**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT – CÔNG NGHỆ**



**BÁO CÁO**

**NGHIÊN CỨU KHOA HỌC**

**XÂY DỰNG HỆ THỐNG THỰC TẾ ẢO TĂNG CƯỜNG HỖ TRỢ HIỂN THỊ MÔ HÌNH TRÊN DI ĐỘNG TRONG HỌC TẬP**

| **Giảng viên hướng dẫn:** | **ThS. Hồ Đắc Hưng** |
| --- | --- |
| **Sinh viên:** | **Trần Tuấn Anh** |
| **MSSV:** 1825202010102; | **Ngành:** Kỹ thuật điện – điện tử |
| **Khóa:** D18; | Hệ chính quy |

Bình Dương, tháng 04/2021

**🙢🕮🙠**

# **LỜI CẢM ƠN**

*Chỉ thông qua tự do và trải nghiệm trong môi trường thực tế mà sự phát triển của con người mới có thể xảy ra.*

*- Maria Montessori -*

Chín tháng làm đề tài nghiên cứu khoa học với em là một trải nghiệm quý báu. Quãng thời gian này tuy ngắn ngủi nhưng cũng đủ để em tích lũy thêm cho riêng bản thân mình nhiều kinh nghiệm mới và nhìn nhận đúng đắn hơn về công việc tương lai. Cũng nhờ đó, em nhận thấy được rõ ràng khả năng và hướng đi sắp tới trong tương lai.

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến Thầy – ThS. Hồ Đắc Hưng – giảng viên hướng dẫn đề tài nghiên cứu khoa học của em trong chặng đường thực hiện đề tài này. Xin chân thành cảm ơn thầy vì những kiến thức, kinh nghiệm quý báu cùng sự đồng hành của thầy không chỉ trong quá trình thực hiền đề tài này mà từ những buổi tư vấn định hướng nghề nghiệp của em trong tương lai. Em sẽ không thể nào quên sự tận tâm, tận tình của thầy trong từng buổi trao đổi và hướng dẫn. Mỗi lần thầy giúp chúng em chỉnh phát âm tiếng anh thuật ngữ chuyên ngành, hướng dẫn lớp cách trích dẫn tài liệu đúng chuẩn, tìm báo cáo… là một lần em được tiếp thêm ngọn lửa nhiệt huyết để luôn chủ động trong công việc và sẵn sàng học hỏi, thay đổi để thích nghi. Chính nhờ tinh thần này, em đã rất tự tin và tự chủ trong công việc.

Từ tận trái tim, em xin kính chúc mọi người có thật nhiều sức khỏe, niềm vui và thành công trong cuộc sống!

Trân trọng.

| Sinh viên thực hiện |
| --- |
| Trần Tuấn Anh |

# **MỤC LỤC**

[**LỜI CẢM ƠN** - 1 -](#_heading=h.gjdgxs)

[**MỤC LỤC** - 2 -](#_heading=h.30j0zll)

[**DANH MỤC BẢNG, HÌNH VẼ, SƠ ĐỒ** - 5 -](#_heading=h.1fob9te)

[**DANH MỤC TỪ VIẾT TẮT** - 7 -](#_heading=h.2et92p0)

[**PHẦN MỞ ĐẦU** - 8 -](#_heading=h.tyjcwt)

[**1.1. Lý do chọn đề tài - 8 -**](#_heading=h.3dy6vkm)

[**1.2.** **Mục tiêu - 9 -**](#_heading=h.1t3h5sf)

[**1.3. Đối tượng và phạm vi đề tài - 9 -**](#_heading=h.4d34og8)

[**1.4. Mong đợi về đề tài - 10 -**](#_heading=h.2s8eyo1)

[**1.5. Phương pháp thực hiện - 11 -**](#_heading=h.17dp8vu)

[**1.6. Cách tiếp cận - 12 -**](#_heading=h.3rdcrjn)

[**PHẦN NỘI DUNG** - 13 -](#_heading=h.26in1rg)

[**CHƯƠNG 1. GIỚI THIỆU VỀ THỰC TẾ ẢO TĂNG CƯỜNG** - 13 -](#_heading=h.lnxbz9)

[**1.1. Khái niệm - 13 -**](#_heading=h.35nkun2)

[**1.2. Lịch sử hình thành và phát triển - 13 -**](#_heading=h.1ksv4uv)

[**1.3.** **Đặc trưng và kiến trúc hệ thống - 17 -**](#_heading=h.44sinio)

[**1.4. Một số bộ SDK hỗ trợ - 18 -**](#_heading=h.2jxsxqh)

[***1.4.1. Metaio* - 18 -**](#_heading=h.z337ya)

[***1.4.2. Vuforia* - 19 -**](#_heading=h.3j2qqm3)

[***1.4.3. Wikitude* - 19 -**](#_heading=h.1y810tw)

[***1.4.4. Blippar* - 20 -**](#_heading=h.4i7ojhp)

[**1.5. Các lĩnh vực ứng dụng - 20 -**](#_heading=h.1ci93xb)

[***1.5.1.******Lĩnh vực trò chơi* - 21 -**](#_heading=h.3whwml4)

[***1.5.2.******Lĩnh vực quảng cáo* - 21 -**](#_heading=h.2bn6wsx)

[***1.5.3.******Lĩnh vực bảo trì – sửa chữa sản phẩm* - 22 -**](#_heading=h.qsh70q)

[***1.5.4.* *Lĩnh vực y học* - 22 -**](#_heading=h.3as4poj)

[**CHƯƠNG 2. PHẦN MỀM VÀ CÔNG NGHỆ** - 24 -](#_heading=h.1pxezwc)

[**2.1. Phần mềm Solidworks - 24 -**](#_heading=h.49x2ik5)

[***2.1.1. Tổng quan phần mềm Solidworks* - 24 -**](#_heading=h.2p2csry)

[***2.1.2. Lịch sử phát triển* - 24 -**](#_heading=h.147n2zr)

[***2.1.3. Tính năng của Solidworks* - 25 -**](#_heading=h.3o7alnk)

[**2.2. Phần mềm Blender - 26 -**](#_heading=h.23ckvvd)

[***2.2.1. Tổng quan phần mềm Blender* - 26 -**](#_heading=h.ihv636)

[***2.2.2. Lịch sử phát triển* - 27 -**](#_heading=h.32hioqz)

[***2.2.3. Tính năng của Blender* - 29 -**](#_heading=h.1hmsyys)

[**2.3. Phần mềm PowerPoint - 32 -**](#_heading=h.41mghml)

[***2.3.1. Tổng quan phần mềm PowerPoint* - 32 -**](#_heading=h.2grqrue)

[***2.3.2. Lịch sử phát triển* - 33 -**](#_heading=h.vx1227)

[***2.3.3. Tính năng của PowerPoint* - 35 -**](#_heading=h.3fwokq0)

[**2.4. Phần mềm Unity - 36 -**](#_heading=h.1v1yuxt)

[***2.4.1. Tổng quan phần mềm Unity* - 36 -**](#_heading=h.4f1mdlm)

[***2.4.2. Lịch sử phát triển* - 37 -**](#_heading=h.2u6wntf)

[***2.4.3. Giới thiệu giao diện và tính năng của Unity* - 39 -**](#_heading=h.19c6y18)

[***2.4.4. Các thành phần khác của Unity* - 44 -**](#_heading=h.3tbugp1)

[**2.5. Phần mềm Microsoft Visual Studio - 52 -**](#_heading=h.28h4qwu)

[***2.5.1. Tổng quan phần mềm Microsoft Visual Studio* - 52 -**](#_heading=h.nmf14n)

[***2.5.2. Lịch sử phát triển* - 53 -**](#_heading=h.37m2jsg)

[***2.5.3. Tính năng của Microsoft Visual Studio* - 56 -**](#_heading=h.1mrcu09)

[**2.6. Vuforia Engine - 62 -**](#_heading=h.46r0co2)

[***2.6.1. Tổng quan Vuforia Engine* - 62 -**](#_heading=h.2lwamvv)

[***2.6.2. Lịch sử phát triển* - 63 -**](#_heading=h.111kx3o)

[***2.6.3. Tính năng của Vuforia Engine* - 63 -**](#_heading=h.3l18frh)

[**CHƯƠNG 3. XÂY DỰNG HỆ THỐNG THỰC TẾ ẢO TĂNG CƯỜNG HỖ TRỢ HIỂN THỊ MÔ HÌNH TRÊN DI ĐỘNG TRONG HỌC TẬP** - 66 -](#_heading=h.206ipza)

[**3.1. Cấu trúc tổng quan của một hệ thống thực tế ảo tăng cường trên điện thoại - 66 -**](#_heading=h.4k668n3)

[**3.2. Quy trình xây dựng hệ thống thực tế ảo tăng cường trên điện thoại - 66 -**](#_heading=h.2zbgiuw)

[**3.3. Tạo cơ sở dữ liệu 2D - 68 -**](#_heading=h.1egqt2p)

[**3.4. Tạo cơ sở dữ liệu 3D - 69 -**](#_heading=h.3ygebqi)

[**3.5. Kết nối các thành phần của hệ thống dựa trên Unity và xuất file APK/IOS - 70 -**](#_heading=h.2dlolyb)

[**CHƯƠNG 4: LẬP TRÌNH TÍNH NĂNG VÀ TẠO GIAO DIỆN** - 76 -](#_heading=h.sqyw64)

[**4.1. Lập trình các tính năng - 76 -**](#_heading=h.3cqmetx)

[**4.2. Tạo giao diện - 78 -**](#_heading=h.1rvwp1q)

[**4.2.1. Canvas - 79 -**](#_heading=h.4bvk7pj)

[**4.2.2. Text - 81 -**](#_heading=h.2r0uhxc)

[**4.2.3. Image - 82 -**](#_heading=h.1664s55)

[**4.2.4. Button - 83 -**](#_heading=h.3q5sasy)

[**4.3. Build hệ thống thành file apk - 85 -**](#_heading=h.25b2l0r)

[**CHƯƠNG 5: TỔNG KẾT** - 87 -](#_heading=h.kgcv8k)

[**TÀI LIỆU THAM KHẢO** - 91 -](#_heading=h.34g0dwd)

[**PHỤ LỤC** - 92 -](#_heading=h.1jlao46)

[**I. MỤC TIÊU CẦN ĐẠT ĐƯỢC TRONG THỜI GIAN THỰC HIỆN ĐỀ TÀI - 92 -**](#_heading=h.43ky6rz)

[**II. KẾ HOẠCH THỰC HIỆN ĐỀ TÀI - 92 -**](#_heading=h.2iq8gzs)

[**III. CODE CỦA HỆ THỐNG - 92 -**](#_heading=h.xvir7l)

# **DANH MỤC BẢNG, HÌNH VẼ, SƠ ĐỒ**

* **HÌNH ẢNH**

Hình 1.1. Một ứng dụng sử dụng Metaio

Hình 1.2. Một ứng dụng sử dụng Vuforia

Hình 1.3. Một ứng dụng sử dụng Wikitude

Hình 1.4. Một ứng dụng sử dụng Blippar

Hình 1.5. Game đua xe AR - STDIO Solutions

Hình 1.6. Thử đồng hồ bằng AR

Hình 1.7. Thông tin hóa các chi tiết

Hình 1.8. Phẫu thuật ít xâm lấn sử dụng AR

Hình 3.1. Tổng quan cấu trúc hệ thống thực tế ảo tăng cường trên điện thoại

Hình 3.2. Sơ đồ quy trình xây dựng hệ thống thực tế ảo tăng cường trên điện thoại

Hình 3.3. Các bước tạo dữ liệu 2D

Hình 3.4. Các bước tạo dữ liệu 3D

Hình 3.5. Đưa dữ liệu vào trong nơi chứa các tệp dự án

Hình 3.6. Tạo tệp C# để lập trình cho hệ thống

Hình 3.7. Lập trình các tính năng cho hệ thống

Hình 3.8. Thêm Vuforia Engine vào Unity

Hình 3.9. Thêm key vào AR Camera

Hình 3.10. Công cụ hỗ trợ thiết kế giao diện cho Unity

Hình 3.11. Thiết lập cho Unity

Hình 3.12. Thiết lập các thông số và thông tin cho hệ thống

Hình 4.1. Lấy Canvas trong UI của Unity

Hình 4.2. Render Mode của Canvas

Hình 4.3. Inspector của Text

Hình 4.4. Inspector của Image

Hình 4.5. Inspector của Button

Hình 4.6. Transition của Button

Hình 4.7. On Click của Button

Hình 4.8. Giao diện của hệ thống

Hình 4.9. Hộp thoại build settings

Hình 4.10. Hộp thoại player

* **BẢNG**

Bảng 5.1. Thông tin về hệ thống

Bảng 1.pl. Kế hoạch thực hiện đề tài

* **SƠ ĐỒ**

Sơ đồ 4.1. Lưu đồ giải thuật các tính năng cơ bản

Sơ đồ 4.2. Lưu đồ giải thuật các tính năng ẩn hiện thông tin

Sơ đồ 4.3. Lưu đồ giải thuật tính năng tương tác cao (chạm đối tượng)

# **DANH MỤC TỪ VIẾT TẮT**

| **STT** | **TỪ VIẾT TẮT** | **TỪ ĐẦY ĐỦ** |
| --- | --- | --- |
| 1 | AR | Thực tế tăng cường (Augmented Reality) |
| 2 | VR | Thực tế ảo (Virtual Reality) |
| 3 | 2D | 2 chiều (2 direction) |
| 4 | 3D | 3 chiều (3 direction) |
| 5 | XR | Thực tế mở rộng (Extended Reality) |
| 6 | MR | Thực tế hỗn hợp (Mixed Reality) |
| 7 | SDK | Bộ công cụ phát triển phần mềm (Software development kit) |

# **PHẦN MỞ ĐẦU**

## **Lý do chọn đề tài**

Hiện nay, thực tại ảo (VR) và thực tại ảo tăng cường (AR), một trong những lĩnh vực quan trọng của cách mạng công nghiệp 4.0, đang phát triển và ứng dụng mạnh mẽ trong nhiều lĩnh vực khác nhau. Sự triển khai mạng 5G kết hợp với công nghệ đám mây càng làm cho việc triển khai các ứng dụng VR vào cuộc sống xã hội được đẩy mạnh hơn bao giờ hết.

Trong giảng dạy, VR và AR ngày nay được ứng dụng mạnh mẽ nhằm nâng cao chất lượng giáo dục, đào tạo vì những ưu điểm của nó đặc biệt là khả năng vận dụng mô hình mô phỏng vào giảng dạy để tăng tính trực quan cũng như tạo điều kiện để sinh viên có nhiều điều kiện thực hành, thực tập hơn.

Ứng dụng công nghệ VR và AR trong giảng dạy nói chung, dạy kỹ thuật nói riêng cung cấp cho người học một nền tảng học tập mới và hiệu quả hơn so với các phương pháp dạy học truyền thống. Đó là người học có thể thực hành thông qua việc tương tác để đạt được khả năng trải nghiệm, quan sát, năng lực đổi mới qua các bài tập thực hành đầy đủ với sự hỗ trợ của công nghệ VR. Từ đó giúp cho người học có được các kỹ năng nghề nghiệp, hiệu quả giáo dục công nghệ được nâng cao.

Trong phương pháp giảng dạy truyền thống đã gặt hái được nhiều kết quả tích cực. Nhưng trước sự phát triển và ứng dụng nhanh chóng của khoa học kỹ thuật hiện đại vào cuộc sống ngày nay thì nó đã bộc lộ những hạn chế trong việc giảng dạy đào tạo cung ứng nguồn nhân lực chất lượng cao phù hợp với nhu cầu thực tế, nhất là trong giảng dạy các khái niệm, các chi tiết, các quy trình kỹ thuật phức tạp như trong các lĩnh vực hàng không, lái xe, lĩnh vực cơ khí, điện tử, tự động hóa và truyền thông.

Việc ứng dụng công nghệ VR và AR trong giảng dạy đã khắc phục được các hạn chế của phương pháp giảng dạy truyền thống. Nó cho phép tạo các đối tượng mà người học có thể tương tác với các khái niệm trừu tượng; cho phép quan sát, tương tác với các đối tượng kỹ thuật, các qui trình kỹ thuật phức tạp, tinh vi và nhất là trong điều kiện hiện nay các thiết bị thực hành còn quá đắt…

Các hệ thống thực tế ảo tăng cường cho học tập ở các nước đã được áp dụng vào trong thực tiễn như đào tạo lái xe, lái máy bay, lái tàu; huấn luyện nghiệp vũ, chiến thuật cho công an; đào tạo bác sĩ; ...

Hiện nay tại Việt Nam, các dự án, hệ thống thực tế ảo tăng cường trong học tập còn khá mới và vừa được triển khai trong thời gian gần 5 năm trở lại đây với các sản phẩm đang được nghiên cứu và phát triển như hệ thống đào tạo lái xe bằng thực tế ảo tăng cường (VRtech), các lớp học bằng thực tế ảo 360o, du lịch kinh thành Huế bằng thực tế ảo, tham quan bảo tàng thực tế ảo, ... và các sản phẩm đã được thương mại hóa như “Tranh tô màu 4D KOLORFUN” bởi Vrtech,....

Đề tài xây dựng hệ thống thực tế ảo tăng cường hỗ trợ hiển thị mô hình trên di động trong học tập sẽ tăng tính tương tác và mở rộng các mô hình phục vụ cho việc học tập.

Ngoài ra, đề tài còn là nền tảng để phát triển thêm các hệ thống mô hình, mô phỏng để hỗ trợ cho các mục đích giáo dục khác của Đại học Thủ Dầu Một. Trên cơ sở nghiên cứu các quy trình thiết kế, xây dựng các hệ thống thực tế ảo tăng cường, nhóm nghiên cứu sẽ đề xuất một quy trình phát triển hệ thống thực tế ảo tăng cường phù hợp với mục đích mô hình, mô phỏng trong giáo dục nói chung và với Đại học Thủ Dầu Một nói riêng.

## **Mục tiêu**

Xây dựng hệ thống thực tế ảo tăng cường hỗ trợ hiển thị mô hình trên di động trong học tập có tính tương tác cao.

## **1.3. Đối tượng và phạm vi đề tài**

Căn cứ vào tình hình phát triển của thực tế tăng cường, em khoanh vùng đối tượng và phạm vi đề tài như sau:

- ***Đối tượng nghiên cứu:*** Mô hình 2D, mô hình 3D, phần mềm hỗ trợ (Unity, PowerPoint, Solidworks, Blender, Visual Studio), công nghệ Vuforia.

*-* ***Phạm vi nghiên cứu:***

* *Phạm vi về thời gian:* Đề tài được thực hiện trong khoảng 09 tháng, từ ngày 1/11/2020 cho đến 01/08/2021.
* *Phạm vi ứng dụng:* Ứng dụng trong giảng dạy cơ khí, chi tiết máy.

## **1.4. Mong đợi về đề tài**

Qua quá trình học tập và thực hiện những đề tài, em đã có một trải nghiệm khá đa dạng về việc ứng dụng thực tế tăng cường vào trong học tập. Tuy nhiên em vẫn chưa có cơ hội trải nghiệm việc về việc xây dựng các hệ thống thực tế tăng cường. Vì vậy, trước khi làm đề tài, cá nhân em có rất nhiều kỳ vọng trong đề tài này.

Nhận thức được tầm quan trọng của học tập và phát triển cá nhân chính là lý do em chọn đề tài về thực tế tăng cường. Có thể nói, đây sẽ là tiền đề giúp em xác định liệu con đường đang chọn có thực sự phù hợp với bản thân. Ngoài ra, em cũng mong muốn trải nghiệm những thách thức sự tìm tòi, học hỏi của bản thân. Đây sẽ là cơ hội để em nhận ra những điểm mạnh và cả những thiếu sót của bản thân để cải thiện trước khi bước vào môi trường làm việc thực tế.

Bên cạnh đó, em mong muốn được biết và hiểu nhiều hơn về cách xây dựng một hệ thống thực tế tăng cường. Ngoài ra, quan trọng không kém là sự đón nhận cùng những chờ đợi, mong mỏi của bản thân khi thực hiện đề tài. Nhất là trong bối cảnh mà năng lượng tuổi trẻ trong bản thân em đang bùng cháy.

Nhìn chung, đề tài thực hiện trong 9 tháng, nhưng tự bản thân em cảm nhận rằng bản thân mình đã trưởng thành hơn và tích lũy thêm được nhiều bài học giá trị. Và dù đã cố gắng và nỗ lực tìm hiểu rất nhiều nhưng cũng không thể nào tránh khỏi những sai sót, những thiếu hụt về nhiều mặt không chỉ là kĩ năng và kiến thức. Do đó với bài báo cáo này, em rất mong sẽ được đón nhận những ý kiến, góp ý phê bình từ các thầy để ngày càng hoàn thiện hơn.

## **1.5. Phương pháp thực hiện**

Dựa trên mục tiêu, đối tượng và phạm vi được nêu ra ở những nội dung trên, đề tài phối hợp sử dụng nhiều phương pháp, bao gồm:

***1.5.1. Phương pháp thu thập ghi chép số liệu***

Đối tượng: người sử dụng.

Mục đích: thu thập dữ liệu của người sử dụng để nâng cao hệ thống trong các phiên bản sau.

Cách thức thực hiện: tiến hành lắng nghe, quan sát, ghi chú các góp kiến của người sử dụng.

***1.5.2. Phương pháp nghiên cứu tài liệu***

Đối tượng: các bài báo về thực tế tăng cường.

Mục đích: thu thập và tập hợp các phương pháp để xây dựng hệ thống thực tế tăng cường.

Cách thức thực hiện: đọc các bài báo về thực tế tăng cường.

***1.5.3. Phương pháp phân tích, tổng hợp***

Đối tượng: tất cả dữ liệu thu thập về (định tính lẫn định lượng).

Mục đích: tổng hợp, so sánh, đối chiếu các dữ liệu nhằm tìm ra những vấn đề, cơ hội, thách thức, điểm mạnh cần phát huy, điểm yếu cần khắc phục, đánh giá thực trạng…

Cách thức thực hiện: tổng hợp dữ liệu từ tài liệu, góp ý của người dùng để so sánh và đối chiếu.

***1.5.4. Phương pháp đối chiếu và so sánh***

Đối tượng: hoạt động của hệ thống.

Mục đích: kiểm tra hệ thống hoạt động đúng theo yêu cầu của người sử dụng hay không để tiến hành sửa lỗi.

Cách thức tiến hành: chạy thử hệ thống với tất cả tính năng.

**1.6. Cách tiếp cận**

* Tìm hiểu quy trình xây dựng hệ thống thực tế ảo tăng cường trên nền tảng di động.
* Thiết kế và thi công các thẻ nhận diện mô hình 2D.
* Từ các mô hình 2D đã thiết kế từ trước thiết kế thêm những mô hình 3D tương ứng và lưu trong hệ thống dữ liệu.
* Thiết kế và thi công App có khả năng quét ảnh 2D, thu thập thông tin và hiển thị hình ảnh 3D tương ứng lên thiết bị di động.
* Thử nghiệm và kiểm tra kết quả.

**PHẦN NỘI DUNG**

# **CHƯƠNG 1. GIỚI THIỆU VỀ THỰC TẾ ẢO TĂNG CƯỜNG**

## **1.1. Khái niệm**

Thực tế ảo tăng cường (tiếng Anh: augmented reality, viết tắt là AR) là góc nhìn trực tiếp hay gián tiếp về môi trường vật lý, thực tế nơi mà các yếu tố được "tăng cường" bởi thông tin nhận thức do máy tính tạo ra, lý tưởng trên nhiều phương thức cảm quan bao gồm thị giác, [thính giác](https://vi.wikipedia.org/wiki/Th%C3%ADnh_gi%C3%A1c), [xúc giác](https://vi.wikipedia.org/wiki/X%C3%BAc_gi%C3%A1c) và [khứu giác](https://vi.wikipedia.org/wiki/Kh%E1%BB%A9u_gi%C3%A1c). Các phủ thông tin cảm giác có thể được xây dựng (tức là phụ để các môi trường tự nhiên) hoặc phá hoại (tức là mặt nạ của môi trường tự nhiên) và là không gian đã đăng ký với thế giới vật chất như vậy mà nó được coi như là một nhập vai khía cạnh của môi trường.

## **1.2. Lịch sử hình thành và phát triển**

Thực tế ảo tăng cường bắt đầu vào năm 1968 bởi Sutherland, người đã sử dụng nó trong việc xây dựng các hệ thống hiển thị gắn trên đầu. Những tiến bộ trong hiệu suất điện toán của những năm 1980 và đầu những năm 1990 cuối cùng đã đạt được yêu cầu để AR nổi lên như một lĩnh vực nghiên cứu độc lập. Trong suốt những năm 1970 và 1980, Myron Krueger, Dan Sandin, Scott Fisher và những người khác đã thử nghiệm nhiều khái niệm pha trộn tương tác của con người với các lớp phủ do máy tính tạo ra trên video để trải nghiệm nghệ thuật tương tác. Về các chú thích đồ họa giữa các hình bóng người tham gia trong các bản cài đặt Videoplace của mình vào khoảng năm 1974.

Năm 1992 đánh dấu sự ra đời của thuật ngữ thực tế tăng cường. Thuật ngữ này lần đầu tiên xuất hiện trong tác phẩm của Caudell và Mizell tại Boeing, nhằm tìm cách hỗ trợ công nhân trong một máy bay trực thăng, bằng cách hiển thị sơ đồ lắp ráp dây. Nối tiếp đó, năm 1993, Feiner et al. đã giới thiệu KARMA, một hệ thống kết hợp AR dựa trên tri thức. Hệ thống này có khả năng tự động suy ra các chuỗi hướng dẫn phù hợp cho các quy trình sửa chữa và bảo trì.

Cũng trong năm 1993, Fitzmaurice đã tạo ra màn hình nhận biết không gian cầm tay đầu tiên, đóng vai trò là tiền thân của AR cầm tay. The Chameleon bao gồm một màn hình tinh thể lỏng cầm tay (LCD) có dây buộc. Màn hình hiển thị đầu ra video của máy trạm đồ họa SGI thời đó và được theo dõi không gian bằng thiết bị theo dõi từ tính. Hệ thống này có khả năng hiển thị thông tin theo ngữ cảnh khi người dùng di chuyển thiết bị xung quanh, ví dụ, cung cấp thông tin chi tiết về vị trí trên bản đồ treo tường.

Năm 1994, State et al. at the University of North Carolina tại Chapel Hill đã trình bày một ứng dụng AR y tế hấp dẫn, có khả năng cho phép bác sĩ quan sát trực tiếp thai nhi trong một bệnh nhân đang mang thai. Mặc dù việc đăng ký chính xác đồ họa máy tính lên trên một vật thể biến dạng như cơ thể người vẫn là một thách thức ngày nay, công việc tinh thần này gợi ý về sức mạnh của AR đối với y học và các nhiệm vụ tế nhị khác.

Vào năm 1995, Rekimoto và Nagao đã tạo ra màn hình AR cầm tay thật sự đầu tiên mặc dù có màn hình AR cầm tay. NaviCam của họ đã được kết nối với một máy trạm, nhưng được trang bị một camera phía trước. Từ nguồn cấp dữ liệu video, nó có thể phát hiện các điểm đánh dấu được mã hóa màu trong hình ảnh camera và thông tin hiển thị trên chế độ xem qua video.

Năm 1996, Schmalstieg và cộng sự đã phát triển Studierstube, hệ thống AR hợp tác đầu tiên. Với hệ thống này, nhiều người dùng có thể trải nghiệm các đối tượng ảo trong cùng một không gian chung. Mỗi người dùng có một HMD được theo dõi và có thể nhìn thấy hình ảnh lập thể chính xác theo quan điểm từ một quan điểm cá nhân. Không giống như trong multi-user VR, các tín hiệu giao tiếp tự nhiên như giọng nói, tư thế cơ thể và cử chỉ, không bị ảnh hưởng trong Studierstube, vì nội dung ảo đã được thêm vào một tình huống hợp tác thông thường theo cách gây khó chịu một cách tối thiểu. Một trong những ứng dụng được giới thiệu là một khóa học hình học (Kaufmann và Schmalstieg 2003), đã được thử nghiệm thành công với giáo dục.

Từ 1997 đến 2001, chính phủ Nhật Bản và Canon Inc. đã cùng tài trợ cho Mixed Reality Systems Laboratory như một công ty nghiên cứu tạm thời. Liên doanh này là cơ sở nghiên cứu thử nghiệm lớn nhất cho nghiên cứu mixed reality (MR) cho đến thời điểm đó (Tamura 2000). Một trong những thành tựu đáng chú ý nhất của nó là thiết kế video âm thanh nổi đồng trục đầu tiên nhìn xuyên qua HMD, COASTAR. Nhiều hoạt động được thực hiện trong phòng thí nghiệm cũng được hướng tới thị trường giải trí kỹ thuật số, đóng vai trò rất nổi bật tại Nhật Bản.

Năm 1997, Feiner et al. đã phát triển hệ thống AR ngoài trời đầu tiên, Touring Machine, tại Đại học Columbia. Máy Touring sử dụng thiết bị HMD nhìn xuyên qua có GPS và theo dõi định hướng. Cung cấp đồ họa 3D di động thông qua hệ thống này yêu cầu một chiếc ba lô cầm chứa một máy tính, nhiều cảm biến khác nhau và một máy tính bảng đầu tiên để nhập liệu.

Chỉ một năm sau, năm 1998, Thomas và cộng sự. công bố công trình của họ về việc xây dựng một hệ thống định vị AR ngoài trời, Map-in-the-Hat. Nền tảng này đã được sử dụng cho các ứng dụng nâng cao, như khảo sát 3D, nhưng nổi tiếng nhất là cung cấp trò chơi AR ngoài trời đầu tiên, ARQuake. Trò chơi này, là một cổng của ứng dụng bắn súng góc nhìn thứ nhất nổi tiếng Quake to Tinmith, đặt người dùng vào giữa một cuộc tấn công zombie trong một bãi đậu xe thực sự.

Trong cùng năm đó, Raskar et al. tại Đại học Bắc Carolina tại Đồi Chapel gửi tới The Office of the Future, một hệ thống từ xa được xây dựng dựa trên ý tưởng về hệ thống quét ánh sáng và máy chiếu có cấu trúc. Mặc dù phần cứng cần thiết không thực sự thiết thực để sử dụng hàng ngày vào thời điểm đó, các công nghệ liên quan, chẳng hạn như cảm biến độ sâu và khớp nối máy ảnh, đóng một vai trò nổi bật trong AR và các lĩnh vực khác hiện nay.

Cho đến năm 1999, không có phần mềm AR nào có sẵn bên ngoài các phòng thí nghiệm nghiên cứu chuyên ngành. Tình trạng này đã thay đổi khi Kato và Billinghurst phát hành ARToolKit, nền tảng phần mềm nguồn mở đầu tiên cho AR. Nó nổi bật với một thư viện theo dõi 3D sử dụng các ống kính đen trắng, có thể dễ dàng sản xuất trên máy in laser. Thiết kế phần mềm thông minh, kết hợp với sự sẵn có của webcam, khiến ARToolKit trở nên phổ biến rộng rãi.

Trong cùng năm đó, Bộ Giáo dục và Nghiên cứu Liên bang Đức đã khởi xướng một chương trình trị giá 21 triệu euro cho AR công nghiệp, được gọi là ARVIKA (Thực tế tăng cường cho phát triển, sản xuất và phục vụ). Hơn 20 nhóm nghiên cứu từ ngành công nghiệp và học viện đã nghiên cứu phát triển các hệ thống AR tiên tiến cho ứng dụng công nghiệp, đặc biệt là trong ngành công nghiệp ô tô Đức. Chương trình này đã nâng cao nhận thức về AR trên toàn thế giới trong các cộng đồng chuyên nghiệp và được theo dõi bởi một số chương trình tương tự được thiết kế để tăng cường ứng dụng công nghiệp của công nghệ.

Một ý tưởng đáng chú ý khác cũng xuất hiện vào cuối năm 1999: nhà nghiên cứu IBM Spohrer đã viết một bài tiểu luận về Worldboard, một cơ sở hạ tầng có thể mở rộng để biết thông tin đăng ký không gian liên kết, mà Spohrer đã đề xuất đầu tiên khi ông làm việc với Apple. Công việc này có thể được xem là khái niệm đầu tiên cho AR trên browser.

Và từ năm 2000 trở đi, các bạn cũng thấy sự phát triển của điện thoại di động và điện toán di động, AR cũng vậy, nó được kết hợp với mobile và cho ra các sản phầm ví dụ như: Pin- taric et al. (2005), một trò chơi AR cầm tay nhiều người chơi, đã được trải nghiệm bởi hàng ngàn khách truy cập tại SIGGRAPH.

Phải mất vài năm, cho đến năm 2008, hệ thống theo dõi tính năng tự nhiên thực sự có thể sử dụng đầu tiên cho điện thoại thông minh được giới thiệu Wagner et al. 2008. Công việc này trở thành tổ tiên của bộ công cụ Vuforia nổi tiếng dành cho các nhà phát triển AR. Những thành tựu đáng chú ý khác trong những năm gần đây trong lĩnh vực theo dõi bao gồm hệ thống theo dõi và lập bản đồ song song (PTAM) của Klein và Murray (2007), có thể theo dõi mà không cần chuẩn bị trong môi trường không xác định và hệ thống KinectFusion do Newcombe et al phát triển. (2011), xây dựng các mô hình 3D chi tiết từ một cảm biến độ sâu không tốn kém.

Ngày nay, các nhà phát triển AR có thể chọn trong số nhiều nền tảng phần mềm, nhưng các hệ thống mô hình này tiếp tục đại diện cho các hướng quan trọng cho các nhà nghiên cứu như là Pokemon GO một trò chơi rất nổi năm trước cũng sử dụng mô hình này để phát triển.

## **Đặc trưng và kiến trúc hệ thống**

Một hệ thống AR có 3 đặc trưng sau:

* Kết hợp thực tại và ảo.
* Tạo ra tương tác theo thời gian.
* Thể hiện trong không gian ba chiều.

Khác với [thực tế ảo - VR](https://www.stdio.vn/technical-artist/tong-quan-ve-thuc-tai-ao-virtual-reality-e1Eg3U), các thông tin tăng cường trong hệ thống AR liên hệ chặt chẽ với môi trường thực, sự xuất hiện của các thông tin thay đổi theo cách người dùng di chuyển cũng như xem xét các thành phần trong môi trường thực.

Một hệ thống AR gồm có 3 phần:

* + - Thế giới thực.
    - Các thiết bị phần cứng nhận diện các thành phần của môi trường.
    - Phần mềm hỗ trợ.

Thiết bị phần cứng cho AR gồm có: bộ vi xử lý, màn hình, cảm biến, các thiết bị đầu vào, các thiết bị điện toán di động như smartphone, tablet có chứa camera, cảm biến gia tốc, GPS, la bàn, ...

Các thiết bị AR hiện tại nổi bật có thể kể đến như Microsoft Hololens 2, Vuzix Blade AR, hoặc các dòng điện thoại cao cấp.

Chìa khóa của các phần mềm AR là làm cách nào để tích hợp được các thông tin tăng cường vào thế giới thực. Mỗi phần mềm AR phải lấy được tọa độ thế giới thực từ các hình ảnh camera, tọa độ này không phụ thuộc vào các camera. Quá trình này gọi là đăng ký hình ảnh, sử dụng các phương pháp khác nhau về thị giác máy tính, quá trình này có thể chia làm 2 bước:

* Bước 1: Phát hiện các điểm cần quan tâm, hoặc các đánh dấu chuẩn, các dòng quang học trong các hình ảnh thu được từ camera.
* Bước 2: Xây dựng lại thế giới thực dựa trên các thông tin thu thập từ bước 1, có thể tính toán trước 1 số cấu trúc 3D trong trường hợp dựng cảnh quá phức tạp. Sau đó bổ sung thêm các thông tin cần tăng cường.

Các phần mềm này có quá trình xây dựng khá phức tạp, do đó để phát triển nhanh các ứng dụng AR một số hãng phần mềm đã cho ra đời các bộ SDK, tiêu biểu như: Metaio, Vuforia, Mobinet AR, Wikitude, Blippar và Layar.

## **1.4. Một số bộ SDK hỗ trợ**

***1.4.1. Metaio***

[Metaio SDK](http://www.metaio.com/) là một trong những bộ SDK có nhiều tính năng cho phát triển ứng dụng AR. SDK này hỗ trợ xây dựng ứng dụng AR hoàn chỉnh, với các tính năng tracking 2-D, Location-based, Extended 2-D, 3-D. Đây là bộ SDK có tính phí, bản thương mại với giá từ 3,490 đô.



***Hình 1.1.*** Một ứng dụng sử dụng Metaio

***1.4.2. Vuforia***

[Vuforia SDK](https://www.qualcomm.com/) do hãng Qualcomm phát triển. Điểm mạnh của Vuforia chính là tính năng Recognition: Objects, Images, User-Defined Images, Cylinders, Text, Boxes, Frame Markers. Gần đây, các tính năng mới của Vuforia cũng thu hút rất nhiều sự quan tâm của giới lập trình viên như các tính năng: Smart Terrain, Extended Tracking, Background Effects, Video Playback, Virtual Button, Occlusion Management.

Bên cạnh đó Vuforia cũng có các tính năng mở rộng tích hợp cho các engine như Unity3D…, phát triển các ứng dụng đa nền tảng. Vuforia cung cấp bản miễn phí cho lập trình viên và thu phí trên hệ thống cloud lưu trữ hình ảnh với số lượng lớn hơn 1000 ảnh.



***Hình 1.2.*** Một ứng dụng sử dụng Vuforia

***1.4.3. Wikitude***

[Wikitude SDK](http://www.wikitude.com/app) hỗ trợ khá nhiều tính năng cho AR, nổi bật là các tính năng về augmentation và visualization cho các mô hình 3D tĩnh và động. Bên cạnh đó SDK này cũng hỗ trợ phát triển ứng dụng đa nền tảng cùng với hệ thống cloud. Wikitude phát triển nhiều bản phần mềm thương mại, với mức phí từ 590 € trở lên.



***Hình 1.3.*** Một ứng dụng sử dụng Wikitude

***1.4.4. Blippar***

[Blippar](https://blippar.com/) cung cấp các nền tảng AR để phát triển các ứng dụng quảng cáo và phát hành nội dung. Ưu điểm của Blippar là công nghệ nhận diện hình ảnh.



***Hình 1.4.*** Một ứng dụng sử dụng Blippar

**1.5. Các lĩnh vực ứng dụng**

Thực tại tăng cường được ứng dụng trong nhiều lĩnh vực khác nhau, nổi bật là các lĩnh vực trò chơi, quảng cáo, bảo trì – sửa chữa sản phẩm, y học, du lịch, ....

* + 1. ***Lĩnh vực trò chơi***

Cùng với sự phát triển ngày càng mạnh mẽ của smartphone và tablets, thị trường game cho nền tảng di động càng ngày sôi động. Bên cạnh các game truyền thống, thể loại game AR đã mang đến cho người chơi nhiều trải nghiệm mới mẻ.



***Hình 1.5.*** Game đua xe AR - STDIO Solutions

* + 1. ***Lĩnh vực quảng cáo***

Các dịch vụ quảng cáo phát triển trên nền tảng AR đã chứng tỏ được tiềm năng của mình trong những năm gần đây. Bên cạnh những ứng dụng nhận diện sản phẩm, các ứng dụng AR hỗ trợ bán hàng qua mạng của các nhà sản xuất lớn cũng có một tầm ảnh hưởng nhất định.



***Hình 1.6.*** Thử đồng hồ bằng AR

* + 1. ***Lĩnh vực bảo trì – sửa chữa sản phẩm***

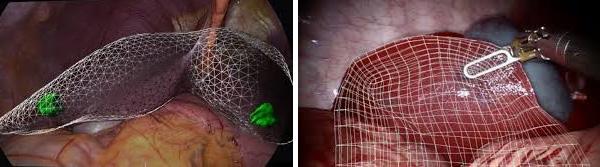
Là tiền đề cho sự ra đời của AR, các ứng dụng bảo trì – sửa chữa sản phẩm ngày càng được phát triển chi tiết hóa các thành phần của sản phẩm.



***Hình 1.7.*** Thông tin hóa các chi tiết

* + 1. ***Lĩnh vực y học***

Trong các công bố khoa học gần đây, các nhà khoa học máy tính đã kết hợp hỗ trợ các bệnh viện tại Mỹ trong việc ứng dụng AR trong các phẫu thuật ít xâm lấn, nhận diện hình ảnh, bổ sung thông tin liên tục cho các bác sĩ trong quá trình phẫu thuật. Các kết quả thực nghiệm đã chứng minh được tính hiệu quả của các phương pháp này.

***Hình 1.8.*** Phẩu thuật ít xâm lấn sử dụng AR

Xu thế hiện tại AR sẽ là hướng phát triển mới cho các ứng dụng, ngày càng gần với thực tế. Hi vọng những chia sẻ này giúp các bạn có cái nhìn rõ ràng hơn về AR và có thể phát triển các ứng dụng AR của chính mình.

# **CHƯƠNG 2. PHẦN MỀM VÀ CÔNG NGHỆ**

## **2.1. Phần mềm Solidworks**

***2.1.1. Tổng quan phần mềm Solidworks***

Solidworks là phần mềm thiết kế 3D chạy trên hệ điều hành Windows và có mặt từ năm 1997, được tạo bởi công ty Dassault Systèmes SOLIDWORKS Corp., là một nhánh của Dassault Systèmes, S. A. (Vélizy, Pháp). Solidworks hiện tại được dùng bởi hơn 2 triệu kỹ sư và nhà thiết kế với hơn 165,000 công ty trên toàn thế giới.

Phần mềm Solidworks được biết đến từ phiên bản Solidworks 1995. ViHoth phân phối phần mềm này từ phiên bản 2011 cho đến nay. Solidworks đã có nhiều bước phát triển vượt bậc về tính năng, hiệu suất và khả năng đáp ứng các nhu cầu thiết kế 3D trong các ngành kỹ thuật, công nghiệp. Solidworks còn được phát triển và ứng dụng rộng rãi trong các ngành khác như: đường ống, kiến trúc, nội thất, xây dựng… nhờ tính năng thiết kế 3D mạnh mẽ và danh mục các giải pháp hỗ trợ đa dạng.

Các dòng sản phẩm phân tích, mô phỏng của Solidworks giúp giải quyết các vấn đề liên quan đến lắp ghép, truyền động, động học (Motion), độ bền, ứng suất, mô phỏng dòng chảy và áp suất… Solidworks CAM là sản phẩm mới của Solidworks hỗ trợ lập trình gia công phay, tiện 2 – 5 trục.

Với trên 7.600 đối tác giải pháp, Solidworks có thể thực hiện add-on nhiều giải pháp phần mềm khác như: phần mềm SolidCAM, MasterCAM cho lập trình gia công CNC, phần mềm SolidPlant cho thiết kế đường ống, kết cấu, 3D QuicPress, 3D QuickMold cho thiết kế khuôn và khuôn dập.

***2.1.2. Lịch sử phát triển***

Công ty SolidWorks được thành lập vào tháng 12 năm 1993 bởi Hirschtick, tốt nghiệp trường MIT nổi tiếng - Massachusetts Institute of Technology; Hirschtick sử dụng1 triệu đô mà anh ta gây dựng được khi là thành viên MIT Blackjack Team để thành lập công ty. Trụ sở ban đầu ở Waltham, Massachusetts, USA, Hirschtick tuyển dụng một nhóm kỹ sư nhằm tạo một phần mềm 3D CAD dễ sử dụng, giá cả phải chăng, và có thể tùy biến trên Windows desktop. Sau này đổi địa chỉ là Concord, Massachusetts, SolidWorks đã phát hành phiên bản đầu tiên SolidWorks 95, năm 1995. Năm 1997 Dassault, Công ty nổi tiếng nhất với phần mềm CATIA, đã mua lại SolidWorks với 310 triệu đô la cổ phiếu. SolidWorks hiện tại có một số phiên bản như SolidWorks CAD, eDrawings một công cụ hỗ trợ và DraftSight, một sản phẩm 2D CAD. SolidWorks được điều hành bởi John McEleney từ 2001 tới July 2007 và Jeff Ray từ 2007 tới tháng 1-2011. CEO hiện tại là Bertrand Sicot.

***2.1.3. Tính năng của Solidworks***

***2.1.3.1. Thiết kế mô hình 3D***

Trong phần mềm Solidworks thì đây được coi là tính năng nổi bật với việc thiết kế các các biên dạng 2D bạn sẽ dựng được các khối 3D theo yêu cầu.

***2.1.3.2. Lắp ráp các chi tiết***

Các chi tiết 3D sau khi được thiết kế xong bởi tính năng thiết kế có thể lắp ráp lại với nhau tạo thành một bộ phận máy hoặc một máy hoàn chỉnh. Tính năng này giúp bạn dễ dàng chỉnh sửa, thỏa sức sáng tạo và nghiên cứu dễ dàng cho những sản phẩm mới.

***2.1.3.3. Xuất bản vẽ dễ dàng***

Phần mềm Solidworks cho phép người dùng xuất bản vẽ dưới nhiều định dạng khác nhau như stl, asm, ….

***2.1.3.4. Tính năng Tab và Slot***

Phần mềm Solidworks cho phép người dùng tự động tạo ra các tính năng tab và slot được sử dụng để tự lắp ghép các bộ phận hàn. Các tính năng cải tiến kim loại khác bao gồm tính năng Normal Cut mới đảm bảo duy trì khoảng cách thích hợp cho sản xuất, và khả năng uốn mới cho phép người dùng tạo mới và trải phẳng góc uốn.

***2.1.3.5. Cải tiến quản lý dự án và quy trình***

Solidworks Manage cung cấp công cụ quản lý dữ liệu, dự án, và quản lý quy trình trong một gói phần mềm quen thuộc. Các khả năng quản lý các dự án, và quản lý quy trình được thêm vào Solidworks PDM Professional.

***2.1.3.6. Các tiện ích cải tiến***

Online Licensing giúp cho việc sử dụng các license trên nhiều máy tính tiện lợi hơn trước rất nhiều. Solidworks Login sẽ chuyển các nội dung và cài đặt các tùy chịn đến bất kỳ máy tính nào được cài Solidworks, trong khi Admin Portal cho phép quản lý các sản phẩm và dịch vụ của Solidworks dễ dàng hơn.

***2.1.3.7. Tính năng gia công***

Giải pháp gia công CAD CAM kết hợp, giải pháp có tên Solidworks CAM, nó được tách ra để bán riêng. Giải pháp này khá đơn giản và dễ dùng. Các modul đơn giản thân thiện.

***2.1.3.8. Phân tích động lực học***

Solidworks Simulation cung cấp các công cụ mô phỏng để kiểm tra và cải thiện chất lượng bản thiết kế của bạn. Các thuộc tính vật liệu, mối ghép, quan hệ hình học được định nghĩa trong suốt quá trình thiết kế được cập nhật đầy đủ trong mô phỏng.

## **2.2. Phần mềm Blender**

***2.2.1. Tổng quan phần mềm Blender***

Blender là một chương trình máy tính dùng để tạo ra các đồ họa 3D, với các công cụ như tạo mô hình, áp vật liệu, mô phỏng khói lửa, chất lỏng, biên tập video và kết xuất ra các đoạn phim hoạt hình, các đoạn phim kỹ xảo và các nhân vật, môi trường trong các trò chơi tương tác.

Blender là một phần mềm mã nguồn mở, hoàn toàn miễn phí của Hà Lan. Tuy miễn phí, nhưng Blender sở hữu các công cụ vô cùng mạnh mẽ và có hầu hết các tính năng của các phần mềm thương mại khác trên thị trường trong cùng lĩnh vực. Với các sự ưu việt đó, đồng thời luôn được phát triển và hỗ trợ bởi cộng đồng đông đảo, Blender là công cụ luôn có mặt trong danh sách bình chọn các phần mềm 3D phổ biến nhất ở cấp độ dành cho người dùng chuyên nghiệp.

Hiện nay, Blender đang dẫn đầu xu hướng về các phần mềm đồ họa 3D và là công cụ chính của đa số các studio vừa và nhỏ trên thế giới. Tại Việt Nam, Blender đang được triển khai thành chương trình học tại một số trường đại học. Các công ty sử dụng Blender cho hoạt hình, game và kiến trúc đã xuất hiện và nhu cầu về kỹ năng Blender ngày càng tăng.

***2.2.2. Lịch sử phát triển***

Năm 1988, Ton Roosendaal đồng sáng lập hãng phim hoạt hình NeoGeo tại Hà Lan. NeoGeo nhanh chóng trở thành xưởng phim hoạt hình 3D lớn nhất ở Hà Lan và là một trong những hãng phim hoạt hình hàng đầu ở châu Âu. NeoGeo đã tạo ra các sản phẩm đoạt giải thưởng (Giải thưởng Video doanh nghiệp châu Âu 1993 và 1995) cho các khách hàng doanh nghiệp lớn như công ty điện tử đa quốc gia Philips. Trong NeoGeo, Ton chịu trách nhiệm cho cả định hướng nghệ thuật và phát triển phần mềm nội bộ. Sau khi cân nhắc kỹ lưỡng, Ton đã quyết định rằng bộ công cụ 3D hiện tại dành cho NeoGeo quá cũ và cồng kềnh và cần phải viết lại từ đầu. Năm 1995, việc viết lại này đã bắt đầu và được định sẵn để trở thành phần mềm 3D mà chúng ta đều biết là Blender. Khi NeoGeo liên tục cải tiến Blender, Ton cho rằng Blender có thể được sử dụng như một công cụ cho các nghệ sĩ khác bên ngoài hãng phim NeoGeo.

Năm 1998, Ton quyết định thành lập một công ty mới có tên Not a Number (NaN) như một công ty con của NeoGeo để tiếp thị và phát triển Blender. Cốt lõi của NaN là mong muốn tạo và phân phối một ứng dụng 3D đa nền tảng nhỏ gọn và miễn phí. Vào thời điểm đó, đây là một khái niệm mang tính cách mạng vì hầu hết các ứng dụng 3D thương mại có giá hàng ngàn đô la. NaN hy vọng sẽ mang các công cụ mô hình hóa và hoạt hình 3D cấp độ chuyên nghiệp đến tầm tay của công chúng điện toán nói chung. Mô hình kinh doanh của NaN liên quan đến việc cung cấp các sản phẩm và dịch vụ thương mại xung quanh Blender. Năm 1999, NaN đã tham dự hội nghị SIGGRAPH đầu tiên của mình với nỗ lực quảng bá rộng rãi hơn Blender. Hội nghị SIGGRAPH đầu tiên của Blender là một thành công lớn và thu hút được sự quan tâm rất lớn từ cả báo chí và người tham dự. Blender là một đột phá và tiềm năng to lớn của nó đã được xác nhận!

Sau thành công của hội nghị SIGGRAPH vào đầu năm 2000, NaN đã có khoản tài trợ 4,5 triệu euro từ các nhà đầu tư mạo hiểm. Dòng tiền mặt lớn này cho phép NaN nhanh chóng mở rộng hoạt động. Chẳng mấy chốc, NaN đã công bố có tới 50 nhân viên làm việc trên khắp thế giới đang cố gắng cải thiện và quảng bá Blender. Vào mùa hè năm 2000, Blender 2.0 đã được phát hành. Phiên bản Blender này đã tích hợp thêm công cụ trò chơi vào ứng dụng 3D. Đến cuối năm 2000, số lượng người dùng đăng ký trên trang web NaN đã vượt quá 250.000 người.

Thật không may, tham vọng và cơ hội của NaN không phù hợp với khả năng của công ty và thực tế thị trường thời đó. Việc mở rộng quá mức này đã dẫn đến việc khởi động lại NaN bằng nguồn vốn của nhà đầu tư mới và một công ty nhỏ hơn vào tháng 4 năm 2001. Sáu tháng sau, sản phẩm phần mềm thương mại đầu tiên của NaN, Blender Publisher đã được ra mắt. Sản phẩm này được nhắm mục tiêu vào thị trường mới nổi của phương tiện truyền thông 3D dựa trên web tương tác. Do doanh số đáng thất vọng và môi trường kinh tế khó khăn đang diễn ra, các nhà đầu tư mới quyết định đóng cửa tất cả các hoạt động của NaN. Việc ngừng hoạt động cũng bao gồm việc ngừng phát triển Blender. Rõ ràng có những thiếu sót trong phiên bản Blender hiện tại, chẳng hạn như kiến ​​trúc phần mềm nội bộ phức tạp, các tính năng chưa hoàn thành và cách cung cấp giao diện không chuẩn, sự hỗ trợ từ cộng đồng người dùng và khách hàng đã mua Blender Publisher. Ton không thể biện minh cho việc Blender trở nên nhạt nhòa. Khởi động lại một công ty với một nhóm các nhà phát triển đủ lớn là không khả thi, Ton Roosendaal đã thành lập tổ chức phi lợi nhuận Blender Foundation vào tháng 3/2002.

Mục tiêu chính của Foundation Blender là tìm cách tiếp tục phát triển và quảng bá Blender như một nguồn mở dựa vào cộng đồng. Vào tháng 7 năm 2002, Ton đã xoay sở để khiến các nhà đầu tư NaN đồng ý với kế hoạch Blender Foundation là đơn vị duy nhất phát hành Blender dưới dạng nguồn mở. Chiến dịch Blender miễn phí trên YouTube đã tìm cách tăng 100.000 euro để Foundation Blender có thể mua bản quyền đối với mã nguồn Blender, quyền sở hữu trí tuệ từ các nhà đầu tư NaN và sau đó phát hành Blender cho cộng đồng nguồn mở. Với một nhóm tình nguyện viên nhiệt tình, trong số đó có một vài nhân viên cũ của NaN, một chiến dịch gây quỹ đã được triển khai cho Blender miễn phí. Trước sự ngạc nhiên và thích thú của mọi người, chiến dịch đã đạt được mục tiêu 100.000 euro chỉ trong bảy tuần ngắn ngủi. Vào Chủ nhật, ngày 13 tháng 10 năm 2002, Blender được phát hành ra thế giới theo các điều khoản của GPL GNU. Sự phát triển của Blender tiếp tục cho đến ngày nay, được thúc đẩy bởi một nhóm các tình nguyện viên tận tâm từ khắp nơi trên thế giới do người sáng tạo ban đầu của Blender, Ton Roosendaal.

***2.2.3. Tính năng của Blender***

***2.2.3.1. Mô hình hóa - Modling***

Việc tạo ra một cảnh 3D cần ít nhất ba thành phần chính: Mô hình, vật liệu và ánh sáng. Trong phần này, phần đầu tiên được đề cập, đó là mô hình hóa. Mô hình hóa chỉ đơn giản là nghệ thuật và khoa học tạo ra một bề mặt mô phỏng hình dạng của một đối tượng trong thế giới thực hoặc thể hiện trí tưởng tượng của bạn về các đối tượng trừu tượng.

Chế độ mô hình:

* Tùy thuộc vào loại đối tượng bạn đang cố gắng mô hình hóa, có các loại chế độ mô hình hóa khác nhau. Vì các chế độ không dành riêng cho việc lập mô hình nên chúng được đề cập trong các phần khác nhau của sách hướng dẫn.
* Chuyển đổi giữa các chế độ trong khi mô hình hóa là phổ biến. Một số công cụ có thể khả dụng ở nhiều chế độ trong khi những công cụ khác có thể chỉ dành cho một chế độ cụ thể.

Chế độ biên soạn - chỉnh sửa:

* Chế độ chỉnh sửa là chế độ chính nơi mô hình hóa diễn ra. Chế độ Chỉnh sửa được sử dụng để chỉnh sửa các loại đối tượng sau:
* Grid
* Đường Cong
* The Face - Surfaces
* Đối tượng văn bản
* Giàn lưới

Bạn chỉ có thể sửa đổi lưới của các đối tượng bạn đang chỉnh sửa. Để sửa đổi các đối tượng khác, bạn có thể rời khỏi Chế độ chỉnh sửa, chọn một đối tượng khác và vào Chế độ chỉnh sửa hoặc sử dụng Chỉnh sửa đa đối tượng.

***2.2.3.2. Kết xuất - Rendering***

Rendering là quá trình biến một cảnh 3D thành một hình ảnh 2D. Blender bao gồm ba động cơ kết xuất với các sức mạnh khác nhau:

* Eevee là một trình kết xuất thời gian thực dựa trên vật lý.
* Cycles là một công cụ đánh dấu đường đi dựa trên vật lý.

Bàn làm việc được thiết kế để bố trí, lập mô hình và xem trước.

Nhiều trình kết xuất hơn từ các nhà phát triển bên thứ ba có sẵn dưới dạng tiện ích bổ sung. Mỗi trình kết xuất có cài đặt kết xuất riêng để kiểm soát chất lượng và hiệu suất kết xuất.

Kết xuất trông như thế nào được xác định bởi máy ảnh, ánh sáng và vật liệu. Chúng được chia sẻ giữa Eevee và Cycles, tuy nhiên một số tính năng chỉ được hỗ trợ trong cái này hay cái kia.

Các kết xuất có thể được chia thành các lớp và chuyển, sau đó có thể được kết hợp với nhau để kiểm soát sáng tạo hoặc để kết hợp với cảnh quay thực. Freestyle có thể được sử dụng để thêm kết xuất đường thẳng không ảnh thực.

Blender hỗ trợ kết xuất khung nhìn 3D tương tác cho tất cả các công cụ kết xuất, để lặp lại nhanh chóng về ánh sáng và đổ bóng. Khi điều này được thực hiện, hình ảnh hoặc hoạt ảnh chất lượng cuối cùng có thể được hiển thị và xuất ra.

***2.2.3.3. Hoạt họa và dàn dựng - Animation & Rigging***

Hoạt hình: là làm cho một đối tượng di chuyển hoặc thay đổi hình dạng theo thời gian. Các đối tượng có thể được làm động theo nhiều cách:

* + Di chuyển toàn bộ như một vật thể: Thay đổi vị trí, hướng hoặc kích thước của chúng kịp thời;
  + Biến dạng chúng: Hoạt hình các đỉnh hoặc điểm kiểm soát của chúng;
  + Thừa Kế: Làm cho đối tượng di chuyển dựa trên chuyển động của một đối tượng khác (ví dụ: cha, móc, phần ứng của nó, v.v.).

Riggdering: là một thuật ngữ chung được sử dụng để thêm các điều khiển vào các đối tượng, thường cho mục đích hoạt ảnh. Rigging thường liên quan đến việc sử dụng một hoặc nhiều tính năng sau:

* Trang bị vũ khí: điều này cho phép các đối tượng lưới có các khớp linh hoạt và thường được sử dụng cho hoạt ảnh xương.
* Ràng buộc: để kiểm soát các loại chuyển động có ý nghĩa và thêm chức năng cho giàn khoan.
* Công cụ sửa đổi đối tượng: biến dạng lưới có thể khá liên quan, có nhiều công cụ sửa đổi giúp kiểm soát điều này.
* Phím hình dạng: để hỗ trợ các hình dạng mục tiêu khác nhau (chẳng hạn như nét mặt) được kiểm soát.
* Trình điều khiển: Vì vậy, giàn khoan của bạn có thể kiểm soát nhiều giá trị khác nhau cùng một lúc, cũng như làm cho một số thuộc tính tự động cập nhật dựa trên những thay đổi ở nơi khác.

Rigging có thể nâng cao như dự án của bạn yêu cầu, các giàn đang xác định một cách hiệu quả giao diện người dùng của riêng người làm hoạt hình để sử dụng mà không cần phải quan tâm đến các cơ chế cơ bản.

***2.2.3.4. Biên soạn phim - Video Editing***

Ngoài việc tạo mô hình và hoạt hình, Blender có thể được sử dụng để chỉnh sửa video. Có hai phương pháp khả thi cho việc này, một là Compositor. Tuy nhiên, chương này mô tả chương kia, Trình chỉnh sửa trình tự video (VSE), đôi khi được rút ngắn thành "Trình tự trình tự". Sequencer trong Blender là một hệ thống chỉnh sửa video hoàn chỉnh cho phép bạn kết hợp nhiều kênh video và thêm hiệu ứng cho chúng. Bạn có thể sử dụng các hiệu ứng này để tạo các chỉnh sửa video mạnh mẽ, đặc biệt khi bạn kết hợp nó với sức mạnh hoạt hình của Blender!

Để sử dụng VSE, bạn tải nhiều video clip và xếp chúng từ đầu đến cuối (hoặc trong một số trường hợp, chồng lên chúng), chèn các phần mờ và chuyển tiếp để liên kết phần cuối của một clip với phần đầu của một clip khác. Cuối cùng, bạn có thể thêm âm thanh và đồng bộ hóa thời gian của chuỗi video để khớp với nó.

***2.2.3.5. Tạo chất liệu – Texturing***

Khi dựng xong model thì ta sẽ phải tô màu cho nó. Để tô màu ta sử dụng Texturing. Chức năng này giúp chúng ta tô màu được nhanh chóng, tỉ mỹ và trông thực hơn. Chức năng này hỗ trợ người dùng các chức năng tô màu bằng ảnh, pixel, khối, … và chỉnh các thông số màu sắc qua các tập lệnh được thiết lập sẵn.

***2.2.3.6. Mô phỏng – Simulator***

Chức năng này giúp người sử dụng chạy mô phỏng các hiệu ứng, ánh sáng, hoạt động, … được thiết lập trên các model.

***2.2.3.7. Nhập và xuất bản vẽ***

Người dùng có thể nhập và xuất dưới nhiều định dạng khác nhau như stl, obj, fbx, ….

**2.3. Phần mềm PowerPoint**

### ***2.3.1. Tổng quan phần mềm PowerPoint***

Microsoft PowerPoint (gọi tắt là PowerPoint) là một phần mềm trình chiếu do hãng Microsoft phát triển. PowerPoint là một phần của gói ứng dụng văn phòng Microsoft Office. Nó có thể cài đặt và sử dụng được trên cả máy tính dùng hệ điều hành Windows lẫn Mac OS X. Bản dùng cho hệ điều hành Windows còn có thể dùng cho cả các máy tính với hệ điều hành Linux nhờ lớp tương thích Wine.

Trong quá trình phát triển hiện nay phần mềm PowerPoint đã có them nhiều tính năng mới hỗ trợ thiết kế Slide 2D và xuất Slider ra các dạng định dạng khác nhau như mp4, png, jpg, … Vì sự dễ dàng sử dụng và có khá đầy các tính năng cho thiết kế 2D nên đã được các nhà thiết kế 2D nghiệp dư sử dụng để tạo ra các sản phẩm như thiệp mời, poster, …

***2.3.2. Lịch sử phát triển***

PowerPoint là một trong những phần mềm của bộ Microsoft Office. Vì vậy quá trình phát triển của PowerPoint sẽ được viết theo Microsoft Office.

Từ thời điểm Windows xuất hiện vào năm 1990, MS Office 1.0 xuất hiện như một cuộc cách mạng trong soạn thảo văn bản với thiết bị trỏ đi kèm làm cho việc soạn thảo và định dạng dễ dàng hơn bao giờ hết với giao diện đồ họa. Bộ Office 1.0 bao gồm các ứng dụng Word 1.1, Excel 2.0 and PowerPoint 2.0 được phát hành vào tháng 11 năm 1990.

Năm 1991, Microsoft phát hành phiên bản mở rộng Office 1.5 vẫn giữ bộ MS Word và PowerPoint nhưng cung cấp các nâng cấp đáng kể trên ứng dụng Excel với tên gọi Excel 3.0 cung cấp khả năng định dạng bảng tính linh hoạt với thiết bị trỏ (mouse – chuột máy tính).

Năm 1992, Office 3.0 được phát hành trên CDROM: Word 2.0; Excel 4.0A và PowerPoint 4.0. cung cấp những thay đổi đáng kể trong giao diện và làm tiền đề để phát triển cho bộ Office 95 sau này. Phiên bản này có 2 bộ cài đặt trên CDROM.

Năm 1994, Office 4.0: phiên bản này không có nhiều nâng cấp đáng kể nào trên Excel và PowerPoint ngoại trừ sự thay đổi lớn trong Word với phiên bản Word 6.0.

Năm 1995, sau thời gian háo hức chờ đợi sự xuất hiện của Windows 95, người dùng đón nhận phiên bản Office 95 (tên gọi khác của Office 7.5) với bộ ứng dụng gọi thân thuộc là Word 95, Excel 95 và Presentation/Powerpoint 95, Access 95, không có nhiều thay đổi về phần giao diện, riêng phần soạn thảo văn bản được cung cấp rất nhiều kiểu chữ mới và bắt đầu xuất hiện các kiểu chữ tiếng Việt và nhiều bộ gõ chữ mới như (VietKey, VNI Tân Kỳ,…) cũng như nhiều trường phái phân biệt Bắc – Nam, ví dụ miền Bắc dùng chủ yếu kiểu chữ ABC (TCVN); miền Nam sử dụng kiểu chữ VNI gây không ít khó khăn cho người dùng khi di chuyển văn bản từ Bắc vào Nam và ngược lại.

Mùa thu năm 1996, Microsoft phát hành phiên bản nâng cấp Office 97 với sự xuất hiện của anh bạn Office Assistant vô tích sự và đáng ghét, thay đổi cơ bản giao diện nút lệnh sang giao diện phẳng.

Năm 1999, sau một thời gian dài, Microsoft mới phát hành phiên bản Office 2000, tuy nhiên chỉ là bản nâng cấp và sửa lỗi cho phiên bản trước đó với một câu hiệu “Better User Experience”, với giao diện được cải thiện.

Giữa năm 2001, Windows chuyển từ nền tảng 9x sang tên gọi XP đồng thời bộ Office cũng được nâng cấp lên phiên bản cũng XP với giao diện mượt mà thời điểm đó nhưng nhưng hiện tại nó được xem là “sếnh súa”. Đây là thời điểm kiện tụng việc độc quyền trong việc phát triển trình duyệt web đi kèm theo hệ điều hành và nhà phát triển phần mềm khác (Nescape, Opera, …) nên trình duyệt web không được kèm vào Windows mà được kèm vào bộ phần mềm Office XP.

Office 2003, được phát hành vào mùa thu năm 2003, phiên bản được dùng nhiều nhất trong tất cả phiên bản, với giao diện dễ dùng, xúc tích, gần như đáp ứng tốt mọi nhu cầu của người sử dụng với rất nhiều chức năng và bảo mật tốt. Giao diện dễ dùng đến mức người dùng không muốn từ bỏ bộ Office này.

Dần dần các phiên bản Office được đặt theo năm phát hành, Microsoft lần đầu tiên giới thiệu giao diện Ruban (ribbon) khi phát hành Office 2007. Giao diện ribbon ban đầu những tưởng cung cấp nhiều lựa chọn, hiện thực hóa các lệnh từ menu như các phiên bản trước bằng các nút lệnh, do đó Microsoft phải sắp xếp các nhóm lệnh vào từ tab mà Microsoft đặt cho giao diện này tên gọi là Ribbon Interface. Tuy nhiên vì sự sắp xếp mới này mà người dùng lúng túng và không chịu chấp nhận nó đến khi Microsoft ép buộc người dùng phải từ bỏ nó khi chuyển sang các thế hệ kế tiếp và tuyên bố dừng hỗ trợ Office 2003. Microsoft chính thức chấp nhận định dạng PDF.

Office 2010, bộ ứng dụng văn phòng bước vào giai đoạn phát triển mới cho môi trường cộng tác và điện toán đám mây, cơ bản không có nhiều thay đổi so với phiên bản Office 2007, giao diện mới tươi sáng hơn.

Office 2013 / Office 365, phiên bản mới hướng đến môi trường doanh nghiệp và tích hợp cloud vào ứng dụng, cho phép người dùng lưu thẳng tài liệu của mình lên tài khoản cá nhân, và có thể làm việc bất kì nơi nào có môi trường Internet.

Từ phiên bản Office 2016 trở lên, Microsoft đã có một quá trình tiến hóa để thay đổi và thích nghi với người dung với những phát triển của thời đại công nghệ mà mọi thứ có thể khác đi vào sáng hôm sau.

### ***2.3.3. Tính năng của PowerPoint***

***2.3.3.1. Tính năng cơ bản***

Người dùng có thể tạo, chèn, xóa slide; thêm các khối hình học; text; căn lề; khoảng cách dòng và đoạn; thiết lập độ ưu tiên hiển thị cho đối tượng; thiết lập fort, cỡ và màu chữ; gợi ý về thiết kế slider.

***2.3.3.2. Tính năng Insert***

Người dùng có thể chèn bảng, ảnh, biểu đồ, đối tượng 3D, video, link, công thức, kí tự đặc biệt, ngày tháng, số thứ tự slide, comment, heart & Footer, WordArt, …

***2.3.3.3. Tính năng Draw***

Người dùng có thể vẽ, sử dụng thước, tạo link tới text, link tới Shape, link tới Math, lnk Replay.

***2.3.3.4. Tính năng Design***

Người dùng có thể thiết kế background như tùy chỉnh kích thước, chèn ảnh nền, sử dụng thiết kế có sẵn hoặc thiết kế từ ngoài. Đặc biệt là tính năng gợi ý thiết kế cho người dùng.

***2.3.3.5. Tính năng Transitions và Animation***

Giúp người dùng tạo các hiệu ứng cho bài thuyết trình hoặc video như hiệu ứng chuyển slide, âm thanh, xuất hiện, trình diễn và rời khỏi slide.

***2.3.3.6. Tính năng Slide Show***

Giúp trình chiếu, quay video, thay đổi các chế độ trình chiếu tùy chỉnh, cài đặt các phím và tính năng hỗ trợ như thời gian chuyển slide, vẽ trong lúc trình chiếu, …

***2.3.3.7. Tính năng Review***

Giúp người dùng kiểm tra thay đổi và sửa chữa, viết comment, …

***2.3.3.8. Tính năng View và hỗ trợ khác***

Giúp người dùng thay đổi giao diện làm việc, xem thông tin bài thuyết trình, lưu bài dưới các dạng khác nhau, in ấn, chia sẻ và xuất file dưới các dạng đặc biệt như mp4, png, jpg, gif, cd, ….

## **2.4. Phần mềm Unity**

### ***2.4.1. Tổng quan phần mềm Unity***

Unity3D là phần mềm làm games trực tiếp theo thời gian thực, mà không cần render, cho phép người  design game có thể thiết kế Interface Graphic, map hay character … từ một phần mềm thứ 2 (thường là các phần mềm thiết kế đồ họa chuyên nghiệp như 3Dsmax, Blender, Maya, XSL, Cinema4D, Cheetah3D, Modo, Autodesk FBX, LightWave…) sau đó chỉ việc import nó vào trong Unity với định dạng của tập tin là \*.FBX hay \*.dae, \*.3DS, \*.dxf và \*.obj, nhưng định dạng \*.FBX hay được dùng vì được tối ưu hóa hơn, còn dùng các định dạng khác ngoài FBX thì phải cài phần mềm thiết kế character tưng ứng thì mới dùng được (tức là dùng Maya có định dạng \*.mb, \*.ma thì muốn dùng được phải cài Maya)

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

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

Unity3D có 2 loại phiên bản, một cho người dùng free, và một phiên bản pro thì mất phí. Khi tải Unity về các bạn cài đặt bình thường, đến khi kết thúc cài đặt nền các bạn chọn phiên bản Pro thì cần phải mua, còn nếu là người dùng free thì các bạn chọn kích  hoạt qua web, chỉ cần có kết nối mạng internet, sau đó chọn cài đặt thủ công và nhập Email, chọn free là bạn có thể sử dụng một bản Unity free.

Unity được tạo ra để sản xuất game nhưng trong quá trình phát triển đặc biệt là sự phát triển mạnh mẽ của XR thì nó còn được dùng để tạo ra các sản phẩm XR một cách nhanh chóng, dễ dàng và mang lại lợi nhuận cao.

### ***2.4.2. Lịch sử phát triển***

Unity game engine khởi động năm 2005, nhằm "dân chủ hóa" phát triển game bằng cách cung cấp cho nhiều nhà phát triển hơn. Năm tiếp theo, Unity được vinh danh là á quân trong hạng mục Best Use of Mac OS X Graphics tại Apple Design Awards của [Apple Inc.](https://vi.wikipedia.org/wiki/Apple_Inc.) Unity ban đầu được phát hành cho Mac OS X, sau đó bổ sung hỗ trợ cho Microsoft Windows và trình duyệt Web.

Unity 2.0 ra mắt năm 2007 với khoảng 50 tính năng mới. Bản phát hành bao gồm một công cụ địa hình được tối ưu hóa cho môi trường 3D, real-time dynamic shadows, directional lights và spotlights, phát lại video và các tính năng khác. Bản phát hành cũng bổ sung các tính năng nhờ đó các nhà phát triển có thể cộng tác dễ dàng hơn. Nó bao gồm một Networking Layer để các nhà phát triển tạo game nhiều người chơi dựa trên [User Datagram Protocol](https://vi.wikipedia.org/wiki/UDP), cung cấp [Network Address Translation](https://vi.wikipedia.org/wiki/Bi%C3%AAn_d%E1%BB%8Bch_%C4%91%E1%BB%8Ba_ch%E1%BB%89_m%E1%BA%A1ng), State Synchronization, và [Remote Procedure Calls](https://vi.wikipedia.org/w/index.php?title=Remote_procedure_call&action=edit&redlink=1).

Khi Apple ra mắt [App Store](https://vi.wikipedia.org/wiki/App_Store_(iOS)) của họ năm 2008, Unity nhanh chóng bổ sung các hỗ trợ cho [iPhone](https://vi.wikipedia.org/wiki/IPhone). Trong vài năm, công cụ này đã không được thử nghiệm trên iPhone và nó trở nên nổi tiếng với các nhà phát triển trò chơi iOS.

Unity 3.0 ra mắt tháng 9 năm 2010 với tính năng mở rộng các tính năng đồ họa của engine cho [máy tính để bàn](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_%C4%91%E1%BB%83_b%C3%A0n) và [video game consoles](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%C6%A1i_game_console). Ngoài hỗ trợ Android, Unity 3 còn có tính năng tích hợp công cụ Beast Lightmap của Illuminate Labs, deferred rendering, một built-in tree editor, kết xuất phông chữ gốc, ánh xạ UV tự động và bộ lọc âm thanh, cùng nhiều thứ khác.

Năm 2012, VentureBeat phát biểu rằng "rất ít công ty đóng góp vào sự phát triển trò chơi độc lập như Unity Technologies.... hơn 1,3 triệu người dùng đang sử dụng Unity để phát triển game trên iOS, Android, máy chơi game, PC và trò chơi trên web. Unity sẽ mở ra thời kỳ phát triển công cụ phát triển trò chơi đa nền tảng". Một khaỏ sát vào tháng 5/2012 của tạp chí Game Developer đã chỉ ra Unity là [game engine](https://vi.wikipedia.org/wiki/Game_engine) hàng đầu dành cho nền tảng di động for mobile platforms. Tháng 11 năm 2012, Unity Technologies phát hành Unity 4.0. Phiên bản này bổ sung các hỗ trợ cho [DirectX 11](https://vi.wikipedia.org/wiki/DirectX) và [Adobe Flash](https://vi.wikipedia.org/wiki/Adobe_Flash), các công cụ hoạt ảnh mới có tên Mecanim, và quyền truy cập vào bản xem trước Linux.

Đối với Apple Design Awards tại cuộc triển lãm thương mại WWDC 2006, [Apple, Inc.](https://vi.wikipedia.org/wiki/Apple_Inc.) đã gọi Unity là Best Use of Mac OS X Graphics. Một năm sau khi Unity ra mắt tại cùng một triển lãm thương mại Unity Technologies cho biết đây là lần đầu tiên một công cụ phát triển game được trao giải thưởng này. Một cuộc khảo sát tháng 5 năm 2012 của tạp chí Game Developer đã khẳng định Unity là công cụ phát triển trò chơi hàng đầu trên di động. Vào tháng 7/2014, Unity đã đạt giải "Best Engine" tại liên hoan trao giải thưởng công nghệ hàng năm của Anh.

Unity 5 cũng nhận được những lời khen tương tự, với The Verge đã nói rằng "Unity giúp việc phát triển game trở nên dễ dàng và phổ biến. Unity 5 là một biến tiến đáng mong đợi trong tương lai."

Sau khi phát hành Unity 5, Unity Technologies bị chỉ trích rằng khổi lượng lớn các game phát hành trên Steam là của các nhà phát triển thiếu kinh nghiệm.[[33]](https://vi.wikipedia.org/wiki/Unity_(ph%E1%BA%A7n_m%E1%BB%81m_l%C3%A0m_game)#cite_note-33) CEO John Riccitiello giải thích trong một cuộc họp báo rằng đây là tác dụng phụ khi chúng ta xã hội hóa việc phát triển game: "If I had my way, I'd like to see 50 million people using Unity – although I don't think we're going to get there any time soon. I'd like to see high school and college kids using it, people outside the core industry. I think it's sad that most people are consumers of technology and not creators. The world's a better place when people know how to create, not just consume, and that's what we're trying to promote." (tạm dịch: nếu chúng tôi có cách, tôi muốn 50 triệu người sử dụng Unity- dù chúng tôi biết rất khó đạt được điều đó. Tôi muốn thấy những đứa trẻ ở trường Trung học và Đại học sử dụng nó, những người ngoài ngành. Tôi nghĩ rằng thật buồn khi hầu hết mọi người là người tiêu dùng về công nghệ, chứ không phải người sáng tạo. Thế giới là một nơi tốt đẹp hơn khi người ta biết làm thế nào để tạo ra, không chỉ tiêu thụ, và đó là những gì chúng tôi đang cố gắng để thúc đẩy).

Vào tháng 12 năm 2016, Unity Technologies thông báo họ sẽ thay đổi hệ thống số phiên bản trên Unity từ định theo thứ tự sang năm phát hành để sắp xếp các phiên bản với nhịp cập nhật thường xuyên hơn.

***2.4.3. Giới thiệu giao diện và tính năng của Unity***

Sau khi cài đặt Unity thành các bạn khởi động Unity:

* Đối với Window: Start -> Program -> Unity, hoặc chọn biểu tượng Unity trên screen.
* Đối với người dùng MacOs: Application -> Unity.

Sau khi khởi động Unity cửa sổ Editor xuất hiện: cửa sổ Editor của Unity gồm nhiều Tab gọi là các view và ta có thể tùy ý kéo thả sắp xếp các view theo ý thích.

***2.4.3.1. Cửa sổ project***

Mỗi Project của Unity đều chứa một thư mục Assets. Nội dung của thư mục này được hiển thị trong Project View. Đây là nơi chứa tất cả các assets để tạo Game của bạn như Scenes, Script, 3D models, Textures, Audio, Prefabs. Chúng ta không nên di chuyển assets bằng cách sử dụng Window Explorer (hay Finder trong MAC), vì nó sẽ phá vỡ cấu trúc siêu dữ liệu của assets với nhau, và luôn sử dụng Project View để cấu trúc assets của mình.

Để thêm assets vào project của mình, chúng ta có thể kéo thả bất kỳ file nào vào trong project view hoặc vào công cụ Assets – Import new Assets (Click chuột phải vào Project View). Scenes cũng được lưu trữ trong Project view, và đây là một level độc lập mang tính cá nhân. Chúng ta dễ dàng tạo một assets game trong Unity bằng cách chọn hình tam giác nhỏ nằm bên phải Create trong cửa sổ Project hoặc click chuột phải trong Project View, sau đó chọn assets tương ứng. Ngoài ra bạn cũng có thể thêm các assets và đổi tên các assets một cách dễ dàng.

***2.4.3.2. Hierarchy (hệ thống phân cấp)***

Trong Hierarchy chứa các GameObject hiện thời, một số có thể trỏ trực tiếp tới những file assets như 3D models, một số khác đại diện cho Prefabs – những đối tượng đã được tùy biến, dùng làm các công việc khác nhau sau này trong Game của bạn. Bạn có thể chọn và parenting Object trong Hierarchy. Một Object có thể được thêm vào hay loại bỏ trong scene và có thể thấy nó mất đi hay xuất hiện trong Hierarchy.

***2.4.3.3. Parenting***

          Tức là thư mục chứa hay thư mục gốc, bất kỳ một gameobject nào muốn là đối tượng con (child) thì ta chỉ việc kéo thả đối tượng đó vào trong đối tượng dự tính làm Parenting trong Hierarchy và nó sẽ kế thừa chuyển động và quay của parenting.

***2.4.3.4. Toolbar***

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

* + Transform Tool: được dung với Scene view, như quay trái, phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.
  + Transform Gizmo Toggles: dùng cho việc thể hiện Scene view.
  + Play/Pause/Step Buttons: dùng cho view game, chơi game ngay trong Editor để kiểm tra.
  + Layer Drop - down kiểm soát đối tượng nào đang được thực hiện trong Scene view
  + Layout Drop - up kiểm soát sự sắp xếp của các Views.

***2.4.3.5. Scene View***

Là nơi bạn Design Game của bạn, đối tượng Maneuvering và Important In trong Scene view (chuyển động và điều khiển) là hai trong số các chức năng quan trọng của Unity, ở góc bên phải của Scene là Scene Gizmo, nó thể hiện hướng nhìn trong không gian của camera trong Scene View hiện thời, cho phép thay đổi góc nhìn trực quan và nhanh chóng.

         Click lên các nhánh hình nón để chuyển qua các góc nhìn khác nhau có thể xem ở chế độ Isometric Model (tức ở dạng mặt cắt hai chiều), để chuyển qua chế độ 3D bạn Click vào hình vuông ở giữa hay giữ phím Shift + Click để chuyển đổi chế độ nhìn.

Khi xây dựng một Game, bạn sẽ đặt rất nhiều đối tượng vào trong Game của bạn. Bạn có thể dử dụng các công cụ Transform Tools ở trong Toolbar để di chuyển, xoay, phóng to tu nhỏ từng đối tượng.Khi bạn chọn một đối tượng trong Scene View, xung quanh đối tượng được chọn sẽ có những thay đổi tương ứng với từng chế độ trong Transform Tools. Sau đó bạn thay đổi đối tượng tùy ý, nếu muốn chính xác bạn có thể chỉnh chi tiết ở bảng Inspector.

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

***2.4.3.6. Game View***

Game View được rendered từ những Camera trong Game. Đó là những gì được nhìn thấy khi hoàn tất, khi Game được xuất bản. Bạn sẽ cần ít nhất là một hoặc nhiều hơn số lượng các Camera để quyết định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game.

***2.4.3.7. Play Mode***

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

***2.4.3.8. Inspector***

      Games trong Unity được tạo ra bởi tập hợp rất nhiều GameObject, trong đó bao gồm meshes, scripts, âm thanh, hay những đối tượng Graphic như nguồn sáng, … Inspector sẽ hiển thị mọi thông tin về đối tượng đang làm việc một cách chi tiết, kể cả những Components được đính kèm và những thuộc tính của nó. Tại đây bạn có thể điều chỉnh, thiết lập mọi thông số chức năng của những mối liên kết GameObject-Component.

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

Chúng ta có thể click lên icon hình bánh răng nhỏ bên phải hay click chuột phải lên tên của Component để xuất hiện context menu dành cho những thiết lập của Component. Inspector cũng sẽ thể hiện mọi thông số Import Setting của assets đang làm việc.

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

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

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

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

Bất kỳ lúc nào bạn cũng có thể right-click lên một tab của bất kỳ view nào để xem những tính năng khác như Maximize hay thêm vào một tab mới trong cùng một cửa sổ.

***2.4.3.10. Quản lý Asset***

Workflow Là “tiến trình công việc” hay “quy trình làm việc”, tức là khi bạn import hoặc tạo ra asset, bạn không cùng một lúc tạo ra toàn bộ, mà là theo trình tự, tùy theo nhu cầu, theo kịch bản, bạn import hoặc tạo ra asset theo thứ tự.

Tạo một Asset thô: Unity hỗ trợ import hầu như tất cả các định dạng 3D model hiện nay. Khi một đối tượng hay vật thể 3D được import hoặc tạo ra trong Unity, nó đều được gọi là asset.

Import Settings: khi bạn chọn một asset trong Project View, những thông số import cho asset sẽ hiển thị ở Inspector. Mỗi loại asset sẽ có mỗi kiểu thông số khác nhau.

Thêm Asset vào trong Scene: chỉ cần click-drag các khung lưới được vẽ trong 3D (mesh) từ Project View vào cửa sổ Hierarchy hoặc Scene View để thêm nó vào trong Scene. Khi thêm một mesh vào trong scene, tức là bạn đã tạo ra một GameObject có Mesh Renderer Component. Tương tự khi thêm vật liệu, hoặc âm thanh, bạn cũng phải gán nó vào một GameObject đã có sẵn trong Scene hay Project.

Mối liên hệ giữa một số kiểu assets thông dụng:

* Một Texture (kết cấu) được áp cho một Material (vật liệu).
* Một Material được áp cho một GameObject (với một Mesh Renderer Component).
* Một Animation được áp cho một GameObject (với một Animation Component).
* Một file âm thanh được áp cho một GameObject (với một Audio Source Component).

***2.4.3.11. Khởi tạo Scene***

Scene chứa tất cả các đối tượng trong Game, nó thường được dùng để tạp màn hình chính trong game, những level độc lập, và mọi thứ khác trong game. Tức là mỗi Scene là một tầng độc lập, và trong mỗi Scenes bạn có thể lập môi trường, vật thể, chi tiết, kể cả design và xây dựng game từ những thứ nhỏ nhất.

Instancing Prefabs: khi bạn đã có một prefabs bạn có thể copy nó ra thành nhiều bản một cách dễ dàng, đó gọi là Instancing prefabs. Để tạo Instancing của bất kỳ một Prefabs nào, chỉ cần drag nó từ trong Project view qua Hierarchy hay trực tiếp trong Scene view.

Thêm Component and Scripts: khi bạn đã có một Prefabs hoặc gameObject đang làm việc bạn có thể thêm các tính năng cho nó bằng cách sử dụng Component (Script cũng là một Component). Để thêm một Component chỉ cần chọn đối tượng rồi chọn Component tương ứng trong menu Component, bạn sẽ thấy xuất hiện các thông số trên cửa sổ Inspector. Script cũng được mặc định trong menu Component.

Trong trường hợp việc thêm Component làm gãy liên kết giữa những GameObject với prefab của nó thì bạn có thể dùng Gameobject -> Apply Changes to Prefab từ menu để thiết lập lại liên kết đó.

***2.4.4. Các thành phần khác của Unity***

***2.4.4.1. Placing GameObject***

Khi đã có một GameObject trong Scene bạn có thể dùng công cụ Transform Tools để di chuyển, hoặc thay đổi giá trị Transform trong Inspector để có vị trí chính xác hơn.

***2.4.4.2. Working Camera***

Camera chính là mắt trong Game, mọi thứ người chơi có thể nhìn thấy đều thông qua Camera, nó cũng được hiển thị và thay đổi, di chuyển như bất kỳ một GameObject nào, và nó thực chất cũng là một GameObject và nó có đầy đủ thuộc tính của GameObject ngoài ra nó còn có các thuộc tính riêng cho Camera. Unity đã cung cấp một số Script sẵn kèm theo gói cài đặt mẫu Asset khi bạn khởi tạo mới một project.

***2.4.4.3. Nguồn sáng***

Hầu như chúng ta phải thêm nguồn sáng cho các bối cảnh của Game. Có ba kiểu nguồn sáng cơ bản, mỗi nguồn sáng khác nhau sẽ thay đổi sắc thái của game rất nhiều.

* *Directional Light:* có thể hiểu là với nguồn sáng này ta có thể định hướng chiếu cho nó, các tia chiếu là song song.
* *Point Light:* là điểm sáng, thường dùng để tạo các điểm sáng nhỏ trên các Character, hay các cảnh trong Game theo đúng ý tưởng của chủ nhân.
* *Spotlight:* đèn tại chỗ chỉ chiếu theo một chiều hình nón, thường dùng là hiệu ứng chiếu cho đèn các loại xe, đèn pin, ánh sáng hắt vào object trong các cảnh…

***2.4.4.4. Camera***

Như Camera được sử dụng trong các bộ phim , Camera trong Unity cũng được dùng để hiển thị game trên thế giới cho người chơi. Nó cũng được coi là một Gameobject trong Unity. Bạn có thể xoay, di chuyển …. tùy chỉnh nó theo ý tưởng của bạn.

