo sự phong phú chức năng cho ứng dụng.



Hình 113 – Thư viện SDK cung cấp hỗ trợ Reticle

Sau khi đã thêm GvrReticle màn hình sẽ tự động xuất hiện một điểm giữa màn hình. Ta có thể di chuyển điểm này cách di chuyển Camera VR.



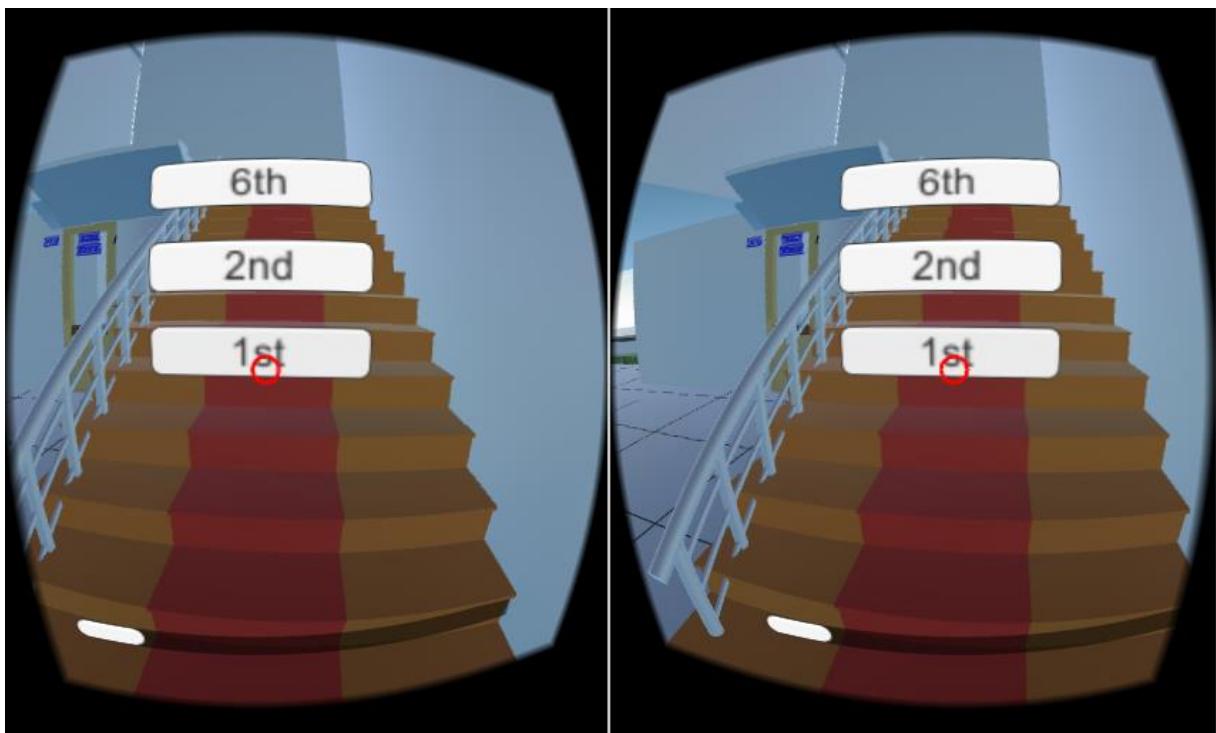
Hình 114 – Reticle trong ứng dụng

3.4.2.4 ShowMenu, Slider và ChangeScene

Tiếp theo thiết kế Menu hiển thị các danh sách Button cho phép khi chọn sẽ lên lầu trong ứng và thanh Slider sẽ giúp hiển thị thời gian chờ. Để thiết kế Menu, Button và Slider ta sử dụng UI Controls của Unity

Với chức năng ShowMenu thực hiện tương tự như cửa, ta sẽ gắn Script ShowMenu tại mỗi cầu thang khi xác định được Player tiến lại gần và hiển thị lên.

Slider sẽ được hiển thị và bắt đầu chạy theo thời gian quy định khi đưa hồng tâm vào Button



Hình 115 – ShowMenu, ClickButton, Slider và ChangeScene

3.4.3 Một số khó khăn gặp phải

Dù ứng dụng đã hoàn thành tuy nhiên trong quá trình cũng gặp một số khó khăn. Một vài khó khăn đã khắc phục được, nhưng cũng có một vài khó khăn cần phải suy nghĩ thêm hướng giải quyết.

Khó khăn	Kinh nghiệm
Thư viện của Google bị lỗi	Thật tế thư viện Google không hề bị lỗi, có thể lúc Import bị thiếu file hoặc trong quá trình thao tác đã lỡ xóa mất file hay thay đổi cấu trúc, nội dung file.
Player không di chuyển được trên mặt đất	Các Object tiếp xúc trực tiếp với người như mặt đất, tường, cửa, bàn, ghế... phải luôn đi kèm một thuộc tính quan trọng là Mesh Collider để đảm bảo giữa Player và Object va chạm được với nhau
Cửa xoay không đúng ý hoặc xoay lung tung	Dù đã sử dụng phương thức Quaternion nhưng cửa vẫn xoay không đúng có thể do 2 nguyên do: Số liệu về góc mở cửa sai có thể thử nhập các con số khác hoặc điểm trực không nằm đúng bản lề có thể vào Blender chỉnh lại điểm trực rồi update lại vào Unity
Project quá lag, di chuyển và sử dụng chức năng không được mượt mà	Nguyên nhân bắt nguồn từ việc thiết kế. Do chưa có kinh nghiệm thiết kế dẫn đến bản vẽ mô hình khá nặng và không có cách nào giảm dung lượng. Dẫn tới việc project Unity cũng nặng theo, ảnh hưởng đến hiệu suất chạy của thiết bị. Trong tương lai có thể sẽ thiết kế một mô hình hợp lý về số liệu, xử lý các kỹ thuật vẽ điêu luyện hơn, và áp dụng việc tái sử dụng các GameObject trong Unity.

Bảng 6 – Những khó khăn trong Unity

CHƯƠNG 4 – KẾT QUẢ ĐẠT ĐƯỢC

4.1 Hình ảnh ứng dụng

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Nguyễn Huy Sơn (2006), Virtual Reality Technologie - Công nghệ Thực Tế Ảo.
2. Wikipedia, Thực tế ảo - https://vi.wikipedia.org/wiki/Thực_tế_ảo

Tiếng Anh

3. Nadia Magnenat Thalmann , Miralab Centre Universitaire , Daniel Thalmann (1999); Virtual Reality Software and Technology.
4. Tomasz Mazuryk & Michael Gervautz (1992), History, Applications, Technology and Future, Virtual Reality.
5. Jessica J.Márquez (2002), Humans and Automation Seminar, An Introduction to Virtual Reality.
6. Steven M. LaValle from University of Illinois (2016), Virtual Reality.
7. Wikipedia, Virtual reality - https://en.wikipedia.org/wiki/Virtual_reality

PHỤ LỤC

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi* thì *bảng câu hỏi* này phải được đưa vào phần Phụ lục ở dạng *nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn