

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

LÊ ĐÌNH HÂN – TRẦN PHƯỚC HIỀN

ỨNG DỤNG ABC

KHOÁ LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

X

TP.HCM, 2018

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

LÊ ĐÌNH HÂN	1412153
TRẦN PHƯỚC HIỀN	1412171

ỨNG DỤNG ABC

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN TIN HỌC

GIÁO VIÊN HƯỚNG DẪN
ThS. TRẦN QUANG TÁNH – ThS. ĐẶNG BÌNH PHƯƠNG

NIÊN KHÓA 2014 – 2018

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.

TP.HCM, ngày tháng năm 2018

Giáo viên hướng dẫn

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.

TP.HCM, ngày tháng năm 2018

Giáo viên phản biện

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, Tp.HCM đã tạo điều kiện tốt cho chúng em thực hiện đề tài này.

Chúng em xin chân thành cảm ơn Thầy Trần Quang Tánh và Thầy Đặng Bình Phương đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian thực hiện đề tài.

Chúng em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu trong những năm học vừa qua.

Chúng em xin gửi lòng biết ơn sâu sắc đến Ba, Mẹ, các anh chị và bạn bè đã ủng hộ, giúp đỡ và động viên chúng em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Mặc dù chúng em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Nhóm thực hiện

ĐỀ CƯƠNG CHI TIẾT

Tên Đề Tài : Ứng dụng abc
Giáo viên hướng dẫn: ThS. Trần Quang Tánh – ThS. Đặng Bình Phương
Thời gian thực hiện : từ ngày aa/bb/2018 đến ngày aa/bb/2018
Sinh viên thực hiện : Lê Đình Hân (1412153) – Trần Phước Hiền (1412171)
Loại đề tài : Tìm hiểu công nghệ và xây dựng ứng dụng
Nội Dung Đề Tài : Mục tiêu của đề tài được xác định là ??? Nội dung chi tiết của luận văn bao gồm :

Kế Hoạch Thực Hiện: - aa/aa/2018-aa/aa/2018 -	
Xác nhận của GVHD	Ngày tháng 7 năm 2018 Nhóm SV Thực hiện Lê Đình Hân – Trần Phước Hiền

MỤC LỤC

LỜI CẢM ƠN	iii
ĐỀ CƯƠNG CHI TIẾT	iv
MỤC LỤC	vi
DANH MỤC CÁC HÌNH	ix
DANH MỤC MÃ	xi
DANH MỤC CÁC SƠ ĐỒ	xii
TÓM TẮT KHÓA LUẬN	xiii
Chương 1 Mở đầu	1
1.1. Giới thiệu chung	1
1.1.1. Tangible và môi trường tương tác Tangible	1
1.1.2. Các kỹ thuật tương tác trong môi trường hỗ trợ tương tác Tangibles	2
1.2. Lý do thực hiện đề tài	9
1.3. Mục tiêu đề tài	10
1.4. Nội dung luận văn.....	11
Chương 2 Các phương pháp nhận dạng vật thể	13
2.1. Tổng quan về vấn đề nhận dạng	13
2.2. Nhận dạng vật bằng ảnh màu	13
2.3. Áp dụng Vuforia trong việc nhận dạng	16
2.3.1. Giới thiệu	16
2.3.2. Các vấn đề kỹ thuật khi sử dụng Vuforia	17
2.4. Áp dụng Kinect trong việc nhận dạng	23
2.4.1. Giới thiệu Kinect	23
2.4.2. Cấu tạo Kinect v2	25

2.4.3.	Các vấn đề kỹ thuật khi sử dụng Kinect	27
2.5.	Kết luận.....	28
Chương 3	Các vấn đề kỹ thuật và giải pháp về đồ họa	29
3.1.	Mở đầu.....	29
3.2.	Các vấn đề kỹ thuật và giải pháp trong Unity	30
3.2.1.	Camera	30
3.2.2.	Material	31
3.2.3.	Collider	33
3.2.4.	Particle system	34
3.2.5.	Raycasting.....	35
3.2.6.	Lighting.....	36
3.2.7.	Position	38
3.3.	Kết luận.....	39
Chương 4	Hệ thống bản đồ địa lý hỗ trợ Tangible	40
4.1.	Giới thiệu tổng quan	40
4.2.	Cấu hình hệ thống.....	41
4.3.	Kịch bản sử dụng hệ thống	43
4.4.	Một số vấn đề gặp phải và kỹ thuật áp dụng	43
4.4.1.	Cách bố trí máy chiếu	43
4.4.2.	Kỹ thuật ánh xạ tọa độ giữa các không gian tọa độ.....	47
4.4.3.	Kiến trúc hệ thống sự kiện liên hoàn	49
4.4.4.	Giao tiếp giữa các thiết bị tương tác.....	54
4.4.5.	Xử lý hành động tương tác từ phía người dùng.....	56
4.5.	Các chức năng của hệ thống	57

4.5.1.	Xem hình ảnh tại một địa điểm	58
4.5.2.	Tương tác với dữ liệu theo thời gian thực	59
4.5.3.	Thống kê dữ liệu	59
4.6.	Kết luận.....	60
Chương 5 Hệ thống chia sẻ dữ liệu hỗ trợ tangible		61
5.1.	Giới thiệu tổng quan	61
5.2.	Cấu hình hệ thống.....	61
5.3.	Kịch bản sử dụng hệ thống	61
5.4.	Một số vấn đề gặp phải và kỹ thuật áp dụng	62
5.4.1.	Cách bố trí máy chiếu	62
5.4.2.	Xử lý hành động tương tác từ phía người dùng.....	62
5.4.3.	Giao tiếp giữa các thiết bị tương tác	63
5.5.	Các chức năng của hệ thống	64
5.5.1.	Các cách thức tương tác.....	64
5.5.2.	Màn hình giao diện chính	65
5.6.	Kết luận.....	65
Chương 6 Kết luận		67
6.1.	Các kết quả đạt được	67
6.2.	Hướng phát triển của đề tài	68
TÀI LIỆU THAM KHẢO		70

DANH MỤC CÁC HÌNH

Hình 1.1 Du khách ở Lawrence Hall of Science khám phá các khái niệm dòng chảy nước [9].....	3
Hình 1.2 Mô hình hồ cát render màu theo độ cao	3
Hình 1.3 Đối tượng chiếu màu trắng (trái), Đối tượng được chiếu hoa văn phụ thuộc vào vị trí và góc nhìn của người xem (giữa, phải) [11].....	5
Hình 1.4 Chiếu phối cảnh lên đĩa thức ăn	5
Hình 1.5 Hành động Flip	6
Hình 1.6 Hành động Shake	7
Hình 1.7 Hành động Erect	7
Hình 1.8 Hành động Shake Down	7
Hình 1.9 Hành động Shake Up	8
Hình 2.1 Kết quả các bước trong quy trình nhận dạng vật thể.....	15
Hình 2.2 Ví dụ về toán tử erosion.....	16
Hình 2.3 Đăng kí tài khoản Vuforia	17
Hình 2.4 Cho phép sử dụng Vuforia trong project	18
Hình 2.5 Giao diện lấy App Key	19
Hình 2.6 Gắn App Key vào project	19
Hình 2.7 Tạo Database hình ảnh nhận diện.....	20
Hình 2.8 Tải Database về project	20
Hình 2.9 Chọn Target theo hình ảnh từ Database	20
Hình 2.10 Dựa vào độ tương phản	21
Hình 2.11 Độ tương phản ảnh Grayscale tương ứng với histogram.....	21
Hình 2.12 Cấu tạo Kinect v2	24
Hình 2.13 Image stream Kinectv2	26

Hình 2.14 Depth stream kinectv2	26
Hình 2.15 Body stream Kinectv2	27
Hình 3.1 Giao diện Unity Engine	29
Hình 3.2 Thông số cơ bản của Camera.....	31
Hình 3.3 Sử dụng Material để lọc màu cho Object	33
Hình 3.4 Colider với Object Cube	34
Hình 3.5 Minh họa Raycasting trong Unity	36
Hình 3.6 Các thành phần cơ bản của Light Component.....	37
Hình 3.7 Các loại Light trong Unity.....	37
Hình 3.8 Mô phỏng cách sử dụng position trong hệ thống	39
Hình 4.1 Cấu hình thiết lập của hệ thống	41
Hình 4.2 Hình ảnh thực tế của hệ thống	42
Hình 4.3 Cách bố trí máy chiếu trên cao	44
Hình 4.4 Cách bố trí máy chiếu dưới đất theo phương vuông góc.....	45
Hình 4.5 Cách bố trí máy chiếu dưới đất kém theo gương	46
Hình 4.6 Xác định vùng thao tác của hệ thống.....	47
Hình 4.7 Lấy mẫu Calibration theo 2 trục X, Y	48
Hình 4.8 Hệ tọa độ của hệ thống cảm biến trên thiết bị di động	56
Hình 4.9 Xem chi tiết hình ảnh bản đồ tại một địa điểm.....	58
Hình 4.10 Ảnh Heat map thể hiện mức độ ô nhiễm không khí.....	59
Hình 4.11 Thống kê thông tin theo tập các địa điểm đã chọn	60

DANH MỤC MÃ

Mã 1 Minh họa một Comparator	52
Mã 2 Minh họa một Action	53
Mã 3 Minh họa một Operator	54

DANH MỤC CÁC SƠ ĐỒ

Sơ đồ 2-1 Quy trình nhận dạng vật thể bằng xử lý ảnh cơ bản	14
Sơ đồ 3-1 Cơ chế kết xuất đồ họa trong Unity	32
Sơ đồ 4-1 Hệ thống sự kiện	50
Sơ đồ 4-2 Hệ thống sự kiện liên hoàn	51
Sơ đồ 4-3 Sơ đồ lớp của Comparator	52
Sơ đồ 4-4 Sơ đồ lớp của Action.....	53
Sơ đồ 4-5 Sơ đồ lớp của Operator	54
Sơ đồ 4-6 Cấu trúc module của hệ thống	55
Sơ đồ 5-1 Sơ đồ kiến trúc hệ thống	63

TÓM TẮT KHÓA LUẬN

Môi trường tương tác hỗ trợ Tangible là môi trường cho phép người dùng có thể tương tác với các thông tin kỹ thuật số mà có sự hỗ trợ của yếu tố vật lý. Đã có nhiều nhóm nghiên cứu trên thế giới ứng dụng tương tác Tangible vào các lĩnh vực trong cuộc sống như giảng dạy, thuyết trình hay trong các cuộc trao đổi, thảo luận nhóm,... Điều đó cho thấy tiềm năng của ứng dụng Tangible vào cuộc sống là rất lớn, có thể mở ra nhiều xu thế mới trong tương lai.

Với điều kiện kinh tế xã hội đang phát triển, sự phát triển của lĩnh vực Công nghệ Thông tin đang được đẩy mạnh, nhiều ứng dụng khoa học công nghệ đã được áp dụng vào trong đời sống và tạo được nhiều thay đổi rõ rệt, nâng cao chất lượng cuộc sống của mọi người. Mọi người có thể tiếp cận dễ dàng hơn với các công nghệ mới, chẳng hạn như tương tác hỗ trợ Tangible. Tiềm năng to lớn và khả năng ứng dụng thực tiễn cao của hướng nghiên cứu này đã tạo động lực cho chúng em thực hiện luận văn này.

Trong nội dung luận văn này, chúng em tập trung vào việc khảo sát, nghiên cứu và thực nghiệm các kỹ thuật tương tác khác nhau trong môi trường hỗ trợ Tangible. Cụ thể, chúng em sử dụng Kinect kết hợp với máy chiếu để tạo nên không gian tương tác, cung cấp cho người dùng người một loạt các cử chỉ, thao tác để có thể giao tiếp với máy tính. Ngoài ra, chúng em dùng Unity 3D Engine làm môi trường để phát triển phần mềm hệ thống.

Theo như kết quả ban đầu đạt được, chúng em đã xây dựng 2 hệ thống để chứng minh tính thực tiễn của các cách tương tác là hệ thống bản đồ hỗ trợ tương tác Tangible và hệ thống chia sẻ dữ liệu hỗ trợ tương tác Tangible. Hai hệ thống mà chúng em phát triển có thể áp dụng linh hoạt trong các lĩnh vực khác nhau như địa lý, xã hội, sinh học, lịch sử hay hỗ trợ cộng tác, làm việc nhóm với nhau,.. Ngoài sự hệ thống còn có khả năng nâng cấp, thay đổi linh hoạt về mặt phần cứng, để có thể triển khai được trong nhiều môi trường khác nhau.

Chương 1

Mở đầu

✍ Nội dung Chương 1 trình bày tổng quan về đề tài, lý do thực hiện đề tài, từ đó nêu rõ mục tiêu của khóa luận. Nội dung tóm tắt của từng chương trong khóa luận được trình bày ở cuối chương này.

1.1. Giới thiệu chung

1.1.1. *Tangible và môi trường tương tác Tangible*

Tangible User Interface (TUI) là giao diện người dùng giúp con người có thể tương tác với các thông tin kỹ thuật số thông qua môi trường vật lý. Mục đích của sự phát triển TUI là để tăng cường tính cộng tác, học hỏi, thiết kế bằng cách vật lý hóa những thông tin kỹ thuật số ví dụ như cầm nắm đồ vật, vật liệu [1].

Tương Tác Tangible (Tangible Interaction) là một cụm từ nói về mối quan hệ giữa giao diện người dùng và cách mà người dùng tương tác với giao diện đó. Hornecker và Burr [2] đã đề ra bốn tiêu chí để xây dựng một kỹ thuật tương tác: sự chân thật, cách thể hiện thông tin, hành vi tương tác của cơ thể và cách thể hiện của kiểu tương tác đó trong thế giới thực. Một số ví dụ về các cách tương tác như Scott R. Klemmer và các cộng sự của ông [3] đã kết hợp giấy note và bảng chiếu để xây dựng bảng làm việc nhóm, nhóm MIT Media Lab [3] đã dùng nút vặn như 1 token vật lý để truy xuất cơ sở dữ liệu, thay đổi thông tin hiển thị.

Một trong những nhóm người đi đầu trong việc nghiên cứu về thiết bị tangible và tương tác tangible là MIT Media Lab – Tangible Media Group. MIT Media Lab được thành lập bởi Nicholas Negroponte và Jerome Wiesner vào năm 1985 nhằm thúc đẩy nghiên cứu cách sử dụng mới mẻ công nghệ máy tính. Họ bắt đầu từ những hệ thống đơn giản từ cuối thế kỉ 20 như hệ thống PingPongPlus – một hệ thống số hóa trò chơi Ping Pong cổ điển của Hiroshi Ishii và các cộng sự [4], các tác giả đã dùng một mặt bàn tương tác có sự kết hợp của hệ thống cảm ứng va chạm được tích hợp vào bàn và công nghệ chiếu ảnh. Ngày nay, các thiết bị Tangible ngày càng phát triển với nhiều kỹ thuật mới hơn được tích hợp vào như: giao diện Tabletop tangible sử dụng nền

tảng theo dõi đối tượng (object tracking platform) - được đề xuất từ hệ thống Sensetable của James Patten và cộng sự [5] có thể theo dõi vị trí và hướng của nhiều vật có kết nối không dây trên một mặt bàn hiển thị, hay cho thiết bị điều khiển âm nhạc điện tử của nhóm Ben Recht [6]. Thêm vào đó, với sự gia tăng băng thông rộng và kết nối Internet tốc độ cao cũng hỗ trợ cho việc mở rộng các hệ thống và thiết bị hỗ trợ Tangible. Anthony DeVincenzi và các cộng sự [7] đã tạo ra hệ thống Kinected Conference có thể tăng cường hiệu quả giao tiếp trong vấn đề Hội thoại Video (Video Conference) bằng cách tạo ra các hình ảnh ba chiều giàu ngữ nghĩa chứa thông tin về thành phần vật chất và vị trí của chúng. Sự phát triển trong khoa học vật liệu cũng là một tiền đề để nâng cao khả năng của các thiết bị Tangible, dự án PneuUI của Lining Yao [8] giới thiệu công nghệ cho phép xây dựng những thiết bị mà bề mặt tương tác có thể thay đổi hình dáng bằng các vật liệu mềm được nén khí.

Sự tiên phong, đột phá trong các nghiên cứu của MIT Media Lab không chỉ về Tangible Interfaces, Wearable Computing, Affective Computing mà còn trải rộng từ việc áp dụng cách tiếp cận kỹ thuật số trong việc chữa các bệnh rối loạn thần kinh đến giao thông vận tải hiệu quả tiết kiệm năng lượng. Điều này đã khơi gợi nguồn cảm hứng cho chúng em muốn được nghiên cứu, thí nghiệm về một trong những hướng đi của MIT Media Lab để đóng góp cho việc cải thiện cách mà con người sống, học tập, làm việc và giải trí. Các kỹ thuật tương tác trong môi trường hỗ trợ tương tác Tangible

1.1.2. Các kỹ thuật tương tác trong môi trường hỗ trợ tương tác Tangibles

❖ Tương tác trực tiếp vào môi trường

Đối với kỹ thuật tương tác trực tiếp vào môi trường, người dùng trực tiếp tiếp xúc, chạm vào môi trường làm biến đổi môi trường. Sự biến đổi của môi trường làm kéo theo sự biến đổi của thông tin được thể hiện tạo ra tương tác giữa người dùng và hệ thống.

Kỹ thuật tương tác này tạo nên trải nghiệm người dùng rất tốt, vì người dùng nhận thức được sự thay đổi thông tin dữ liệu do chính tác động của mình vào môi trường. Kỹ thuật này đã được một số nhóm nghiên cứu trên thế giới ứng dụng vào việc biểu

diễn thông tin và thu được kết quả rất tích cực. Oliver Kreylo cùng các cộng sự đã thực hiện dự án Sandbox [9], dự án gồm 1 hồ cát để tương tác và 1 máy chiếu phía trên để chiếu lên hình ảnh có các sắc thái màu sắc tương ứng với độ cao của bề mặt cát trong hồ. Dự án nào có thể được ứng dụng vào lĩnh vực địa lý để dạy học, chẳng hạn như minh họa thay đổi độ cao của bề mặt trái đất, hoặc sự chuyển động của nước khi chảy từ nơi cao xuống nơi thấp hơn



Hình 1.1 Du khách ở Lawrence Hall of Science khám phá các khái niệm dòng chảy nước [9]

Để khảo sát và đánh giá độ hiệu quả, chúng em đã lấy cảm hứng ý tưởng của project Sandbox và thử nghiệm thực tế. Sau quá trình nghiên cứu và phát triển, nhóm chúng em đã hiện thực hóa được ý tưởng ban đầu.



Hình 1.2 Mô hình hồ cát render màu theo độ cao

Nhóm chúng em đã dùng Kinect để lấy bản đồ độ sâu của hồ cát, từ đó chuyển giá trị độ sâu thành giá trị màu theo từng ngưỡng độ cao khác nhau. Qua quá trình ánh

xạ tọa độ để căn chỉnh so khớp máy chiếu và Kinect, chúng em thu được kết quả như Hình 1.2. Theo đánh giá và nhận xét, kỹ thuật tương tác này mang lại trải nghiệm người dùng rất tốt, khiến người dùng thích thú, tìm tòi và khám phá. Tuy nhiên, hạn chế của cách làm này cần chuẩn bị nhiều thiết bị, vật liệu. Ngoài ra mô hình này chỉ phù hợp cho một số lĩnh vực như địa lý, khó có thể áp dụng sang nhiều lĩnh vực khác như toán học, sinh học,...

❖ **Tương tác một phần vào môi trường để kích hoạt sự kiện**

Kỹ thuật chính trong cách tương tác này là kỹ thuật trình chiếu so khớp với vật thể. Trong môi trường tương tác, một số vật thể thật được đặt làm các vị trí mốc. Một hoặc nhiều máy chiếu khác nhau sẽ chiếu hình ảnh lên các vật thể sao cho hình ảnh được chiếu trùng khớp với các vật thể tạo nên kịch bản tương tác với người dùng. Người dùng sẽ sử dụng các thao tác được quy định sẵn để tương tác, các thao tác này sẽ làm điều kiện để kích hoạt các sự kiện trong kịch bản trình chiếu.

Kỹ thuật trình chiếu trùng khớp với vật thể (3D projection mapping) là sự kết hợp của việc trình chiếu video và thực tại tăng cường trong không gian [10]. Vật thể để hứng ảnh chiếu thường có hình dạng đơn giản, có thể cong hoặc gấp khúc. Các kỹ thuật quang học như phản xạ gương, thấu kính được áp dụng để tạo nên các ảnh chiếu khác nhau. Ngoài ra, các kỹ xảo hình ảnh, âm thanh được kết hợp trong quá trình trình chiếu giúp tạo nên nhiều hiệu ứng hấp dẫn thu hút người dùng.

Ramesh Raskar và các cộng sự của ông đã dùng kỹ thuật này để thực hiện dự án “Shader Lamp” [11]. Ông dùng 1 chiếc bình để hứng ảnh chiếu và tạo hoa văn cho chiếc bình bằng cách thay đổi nhiều hình chiếu khác nhau (Hình 1.3). Karl D.D. Willis và các cộng sự áp dụng kỹ thuật này với dự án “Hide Out”. Dự án dùng một máy chiếu di động kết hợp các marker được đánh dấu mực hồng ngoại để trình chiếu kịch bản dựa theo các marker đó.



Hình 1.3 Đối tượng chiếu màu trắng (trái), Đối tượng được chiếu hoa văn phụ thuộc vào vị trí và góc nhìn của người xem (giữa, phải) [11]

Chúng em đã thực nghiệm thử kỹ thuật này, vật thể mốc chúng em dùng một chiếc đĩa thức ăn và lắp đặt máy chiếu chiếu trực diện phía trên. Hình ảnh chiếu ra được chiếu nghiêng một góc 80° , theo như thử nghiệm, chúng em nhận thấy góc chiếu này mang lại cảm giác hình ảnh 3D chiếu ra là chân thật nhất (Hình 1.4). Tuy nhiên chúng em còn gặp khó khăn là để hình ảnh chiếu ra trùng với chiếc đĩa, chúng em vẫn phải thực hiện thủ công căn chỉnh mà chưa tự động hóa được công đoạn này.



Hình 1.4 Chiếu phối cảnh lên đĩa thức ăn

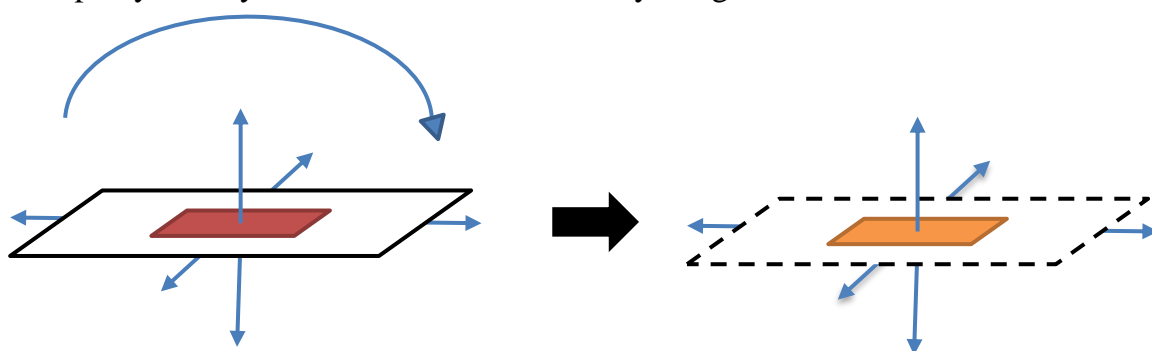
❖ **Tương tác thông qua thiết bị di động**

Với kỹ thuật tương tác này, chúng em dùng một thiết bị di động để hỗ trợ cho quá trình tương tác bằng cách cung cấp các hình thức tương tác (các cử chỉ tương tác, tương tác thông qua giao diện chương trình). Để tương tác với môi trường Tangible

thì việc nhận ra các cử chỉ của người sử dụng để kích hoạt các sự kiện cũng như là các chức năng của hệ thống là rất quan trọng. Thông thường, ta có thể sử dụng việc nhận diện cử chỉ thông qua hình ảnh bằng cách sử dụng camera thu hình và nhận diện hành động đang được thực hiện (như Kinect đã có hỗ trợ chức năng này). Nhưng việc nhận diện thông qua hình ảnh lại có nhiều hạn điểm về độ nhiễu như chất lượng của hình ảnh thu được qua camera, vị trí của người sử dụng đối với camera, các cử chỉ gần giống với nhau, tốc độ thực hiện các cử chỉ. Tất cả những hạn chế trên sẽ ảnh hưởng trực tiếp đến việc nhận diện tương tác và xử lý tương tác, từ đó mang đến trải nghiệm không thoải mái cho người sử dụng. Để giải quyết những hạn chế trên đồng thời với mục đích chia nhỏ xử lý, không dồn tất cả vào việc xử lý hình ảnh, chúng em sử dụng một thiết bị di động để mô phỏng một số cử chỉ đơn giản sẽ được trình bày ở phần tiếp theo. Với sự hỗ trợ về phần cứng và phần mềm của thiết bị di động, chúng em vừa có thể cung cấp cách tương tác truyền thống là tương tác qua giao diện chương trình được vừa có thể hỗ trợ một số hành động tự nhiên để tăng cường hiệu quả về cảm giác sử dụng.

Các cử chỉ được mô phỏng :

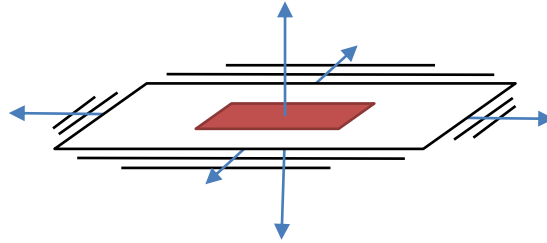
- Lật (Flip) : là hành động thay đổi hướng của mặt thiết bị di động từ hướng ngửa lên đến úp xuống trên một mặt phẳng nằm ngang như Hình 1.5 Hành động Flip. Chúng em mô phỏng hành động này tượng trưng cho việc thay đổi một lựa chọn trong tập các lựa chọn được cho trước. Các lựa chọn trong tập này sẽ thay đổi theo thứ tự tuần tự xoay vòng.



Hình 1.5 Hành động Flip

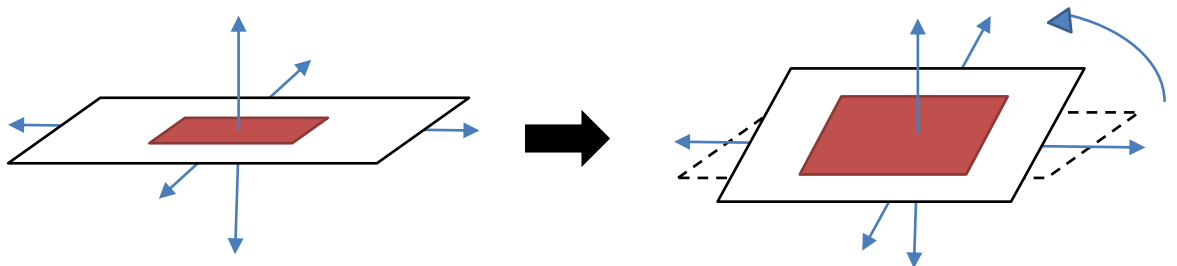
- Lắc (Shake) : là hành động thay đổi vị trí của thiết bị quanh vị trí ban đầu theo 2 hướng nghịch nhau trong thời gian ngắn khi thiết bị được đặt nằm

theo mặt phẳng ngang như hình Hình 1.6 Hành động Shake. Chúng em mô phỏng hành động này tượng trưng cho việc xóa bỏ một trạng thái đang được áp dụng lên thiết bị để trở về bình thường.



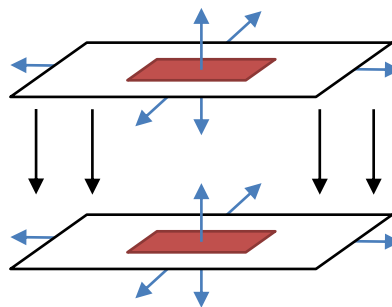
Hình 1.6 Hành động Shake

- Dựng đứng (Erect) : là hành động thay đổi tư thế của thiết bị từ đang nằm theo mặt phẳng ngang đến đứng lên theo mặt phẳng đứng 90^0 như hình Hình 1.7. Chúng em mô phỏng hành động này tượng trưng cho việc xem một thông tin chi tiết trên thiết bị mà người dùng đặc biệt quan tâm, thông tin này không xuất hiện thường xuyên.



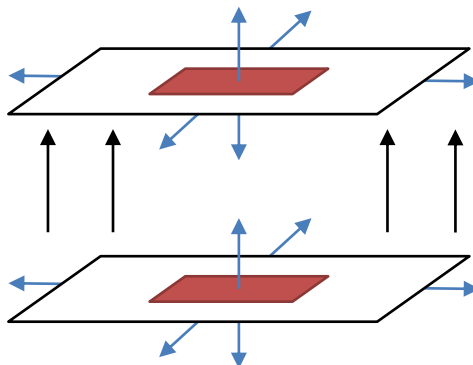
Hình 1.7 Hành động Erect

- Giữ xuống (Shake down) : là hành động thay đổi vị trí của thiết bị một cách đột ngột từ vị trí ban đầu theo hướng thẳng đứng từ trên xuống dưới như Hình 1.8. Chúng em mô phỏng hành động này tượng trưng cho việc gửi đi một thông tin từ thiết bị đến một môi trường khác.



Hình 1.8 Hành động Shake Down

- Hắt lên (Shake Up) : là hành động thay đổi vị trí của thiết bị một cách đột ngột từ vị trí ban đầu theo hướng thẳng đứng từ dưới lên trên như Hình 1.9. Chúng em mô phỏng hành động này tượng trưng cho việc lấy một thông tin từ môi trường bên ngoài vào trong thiết bị của người sử dụng.



Hình 1.9 Hành động Shake Up

❖ **Kết luận**

So sánh kỹ thuật tương tác trực tiếp vào môi trường, kỹ thuật tương tác một phần vào môi trường và tương tác hỗ trợ Tangible với sự trợ giúp của thiết bị di động, chúng em thấy được những hạn chế của hai phương pháp đầu tiên. Ở kỹ thuật đầu tiên, ta cần phải tương tác trực tiếp đến môi trường tương tác và những vật trong môi trường tương tác. Việc xử lý nhận diện các cử chỉ hay hành động tương tác sẽ thiên về xử lý hình ảnh trên camera, như đã nhắc đến những hạn điểm của việc xử lý cử chỉ thông qua hình ảnh ở mục **Tangible với thiết bị di động**, chính điều này mang đến trải nghiệm không thoải mái cho người dùng. Với kỹ thuật thứ hai, thì môi trường tương tác và vật tương tác đóng góp vai trò trong việc vật lý hóa thông tin được trình chiếu theo kịch bản cho trước. Đây là phương pháp tăng cường cảm giác thật của người dùng với một đoạn video cho trước. Ở đây thiếu sự tương tác từ phía người dùng đến hệ thống, chính vì vậy cũng không mang lại trải nghiệm toàn vẹn cho phía người dùng. Vì thế, chúng em quyết định sẽ đi sâu vào phương pháp thứ ba, sử dụng một thiết bị di động để thu thập thông tin về hành động của người dùng sau đó giao tiếp đến hệ thống máy chủ để máy chủ xử lý. Việc sử dụng thiết bị di động tuy có hạn chế về độ đa dạng các cử chỉ và hành động nhận được tuy nhiên sẽ cho ra kết quả chính xác hơn, và tốc độ hồi đáp nhanh hơn so với việc nhận diện cử chỉ qua hình ảnh. Điều này giảm bớt sự nặng nề của hệ thống trong việc xử lý cũng như nhận diện các cử chỉ, hành động. Máy chủ sẽ nhận tín hiệu truyền về từ thiết bị và chỉ quan tâm

đến quản lí kịch bản sự kiện. Cuối cùng sử dụng một thiết bị hỗ trợ sẽ giải quyết vấn đề về sự thiếu tương tác từ phía người dùng như phương pháp thứ hai.

1.2. Lý do thực hiện đề tài

Từ năm 2015, Sở Giáo dục và Đào tạo đã đưa chương trình giáo dục STEM vào giảng dạy thí điểm ở một số địa bàn thành phố, từ các trường tiểu học cho đến các trường cấp 3. Qua thời gian thí điểm và thử nghiệm, chương trình giáo dục STEM đã chứng minh được tính ưu việt và hiệu quả của mình. STEM (viết tắt tiếng Anh của Science - Khoa học, Technology - Công nghệ, Engineering - Kỹ thuật và Math – Toán học) là phương pháp giảng dạy môn khoa học ứng dụng hướng đến liên kết các môn học, mang tính thực tiễn cao [12]. Mục tiêu phương pháp này là biến việc học các môn khoa học cơ bản rời rạc trở nên thú vị, sống động hơn với học sinh. Nhiều doanh nghiệp đã chủ động đầu tư các dự án giáo dục ứng dụng công nghệ ở nước ta và đạt được nhiều kết quả tích cực, có thể kể đến như Samsung với các dự án như Dự án Thư viện thông minh, Dự án phòng học thông minh, hay các trại hè như Robotics Camp do đại sứ quán Mỹ bảo trợ.

Hiểu được tiềm năng to lớn của các việc áp dụng kỹ thuật công nghệ cao để áp dụng vào phương pháp dạy học, nhóm chúng em muốn phát triển các hệ thống hỗ trợ Tangible dễ triển khai và giá thành rẻ. Các hệ thống hỗ trợ tương tác Tangible giúp trực quan hóa nội dung, tạo cảm hứng cho người học, khơi dậy tìm đam mê khoa học, kích thích người dùng tìm tòi, khám phá.

Việc ứng dụng tương tác Tangible vào lĩnh vực giáo dục đã được nhiều nhóm nghiên cứu trên thế giới đề xuất. Điều này chứng minh hướng tiếp cận này là khả thi và có tiềm năng. Sự thành công của các nhóm nghiên cứu trước đã tạo động lực và cảm hứng để chúng em thực hiện đề tài này.

Trong hệ thống hỗ trợ Tangible mà chúng em phát triển, chúng em dùng 1 camera độ sâu, cụ thể là Kinect v2 kết hợp với 1 máy chiếu để tạo nên không gian tương tác hỗ trợ Tangible. Chi phí cho hệ thống này khi triển khai thực tế không quá cao và có thể lắp đặt nhanh cho tính nhỏ gọn, cấu trúc đơn giản. Hệ thống có thể đặt trong các phòng học, cho phép nhiều người cùng quan sát và thảo luận với nhau. Ngoài ra, hệ thống còn có khả năng nâng cấp, thay vì sử dụng máy chiếu để hiển thị ảnh thì ta có

thể thay bằng bất kì màn hình hiển thị nào khác, điều này mang đến trải nghiệm hình ảnh tốt hơn nhưng kèm theo đó là chi phí triển khai sẽ tốn kém hơn.

1.3. Mục tiêu đề tài

Chúng em thực hiện luận văn này với mục tiêu là **khảo sát, nghiên cứu và phát triển** 1 hệ thống có hỗ trợ tương tác Tangible, triển khai, **nghiên cứu và đề xuất thêm** các thao tác, cử chỉ hỗ trợ quá trình tương tác. Từ đó phát triển và thử nghiệm một số hệ thống ứng dụng để **chứng minh tính khả thi** của việc áp dụng Tangible vào thực tiễn.

Để đạt được mục tiêu trên, chúng em đề ra các nội dung cần thực hiện như sau:

- *Khảo sát vấn đề:*

- Tìm hiểu hướng tiếp cận về Tangible của các nhóm nghiên cứu trên thế giới
- Tìm hiểu các kiểu tương tác giữa người và máy, các cách biểu diễn thông tin ảo trong môi trường thực tế
- Tìm hiểu cách khả năng, cách tích hợp và sử dụng Kinect v2
- Tìm hiểu các công nghệ về đồ họa 3D, xử lý và nhận dạng
- Tìm hiểu các kỹ thuật, khả năng hỗ trợ lập trình trong Unity3D

- *Đề xuất giải pháp và tiến hành thực nghiệm:*

- Đề xuất các thao tác tương tác có thể ứng dụng được, chia ra thành 3 nhóm tương tác chính: tương tác trực tiếp vào môi trường, tương tác 1 phần vào môi trường để kích hoạt sự kiện và tương tác thông qua thiết bị trung gian, cụ thể là điện thoại thông minh.
- Tiến hành thực nghiệm đối với từng loại tương tác nêu trên để phân tích, đánh giá và đưa ra kết luận.
- Đề xuất và đánh giá các phương pháp nhận dạng vật thể và theo vết vật thể dựa vào ảnh màu, ảnh hồng ngoại và ảnh độ sâu thu được từ Kinect.
- Đề xuất cách thức giao tiếp giữa các module trong một hệ thống.
- Đề xuất xây dựng kiến trúc linh hoạt để có thể mở rộng, thay đổi nhanh hệ thống, đáp ứng các nhu cầu đa dạng, phức tạp trong thực tế.
- Đề xuất cấu trúc hệ thống của môi trường tương tác hỗ trợ Tangible, cách bố trí các phần cứng trong hệ thống sao cho hiệu quả, tối ưu.

- *Dựa trên những khảo sát và đề xuất trên, chúng em sẽ triển thực tế 1 thí nghiệm và 2 hệ thống để chứng minh tính thực tiễn của đề tài:*
 - Thí nghiệm hồ cát hỗ trợ tương tác Tangible: Đây là thí nghiệm chúng em sẽ thực nghiệm trong quá trình khảo sát độ hiệu quả của kỹ thuật tương tác trực tiếp vào môi trường để làm biến đổi thông tin. Thí nghiệm gồm một hộp chứa cát trắng và máy chiếu, kinect lắp phía trên, cho phép người dùng dùng tay để tạo hình cho cát, sau đó máy chiếu phía trên sẽ hiển thị hình ảnh phân tầng độ cao tương ứng với độ cao thực tế của bề mặt hồ cát. Thí nghiệm này có thể áp dụng vào các môn học địa lý và kích thích sự tìm tòi, sáng tạo của người học
 - Hệ thống bản đồ hỗ trợ tương tác Tangible: Hệ thống hỗ trợ người dùng xem các thông tin trên bản đồ, tương tác thông qua thiết bị ngoại vi như điện thoại thông minh. Hệ thống cung cấp thông tin toàn diện và tổng quan của một khu vực địa lý, ngoài ra cung cấp cho người dùng các thông tin chi tiết tại một vị trí cụ thể thông qua các hành vi tương tác khác nhau để truy vấn. Hệ thống thích hợp để phục vụ cho các cuộc thảo luận, thuyết trình cần truyền tải thông tin một cách trực quan, sinh động. Ngoài bản đồ địa lý, hệ thống còn có khả năng triển khai trên các lĩnh vực khác nhau như sinh học, lịch sử,.. cho nên tiềm năng của hệ thống là rất cao.
 - Hệ thống chia sẻ dữ liệu hỗ trợ tương tác Tangible: Hệ thống cho phép người dùng chia sẻ dữ liệu như hình ảnh, tài liệu, ghi chú một cách trực quan. Hệ thống đóng vai trò là một môi trường lưu trữ vào cung cấp dữ liệu trực quan. Dữ liệu cần chia sẻ được hiển thị trực tiếp, người dùng có thể truy cập vào để chia sẻ dữ liệu hoặc lấy dữ liệu mà mình muốn có. Thông qua các cử chỉ, thao tác rất quen thuộc trong cuộc sống như hát lên, giữ xuống, người dùng có thể gửi hoặc nhận dữ liệu.

1.4. Nội dung luận văn

Luận văn bao gồm 6 chương, sau đây là nội dung chính của từng chương.

Chương 1: Giới thiệu định nghĩa về Tangible và các kỹ thuật tương tác trong môi trường hỗ trợ Tangible. Khảo sát các cách tiếp cận khi xây dựng môi trường hỗ trợ Tangible của các nhóm nghiên cứu khác trên thế giới. Tiếp theo đó, chúng em nêu lý

do và động lực để thực hiện đề tài, và đề ra các mục tiêu cần thực hiện được trong đề tài này.

Error! Reference source not found.: trình bày về các đề xuất để nhận dạng vật thể và theo vết vật thể. Cụ thể là dùng các phương pháp xử lý ảnh để nhận dạng theo ảnh màu, ảnh hồng ngoại và ảnh độ sâu. Bên cạnh đó, chúng em còn khảo sát các framework hỗ trợ nhận dạng như Vuforia. Ngoài ra, chúng em trình bày tổng quan về lịch sử, cấu trúc phần cứng của Kinect và cách tích hợp Kinect vào hệ thống, sử dụng các thư viện hỗ trợ lập trình như Kinect SDK.

Chương 3: trình bày các vấn đề kỹ thuật trong Unity, lý do cần tìm hiểu các kỹ thuật đó và cách sử dụng trong thực tế như thế nào.


Chương 4: trình bày kiến trúc hệ thống bản đồ hỗ trợ tương tác Tangible, kỹ thuật ánh xạ tọa độ trong hệ thống và các chức năng nổi bật.

Chương 5: trình bày kiến trúc hệ thống chia sẻ dữ liệu hỗ trợ tương tác Tangible và các chức năng nổi bật.

Chương 6Error! Reference source not found.: trình bày các kết quả đã đạt được trong quá trình thực hiện đề tài và hướng phát triển trong tương lai.

Chương 2

Các phương pháp nhận dạng vật thể

 Nội dung *Error! Reference source not found.* đề xuất, phân tích và thực nghiệm các giải pháp để nhận diện vật thể. Từ đó nêu lên ưu điểm, hạn chế của từng phương pháp và áp dụng các phương pháp này để xây dựng một môi trường tương tác hỗ trợ Tangible

2.1. Tổng quan về vấn đề nhận dạng



Đặt vấn đề

Việc nhận dạng chính xác vật thể đóng vai trò quan trọng trong việc xây dựng môi trường tương tác có hỗ trợ Tangible. Nhận dạng là công cụ gián tiếp để các hành vi, thao tác của người dùng giao tiếp được với hệ thống. Làm sao để xác định sự xuất hiện của một vật ? Làm sao để truy vết được sự thay đổi vị trí của vật đó ?



Giải pháp

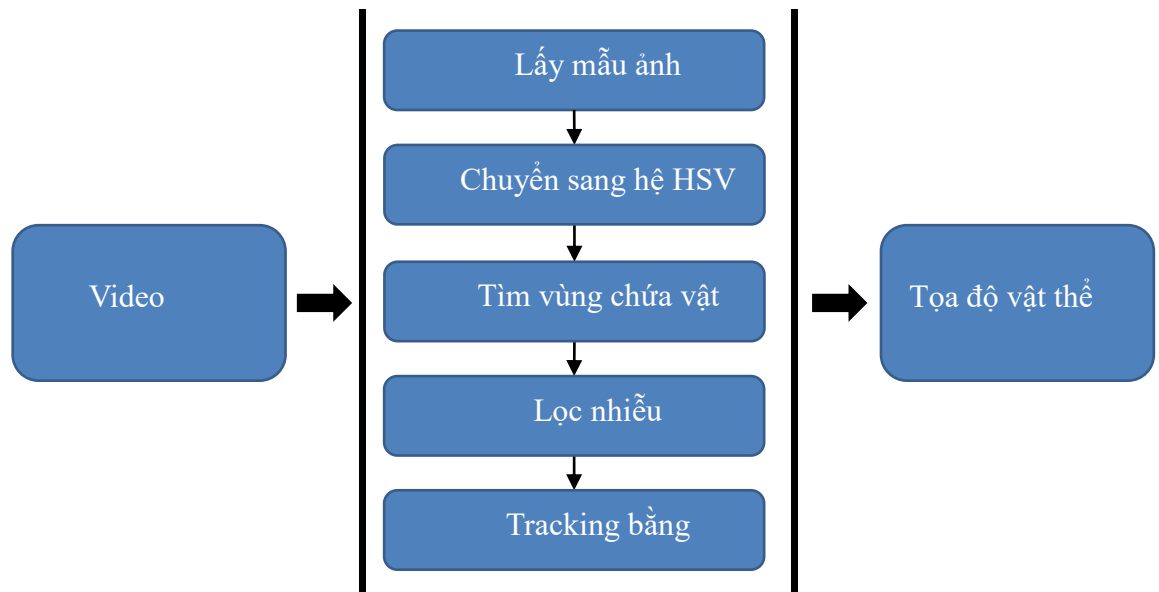
Để trả lời cho các câu hỏi trên, chúng em xin đề xuất ra các phương pháp tiếp cận dưới đây, từ đó tiến thành thực nghiệm, phân tích, đánh giá và đưa ra kết luận phương pháp phù hợp nhất:

- Nhận dạng bằng màu đơn sắc (Chương 2.2)
- Nhận dạng bằng họa tiết ảnh áp dụng Vuforia (Chương 2.3)
- Nhận dạng bằng ảnh hồng ngoại kết hợp thông tin độ sâu (Chương 2.4)

2.2. Nhận dạng vật bằng ảnh màu

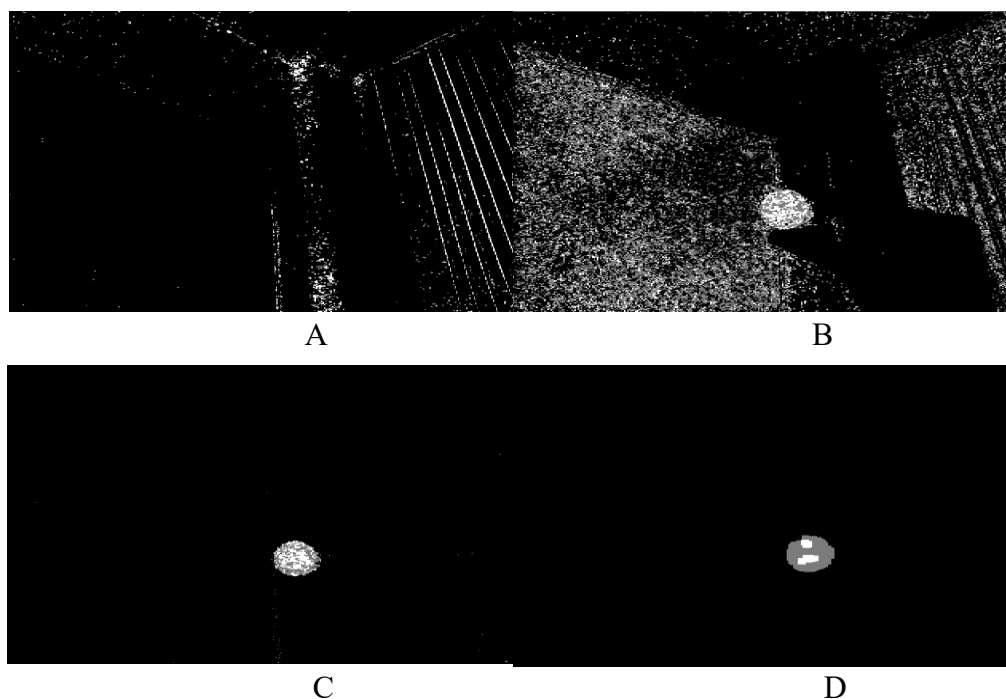
Để có thể triển khai nhanh ứng dụng với chi phí rẻ, chúng em quyết định dùng một camera màu hỗ trợ Full HD (webcam) hiệu để thực hiện lấy ảnh màu cho việc nhận dạng.

Chúng em áp dụng các phương pháp xử lý ảnh cơ bản để nhận dạng vật theo màu sắc theo quy trình sau:



Sơ đồ 2-1 Quy trình nhận dạng vật thể bằng xử lý ảnh cơ bản

Sơ đồ 2-1 mô tả quy trình nhận dạng một vật thể theo kênh màu. Bước đầu tiên ta sẽ lấy vùng ảnh chứa vật thể cần nhận dạng, sau đó chuyển ảnh sang hệ màu HSV rồi tiến hành tính toán biểu đồ Histogram của vùng màu được chọn. Ta bắt đầu tìm vùng ảnh chứa vật thể cần nhận dạng. Dựa vào biểu đồ histogram, ta áp dụng phương pháp Back Projection để lấy được ảnh xác suất màu sắc có thể xuất hiện trong mỗi điểm ảnh như Hình 2.1 **Error! Reference source not found.** B. Sau đó đem chồng lên 1 lớp ảnh Mask (Hình 2.1. A) đã lọc theo ngưỡng màu từ ảnh gốc. Kết quả thu được ta sẽ thu được một ảnh trắng đen như Hình 2.1. C với vùng màu trắng là vùng có kênh màu cần nhận dạng. Tiến hành thí nghiệm nhận dạng một nắp chai màu xanh, ta thấy ảnh thu được sau khi tìm vùng chứa vật thể còn nhiều khá nhiều, nguyên nhân là do các vùng lân cận có thể có kênh màu gần giống với vật thể. Để thu được kết quả chính xác hơn, ta tiến hành bước lọc nhiễu.



Hình 2.1 Kết quả các bước trong quy trình nhận dạng vật thể

Để lọc các giá trị nhiễu này, ta áp dụng phương pháp lọc nhiễu hình thái (morphology) với 2 thao tác cơ bản là co ảnh (erosion) và giãn nở ảnh (dilation). Ý tưởng chính của phương pháp này là dùng phép toán erosion làm giảm kích thước của đối tượng, tách rời các đối tượng gần nhau, từ đó làm tiêu biến các điểm nhiễu nhỏ. Sau đó sẽ dùng phép toán dilation làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước để vùng biên của vật thể tròn trịa, mịn hơn.

❖ **Toán tử erosion :**

$$A \ominus B = \{c \mid (B)_c \subseteq A\}$$

Trong đó:

- A: Ma trận điểm ảnh của ảnh nhị phân.
- B: Là phân tử cấu trúc.

Phép co ảnh erosion sẽ cho ra một tập điểm ảnh c thuộc A, nếu ta đi chuyển phần tử cấu trúc B theo c, thì B nằm trong đối tượng A.

$$I_{src} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad I_{dst} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Hình 2.2 Ví dụ về toán tử erosion

Một ví dụ trực quan hơn như Hình 2.2, ta có ma trận điểm ảnh I_{src} , ma trận điểm ảnh sau phép co I_{dst} và cấu trúc phân tử B . Ứng với công thức ở trên, ta lần lượt đặt phần tử cấu trúc vào các điểm ảnh có giá trị 1 của ma trận điểm ảnh I_{src} . Kết quả thu được là ma trận điểm ảnh I_{dst} .

❖ Toán tử dilation

Toán tử dilation với phương pháp tương tự như toán tử erosion, nhưng cấu trúc phân tử B sẽ mở rộng vùng chứa ô có giá trị

Sau quá trình lọc nhiễu, ta thu được **Error! Reference source not found.** Để tiến hành việc truy vết ảnh. Ở đây chúng em dùng thuật toán Camshift (Continuously Adaptive Meanshift) trong OpenCV. Thuật toán này bản chất là áp dụng thuật toán Meanshift lên mỗi frame ảnh của nguồn video thu được từ webcam.

2.3. Áp dụng Vuforia trong việc nhận dạng

2.3.1. Giới thiệu

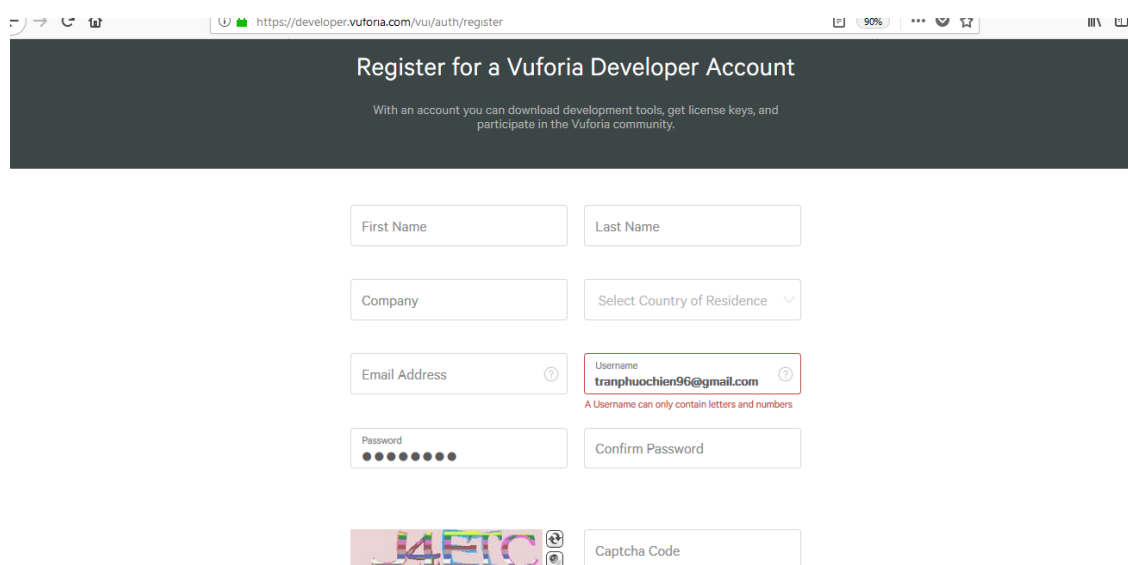
Vuforia là một nền tảng cho phép tạo ra các ứng dụng thực tại tăng cường. Nhưng điểm đặc biệt về thế mạnh của nó là sự hỗ trợ việc nhận diện hình ảnh, vật thể hoặc tái tạo môi trường trong thế giới thực với thư viện ảnh đồ sộ và các công cụ hỗ trợ thiết kế vật thể 3D. Với mục đích sử dụng Vuforia để nhận diện Object trong trường hợp vật cần nhận diện có nhiều họa tiết hoa văn chứ không phải là một mặt phẳng đơn sắc. Chúng em tiến hành sử dụng Vuforia để kiểm tra mức độ hiệu quả của nó so với quy trình nhận dạng.

2.3.2. Các vấn đề kỹ thuật khi sử dụng Vuforia

❖ Tích hợp Vuforia trong Unity

Đăng kí tài khoản sử dụng Vuforia API :

- Bước 1: Truy cập vào <https://developer.vuforia.com/vui/auth/register> để đăng kí tài khoản developer (Hình 2.3). Sau khi điền đủ thông tin vào mẫu đăng kí và xác nhận tạo tài khoản, ta xác thực việc đăng kí với email được gửi đến.



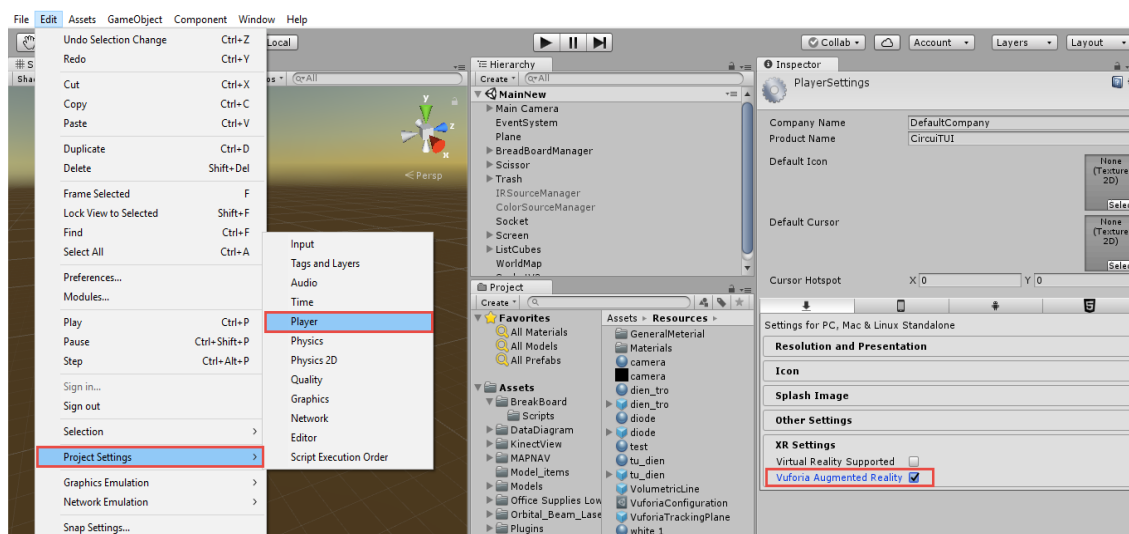
Hình 2.3 Đăng kí tài khoản Vuforia

Sau khi xác nhận tài khoản, ta đăng nhập vào hệ thống với đường dẫn: <https://developer.vuforia.com/vui/auth/login>

- Bước 2: Sử dụng Vuforia API với Unity :

Trước hết, ta cần tải bộ cài từ trang Download của Vuforia với đường dẫn sau: <https://developer.vuforia.com/downloads/sdk>. Cài đặt được tải ở dạng là một package của Unity, khi tải xong sẽ được cài trực tiếp vào Unity. Tuy nhiên đây là cách cài đặt dành cho Unity có phiên bản trước 2017.2, từ phiên bản 2017.2 trở về sau, Vuforia có thể được cài đặt với các package khác khi cài đặt Unity.

Tiếp theo, ta khởi động Unity, Chọn Edit => Project Settings => Player. Ở cửa sổ Inspector, ta đánh dấu vào “Vuforia Augmented Reality” (Hình 2.4)



Hình 2.4 Cho phép sử dụng Vuforia trong project

Để sử dụng các chức năng của Vuforia thì ta chọn Game Object => Vuforia rồi chọn chức năng tương ứng, ở đây chúng em lấy chức năng cơ bản của Vuforia là Image Target. Để sử dụng Image Target thì ở trang developer của Vuforia ta cần tạo một App Key để sử dụng. Khi điền đủ thông tin và xác nhận tạo key, ta có một App key được tạo như hình dưới, ta nhấn vào App Key Name để lấy Key (Hình 2.5)

Ta copy phần App Key, ở màn hình Unity, ta chọn Window => Vuforia Configuration, sau đó Key vào ô App License Key (Hình 2.6).

Please copy the license key below into your app

```
AVIoI3T/////AAAAGU8/wb4rrE+6qDR+e15ZAZ91gaYp1c1eRSDiRCL
1xVt+iH2x24tN8NC/zsOPNN4c923OR2ENcDaebSazy+Ph+W2rQI1p18
0aX8howM5ec5K61ysxX487OxheU8YtLqHc7v1NhD3CQrRclzy8cZnHm
VUbS0W743rjHmhIaY7vLjXKJ/OVkkUpj0HzTpWCHukY2xErrzf671RL8
+W5wDbQSQOCCFDWC03ePQwzes0T/L3UTiXE2/Iz70Kqbs5VuuRM/hIt
Lkvxs02gk5Rv4fbQ17Pdss9v5VsYOWk14lrJeKxx77q6bjNwb2C31lm
+Cj962E91MXASf8exMEGrVXXubga+h1ifgXrnsD9HPfItto9Aoc
```

Type: Develop

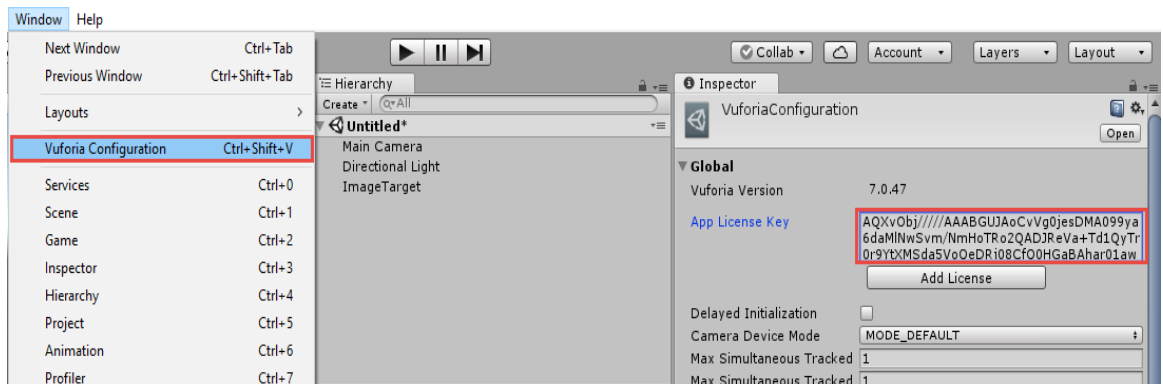
Status: Active

Created: Apr 11, 2017 21:51

History:

License Created - Apr 11, 2017 21:51

Hình 2.5 Giao diện lấy App Key

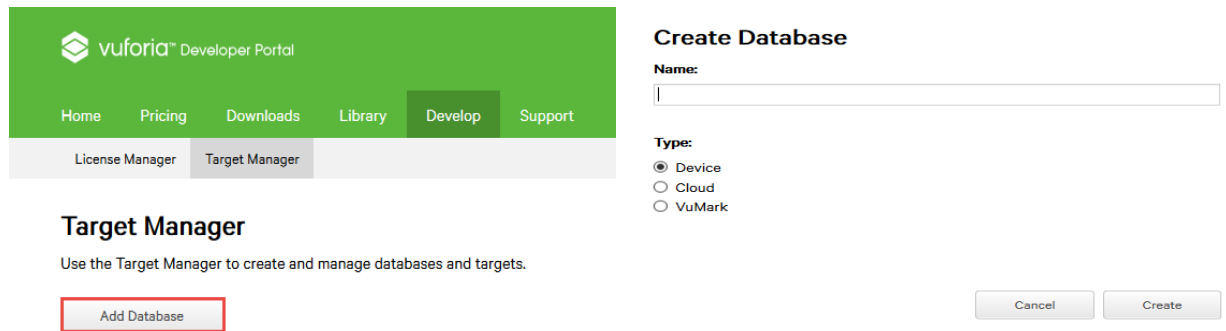


Hình 2.6 Gắn App Key vào project

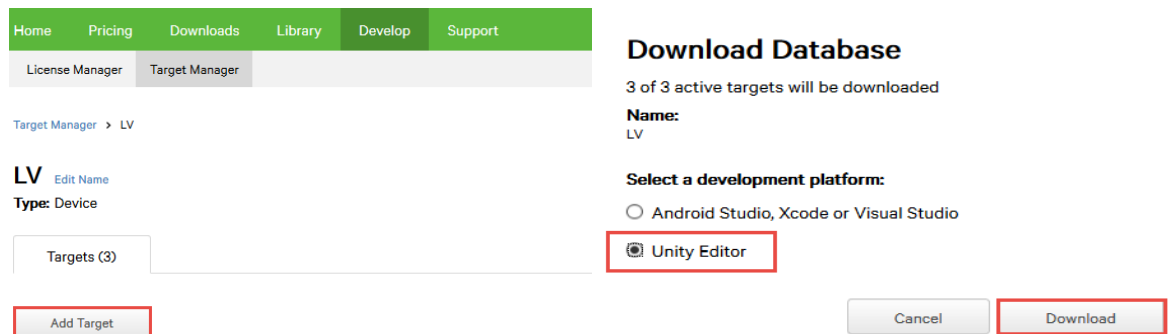
- Bước 3: Chuẩn bị database cho việc nhận dạng

Bây giờ, ta đã có thể sử dụng Image Target, chức năng cơ bản của Image Target là nhận diện một hình ảnh 2D đã được thêm vào database của Vuforia. Để thêm một hình ảnh vào data base ta đăng nhập vào trang chủ Vuforia, ở Tab Develop chọn Target Manager và nhấn Add Database. Đặt tên Database và tạo với type là Device.

Ta nhấn vào tên Database và nhấn Add Target để thêm dữ liệu vào Database. Có thể thêm hình ảnh 2D hoặc các vật thể 3D (Hình 2.7). Sau khi upload dữ liệu, điền kích thước và đặt tên thì ta nhấn Add để xác nhận. Khi dữ liệu được đưa vào Database thì mức độ đánh giá khả năng nhận diện được thể hiện qua phần Rating. Cuối cùng nhấn vào Download Database (All) để tải về Unity. Khi tải về ta chọn Development platform là Unity Editor rồi nhấn Download . File tải về dưới dạng là package của Unity sẽ được cài trực tiếp (Hình 2.8).



Hình 2.7 Tạo Database hình ảnh nhận diện



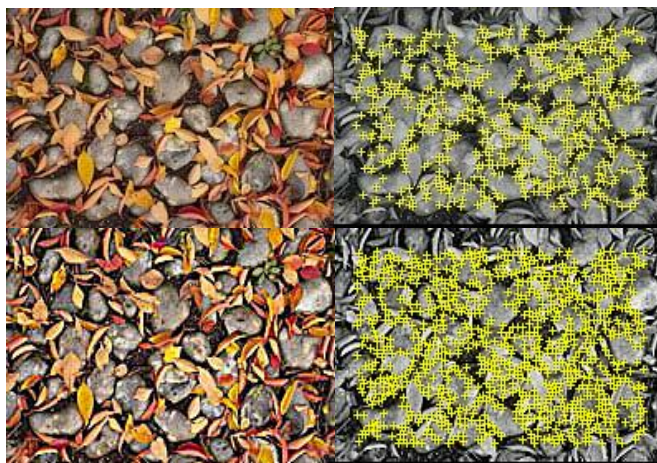
Hình 2.8 Tải Database về project

Ở giao diện Unity, ta chọn Database tương ứng vừa tải về và chọn Image Target là ảnh được đưa vào Database (Hình 12). Sau đó là có thể Start Game Mode của Unity và nhận diện.



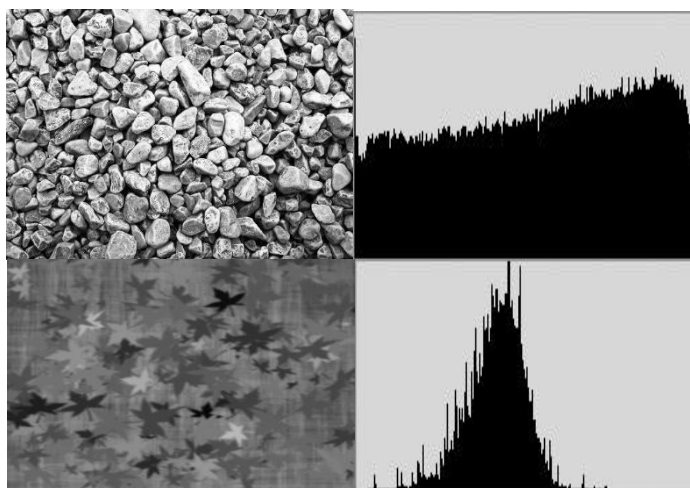
Hình 2.9 Chọn Target theo hình ảnh từ Database

❖ Tiêu chí đánh giá ảnh của Vuforia và các yếu tố ảnh hưởng



Hình 2.10 Dựa vào độ tương phản

Nguồn : <https://www.edrawsoft.com/umldiagramtemplate.php>



Hình 2.11 Độ tương phản ảnh Grayscale tương ứng với histogram

Nguồn : <https://www.edrawsoft.com/umldiagramtemplate.php>

Vuforia có thể nhận diện và theo dấu Targets bằng việc phân tích sự tương phản dựa trên những đặc tính được rút trích từ Target đó thể hiện cho camera như họa tiết của ảnh, tỉ lệ ảnh, chất lượng ảnh, độ cân bằng trong phân bố các chi tiết ảnh... Có thể cải thiện hiệu quả nhận diện bằng cách cải thiện những đặc tính đó (tăng độ chi tiết, độ tương phản màu sắc tốt (Hình 2.10), không bị lấp nội dung, chi tiết có nhiều góc cạnh). Format của ảnh phải là ảnh PNG hoặc JPG 8 hoặc 24 bit, nhỏ hơn 2MB, với ảnh JPG phải dùng hệ màu RGB hoặc grayscale (không sử dụng hệ màu CMYK). Để đánh giá tiêu chí Rating với ảnh Grayscale thì ta dựa vào Grayscale Histogram,

ảnh có độ tương phản cao, có Histogram rộng và phẳng thường là ảnh có Rating cao (Hình 2.11)

Về camera thì nếu có chế độ focus sẽ tốt hơn cho việc nhận diện. Bên cạnh đó, điều kiện ánh sáng môi trường cũng có ảnh hưởng lớn đến việc nhận diện và theo dấu. Vuforia hoạt động tốt nhất với ánh sáng trong phòng vì nó ổn định và dễ dàng điều chỉnh cường độ sáng. Kích thước Target tốt nhất từ 5 inches hoặc 12cm trở lên về chiều rộng cùng với kích thước hợp lí theo tỉ lệ ảnh về chiều cao. Khoảng cách từ ảnh đến camera trong điều kiện hoàn hảo khoảng bằng 10 lần chiều rộng của ảnh. Ảnh được in ra cần đảm bảo độ phẳng, không bị bẻ cong hay nhăn, nên sử dụng giấy bìa cứng để in ảnh nhận diện. Bề mặt ảnh với camera nên là bề mặt nhám, vì bề mặt bóng dưới nguồn sáng và góc chiếu sáng ngẫu nhiên có thể dẫn đến hiện tượng phản xạ, khúc xạ làm mất đi chi tiết của ảnh. Góc nhìn của camera đến ảnh nên trực diện và tránh góc nhìn quá hẹp so với bề mặt ảnh nhận diện.

Sau quá trình thử nghiệm với Vuforia, chúng em thấy được lợi điểm của nó cho phép nhận diện và theo dấu được vật có bề mặt phức tạp. Tuy nhiên, nó cũng bị phụ thuộc vào điều kiện môi trường như quy trình ở Sơ đồ 2-1 và khả năng ánh xạ tọa độ thu được của vật trong thế giới ảo với vật trong thế giới thật khá phức tạp.

2.4. Áp dụng Kinect trong việc nhận dạng

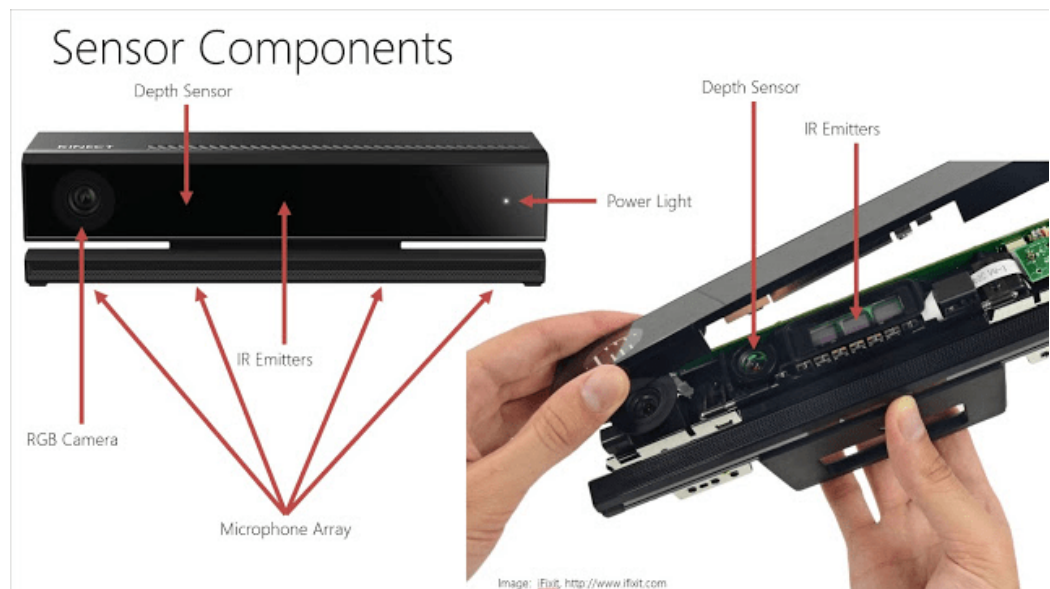
2.4.1. Giới thiệu Kinect

Kinect là thiết bị cảm biến thu nhận chuyển động được Microsoft phát triển dành riêng cho máy chơi game Xbox360 và về sau là nhằm phục vụ máy vi tính sử dụng hệ điều hành Windows. Quá trình phát triển của Kinect có thể tóm tắt như sau: Vào 30/5/2007, trong hội nghị D5, chủ tịch Microsoft là Bill Gates đã đề xuất ra một phương thức điều khiển game từ người chơi mới dựa vào thu nhận dữ liệu từ camera, tương tác trực tiếp bằng hành động cử chỉ của người chơi và vật thể thực tế. Ngày 7/4/2008, MTV News Reports đưa tin Microsoft đang tích cực phát triển một thiết bị điều khiển bằng chuyển động để cạnh tranh với Nintendo's Wii. 1/6/2009, Microsoft công bố dự án Natal tại hội nghị E3, đồng thời đưa ra minh họa cho việc sử dụng thiết bị với công nghệ điều khiển bằng hành động. Natal có thể nhận diện được đồng thời 4 người với 48 khớp xương trên cơ thể. Đến ngày 13/6/2010, tại E3, Microsoft chính thức đổi tên dự án Natal thành Kinect kết hợp giữa 2 từ kinectic và connect. Ngày 20/7/2010, Microsoft thông báo chính thức thời gian phát hành và giá bán của Kinect, nhưng mới chỉ dành cho phiên bản Xbox. Ngày 2/1/2012, Microsoft công bố phiên bản Kinect dành cho windows kèm theo phiên bản thương mại chính thức Kinect SDK. Đến tháng 10/2014, phiên bản thứ 2 của Kinect được bày bán với những cải tiến:

- Độ phân giải Full HD 1080.
- Tích hợp bộ cảm biến 3D, tăng khả năng nắm bắt chi tiết.
- Cảm biến hồng ngoại để nhận biết được đối tượng trong môi trường thiếu sáng, với phạm vi hoạt động rộng hơn.
- Độ sâu trung thực (3x).
- Tăng điểm khung xương lên 25.
- Cải thiện tầm nhìn (0.5m đến 4.5m).
- Đa ứng dụng cùng xử dụng 1 Kinect.
- Hỗ trợ theo dõi các ngón tay, cử chỉ nắm mở bàn tay.
- Có khả năng đo được nhịp tim của người dùng.

❖ So sánh Kinect v2 và Kinect v1 [13]:

Đặc điểm	Kinect v1	Kinect v2
Màu sắc	640×480 @ 30fps	1920×1080 @ 30fps
Độ sâu	320×240 @ 30fps	512×424 @ 30fps
Cảm biến	Structured Light (PrimeSense Light Coding)	Time of Flight (ToF)
Tầm nhìn	0.8~4.0 m	0.5~4.5 m
Góc nhìn (ngang/dọc)	57°/43°	70°/60°
Mic	Có	Có
Số người nhận được	6	6
Số khớp xương	20	25
Trạng thái tay	Open/Close	Open/Close/Lasso
Cử chỉ	Không hỗ trợ	Có



Hình 2.12 Cấu tạo Kinect v2

Nguồn : [https://www.slideshare.net/MarcoDAlessandro/nui-e-biometrics-in-windows-](https://www.slideshare.net/MarcoDAlessandro/nui-e-biometrics-in-windows-10)

10

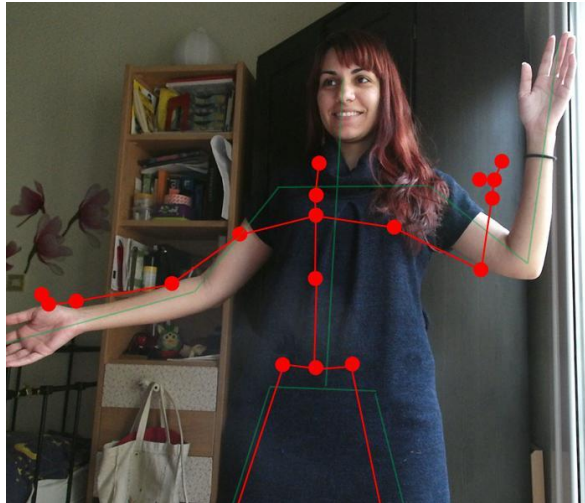
2.4.2. Cấu tạo Kinect v2

Kinect v2 có kích thước 14.8” x 5.9” x 4.8”, có một camera RGB, một cảm biến độ sâu và dãy microphone. Nó cho phép thu được đầy đủ chuyển động cơ thể dưới dạng 3D, nhận dạng khuôn mặt và giọng nói. Camera RGB cho phép nhận diện tối đa 6 người cùng lúc, theo dõi chuyển động tối đa 2 người. Có cấu tạo được thể hiện như Hình 2.12 :

- IR Emitters: thiết bị dùng để phát tia hồng ngoại.
- RGB Camera: camera dùng để thu hình ảnh màu bên ngoài. Camera này hỗ trợ nhiều định dạng ảnh và độ phân giải khác nhau
- IR Camera: có nhiệm vụ bắt tia hồng ngoại được chiếu từ IR Emitters phản xạ từ các vật thể và dùng thông tin đó để tính toán khoảng cách từ camera đến các vật thể được chiếu đến.
- Multi-Array Mic: là một dãy gồm 4 microphone có nhiệm vụ thu lại âm thanh và xác định hướng phát ra âm thanh. Dãy microphone này được dùng trong các ứng dụng cần nhận diện giọng nói.

Các định dạng dữ liệu stream được Kinect v2 hỗ trợ :

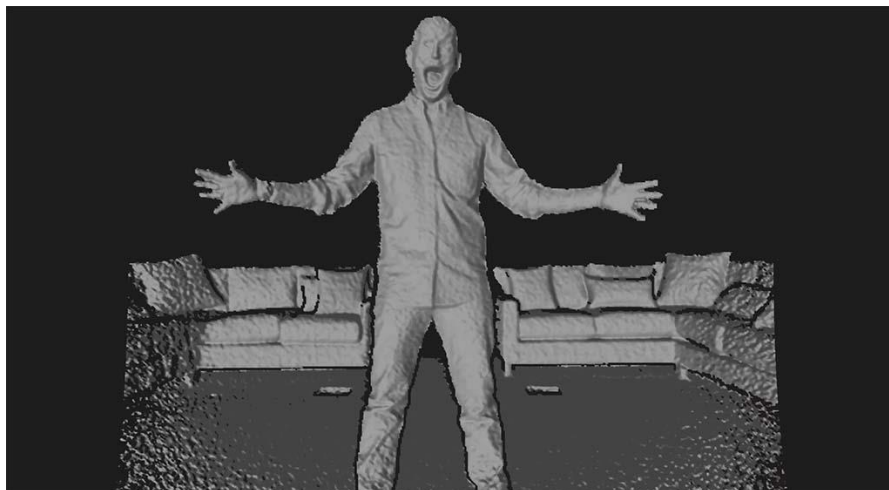
- Image Stream: trả về dữ liệu ảnh màu thu được từ RGB camera, hỗ trợ định dạng hình ảnh RGB_1920x1080(định dạng RGB, độ phân giải 1920x1080, tốc độ khung hình tối đa 30 fps, cho phép khả năng tăng độ sáng, phơi sáng tốt hơn).



Hình 2.13 Image stream Kinectv2

Nguồn : <https://www.mindtreatstudios.com/the-art-of-speech/>

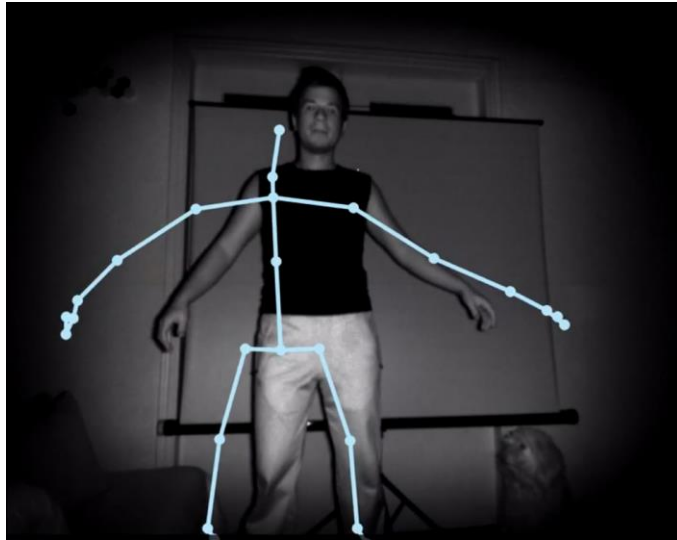
- Depth Stream: trả về dữ liệu độ sâu của khung nhìn và dữ liệu phân đoạn cơ thể bằng cách sử dụng 3D Depth Sensor. Với body tracking, có thể lấy được hình ảnh của 6 người tại một khung hình trong khoảng từ 0.5ms đến 4.5m. Depth Stream hỗ trợ định dạng DEPTH_512x424(độ phân giải 512x424, tốc độ khung hình 30fps).



Hình 2.14 Depth stream kinectv2

Nguồn : <https://brekel.com/kinect-2-details-specifications-observations/>

- Body Stream: trả về dữ liệu độ sâu với mốc thời gian, bao gồm vị trí tổng thể của cơ thể với 25 khớp xương



Hình 2.15 Body stream Kinectv2

Nguồn : <https://pterneas.com/2014/03/13/kinect-for-windows-version-2-body-tracking/>

2.4.3. Các vấn đề kỹ thuật khi sử dụng Kinect

❖ Lấy các kênh ảnh từ Kinect



Đặt vấn đề

Một lợi thế rất lớn khi sử dụng Kinect là Kinect cung cấp cho chúng ta nhiều kênh thông tin khác nhau để phục vụ cho việc nhận dạng. Ở đây, chúng em xác định sử dụng 2 kênh hình ảnh là ảnh hồng ngoại và ảnh độ sâu. Vậy làm sao có thể giao tiếp với Kinect để lấy thông tin từ các kênh ảnh này ?



Giải pháp

Để lấy thông tin ảnh hồng ngoại từ Kinect, ta chỉ cần khởi tạo đối tượng *KinectSensor* và gọi API *InfraredFrameSource*. Trong luận văn này, chúng em sử dụng Kinectv2 nên ảnh trả về có kích thước là 512x424. Tuy nhiên để tăng tốc xử lý, ta có thể giới hạn lại kích thước ảnh trả về để hình ảnh trả về phù hợp với không gian cần nhận dạng.

2.5. Kết luận

Với việc sử dụng ảnh màu để nhận diện vật thể, chúng em đã lấy được kết quả thành công từ quá trình nhận diện. Tuy nhiên, với nhận diện vật thể bằng ảnh màu thì việc giữ màu sắc ổn định là rất quan trọng, bên cạnh đó vật nhận diện phải có bề mặt đơn sắc và ít phản sáng. Điều này không thực tế vì ánh sáng môi trường ảnh hưởng rất nhiều đến việc nhận diện và vật thể nhận diện không phải lúc nào cũng chỉ có một màu.

Với Vuforia, chúng em đã giải quyết được nhược điểm của phương pháp nhận diện bằng ảnh màu phía trên bằng việc nhận diện vật thể có tổ hợp màu đa dạng hoặc có họa tiết phức tạp. Tuy nhiên, Vuforia cũng bị ảnh hưởng bởi ánh sáng môi trường khá nhiều khi hình ảnh nhận diện bị mất chi tiết do ánh sáng chiếu vào. Thêm vào đó việc mapping tọa độ khi sử dụng Vuforia khá khó so với việc mapping tọa độ bằng hình ảnh.

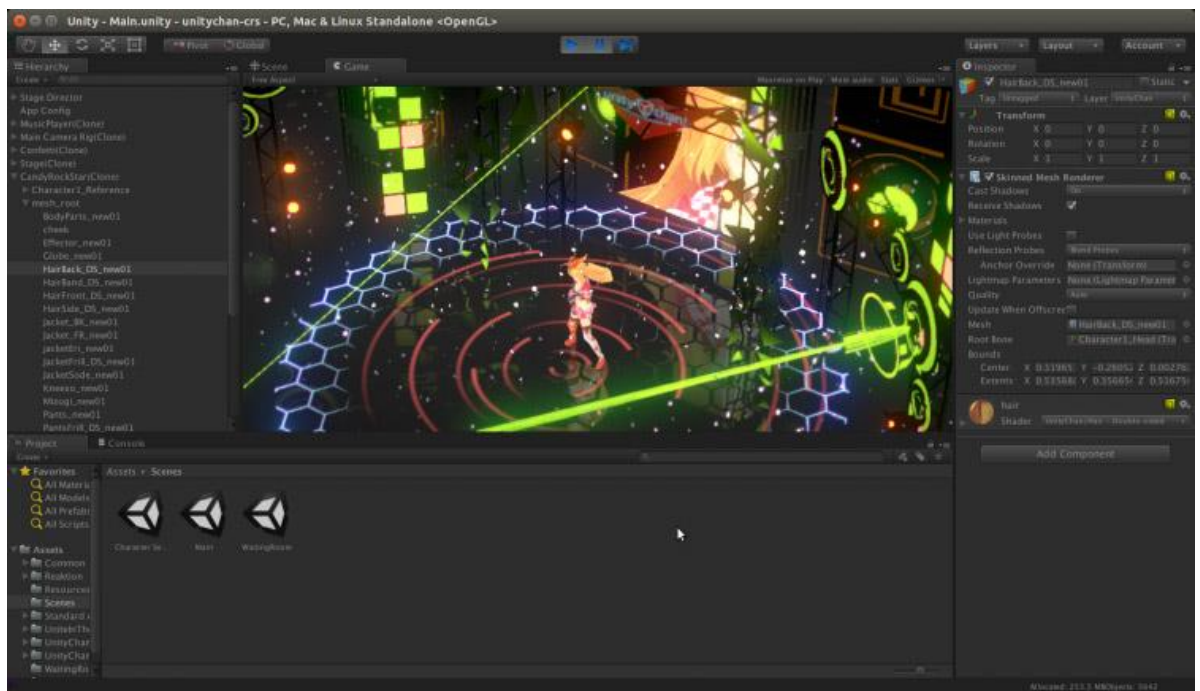
Cuối cùng, với 2 kênh dữ liệu từ Kinect, chúng em có thể hỗ trợ độ chính xác cho việc nhận diện bằng kênh màu với kênh độ sâu. Kết quả thu được chính xác hơn và ít nhiễu hơn so với phương pháp dùng ảnh màu cơ bản, và cách mapping tọa độ cũng giống như cách mapping ở phương pháp dùng ảnh màu phía trên. Chính vì thế chúng em quyết định lựa chọn Kinect cho việc nhận diện vật thể.

Chương 3

Các vấn đề kỹ thuật và giải pháp về đồ họa

✍ Chương 3 sẽ trình bày các vấn đề về kỹ thuật và giải pháp về đồ họa, bao gồm: nhu cầu thực tế phát sinh; giải pháp đưa ra và lý do lựa chọn Unity Engine làm môi trường phát triển; cách thức tiến hành và cài đặt trong Unity và các thông số kỹ thuật trong Unity

3.1. Mở đầu



Hình 3.1 Giao diện Unity Engine

Unity là một nền tảng hỗ trợ phát triển game đa nền tảng được phát triển bởi Unity Technologies [14] Lần đầu tiên Unity được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng. Unity cung cấp một công cụ soạn thảo bằng giao diện người dùng để sử dụng. Đặc biệt Unity hỗ trợ rất tốt các công việc liên quan đến đồ họa 3D, người dùng có thể dễ dàng kéo và thả các đối tượng, tùy chỉnh các thông số của mỗi đối tượng qua hệ thống menu (Hình 3.1 Giao diện Unity Engine). Bên cạnh dùng giao diện, người dùng Unity còn có thể viết kịch bản bằng các ngôn ngữ JavaScript, C # hoặc

Boo. Hơn nữa, Unity có một cộng đồng rất lớn trên Unity Store, nơi cung cấp rất nhiều asset, resource có sẵn từ miễn phí đến có phí.

3.2. Các vấn đề kỹ thuật và giải pháp trong Unity

3.2.1. Camera



Vấn đề

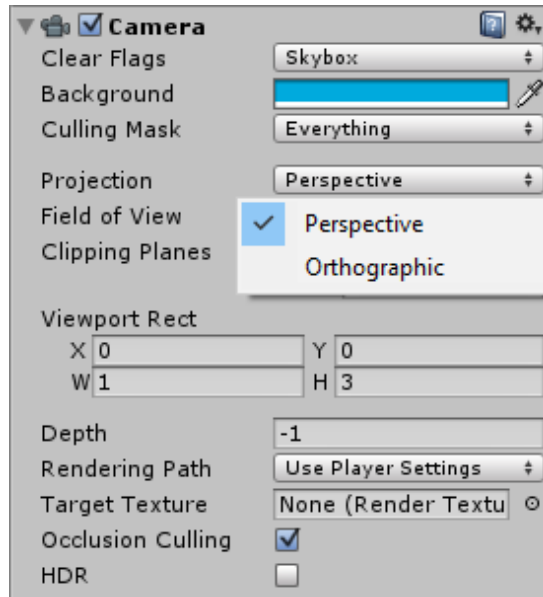
Để có thể linh hoạt trong việc trình chiếu cảnh vật, đối tượng, ta cần phải điều chỉnh hướng góc nhìn, tầm nhìn hoặc thay đổi cả vị trí của góc nhìn như góc nhìn thứ nhất, góc nhìn thứ ba. Ở mỗi góc nhìn khác nhau, người dùng sẽ có những trải nghiệm khác nhau. Vậy trong Unity có công cụ nào để tùy chỉnh góc nhìn hay không ?



Giải pháp

Camera là thành phần cơ bản nhất trong Unity. Điều chỉnh camera sẽ tác động đến nội dung được chiếu ra trên màn ảnh. Trong mô trường Unity, một ứng dụng có thể có một hoặc nhiều camera, và có 1 camera đóng vai trò chính. Việc có thể có nhiều camera trong 1 ứng dụng giúp linh hoạt chuyển các góc nhìn khác nhau hoặc hỗ trợ chế độ 2 người chơi cùng lúc. Sau đây ta sẽ đi vào chi tiết thành phần của 1 camera. Có thể nói đến một số thành phần cơ bản như Hình 3.2 :

- Góc chiếu của camera (Projection): Gồm phép chiếu song song (Orthographic) và phép chiếu xa gần (Perspective). Phép chiếu song song được sử dụng trong game 2D, khi tỉ lệ các đối tượng sẽ được thể hiện không đổi ở mọi chiều sâu. Phép chiếu xa gần được sử dụng trong game 3D, mô phỏng góc nhìn theo thế giới thực, tỉ lệ các đối tượng sẽ theo luật xa gần.
- Góc mở của camera (Field Of View): thuộc tính của phép chiếu xa gần, cùng một đối tượng và cùng một vị trí, nếu góc mở càng lớn thì đối tượng càng nhỏ và ngược lại. Đây là kỹ thuật được dùng cho hiệu ứng zooming.
- Mặt phẳng xa nhất (Far) và mặt phẳng gần nhất (Near): vật chỉ được hiển thị nếu nó nằm giữa 2 mặt phẳng trên và ngược lại.



Hình 3.2 Thông số cơ bản của Camera

3.2.2. Material

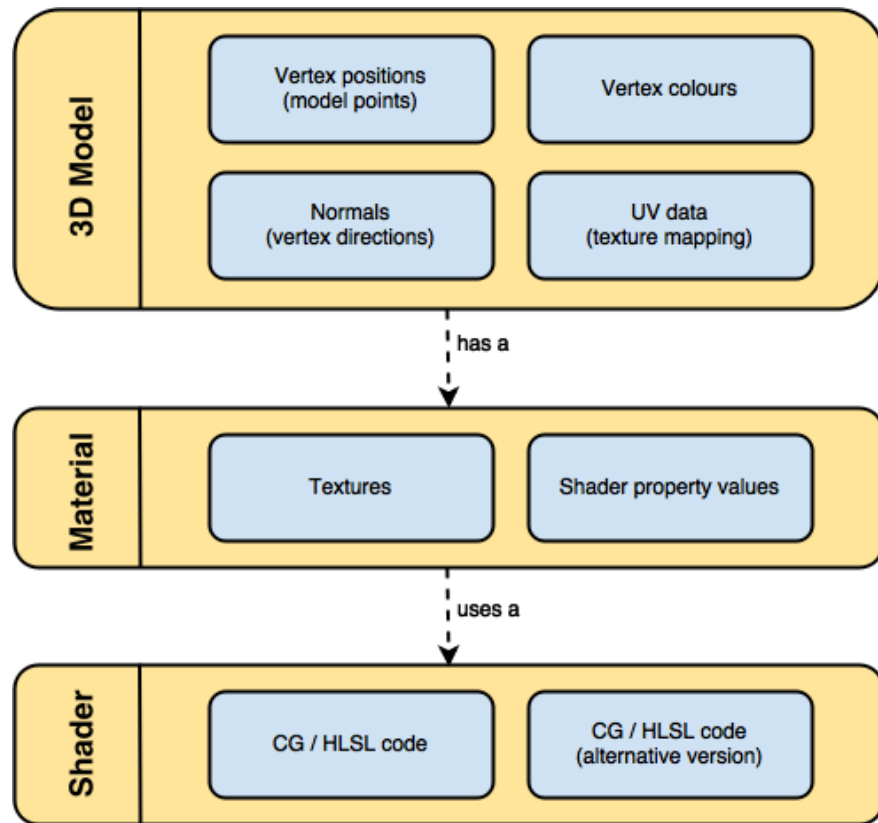


Vấn đề

Với một Object được tạo ra thì thành phần cơ bản của nó là Mesh giúp nó xác định được hình dạng, để hỗ trợ cho việc thể hiện thêm thông tin màu sắc thì theo cơ chế kết xuất đồ họa trong Unity (Sơ đồ 7) thì ta cần sử dụng một công cụ giúp ta trang trí vật đó, nó là Material.



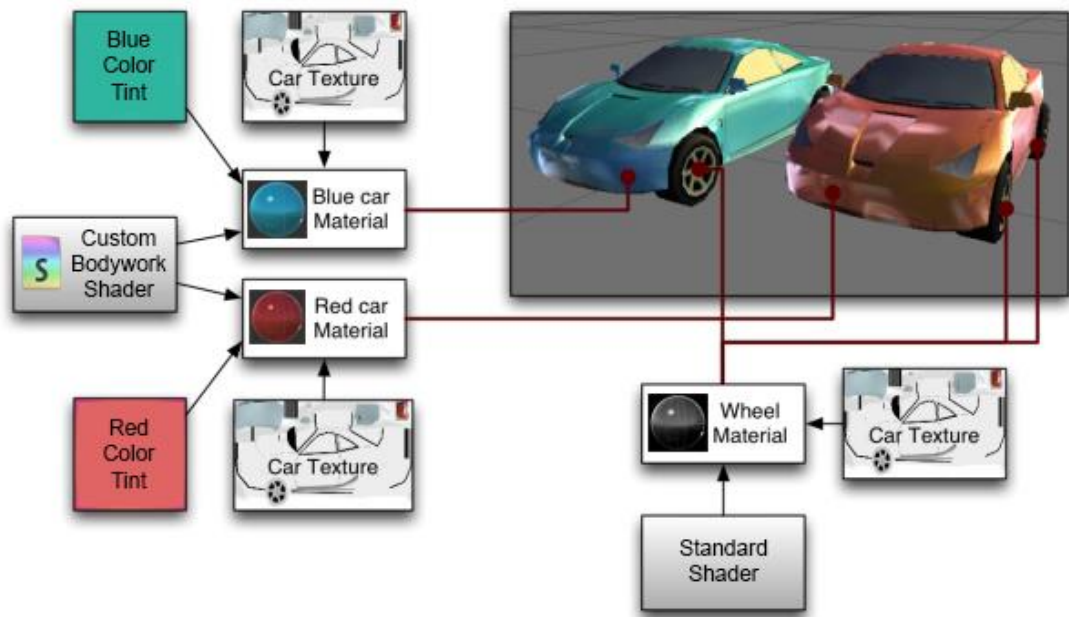
Giải pháp



Sơ đồ 3-1 Cơ chế kết xuất đồ họa trong Unity

Nguồn : <https://unity3d.com/learn/tutorials/topics/graphics/gentle-introduction-shaders>

Material được sử dụng trong các thành phần kết xuất đồ họa trong Unity. Nói cách khác, nó đóng vai trò quan trọng trong việc cho thấy một Object sẽ được phô bày trên màn hình thế nào, nó như là lớp áo của Object đó. Thuộc tính mà Material thể hiện được quyết định bằng Shader. Shader là một chương trình đồ họa đặc biệt được chạy trên GPU có thể xác định bề mặt họa tiết (Texture) và thông tin ánh sáng được thể hiện thế nào khi một Object được tạo ra trên màn hình. Texture là họa tiết được lợp lên trên vật bởi Material đó. Một Shader có thể sử dụng cho nhiều Material và một Material có thể áp dụng cho nhiều 3D Object. Có thể minh họa bằng Hình 17.



Hình 3.3 Sử dụng Material để lợp màu cho Object

Nguồn : <https://docs.unity3d.com/Manual/Materials.html>

3.2.3. Collider



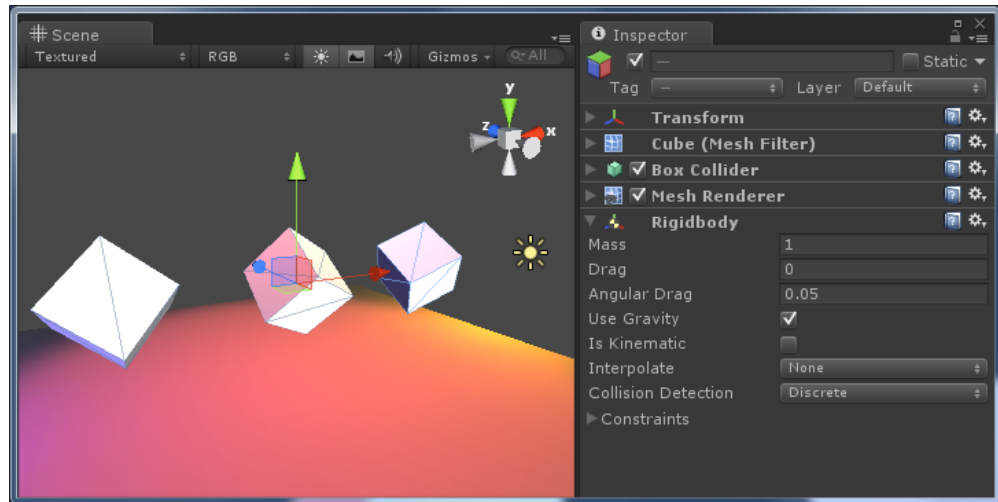
Vấn đề

Để mối quan hệ giữa các đối tượng trong ứng dụng trở nên thực hơn, ta phát sinh ra nhu cầu áp dụng tính chất vật lý lên các đối tượng như rơi, va chạm, lực kéo ma sát,... Vậy để áp dụng các tính chất vật lý này, Unity có công cụ nào để hỗ trợ hay không ?



Giải pháp

Trong Unity việc xử lý các tương tác vật lý sẽ giúp chương trình trở nên thực tế hơn. Trong tương tác vật lý thì xử lý va chạm là một nhu cầu cần thiết. Collider là vật mà Engine vật lý dùng để nhận ra sự va chạm. Đa số các collider có hình dạng đơn giản nhằm mục đích tính toán đơn giản và dễ dàng. Phần lớn các Object cơ bản trong Unity sẽ được gắn collider khi tạo ra (Cube, Sphere, Cylinder,...) (Hình 19). Ở hệ thống của chúng em, collider được dùng để xác định khi một Object trong game bị đè lên bởi một Object khác thông qua Raycasting (được nói ở phần sau).



Hình 3.4 Collider với Object Cube

Nguồn : <https://viblo.asia/p/co-ban-ve-engine-vat-ly-trong-unity-3OEgGj1PM9bL>

3.2.4. Particle system



Vấn đề

Để nâng cao trải nghiệm người dùng khi thao tác trong môi trường tương tác có hỗ trợ Tangible, đôi khi ta cần tạo ra các hiệu ứng đẹp mắt như hào quang, khói, lửa,.. Các hiệu ứng này giúp thu hút sự chú ý của người dùng cũng như là một dấu hiệu đề gợi ý cho người dùng trong các hệ thống hoạt động có kịch bản. Vậy trong Unity có công cụ nào để hỗ trợ tạo nên các hiệu ứng này hay không.



Giải pháp

Particle system (hệ thống hạt hiệu ứng) là kỹ thuật để tạo ra cái hiệu ứng tia lửa, cháy, nổ, khói bụi, hơi nước,.. trong game. Trong game 3D phần lớn các thành phần trong cảnh, các nhân vật đều được đại diện bởi Mesh, tương tự như việc ở game 2D sử dụng sprite để tạo nên các thành phần trong khung hình. Nói cách khác các thành phần này được cấu thành bởi một vật rắn có hình dạng cho trước. Nhưng với những hiệu ứng như chất lỏng, khói, mây, lửa thì phải tiếp cận bằng cách khác, chính vì thế Particle system được tạo ra để giải quyết vấn đề này. Hệ thống của chúng em cũng áp dụng particle system để tạo ra các hiệu ứng màu sắc cho việc chọn một Object để gây sự chú ý của người dùng trong quá trình sử dụng. Một vài thành phần cơ bản của Particle system:

- Duration: thời gian tính bằng giây quãng đời tồn tại của một hạt.
- Looping: có được tạo lại hay không khi hạt cuối cùng bị biến mất.
- Speed, Direction, Rotation: tốc độ, hướng di chuyển, góc quay của các hạt.
- Max particles: số hạt tối đa xuất hiện trong particle system.
- Material/Shader: dùng để thay đổi hình dáng của các hạt.
- Shape: hình dáng đường đi của các hạt tạo thành khi phát ra.

3.2.5. Raycasting



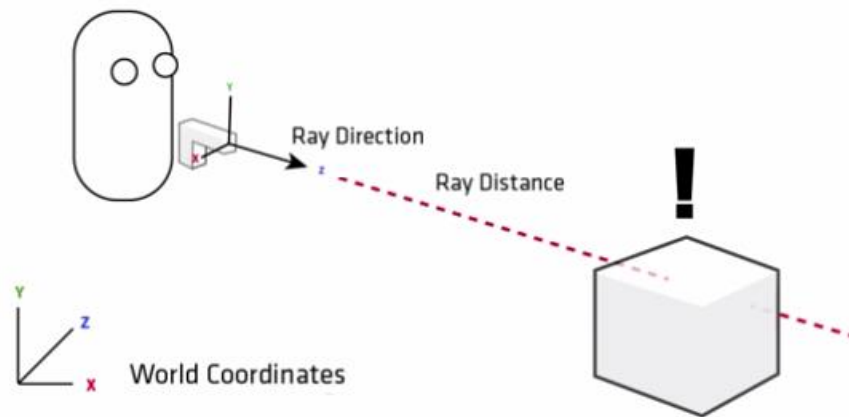
Vấn đề

Trong quá trình phát triển hệ thống, chúng em phát sinh nhu cầu làm thế nào để tìm được hình chiếu của một điểm lên một mặt phẳng cho trước, nhu cầu này gần giống với hành vi teleport trong game hoặc dùng tay chỉ vào một vật để chọn vật đó. Vậy kỹ thuật này được làm như thế nào trong Unity



Giải pháp

Trong Unity, raycasting là kỹ thuật bắn một tia từ điểm *origin*, theo hướng *direction*, có chiều dài *maxDistance* đập vào các colliders (Hình 3.5). Ứng dụng của nó có thể kể đến là bắt va chạm của một viên đạn với một Object khác hoặc là biết tay bạn đang chạm vào vật nào. Nó thường được sử dụng trong việc xác định đường ngắm. Ở hệ thống của chúng em, việc sử dụng Raycasting dùng để tạo một sự kiện khi tọa độ của vật chúng em cần nhận diện qua camera đề lên Object trên màn hình. Để lọc ra những vật đang quan tâm khi có Raycast chiếu trúng thì nên tạo Tag cho các vật đó để không bị nhầm lẫn với các vật khác.



Hình 3.5 Minh họa Raycasting trong Unity

Nguồn : <http://www.clonefactor.com/wordpress/unity3d-show-room/raycast-gizmos-visualizer/>

3.2.6. Lighting



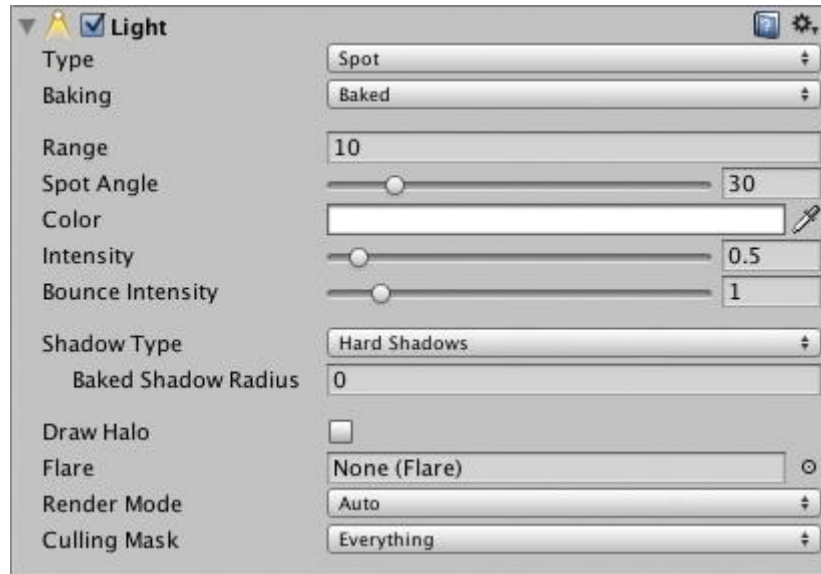
Đặt vấn đề

Trong thế giới thật, ánh sáng là một trong các yếu tố quan trọng nhất để tạo nên hình khối, độ chân thật các vật thể. Một mô hình 2D nếu kết hợp với việc chiếu sáng đúng góc nhìn và cường độ ánh sáng thích hợp sẽ làm cho người nhìn cảm thấy vật thật hơn. Vậy làm sao để điều chỉnh, thiết lập ánh sáng một cách chân thật nhất trong môi trường Unity ?



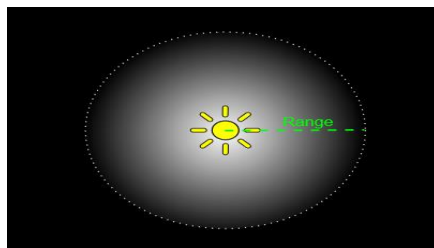
Giải pháp

Ánh sáng là một thành phần thiết yếu trong mọi cảnh, nếu chỉ dùng Mesh và Texture dùng để định hình các thành phần thì vẫn không thể mang lại hiệu quả đồ họa như mong muốn. Chính vì thế ánh sáng được dùng để xác định màu sắc của môi trường 3D, nhờ có ánh sáng ta không những phân biệt được màu sắc bề mặt vật mà còn thấy được vật độ phản xạ ánh sáng của vật (độ bóng-nhám của bề mặt). Một cảnh trong Unity có thể kết hợp nhiều nguồn sáng để tạo ra hiệu ứng đặc biệt về ánh sáng và màu sắc. Các thành phần cơ bản của ánh sáng như Hình 3.6

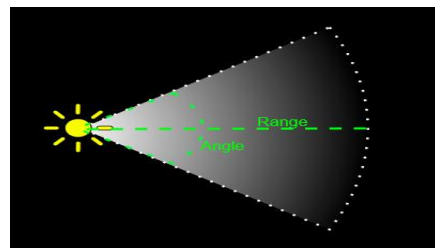


Hình 3.6 Các thành phần cơ bản của Light Component

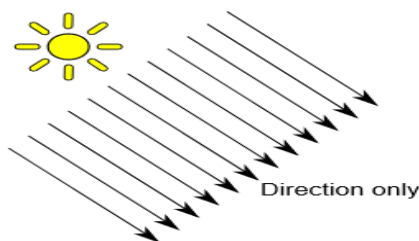
- Type : loại ánh sáng trong Unity, có 4 loại : Directional mô phỏng ánh sáng song song từ mặt trời (Hình 3.7 Các loại Light trong UnityC), Point mô phỏng ánh sáng từ bóng đèn, tâm nguồn sáng ở một điểm và phát tia sáng ra mọi hướng (Hình 3.7 Các loại Light trong Unity), Spot mô phỏng ánh sáng từ đèn pin, từ tâm nguồn sáng sẽ phát ra chùm sáng phân kì theo hình nón (Hình 3.7 Các loại Light trong Unity), Area mô phỏng ánh sáng phát ra từ bảng hiệu, ánh sáng phát ra từ mọi điểm trên 1 mặt của hình chữ nhật (Hình 3.7 Các loại Light trong Unity)



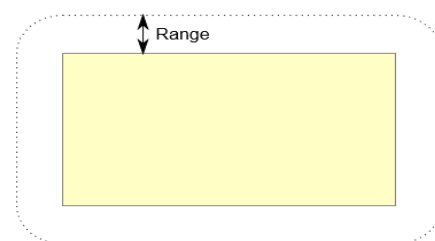
A.Point Light



B.Spot Light



C.Direction Light



D.Area Light

Hình 3.7 Các loại Light trong Unity

Nguồn : <https://docs.unity3d.com/Manual/Lighting.html>

- Range : xác định ánh sáng sẽ được chiếu bao xa.
- Spot Angle : xác định độ mở của chóp nón trong Spot Light
- Color : màu ánh sáng phát ra.
- Shadow Type : cường độ của bóng khi vật bị ánh sáng chiếu vào.

Trong hệ thống chúng em sử dụng Directional Light vì mọi thao tác sẽ được thực hiện trên mặt phẳng nên tại mọi điểm trên mặt phẳng đó cần có góc chiếu sáng, độ đổ bóng, cường độ ánh sáng nhận được là như nhau.

3.2.7. Position



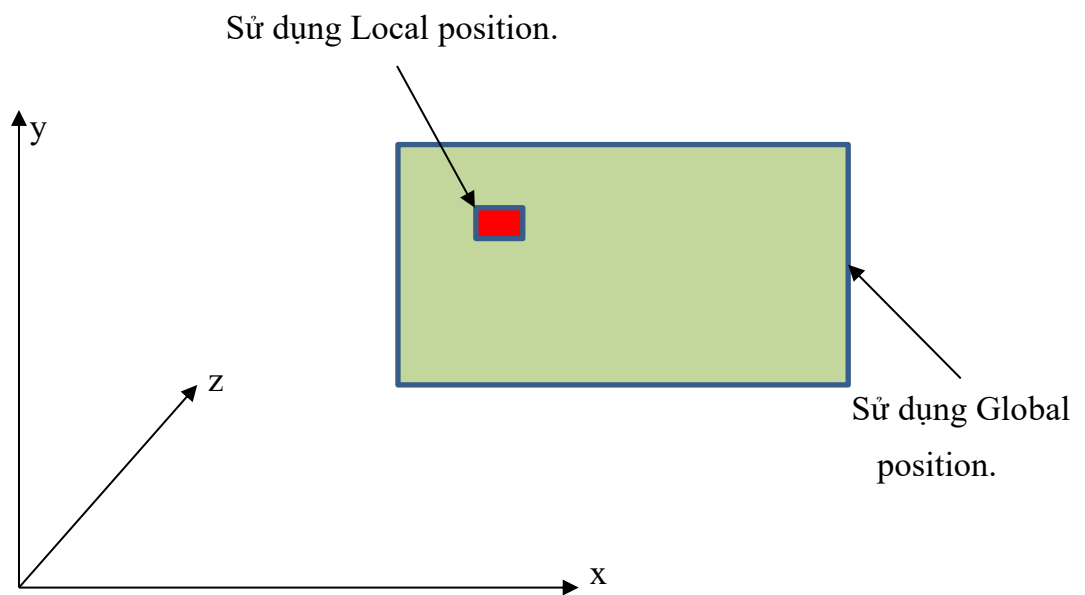
Vấn đề

Hệ thống của chúng em đang tồn tại 2 hệ tọa độ khác nhau: một hệ tọa độ từ camera thu được, một hệ tọa độ trong môi trường Unity. Vậy làm sao để có thể ánh xạ qua lại một tọa độ giữa hai hệ trục tọa độ đó?



Giải pháp

Khi tiến hành mapping tọa độ từ ảnh sang tọa độ Unity thì việc hiểu rõ các loại tọa độ trong Unity là rất quan trọng. Vì nó giúp chúng em dễ dàng hơn trong việc mapping tọa độ chính xác hơn. Để xác định vị trí của một Object trong Unity thì có 2 loại vị trí : position (global) và local position (local). Local position là vị trí của Object tương đối với vị trí của Parent Object chứa nó. Global position là vị trí của Object tính theo gốc tọa độ của World Space. Với những thành phần tách biệt nhau thì nên sử dụng Global position, tuy nhiên với việc ánh xạ tọa độ cho phù hợp với màn hình thì ở đây chúng em sử dụng Local position để dễ tính toán. Chúng em sử dụng một Plane Object để tượng trưng cho màn hình, sau đó đặt các Object con cần thao tác là con của Plane đó. Khi sử dụng Local position thì việc tính tọa độ sẽ dựa vào tâm của Plane giống như hệ tọa độ khi lấy từ camera, chính vì thế sẽ dễ dàng hơn cho việc ánh xạ tọa độ (Hình 3.8 Mô phỏng cách sử dụng position trong hệ thống).



Hình 3.8 Mô phỏng cách sử dụng position trong hệ thống

3.3. Kết luận

Trong chương này, chúng em đã giới thiệu được môi trường phát triển game, ứng dụng 3D đa nền tảng Unity và các kỹ thuật cần thiết trong Unity để xây dựng các ứng dụng đồ họa 3D một cách nhanh chóng. Nền tảng Unity có ưu điểm là dễ sử dụng, giao diện trực quan, dễ hiểu và có cộng đồng phát triển lớn mạnh. Những ưu điểm đó giúp chúng em phát triển nhanh được các ứng dụng được trình bày ở những chương tiếp sau.

Chương 4

Hệ thống bản đồ địa lý hỗ trợ Tangible

✍ Chương 4 này chúng em sẽ giới thiệu hệ thống bản đồ hỗ trợ Tangible, minh họa cụ thể ứng dụng của các thao tác đã nêu trên ở Chương 2. Hệ thống bản đồ chúng em xây dựng cung cấp cho người dùng 1 môi trường hỗ trợ Tangible, giúp người dùng có thể xem camera giao thông, xem thông tin tăng cường như độ ẩm không khí, nhiệt độ, lượng mưa,... bằng nhiều thao tác khác nhau.

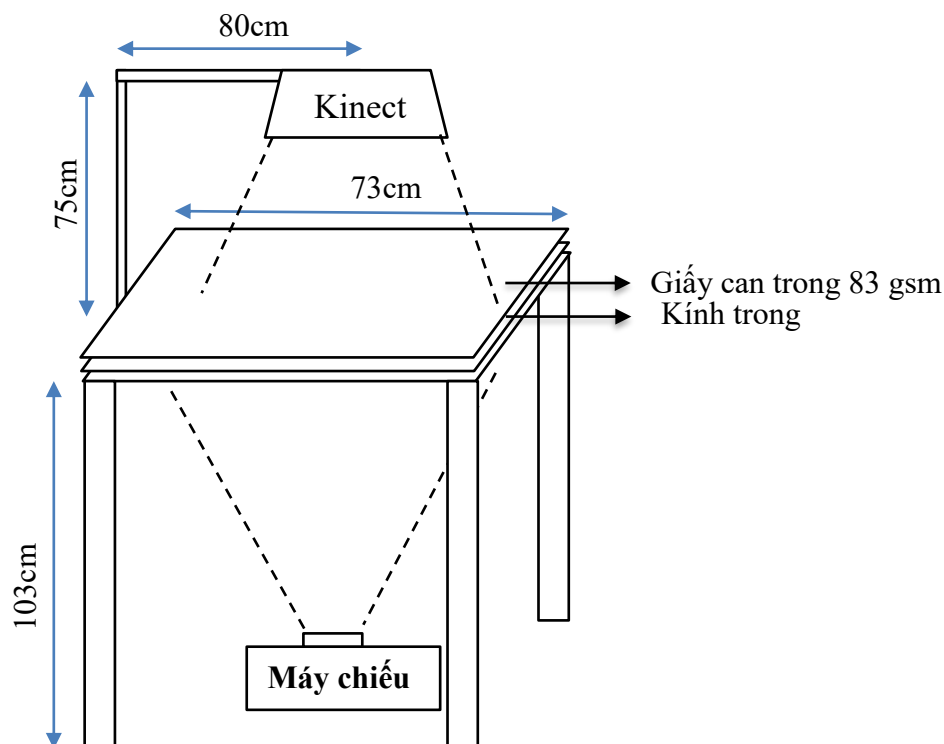
4.1. Giới thiệu tổng quan

Hiện nay tại các trường trung học, đa số các em học sinh đều chú trọng vào các môn tự nhiên hơn là các môn xã hội. Một trong những lý do chính là các môn xã hội quá nhiều lý thuyết khô khan mà ít thấy được ứng dụng cũng như minh họa cụ thể. Điều này khiến việc tiếp thu kiến thức trở nên nhàm chán và khó khăn hơn. Hiểu được điều này, nhóm chúng em đã đề xuất hệ thống bản đồ có hỗ trợ Tangible. Hệ thống giúp việc truyền tải thông tin về địa lý, xã hội trở nên trực quan, hấp dẫn hơn, thích hợp với việc trình diễn trước nhiều người.

Hệ thống gồm một bàn chiếu và 1 thiết bị di động như điện thoại thông minh, tablet,.. Thông qua thiết bị điện tử, người dùng có thể xem chi tiết hình ảnh tại một khu vực trên bản tương tác. Ngoài ra còn có thể xem các thông tin tăng cường khác như các loại bản đồ khác nhau, mức độ ô nhiễm không khí, ô nhiễm nguồn nước, xem thống kê mật độ dân số, bản đồ heatmap tại khu vực,... Hơn thế nữa, hệ thống còn cho phép người dùng thực hiện thống kê dữ liệu trên 1 tập các địa điểm khác nhau và cung cấp cho người dùng 1 loại các cách thức tương tác khác nhau.

Ngoài lĩnh vực cụ thể là về địa lý xã hội, hệ thống hứa hẹn còn có thể ứng dụng vào các lĩnh vực khác như sinh học, kinh tế,... Với thiết kế nhỏ gọn, lắp đặt nhanh chóng và dễ dàng, hệ thống có thể triển khai nhanh ở các trường học, showroom, khu triển lãm. Các thao tác phong phú và đa dạng sẽ giúp mang lại cảm giác ấn tượng, thu hút người sử dụng.

4.2. Cấu hình hệ thống



Hình 4.1 Cấu hình thiết lập của hệ thống

Error! Reference source not found. minh họa cấu hình tổng thể của hệ thống.
Các thành phần chính của hệ thống này bao gồm :

- Một khung bàn nhôm có chiều rộng 73cm và chiều dài là 73cm, bàn cao 103cm. Mặt bàn chỉ có viền cạnh và rỗng bên trong. Đặt phía trên khung là một tấm kính trong suốt và phía trên tấm kính là giấy trắng để hứng ảnh
- Một máy chiếu Sanyo PLC-XW60 với cường độ sáng 2000 Ansi Lumens, độ phân giải 1024x768, kích thước hình ảnh từ 80 đến 300 inches. Máy chiếu theo góc 90 độ hướng lên từ dưới chân bàn lên mặt kính trong, hình ảnh sẽ được giữ lại ở mặt giấy trắng, để hình ảnh thuận chiều với mắt người nhìn từ phía trên thì chúng em tiến hành lật màn hình phát ra từ máy chiếu theo chiều dọc. Với máy chiếu trên có cảm biến về góc chiếu nên chúng em đã điều chỉnh keystone để hình ảnh phát ra là hình chữ nhật với góc người nhìn, cuối cùng là điều chỉnh độ thu phóng và focus của thấu kính. Để hình ảnh phát ra hứng được là rõ nhất thì ngoài việc chỉnh độ focus của thấu kính

chúng em có 2 hướng giải quyết: Một là giảm ánh sáng môi trường bên ngoài tác động vào mặt bàn – Đây là cách giải quyết chúng em đang thực hiện – với mức sáng hiện tại chúng em đang sử dụng là 4 lux trở xuống (ánh sáng phòng thí nghiệm ban ngày không mở đèn phòng và đóng rèm là 8 lux). Hai là thay đổi chất liệu giấy hứng ánh để thu được hình ảnh tốt hơn, theo thực nghiệm loại giấy tốt nhất để hứng sáng là giấy can trong 83 gsm.

- Kinect 2 có khả năng quay video hồng ngoại 512x424 pixel, USB 3.0 để thu hình từ phía trên. Kinect được lắp cao cách 75cm so với mặt bàn, có góc nhìn sao cho lấy đủ mặt bàn và hình ảnh hứng được. Kinect được dùng để thu ảnh và nhận diện vật thể trong quá trình sử dụng.
- Thiết bị di động để hỗ trợ tương tác trong quá trình hoạt động của hệ thống. Ở đây chúng em sử dụng điện thoại Xiaomi có màn hình 5.5 inches Full HD, hệ điều hành Android 7.0, CPU Snapdragon 625 8 nhân 2.0 Hz, RAM 3 GB, chip đồ họa Adreno 506 và máy tính bản Huawei Mediapad T3 màn hình 8 inches, độ phân giải 1280x800 pixels, hệ điều hành Android 7.0, CPU Qualcomm MSM8917 1.4 Hz, RAM 2 GB, chip đồ họa Adreno 308.

Sau khi xây dựng theo các thông số trên, ta sẽ có được hệ thống thực tế như Hình 4.2.