Camera được sử dụng để hiển thị cảnh trong game, chúng ta có thể làm cho game của mình trở nên độc đáo hơn nhờ tùy chỉnh Camera. Trong một cảnh, chúng ta có thể có một hoặc rất nhiều Camera. Đặc điểm Camera:

* Clear Flags: xác định các bộ phận mà màn hình sẽ bị xóa. Thuận tiện khi sử dụng nhiều máy ảnh và để vẽ nhiều đối tượng khác nhau. Không xóa nó sẽ hiển thị màu đen xì.
* Background: màu nền cho phần màn hình còn lại.
* Culling Mask: chỉ định các lớp đối tượng của bạn trong Inspector. Cho phép hoặc bỏ qua các đối tượng được hiển thị trong Camera.
* Projection:
* Perspective: Camera hiển thị các đối tượng theo phối cảnh trọn vẹn.
* Orthographic: hiển thị các đối tượng như một thể thống nhất, không có theo nghĩa của phối cảnh(Perspective).
* Size: kích thước quan sát của Camera khi chọn phép chiếu là: Orthographic.
* Field of view: chiều rộng của góc nhìn Camera. Được đo bằng độ dọc theo trục Local Y.
* Clipping plane:
  + - Near: khoảng cách gần nhất hiển thị trong Camera.
    - Far: khoảng cách xa nhất hiển thị được trong tầm nhìn của Camera.
* Normalized ViewPort Rect:
  + - X: bắt đầu từ vị trí ngang mà Camera hiển thị.
    - Y: bắt đầu từ vị trí thẳng đứng mà Camera hiển thị.
* Depth: vị trí Camera có một giá trị lớn hơn sẽ được hiển thị lên đầu. Tức là cái nào có giá trị lớn hơn thì nó sẽ được ưu tiên được hiển thị lên màn hình.
* Rendering path: tùy chọn cho việc xác định phương pháp vẽ những gì sẽ được hiển thị lên Camera:
* Use Player Setting: sử dụng cài đặt Player.
* Vertex Lit: tất cả các đối tượng được đưa ra bởi máy ảnh sẽ được trả lại như đối tượng Vertex-Lit.
* Forward: tất cả các đối tượng sẽ được trả về như một tài liệu. Giống như tiêu chuẩn Unity 2.x.
* Deferred Lighting (Unity Pro only): tất cả các đối tượng sẽ được rút ra mà không có ánh sáng, sau đó tất cả ánh sáng được trả lại trong hàng đợi Render (mất phí).
* Target Texture: phiên bản free 4.0 đã có. Tham chiếu đến một texture. Cho phép dựng hình High Dynamic Range cho máy ảnh này.

***2.4.4.5. Địa hình***

Trên Menu chính, chọn Create Terrain:

* Terrain Script: tại đây chúng ta có thể vẽ địa hình, nâng/hạ địa hình, vẽ cây, cỏ, cài đặt cho Terrain.
* Brushes: gồm một số dạng bút lông phục vụ việc vẽ địa hình.

***2.4.4.6. Mắt lưới***

Khi một đối tượng mô hình 3D được Import. Unity sẽ gán cho nó một đại diện như một bộ lưới. Lưới phải được đính kèm với một Gameobject bằng cách sử dụng một thành phần Filter Meshes.

***2.4.4.7. Định dạng***

  Export file định dạng 3D như: \*.FBX, \*.OBJ.

* Ưu điểm:
* Chỉ xuất ra dữ liệu bạn cần.
* Kiểm tra lại dữ liệu.
* Nhìn chung tập tin rất nhẹ.
* Khuyến khích phương pháp Module.
* Hỗ trợ các gói 3D.
* Nhược điểm:
* Có thể gây chậm trong việc tạo mẫu và sử dụng lại.
* Khó có thể theo dõi

  Sở hữu các file ứng dụng 3D như: \*.Max, \*.Blend.

* Ưu điểm:
* Nhanh chóng lặp đi lặp lại các quá trình.
* Đơn giản việc sử dụng ban đầu.
* Nhược điểm:
* Công cụ hay bản sao của định dạng đó phải được cài đặt trên máy.
* Tập tin có thể cồng kềnh với các file dữ liệu không cần thiết.
* Các tập tin lớn có thể làm chậm cập nhật của Unity.
* Chưa xác nhận khó khăn để khắc phục sự cố các vấn đề.

***2.4.4.8. Animation***

Để sử dụng Animation cơ bản, trước tiên chọn Create new clip. Animation sẽ hiển thị các tùy chọn trên Animation View hoặc trên Inspector. Để áp dụng animation cho một đối tượng nào đó bạn phải chọn vào đối tượng đó và thuộc tính animation tương ứng sẽ hiển thị trên Animation View tương ứng.

Trên Animation view có thể chọn một thuộc tính hoặc một vào thuộc tính để tùy biến hoạt hình theo các thuộc tính của đối tượng. Chính sửa Curves bằng cách tại các frame đã được tạo, kéo đường kẻ theo ý bạn muốn. Nó sẽ hiển thị ngay trên Scene View hoặc trên Inspector.

Tại các điểm Frame, có thể xóa frame, chỉnh/chọn tiếp tuyến, …. Có thể sử dụng button nhỏ (+) để thêm Frame hoặc Event cho Animation hoặc click-right lên thanh frame.

* Animation Scripting (Legacy): hệ thống Animation của Unity cho phép bạn tạo đẹp nhân vật của mình bằng các pha trộn các vật liệu, chất phụ gia, đồng bộ hóa thời gian, chu kì, các lớp ảnh động, kiểm soát tất cả các các khía cạnh phát lại của hình ảnh động(thời gian, tốc độ, sự pha trộn – trọng lượng). Một nhân vật hoạt hình luôn có hai vấn đề chính đó là sự di chuyển của nó trong thế giới và tạo hiệu ứng cho phù hợp.
* Animation Blending: trong các trò chơi ngày nay, hình ảnh động hòa trộn làm cho nhân vật trở nên trơn tru hơn. Tại bất cứ thời điểm nào trong game, bạn đều có thể chuyển đổi qua lại giữa hình ảnh động và hình ảnh tĩnh. Đây là nơi các hình ảnh động được Blending. Tất cả các hình ảnh động được thêm, pha trộn vào nhau để được hình ảnh động cuối cùng.Ví dụ: tạo một chuyển động đơn giản để xem nhân vật chuyển động nhanh thế nào, và rồi chuyển đổi giữa hoạt hình động và hoạt hình tĩnh.
* Animation Layers: là một khái niệm hữu ích giúp bạn nhóm các hình ảnh động và ưu tiên cho chúng. Hệ thống hoạt hình của Unity có thể pha trộn giữa nhiều Clip hoạt hình theo ý bạn muốn. Có thể gán lượng pha trộn bằng tay hay sử dụng animation.CrossFade(). Lượng Blend luôn được bình thường hóa trước khi được áp dụng. Tưởng tượng như bạn có một chu kỳ Walk và một chu kỳ chạy, cả hai đều có lượng là 1(100%). Khi bình thường hóa thì mỗi chu kỳ chiếm 50% lượng. Tuy nhiên nếu bạn muốn cho bên nào đó có lượng lớn hơn thì bạn có thể Blend bằng tay.
* Animation Mixing: cho phép cắt giảm số Animation mà bạn cần trong game bởi chỉ cần một số Animation được áp dụng trên thân đối tượng. Điều đó có nghĩa là đối tượng hoạt hình có thể được sử dụng chung với các đối tượng hoạt hình khác. Bạn thêm một Animation Mixing biển đổi tới một Animation bởi lời gọi AddMixingTransform(). Ví dụ Mixing: trong hoạt hình, giống như khi bạn nghe thấy những làn sóng vỗ tay. Bạn có thể tạo ra những làn sóng vỗ tay hoặc khi nhàn rỗi tùy từng trường hợp đối tượng ở trạng thái nào.
* Animating Characters Procedurally: nhiều khi chúng ta sẽ cần hoạt hình hóa xương của nhân vật.
* Animation Playback and Sampling: làm thế nào để lấy lại được mẫu hoạt hình trong Unity khi người chơi sử dụng lại máy đó.

***2.4.4.9. Sound***

Audio Listener: giống như một thiết bị Microphone. Nó nhận đầu vào từ bất cứ một nguồn âm thanh nào trong Scenes và các âm thanh thông qua máy tính. Đối với hầu hết các ứng dụng, nó là ý nghĩa nhất để gắn tai nghe lên Main Camera.

Ranh giới / độ ảnh hưởng âm thanh nghe của Reverb Zone được áp dụng cho toàn Scene thì có thể nghe thấy âm thanh trên bất cứ địa điểm nào của Scene.

Properties: phải thêm âm thanh vào Scene và chỉnh sửa thuộc tính của âm thanh bên Inspector View.

* Audio Clip: chọn file âm thanh cho Scene.
* Mute: bật/tắt âm thanh.
* Bypass Effects: lọc nhanh hiệu ứng “bypass” to audio source. Một cách dễ dàng nhất để bật/tắt hiệu ứng(effect).
* Play on Wake: nếu enable, âm thanh sẽ được chạy ngay khi ra mắt Scene. Nếu để Disable, khi cần chạy âm thanh chúng ta phải gọi phương thức/chức năng Play() từ Script.
* Priority: xác định độ ưu tiên của Audio Source trong số tất cả các Audio Source có trong Scene.
* Priority = 0: rất quan trọng. Sử dụng ở mức 0 cho bài nhạc để tránh bị thường xuyên trao đổi.
* Priority =256: độ quan trọng thấp nhất(Độ ưu tiên ở mức thấp nhất).
* Mặc định (default) = 128.
* Volume: làm thế nào để âm thanh lớn trên một bộ phận của thế giới Scene từ Audio Source.
* Pitch: số dùng thay đổi tốc độ của âm thanh. Tốc độ bằng 1 là ở mức chạy bình thường.
* 3D Source Setting: cài đặt, cái mà được áp dụng cho Audio source nếu Audio Clip là một file âm thanh 3D.
* Pan Lever: cài đặt, làm thế nào để máy 3D có được hiệu ứng từ Audio source.
* Spread: cài đặt góc ảnh hưởng tới âm thanh 3D Stereo nếu Audio Clip là một âm thanh 3D.
* Doppler Lever: xác định bao nhiêu hiệu ứng âm thanh Dopper sẽ được áp dụng cho Audio Source (Nếu cài đặt là 0 thì sẽ không có hiệu ứng nào được áp dụng).
* Min Distance: với Min Distance, âm thanh sẽ đạt mức to nhất có thể. Ngoài Min Distance, nó sẽ bắt đầu suy yếu đi. Tăng MinDistance của âm thanh để được âm thanh lớn hơn trong thế giới 3D và giảm MinDistance để cho âm thanh nhỏ hơn.
* Max Distance: khoảng cách mà âm thanh dừng suy giảm. Sau khoảng cách này nó sẽ tạm ngừng không cho MinDistance tăng nữa. Tức là đây là giá trị Max mà MinDistance sẽ đạt tới.
* Rolloff Mode: làm thế nào nhanh chóng mất dần âm thanh. Giá trị cao hơn, gần gũi hơn Listener có được trước khi nghe âm thanh.(Điều này được xác định bởi một Graph).
* Logarithmic Rolloff: âm thanh là lớn khi bạn ở gần nguồn âm thanh. Nhưng khi chúng ta nhận được từ các đối tượng nó sẽ giảm đi nhanh chóng một cách đáng kể.
* Linear Rolloff: càng xa nguồn âm thanh, bạn nghe thấy càng ít.
* 2D Sound Setting: cài đặt áp dụng cho nguồn âm thanh, nếu Audio Clip là âm thanh 2D.
* Pan 2D: cài đặt hiệu ứng trên nguồn âm thanh.

***2.4.4.10. Script***

Scripting là một thành phần thiết yếu trong tất cả các ứng dụng bạn tạo trong Unity. Hầu hết các ứng dụng cần tập lệnh để phản hồi thông tin đầu vào từ người chơi và sắp xếp các sự kiện trong trò chơi xảy ra khi họ cần. Ngoài ra, các tập lệnh có thể được sử dụng để tạo hiệu ứng đồ họa, kiểm soát hành vi vật lý của các đối tượng hoặc thậm chí triển khai hệ thống AI tùy chỉnh cho các nhân vật trong trò chơi. Hiện này Script Unity chỉ sử dụng ngôn ngữ C# với phần mềm hỗ trợ đi kèm sẵn là Microsoft Virual Studio.

## **2.5. Phần mềm Microsoft Visual Studio**

***2.5.1. Tổng quan phần mềm Microsoft Visual Studio***

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

Microsoft cung cấp phiên bản "Express" (đối với phiên bản Visual Studio 2013 trở về trước) và "Community" (đối với bản Visual Studio 2015 trở về sau) là phiên bản miễn phí của Visual Studio.

***2.5.2. Lịch sử phát triển***

Vào năm 1997, Visual Studio ra mắt, tên mã là [Boston](https://vi.wikipedia.org/wiki/Boston) (tên thành phố, Microsoft có thói quen đặt tên mã của Visual Studio theo tên địa danh). Visual Studio 97 là sự kết hợp của nhiều công cụ lập trình với nhau. Visual Studio 97 xuất hiện trong hai phiên bản: Visual Studio Professional và Visual Studio Enterprise, phiên bản chuyên nghiệp chứa trên ba đĩa [CD](https://vi.wikipedia.org/wiki/CD) và phiên bản doanh nghiệp chứa trên bốn đĩa CD. Nó bao gồm Visual J++ 1.1 cho các lập trình viên Java, cũng như đi kèm với Visual InterDev để tạo ra các trang web được tạo tự động bằng [Active Server Pages](https://vi.wikipedia.org/wiki/Active_Server_Pages). Đi kèm các CD cài đặt là một CD chứa thư viện [Mạng Microsoft Developer Network](https://vi.wikipedia.org/wiki/Microsoft_Developer_Network).

Visual Studio 97 là nỗ lực đầu tiên của Microsoft trong việc sử dụng một môi trường phát triển cho nhiều ngôn ngữ lập trình khác nhau. Visual J++, InterDev, và Thư viện MSDN đã sử dụng cùng một "môi trường", gọi là Developer Studio. Visual Studio cũng được bán dưới dạng các phiên bản con với các IDE riêng biệt được sử dụng cho [Visual C++](https://vi.wikipedia.org/wiki/Visual_C%2B%2B), [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic) và [Visual FoxPro](https://vi.wikipedia.org/wiki/Visual_FoxPro).

Phiên bản kế tiếp có tên chính thức là Visual Studio 6.0, tên mã [Aspen](https://vi.wikipedia.org/wiki/Aspen,_Colorado) (khu nghỉ mát trượt tuyết ở [Colorado](https://vi.wikipedia.org/wiki/Colorado)), được phát hành vào tháng 6 năm 1998 và là phiên bản cuối cùng chạy trên nền Windows 9x.[[43]](https://vi.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-43) Mỗi phiên bản của mỗi thành phần cũng được nâng lên thành v6.0, bao gồm Visual J++ trước đó là v1.1 và Visual InterDev tại lần phát hành đầu tiên. Phiên bản v6 của Microsoft là môi trường cốt lõi cho bốn phiên bản tiếp theo nhằm cung cấp cho các lập trình một nền tảng lập trình giống nhau. Điều này đã khiến Microsoft tập trung sự phát triển vào nền tảng độc lập [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework).

