

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng phần mềm hỗ trợ thiết kế nội thất sử dụng công nghệ thực tại hỗn hợp

TÔN THIỆN VĨNH

vinh.tt164769@sis.hust.edu.vn

Ngành Công nghệ thông tin

Chuyên ngành Kỹ thuật máy tính

Giảng viên hướng dẫn:

TS. Ngô Lam Trung

Chữ ký của GVHD

Bộ môn:

Kỹ thuật máy tính

Viện:

Công nghệ Thông tin và Truyền Thông

HÀ NỘI, 12/2020

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Tôn Thiện Vĩnh

MSSV: 20164769

Điện thoại liên lạc: 0902294342

Email: vinh.tt164769@sis.hust.edu.vn

Lớp: CNTT1.02-K61

Hệ đào tạo: Chính quy

Thời gian làm ĐATN: Từ ngày 01/09/2020 đến ngày 30/12/2020

2. Mục đích nội dung của ĐATN

Xây dựng phần mềm hỗ trợ thiết kế nội thất nhà sử dụng công nghệ thực tại ảo tăng cường, thực tại hỗn hợp.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu và nghiên cứu các công nghệ kỹ thuật liên quan
- Thiết kế và xây dựng phần mềm hỗ trợ thiết kế nội thất.
- Thiết kế phương thức để phần mềm dễ dàng sử dụng mô hình của các phần mềm thiết kế ngoài.
- Sử dụng các công nghệ thực tại ảo để tăng trải nghiệm tương tác với các thiết kế đây.
- Lưu trữ các mẫu mô hình và xuất dữ liệu mỗi lần thiết kế.
- Lưu trữ và quản lý cơ sở dữ liệu liên quan.

4. Lời cam đoan của sinh viên

Tôi – *Tôn Thiện Vĩnh* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Tiến sĩ Ngô Lam Trung*. Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Tôn Thiện Vĩnh

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ

Hà Nội, ngày tháng năm

Giáo viên hướng dẫn

TS. Ngô Lam Trung

LỜI CẢM ƠN

Sau năm học thứ nhất em đã được phân vào chuyên ngành Kỹ thuật máy tính, nơi em theo đuổi hết thời sinh viên, nơi gửi gắm những ước mơ và đam mê của bản thân. Em đã nhận được sự giúp đỡ nhiệt tình từ thầy cô và bạn bè đang làm việc và học tập tại đây trong suốt những năm tháng học tập tại Bộ môn.

Trước tiên, em xin gửi lời cảm ơn sâu sắc nhất đến người thầy, tiến sĩ Ngô Lam Trung, người đã tận tình chỉ bảo, hướng dẫn em suốt những năm qua, từ lúc em vẫn còn là một cậu sinh viên năm ba cho đến những ngày cuối cùng của đồ án. Thầy luôn là một người tạo cảm hứng, giúp đỡ và giải đáp các thắc mắc của em trong quá trình học tập và nghiên cứu. Một lần nữa em xin chân thành gửi lời cảm ơn tới thầy!

Em cũng xin gửi lời cảm ơn đến các thầy cô đang công tác tại Bộ môn Kỹ thuật máy tính và Viện Công nghệ Thông tin và Truyền thông, những người đã truyền đạt cho em những kiến thức tuy cơ bản nhưng cũng rất cần thiết trong việc hoàn thành đồ án và công việc mai sau.

Cuối cùng em xin cảm ơn các anh chị, bạn bè đang làm việc và học tập tại phòng thí nghiệm Hệ thống máy tính – CS Lab 505 B1 đã luôn giúp đỡ, động viên em trong quá trình nghiên cứu, học tập tại đây.

Cảm ơn tất cả mọi người! Cảm ơn Kỹ thuật máy tính! Cảm ơn Bách Khoa!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Với yêu cầu thực tế cần phải thiết kế nội thất cho một căn phòng cho sẵn nhằm giúp cho người dùng trải nghiệm và phản hồi thiết kế trước khi đưa ra sản phẩm nội thất đến tay người dùng. Tác giả đã xây dựng phần mềm hỗ trợ thiết kế nội thất nhà sử dụng công nghệ thực tại ảo. Phần mềm được xây dựng trên Unity và được hỗ trợ bởi các bộ công cụ: AR Foundation cho thiết bị di động và Mixed Reality Toolkit cho HoloLens.

Dưới đây là đề mục các chương tác giả sẽ trình bày trong đồ án và sẽ bám sát vào đó để triển khai nội dung. Đồ án được chia làm 5 chương:

Chương 1. Giới thiệu về đề tài

Chương 2. Tổng quan về các công nghệ thực tại ảo

Chương 3. Tổng quan về AR Foundation và Unity

Chương 4. Xây dựng ứng dụng

Chương 5. Kết quả và tổng kết

Trong mỗi chương, nội dung sẽ được trình bày thành các đề mục nhỏ hơn nhằm phân tích rõ vấn đề và các bước cần thiết để triển khai từng nhiệm vụ.

Sinh viên thực hiện
Ký và ghi rõ họ tên

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU VỀ ĐỀ TÀI.....	1
1.1 Đặt vấn đề	1
1.2 Các nội dung thực hiện trong đề án.....	2
1.2.1 Tìm hiểu về các công nghệ thực tại ảo.....	2
1.2.2 Khái quát hệ thống phần cứng	2
1.2.3 Khái quát cách thức làm việc của hệ thống	6
1.2.4 Tìm hiểu về các SDK hỗ trợ	7
1.2.5 Xây dựng ứng dụng hỗ trợ thiết kế nội thất.....	7
1.2.6 Thử nghiệm trên điện thoại và Hololens.....	7
1.3 Công nghệ và công cụ phát triển	7
CHƯƠNG 2. TỔNG QUAN VỀ CÁC CÔNG NGHỆ THỰC TẠI ẢO.....	8
2.1 Thực tại mở rộng	8
2.1.1 Nhập vai vào các công việc khác	9
2.1.2 Hỗ trợ thiết kế và sản xuất trong nhà máy tương lai.....	10
2.1.3 Tạo một chuyến tham quan ảo	12
2.2 Thực tại ảo	13
2.3 Thực tại tăng cường	15
2.3.1 So sánh với thực tại ảo	16
2.3.2 Các công nghệ phần cứng	17
2.3.3 Các công nghệ phần mềm	19
2.4 Thực tại hỗn hợp.....	21
2.4.1 Tính liên tục ảo/trung (Virtuality/Mediality continuum).....	21
2.4.2 Sự khác biệt so với các công nghệ khác	22
2.4.3 Vật lý liên thực tại (Interreality physics)	23
2.4.4 Ứng dụng của MR.....	23
CHƯƠNG 3. TỔNG QUAN VỀ AR FOUNDATION VÀ UNITY	24
3.1 Giới thiệu về Unity	24
3.1.1 Scene View	25
3.1.2 Game View	25
3.1.3 Cửa sổ phân cấp	26
3.1.4 Component	26
3.1.5 Prefab	27
3.1.6 Assets	27

3.1.7	Luồng chạy của component	27
3.2	Giới thiệu về AR Foundation.....	28
3.2.1	Giới thiệu về ARCore	28
3.2.2	Giới thiệu về ARKit	30
3.2.3	Giới thiệu về AR Foundation.....	31
3.3	Giới thiệu về Mixed Reality Toolkit.....	33
CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG.....		35
4.1	Phân tích thiết kế.....	35
4.1.1	Quản lý dữ liệu người dùng	35
4.1.2	Quản lý dữ liệu chương trình	36
4.1.3	Thiết kế component để quản lý vật thể	38
4.1.4	Thiết kế component cho việc tương tác	39
4.1.5	Thiết kế component quản lý lưu trữ dữ liệu.....	44
4.1.6	Mô tả quy trình thao tác	46
4.2	Cài đặt chương trình	49
4.2.1	Tạo gói nội thất mới.....	49
4.2.2	Tạo game object quản lý phiên làm việc cho thiết bị di động ..	51
4.2.3	Tạo game object quản lý phiên làm việc cho Hololens	52
4.2.4	Xây dựng hệ thống	53
CHƯƠNG 5. KẾT QUẢ VÀ TỔNG KẾT		54
5.1	Giao diện của chương trình.....	54
5.2	Thử nghiệm trên phòng lab 505, 802.....	57
5.3	Tổng kết	59
5.3.1	Các kết quả đạt được.....	59
5.3.2	Vấn đề còn tồn tại	59
5.3.3	Định hướng phát triển	59
5.3.4	Kết luận	59

DANH MỤC HÌNH VẼ

Hình 1.1 Số liệu thống kê quá trình đô thị hoá	1
Hình 1.2 Bộ công cụ google cardboard	2
Hình 1.3 Các dòng sản phẩm của Oculus Rift	3
Hình 1.4 a) Cấu tạo của Google Glass b) Một ví dụ sử dụng Google Glass để quét mã QR thiết bị y tế.....	4
Hình 1.5 Các thành phần của Hololens đời đầu	5
Hình 1.6 Khái quát cách thức làm việc của hệ thống AR	6
Hình 2.1 Tỷ lệ các ngành kinh doanh được tin là ảnh hưởng bởi XR	8
Hình 2.2 Tỷ lệ các ngành tiêu dùng được tin là ảnh hưởng bởi XR	8
Hình 2.3 Một căn buồng giả lập lái máy bay cho phi công.....	9
Hình 2.4 Tạo mẫu thử bằng AR	10
Hình 2.5 Kiểm tra các thành phần của động cơ máy bay bằng AR	10
Hình 2.6 Một bản sao kỹ thuật số của tua bin máy bay	11
Hình 2.7 Kỹ thuật viên kiểm tra bộ phận của động cơ bằng Hololens	11
Hình 2.8 Sử dụng công nghệ AR để xem trước căn hộ.....	12
Hình 2.9 Một người đang trải nghiệm VR	13
Hình 2.10 Một thiết bị HMD tiêu biểu	14
Hình 2.11 Sử dụng AR để quản lý nhà kho và tối ưu số lần chờ hàng	15
Hình 2.12 Hệ thống AR đầu tiên của không quân Mỹ.....	16
Hình 2.13 Một người đàn ông đang thử nghiệm Smartglasses	17
Hình 2.14 Một số điểm đánh dấu fiducial cho thị giác máy tính	19
Hình 2.15 Con thuyền xuất hiện do có điểm đánh dấu là bìa tờ tạp chí	20
Hình 2.16 Các AR SDK nổi tiếng	20
Hình 2.17 Sự tương tác giữa con người, máy tính và môi trường	21
Hình 2.18 Mặt phẳng thực tế trung gian liên tục bao gồm 4 điểm.....	22
Hình 2.19 Dải phổ thực tại hỗn hợp	22
Hình 2.20 Các game ứng dụng MR.....	23
Hình 3.1 Biểu tượng của unity	24
Hình 3.2 Cảnh sắp xếp model nhà trong scene view	25
Hình 3.3 Giao diện đầy đủ của Unity Editor	26
Hình 3.4 Thứ tự chạy các hàm update trong Unity	27
Hình 3.5 Một plane (màu vàng) và các điểm đặc trưng (màu đỏ)	29
Hình 3.6 Quy tắc bàn tay phải cho hệ trục Ox (+)	30
Hình 3.7 Lớp phủ các hình ảnh mà từng cảm biến ghi lại được	34
Hình 4.1 Lưu đồ thuật toán của DragMove.....	41
Hình 4.2 Lưu đồ thuật toán của DragRotate	42

Hình 4.3 Lưu đồ thuật toán của DragResize	43
Hình 4.4 Biểu đồ activity thao tác đăng nhập	46
Hình 4.5 Biểu đồ activity thao tác lưu kết quả	47
Hình 4.6 Biểu đồ activity thao tác thoát ra màn hình đăng nhập	48
Hình 4.7 Thiết kế nội thất bằng blender	49
Hình 4.8 So sánh cấu trúc mô hình trong Blender và trong Unity	49
Hình 4.9 Cấu trúc một prefab chứa mô hình nội thất	49
Hình 4.10 Bảng dán nhãn asset bundle cho asset mong muốn	50
Hình 4.11 Giao diện của AssetBundle Browser	50
Hình 4.12 Menu khởi tạo phiên làm việc AR	51
Hình 4.13 Menu import các thư viện của MRTK	52
Hình 4.14 Cách thêm và các game object đảm nhận phiên làm việc của MRTK	52
Hình 5.1 Giao diện đăng nhập của phần mềm	54
Hình 5.2 Giao diện chính của phần mềm	54
Hình 5.3 Giao diện chính của phần mềm kèm bảng danh sách nội thất	55
Hình 5.4 Giao diện khi chọn một vật thể	55
Hình 5.5 Bảng xác nhận khi ấn lưu	56
Hình 5.6 Bảng xác nhận khi ấn thoát	56
Hình 5.7 Kết quả lưu được trên CSDL	56
Hình 5.8 Thử nghiệm trên phòng 505 (tắt hiện mặt phẳng)	57
Hình 5.9 Thử nghiệm trên phòng 505 (bật hiện mặt phẳng)	57
Hình 5.10 Thử nghiệm trên phòng 505 khi đặt ở vị trí không phù hợp	57
Hình 5.11 Thử nghiệm trên phòng 505 khi đặt ở vị trí phù hợp	58
Hình 5.12 Thử nghiệm ở phòng 802	58
Hình 5.13 Thử nghiệm ở phòng 802 khi có vật thể chắn	58

DANH MỤC BẢNG BIỂU

Bảng 1-1 Cấu hình tối thiểu và cấu hình đề nghị của Rift S	3
Bảng 3-1 Các tính năng được hỗ trợ trên từng nền tảng	31
Bảng 3-2 Thông số kỹ thuật của sensor Hololens	33
Bảng 4-1 Mô tả các trường trong object user	35
Bảng 4-2 Mô tả các thuộc tính trong ObjectInfo	36
Bảng 4-3 Mô tả các trường trong object cached bundle	36
Bảng 4-4 Mô tả các trường trong object furniture	37
Bảng 4-5 Các thuộc tính quản lý vật thể được chọn	38
Bảng 4-6 Các hàm trong component ObjectManager	39
Bảng 4-7 Các thuộc tính quản lý hình bao của vật thể	39
Bảng 4-8 Các hàm vẽ hình bao của vật thể	39
Bảng 4-9 Các cặp component phụ trách tương tác giữa người dùng với vật thể	40
Bảng 4-10 Các event và event handler tương ứng	40
Bảng 4-11 Các thuộc tính lưu trữ thông tin người dùng	44
Bảng 4-12 Các hàm liên quan đến lưu trữ thông tin người dùng	44
Bảng 4-13 Các thuộc tính dùng để lưu các asset bundle đã tải	45
Bảng 4-14 Các hàm dùng để tải cũng như lưu asset bundle	45

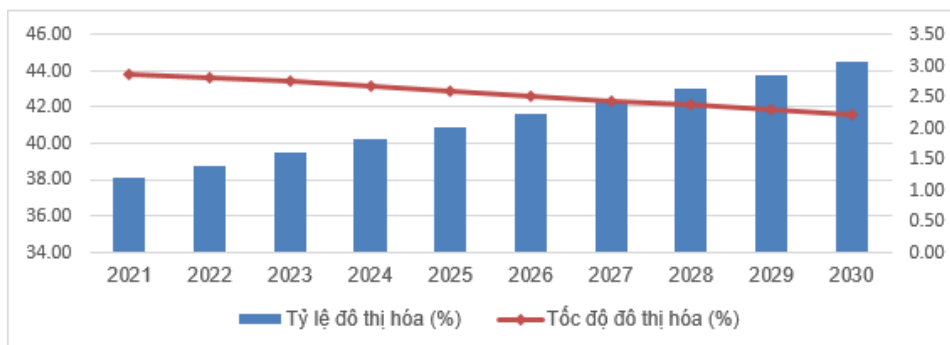
DANH MỤC CÁC TỪ VIẾT TẮT

XR	Extended Reality
VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
API	Appilcation Programming Interface
SDK	Software Development Kit
DLL	Dynamic Linker Library
CSDL	Cơ Sở Dữ Liệu

CHƯƠNG 1. GIỚI THIỆU VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Ngày nay, xã hội phát triển, đời sống người dân ngày càng được cải thiện kéo theo quá trình đô thị hoá cũng diễn ra mạnh mẽ, tạo hiệu ứng thúc đẩy đô thị hoá nhanh lan toả trên khắp phạm vi cả nước. Theo Trung tâm Thông tin & Dự báo kinh tế - xã hội quốc gia, xu hướng đô thị hóa ở Việt Nam đã, đang và sẽ được tiếp tục mở rộng đến các vùng lân cận, các thành phố vệ tinh, các thành phố nhỏ và vừa.



Nguồn: Dự báo của nhóm nghiên cứu từ số liệu của UN.

Hình 1.1 Số liệu thống kê quá trình đô thị hoá

Số lượng người dân khu vực thành thị tăng, đồng nghĩa với việc nguồn cầu căn hộ ở các thành phố lớn cũng tăng theo. Không chỉ vậy, ngày nay, cuộc sống vật chất và tinh thần trên mặt bằng chung xã hội ngày một tiến bộ, không chỉ đáp ứng những nhu cầu cơ bản, con người còn muốn được hưởng thụ chất lượng nhiều và cao hơn từ cuộc sống, chính vì lí do đó mà các nhu cầu về nội thất cũng tăng cao. Thiết kế nội thất đã trở thành một việc không thể thiếu.

Mặc dù những năm qua công nghệ đã có sự phát triển nhanh và đột phá nhưng khi người tiêu dùng muốn thiết kế nội thất vẫn phải đi qua các bước thủ công: các nhà thiết kế sẽ xây dựng mô hình 3D của cả căn phòng cùng với các nội thất bên trong. Điều này dẫn đến việc mọi thứ đều là mô hình máy tính, không đem lại cho khách hàng một trải nghiệm chân thật đồng thời có thể dẫn đến những sự cố khách quan, ví dụ như không thể kiểm tra được màu sắc trong không gian dẫn đến việc khi ở trong mô hình thì đồ vật rất đẹp và bắt mắt nhưng khi được đặt trong tổng thể không gian trong nhà thì nó lại trở nên lạc lõng.

Do những hạn chế trên, tác giả đã xây dựng một phần mềm thiết kế nội thất mới sử dụng công nghệ thực tại tăng cường. Khi đó căn phòng người dùng xem xét sẽ là một căn phòng thật tuy nhiên nội thất đặt trong đó là một mô hình 3D ảo. Và với thực tại hỗn hợp, chương trình sẽ tăng tính chân thực nhằm đem lại một trải nghiệm tốt hơn.

1.2 Các nội dung thực hiện trong đề án

1.2.1 Tìm hiểu về các công nghệ thực tại ảo

Các công nghệ thực tại ảo đang là xu hướng của tương lai. Bằng sức mạnh của công nghệ, chỉ với tai nghe và các cảm biến chuyển động, người dùng đã có thể hoà mình vào một không gian ảo như thể đang thực sự ở trong đó. Trong những năm gần đây, nó đã có một sự hấp dẫn nhất định nhờ vào các trò chơi, phim, ảnh với một trải nghiệm hấp dẫn. Trong chương sau, tác giả sẽ tìm hiểu về các công nghệ đó.

1.2.2 Khái quát hệ thống phần cứng

Các công nghệ thực tại ảo hỗ trợ một hệ thống phần cứng đa dạng và phong phú, từ một chiếc điện thoại thông minh với giá thành phải chăng cho đến một bộ thiết bị bao gồm các cảm biến phức tạp.

Với chiếc điện thoại thông minh, chỉ cần có những thành phần sau đã có thể trải nghiệm AR:

- Bộ xử lý ARM 64-bit
- Camera
- Cảm biến độ sâu (nếu có)
- Cảm biến màu sắc
- Con quay hồi chuyển
- La bàn thể rắn

Và với một khung giấy các-tông đơn giản, người dùng có thể biến chiếc điện thoại của mình thành một bộ thiết bị đeo tai để hoà mình vào không gian ảo với công nghệ VR:



Hình 1.2 Bộ công cụ google cardboard

Oculus Rift là một dòng sản phẩm kính VR giúp tăng trải nghiệm của người dùng.



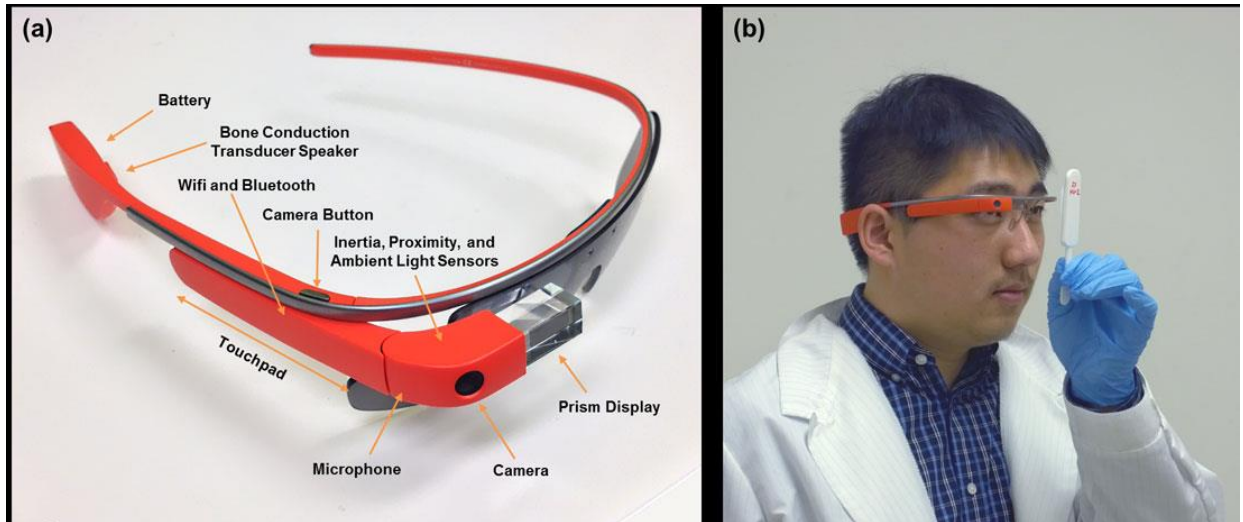
Hình 1.3 Các dòng sản phẩm của Oculus Rift

Rift S có màn hình LCD 1280x1440 với tần số quét 80Hz với một trường nhìn 115°. Rift S theo dõi vị trí của chính nó và các bộ điều khiển của nó trong không gian 3D bằng cách sử dụng một hệ thống được gọi là Oculus Insight, sử dụng 5 camera trên thiết bị để theo dõi các điểm trong môi trường và đèn LED hồng ngoại trên bộ điều khiển, thông tin từ gia tốc kế và thị giác máy tính để dự đoán con đường mà nó và bộ điều khiển có nhiều khả năng đi theo. Để sử dụng Rift S người dùng phải kết nối với một máy tính có cấu hình:

Component	Recommended Specs	Minimum Specs
Processor	Intel i5-4590 / AMD Ryzen 5 1500X or greater	Intel i3-6100 / AMD Ryzen 3 1200, FX4350 or greater
Graphics Card	NVIDIA GTX 1060 / AMD Radeon RX 480 or greater	NVIDIA GTX 1050 Ti / AMD Radeon RX 470 or greater
Alternative Graphics Card	NVIDIA GTX 970 / AMD Radeon R9 290 or greater	NVIDIA GTX 960 4GB / AMD Radeon R9 290 or greater
Memory	8GB+ RAM	8GB+ RAM
Operating System	Windows 10	Windows 10
USB Ports	1x USB 3.0 ports	1x USB 3.0 port
Video Output	Compatible DisplayPort video output	Compatible miniDisplayPort video output (miniDisplayPort to DisplayPort adapter included with Rift S)

Bảng 1-1 Cấu hình tối thiểu và cấu hình đề nghị của Rift S

Google Glass là thiết bị công nghệ đeo được với màn hình quang học gắn trên đầu (OHMD). Google Glass được phát triển bởi Google với nhiệm vụ phổ biến một thị trường máy tính đại trà và đã tạo nên một làn sóng mới trong thị trường kính AR. Google Glass hiển thị thông tin như trên smartphone ở chế độ rảnh tay. Người đeo giao tiếp với thiết bị thông qua điều khiển giọng nói



Hình 1.4 a) Cấu tạo của Google Glass
b) Một ví dụ sử dụng Google Glass để quét mã QR thiết bị y tế

Google Glass có các thông số kỹ thuật như sau:

- Hệ điều hành: Android (4.4.2)
- CPU: OMAP 4430 SoC, lõi-kép
- Bộ nhớ: 1GB RAM (682MB sẵn cho nhà phát triển)
- Bộ nhớ lưu trữ: 16 GB Flash (12 GB có thể sử dụng)
- Hiển thị: Lăng kính chiếu, 640×360 pixels (tương đương với màn hình từ 25 in/64 cm đến 8 ft/2,4 m)
- Âm thanh: Bone conduction transducer
- Đầu vào: Điều khiển bằng giọng nói thông qua microphone, gia tốc, con quay, từ kế, cảm biến ánh sáng xung quanh, cảm biến gần
- Điều khiển: Touchpad, MyGlass phone app
- Máy ảnh: Ảnh – 5 MP, video – 720p
- Kết nối: Wi-Fi 802.11b/g, Bluetooth, micro USB
- Khả năng tương thích ngược: tất cả điện thoại hỗ trợ bluetooth; MyGlass tương thích với ứng dụng Android 4.0.3 (Ice Cream Sandwich) hoặc cao hơn hoặc iOS 7.0 hoặc cao hơn

Đối với thực tại hỗn hợp, Microsoft đã giới thiệu kính thực tế ảo HoloLens:

- Bộ xử lý Holographic được Microsoft xây dựng riêng
- 1 đơn vị đo lường quán tính (IMU)
- 4 camera nhận diện môi trường
- 1 camera nhận diện độ sâu
- 1 camera 2MP / quay phim HD
- 1 module chụp thực tại hỗn hợp
- 4 micrô
- 1 cảm biến ánh sáng xung quanh

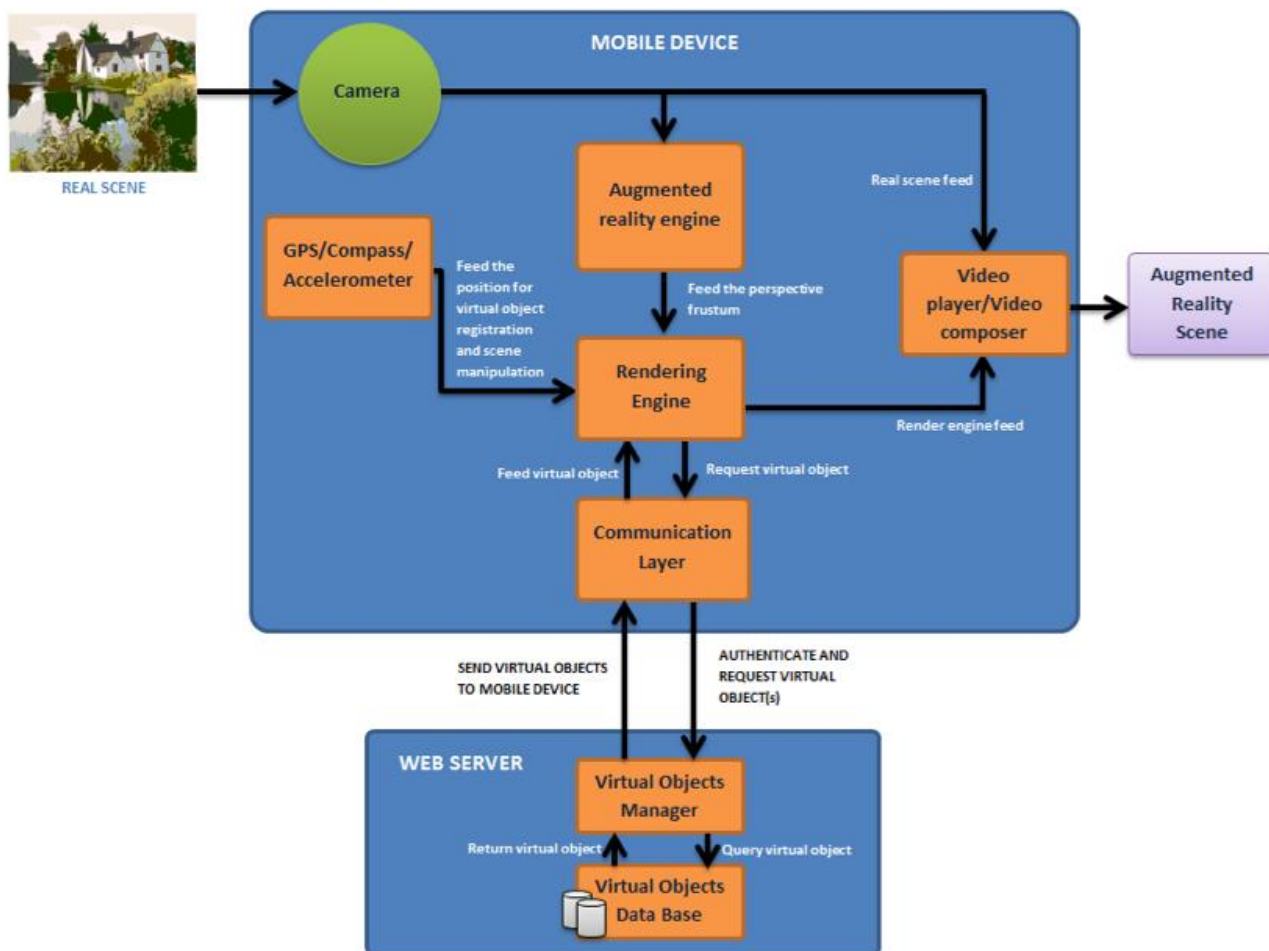


Hình 1.5 Các thành phần của HoloLens đời đầu

1.2.3 Khái quát cách thức làm việc của hệ thống

Camera và các cảm biến sẽ thu nhập các tín hiệu từ môi trường xung quanh rồi gửi đến bộ xử lý để tính toán được môi trường xung quanh người đeo, từ đó có thể bao phủ một lớp kỹ thuật số lên trên các vật thể ở thế giới thực thông qua màn hình hiển thị.

Người dùng có thể tương tác được với các vật thể kỹ thuật số đó thông qua các đầu vào, với AR sẽ là màn hình của thiết bị, với MR sẽ là thông qua cử chỉ, giọng nói cũng như ánh mắt của người dùng mà các cảm biến sẽ chuyển thành các đầu vào tương ứng.



Hình 1.6 Khái quát cách thức làm việc của hệ thống AR

Các vật thể ảo sẽ tương tác với thế giới thực với các tính chất tương tự như vật thể ở thế giới thực như đổ bóng, đứng trên mặt đất,...

1.2.4 Tìm hiểu về các SDK hỗ trợ

Để giúp các nhà phát triển có thể xây dựng ứng dụng một cách nhanh chóng và thuận tiện, các nhà phát hành như Google, Microsoft, Facebook,... đã tung ra các SDK như AR Core, Mixed Reality Toolkit hay Oculus SDK. Các bộ công cụ này hỗ trợ các chức năng như theo dõi đối tượng 3D, multi-tracking (đa theo dõi), nhận dạng hình ảnh,... Những SDK này thường dành cho các framework (khung chương trình) và phần cứng cụ thể; tuy nhiên, một số SDK cũng hỗ trợ trên nhiều nền tảng tạo ra một sự linh hoạt.

1.2.5 Xây dựng ứng dụng hỗ trợ thiết kế nội thất

Phần mềm được xây dựng ra với mục đích chính là mang lại trải nghiệm chân thực cho người sử dụng khi thiết kế nội thất, giúp người dùng có thể tương tác được với các mẫu mô hình kỹ thuật số của các đồ nội thất. Vì vậy sẽ có sẵn các mẫu mô hình được lưu trong thiết bị. Đồng thời ứng dụng cũng có thể cập nhật thêm các gói mô hình do nhà cung cấp nội thất bên ngoài phân phối đến.

Với mỗi loại nội thất sẽ có những tính chất khác nhau, do đó người dùng phải chọn đúng đồ vật và đặt ở đúng nơi qui định.

Sau khi sắp xếp một cách ưng ý người dùng có thể lưu lại thông tin như ngày tạo, số lượng các vật phẩm cũng như thuộc tính của từng vật phẩm đó.

Do đó cần xây dựng phần mềm để phục vụ cho phần cứng với các nhiệm vụ sau:

- Nhận diện được các mặt phẳng kỹ thuật số đúng với sàn và tường trong thế giới thực được hiển thị trên màn hình
- Có thể tương tác với các mô hình 3D của nội thất
- Có thể báo cho người dùng xem sự sắp xếp này có hợp lý hay không
- Có thể thêm/xoá các gói mô hình từ nguồn ngoài
- Lưu trữ và xử lý dữ liệu thu thập được:
 - o Quản lý dữ liệu của lần sắp xếp
 - o Xuất kết quả ra CSDL bên ngoài

1.2.6 Thử nghiệm trên điện thoại và Hololens

Sau khi xây dựng và cài đặt xong ứng dụng, tác giả sẽ thử nghiệm trên thiết bị tương ứng trong môi trường xác định. Cụ thể với điện thoại sẽ thử nghiệm trên phòng lab 505, 802 còn với Hololens sẽ tạm thời thử nghiệm trên giả lập Hololens Emulator.

1.3 Công nghệ và công cụ phát triển

Có khá nhiều framework hỗ trợ phát triển các công nghệ thực tại mở rộng hỗ trợ đa nền tảng như Unity Engine, Unreal Engine, 3ds Max,... Nhưng phù hợp nhất cũng như để tối ưu thời gian phát triển, tính đơn giản, linh hoạt trong lập trình, tác giả đã chọn sử dụng Unity Engine với ngôn ngữ lập trình C#.

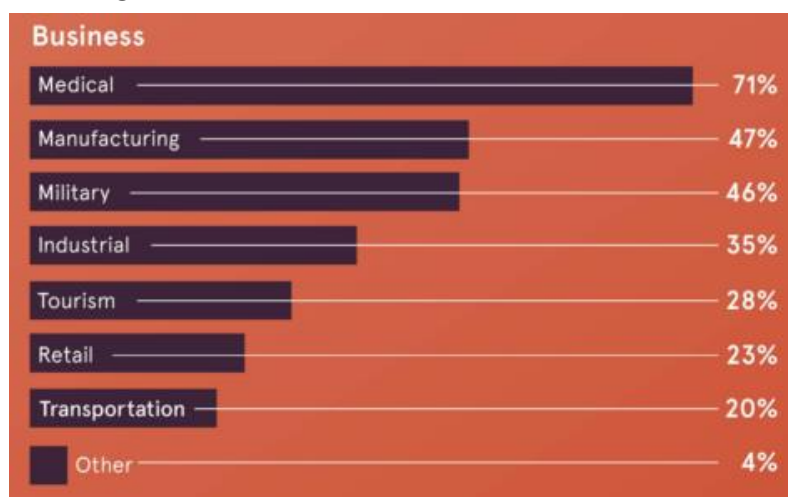
CHƯƠNG 2. TỔNG QUAN VỀ CÁC CÔNG NGHỆ THỰC TẠI ẢO

2.1 Thực tại mở rộng

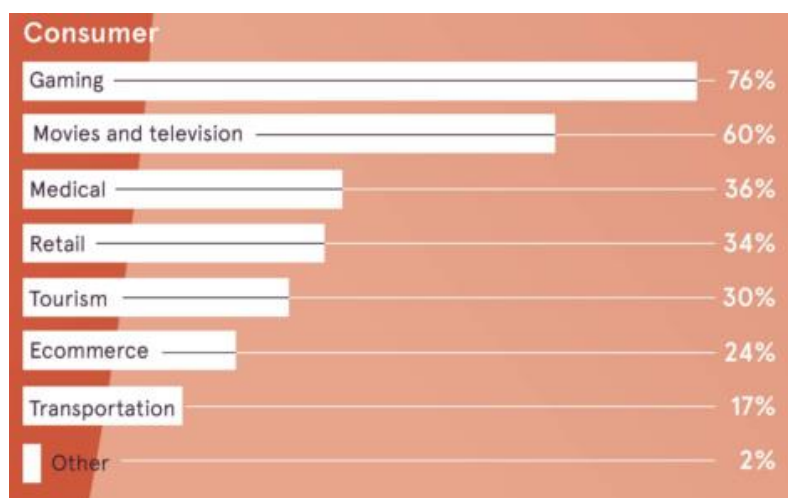
Thực tại mở rộng (XR) là một thuật ngữ đề cập đến tất cả các môi trường kết hợp thực và ảo và các tương tác giữa người và máy được tạo ra bởi máy tính và các thiết bị đeo được, trong đó 'X' đại diện cho một biến số cho bất kỳ công nghệ điện toán không gian hiện tại hoặc tương lai. Ví dụ: Nó bao gồm các hình thức như thực tại tăng cường (AR), thực tại hỗn hợp (MR) và thực tại ảo (VR).

XR là một tập bao gồm từ "thực hoàn toàn" đến "ảo hoàn toàn" trong khái niệm liên tục thực - ảo do Paul Milgram đưa ra. Tuy nhiên, ý nghĩa của nó nằm ở việc mở rộng trải nghiệm của con người, đặc biệt là liên quan đến việc cảm nhận sự tồn tại (được đại diện bởi VR) và thu nhận nhận thức (được đại diện bởi AR). Với sự phát triển liên tục trong tương tác giữa con người và máy tính, ý nghĩa này vẫn đang tiếp tục phát triển.

XR là một lĩnh vực đang phát triển nhanh chóng đang được áp dụng theo nhiều cách, chẳng hạn như giải trí, tiếp thị, bất động sản, đào tạo và làm việc từ xa. Đến năm 2022, thị trường XR được dự đoán đạt mốc 209 tỉ USD.



Hình 2.1 Tỷ lệ các ngành kinh doanh được tin là ảnh hưởng bởi XR



Hình 2.2 Tỷ lệ các ngành tiêu dùng được tin là ảnh hưởng bởi XR

2.1.1 Nhập vai vào các công việc khác

XR ngày càng được sử dụng nhiều bởi các công ty để sử dụng cho mục đích đào tạo một cách sáng tạo và an toàn. Ví dụ như tìm hiểu về một nghề, học các thực hành nghề nghiệp và hướng dẫn an toàn, phát triển kỹ năng,...

Lợi thế của các công nghệ nhập vai là chúng có thể tái tạo môi trường làm việc và tạo ra rất nhiều kịch bản, bao gồm cả những kịch bản nhiều rủi ro hoặc khó để tạo ra trong thực tế, bằng cách cung cấp cho người sử dụng một sự nhập vai có độ tương tác cao.

Người học có thể phát triển trong một môi trường mô phỏng siêu thực tế, hoặc nắm bắt một hệ thống phức tạp và thực hiện các bài tập được giao. Người huấn luyện có thể phân tích các chiến lược đã được triển khai. Khi ở trong điều kiện thực tế, người được đào tạo sẽ hiểu rõ hơn các phương pháp chuyên môn và áp dụng chúng dễ dàng hơn.

Đối với những công việc đôi khi nguy hiểm (như trong các nhà máy điện, công trường xây dựng,...) hoặc đối với những ngành nghề có trách nhiệm cao (bác sĩ phẫu thuật, phi công hàng không,...), đào tạo XR là một cách chuẩn bị cho sinh viên đối với các tình huống khẩn cấp, chẳng hạn như can thiệp tại chỗ mà không lo sợ các rủi ro liên quan.



Hình 2.3 Một cabin giả lập lái máy bay cho phi công

XR cũng góp phần vào việc học tập hiệu quả hơn. Trong lĩnh vực giáo dục và đào tạo, lợi ích của mô phỏng đã được đề cao từ lâu. “Được tăng cường” bởi XR, chúng cho phép việc học ít lý thuyết hơn mà dựa trên kinh nghiệm thực tế, điều này sẽ thúc đẩy người học nhiều hơn và củng cố kiến thức lâu dài hơn.

2.1.2 Hỗ trợ thiết kế và sản xuất trong nhà máy tương lai

Được xác định là chiến lược cho cách mạng công nghiệp 4.0, công nghệ nhập vai tham gia vào tất cả các giai đoạn của chuỗi sản xuất, từ nghiên cứu và phát triển đến đưa sản phẩm ra thị trường.

Trong quá trình lên ý tưởng, họ giúp các nhà thiết kế và kỹ sư xây dựng cũng như thử nghiệm các sản phẩm công nghiệp. Ví dụ: tạo mẫu thử bằng thực tế ảo cung cấp cơ hội để trực quan hóa mô hình 3D của các vật thể thực được thiết kế bằng phần mềm hỗ trợ máy tính, qua đó mô phỏng hoạt động của chúng trong điều kiện thực tế, phân tích kết quả và tối ưu hóa thiết kế của chúng.



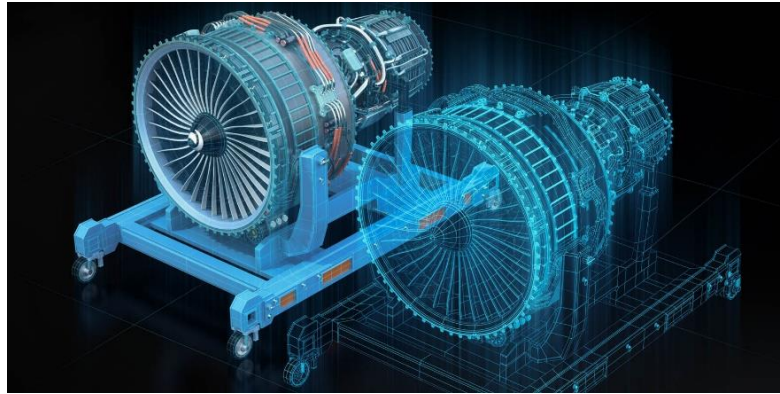
Hình 2.4 Tạo mẫu thử bằng AR

Từ đó giúp hạn chế các nguyên mẫu và thử nghiệm vật lý, việc sử dụng nguyên mẫu ảo ở giai đoạn đầu của chu kỳ phát triển giúp giảm chi phí thiết kế và độ trễ, đồng thời tối ưu hóa chất lượng sản phẩm. Trên thực tế, chúng cung cấp cho các kỹ sư cơ hội so sánh hiệu suất của các mô hình khác nhau nhanh hơn và xác định quy trình sản xuất tốt nhất.



Hình 2.5 Kiểm tra các thành phần của động cơ máy bay bằng AR

Trong lĩnh vực bảo trì, bản sao kỹ thuật số được sử dụng để giám sát và tối ưu hóa hoạt động của các thiết bị công nghiệp. Là một bản sao kỹ thuật số của một hệ thống thực (máy móc, quy trình công nghiệp, dây chuyền lắp ráp hoặc thậm chí là toàn bộ nhà máy) được cung cấp dữ liệu được thu thập bởi các cảm biến IoT, bản sao kỹ thuật số cung cấp thông tin về tình trạng và hiệu suất của hệ thống này trong suốt vòng đời của nó. Nhờ công nghệ máy học, nó cũng có thể đưa ra dự đoán để dự đoán bất kỳ sự cố tiềm ẩn nào.



Hình 2.6 Một bản sao kỹ thuật số của tua bin máy bay

Điều này dẫn đến một ứng dụng thứ ba của XR trong ngành: hỗ trợ hoạt động. Đặc biệt MR giúp hướng dẫn người vận hành thực hiện nhiệm vụ bằng cách cung cấp thông tin theo ngữ cảnh trong thời gian thực. Ví dụ: trong quá trình kiểm soát chất lượng động cơ, kỹ thuật viên - được trang bị Hololens - có thể trực quan hóa biểu diễn kỹ thuật số của mọi bộ phận của động cơ, chồng lên động cơ thực, cũng như hiển thị kế hoạch, hướng dẫn, lời khuyên,... trong suốt quá trình, đồng thời vẫn giữ cho họ rảnh tay.



Hình 2.7 Kỹ thuật viên kiểm tra bộ phận của động cơ bằng Hololens

2.1.3 Tạo một chuyến tham quan ảo

Trong lĩnh vực dịch vụ, các nhà phát triển bất động sản và đại lý bất động sản cũng đang chuyển sang sử dụng XR để thu hút người mua tiềm năng. Mục đích là tạo ra trải nghiệm sống động và tương tác để giúp họ tưởng tượng chính mình trong ngôi nhà tương lai của họ.

Do đó, chuyến tham quan ảo trước hết cho phép người mua tiềm năng có một cái nhìn ban đầu về bất động sản trước khi lên kế hoạch cho một chuyến tham quan thực tế, đặc biệt hữu ích khi họ định mua ở một nơi xa. Quan trọng hơn, nó cho phép họ khám phá bất động sản như thể họ đang ở tại chỗ, mặc dù nó có thể chưa được xây dựng, đây là một khía cạnh đáng kể khi xem xét mua một ngôi nhà hoặc căn hộ mới.



Hình 2.8 Sử dụng công nghệ AR để xem trước căn hộ

Mặc dù ảnh 360 độ đã cho phép người mua thể hiện trung thực nội thất, nhưng VR còn tiến xa hơn nữa bằng cách “dịch chuyển” chúng thẳng vào nhà, nơi họ có thể đi từ phòng này sang phòng khác. Nhờ AR, chủ sở hữu có thể hình dung một tòa nhà tại chỗ trước khi nó được xây dựng. Được trang bị điện thoại thông minh hoặc máy tính bảng, họ có thể đến vùng đất nơi công việc đang được thực hiện để hình dung ngôi nhà hoặc nơi ở tương lai của họ ở dạng 3D với kích thước thực.

Không chỉ vậy người bán có thể cung cấp các tùy chỉnh của bất động sản, từ sàn đến trần. Trong chuyến tham quan một ngôi nhà, đại lý bất động sản có thể cung cấp cho khách hàng cái nhìn về các kế hoạch khả thi (kéo vách ngăn xuống để làm nhà bếp không gian mở, xây tủ quần áo không cửa ngăn,...) và các tùy chọn tùy chỉnh. Nhờ thực tại hỗn hợp, người mua thậm chí có thể thiết kế nội thất, bằng cách thêm đồ nội thất vào một căn phòng trống và thử tất cả các loại kết cấu trên sàn và tường.

2.2 Thực tại ảo

Thực tại ảo (VR) là một môi trường giả lập mô phỏng do máy tính tạo ra, trong đó một người có thể tương tác trong môi trường ba chiều nhân tạo bằng các thiết bị điện tử, chẳng hạn như kính bảo hộ đặc biệt có màn hình hoặc găng tay có gắn cảm biến. Trong môi trường nhân tạo giả lập này, người dùng có thể có trải nghiệm cảm giác thực tế.



Hình 2.9 Một người đang trải nghiệm VR

Khái niệm thực tại ảo được xây dựng dựa trên sự kết hợp tự nhiên của hai từ: “ảo” và “thực”. Trong đó từ “ảo” có nghĩa là "gần như" hoặc "là một sự vật hoặc hiện tượng nào đó, mặc dù không thực sự có thật", dẫn đến trải nghiệm gần như thực tế thông qua việc sử dụng công nghệ. Phần mềm tạo ra một thế giới ảo được trải nghiệm bởi những người dùng đeo thiết bị phần cứng đặc biệt như kính bảo hộ, tai nghe và găng tay. Qua đó, người dùng có thể xem và tương tác với thế giới ảo như thể từ bên trong.

Chúng ta hiểu môi trường xung quanh thông qua các giác quan và cơ chế nhận thức của cơ thể. Các giác quan bao gồm vị giác, xúc giác, khứu giác, thị giác và thính giác, cũng như nhận thức và cân bằng không gian. Các yếu tố đầu vào do các giác quan này thu thập được bộ não của chúng ta xử lý để đưa ra các diễn giải về môi trường khách quan xung quanh chúng ta. Thực tại ảo cố gắng tạo ra một môi trường ảo tưởng có thể hiển thị cho các giác quan của chúng ta bằng thông tin nhân tạo, khiến tâm trí chúng ta tin rằng nó (gần như) là một thực tại.

Ví dụ thực tế nhất về VR là phim ba chiều (3D). Sử dụng kính 3D đặc biệt, người ta có được trải nghiệm đắm chìm như một phần của bộ phim mặc dù chỉ ngồi tại chỗ xem. Chiếc lá rơi từ trên cây dường như trôi lơ lửng ngay trước mặt người xem, hoặc cảnh một chiếc xe ô tô phóng nhanh vượt qua vách đá khiến người xem cảm nhận được độ sâu của vực sâu và có thể khiến một số người xem cảm giác như thể muốn rơi xuống. Về cơ bản, hiệu ứng ánh sáng và âm thanh của một bộ phim 3D làm cho thị giác và thính giác của chúng ta tin rằng tất cả đang diễn ra ngay trước mặt chúng ta, mặc dù không có gì tồn tại trong thực tế vật lý.

Các tiến bộ công nghệ đã cho phép nâng cao trải nghiệm hơn nữa chứ không còn chỉ bị giới hạn bởi kính 3D tiêu chuẩn. Giờ đây, người ta có thể tìm thấy các bộ màn hình hiển thị gắn trên đầu (head-mounted display – HMD) để khám phá nhiều hơn nữa. Với sự hỗ trợ của hệ thống máy tính, giờ đây một người có thể chơi quần vợt "thực" (hoặc các môn thể thao khác) ngay trong phòng khách của họ bằng cách cầm những cây vợt được trang bị cảm biến để chơi trong một trò chơi mô phỏng trò chơi do máy tính điều khiển. Tai nghe VR mà người chơi đeo trên mắt mang lại cảm giác như đang ở trên sân quần vợt. Họ di chuyển và cố gắng đánh tùy thuộc vào tốc độ và hướng của bóng tới và đánh nó bằng các cây vợt được trang bị cảm biến. Độ chính xác của cú đánh được đánh giá bởi máy tính điều khiển trò chơi, được hiển thị trong trò chơi VR tương ứng - cho biết quả bóng bị đánh quá mạnh và đi ra ngoài biên hay bị đánh quá mềm và bị lưới chặn lại.



Hình 2.10 Một thiết bị HMD tiêu biểu

Các ứng dụng khác của công nghệ VR này liên quan đến đào tạo và mô phỏng. Ví dụ: những người muốn lấy bằng lái xe có thể trải nghiệm trực tiếp lái xe đường trường bằng cách sử dụng thiết lập VR liên quan đến việc xử lý các bộ phận của ô tô như vô lăng, phanh và chân ga. Nó mang lại lợi ích của trải nghiệm mà không có khả năng gây ra tai nạn, vì vậy học viên có thể phát triển một trình độ chuyên môn nhất định về lái xe trước khi thực sự tham gia trên đường.

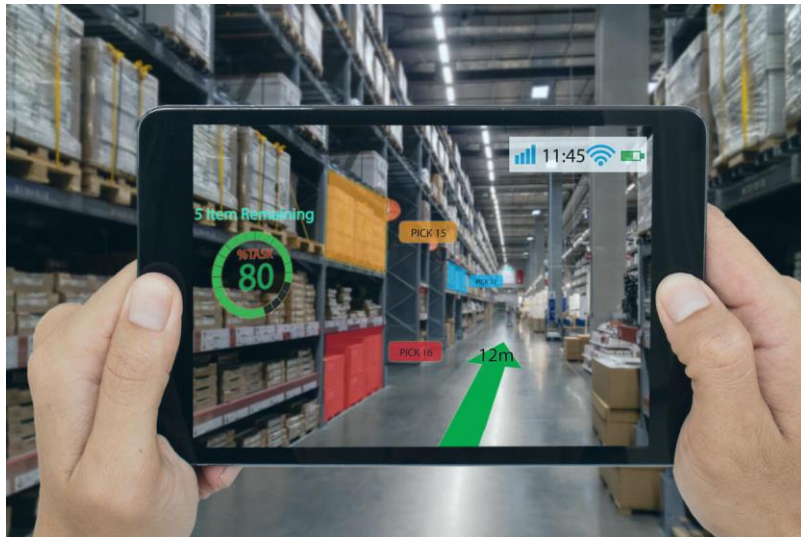
Người bán bất động sản cũng có thể sử dụng các hướng dẫn về ngôi nhà hoặc căn hộ có hỗ trợ thực tế ảo để đưa ra cảm nhận về một bất động sản mà không cần phải thực sự có mặt tại địa điểm với người mua tiềm năng.

Các ứng dụng đang phát triển khác là đào tạo các phi hành gia để du hành không gian, khám phá sự phức tạp của các vật thể thu nhỏ và cho phép sinh viên y khoa thực hành phẫu thuật trên các chủ đề do máy tính tạo ra.

2.3 Thực tại tăng cường

Thực tại tăng cường (AR) là một trải nghiệm tương tác ngay tại môi trường thế giới thực, nơi các đối tượng được tăng cường bởi thông tin dữ liệu do máy tính tạo ra, có thể qua nhiều phương thức cảm giác khác nhau, bao gồm thị giác, thính giác, xúc giác, khứu giác và khứu giác.

AR có thể được định nghĩa như là một hệ thống đáp ứng được ba tính năng cơ bản bao gồm: sự kết hợp giữa hai thế giới thực và ảo, tương tác thời gian thực trên đó và hiển thị một chính xác các đối tượng ảo dưới dạng 3D và nhận diện được các đối tượng thực. Thông tin cảm quan được thể hiện cho người dùng có thể mang tính xây dựng (tức là thêm vào trong môi trường tự nhiên) hoặc phá hủy (tức là che lấp/xoá bỏ trong môi trường tự nhiên). Trải nghiệm này được đan xen một cách liền mạch với vật chất hiện hữu sao cho đối với môi trường thực tế nó được coi như là một khía cạnh nhập vai. Với cách này, thực tại tăng cường liên tục thay đổi nhận thức của người dùng về môi trường thế giới thực, trong khi thực tại ảo lại thay thế hoàn toàn môi trường thế giới thực của người dùng bằng một môi trường giả lập.



Hình 2.11 Sử dụng AR để quản lý nhà kho và tối ưu số lần chờ hàng

Thực tại tăng cường mang giá trị to lớn nằm ở cách thức mà các thành phần của thế giới ảo (thế giới kỹ thuật số) hòa vào nhận thức của một người về thế giới thực, không phải chỉ là một màn hình hiển thị dữ liệu đơn giản, mà thông qua việc tích hợp các cảm giác nhập vai vốn là các phần tự nhiên của một môi trường.

Các hệ thống AR sớm nhất nhằm cung cấp trải nghiệm thực tại hỗn hợp cho người dùng được phát minh vào đầu những năm 1990, với hệ thống đồ đặc ảo được phát triển tại Phòng thí nghiệm Armstrong của Không quân Hoa Kỳ vào năm 1992. Trải nghiệm của thực tại tăng cường lần đầu tiên được giới thiệu và thương mại trong các doanh nghiệp giải trí, trò chơi. Sau đó, các ứng dụng của nó đã được mở rộng sang các ngành như giáo dục, truyền thông, y học và giải trí.

Trong giáo dục, nội dung có thể được truy cập bằng cách quét hoặc xem hình ảnh bằng thiết bị di động.



Hình 2.12 Hệ thống AR đầu tiên của không quân Mỹ

Thực tại tăng cường được sử dụng để cải thiện môi trường thực hoặc các tình huống tự nhiên và cung cấp trải nghiệm phong phú về mặt tri giác. Với sự trợ giúp của các công nghệ AR tiên tiến (ví dụ: thêm tầm nhìn máy tính, kết hợp camera AR vào các ứng dụng điện thoại thông minh và nhận dạng đối tượng), thông tin về thế giới thực xung quanh của người dùng có thể tương tác được và được thao tác dưới dạng kỹ thuật số. Thông tin về môi trường và các đối tượng của nó được phủ lên thế giới thực. Thông tin này có thể là ảo hoặc thực, ví dụ: nhìn thấy các thông tin đo lường hoặc cảm nhận thực thể khác như sóng vô tuyến điện từ được thể hiện một cách chính xác so với vị trí thực sự của chúng trong không gian.

Thực tại tăng cường cũng có rất nhiều tiềm năng trong việc thu thập và chia sẻ kiến thức ngầm. Các kỹ thuật tăng cường thường được thực hiện trong thời gian thực và trong bối cảnh ngữ nghĩa với các yếu tố môi trường. Thông tin tri giác sâu sắc đôi khi được kết hợp với thông tin bổ sung, ví dụ như điểm số qua nguồn cấp video trực tuyến về một sự kiện thể thao. Điều này kết hợp những lợi ích của cả công nghệ thực tại tăng cường và công nghệ hiển thị màn hình ngẩng đầu (heads up display - HUD).

2.3.1 So sánh với thực tại ảo

Trong thực tại ảo (VR), cảm nhận của người dùng về thực tại hoàn toàn dựa trên thông tin ảo. Trong thực tại tăng cường (AR), người dùng được cung cấp thêm các thông tin do máy tính tạo ra để nâng cao nhận thức của họ về thực tế đó.

Ví dụ, trong kiến trúc, VR có thể được sử dụng để tạo ra mô phỏng nội thất bên trong của một tòa nhà mới; và AR có thể được sử dụng để hiển thị cấu trúc và hệ thống của một tòa nhà qua việc áp đặt vào chế độ xem ngoài đời thực.

Một ví dụ khác là thông qua việc sử dụng các ứng dụng tiện ích. Một số ứng dụng AR, chẳng hạn như Augment, cho phép người dùng đặt các đối tượng kỹ thuật số vào môi trường thực, cho phép các doanh nghiệp sử dụng thiết bị AR như một cách để xem trước sản phẩm của họ trong thế giới thực. Tương tự, nó

cũng có thể được sử dụng như một cách để giới thiệu sản phẩm trông như thế nào cho khách hàng, tiêu biểu như các công ty như Mountain Equipment Co-op hoặc Lowe's, những người sử dụng thực tế tăng cường để cho phép khách hàng xem trước sản phẩm của họ trông như thế nào tại nhà thông qua việc sử dụng các mô hình 3D.

Thực tại tăng cường (AR) khác với thực tại ảo (VR) ở chỗ là trong AR, một phần của môi trường xung quanh thực sự là “thực” và chỉ thêm các lớp vật thể ảo vào môi trường thực. Mặt khác, trong VR, môi trường xung quanh hoàn toàn là ảo.

2.3.2 Các công nghệ phần cứng

Các thành phần phần cứng cần thiết cho thực tại tăng cường là: bộ xử lý, màn hình, cảm biến và thiết bị đầu vào. Các thiết bị di động hiện đại như điện thoại thông minh và máy tính bảng chứa các yếu tố này, thường bao gồm máy ảnh và các cảm biến của hệ thống vi cơ điện tử (MEMS) như gia tốc kế, GPS và la bàn trạng thái rắn, làm cho chúng trở thành thiết bị phù hợp với nền tảng AR. Có hai công nghệ được sử dụng trong thực tế tăng cường: ống dẫn sóng nhiễu xạ và ống dẫn sóng phản xạ.



Hình 2.13 Một người đàn ông đang thử nghiệm Smartglasses

Các thành phần màn hình:

- Màn hình gắn đầu (HMD): là một thiết bị hiển thị đeo trên trán, chẳng hạn như dây nịt hoặc gắn trên mũ bảo hiểm. HMD đưa hình ảnh của cả thế giới thực và các đối tượng ảo lên tầm nhìn của người dùng. Các HMD hiện đại thường sử dụng các cảm biến để giám sát sáu bậc tự do cho phép hệ thống điều chỉnh thông tin ảo với thế giới thực và điều chỉnh phù hợp với chuyển động đầu của người dùng. HMD có thể cung cấp cho người dùng VR trải nghiệm di động và cộng tác. Các nhà cung cấp cụ thể, chẳng hạn như uSens và Gestigon, bao gồm các điều khiển cử chỉ để hoàn toàn đắm chìm trong ảo.

- Kính mắt: màn hình AR có thể được hiển thị trên các thiết bị giống như kính đeo mắt. Bao gồm các phiên bản:
 - Kính mắt sử dụng máy ảnh để chặn thế giới quan thực và hiển thị lại chế độ xem tăng cường của nó thông qua thị kính
 - Các thiết bị mà trong đó hình ảnh AR được chiếu qua hoặc phản chiếu ra khỏi bề mặt của các mảnh thấu kính của kính mắt.
- Màn hình ngẩng đầu (HUD): là một màn hình trong suốt hiển thị dữ liệu mà không yêu cầu người dùng nhìn ra khỏi điểm nhìn thông thường của họ. Là công nghệ tiền thân của thực tại tăng cường, HUD được phát triển lần đầu tiên cho phi công vào những năm 1950, chiếu dữ liệu chuyến bay đơn giản vào tầm nhìn của họ, nhờ đó giúp họ luôn "ngẩng cao đầu" và không nhìn xuống thiết bị. Thiết bị thực tại tăng cường dành cho mắt cận có thể được sử dụng làm màn hình hiển thị di động vì chúng có thể hiển thị dữ liệu, thông tin và hình ảnh trong khi người dùng xem thế giới thực.
- Kính áp tròng: kính áp tròng hiển thị hình ảnh AR đang được phát triển. Những kính áp tròng bionic này có thể chứa các yếu tố để hiển thị được nhúng vào ống kính bao gồm mạch tích hợp, đèn LED và ăng-ten để giao tiếp không dây. Màn hình kính áp tròng đầu tiên được cấp bằng sáng chế vào năm 1999 bởi Steve Mann và dự định sẽ hoạt động kết hợp với kính AR, nhưng dự án đã bị bỏ dở, sau đó 11 năm vào năm 2010–2011 một phiên bản khác của kính áp tròng, đang được phát triển cho quân đội Hoa Kỳ, được thiết kế để hoạt động với kính AR, cho phép binh lính tập trung vào các hình ảnh AR ở gần mắt trên kính và các vật thể ở xa trong thế giới thực cùng một lúc.
- Màn hình võng mạc ảo (VRD): một thiết bị hiển thị cá nhân đang được phát triển tại Phòng thí nghiệm Công nghệ Giao diện Con người của Đại học Washington dưới thời Tiến sĩ Thomas A. Furness III. Với công nghệ này, màn hình được quét trực tiếp lên võng mạc của mắt người xem. Điều này dẫn đến hình ảnh tươi sáng với độ phân giải cao và độ tương phản cao. Người xem nhìn thấy những gì dường như là một màn hình thông thường lơ lửng trong không gian.

Các thành phần cảm biến: hệ thống thực tại tăng cường di động hiện đại sử dụng một hoặc nhiều công nghệ theo dõi chuyển động như: máy ảnh kỹ thuật số và/hoặc cảm biến quang học, gia tốc kế, GPS, con quay hồi chuyển, la bàn trạng thái rắn, nhận dạng tần số vô tuyến (RFID). Những công nghệ này cung cấp các mức độ chính xác và độ chính xác khác nhau. Tuy nhiên quan trọng nhất vẫn là vị trí và hướng của đầu người dùng. Theo dõi (các) tay của người dùng hoặc thiết bị cầm tay có thể cung cấp kỹ thuật tương tác 6DOF.

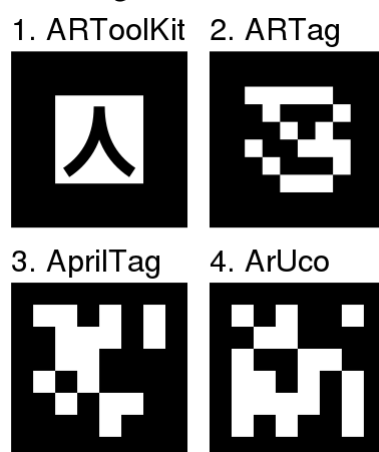
Các thành phần thiết bị đầu vào: các kỹ thuật như hệ thống nhận dạng giọng nói chuyển lời nói của người dùng thành các lệnh điều khiển trên máy tính và hệ thống nhận dạng cử chỉ diễn giải chuyển động cơ thể của người dùng bằng cách phát hiện trực quan hoặc từ các cảm biến được nhúng trong một thiết bị ngoại vi như đĩa phép, bút stylus, con trỏ, găng tay hoặc các đồ dùng cơ thể khác.

2.3.3 Các công nghệ phần mềm

Một thước đo quan trọng của các hệ thống AR là mức độ thực tế chúng tích hợp các tính năng tăng cường với thế giới thực. Phần mềm phải lấy tọa độ thế giới thực, không phụ thuộc vào máy ảnh và hình ảnh máy ảnh. Quá trình đó được gọi là đăng ký hình ảnh và sử dụng các phương pháp thị giác máy tính khác nhau, chủ yếu liên quan đến theo dõi video. Nhiều phương pháp thị giác máy tính của thực tại tăng cường được kế thừa từ phép đo thị giác.

Thông thường những phương pháp đó gồm hai phần. Giai đoạn đầu là phát hiện điểm quan trọng, điểm đánh dấu fiducial hoặc luồng quang học trong hình ảnh camera. Bước này có thể sử dụng các phương pháp phát hiện nổi bật như phát hiện góc, phát hiện đốm màu, phát hiện biên hoặc ngưỡng và các phương pháp xử lý hình ảnh khác. Giai đoạn thứ hai là khôi phục lại hệ tọa độ trong thế giới thực từ những dữ liệu thu được trong giai đoạn đầu tiên. Một số phương pháp sẽ giả định rằng các đối tượng có dạng hình học đã biết (hoặc các điểm đánh dấu fiducial) có hiện diện trong cảnh. Trong một số trường hợp, cấu trúc cảnh 3D nên được tính toán trước. Nếu một phần của cảnh là không xác định thì định vị hóa và ánh xạ đồng thời (SLAM) có thể lập bản đồ các vị trí tương đối. Nếu không có thông tin về bối cảnh, cấu trúc từ các phương pháp chuyển động như điều chỉnh bó (Bundle adjustment) sẽ được sử dụng. Các phương pháp toán học được sử dụng trong giai đoạn thứ hai bao gồm: hình học chiếu ảnh (epipolar), đại số hình học, biểu diễn phép quay với ánh xạ hàm mũ, bộ lọc kalman và hạt, tối ưu hóa phi tuyến, thống kê vững.

Trong thực tại tăng cường, sự phân biệt được thực hiện giữa hai chế độ theo dõi riêng biệt, được gọi là điểm đánh dấu và không có điểm đánh dấu. Điểm đánh dấu là các dấu hiệu trực quan kích hoạt hiển thị thông tin ảo. Có thể sử dụng một tờ giấy với một số hình học riêng biệt. Máy ảnh nhận dạng các hình học bằng cách xác định các điểm cụ thể trong hình vẽ.



Hình 2.14 Một số điểm đánh dấu fiducial cho thị giác máy tính



Hình 2.15 Con thuyền xuất hiện do có điểm đánh dấu là bìa tờ tạp chí

Theo dõi không đánh dấu, còn được gọi là theo dõi tức thì, không sử dụng điểm đánh dấu. Thay vào đó, người dùng đặt đối tượng trong chế độ xem camera tốt nhất là trên một mặt phẳng nằm ngang. Nó sử dụng các cảm biến trong thiết bị di động để phát hiện chính xác môi trường trong thế giới thực, chẳng hạn như vị trí của các bức tường và các điểm giao nhau.

Ngôn ngữ đánh dấu thực tế tăng cường (ARML) là một tiêu chuẩn dữ liệu được phát triển trong Hiệp hội không gian địa lý mở (OGC), bao gồm cú pháp từ ngôn ngữ đánh dấu có thể mở rộng (XML) để mô tả vị trí và sự xuất hiện của các đối tượng ảo trong cảnh, cũng như các ràng buộc ECMAScript để cho phép truy cập động vào các thuộc tính của các đối tượng ảo.

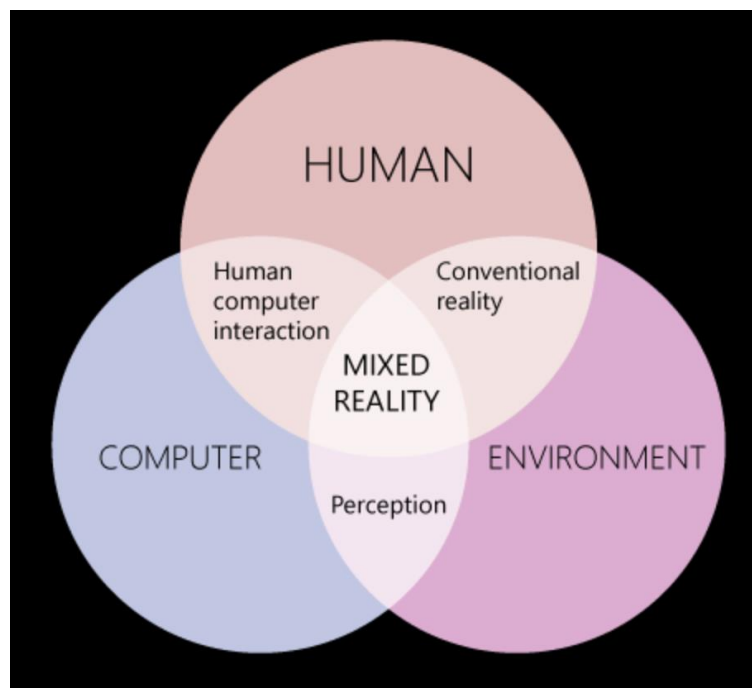
Để cho phép phát triển nhanh chóng các ứng dụng thực tế tăng cường, một số bộ công cụ phát triển phần mềm (SDK) đã ra đời.



Hình 2.16 Các AR SDK nổi tiếng

2.4 Thực tại hỗn hợp

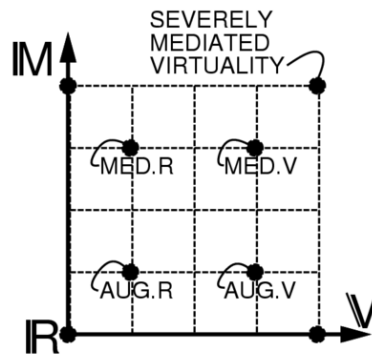
Thực tại hỗn hợp (MR) là sự hợp nhất của thế giới thực và ảo để tạo ra môi trường và hình dung mới, nơi các đối tượng vật lý và kỹ thuật số cùng tồn tại và tương tác trong thời gian thực. Thực tại hỗn hợp không chỉ diễn ra trong thế giới vật chất hay thế giới ảo, mà là sự kết hợp giữa thực tại và thực tại ảo. Có rất nhiều ứng dụng thực tế của thực tại hỗn hợp, bao gồm giải trí, huấn luyện quân sự và làm việc từ xa. Ngoài ra còn có các công nghệ hiển thị khác nhau được sử dụng để tạo điều kiện tương tác giữa người dùng và các ứng dụng thực tại hỗn hợp.



Hình 2.17 Sự tương tác giữa con người, máy tính và môi trường

2.4.1 Tính liên tục ảo/trung (Virtuality/Mediality continuum)

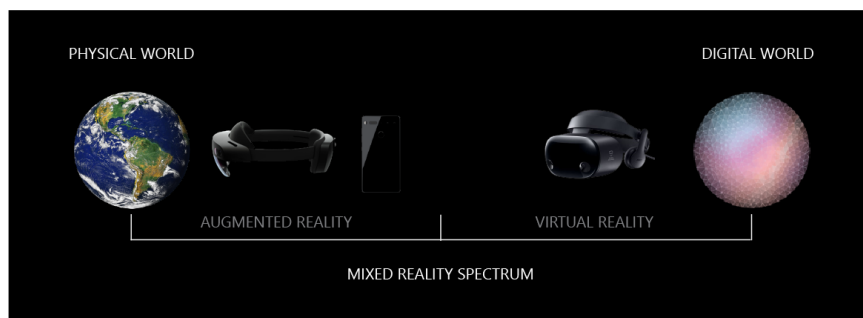
Thực tế hỗn hợp lần đầu tiên được Paul Milgram và Fumio Kishino định nghĩa vào năm 1994 là "bất cứ nơi nào giữa cực điểm của sự ảo hóa liên tục", nơi mà tính ảo hóa liên tục kéo dài từ hoàn toàn thực đến môi trường hoàn toàn ảo, với thực tại tăng cường và ảo hóa tăng cường nằm ở giữa. Tính liên tục của độ trung gian thực ảo có thể được thực hiện trong một mũ trùm hoặc kính đeo mắt có thể chặn các thông tin thừa hoặc thay thế chúng trong thế giới thực bằng thông tin hữu ích hơn. Khoảng không gian thực tế trung gian này là cơ sở để mô tả cách các đối tượng trong cả thế giới vật lý và ảo đang tương tác với nhau. Thay vì chỉ dựa vào thực tế và ảo như hai thực thể hoàn toàn riêng biệt, người ta đã chấp nhận rằng có sự liên tục giữa hai khái niệm này và các ứng dụng của thực tại hỗn hợp có thể nằm ở bất cứ đâu giữa hai khái niệm này. Trong bài báo lần đầu tiên giới thiệu thuật ngữ thực tại hỗn hợp, Milgram và Kishino lập luận rằng một thuật ngữ như vậy là cần thiết để đề cập đến "một lớp con cụ thể của các công nghệ liên quan đến VR mà sẽ diễn ra việc hợp nhất thế giới thực và ảo", một thông số mà trước đây không được đưa vào.



Hình 2.18 Mặt phẳng thực tế trung gian liên tục bao gồm 4 điểm

2.4.2 Sự khác biệt so với các công nghệ khác

Thực tại hỗn hợp đề cập đến mọi thứ trong chuỗi liên tục thực tế - ảo ngoại trừ các ứng dụng ở hai thái cực. Điều này bao gồm thực tại ảo (VR), thực tại tăng cường (AR) và ảo tăng cường (AV). Ở một đầu của dải là thế giới thực không có lớp phủ công nghệ kỹ thuật số. Ở đầu kia của dải là thực tế ảo, đề cập đến "một môi trường nhân tạo được trải nghiệm thông qua các kích thích cảm giác (chẳng hạn như cảnh quan và âm thanh) do máy tính cung cấp và trong đó hành động của một người xác định một phần những gì xảy ra trong môi trường." Thực tại tăng cường nằm giữa hai điểm đó và đề cập đến "một phiên bản nâng cao của thực tế được tạo ra bằng cách sử dụng công nghệ để phủ thông tin kỹ thuật số lên hình ảnh của một thứ đang được xem qua một thiết bị." Thực tại hỗn hợp có sự độc nhất ở chỗ nó thường đề cập đến các sản phẩm nhân tạo tương tác với người dùng trong thế giới thực. Ảo giác tăng cường (AV) là một danh mục con của thực tại hỗn hợp đề cập đến việc hợp nhất các đối tượng trong thế giới thực vào thế giới ảo.



Hình 2.19 Dải phổ thực tại hỗn hợp

Là một trường hợp trung gian trong dải liên tục ảo, nó được đề cập chủ yếu đến là không gian ảo, nơi các yếu tố vật lý (chẳng hạn như vật thể hoặc con người) được tích hợp vào và có thể tương tác với thế giới ảo trong thời gian thực. Sự tích hợp này đạt được với việc sử dụng các kỹ thuật khác nhau, chẳng hạn như phát trực tuyến video từ không gian vật lý, như qua webcam, hoặc sử dụng kỹ thuật số hóa 3D của các đối tượng vật lý. Việc sử dụng thông tin cảm biến trong thế giới thực, chẳng hạn như con quay hồi chuyển, để điều khiển môi trường ảo là một dạng bổ sung của ảo tăng cường, trong đó các đầu vào bên ngoài cung cấp ngữ cảnh cho chế độ xem ảo.

2.4.3 Vật lý liên thực tại (Interreality physics)

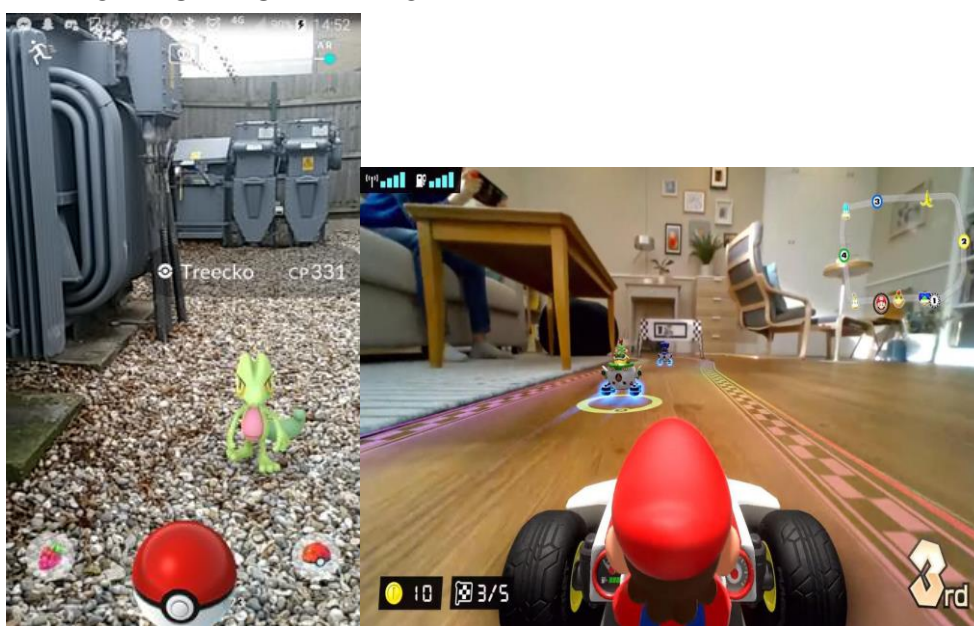
Trong một ngữ cảnh vật lý, thuật ngữ "hệ thống liên thực tại" dùng để chỉ một hệ thống thực tại ảo kết hợp với hệ thống thực tế của nó. Nó có thể được mô tả như một hệ thống bao gồm một con lắc vật lý thực kết hợp với một con lắc chỉ tồn tại trong thực tế ảo. Hệ thống này có hai trạng thái chuyển động ổn định: trạng thái "thực tại kép" trong đó chuyển động của hai mặt con lắc không tương quan và trạng thái "thực tại hỗn hợp" trong đó hai con lắc thể hiện chuyển động có khóa pha ổn định, có tương quan cao. Việc sử dụng các thuật ngữ "thực tại hỗn hợp" và "thực tại" được xác định rõ ràng trong ngữ cảnh vật lý và có thể hơi khác trong các lĩnh vực khác, tuy nhiên, nó thường được coi là, "cầu nối giữa thế giới vật lý và ảo".

2.4.4 Ứng dụng của MR

Một giao diện quản lý nội dung sản phẩm tương tác được (IPCM): việc quản lý nội dung sản phẩm trước khi thực tại hỗn hợp ra đời chủ yếu bao gồm các tài liệu và ít tương tác giữa khách hàng và sản phẩm. Với những cải tiến công nghệ thực tại hỗn hợp, các hình thức quản lý nội dung sản phẩm có thể tương tác được đã xuất hiện. Đáng chú ý nhất, kỹ thuật dựng hình ảnh 3 chiều của các sản phẩm mà thông thường là 2 chiều đã tăng khả năng tiếp cận và hiệu quả của tương tác giữa người tiêu dùng và sản phẩm.

Giáo dục: các phương pháp học tập dựa trên mô phỏng, như đào tạo dựa trên VR và AR đã làm tăng sự tương tác, trải nghiệm. Có rất nhiều tiềm năng trong việc sử dụng thực tại hỗn hợp trong cả môi trường giáo dục và cơ sở đào tạo chuyên nghiệp. Đáng chú ý là trong giáo dục, AR đã được sử dụng để mô phỏng các trận chiến lịch sử, mang lại trải nghiệm nhập vai cao cho học sinh và nâng cao trải nghiệm học tập.

Giải trí: từ các chương trình truyền hình đến máy chơi game, thực tại hỗn hợp có rất nhiều ứng dụng trong lĩnh vực giải trí.



Hình 2.20 Các game ứng dụng MR

CHƯƠNG 3. TỔNG QUAN VỀ AR FOUNDATION VÀ UNITY

3.1 Giới thiệu về Unity



Hình 3.1 Biểu tượng của unity

Unity là một công cụ để tạo trò chơi trên nhiều nền tảng. Unity được Unity Technologies phát hành vào năm 2005. Trọng tâm của Unity nằm ở việc phát triển cả trò chơi 3D và 2D và nội dung tương tác. Unity hiện hỗ trợ 27 nền tảng mục tiêu khác nhau để triển khai. Các nền tảng phổ biến nhất là hệ thống Android, PC và iOS.

Unity là một nền tảng tích hợp được sử dụng như một công cụ và framework game.

Unity cho phép phát triển một lần và phát hành ở mọi nơi.

Unity thiết kế một hệ thống module để xây dựng các **scene** và **game object**

Mặc dù Unity được coi là thích hợp để tạo trò chơi 3D hơn, nó cũng có thể được sử dụng tương tự để phát triển trò chơi 2D.

Trong Unity, có thể phát triển các trò chơi có nội dung nặng mà không phụ thuộc vào các framework hoặc công cụ bổ sung. Giúp nâng cao trải nghiệm của người dùng.

Với sự trợ giúp của Unity, các nhà phát triển trò chơi có thể truy cập vô số tài nguyên như các công cụ trực quan, nội dung có sẵn, tài liệu rõ ràng, cộng đồng trực tuyến,... một cách miễn phí để tạo nội dung 3D trong trò chơi.

Theo dõi tài nguyên và kết xuất đồ họa, lập trình là một số tính năng phát triển trò chơi của Unity dùng để giảm thời gian và chi phí.

3.1.1 Scene View

Scene View là nơi lập trình viên xây dựng trò chơi của mình. Đó là nơi bạn thêm tất cả các mô hình, máy ảnh và các phần khác tạo nên trò chơi của bạn. Đây là cửa sổ 3D nơi bạn có thể đặt trực quan tất cả các nội dung bạn đang sử dụng.



Hình 3.2 Cảnh sắp xếp model nhà trong scene view

Khi người dùng kiểm tra trò chơi của mình trong Unity, Scene View sẽ tự cập nhật trạng thái của trò chơi hiện tại. Người dùng có thể thêm các yếu tố mới vào scene.

Khi người dùng dừng trò chơi, scene sẽ trở lại trạng thái ban đầu. Mọi thay đổi bạn thực hiện đối với Scene View trong khi chơi trò chơi sẽ bị mất khi nó dừng lại.

3.1.2 Game View

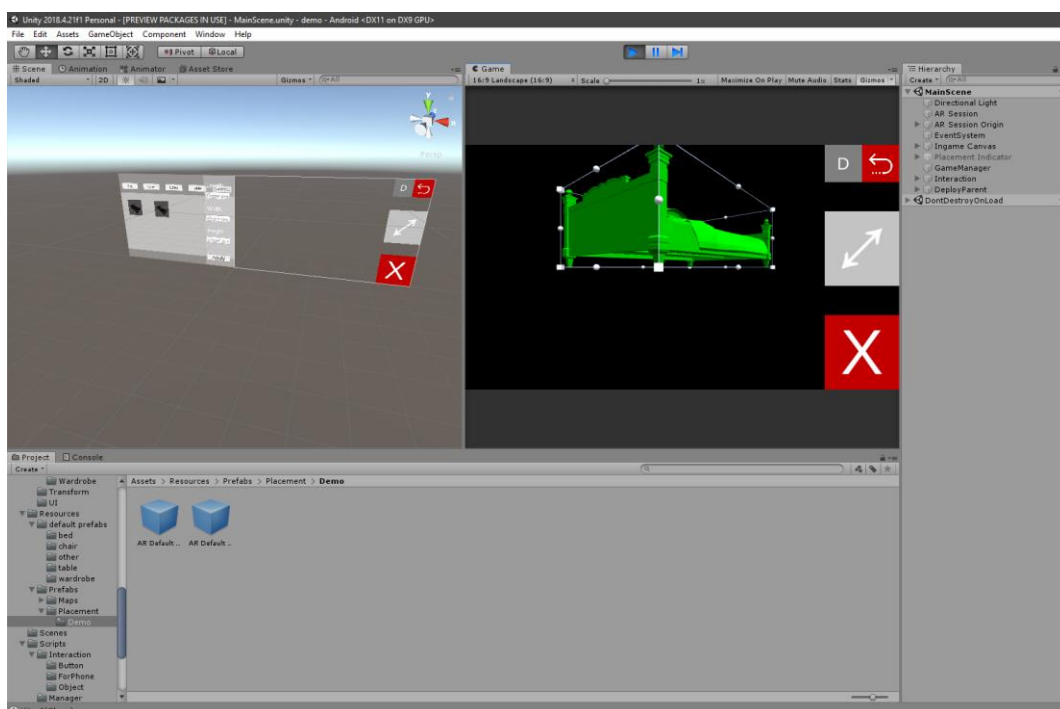
Game View thể hiện góc nhìn của người chơi về trò chơi. Đây là nơi người dùng có thể chơi trò chơi của mình và xem tất cả các game object hoạt động với nhau như thế nào. Chế độ xem trò chơi có bộ chọn tỷ lệ khung hình, cho phép bạn thay đổi kích thước của chế độ xem để khớp với kích thước của một tỷ lệ khung hình màn hình cụ thể (ví dụ: 4:3, 16:9, 16:10...) hoặc kích thước màn hình thiết bị chẳng hạn như của iPhone hoặc iPad. Điều này cho phép lập trình viên đảm bảo trò chơi trông hợp lý trên tất cả các tỷ lệ khung hình và độ phân giải hỗ trợ.

3.1.3 Cửa sổ phân cấp

Cửa sổ phân cấp chứa danh sách tất cả các game object hiện tại được sử dụng trong trò chơi, game object là một đối tượng trong trò chơi.

Về bản chất, game object là các vùng chứa trống mà lập trình viên có thể tùy chỉnh bằng cách thêm các thành phần. Các thành phần cho phép game object chiếu hình học (từ một khối đơn giản đến các mô hình 3D phức tạp hơn về thấp hoặc quái vật), phát ra ánh sáng, hoạt động như một máy ảnh hoặc thậm chí tạo ra hành vi phức tạp thông qua các tập lệnh.

Game object cũng có thể hoạt động giống như các thư mục, chứa các game object khác, điều này làm cho chúng khá hữu ích để tổ chức scene.



Hình 3.3 Giao diện đầy đủ của Unity Editor

3.1.4 Component

Do bản chất của unity là phân thành các module nên mỗi một game object lại gồm nhiều phần nhỏ hơn hợp thành gọi là các component, các component có các chức năng riêng biệt như <Image> dùng để hiển thị hình ảnh hay <Collider> có chức năng tạo một hình bao để dùng để xác định va chạm giữa các vật thể trong game, hay <Script> có chức năng như một class con của game object chứa các biến private public như một class bình thường và có thể được truy cập bởi những component khác.

3.1.5 Prefab

Hệ thống Prefab của Unity cho phép người lập trình tạo, định cấu hình và lưu trữ một game object hoàn chỉnh với tất cả các thành phần, giá trị thuộc tính và các game object con của nó như một tài nguyên có thể tái sử dụng. Prefab hoạt động như một khuôn mẫu mà từ đó bạn có thể tạo các phiên bản Prefab mới trong scene.

Hệ thống Prefab cho phép tự động giữ đồng bộ tất cả các bản sao. Bất kỳ chỉnh sửa nào được thực hiện đối với nội dung prefab đều được tự động thực hiện đối với các bản sao của prefab đó, cho phép dễ dàng thực hiện các thay đổi rộng rãi trên toàn bộ scene mà không cần phải thực hiện lại cùng một chỉnh sửa đối với mọi bản sao đó.

3.1.6 Assets

Assets là những tài nguyên được sử dụng trong việc phát triển game trong Unity. Các Assets này có thể là hình ảnh, mô hình 3D, âm thanh, hiệu ứng, ... được tạo ra bởi các nhà phát triển, có thể được download miễn phí hoặc trả phí. Tính năng này giúp giảm thiểu rất nhiều thời gian cho việc thiết kế và lập trình game.

3.1.7 Luồng chạy của component

Khi lần đầu component được khởi tạo nó sẽ được gọi vào hàm Awake()

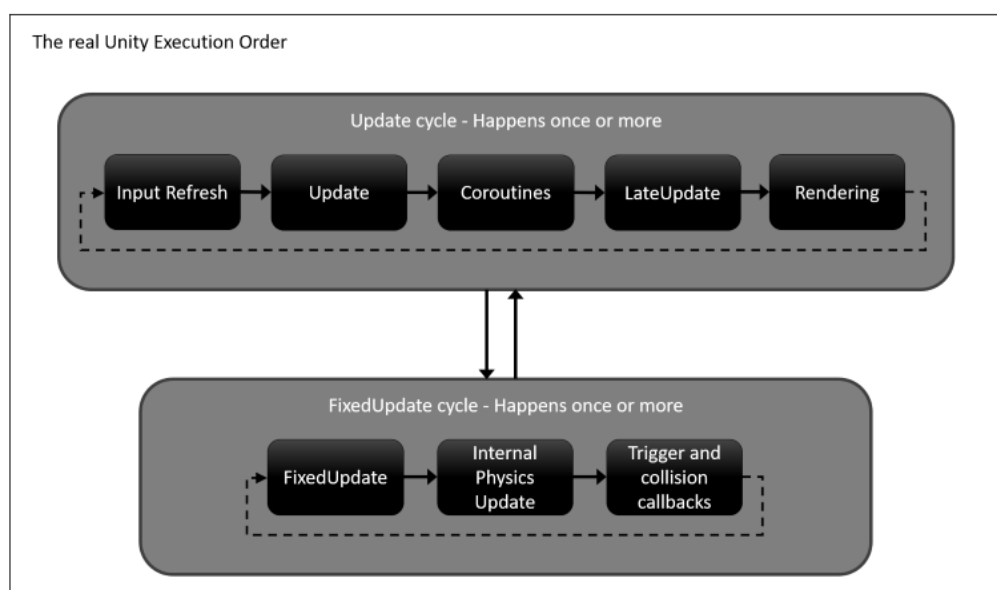
Tiếp theo nếu được bật thì component sẽ chạy hàm OnEnable() rồi đến hàm Start()

Sau khi hoàn thành các bước trên unity sẽ bước vào vòng lặp:

- Update(): chạy sau mỗi một khung hình
- FixedUpdate(): chạy sau mỗi một khoảng thời gian

Nếu component bị phá hủy thì sẽ chạy hàm OnDestroy()

Nếu component bị tắt thì sẽ chạy hàm OnDisable()



Hình 3.4 Thứ tự chạy các hàm update trong Unity

3.2 Giới thiệu về AR Foundation

3.2.1 Giới thiệu về ARCore

ARCore là nền tảng của Google để xây dựng trải nghiệm thực tại tăng cường. Sử dụng các API khác nhau, ARCore cho phép điện thoại của bạn cảm nhận môi trường xung quanh nó, hiểu biết đến thế giới và tương tác với thông tin. Một số API có sẵn trên Android và iOS để cho phép chia sẻ trải nghiệm AR.

ARCore sử dụng ba khả năng chính để tích hợp nội dung ảo với thế giới thực khi được nhìn thấy qua camera trên điện thoại của bạn:

- Theo dõi chuyển động (motion tracking) cho phép điện thoại hiểu và theo dõi vị trí của nó so với thế giới.
- Hiểu biết về môi trường (environmental understanding) cho phép điện thoại phát hiện kích thước và vị trí của tất cả các loại bề mặt: bề mặt ngang, dọc và góc cạnh như mặt đất, bàn cà phê hoặc tường.
- Ước lượng ánh sáng (light estimation) cho phép điện thoại ước tính điều kiện ánh sáng hiện tại của môi trường.

Về cơ bản, ARCore đang làm hai việc: theo dõi vị trí của thiết bị di động khi nó di chuyển và xây dựng hiểu biết của riêng nó về thế giới thực.

Công nghệ theo dõi chuyển động của ARCore sử dụng camera của điện thoại để xác định các điểm thú vị (interesting point), được gọi là các đặc trưng và theo dõi cách các điểm đó di chuyển theo thời gian. Với sự kết hợp giữa chuyển động của những điểm này và số đọc từ cảm biến quán tính của điện thoại, ARCore xác định cả vị trí và hướng của điện thoại khi nó di chuyển trong không gian.