Hình 4.2 Hình ảnh thực tế của hệ thống

4.3. Kịch bản sử dụng hệ thống

Hệ thống bao gồm 1 bàn tương tác với hình ảnh chiếu ra từ 1 máy tính đóng vai trò là trung tâm xử lý. Ở lần đầu tiên sử dụng hệ thống, người dùng cần thực hiện lại quá trình căn chỉnh tọa độ để hệ thống ánh xạ tọa độ giữa các không gian tọa độ khác nhau. Sau quá trình căn chỉnh, người dùng có thể dùng các thiết bị thông minh như điện thoại, máy tính bảng đã được cài sẵn ứng dụng để tương tác. Thông qua các thiết bị di động, người dùng có thể giao tiếp với hệ thống và sử dụng các chức năng của hệ thống.

4.4. Một số vấn đề gặp phải và kỹ thuật áp dụng

4.4.1. Cách bố trí máy chiếu



Vấn đề

Bài toán cần giải quyết là lắp đặt vị trí máy chiếu làm sao cho thích hợp để thỏa mãn 2 tiêu chí:

- Chi phí lắp đặt rẻ, dễ và nhanh chóng lắp đặt
- Nâng cao trải nghiệm người dùng, dễ sử dụng và đẹp mắt

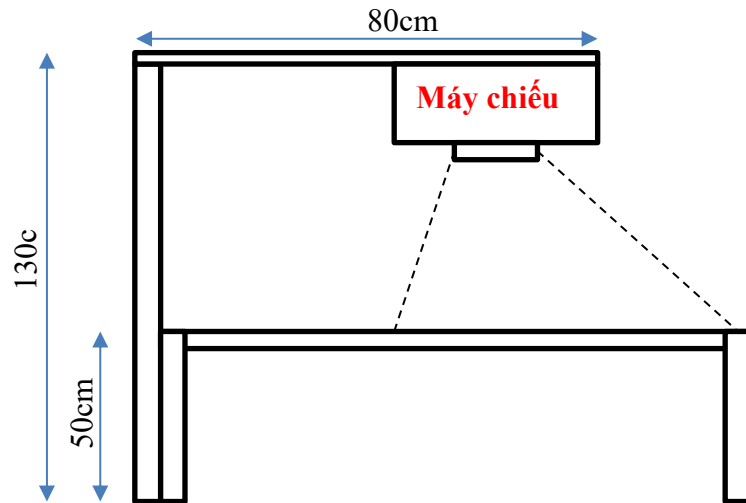


Giải pháp

Qua phân tích và thử nghiệm thực tế, chúng em đề xuất 3 cách bố trí máy chiếu khả thi nhất

- Cách 1: Đặt máy chiếu trên cao, chiếu từ trên xuống
- Cách 2: Đặt máy chiếu dưới mặt đất, phương vuông góc với mặt đất, chiếu từ dưới lên
- Cách 3: Đặt máy chiếu dưới đất, áp dụng phản xạ gương để tạo góc chiếu 90° so với mặt đất

4.4.1.1. Đặt máy chiếu trên cao



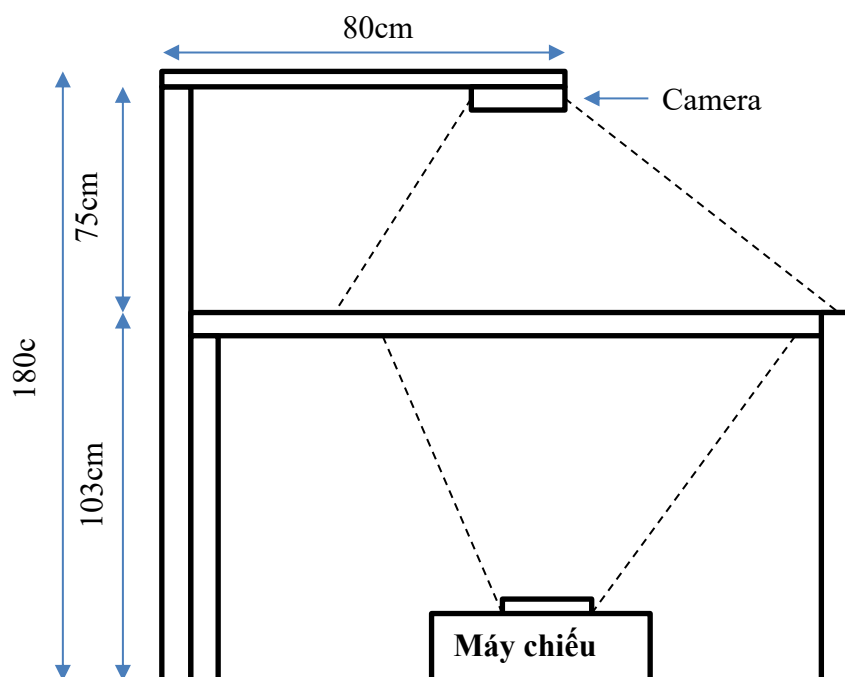
Hình 4.3 Cách bố trí máy chiếu trên cao

Thông số lắp đặt cụ thể được mô tả ở Hình 4.3. Với cách lắp đặt này, ta tận dụng được giá treo hiện tại của Camera. Hình ảnh chiếu ra rất rõ nét, chân thật và sống động, mang đến cho người dùng trải nghiệm rất tốt.

Tuy nhiên, nhược điểm của cách lắp đặt này là ta cần gia cố giá đỡ thêm chắc chắn vì máy chiếu khá nặng. Hơn thế nữa, việc lắp đặt khá khó khăn do phải lắp đặt cố định để máy chiếu không rung lắc trong quá trình sử dụng. Việc lắp đặt cứng này gây khó khăn nếu về sau có nhu cầu thay đổi, nâng cấp về phần cứng.

Ngoài ra, việc chiếu ánh sáng từ trên cao xuống sẽ gây nên vấn đề là gây đổ bóng lên mặt bàn khi người dùng thao tác dưới máy chiếu, làm cho thông tin chiếu ra bị bóp méo, sai lệch với kích thước ban đầu.

4.4.1.2. Đặt máy chiếu dưới đất theo phương vuông góc mặt đất

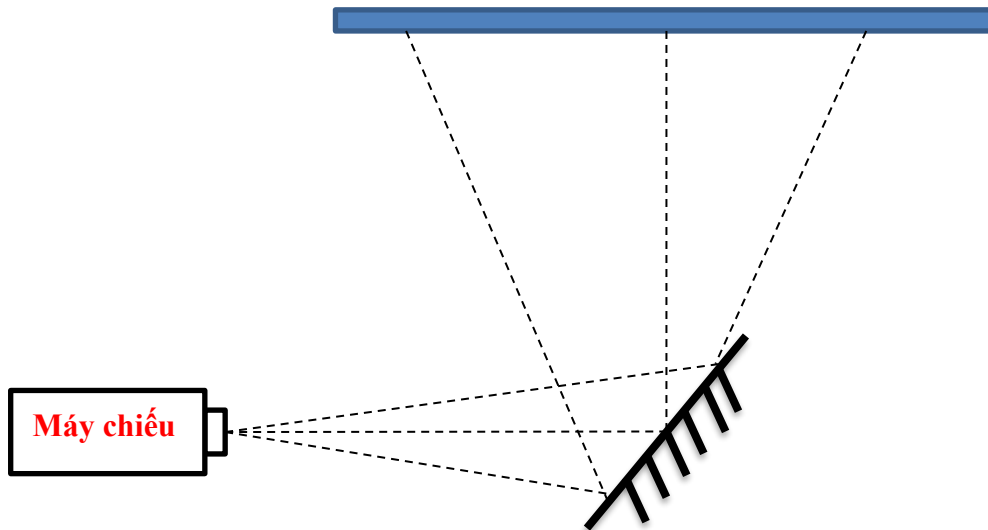


Hình 4.4 Cách bố trí máy chiếu dưới đất theo phương vuông góc

Theo cách bố trí ở Hình 4.4, ta thấy rõ ràng cách lắp đặt này đã khắc phục được nhược điểm gây độ bóng của cách lắp đặt trước. Khung giá đỡ cũng không cần phải gia cố thêm, giúp tiết kiệm chi phí. Do mặt bàn trong trường hợp này là kính trong suốt có phủ giấy mỏng, nên hình ảnh chiếu ra từ máy chiếu tuy không sắc sảo, thu hút như cách lắp đặt trước, nhưng vẫn hiển thị rõ thông tin ở mức chấp nhận được. Nếu thay đổi chất liệu kính trong suốt và giấy trắng thành kính đục để trình chiếu thì chất lượng ảnh sẽ tốt hơn nhiều. Tuy nhiên, trong giới hạn luận văn này, chúng em chỉ tiến hành thí nghiệm trên kính trong suốt và giấy A4.

Tuy nhiên, cách bố trí này có khuyết điểm là việc giữ máy chiếu ở góc 90° gây ra một số khó khăn như : mặt tản nhiệt của máy nằm ở phần sau, nếu trình chiếu trong thời gian dài sẽ làm nóng máy do mặt tản nhiệt áp xuống đất, cần làm thêm một giá đỡ để giữ máy chiếu cố định. Ngoài ra vùng chiếu ra của máy chiếu nhỏ nên nếu muốn hứng được ảnh lớn hơn, ta phải nâng cao khoảng cách màn hứng và máy chiếu. Việc này khiến tổng chiều cao của cả ứng dụng đồ sộ hơn.

4.4.1.3. Đặt máy chiếu dưới đất và áp dụng phản xạ gương



Hình 4.5 Cách bố trí máy chiếu dưới đất kém theo gương

Với cách bố trí như hình Hình 4.5, cơ bản cách bố trí này cho kết quả như là cách ở mục 4.4.1.2, nhưng có cải tiến về độ hiệu quả của của phương pháp chiếu. Việc áp dụng tính chất phản xạ của gương giúp ta có thể điều chỉnh độ rộng của vùng chiếu ra mà không cần phải tăng khoảng cách của mặt bàn và máy chiếu. Ngoài ra máy chiếu có thể tản nhiệt tốt hơn khi đặt nằm.

Theo thực nghiệm, chúng em chọn góc đặt gương là góc 45° sẽ thi được hình ảnh to hơn trên màn hứng mà chất lượng ảnh vẫn không đổi.

4.4.1.4. Kết luận

Theo qua trình phân tích và thực nghiệm ở cả 3 phương pháp trên, chúng em quyết định chọn phương pháp thứ 2: Đặt máy chiếu dưới đất theo phương vuông góc 90° để tiến hành xây dựng ứng dụng. Phương pháp 1 không mang lại trải nghiệm tốt do vấn đề đổ bóng. Phương pháp 2 và 3 gần tương đương nhau về thời gian lắp đặt và độ hiệu quả sử dụng. Tuy phương pháp 3 có thể tối ưu phần cứng của hệ thống hơn nhưng cần phải tính toán và thực nghiệm nhiều hơn. Nên chúng em sẽ chọn phương pháp 2.

Về bản chất, việc dùng máy chiếu và màn hứng để giả lập lại một màn chiếu tivi, hoặc màn hình cảm ứng kích thước lớn, nhưng do giá thành các thiết đó quá đắt đỏ, nên việc dùng phương pháp này sẽ giảm chi phí thực hiện đi rất nhiều lần.

4.4.2. Kỹ thuật ánh xạ tọa độ giữa các không gian tọa độ



Vấn đề

Với cấu hình hệ thống hiện tại như Hình 4.1, camera sẽ ghi hình các thao tác của người dùng và chuyển ảnh đó về module xử lý, máy chiếu sẽ chiếu ra hình ảnh sau khi được xử lý. Như vậy hiện đang tồn tại 2 hệ tọa độ:

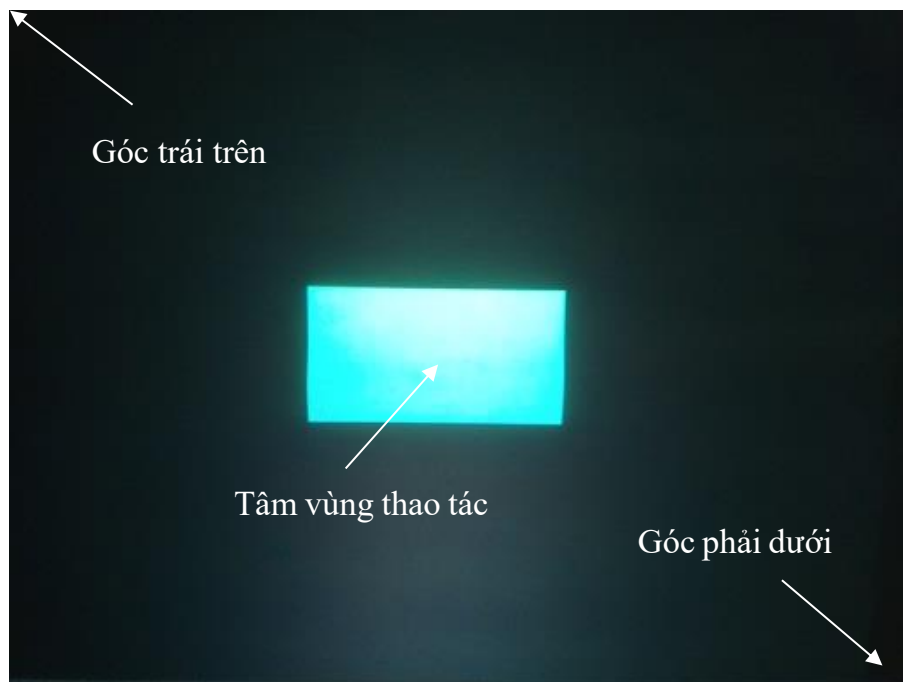
- Tọa độ trên ảnh ghi được bởi camera, chiều dài và chiều rộng bằng với kích thước mà camera trả về, đơn vị là *pixel*
- Tọa độ trong môi trường Unity

Vấn đề đặt ra là làm sao ánh xạ tọa độ giữa 2 hệ trục tọa độ này ?



Giải pháp

Qua quá trình nhận dạng vật thể, ta lấy được tọa độ tâm của vật thể đó trên bức ảnh theo đơn vị pixel, góc tọa độ của Frame ảnh đó ở góc trái trên. Nhưng ở Unity, để làm việc thì tọa độ tương đối giữa các Object được tính từ tâm của Object này đến Object kia, chính vì thế cần có một công thức để ánh xạ tọa độ nhận được sang tọa độ cần xử lý ở Unity.



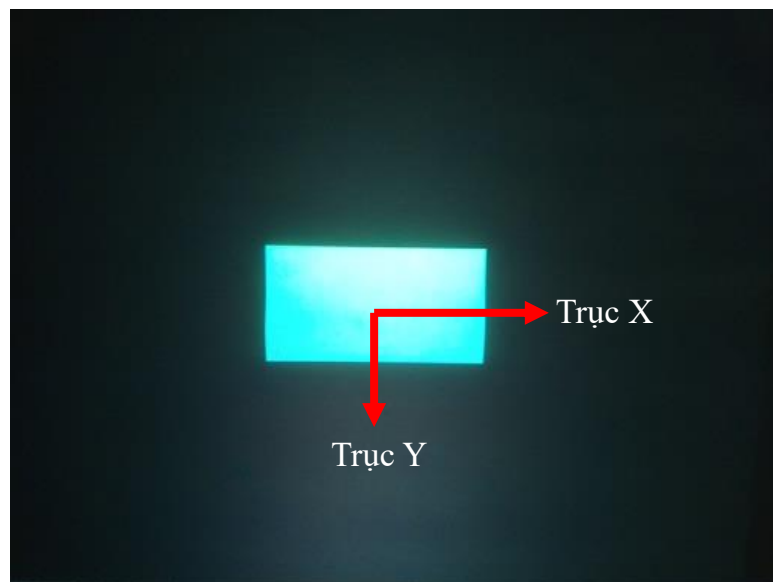
Hình 4.6 Xác định vùng thao tác của hệ thống

Đầu tiên, chúng em chuyển góc tọa độ ở góc trái trên bức ảnh thu được vào tâm của vùng thao tác để giống với vị trí góc tọa độ trong Unity. Chúng em đo tọa độ của góc trái trên và góc phải dưới của vùng screen (vùng thao tác) sẽ được phát ra theo tọa độ trên tấm ảnh (Hình 4.6). Từ đó có thể tính ra được tọa độ tâm của vùng thao tác và dùng một phép tịnh tiến tọa độ để tính tọa độ của một điểm theo góc tọa độ mới như sau:

Gọi (x_0, y_0) là tọa độ tâm của vùng thao tác ở hệ tọa độ cũ, (x, y) là tọa độ của một điểm trong hệ tọa độ cũ, (X, Y) là tọa độ của điểm đó trong hệ tọa độ mới thì ta có:

$$\begin{cases} X = x - x_0 \\ Y = y - y_0 \end{cases}$$

Sau khi đã chuyển hệ tọa độ, chúng em tiến hành mapping tọa độ trên bức ảnh ở hệ trục mới, với tọa độ của màn hình trong Unity. Tư tưởng tiến hành mapping như sau: Đầu tiên chúng em tạo 1 GameObject có màu đỏ tại tâm của vùng thao tác, với kích thước 4cm x 4cm. Chúng em dùng Blob để gửi tọa độ ảnh của Object đó liên tục từ frame hình thu được qua socket vào Unity, với mọi tọa độ nhận được thì tiến hành chuyển hệ trục như công thức trên (vẫn là tọa độ ảnh).



Hình 4.7 Lấy mẫu Calibration theo 2 trục X, Y

Chúng em lấy 9 điểm tọa độ theo trục X để mapping là $(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0), (8, 0)$; 6 điểm theo trục Y (vì tỉ lệ màn hình nên theo chiều dọc lấy được 6 điểm từ tâm màn hình) là $(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5)$ (Hình 4.7). Đây là các tọa độ màn hình trong Unity, với mỗi một tọa độ màn hình sẽ tương ứng với 1 tọa độ ảnh từ Blob gửi về. Để lọc nhiễu trong quá trình lấy tọa độ, thì với mỗi tọa độ màn hình, chúng em sẽ lấy 90 tọa độ ảnh từ Blob và lấy trung bình của 90 giá trị đó. Sau khi đã lấy hoàn tất 15 điểm thì chúng em tiến hành mapping theo công thức sau:

Gọi (X, Y) là tọa độ nhận được và đã qua việc chuyển tọa độ. (x_1, y_1) là tọa độ trong màn hình Unity tương ứng với tọa độ nhận được, chúng em thiết lập công thức sau:

$$\begin{cases} X = a + \Delta x \cdot x_1 \\ Y = b + \Delta y \cdot y_1 \end{cases}$$

Chúng ta cần tìm được $\Delta x, \Delta y, a, b$ trung bình. Với kết quả các cặp tọa độ thu được trong việc đo đạc phía trên, chúng em có thể tính được giá trị trung bình chấp nhận được của $\Delta x, \Delta y, a, b$. Từ đó suy ngược được công thức tính tọa độ 1 điểm trong Unity theo tọa độ nhận được như sau:

$$\begin{cases} x_1 = \frac{X - a}{\Delta x} \\ y_1 = \frac{Y - b}{\Delta y} \end{cases}$$

4.4.3. Kiến trúc hệ thống sự kiện liên hoàn



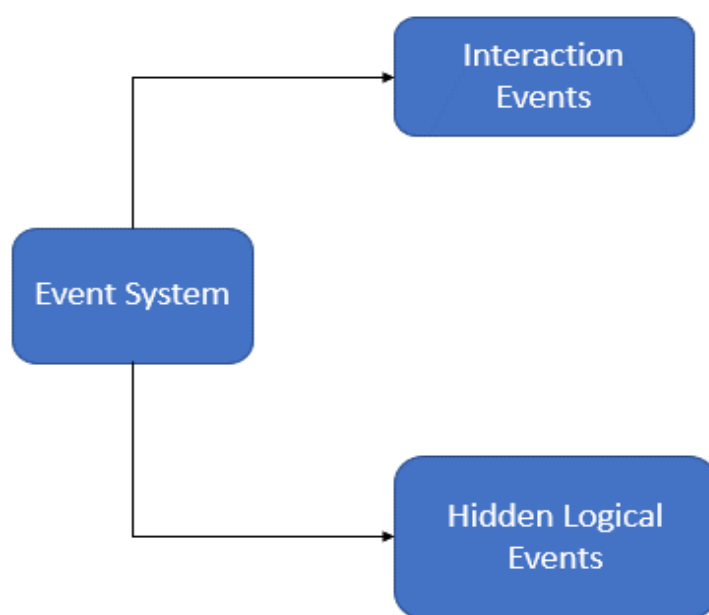
Đặt vấn đề

Để có thể phát triển hệ thống theo hướng mở rộng, tích hợp, phát triển nhanh chóng các tính năng về sau, ta cần xây dựng một kiến trúc có tính linh hoạt cao, không bị phụ thuộc vào các thành phần con của hệ thống. Một ví dụ cụ thể như sau: giả sử sau này ta phát triển được một hệ thống nhận dạng chính xác, hiệu quả hơn hệ thống hiện tại, ta chỉ cần thay thế module nhận dạng mà không cần chỉnh sửa các phần còn lại của hệ thống mà vẫn đảm bảo được toàn hệ thống hoạt động bình thường.



Giải pháp

Hệ thống sự kiện (Event System) được thiết kế với mục đích quản lí kịch bản hoạt động của hệ thống một cách hiệu quả và tách biệt các thành phần của hệ thống với nhau để chúng hoạt động độc lập và dễ dàng thay đổi, mở rộng. Các thành phần ở đây là việc xử lí một sự kiện xảy ra, và việc kiểm tra điều kiện để kích hoạt sự kiện đó. Nó cho phép việc tải nhanh một kịch bản và có thể thay đổi kịch bản đó một cách linh hoạt mà không phụ thuộc vào kiến trúc xử lí sự kiện bên trong. Bên cạnh đó hệ thống sự kiện cho phép việc kích hoạt liên hoàn sự kiện một cách dễ dàng. Hệ thống sự kiện không chỉ có kịch bản sự kiện ẩn liên hoàn mà còn bao gồm cả các sự kiện và việc xử lí sự kiện do người dùng tạo ra (như cách xử lí sự kiện thông thường). Ở đây chúng em tập trung vào việc xử lí liên hoàn sự kiện theo kịch bản. Minh họa hệ thống sự kiện như Sơ đồ 4-1.



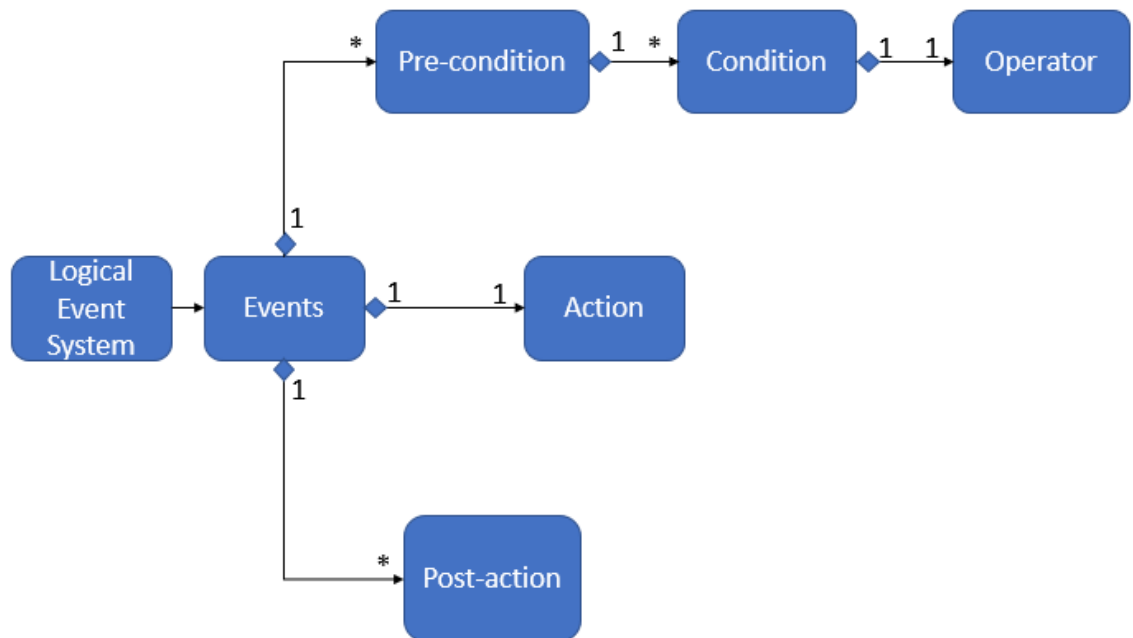
Sơ đồ 4-1 Hệ thống sự kiện

4.4.3.1. Thành phần của hệ thống sự kiện liên hoàn

Một hệ thống sự kiện liên hoàn có các thành phần sau :

- Một mảng các đối tượng Event, mỗi Event tương ứng với một bộ các hành động tác động lên các Object tùy thuộc vào kịch bản.
- Một Dictionary chứa các cờ hiệu (Flags), trong quá trình hoạt động, chương trình sẽ duyệt Dictionary này một cách xuyên suốt. Nếu có một tổ hợp cờ nào phù hợp với một Event tương ứng thì Event đó sẽ được kích hoạt.
- Một Dictionary các Object sẽ bị tác động trong kịch bản, ở đầu chương trình sẽ tiến hành đăng kí các Object này vào Dictionary.
- Một kịch bản đầu vào ở dạng JSON File, mô tả các Event. Cấu trúc gồm tên của Event, Action, Pre-conditions, Post-conditions.

4.4.3.2. Cấu trúc của một hệ thống sự kiện liên hoàn

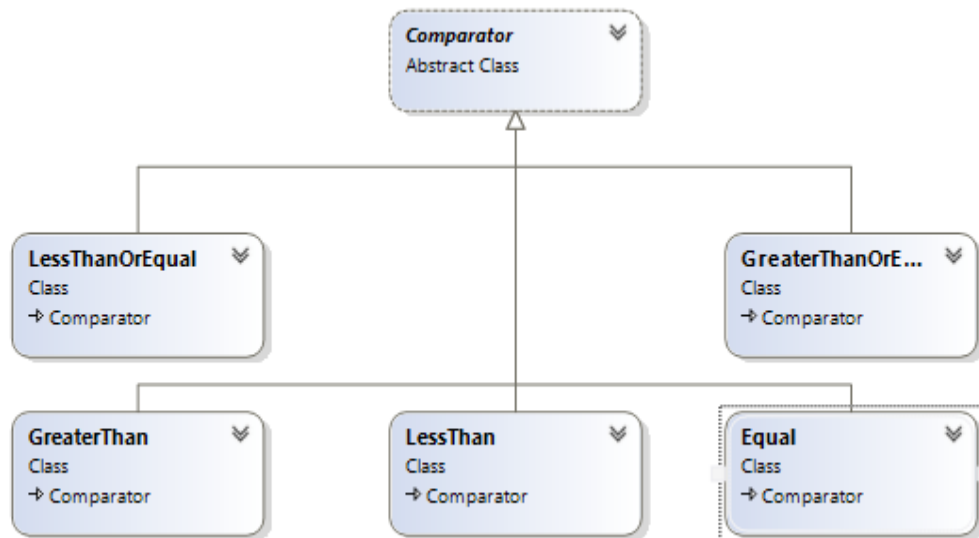


Sơ đồ 4-2 Hệ thống sự kiện liên hoàn

Sơ đồ 4-2 mô tả mối quan hệ giữa các thành phần trong hệ thống Event, chi tiết các thành phần như sau :

- **Pre-conditions:** Tập hợp các điều kiện đầu vào, một Pre-condition gồm các Conditions. Sự kiện được kích hoạt khi một Pre-condition thỏa.
- **Conditions:** Là một chuỗi các điều kiện được nối với nhau bằng toán tử &&, một Pre-condition được kích hoạt khi chuỗi các điều kiện đó đều thỏa.

- **Comparators:** Là các toán tử so sánh dùng để so sánh các cờ hiệu với một giá trị nào đó, nhằm kích hoạt Conditions từ đó kích hoạt Pre-condition và cuối cùng là kích hoạt sự kiện. Các toán tử này bao gồm: Equal, GreaterThan, LessThan, LessThanOrEqual, GreaterThanOrEqual. Các toán tử này kế thừa Abstract Class Comparator và được tạo ra một lần duy nhất ở đầu chương trình để sử dụng (Sơ đồ 4-3). Nói cách khác đây là một Global Helper Class.



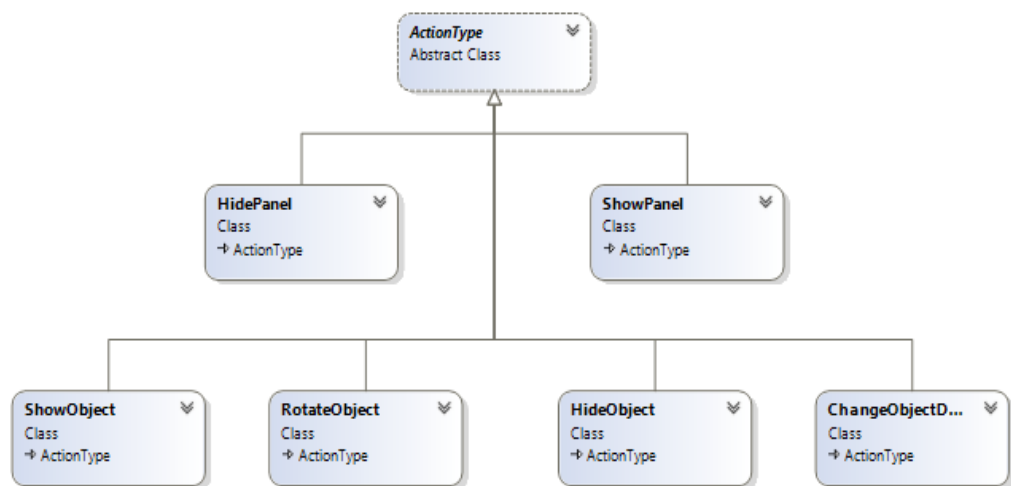
Sơ đồ 4-3 Sơ đồ lớp của Comparator

```

public class GreaterThanOrEqual : Comparator
{
    public override bool compare(string FlagId, int value)
    {
        return Global.getFlag(FlagId) >= value;
    }
}
  
```

Mã 1 Minh họa một Comparator

- **Action:** Là một hành động sẽ tác động lên một GameObject trong hệ thống khi sự kiện được kích hoạt (ví dụ như ẩn, hiện, di chuyển vị trí, thu nhỏ, phóng to,...). Các class này kế thừa class Action Type là Base Abstract Class cho các Action này (Sơ đồ 4-4).



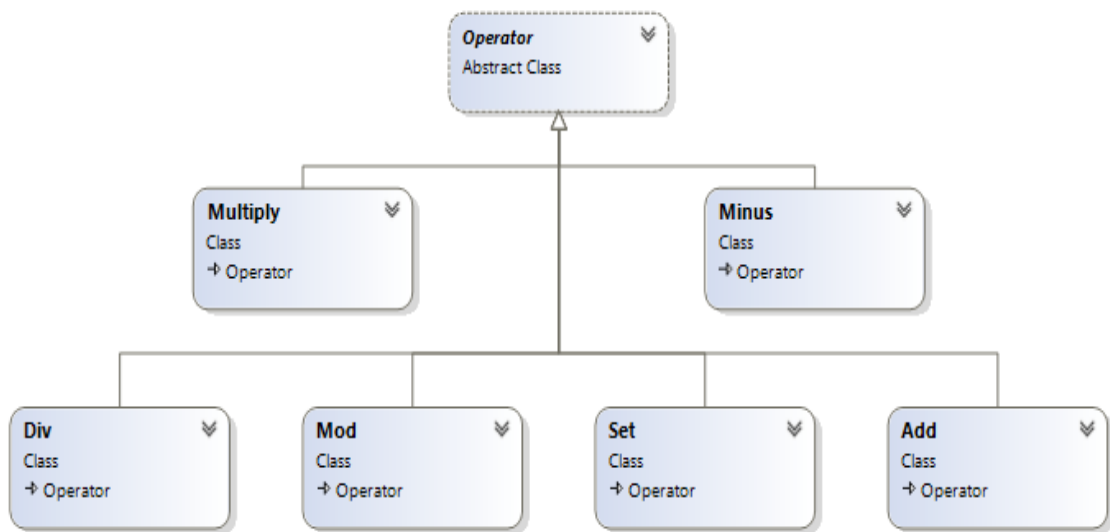
Sơ đồ 4-4 Sơ đồ lớp của Action

```

public class RotateObject : ActionType
{
    public override void DoAction(GameObject gameObject,
ParamsRaw paramsRaw)
    {
        gameObject.transform.Rotate(0, 0, 30 * Time.deltaTime,
Space.World);
    }
}
  
```

Mã 2 Minh họa một Action

- **Post-actions:** Là chuỗi các hành động dùng để cập nhật các phần tử của Dictionary Flags sau khi Action của Event được thực hiện. Đây là cơ chế để thực hiện việc kích hoạt liên hoàn sự kiện trong hệ thống. Vì Dictionary Flags sẽ luôn được thăm qua mỗi lần lặp Update, dựa vào trạng thái cờ mà sẽ kích hoạt sự kiện, nếu một sự kiện được kích hoạt, sau đó Post-actions cập nhật lại Dictionary Flags tạo ra một tổ hợp trạng thái mới kích hoạt một sự kiện khác, từ đó tạo ra hiệu ứng liên hoàn.
- **Operator:** Mỗi hành động trong Post-actions là một Operator (cộng, trừ, nhân, chia, gán,...) tác động đến giá trị của Dictionary Flags như Sơ đồ 4-5.



Sơ đồ 4-5 Sơ đồ lớp của Operator

```

public class Mod : Operator
{
    public override int Operate(string FlagId, int value)
    {
        return Global.setFlag(FlagId, Global.getFlag(FlagId) % value);
    }
}
  
```

Mã 3 Minh họa một Operator

4.4.4. Giao tiếp giữa các thiết bị tương tác

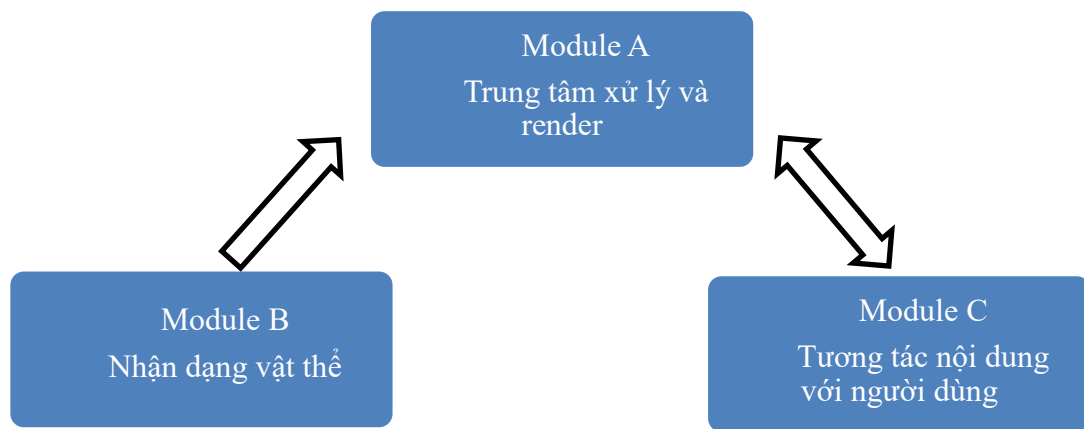


Đặt vấn đề

Một hệ thống thường được chia làm nhiều module nhỏ khác nhau để dễ quản lý và thuận tiện cho việc bảo trì, nâng cấp về sau. Việc chia nhỏ các module làm phát sinh một vấn đề là vấn đề giao tiếp giữa chúng. Vậy các module đây giao tiếp với nhau như thế nào, thông qua phương thức nào ?



Giải pháp



Sơ đồ 4-6 Cấu trúc module của hệ thống

Theo như kiến trúc ứng dụng trên được trình bày ở Sơ đồ 4-6, ứng dụng chia ra thành các module độc lập với nhau, mỗi module chịu một trách nhiệm riêng biệt như: module xử lý dữ liệu và render hình ảnh (A), module nhận dạng vật thể (B), module tương tác nội dung với người dùng như điện thoại, máy tính bảng (C). Thêm vào đó, các module cần một phương thức để giao tiếp, truyền đạt dữ liệu qua lại lẫn nhau một cách hiệu quả.

Trong giới hạn luận văn này, ứng dụng được triển khai trên môi trường mạng cục bộ LAN, nên chúng em quyết định dùng phương thức Socket để giao tiếp giữa các module. Socket là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ (client) và một chương trình cung cấp dịch vụ (server) trên mạng LAN, WAN hay Internet.

Module (A) và (B) chạy cùng trên một máy tính và (B) đóng vai trò là client, chỉ có nhiệm vụ gửi tọa độ truy vết được cho (A) xử lý. (B) được viết bằng ngôn ngữ Python và dùng thư viện OpenCV để hỗ trợ việc truy vết. (A) đóng vai trò là server với (B) và chỉ tiếp nhận dữ liệu từ (B) nếu có và không có phản hồi lại. (A) sẽ lấy dữ liệu tọa độ truy vết được từ (B) để tiến hành căn chỉnh tọa độ phục vụ cho việc render hình ảnh.

Module (A) và (C) chạy trên 2 thiết bị khác nhau và (A) đóng vai trò là server, (C) là client, giao tiếp với nhau theo kịch bản

Ở đây, module (B) chỉ có nhiệm vụ gửi vị trí tọa độ vật thể truy vết được 1 cách liên tục mà không cần quan tâm việc phản hồi, nên chúng em đã dùng phương thức gửi UDP, cụ thể là dùng tham số socket.SOCK_DGRAM để thiết lập việc gửi theo hình thức UDP. Phương thức gửi UDP tuy không thể xác nhận việc gửi có thành công hay không, và giới hạn kích thước gói tin gửi bị giới hạn, tuy nhiên tốc độ gửi sẽ nhanh hơn nhiều so với TCP. Ở ngữ cảnh này, việc dùng UDP là hợp lý.

4.4.5. Xử lý hành động tương tác từ phía người dùng



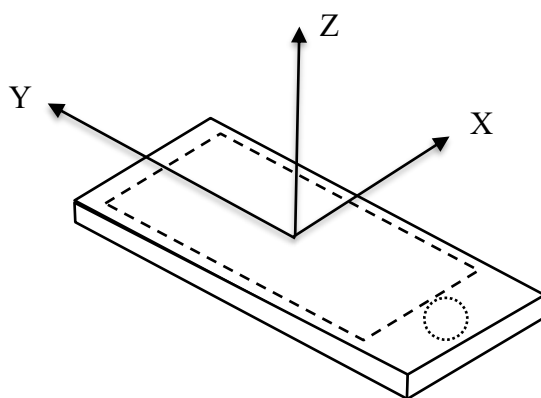
Đặt vấn đề

Với những chức năng đã đề ra, thì người dùng sẽ tương tác với hệ thống chủ yếu thông qua thiết bị di động. Tuy nhiên, không thể sử dụng tương tác chạm truyền thống làm phương pháp tương tác chủ yếu, điều đó sẽ làm mất đi cảm giác thật khi tiếp xúc với môi trường tangible. Chính vì thế cần có những cách tương tác mới hơn để giải quyết vấn đề này.



Giải pháp

Để tạo ra những cử chỉ mới để tự nhiên hóa tương tác từ phía người dùng, chúng em đã nghiên cứu hệ thống cảm biến trên thiết bị di động. Với sự hỗ trợ của hệ thống cảm biến, mà đặc biệt ở đây là cảm biến gia tốc, chúng em đã đề xuất các hành động mới cho cử chỉ tương tác từ phía người dùng.



Hình 4.8 Hệ tọa độ của hệ thống cảm biến trên thiết bị di động

Với hệ tọa độ mà cảm biến gia tốc nhận được (**Error! Reference source not found.**) kết hợp với cảm biến nhận diện được tư thế úp, ngửa, đứng đứng của thiết bị, chúng em đã định nghĩa 3 cử chỉ hỗ trợ cho hệ thống của chúng em như sau:

- Flip: Với điều kiện thiết bị đang đặt nằm trên mặt phẳng ngang, màn hình đang ngửa ở vị trí ban đầu, sau đó màn hình bị lật úp xuống (xoay theo trục Y của hệ thống cảm biến gia tốc). Khi thỏa mãn các điều kiện trên thì thiết bị nhận diện đó là một cử chỉ Flip và gửi tín hiệu đến máy chủ để thay đổi loại bản đồ đang hiển thị trên mặt bàn.
- Shake: Với điều kiện thiết bị đang đặt nằm trên mặt phẳng ngang, màn hình đang ngửa, người dùng lắc đột ngột thiết bị theo trục X theo chiều âm hoặc chiều dương với giá trị gia tốc trả về có trị tuyệt đối lớn hơn 1.5. Khi thỏa mãn các điều kiện trên thì thiết bị nhận diện đó là một hành động Shake và gửi tín hiệu đến máy chủ để xóa bỏ trạng thái hiện tại trên thiết bị (với hệ thống thì khi Shake sẽ cho thiết bị nhận lại tọa độ đang dò trên mặt bàn sau khi Freeze).
- Erect: Với điều kiện thiết bị đang nằm trên mặt phẳng ngang, màn hình đang ngửa, người dùng dựng thiết bị lên với hướng màn hình về phía người dùng (xoay theo trục X của hệ thống cảm biến gia tốc). Khi thỏa mãn các điều kiện trên thì thiết bị nhận diện đó là một cử chỉ Erect và gửi tín hiệu đến máy chủ để hiển thị thông tin chi tiết (với hệ thống là hiển thị thông tin biểu đồ về độ sạch không khí, nguồn nước, tiếng ồn tại một địa điểm).

4.5. Các chức năng của hệ thống

Hệ thống bản đồ sẽ cung cấp cho người dùng các chức năng cụ thể sau :

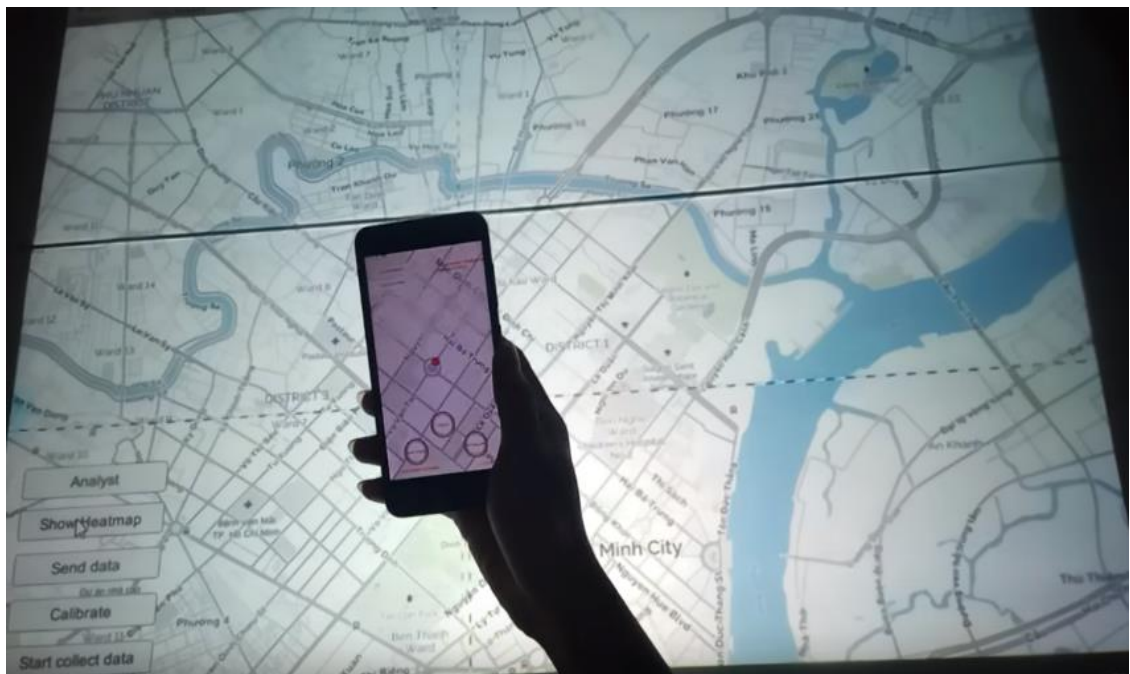
- Phóng to hình ảnh tại một địa điểm trên mặt bàn, hình ảnh được phóng to sẽ hiển thị trên màn hình điện thoại
- Thực hiện Shake để phóng to một vị trí bất kì tương ứng với vị trí của điện thoại trên mặt bàn.
- Thực hiện Freeze để ngừng phóng to hình ảnh theo vị trí điện thoại, người dùng có thể mang điện thoại ra không gian khác để tương tác với địa điểm hiện hành, chẳng hạn như xem các thông tin về dân số, tình trạng nhập cư,...
- Thực hiện Flip điện thoại để thay đổi loại hiển thị bản đồ như bản đồ giao thông, bản đồ vệ tinh,...
- Cho phép xem bản đồ heatmap mức độ ô nhiễm không khí của vùng lãnh thổ hiện tại.

- Cho phép xem camera giao thông tại thời điểm hiện tại tại một vị trí cụ thể trên bản đồ.
- Cho phép đánh dấu 1 tập các vị trí trên bản đồ, sau đó thực hiện thống kê trên tập địa điểm đó.

4.5.1. Xem hình ảnh tại một địa điểm

Sau khi khởi động hệ thống, mặt bàn sẽ hiển thị hình ảnh bản đồ tổng quát của một vùng lãnh thổ.

Người dùng dùng điện thoại đưa vào vùng không gian của bàn tương tác, thực hiện hành động Shake điện thoại, lập tức màn hình điện thoại sẽ hiển thị hình ảnh được phóng to tương ứng với vị trí của đặt của điện thoại trên mặt bàn. Người dùng có thể nhấn vào nút Freeze trên màn hình để ngưng nhận tín hiệu từ hệ thống gửi đến. Khi ở trạng thái Freeze, hình ảnh được phóng to cuối cùng sẽ được cố định. Người dùng có thể mang điện thoại sang 1 nơi khác làm việc và tương tác với địa điểm đó, chẳng hạn như xem các thống kê về địa điểm như thống kê về dân số, mức độ ô nhiễm không khí,..



Hình 4.9 Xem chi tiết hình ảnh bản đồ tại một địa điểm

4.5.2. Tương tác với dữ liệu theo thời gian thực

Hệ thống cho phép người dùng có thể xem camera giao thông ở thời điểm hiện tại tại một vị trí cụ thể. Người dùng chỉ cần lia điện thoại đến các vị trí gắn các marker trên mặt bàn, điện thoại lập tức sẽ mở kênh video trực tiếp giao thông tại vị trí đó.

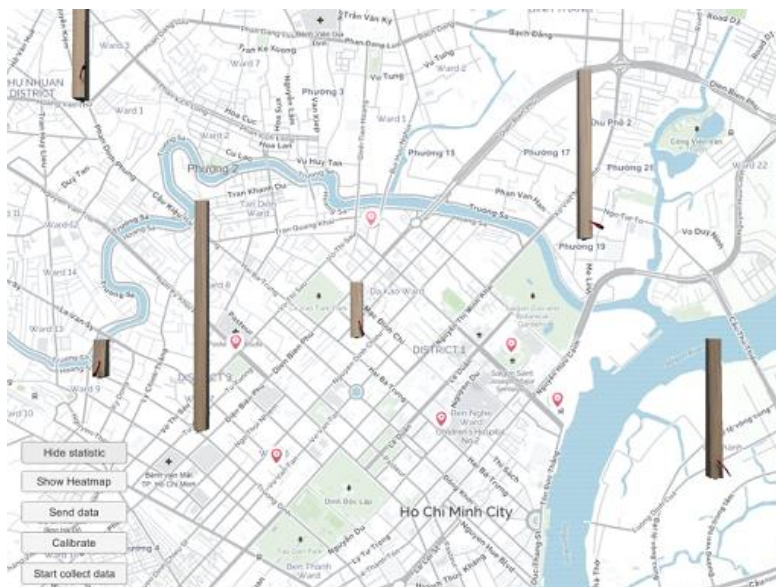
Ngoài ra hệ thống còn cho phép người dùng xem thống kê dữ liệu dưới dạng bản đồ nhiệt (Heat map). Do trong giới hạn luận văn, nhóm chúng em không có dữ liệu thực tế thống kê về mặt như dân số, mức độ kẹt xe, ô nhiễm không khí theo thời gian thực, nên chúng em sẽ sinh ra ngẫu nhiên dữ liệu, rồi chuyển dữ liệu đó về dạng ảnh heat map rồi hiển thị lên.