Ngoài việc xác định các điểm trên, ARCore có thể phát hiện các bề mặt phẳng, như bàn hoặc sàn nhà và cũng có thể ước tính ánh sáng trung bình ở khu vực xung quanh nó. Những khả năng này kết hợp với nhau để cho phép ARCore xây dựng sự hiểu biết của riêng mình về thế giới xung quanh.

Sự hiểu biết của ARCore về thế giới thực cho phép bạn đặt các đối tượng, chú thích hoặc thông tin khác theo cách tích hợp hoàn toàn với thế giới thực.

Khi điện thoại của bạn di chuyển khắp thế giới, ARCore sử dụng một quy trình được gọi là bản địa hóa và ánh xạ đồng thời, hoặc SLAM, để hiểu vị trí của điện thoại so với thế giới xung quanh. ARCore phát hiện các đặc điểm khác biệt trực quan trong hình ảnh camera đã chụp được gọi là điểm đặc trưng và sử dụng các điểm này để tính toán sự thay đổi về vị trí của nó. Thông tin hình ảnh được kết hợp với các phép đo quán tính từ IMU của thiết bị để ước tính **pose** (tư thế - vị trí và hướng) của máy ảnh so với thế giới theo thời gian. Bằng cách căn chỉnh tư thế của máy ảnh ảo hiển thị nội dung 3D của bạn với tư thế của máy ảnh của thiết bị do ARCore cung cấp, các nhà phát triển có thể hiển thị nội dung ảo từ góc nhìn chính xác. Hình ảnh ảo được hiển thị có thể được phủ lên trên hình ảnh thu được từ máy ảnh của thiết bị, làm cho nó xuất hiện như thể nội dung ảo là một phần của thế giới thực.

ARCore sau đó sẽ tìm kiếm các cụm điểm đặc trưng dường như nằm trên các bề mặt ngang hoặc dọc phổ biến, như bảng hoặc tường và cung cấp các bề mặt này cho ứng dụng của bạn dưới dạng **plane** (mặt phẳng). ARCore cũng có thể xác định ranh giới của từng mặt phẳng và cung cấp thông tin đó cho ứng dụng của bạn. Bạn có thể sử dụng thông tin này để đặt các đối tượng ảo nằm yên trên bề mặt phẳng.



Hình 3.5 Một plane (màu vàng) và các điểm đặc trưng (màu đỏ)

ARCore sử dụng **hit-testing** (thử nghiệm va chạm) để lấy tọa độ (x, y) tương ứng với màn hình của điện thoại (được cung cấp bằng cách nhấn hoặc bất kỳ tương tác nào khác mà bạn muốn ứng dụng của mình hỗ trợ) và chiếu tia từ điểm nhìn của máy ảnh, trả lại bất kỳ plane hoặc điểm đặc trưng mà tia này đâm vào, cùng với vị trí của giao điểm đó trong không gian thế giới. Điều này cho phép người dùng chọn hoặc tương tác với các đối tượng trong môi trường.

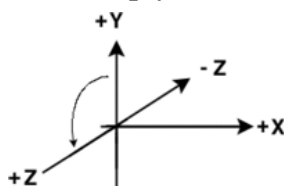
Các pose có thể thay đổi khi ARCore nâng cao hiểu biết về vị trí của chính mình và môi trường của nó. Khi bạn muốn đặt một đối tượng ảo, bạn cần xác định một **anchor** (mỏ neo) để đảm bảo rằng ARCore theo dõi vị trí của đối tượng theo thời gian. Thông thường, bạn tạo một anchor dựa trên pose được trả lại bởi một bài hit-testing, như được mô tả trong tương tác với người dùng.

Thực tế là các pose có thể thay đổi có nghĩa là ARCore có thể cập nhật vị trí của các đối tượng môi trường như mặt phẳng và các điểm đặc trưng theo thời gian. Plane và điểm là một loại đối tượng đặc biệt được gọi là **trackable** (đối tượng có thể theo dõi). Giống như tên gọi, đây là những đối tượng mà ARCore sẽ theo dõi theo thời gian. Người dùng có thể neo các đối tượng ảo vào các trackable cụ thể để đảm bảo rằng mối quan hệ giữa đối tượng ảo và trackable vẫn ổn định ngay cả khi thiết bị di chuyển xung quanh.

3.2.2 Giới thiệu về ARKit

Được giới thiệu tại sự kiện WWDC tháng 06/2017, ARKit cho phép bạn tích hợp và sử dụng camera các thiết bị iOS và các cảm biến chuyển động để tạo ra trải nghiệm tăng cường thực tế trong các game và ứng dụng, bản chất là thêm các yếu tố hình ảnh 2D hoặc 3D vào live view của camera thiết bị, làm cho các yếu tố đó xuất hiện trong thế giới thực và là một phần trong đó. ARKit kết hợp việc theo dõi chuyển động của thiết bị, kết hợp với chụp ảnh camera, xử lý hình ảnh thu được và công nghệ hiển thị nhằm đơn giản hoá việc xây dựng một trải nghiệm AR.

Đặc điểm cơ bản của AR là khả năng tạo ra và theo dõi chuyển động, phản hồi giữa không gian ảo với các object ảo được đặt lên màn hình live view điện thoại khi người di chuyển thiết bị trong thế giới thực. Khi đó, hình ảnh object ảo được hiển thị trên màn hình điện thoại trên nền hình ảnh của thế giới thực thu được từ camera chân thực, sống động như thật. Người dùng sẽ có cảm giác thật như chính vật đó đang tồn tại giữa đời thực. ARKit sử dụng hệ trục tọa độ không gian 3 chiều Oxyz theo quy tắc bàn tay phải: trục Oy (+) hướng lên trên theo ngón giữa, trục Ox (+) hướng sang phải theo ngón trỏ và trục Oz (+) vuông góc với lòng bàn tay, hướng về mặt người nhìn. Một AR session có nhiều anchor. Mỗi anchor cũng có hệ trục tọa độ riêng, tuân theo quy tắc bàn tay phải.



Hình 3.6 Quy tắc bàn tay phải cho hệ trục Ox (+)

Mỗi khi sử dụng ARKit, ta cần tạo ra một session để quản lý việc tracking, xử lý, hiển thị... Chúng ta cũng có thể cấu hình cho session, thay đổi tọa độ gốc và hướng của hệ tọa độ phản ánh thế giới thực qua property `worldAlignment`

Để tạo ra sự hài hoà, tương xứng giữa không gian thực và không gian ảo, ARKit sử dụng một kỹ thuật được gọi là "visual-inertial odometry". Kỹ thuật này kết hợp thông tin dữ liệu thu được từ các cảm biến chuyển động thiết bị iOS với khả năng phân tích, xử lý hình ảnh cảnh quan từ camera. ARKit nhận biết các điểm đặc biệt trong các bức ảnh không gian thực, đồng thời theo dõi sự thay đổi về vị trí của các điểm này qua các frame theo thời gian, sau đó so sánh với thông tin dữ liệu cảm biến chuyển động. Kết quả là một mô hình có độ chính xác cao của vị trí và chuyển động của thiết bị với cảnh quan thực xung quanh. World tracking cũng phân tích và tìm hiểu nội dung của cảnh đó. Sử dụng các method hit-testing để tìm kiếm những bề mặt trong không gian thực tương ứng với một điểm trên ảnh từ camera. Nếu bạn enable property `planeDetection` trong session configuration, ARKit sẽ tự động phát hiện những mặt phẳng trong hình ảnh camera thu được cùng với vị trí và kích thước của chúng. Sau đó, bạn có thể đặt các đối tượng ảo vào các mặt phẳng tìm được trong thế giới thực và có thể tương tác với chúng, như xoay, di chuyển các đối tượng này.

3.2.3 Giới thiệu về AR Foundation

3.2.3.1. Tổng quan về AR Foundation

AR Foundation cho phép bạn làm việc với các nền tảng thực tại tăng cường theo cách đa nền tảng trong Unity. Gói này cung cấp giao diện cho các nhà phát triển Unity sử dụng, nhưng không tự triển khai bất kỳ tính năng AR nào. Để sử dụng AR Foundation trên thiết bị mục tiêu, bạn cũng cần các gói riêng biệt cho các nền tảng mục tiêu được Unity chính thức hỗ trợ:

- ARCore XR Plugin cho Android
- ARKit XR Plugin cho iOS
- Magic Leap XR Plugin cho Magic Leap
- Windows XR Plugin cho HoloLens

	ARCore	ARKit	Magic Leap	HoloLens
Device tracking	✓	✓	✓	✓
Plane tracking	✓	✓	✓	
Point clouds	✓	✓		
Anchors	✓	✓	✓	✓
Light estimation	✓	✓		
Environment probes	✓	✓		
Face tracking	✓	✓		
2D Image tracking	✓	✓	✓	
3D Object tracking		✓		
Meshing		✓	✓	✓
2D & 3D body tracking		✓		
Collaborative participants		✓		
Human segmentation		✓		
Raycast	✓	✓	✓	
Pass-through video	✓	✓		
Session management	✓	✓	✓	✓
Occlusion	✓	✓		

Bảng 3-1 Các tính năng được hỗ trợ trên từng nền tảng

AR Foundation được xây dựng trên các **subsystem** (hệ thống con). Subsystem là một interface (interface) không phụ thuộc vào nền tảng để hiển thị các loại thông tin khác nhau. Các hệ thống con liên quan đến AR được định nghĩa trong gói AR Subsystems và sử dụng namespace `UnityEngine.XR.ARSubsystems`. Đôi khi sẽ cần phải tương tác với các loại trong gói subsystem AR. Mỗi subsystem xử lý chức năng cụ thể. Ví dụ: `XRPlaneSubsystem` cung cấp interface phát hiện mặt phẳng

3.2.3.2. Vấn đề về việc nhận diện mặt phẳng dọc

ARKit và ARCore có thể phân tích môi trường có thể nhìn thấy trong chế độ camera và phát hiện vị trí của các mặt phẳng nằm ngang như bàn, sàn nhà hoặc mặt đất. Tuy nhiên, các framework của ARKit và ARCore không thể phát hiện trực tiếp các mặt phẳng thẳng đứng như tường.

Lý do tại sao việc phát hiện mặt phẳng thẳng đứng bị hạn chế là do điện thoại thông minh hiện tại không có các cảm biến bổ sung cần thiết để nhận diện độ sâu một cách chính xác. Nhiều khả năng, tính năng phát hiện mặt phẳng dọc sẽ được triển khai trong các thế hệ thiết bị di động tiếp theo. Tuy nhiên, cho đến khi điều này xảy ra, khả năng AR sẽ vẫn bị hạn chế phần nào do thiếu chức năng này.

Tuy nhiên, có một số giải pháp cho vấn đề này và nhiều cách tiếp cận đã được sử dụng để giải quyết nó, mỗi cách đều có những ưu và nhược điểm riêng:

- Sử dụng đám mây điểm neo: Nếu một bức tường có các màu sắc hoặc hoa văn tương phản hoặc đã treo các đồ vật có độ tương phản cao trên tường, thì những vật này cung cấp các tính năng tốt có thể được quét. Bất cứ khi nào các điểm đặc trưng được phát hiện (hiển thị bằng màu vàng trong ảnh chụp màn hình ở trên), AR Foundation có thể đặt neo trong cảnh thế giới thực. Điều này cho phép nắm lấy vị trí của một điểm hoặc thậm chí toàn bộ bề mặt bằng cách sử dụng đám mây điểm. Ví dụ trên cho thấy rằng nhận dạng tường hoạt động tốt trong các tình huống có bề mặt tường tương phản mạnh và do đó không cần thêm khả năng phát hiện mặt phẳng thẳng đứng.
- Tìm kiếm mặt phẳng khung hình theo điểm: Cách tiếp cận này dựa trên việc tìm kiếm đường viền của một mặt phẳng đã được phát hiện, giả sử rằng mặt phẳng này là bức tường thực. Bất kỳ điểm 2D nào trong không gian tọa độ của cảnh trước mặt camera đều có thể tham chiếu đến các điểm dọc theo một đoạn thẳng trong không gian tọa độ 3D. Phương pháp này giả định rằng có thể tìm thấy một mặt phẳng của khung hình hiện tại trên thế giới thực nằm tại một điểm giao với khung này. Quá trình này bao gồm việc đi qua từng điểm ảnh dọc theo trục Y theo pixel để tìm điểm giao với mặt phẳng được phát hiện. Phương pháp này gặp sự cố khi không có mặt phẳng nhìn thấy trong khung hình hiện tại (Ví dụ khi đứng quá gần).
- Tìm kiếm một mặt phẳng bằng cách sử dụng ray-casting: ray-casting là việc sử dụng một tia phóng từ một điểm để xác định vật thể đầu tiên tia đó chạm đến. Phương pháp này liên quan đến việc ray-casting dọc theo trục Y cho đến khi tìm thấy một lần bắn trúng hoặc tới khi tia vuông góc với mặt đất, nghĩa là không có mặt phẳng nào ở phía trước. Cách tiếp cận này yêu cầu phần mềm thực hiện nhiều phép tính khác nhau để có được tọa độ của các tia trong không gian 3D.

3.3 Giới thiệu về Mixed Reality Toolkit

MRTK là một bộ công cụ mã nguồn mở đã có từ khi Hololens được phát hành lần đầu tiên. MRTK-Unity là một bộ công cụ phát triển đa nền tảng, mã nguồn mở cho các ứng dụng thực tế hỗn hợp. Bộ công cụ cung cấp hệ thống đầu vào đa nền tảng, các thành phần nền tảng và các khối xây dựng chung cho các tương tác không gian. MRTK phiên bản 2 dự định tăng tốc độ phát triển ứng dụng cho Microsoft Hololens, tai nghe nhập vai Thực tế hỗn hợp (VR) Windows và nền tảng OpenVR. Dự án nhằm mục đích giảm bớt các rào cản gia nhập, tạo ra các ứng dụng thực tại hỗn hợp.

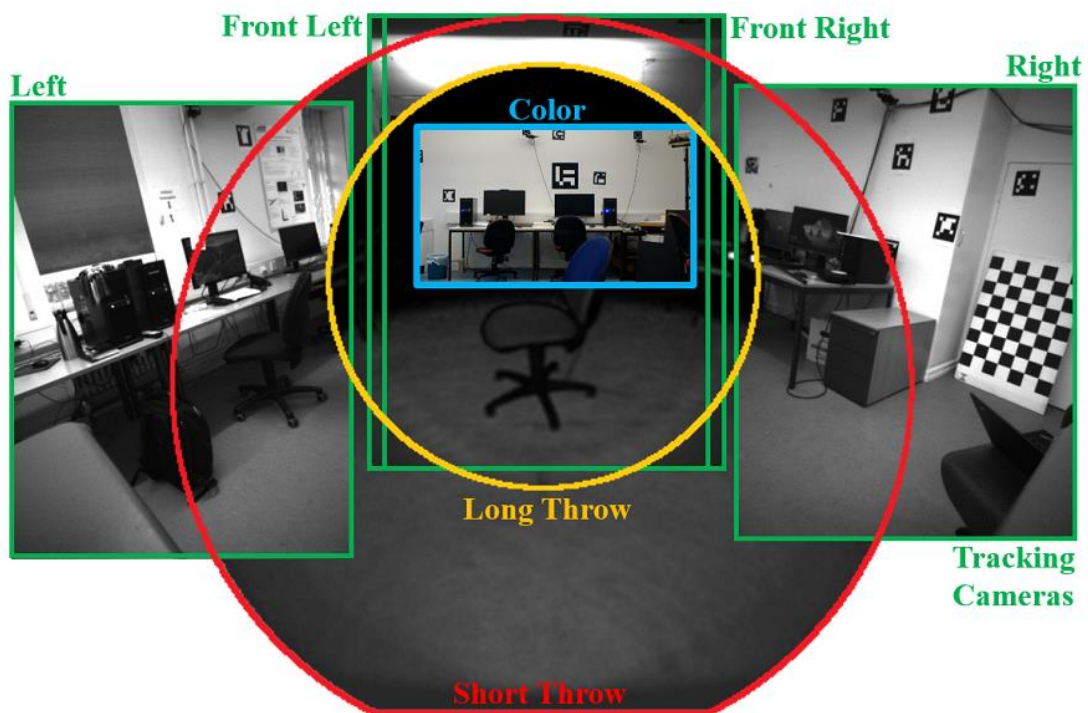
Camera	Type	Field-of-View [°]	Image Size [pixels]	Effective Pixels [%]	Frame Rate [fps]	Data Type
Photo Video	Color	40×25	1408×792 1344×756 1280×720 896×504	100	30	BGRA8
Long Throw	Depth	60×54	448×450	24	1	Gray16
Long Throw	Intensity	60×54	448×450	24	1	Gray8
Short Throw	Depth	78×77	448×450	71	15	Gray16
Short Throw	Intensity	78×77	448×450	71	15	Gray8
4 × Tracking	Grayscale	60×50	160×480	400	30	BGRA8

Bảng 3-2 Thông số kỹ thuật của sensor Hololens

Hololens được trang bị các cảm biến hình ảnh khác nhau cung cấp dữ liệu cần thiết để thực hiện các nhiệm vụ khác nhau cấu thành hệ thống thực tại tăng cường như theo dõi, tái định vị trong các môi trường đã biết và nắm bắt cấu trúc hình học của môi trường xung quanh bằng phương pháp cảm biến độ sâu.

Bảng 3-2 Thông số kỹ thuật của sensor Hololens cung cấp tổng quan về các cảm biến máy ảnh này và các đặc điểm tương ứng của chúng.

Hình 3.7 Lớp phủ các hình ảnh mà từng cảm biến ghi lại được cho thấy lớp phủ các hình ảnh được ghi lại bởi các cảm biến đó để tạo ấn tượng về cách sắp xếp của chúng trên thiết bị.



Hình 3.7 Lớp phủ các hình ảnh mà từng cảm biến ghi lại được

CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG

4.1 Phân tích thiết kế

Mục tiêu chính của phần mềm là thực hiện việc thiết kế, sắp xếp đồ nội thất cũng như đánh giá đúng vị trí tương đối của các vật thể. Để làm được điều này yêu cầu thiết bị luôn phải nhận diện được môi trường xung quanh. Do đó phần mềm phải được tích hợp các game object (module) đảm nhận việc nhận diện môi trường, tương tác giữa người dùng và các vật, tương tác giữa các vật với nhau đồng thời lưu trữ và truy xuất dữ liệu.

Khái quát lại, phần mềm cần thực hiện các nhiệm vụ sau:

- Nhận diện được mặt phẳng từ môi trường xung quanh
- Tương tác, điều khiển các mô hình 3D
- Quản lý dữ liệu chương trình
- Quản lý dữ liệu người dùng

Để đáp ứng các chức năng nêu trên, tác giả đã xây dựng các game object sau:

- Game object quản lý phiên làm việc
- Component quản lý thông tin vật thể
- Component quản lý việc tương tác
- Component quản lý dữ liệu chương trình
- Component quản lý dữ liệu người dùng

4.1.1 Quản lý dữ liệu người dùng

Dữ liệu người dùng được sử dụng để định danh người thực hiện thiết kế cũng như ngày, tháng và số lượng nội thất mà người này đã sắp xếp. Bằng những phân tích ở trên, đối tượng người dùng được thiết kế gồm những trường sau:

Tên trường trong CSDL	Kiểu dữ liệu	Mô tả
userId	string	Định danh người dùng
date	DateTime	Ngày thực hiện sắp xếp
list	Dictionary<int, ObjectInfo>	Mô tả danh sách nội thất người này đã sử dụng

Bảng 4-1 Mô tả các trường trong object user

Trong đó ObjectInfo gồm các thuộc tính sau:

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	int	Định danh của nội thất này
width	float	Chiều rộng của nội thất
length	float	Chiều dài của nội thất
height	float	Chiều cao của nội thất

Bảng 4-2 Mô tả các thuộc tính trong ObjectInfo

4.1.2 Quản lý dữ liệu chương trình

4.1.2.1. Dữ liệu các gói nội thất đã tải về

Do chương trình hỗ trợ tải về các gói nội thất bên ngoài nên cần một cơ sở dữ liệu để lưu trữ lại thông tin các gói đã tải về để có thể cache lại một cách dễ dàng tránh việc lần sau phải tải lại gói đấy. Unity đã tích hợp sẵn chức năng cache các gói tin tải về, chỉ cần biết được địa chỉ tải xuống và số hiệu phiên bản là có thể dễ dàng cache lại, nếu muốn tải lại chỉ cần thay đổi số hiệu phiên bản đó. Với những phân tích này tác giả đã tổ chức các gói nội thất gồm các trường như sau:

Tên trường trong CSDL	Kiểu dữ liệu	Mô tả
name	string	Tên gói nội thất
url	string	Địa chỉ tải xuống gói nội thất
version	uint	Phiên bản gói nội thất

Bảng 4-3 Mô tả các trường trong object cached bundle

4.1.2.2. Dữ liệu đồ nội thất

Mỗi đồ nội thất sẽ có các thông số và đặc tính khác nhau tùy thuộc vào từng loại đồ vật. Từ những yêu cầu trên các đồ nội thất có các trường và mô tả như bảng dưới đây:

Tên trường trong CSDL	Kiểu dữ liệu	Mô tả
id	int	Định danh đồ nội thất
tag	string	Loại nội thất
width	float	Chiều rộng của nội thất
length	float	Chiều dài của nội thất
height	float	Chiều cao của nội thất
isCanResize	bool	Có thể thay đổi kích cỡ được không
isOnVerticalPlane	bool	Có nằm trên tường không
isOnCeil	bool	Có nằm trên trần không

Bảng 4-4 Mô tả các trường trong object furniture

4.1.3 Thiết kế component để quản lý vật thể

ObjectManager là một component để quản lý vật thể gồm có 2 chức năng:

Khi một đối tượng được sinh ra (người dùng ấn chọn thêm 1 đồ nội thất) thì sẽ gọi đến component này để thiết lập các tham số ban đầu

Component này sẽ được thay đổi thuộc tính khi có một đồ nội thất được chọn làm đối tượng:

- Khi có đối tượng được chọn: ObjectManager sẽ lưu lại các thông số liên quan đến đối tượng đấy, đồng thời bật các component liên quan đến việc hiển thị đối tượng được chọn: BoxTransform dùng để hiện bao của đồ nội thất, ResizeObject dùng để hiển thị UI resize vật thể
- Khi đối tượng được bỏ chọn: ObjectManager sẽ bỏ các thông số liên quan đến đối tượng đấy, đồng thời tắt các component liên quan đến việc hiển thị đối tượng được chọn

Thuộc tính	Mô tả
public GameObject targetGameObject	Game object đang được chọn
public GameObject deployParent	Game object cha của tất cả đồ nội thất
private RoomManager roomManager	Component quản lý thông tin người dùng
private ResizeObject resizeComponent;	Component quản lý thu phóng vật thể
private AddRemoveObject addRemoveObject;	Component quản lý thêm xóa vật thể
private BoxTransform boxComponent;	Component quản lý đường bao vật thể
public Vector3 objectSize;	Vector chứa kích thước vật thể
public bool isCanResize;	Xác định xem vật thể thu phóng được không
public bool isOnVerticalPlane;	Xác định xem vật thể có nằm trên tường không
public bool isOnCeil;	Xác định vật thể có nằm trên trần không

Bảng 4-5 Các thuộc tính quản lý vật thể được chọn

Nguyên mẫu hàm	Mô tả
void setTargetGameObject()	Đặt lại đối tượng được chọn
void deployObject()	Sinh đối tượng nội thất mới

Bảng 4-6 Các hàm trong component ObjectManager

4.1.4 Thiết kế component cho việc tương tác

4.1.4.3. Thiết kế component hiển thị hình bao của vật thể

Component BoxTransform phụ trách việc sinh hình bao cho vật thể dựa vào kích cỡ collider của chúng. Hình bao này sẽ là một hình hộp chữ nhật bao xung quanh vật thể với 8 đỉnh là 8 nút giúp thu phóng vật thể, còn ở trung điểm mỗi cạnh sẽ là nút giúp xoay vật thể.