Hình 4.10 Ảnh Heat map thể hiện mức độ ô nhiễm không khí

4.5.3. Thống kê dữ liệu

Ngoài các tính năng trên, hệ thống còn cho phép người dùng đánh dấu 1 tập các địa điểm trên bản đồ. Người dùng chỉ cần đưa điện thoại đến vị trí muốn đánh dấu và bấm chọn đánh dấu. Lập tức 1 lá cờ được đặt tại vị trí đó trên cả mặt bàn và điện thoại. Sau khi chọn 1 tập các địa điểm, người dùng có thể thực hiện thống kê, so sánh dữ liệu giữa các địa điểm đó 1 cách trực quan, sinh động.



Hình 4.11 Thống kê thông tin theo tập các địa điểm đã chọn

4.6. Kết luận

Hệ thống bản đồ hỗ trợ tương tác Tangible cho phép người dùng tương tác, thảo luận với nhau trên cùng 1 không gian với sự minh họa dữ liệu cụ thể trực quan. Ngoài ra, hệ thống còn cung cấp một loạt các thao tác tương tác giữa thiết bị di động và mặt bàn giúp kích thích người dùng, gây cảm giác hứng thú, cảm thấy mới lạ hơn trong cách giao tiếp.

Kích thước nhỏ gọn, dễ thi công, lắp đặt của hệ thống hoàn toàn phù hợp với các không gian như lớp học, khu triển lãm, khu vực thảo luận. Ngoài ra, dựa vào hệ thống phần cứng được xây dựng sẵn, ta hoàn toàn có thể thiết kế thêm nội dung để ứng dụng vào các lĩnh vực khác như sinh học, lịch sử

Về hướng phát triển về sau, hệ thống có thể áp dụng kết quả của máy học, học sâu vào việc nhận dạng vật thể, cử chỉ để tăng độ chính xác của hệ thống hiện tại và mở rộng, bổ sung thêm 1 số cách tương tác khác nhau. Thêm nữa, chúng em sẽ hỗ trợ thêm nhiều người dùng có thể tương tác cùng lúc để việc trao đổi, thảo luận thông tin có thể thực hiện hiệu quả hơn.

Chương 5

Hệ thống chia sẻ dữ liệu hỗ trợ tangible

✍ Chương 5 này chúng em sẽ giới thiệu các chức năng của hệ thống chia sẻ hình ảnh trên mặt bàn hỗ trợ Tangible. Người dùng có thể chia sẻ dữ liệu, cụ thể là hình ảnh thông qua 1 bàn tương tác.

5.1. Giới thiệu tổng quan

Nhu cầu chia sẻ hình ảnh giữa mọi người luôn là một nhu cầu cần thiết. Với phương pháp chia sẻ thông thường thì một người sẽ truyền hình ảnh từ thiết bị của mình đến một thiết bị khác. Việc chia sẻ này thường diễn ra ở quan hệ 1-1, để tăng tính cộng tác trong một nhóm người, và tăng sự tự nhiên trong việc chia sẻ hình ảnh thì chúng em đã thiết kế Hệ thống chia sẻ hình ảnh trên mặt bàn hỗ trợ tangible. Mọi người có thể cùng xem ảnh từ một người trên mặt bàn, với những ai thích ngắm ảnh đó có thể lấy trực tiếp tại đó. Bên cạnh đó, hệ thống còn có thể sử dụng để thực hiện một lời nhắn với chức năng đính kèm chú thích vào tấm ảnh được gửi.

5.2. Cấu hình hệ thống

Hệ thống chia sẻ hình ảnh sử dụng lại hệ thống đã được cài đặt và dựng sẵn ở Hệ thống bản đồ hỗ trợ ở **Error! Reference source not found..** Với kịch bản khác được thiết kế cho việc chia sẻ hình ảnh qua lại giữa các thiết bị của người dùng.

5.3. Kịch bản sử dụng hệ thống

Hệ thống gồm 2 thành phần chính là máy chủ xử lý và thiết bị di động của người dùng. Trước khi sử dụng hệ thống thì các thiết bị di động cần Connect vào IP của máy chủ. Máy chủ xử lý và mặt bàn đóng vai trò là môi trường trung gian để truyền ảnh. Chương trình điều khiển trên thiết bị di động của người dùng hỗ trợ 2 chức năng chính là truyền ảnh và lấy ảnh. Người dùng đầu tiên sử dụng chức năng truyền ảnh, lựa chọn ảnh muốn truyền trong danh sách ảnh của mình và kèm thêm một chú thích nếu có. Khi đã chọn xong ảnh thì người dùng đó úp thiết bị xuống về phía mặt bàn và giữ mạnh để truyền ảnh xuống máy chủ, máy chủ hiển thị ảnh lên mặt bàn. Với những người dùng khác cùng Connect vào hệ thống, nếu ai muốn lấy ảnh đó về thiết bị của

mình thì có thể chuyển sang chế độ lấy ảnh. Với hành động hất điện thoại lên thì thiết bị đó sẽ được máy chủ gửi ảnh vào thiết bị.

5.4. Một số vấn đề gặp phải và kỹ thuật áp dụng

5.4.1. Cách bố trí máy chiếu

Cách bố trí máy chiếu và mặt bàn được xây dựng như phần **Error! Reference source not found.**

5.4.2. Xử lý hành động tương tác từ phía người dùng



Đặt vấn đề

Với mục tiêu trao việc nhận diện tương tác từ phía người dùng cho thiết bị di động của người dùng đó. Việc cần quan tâm ở đây là làm sao nhận được hành động được định nghĩa trước đã được thực hiện bởi người dùng.



Giải pháp

Với sự hỗ trợ của hệ thống phần cứng trên thiết bị, chúng em đã định nghĩa hành động cần nhận biết bằng việc theo dõi hệ thống cảm biến gia tốc trên thiết bị như Hình 4.8**Error! Reference source not found..**

Hệ thống hỗ trợ 2 hành động chính là giữ xuống (Shake Down) và hất lên (Shake Up). Cả 2 hành động này đều xử lý giá trị của cảm biến gia tốc theo trục Z, bên cạnh đó nhận diện thiết bị đang ngửa màn hình lên hay úp màn hình xuống. Hoạt động của cảm biến theo trục Z như sau:

- Khi màn hình ngửa: việc lắc mạnh thiết bị theo trục Z dù là lắc lên hay lắc xuống đều trả về giá trị gia tốc là âm.
- Khi màn hình úp: việc lắc mạnh thiết bị theo trục Z dù là lắc lên hay lắc xuống đều trả về giá trị gia tốc là dương.

Lợi dụng điều này kèm theo việc có thể nhận diện tư thế úp hay ngửa của thiết bị, chúng em đã định nghĩa 2 hành động Shake Up và Shake Down và hoạt động của 2 hành động đó trong hệ thống như sau:

- Shake Up: Với điều kiện thiết bị đang ở chế độ lấy ảnh, mặt thiết bị đang ngửa và người dùng hất đột ngột thiết bị theo trục Z lên phía trên với giá trị gia tốc trả về từ thiết bị lớn hơn 1.5 (giá trị gia tốc vừa phải không dùng quá

hiều sức theo thí nghiệm của nhóm chúng em). Khi thỏa mãn các điều kiện trên thì thiết bị gửi một tín hiệu yêu cầu lấy ảnh đến máy chủ. Máy chủ nhận được tín hiệu yêu cầu sẽ gửi lại 1 tín hiệu kèm theo thông tin về ảnh đó lại cho thiết bị đó.

- Shake Down: Với điều kiện thiết bị đang ở chế độ truyền ảnh, mặt thiết bị đang úp xuống và người dùng giữ đột ngột thiết bị theo trục Z xuống dưới với giá trị gia tốc trả về từ thiết bị bé hơn -1.5. Khi thỏa mãn các điều kiện trên thì thiết bị gửi tín hiệu kèm theo thông tin ảnh là tấm ảnh đang được hiển thị trong list ảnh và lời nhắn chú thích dưới ảnh nếu có đến máy chủ. Máy chủ nhận được tín hiệu sẽ hiển thị nội dung ảnh và nội dung text lên mặt bàn.

5.4.3. Giao tiếp giữa các thiết bị tương tác



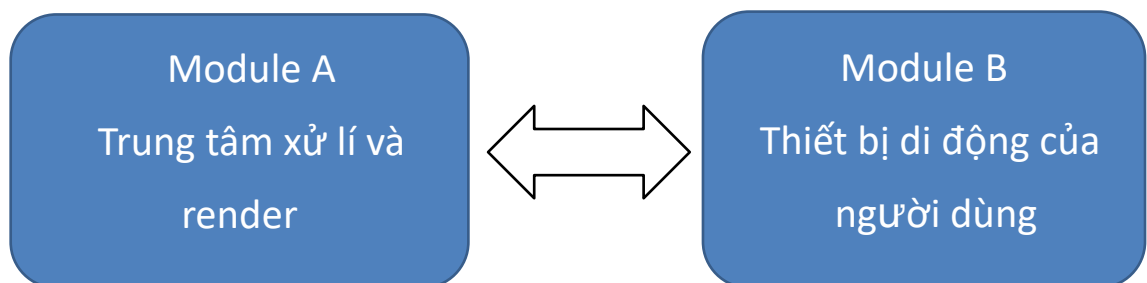
Đặt vấn đề

Với mục tiêu tương tác hỗ trợ cho đa người dùng cùng một lúc thì hệ thống cần xử lý việc nhận biết các thiết bị tương tác. Vậy làm thế nào để biết được những thiết bị nào đang kết nối, và thiết bị của người dùng nào đang thực sự tương tác, hay đang có yêu cầu tương tác?



Giải pháp

Cách tương tác giao tiếp giữa các thiết bị và trung tâm xử lý vẫn theo cách sử dụng truyền dữ liệu qua Socket như ở phần **Error! Reference source not found.** Tuy nhiên hệ thống có sự thay đổi về cấu trúc các module để phù hợp với chức năng được thể hiện với kịch bản trên ở phần 5.3.



Sơ đồ 5-1 Sơ đồ kiến trúc hệ thống

Theo kiến trúc ở Sơ đồ 5-1, ứng dụng được chia thành 2 module riêng biệt nhận 2 nhiệm vụ khác nhau: module xử lý dữ liệu và render hình ảnh (A), module hỗ trợ

người dùng tương tác với hệ thống như điện thoại, máy tính bảng (B). Phương thức giao tiếp và truyền đạt dữ liệu qua lại giữa các module sử dụng môi trường Socket, trong mạng cục bộ LAN như ở **Error! Reference source not found.**

Ở đây module (A) đóng vai trò là server xử lý và (B) đóng vai trò là client, (B) có nhiệm vụ phát hiện cử chỉ được hệ thống định nghĩa từ người dùng, sau đó gửi tín hiệu đến server là hành động nào đã được thực hiện, kèm theo các thông tin đi kèm (hình ảnh và tin nhắn). Cả (A) và (B) đều được viết bằng Unity, với module (A) vẫn sử dụng lại cấu trúc mã nguồn trong **Error! Reference source not found.** chỉ thay đổi lại kịch bản xử lý. (A) tiếp nhận dữ liệu về hành động được thực hiện và thông tin đi kèm từ (B) và thực hiện xử lý thông tin đó theo kịch bản.

Một điểm khác biệt ở hệ thống này là việc quản lý truy cập từ nhiều người dùng. Vì mục đích sử dụng của hệ thống là thiết kế theo kiểu bàn cộng tác nên cần phải quản lý được danh sách truy cập từ các thiết bị. Tất cả các thiết bị truy cập vào hệ thống đều được lưu lại IP vào một mảng. Khi có một tín hiệu đến, với tín hiệu truyền ảnh, thì (A) chỉ quan tâm đến thông tin được gửi đến là hình ảnh và tin nhắn mà không quan tâm đến IP gửi. (A) tiến hành render theo nội dung được nhận. Còn với tín hiệu yêu cầu lấy ảnh thì (A) nhận biết được thiết bị với mã IP nào đang yêu cầu và gửi nội dung đang được hiển thị trên mặt bàn đến đúng thiết bị có mã IP đó. Các tín hiệu truyền về đều có mã IP của thiết bị sở hữu IP đó nên việc quản lý đa thiết bị truy cập là có thể.

5.5. Các chức năng của hệ thống

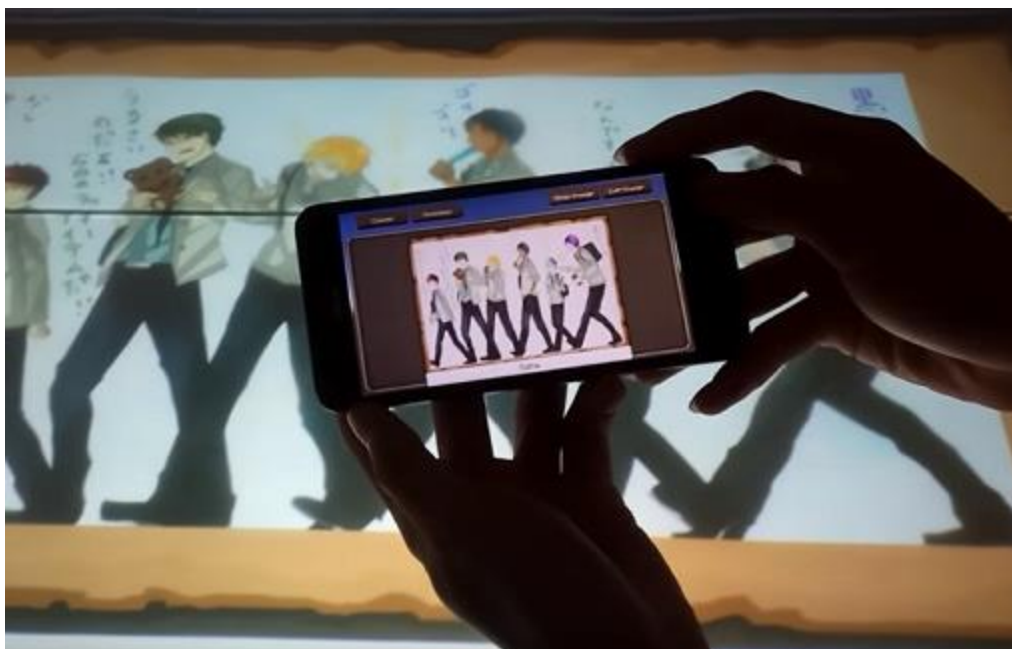
5.5.1. Các cách thức tương tác

Hệ thống hỗ trợ tương tác chủ yếu trên thiết bị di động. Với thiết bị di động thì người dùng sẽ tương tác với hệ thống bằng tương tác chạm truyền thống trên thiết bị và tương tác bằng cử chỉ do chúng em định nghĩa cho hệ thống đã được trình bày ở phần 5.4.2.

Với tương tác chạm thì người dùng sẽ tiến hành quét màn hình (Swipe) để chọn lựa hình ảnh trong danh sách trên thiết bị. Nhấn (Tap) để chọn nút bấm cho các chế độ điều khiển (truyền ảnh và lấy ảnh) và nhập ký tự tin nhắn sẽ gửi.

Với cử chỉ được hệ thống định nghĩa thì người dùng sẽ tiến hành giữ xuống để gửi ảnh đến máy chủ và hát lên để lấy ảnh từ máy chủ đang hiển thị trên mặt bàn.

5.5.2. Màn hình giao diện chính



5.6. Kết luận

Trong chương này nhóm chúng em đã thực hiện nghiên cứu và xây dựng hệ thống chia sẻ hình ảnh với mặt bàn hỗ trợ tangible và các thiết bị di động. Các thiết bị được kết nối qua socket tới server và thực hiện gửi các thông điệp tới nhau để thực hiện tương tác. Hệ thống xử lý cử chỉ tương tác từ phía người dùng để thực hiện thao tác truyền, nhận ảnh. Hệ thống hỗ trợ tương tác từ nhiều người dùng cùng với nhau.

Hệ thống có thể được phát triển bằng cách sử dụng thuật toán thông minh hơn trong việc nhận diện thiết bị qua camera. Bằng cách nhận diện vị trí của thiết bị sẽ hiển thị ảnh rơi xuống mặt bàn tại đúng vị trí đó để tạo cảm giác thật hơn. Đồng thời hệ thống có thể mở rộng để chia sẻ các dữ liệu khác như video, audio và cho phép gửi nhận với nhiều thiết bị ngoại vi khác như loa thông qua Bluetooth hay Wifi.

Chương 6

Kết luận

✍ Chương 6 này chúng em sẽ trình bày tóm lược những nội dung mà nhóm đã thực hiện và các kết quả đạt được trong quá trình thực hiện đề tài, qua đó mở ra hướng phát triển mới của đề tài trong tương lai.

6.1. Các kết quả đạt được

Qua đề tài này, chúng em đã tìm hiểu được thế nào là môi trường tương tác hỗ trợ Tangible, các kỹ thuật có thể triển khai trong môi trường tương tác này, đề đó đề xuất, thực nghiệm các kỹ thuật tương tác mới. Chúng em đã nắm vững và vận dụng được các kỹ thuật lập trình trong Unity3D Engine và sử dụng được, tích hợp Kinect vào hệ thống. Ngoài ra chúng em cũng đã tìm hiểu được các phương pháp nhận dạng vật thể khác nhau, thực nghiệm và so sánh để chọn phương pháp phù hợp nhất. Về phần mềm của hệ thống, chúng em đã áp dụng các mẫu thiết kế hướng đối tượng và xây dựng kiến trúc sự kiện để hệ thống có thể dễ dàng thay đổi tính năng, nội dung trình bày một cách nhanh chóng. Về phần cứng chúng em thiết kế để có thể dễ dàng vận chuyển và chỉnh sửa, phù hợp cho nhiều không gian sử dụng khác nhau. Tổng kết lại, những đóng góp chính của chúng em trong luận văn này là:

- Tìm hiểu về yếu tố Tangible trong tương tác, ảnh hưởng của nó tác động lên trải nghiệm người dùng và độ hiệu quả khi áp dụng vào các thao tác.
- Đề xuất và hiện thực hóa các kỹ thuật tương tác hỗ trợ yếu tố Tangible, chủ yếu tập trung vào 3 nhóm: tương tác trực tiếp vào môi trường, tương tác một phần vào môi trường và tương tác thông qua thiết bị di động.
- Khảo sát và thực nghiệm các cách nhận dạng vật thể bằng cách áp dụng kỹ thuật xử lý ảnh.

Trên cơ sở những kiến thức đã học hỏi được trong quá trình thực hiện luận văn, chúng em đã phát triển hệ thống tương tác hỗ trợ Tangible. Hệ thống chứng minh được tính thực tiễn của yếu tố Tangible khi áp dụng vào cuộc sống:

- Hệ thống bản đồ: Hệ thống hỗ trợ người dùng xem các thông tin trên bản đồ, tương tác thông qua thiết bị ngoại vi như điện thoại thông minh. Hệ thống

cung cấp thông tin toàn diện và tổng quan của một khu vực địa lý, ngoài ra cung cấp cho người dùng các thông tin chi tiết tại một vị trí cụ thể thông qua các hành vi tương tác khác nhau để truy vấn. Hệ thống thích hợp để phục vụ cho các cuộc thảo luận, thuyết trình cần truyền tải thông tin một cách trực quan, sinh động. Ngoài bản đồ địa lý, hệ thống còn có khả năng triển khai trên các lĩnh vực khác nhau như sinh học, lịch sử,... cho nên tiềm năng của hệ thống là rất cao.

- Hệ thống chia sẻ dữ liệu: Hệ thống cho phép người dùng chia sẻ dữ liệu như hình ảnh, tài liệu, ghi chú một cách trực quan. Hệ thống đóng vai trò là một môi trường lưu trữ vào cung cấp dữ liệu trực quan. Dữ liệu cần chia sẻ được hiển thị trực tiếp, người dùng có thể truy cập vào để chia sẻ dữ liệu hoặc lấy dữ liệu mà mình muốn có. Thông qua các cử chỉ, thao tác rất quen thuộc trong cuộc sống như hất lên, giũ xuống, người dùng có thể gửi hoặc nhận dữ liệu.
- Hồ cát hiển thị sự phân tầng độ cao: Đây là thí nghiệm chúng em đã thực nghiệm trong quá trình khảo sát độ hiệu quả của kỹ thuật tương tác trực tiếp vào môi trường để làm biến đổi thông tin. Thí nghiệm cho phép người dùng dùng tay để tạo hình cho cát, sau đó máy chiếu phía trên sẽ hiển thị hình ảnh phân tầng độ cao tương ứng với độ cao thực tế của bề mặt hồ cát. Thí nghiệm này có thể áp dụng vào các môn học địa lý và kích thích sự tìm tòi, sáng tạo của người học.

6.2. Hướng phát triển của đề tài

Hiện tại, lĩnh vực công nghệ thông tin đang phát triển rất nhanh, các ứng dụng khoa học kỹ thuật đang được áp dụng vào nhiều ngành, nghề trong cuộc sống. Với điều kiện thuận lợi đó, mọi người có nhiều cơ hội tiếp xúc với nhiều công nghệ mới. Trong số đó có thể kể đến môi trường tương tác hỗ trợ Tangible. Đây là cách mang đến cho người dùng trải nghiệm mới lạ, kích thích sự tìm tòi, khám phá, thông qua đó có thể truyền tải được thông tin nhanh hơn, hiệu quả hơn đến người dùng. Thay vì các thao tác chạm, chọn truyền thống, yếu tố Tangible còn cho phép người dùng tác động đến môi trường tương tác, thay đổi môi trường để làm thay đổi thông tin. Các

thao tác gần giống với các hành động trong sinh hoạt hằng ngày giúp người dùng cảm thấy quen thuộc, tự nhiên hơn, mang yếu tố ảo trở nên thực hơn.

Hướng phát triển tiếp theo của luận văn, chúng em sẽ khảo sát và nghiên cứu thêm các thao tác khác nhau có thể ứng dụng vào việc tương tác. Đồng thời sẽ thực hiện case study khảo sát trên người dùng để có thể đánh giá, so sánh và đề nghị thao tác nào là tốt nhất trong một ngữ cảnh cụ thể. Ngoài ra, chúng em sẽ tiếp tục cải thiện chất lượng module nhận dạng cũng như hoàn thiện hơn về giao diện người dùng của 2 hệ thống.

TÀI LIỆU THAM KHẢO

- [1] E. Sharlin, B. Watson, Y. Kitamura, F. Kishino, and Y. Itoh, “On tangible user interfaces, humans and spatiality,” *Pers. Ubiquitous Comput.*, vol. 8, no. 5, pp. 338–346, Sep. 2004.
- [2] E. Hornecker and J. Buur, “Getting a grip on tangible interaction,” in *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, 2006, p. 437.
- [3] B. Ullmer, H. Ishii, and R. J. K. Jacob, “Token+constraint systems for tangible interaction with digital information,” *ACM Trans. Comput. Interact.*, vol. 12, no. 1, pp. 81–118, Mar. 2005.
- [4] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso, “PingPongPlus,” in *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*, 1999, pp. 394–401.
- [5] J. Patten, H. Ishii, J. Hines, and G. Pangaro, “Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces.”
- [6] J. Patten, B. Recht, and H. Ishii, “Interaction techniques for musical performance with tabletop tangible interfaces,” in *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology - ACE '06*, 2006, p. 27.
- [7] A. DeVincenzi, L. Yao, H. Ishii, and R. Raskar, “Kinected conference,” in *Proceedings of the ACM 2011 conference on Computer supported cooperative work - CSCW '11*, 2011, p. 621.
- [8] L. Yao, R. Niiyama, J. Ou, S. Follmer, C. Della Silva, and H. Ishii, “PneUI,” in *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, 2013, pp. 13–22.
- [9] “" Shaping Watersheds " Augmented Reality Sandbox Facilitation Guide Shaping Watersheds AR Sandbox Facilitator’s Guide.”
- [10] O. Bimber and R. Raskar, “Spatial Augmented Reality Merging Real and Virtual Worlds.”
- [11] R. Raskar, K.-L. Low, and G. Welch, “Shader Lamps Animating Real Objects with Image-Based Illumination,” 2000.
- [12] J. J. Kuenzi, “Science, Technology, Engineering, and Mathematics (STEM)

Education: A Primer,” 2012.

- [13] S. Paul, S. Basu, and M. Nasipuri, “Microsoft Kinect in Gesture Recognition: A Short Review.”
- [14] J. Haas and B. Moriarty, “A History of the Unity Game Engine An Interactive Qualifying Project.”