Thuộc tính	Mô tả
public float width	Độ dày của cạnh hình bao
private GameObject _vertexPrefab;	Prefab của nút đỉnh
private GameObject _edgePrefab;	Prefab của nút trung điểm cạnh

Bảng 4-7 Các thuộc tính quản lý hình bao của vật thể

Trong hàm Update() chúng ta sẽ chạy 4 hàm dưới đây để cập nhật vị trí hình bao chính xác so với vật thể

Nguyên mẫu hàm	Mô tả
void CalcPositons()	Tính toán vị trí mới của hình bao
void DrawBox()	Vẽ các cạnh của hình bao
void SetEdgesPosition()	Đặt lại vị trí các nút trung điểm cạnh
void SetVerticesPosition()	Đặt lại vị trí các nút đỉnh

Bảng 4-8 Các hàm vẽ hình bao của vật thể

4.1.4.4. Thiết kế component quản lý việc di chuyển, xoay, thu phóng

Các component gồm các cặp tương ứng như sau

Tên component phụ trách tương tác người dùng	Tên component phụ trách logic	Mô tả
DragMove	MoveObject	Di chuyển vật thể
DragRotate	RotateObject	Xoay vật thể
DragResize	ResizeObject	Thu phóng vật thể

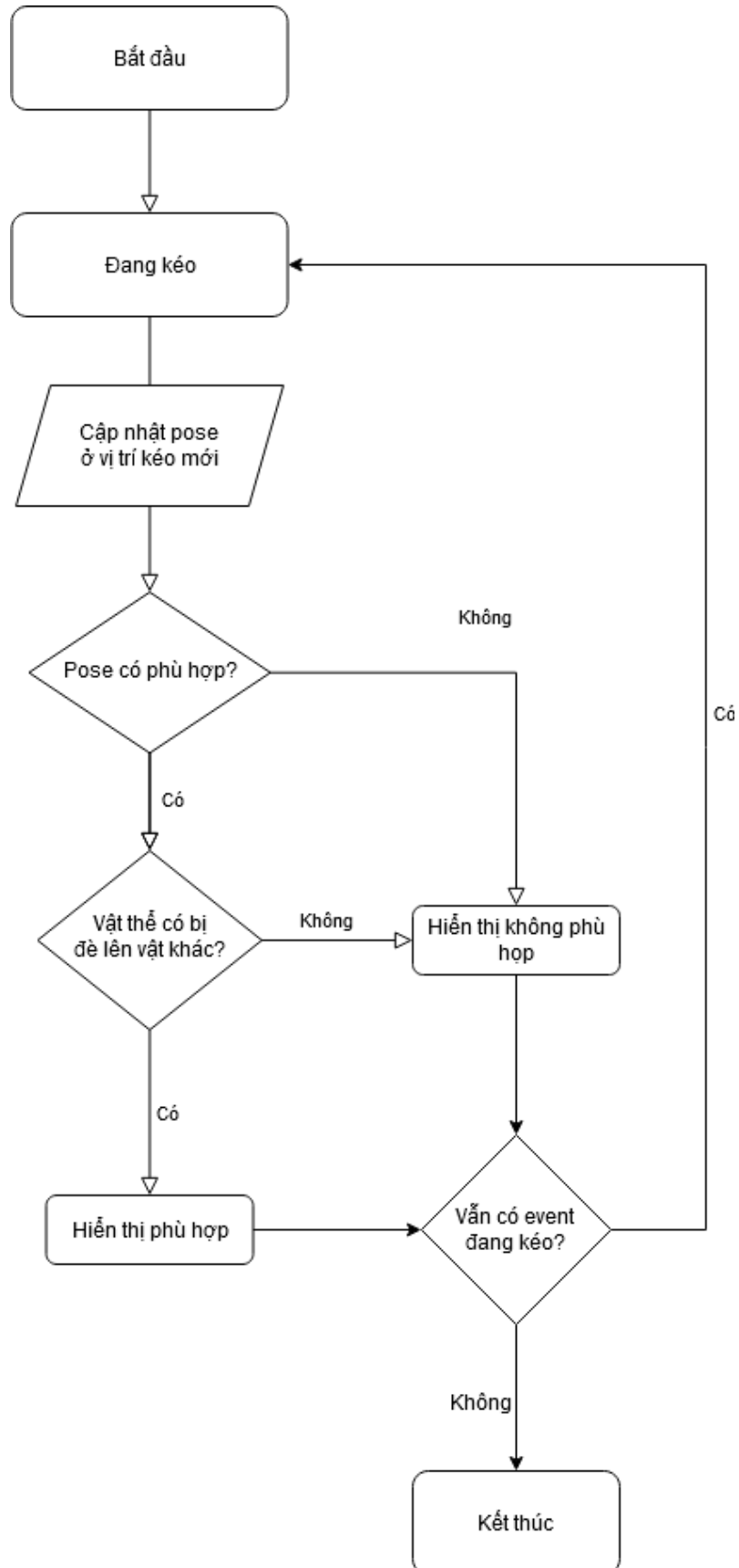
Bảng 4-9 Các cặp component phụ trách tương tác giữa người dùng với vật thể

Trong đó các component DragMove, DragResize và DragRotate liên quan đến việc kéo thả để di chuyển vật/xoay/thu phóng vật thể đều được subscribe vào các event input, nhờ đó khi người dùng tác động vào vật thể với input tương ứng thì component sẽ tự động thực hiện các handler tương ứng.

Tên event	Tên handler	Mô tả
IBeginDragHandler	OnBeginDrag()	Khi bắt đầu kéo
IDragHandler	OnDrag()	Khi đang kéo
IEndDragHandler	OnEndDrag()	Khi thả ra

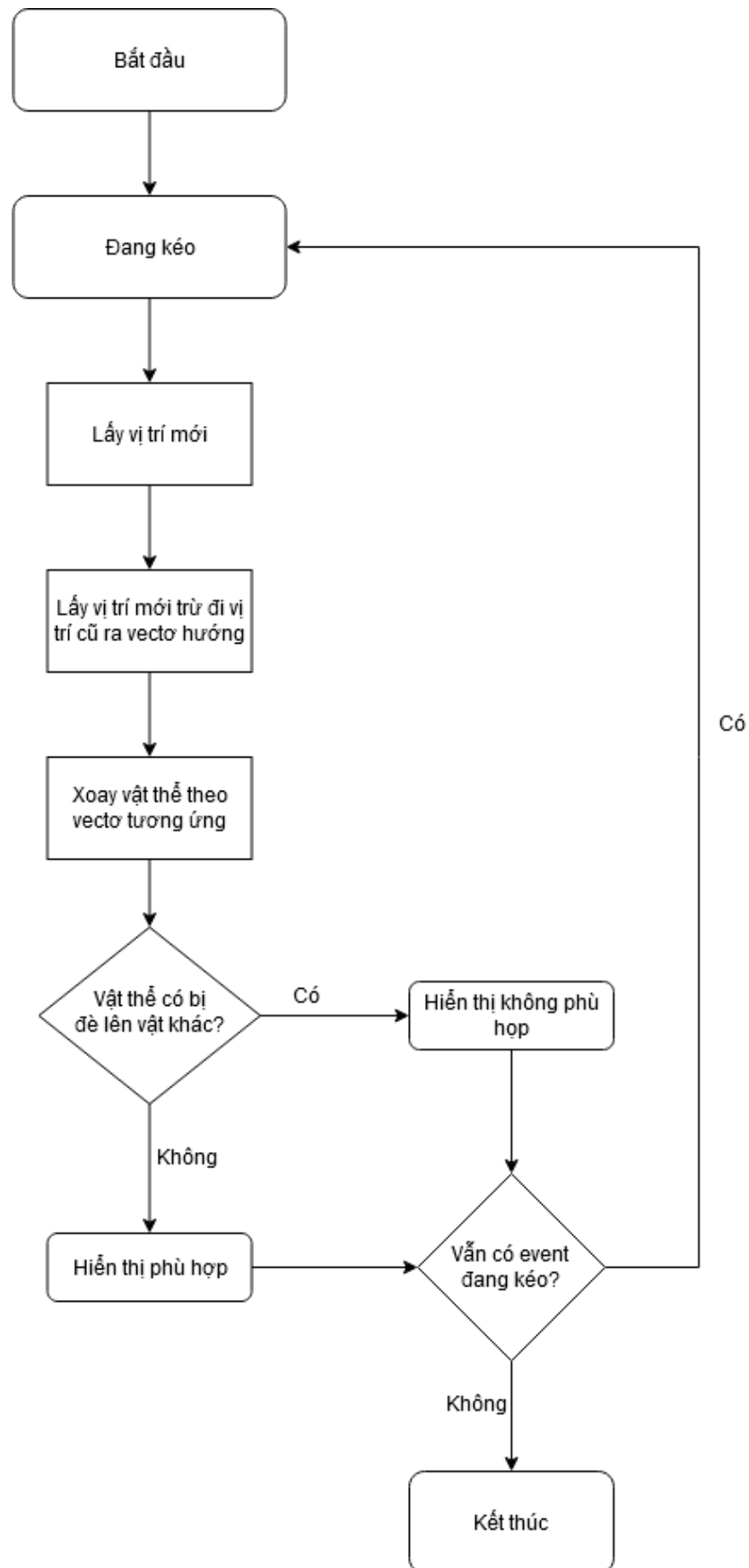
Bảng 4-10 Các event và event handler tương ứng

4.1.4.5. Di chuyển vật thể



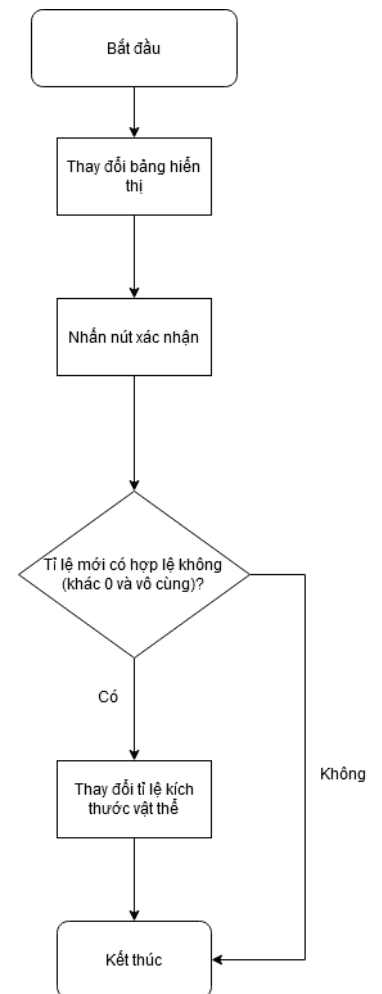
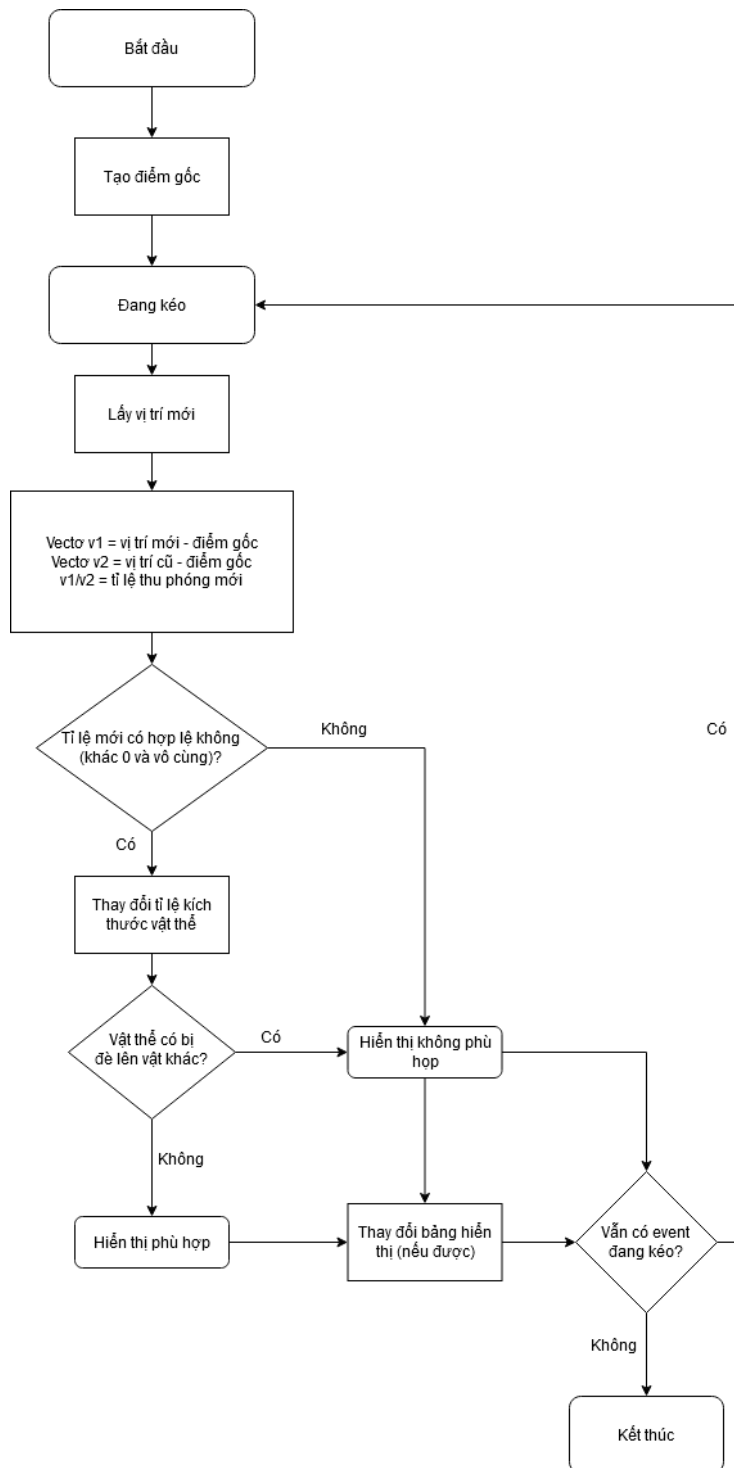
Hình 4.1 Lưu đồ thuật toán của DragMove

4.1.4.6. Xoay vật thể



Hình 4.2 Lưu đồ thuật toán của DragRotate

4.1.4.7. Thu phóng vật thể



Hình 4.3 Lưu đồ thuật toán của DragResize

4.1.5 Thiết kế component quản lý lưu trữ dữ liệu

4.1.5.8. Lưu trữ dữ liệu người dùng

Component RoomManager dùng để tổng hợp dữ liệu về lần thiết kế nội thất này của người chơi và lưu trữ ra một CSDL. Ở đây để tối ưu thời gian phát triển, cũng như linh hoạt, bảo mật chúng ta sử dụng Google Firebase.

Thuộc tính	Mô tả
private GameObject deployParent;	Xác định game object cha của các đối tượng nội thất để tổng hợp thành list
private DatabaseReference mDatabaseRef;	Reference của google firebase realtime database
public bool isEdit;	Kiểm tra xem chương trình có đang chỉnh sửa không
public bool isOnExit;	Kiểm tra xem chương trình có đang thoát không

Bảng 4-11 Các thuộc tính lưu trữ thông tin người dùng

Nguyên mẫu hàm	Mô tả
private void uploadData()	Upload dữ liệu dưới dạng json lên firebase
public void OnExitClick()	Khi người dùng ấn vào nút thoát
public void OnConfirmClick()	Khi người dùng ấn vào nút upload

Bảng 4-12 Các hàm liên quan đến lưu trữ thông tin người dùng

4.1.5.9. Lưu trữ gói nội thất đã tải

Component ABUtils dùng để quản lý các gói asset bundle của chương trình, khi người dùng tải một asset bundle trên mạng về thì component sẽ tự động lưu lại thông tin của gói nội thất đẩy ra một file json lưu trên bộ nhớ của thiết bị, những lần sau thì chương trình sẽ tự động cache các thông tin từ trong file json đẩy ra mà không phải tải lại lần nữa.

Thuộc tính	Mô tả
public static List<> cachedList;	List chứa thông tin các gói asset bundle đã tải về
public static ConcurrentDictionary<> goDict;	Dictionary chứa các mô hình nội thất theo id
public static ConcurrentDictionary<> goPropDict;	Dictionary chứa thông tin các mô hình nội thất theo id
public static ConcurrentDictionary<> tagDict;	Dictionary chứa thông tin các loại nội thất theo id

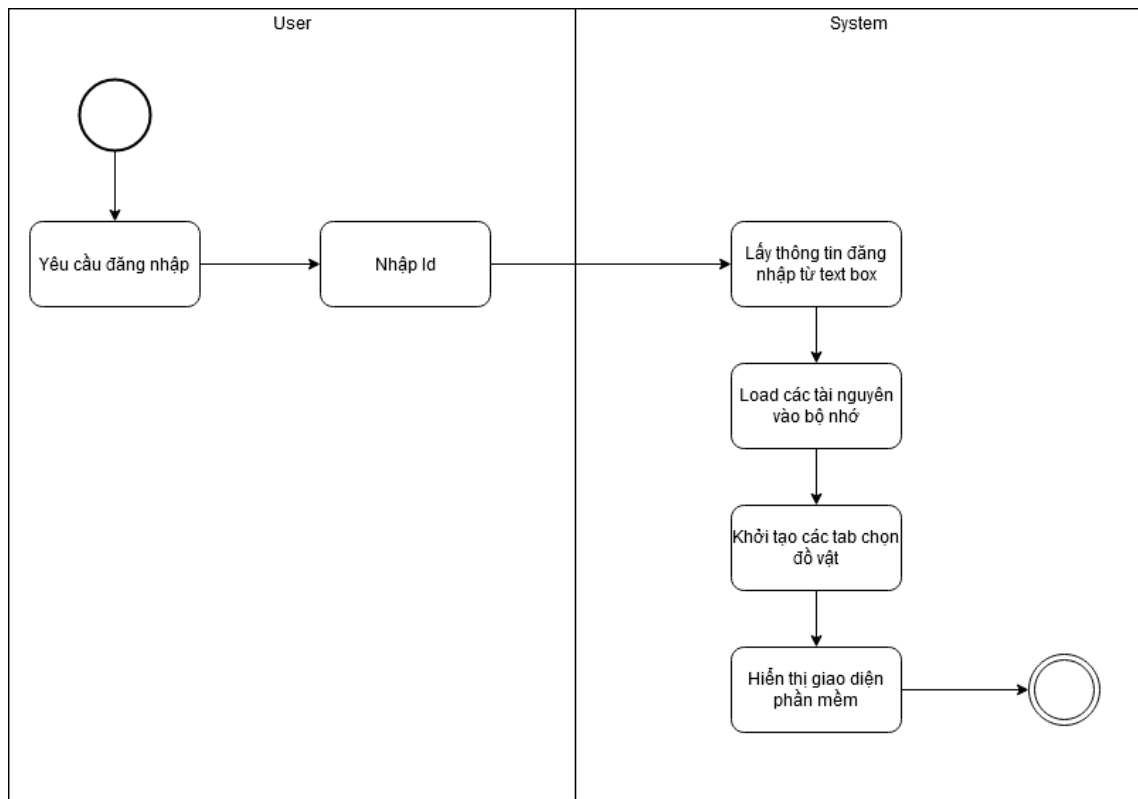
Bảng 4-13 Các thuộc tính dùng để lưu các asset bundle đã tải

Nguyên mẫu hàm	Mô tả
private static void SaveLocal()	Lưu gói nội thất về máy
private static void LoadLocal()	Tải gói nội thất vào bộ nhớ
public static void RemoveLocal()	Xoá gói nội thất khỏi list chứa thông tin gói nội thất đã tải về

Bảng 4-14 Các hàm dùng để tải cũng như lưu asset bundle

4.1.6 Mô tả quy trình thao tác

4.1.6.10. Thao tác đăng nhập

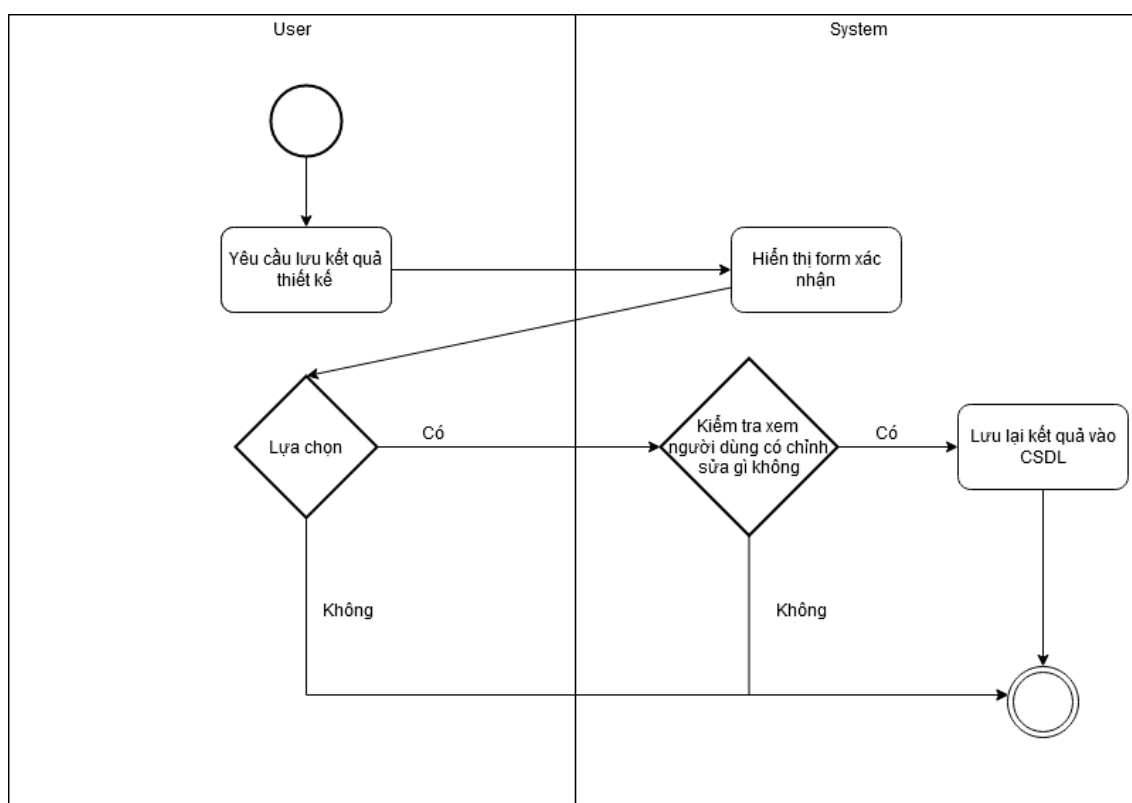


Hình 4.4 Biểu đồ activity thao tác đăng nhập

Mô tả quy trình thao tác:

- Người dùng khởi động phần mềm đã cài trên máy, hệ thống sẽ hiển thị form đăng nhập, yêu cầu người dùng nhập id đăng nhập.
- Người dùng nhập tài khoản và chọn button đăng nhập.
- Khi đăng nhập hệ thống sẽ hiển thị màn hình tải để tải tài nguyên vào bộ nhớ. Khi tải xong bắt đầu khởi tạo các tab tùy theo có bao nhiêu loại nội thất
- Sau khi khởi tạo xong sẽ đóng màn hình tải lại, hiển thị giao diện phần mềm

4.1.6.11. Thao tác lưu kết quả

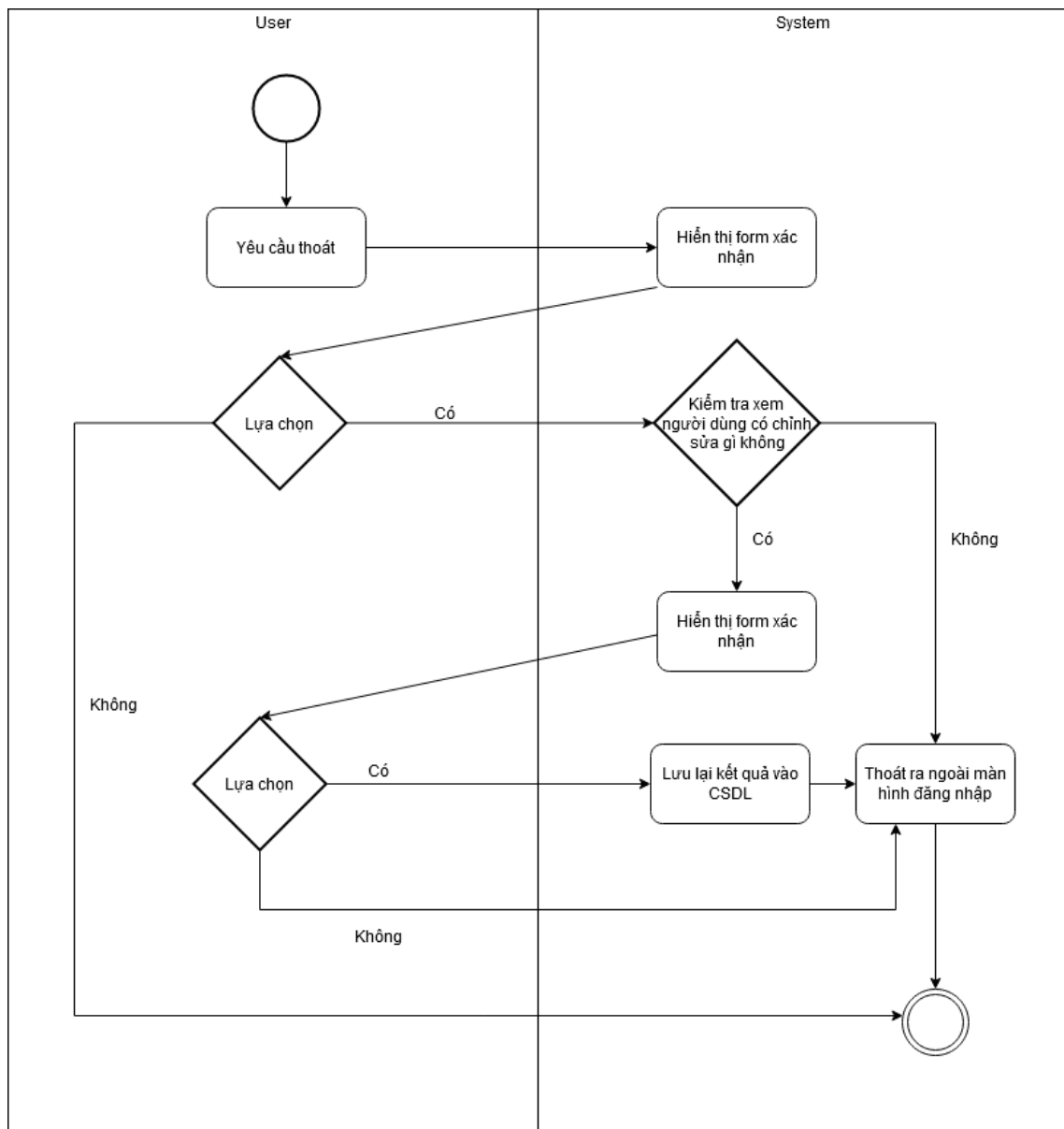


Hình 4.5 Biểu đồ activity thao tác lưu kết quả

Mô tả quy trình thao tác:

- Người dùng chọn nút lưu kết quả thiết kế
- Phần mềm hiển thị bảng chọn xác nhận lưu
- Nếu người dùng nhấn lưu thì sẽ lưu kết quả lên CSDL
- Sau khi lựa chọn xong thì đóng bảng lại

4.1.6.12. Thao tác thoát ra ngoài màn hình đăng nhập



Hình 4.6 Biểu đồ activity thao tác thoát ra màn hình đăng nhập

Mô tả quy trình thao tác:

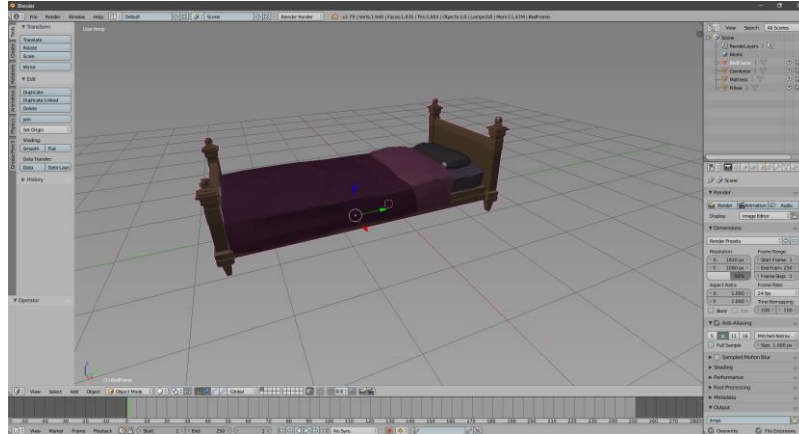
- Người dùng chọn nút thoát
- Phần mềm hiển thị bảng chọn xác nhận thoát
- Nếu người dùng nhấn không thì đóng bảng chọn
- Nếu người dùng nhấn có thì kiểm tra người dùng có chỉnh sửa thêm gì thiết kế không, nếu có chỉnh sửa thêm thì hiện bảng hỏi người dùng có muốn lưu không, nếu người dùng chọn có thì lưu kết quả lên CSDL
- Sau khi lựa chọn xong thì chuyển scene về login

4.2 Cài đặt chương trình

4.2.1 Tạo gói nội thất mới

4.2.1.1. Thiết kế các nội thất

Từ phần mềm thiết kế (ví dụ như Autocad, Blender,...), người thiết kế có thể tạo nên mô hình nội thất mong muốn



Hình 4.7 Thiết kế nội thất bằng blender

Sau đó họ chỉ cần xuất các file nội thất ra định dạng Unity Editor có thể đọc được, ở đây tác giả chọn định dạng FBX.

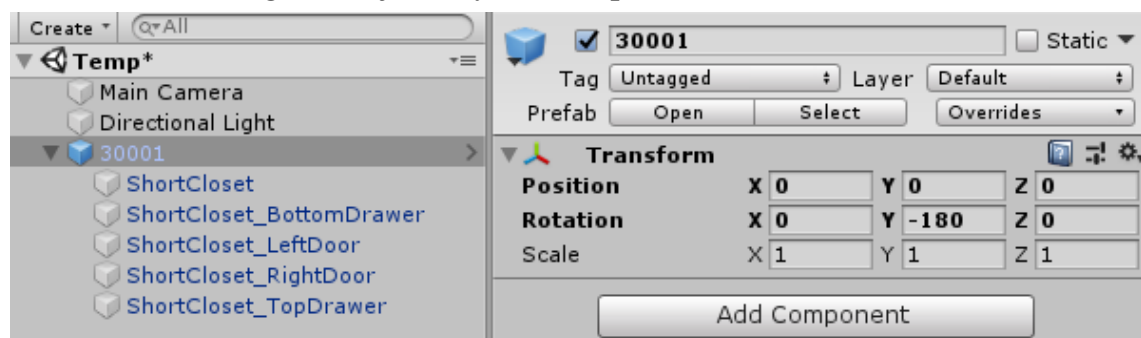
4.2.1.2. Đóng gói nội thất

Unity Editor sẽ đọc các file FBX chứa mô hình nội thất này, tác giả chỉ cần kéo thả vào trong Scene View để tạo thành một game object có cấu trúc tương tự với file thiết kế được tạo trong Blender.



Hình 4.8 So sánh cấu trúc mô hình trong Blender và trong Unity

Để có thể quản lý mô hình dễ dàng, cũng như có thể thêm các tương tác đặc biệt sau này, tác giả sẽ cho game object này là con của 1 game object với tên là id của đồ nội thất và lưu game object này thành 1 prefab.

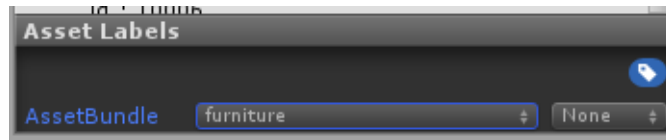


Hình 4.9 Cấu trúc một prefab chứa mô hình nội thất

Các prefab cùng với 1 file cấu hình chứa thông tin cần thiết sẽ được đóng gói thành 1 asset bundle. Tuy nhiên việc tạo nên asset bundle chỉ có thể thực hiện được với Unity Editor. Để tạo 1 asset bundle chúng ta phải có AssetBundles Browser. Tải xuống unitypackage AssetBundles Browser tại:

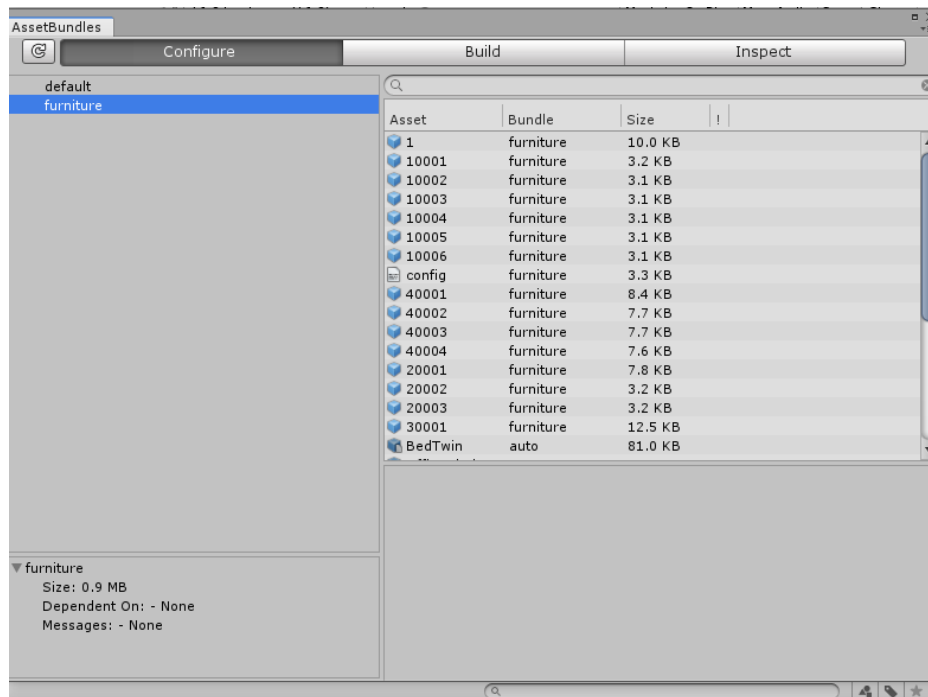
<https://github.com/Unity-Technologies/AssetBundles-Browser>

Trong màn hình của Unity Editor, chọn asset muốn đưa vào trong asset bundle, ở góc phải dưới của editor, phần asset labels, chọn asset bundle muốn thêm asset này vào



Hình 4.10 Bảng dán nhãn asset bundle cho asset mong muốn

Sau khi chọn xong asset muốn thêm vào, chọn Window > AssetBundle Browser để mở cửa sổ AssetBundle Browser



Hình 4.11 Giao diện của AssetBundle Browser

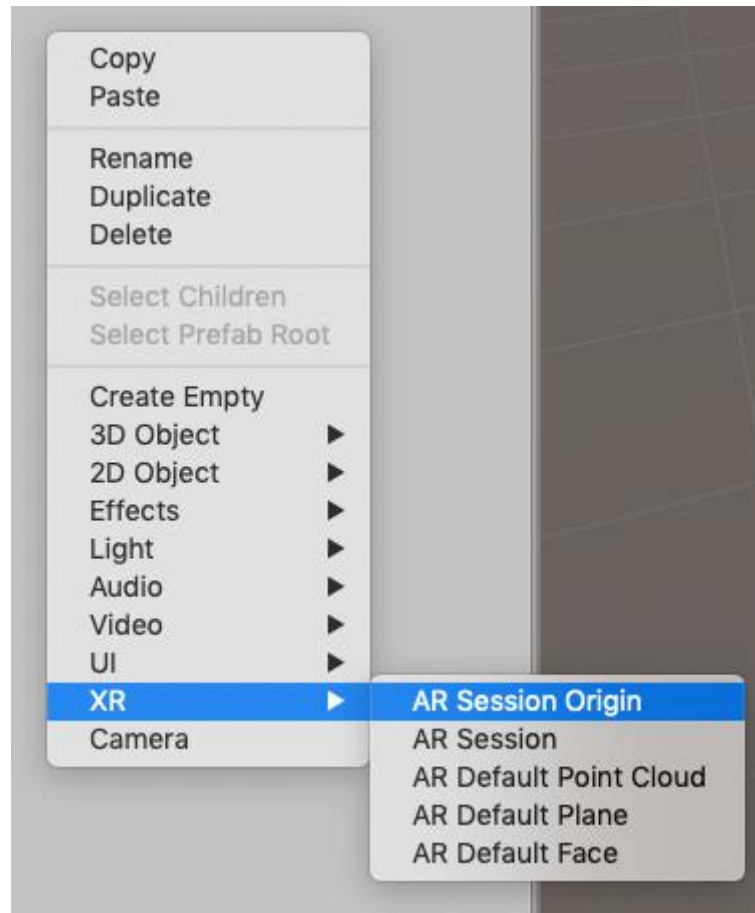
Chọn tab build và ấn nút build để unity bắt đầu tạo asset bundle; sau khi tạo xong sẽ ra một số file có dạng <tên asset bundle> và <tên asset bundle>.manifest thì file không có đuôi là file asset bundle. Đưa file này lên 1 server chia sẻ là có thể sử dụng được.

4.2.1.3. Sử dụng gói nội thất đã tạo

Định dạng gói asset bundle của chương trình sẽ là file config.json cùng với các file prefab mang id của đồ nội thất, trong file config chứa các trường thông tin của đồ nội thất để chương trình khi đọc file config sẽ tự nạp vào.

4.2.2 Tạo game object quản lý phiên làm việc cho thiết bị di động

Để khởi tạo game object liên quan đến việc quản lý phiên AR của thiết bị, chuột phải vào cây phân cấp chọn XR > AR Session. Lúc này sẽ có 2 game object được tạo ra là AR Session và AR Session Origin



Hình 4.12 Menu khởi tạo phiên làm việc AR

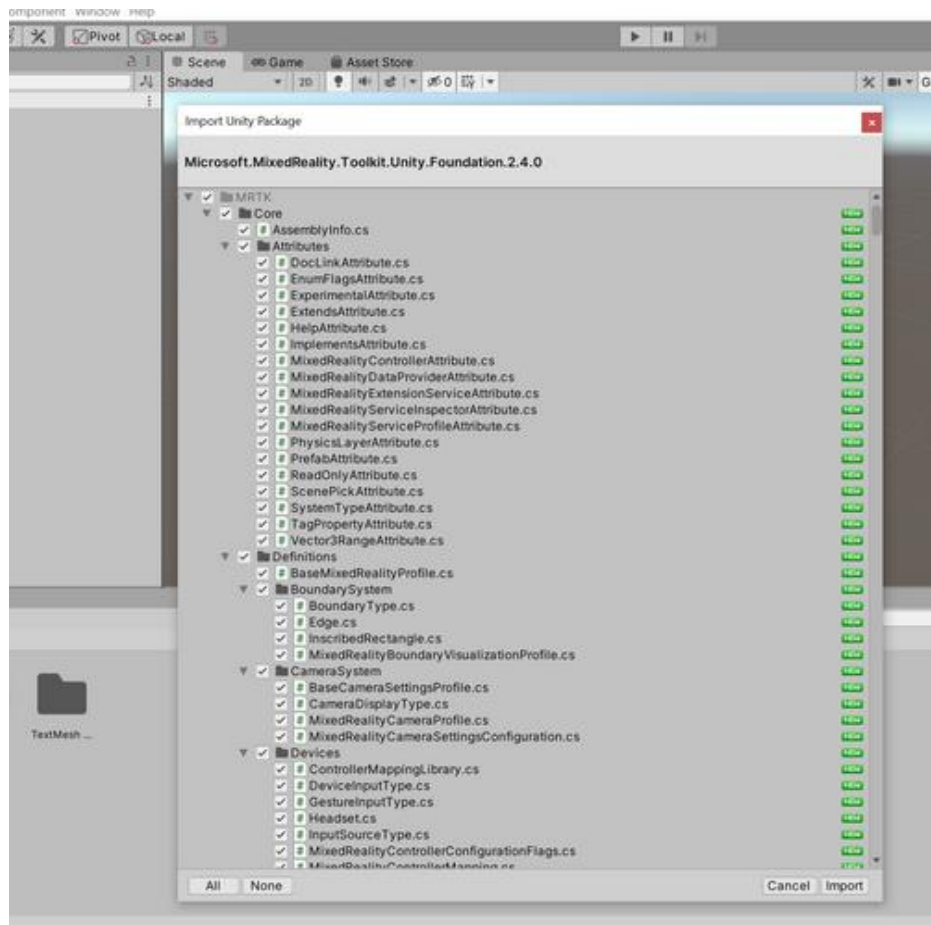
AR Session kiểm soát vòng đời của trải nghiệm AR bằng cách bật hoặc tắt AR trên nền tảng đích. AR Session có thể có trên bất kỳ game object nào. Khi lập trình viên tắt AR Session, hệ thống không còn theo dõi các đặc trưng trong môi trường nữa, nhưng nếu nó được bật lại vào thời điểm sau đó, hệ thống sẽ cố gắng khôi phục và duy trì các đặc trưng đã được phát hiện trước đó.

Mục đích của AR Session Origin là chuyển đổi các đặc trưng của trackable, chẳng hạn như plane và điểm đặc trưng, thành vị trí, hướng và tỷ lệ cuối cùng của chúng trong scene. Vì các thiết bị AR cung cấp dữ liệu của chúng trong "không gian phiên", là không gian chưa được chia tỷ lệ so với đầu phiên AR, AR Session Origin thực hiện chuyển đổi thích hợp thành không gian trong game. Các trackable mà thiết bị AR tạo ra, chẳng hạn như plane, được cung cấp trong "không gian phiên", liên quan đến hệ tọa độ của thiết bị. Khi được khởi tạo trong Unity dưới dạng game object, chúng cũng có một vị trí trong không gian game. Để khởi tạo chúng ở đúng vị trí, AR Foundation cần biết nguồn gốc phiên nên ở đâu trong scene.

4.2.3 Tạo game object quản lý phiên làm việc cho Hololens

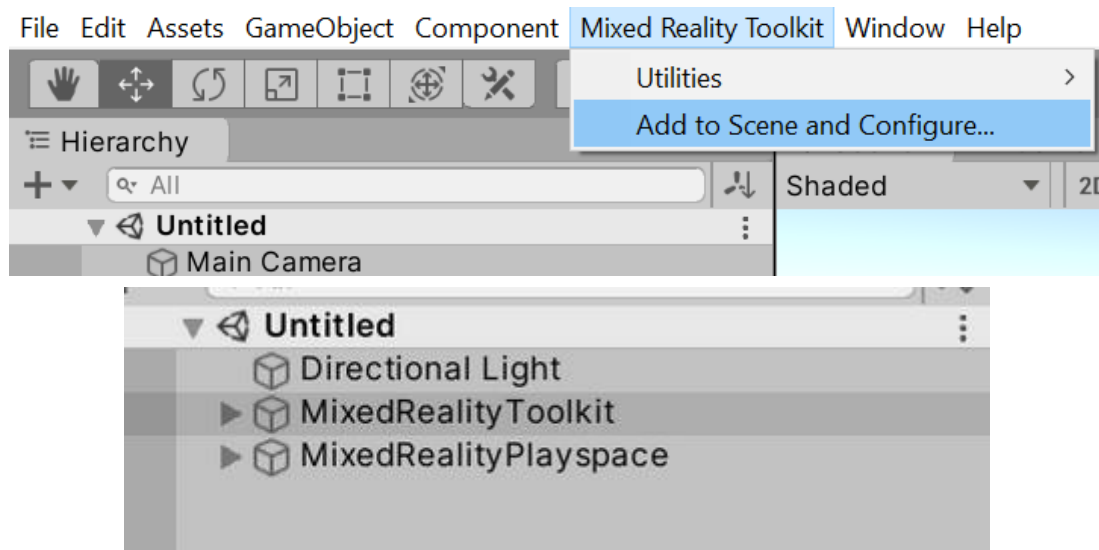
Download unitypackage của MRTK trên trang chủ Microsoft

Sau đó chọn Assets > Import Package > Custom Package để bắt đầu import



Hình 4.13 Menu import các thư viện của MRTK

Sau khi import hoàn tất chọn Mixed Reality Toolkit > Add to Scene and Configure... để add các game object cần thiết của MRTK vào scene



Hình 4.14 Cách thêm và các game object đảm nhận phiên làm việc của MRTK

4.2.4 Xây dựng hệ thống

Khi xây dựng một chương trình lớn, việc chia chương trình thành những module nhỏ sẽ giúp việc quản lý mã nguồn hiệu quả và thuận tiện cho việc bảo trì, nâng cấp phần mềm sau này. Từ đó nâng cao hiệu suất lập trình cũng như thời gian phát triển sản phẩm. Bằng những phân tích và thiết kế ở 4.1, tác giả đã xây dựng hệ thống gồm 4 game object nhỏ:

- AR Session: dùng để quản lý phiên làm việc của AR
- GameManager: chứa các hệ thống quản lý dữ liệu cũng như quản lý vật thể được chọn
- Interaction: dùng để quản lý các tương tác giữa người dùng với vật thể và giữa các vật thể với nhau
- Ingame Canvas: chứa các thành phần giao diện người dùng

Chương trình được xây dựng gồm 2 luồng chính:

- Luồng chạy ứng dụng xử lý giao diện và thao tác với vật thể.
- Luồng điều khiển, xử lý thao tác với dữ liệu và AR.

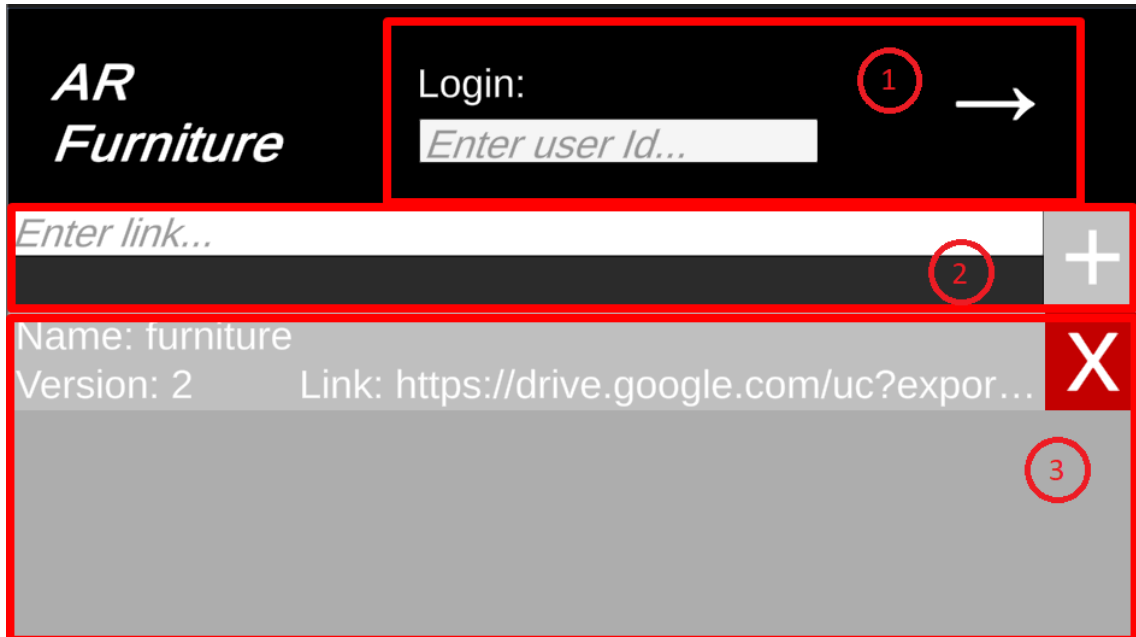
Luồng chạy xử lý giao diện chịu trách nhiệm tương tác với người dùng, xử lý những sự kiện mà người dùng tạo ra như sự kiện người dùng tap, chạm hoặc kéo thả.

Khi chương trình bắt đầu tải các tài nguyên thì hai luồng này hoạt động song song với nhau nên có thể luồng xử lý giao diện không thể lấy được dữ liệu do luồng điều khiển chưa tải xong thì lúc này phải đảm bảo tính toàn vẹn dữ liệu bằng cách ưu tiên cho luồng điều khiển chạy trước luồng xử lý giao diện. Điều này có thể giải quyết bằng cách cho hàm khởi tạo của luồng điều khiển nằm ở Awake() còn hàm khởi tạo của luồng giao diện nằm ở Start() thì sẽ đảm bảo được thứ tự chạy của 2 luồng.

Không chỉ vậy, do khi tải các tài nguyên chúng ta chia thành nhiều thread download đồng thời nên không thể dùng Dictionary thường để lưu trữ dữ liệu được mà phải dùng ConcurrentDictionary để đảm bảo dữ liệu không bị hỏng hay gặp sự cố liên quan đến concurrent. Khi đó thay vì sử dụng các hàm Add() với Get() mà sử dụng các hàm riêng của ConcurrentDictionary là TryAdd() và TryGetValue().

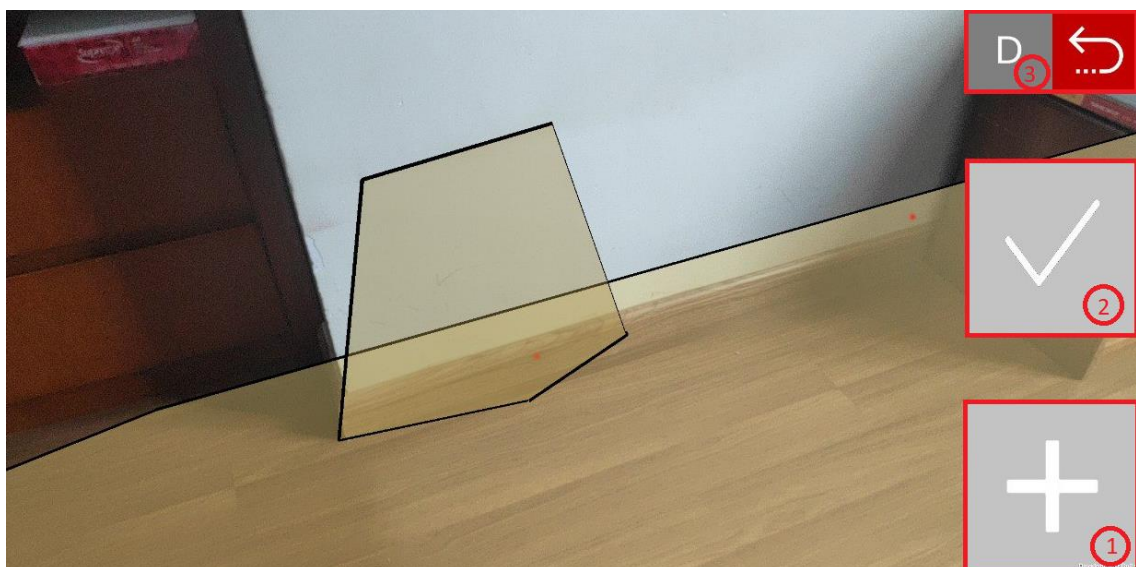
CHƯƠNG 5. KẾT QUẢ VÀ TỔNG KẾT

5.1 Giao diện của chương trình

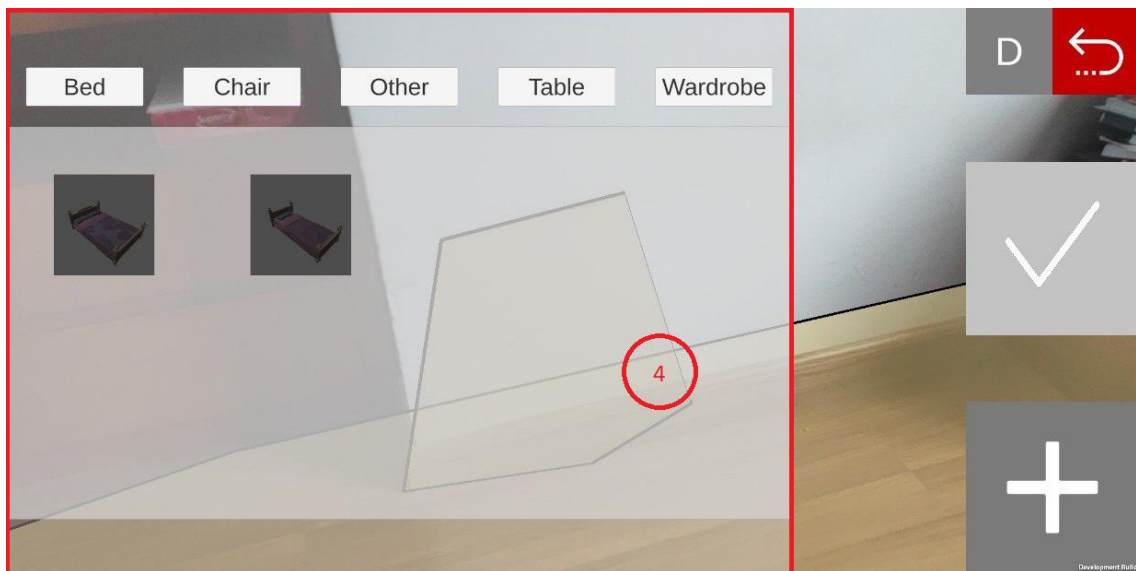


Hình 5.1 Giao diện đăng nhập của phần mềm

1. Ô nhập ID và nút đăng nhập
2. Ô nhập link tải các gói nội thất
3. Bảng thông tin các gói nội thất đã tải xuống

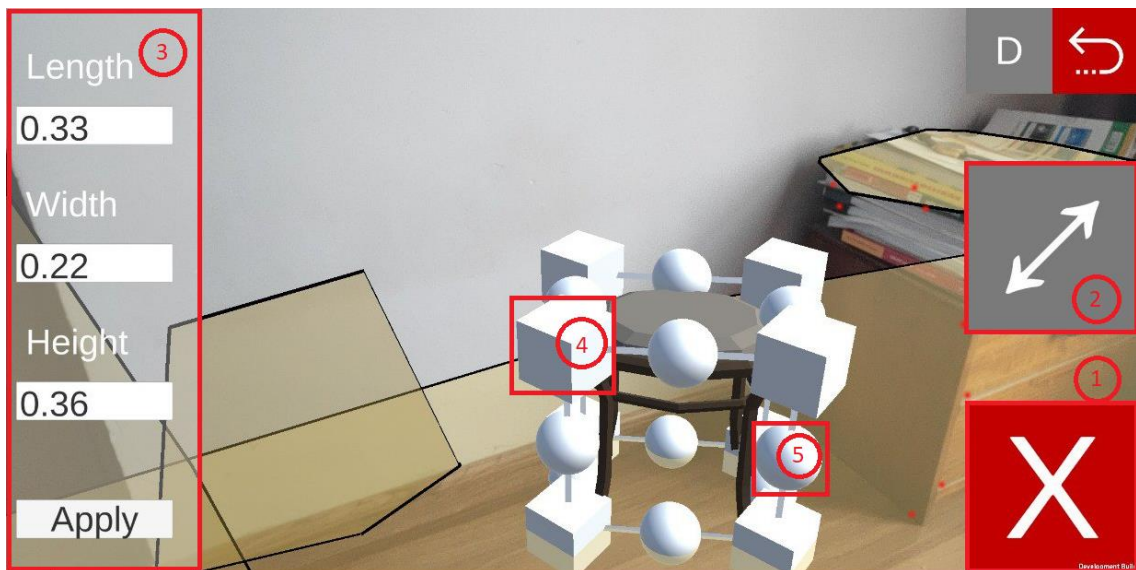


Hình 5.2 Giao diện chính của phần mềm



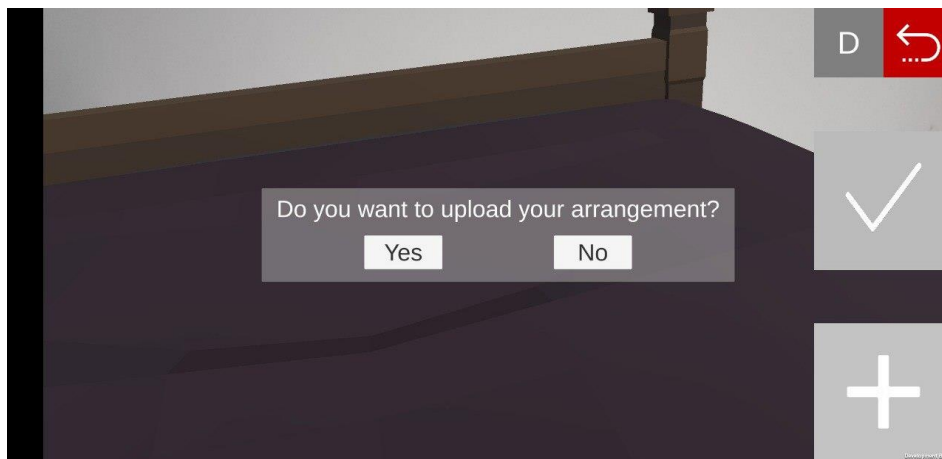
Hình 5.3 Giao diện chính của phần mềm kèm bảng danh sách nội thất

1. Nút mở tab chọn đồ nội thất
2. Nút lưu thiết kế
3. Nút bật tắt chế độ AR và thoát ra màn hình đăng nhập
4. Bảng danh sách nội thất hiện có

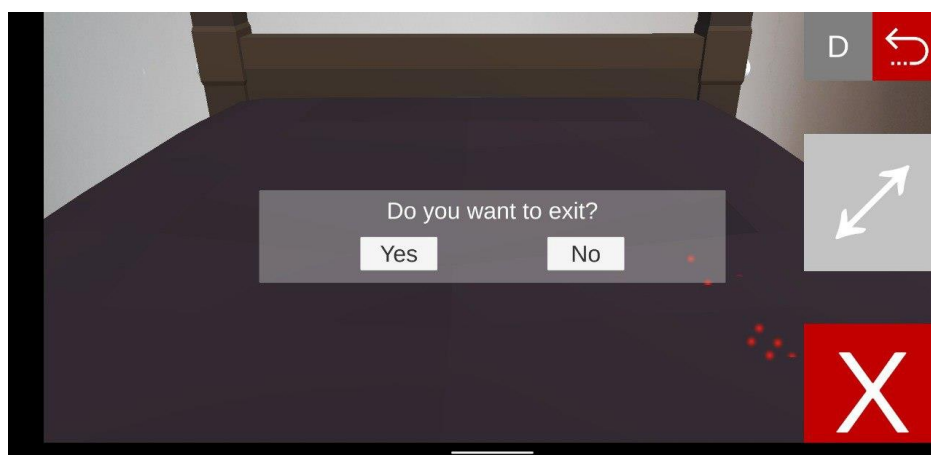


Hình 5.4 Giao diện khi chọn một vật thể

1. Nút xoá vật thể
2. Nút mở thanh thu phóng vật thể
3. Thanh thu phóng vật thể
4. Nút thu phóng vật thể
5. Nút xoay vật thể

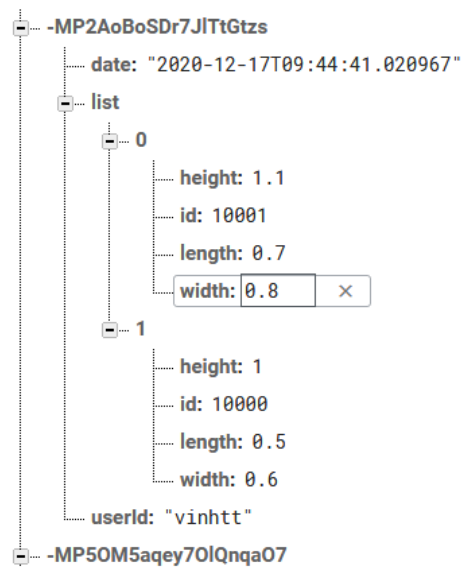


Hình 5.5 Bảng xác nhận khi ấn lưu



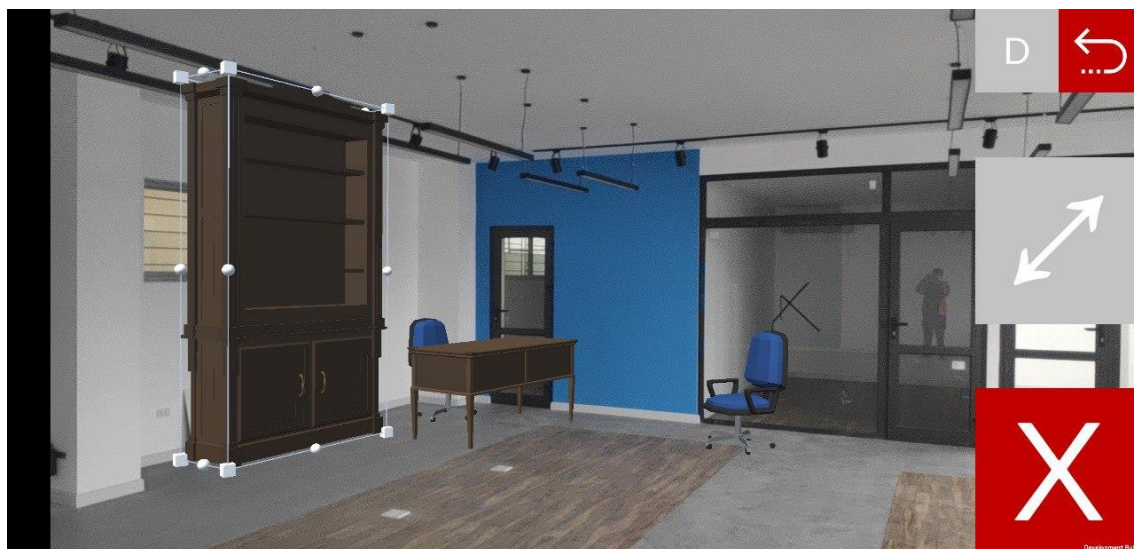
Hình 5.6 Bảng xác nhận khi ấn thoát

Kết quả gửi lên server



Hình 5.7 Kết quả lưu được trên CSDL

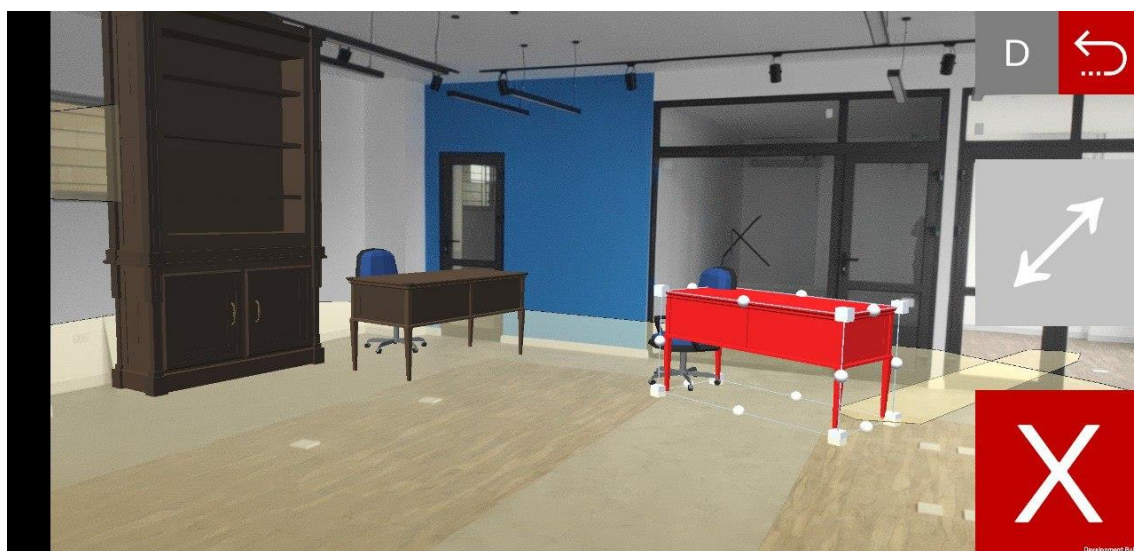
5.2 Thử nghiệm trên phòng lab 505, 802



Hình 5.8 Thử nghiệm trên phòng 505 (tắt hiện mặt phẳng)



Hình 5.9 Thử nghiệm trên phòng 505 (bật hiện mặt phẳng)



Hình 5.10 Thử nghiệm trên phòng 505 khi đặt ở vị trí không phù hợp



Hình 5.11 Thử nghiệm trên phòng 505 khi đặt ở vị trí phù hợp



Hình 5.12 Thử nghiệm ở phòng 802



Hình 5.13 Thử nghiệm ở phòng 802 khi có vật thể chắn

5.3 Tổng kết

5.3.1 Các kết quả đạt được

Phần mềm đã được cài đặt và chạy thử và thực công thành công các chức năng quản lý gói nội thất, thêm, xóa, di chuyển, xoay và thu phóng nội thất.

5.3.2 Vấn đề còn tồn tại

Về giao diện: Trong một số thao tác chuyển đổi, hình ảnh chuyển đổi vẫn chưa được mượt mà, hình dạng nút bấm còn thô sơ, chưa chăm chút. Có cảm giác AR chưa được chân thực lắm.

Về cơ sở dữ liệu: Tuy lượng dữ liệu sinh ra một lần rất nhỏ nhưng về lâu dài (có thể 10 – 15 năm sau) lượng dữ liệu trở nên nhiều hơn có thể làm cho các câu lệnh truy vấn trở nên khó khăn..

Về phân quyền người dùng: Chưa có phân quyền người dùng, ai cũng có thể chỉnh sửa danh sách gói nội thất.

5.3.3 Định hướng phát triển

Để đảm bảo sự tiện lợi cũng như tính ổn định của phần mềm, tác giả có một số định hướng phát triển như sau:

- Khắc phục những vấn đề đã nêu ở mục trên.
- Thêm tính năng undo/redo
- Thêm các tính năng thống kê để đánh giá nhu cầu người dùng, từ đây có các định hướng tốt hơn cho cả người mua lẫn người bán.
- Thêm tính năng tìm kiếm dựa theo loại nội thất
- Thêm tính năng cho phép tải cả những gói nội thất có định dạng khác.

5.3.4 Kết luận

Về tổng thể, phần mềm đã đáp ứng được các yêu cầu của thực tế. Nhưng do kinh nghiệm còn hạn chế nên một số vấn đề trong lập trình cũng như trong báo cáo vẫn chưa được xử lý tốt. Em rất mong được các thầy nhiệt tình góp ý để phần mềm được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

- [1] J. Scribani, "Visual Capitalist," 16 January 2019. [Online]. Available: <https://www.visualcapitalist.com/extended-reality-xr/>. [Accessed 15 11 2020].
- [2] "Hello Future," Orange, 22 August 2020. [Online]. Available: <https://hellofuture.orange.com/en/extended-reality-for-learning-designing-visiting-etc/>. [Accessed 15 11 2020].
- [3] E. Zubkov, "blog.griddynamics.com," 10 January 2018. [Online]. Available: <https://blog.griddynamics.com/how-arkit-and-arcore-recognize-vertical-planes/>.
- [4] P. Hübner, K. Clintworth, Q. Liu, M. Weinmann and S. Wursthorn, "Evaluation of Hololens Tracking and Depth Sensing for Indoor Mapping Applications," *Sensors* 2020, no. 20, p. 1021.
- [5] [Online]. Available: <https://developers.google.com/ar/discover>.
- [6] [Online]. Available: <https://developer.apple.com/documentation/arkit>.
- [7] [Online]. Available: <https://microsoft.github.io/MixedRealityToolkit-Unity/>
- [8] [Online]. Available: docs.unity3d.com/Manual/com.unity.xr.arfoundation