Visual Studio 6.0 là phiên bản cuối cùng bao gồm Visual J++ mà Microsoft đã gỡ bỏ (theo thỏa thuận của Microsoft với [Sun Microsystems](https://vi.wikipedia.org/wiki/Sun_Microsystems), yêu cầu [Microsoft Internet Explorer](https://vi.wikipedia.org/wiki/Internet_Explorer) không hỗ trợ cho máy ảo Java).

Vào tháng 2 năm 2002, Microsoft đã phát hành Visual Studio.NET, có tên mã là Rainier (Mount Rainier của [Washington](https://vi.wikipedia.org/wiki/Washington)). Phiên bản beta đã được phát hành thông qua [MSDN](https://vi.wikipedia.org/wiki/Microsoft_Developer_Network) vào năm 2001. Thay đổi lớn nhất là việc giới thiệu một môi trường phát triển mã được quản lý bằng.NET Framework. Các chương trình phát triển sử dụng.NET không được biên dịch thành ngôn ngữ máy (như C ++ chẳng hạn) mà thay vào đó là một định dạng gọi là Microsoft Intermediate Language (MSIL) hoặc [Ngôn ngữ trung gian dùng chung](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_trung_gian_d%C3%B9ng_chung) (CIL). Khi một ứng dụng CIL được thực thi, nó được biên dịch vào ngôn ngữ máy phù hợp với nền tảng đang chạy, do đó mã nguồn có thể sử dụng cho nhiều nền tảng khác nhau. Các chương trình biên soạn vào CIL chỉ có thể được thực thi trên các nền tảng có [cơ sở ngôn ngữ dùng chung](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_ng%C3%B4n_ng%E1%BB%AF_d%C3%B9ng_chung). Có thể chạy các chương trình CIL trong Linux hoặc Mac OS X sử dụng các ứng dụng không phải Microsoft.NET như Mono và DotGNU. Đây là phiên bản đầu tiên của Visual Studio yêu cầu nền tảng Windows NT. Trình cài đặt thực thi yêu cầu này.

Visual Studio.NET 2002 được vận chuyển theo bốn phiên bản: Học thuật (Academic), Chuyên gia (Professional), Nhà phát triển Doanh nghiệp (Enterprise Developer) và Kiến trúc sư Doanh nghiệp (Enterprise Architect). Microsoft giới thiệu C# ([C-sharp](https://vi.wikipedia.org/wiki/C_th%C4%83ng)), một ngôn ngữ lập trình mới, hướng vào.NET. Họ cũng giới thiệu sự kế thừa cho Visual J++ là Visual J#. Các chương trình Visual J# sử dụng cú pháp ngôn ngữ của Java. Tuy nhiên, không giống như các chương trình Visual J++, chương trình Visual J# chỉ có thể chạy trên.NET Framework chứ không phải Java Virtual Machine.

Vào tháng 4/2003, Visual Studio.NET 2003 ra mắt, có tên mã là [Everett](https://vi.wikipedia.org/wiki/Everett,_Washington). Ở bản này, [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework) được nâng cấp lên phiên bản 1.1. Đây cũng là phiên bản Visual Studio đầu tiên hỗ trợ phát triển các chương trình cho các thiết bị di động, sử dụng ASP.NET hoặc.NET Compact Framework. Các tiêu chuẩn tuân thủ của Visual C++ được cải thiện. Visual C++ Toolkit 2003 được bán kèm với Visual Studio.NET 2003 mà không có IDE. Tính đến năm 2010, bộ này không còn tồn tại nữa và thay thế nó là Express Editions. Số phiên bản của Visual Studio.NET 2003 là phiên bản 7.1, số phiên bản định dạng tệp là 8.0.

Visual Studio.NET 2003 cũng được bán với bốn phiên bản: Học thuật, Chuyên gia, Nhà phát triển Doanh nghiệp và Kiến trúc sư Doanh nghiệp. Phiên bản Visual Studio.NET 2003 Enterprise Architect bao gồm việc triển khai các công nghệ mô hình của Microsoft Visio 2002, bao gồm các công cụ để tạo ra các mô tả trực quan về giải pháp mô hình hoá cơ sở dữ liệu logic.

Visual Studio 2005 có tên mã Whidbey (đảo Whidbey ở Puget Sound), được phát hành trực tuyến từ tháng 10 năm 2005. Kể từ phiên bản này Microsoft loại bỏ tên gọi ".NET" ở tên sản phẩm (cũng như mọi sản phẩm khác có.NET), nhưng Visual Studio vẫn chủ yếu nhắm mục tiêu vào.NET Framework (lúc này đã được nâng cấp lên phiên bản 2.0).

Visual Studio 2008 và Visual Studio Team System 2008 có tên mã Orcas (đảo Orcas, cũng là một hòn đảo ở Puget Sound), bắt đầu phát hành cho các thuê bao MSDN vào ngày 19 tháng 11 năm 2007 cùng với [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework) 3.5. Mã nguồn cho Visual Studio 2008 IDE có sẵn dưới giấy phép nguồn chia sẻ cho một số đối tác của Microsoft. Microsoft phát hành gói dịch vụ 1 cho Visual Studio 2008 vào ngày 11 tháng 8 năm 2008. Số phiên bản nội bộ của Visual Studio 2008 là 9.0, số phiên bản định dạng tệp là 10.0. Visual Studio 2008 là phiên bản cuối cùng hỗ trợ cho các ứng dụng C++ chạy trên Windows 2000.

Ban đầu được gọi là Visual Studio "14", Community Technology Preview (CTP) được phát hành lần đầu vào ngày 3 tháng 6 năm 2014 và bản RC đã được phát hành vào ngày 29 tháng 4 năm 2015. Sau đó, vào ngày 12 tháng 11 năm 2014, Visual Studio 2015 chính thức được công bố là cái tên cuối cùng của phiên bản này. Nhìn chung thì Visual Studio 2015 và các bản cập nhật của nó không đi kèm với tính năng mới, mà chỉ cập nhật các thành phần.

Phiên bản xem trước đầu tiên được phát hành vào ngày 30 tháng 3 năm 2016 với cái tên là Visual Studio "15". Ngày 14 tháng 11 năm 2016, Microsoft đăng một bài blog tiết lộ tên sản phẩm Visual Studio 2017 cùng với các tính năng sắp tới. Sau đó, vào ngày 16 tháng 11 năm 2016, "Visual Studio 2017" đã được công bố là tên cuối cùng của sản phẩm và Visual Studio 2017 RC. Cuối cùng, vào ngày 7 tháng 3 năm 2017, Visual Studio 2017 RTM chính thức ra mắt công chúng. Visual Studio 2017 tổng hợp các cập nhật của các bản preview, cũng như bổ sung.Net Core và ASP.NET Core. Bản này cũng cung cấp các tính năng mới như hỗ trợ EditorConfig (một khuôn khổ cho việc thực thi mã hóa), hỗ trợ NGen, công cụ.NET Core và Docker và Xamarin 4.3. Ở phiên bản này, XAML Editor và IntelliSense được cải tiến, unit test trực tiếp, gỡ lỗi nâng cao, và cải thiện tổng thể hiệu suất của IDE.

Visual Studio 2019 như một cú lột xác hoàn hảo với sự cải thiện toàn diện cả về giao diện thao tác và hiệu suất. Giao diện Start cũng trở nên đơn giản và dễ dàng hơn cho các "coder" để thao tác nhanh hơn (nhất là khi muốn clone một repos từ Git hoặc mở một project/ folder có sẵn), màn hình chọn template code cũng được cải thiện, vùng soạn thảo code được mở rộng không gian hơn, trải nghiệm tìm kiếm mới, trình debug thông minh hơn. Hơn thế nữa, tất cả những tính năng trên đều có thể hoạt động với cả những project đã tồn tại hoặc một project mới - từ các ứng dụng C++ đa nền tảng, cho tới ứng dụng .NET cho Android và iOS sử dụng Xamarin hay những ứng dụng sử dụng nền tảng đám mây với dịch vụ Azure.

***2.5.3. Tính năng của Microsoft Visual Studio***

***2.5.3.1 Biên tập mã***

Giống như bất kỳ IDE khác, nó bao gồm một trình soạn thảo mã hỗ trợ tô sáng cú pháp và hoàn thiện mã bằng cách sử dụng IntelliSense không chỉ cho các biến, hàm và các phương pháp mà còn các cấu trúc ngôn ngữ như vòng điều khiển hoặc truy vấn. IntelliSense được hỗ trợ kèm theo cho các ngôn ngữ như XML, Cascading Style Sheets và JavaScript khi phát triển các trang web và các ứng dụng web. Các đề xuất tự động hoàn chỉnh được xuất hiện trong một hộp danh sách phủ lên trên đỉnh của trình biên tập mã. Trong Visual Studio 2008 trở đi, nó có thể được tạm thời bán trong suốt để xem mã che khuất bởi nó. Các trình biên tập mã được sử dụng cho tất cả các ngôn ngữ được hỗ trợ.

Visual Studio cũng hỗ trợ cài đặt dấu trang trong mã để điều hướng nhanh chóng. Hỗ trợ điều hướng khác bao gồm thu hẹp các khối mã lệnh và tìm kiếm gia tăng, ngoài việc tìm kiếm văn bản thông thường và tìm kiếm Biểu thức chính quy. Các trình biên tập mã cũng bao gồm một bìa kẹp đa mục và một danh sách công việc. Các trình biên tập mã hỗ trợ lưu lại các đoạn mã được lặp đi lặp lại nhằm để chèn vào mã nguồn sử dụng về sau. Một công cụ quản lý cho đoạn mã được xây dựng là tốt. Những công cụ này nổi lên như các cửa sổ trôi nổi có thể được thiết lập để tự động ẩn khi không sử dụng hoặc neo đậu đến các cạnh của màn hình. Các trình biên tập mã Visual Studio cũng hỗ trợ cải tiến mã nguồn bao gồm tham số sắp xếp lại, biến và phương pháp đổi tên, khai thác và đóng gói giao diện các lớp thành viên bên trong những trạng thái giữa những thứ khác.

Visual Studio có tính năng biên dịch nền (còn gọi là biên dịch gia tăng): như mã đang được viết, Visual Studio biên dịch nó trong nền để cung cấp thông tin phản hồi về cú pháp và biên dịch lỗi, được đánh dấu bằng một gạch dưới gợn sóng màu đỏ. Biên dịch nền không tạo ra mã thực thi, vì nó đòi hỏi một trình biên dịch khác hơn là để sử dụng tạo ra mã thực thi. Biên dịch nền ban đầu được giới thiệu với Microsoft Visual Basic nhưng bây giờ đã được mở rộng cho tất cả các ngôn ngữ.

***2.5.3.2. Trình gỡ lỗi***

Visual Studio có một trình gỡ lỗi hoạt động vừa là một trình gỡ lỗi cấp mã nguồn và là một trình gỡ lỗi cấp máy. Nó hoạt động với cả hai mã quản lý cũng như ngôn ngữ máy và có thể được sử dụng để gỡ lỗi các ứng dụng được viết bằng các ngôn ngữ được hỗ trợ bởi Visual Studio. Ngoài ra, nó cũng có thể đính kèm theo quy trình hoạt động và theo dõi và gỡ lỗi những quy trình. Nếu mã nguồn cho quá trình hoạt động có sẵn, nó sẽ hiển thị các mã như nó đang được chạy. Nếu mã nguồn không có sẵn, nó có thể hiển thị các tháo gỡ. Các Visual Studio debugger cũng có thể tạo bãi bộ nhớ cũng như tải chúng sau để gỡ lỗi. Các chương trình đa luồng cao cấp cũng được hỗ trợ. Trình gỡ lỗi có thể được cấu hình sẽ được đưa ra khi một ứng dụng đang chạy ngoài Visual Studio bị treo môi trường.

Trình gỡ lỗi cho phép thiết lập các breakpoint (mà cho phép thực thi được tạm thời dừng lại tại một vị trí nhất định) và watch (trong đó giám sát các giá trị của biến là việc thực hiện tiến bộ). Breakpoint có thể có điều kiện, nghĩa là chúng được kích hoạt khi điều kiện được đáp ứng. Mã có thể được biểu diễn, tức là chạy một dòng (của mã nguồn) tại một thời điểm. Nó có hoặc là bước sang các chức năng để gỡ lỗi bên trong nó, hoặc là nhảy qua nó, tức là, việc thực hiện các chức năng không có sẵn để kiểm tra thủ công. Trình gỡ lỗi hỗ trợ Edit and Continue, nghĩa là, nó cho phép mã được chỉnh sửa khi nó đang được sửa lỗi (chỉ có 32 bit, không được hỗ trợ trong 64 bit). Khi gỡ lỗi, nếu con trỏ chuột di chuyển lên bất kỳ biến, giá trị hiện tại của nó được hiển thị trong phần chú giải ("chú thích dữ liệu"), nơi mà nó cũng có thể được thay đổi nếu muốn. Trong quá trình viết mã, các trình gỡ lỗi của Visual Studio cho phép một số chức năng được gọi ra bằng tay từ cửa sổ công cụ Immediate. Các thông số cho phương thức được cung cấp tại các cửa sổ Immediate.

***2.5.3.3. Thiết kế***

Windows Forms Designer: được sử dụng để xây dựng GUI sử dụng Windows Forms; bố trí có thể được xây dựng bằng các nút điều khiển bên trong hoặc khóa chúng vào bên cạnh mẫu. Điều khiển trình bày dữ liệu (như hộp văn bản, hộp danh sách, vv) có thể được liên kết với các nguồn dữ liệu như cơ sở dữ liệu hoặc truy vấn. Các điều khiển dữ liệu ràng buộc có thể được tạo ra bằng cách rê các mục từ cửa sổ nguồn dữ liệu lên bề mặt thiết kế. Các giao diện người dùng được liên kết với mã sử dụng một mô hình lập trình hướng sự kiện. Nhà thiết kế tạo ra bằng C# hay VB.NET cho ứng dụng.

WPF Designer: có tên mã là Cider, được giới thiệu trong Visual Studio 2008. Giống như Windows Forms Designer, hỗ trợ kéo và thả ẩn dụ. Sử dụng tương tác người-máy nhắm mục tiêu theo Windows Presentation Foundation. Nó hỗ trợ các chức năng WPF bao gồm kết nối dữ liệu và tự động hóa bố trí quản lý. Nó tạo ra mã XAML cho giao diện người dùng. Các tập tin XAML được tạo ra là tương thích với Microsoft Expression Design, sản phẩm thiết kế theo định hướng.

Web designer/development: Visual Studio cũng bao gồm một trình soạn thảo và thiết kế trang web cho phép các trang web được thiết kế bằng cách kéo và thả các đối tượng. Nó được sử dụng để phát triển các ứng dụng ASP.NET và hỗ trợ HTML, CSS và JavaScript. Nó sử dụng mô hình code-behind để liên kết với mã ASP.NET. Từ Visual Studio 2008 trở đi, công cụ bố trí được sử dụng bởi các nhà thiết kế web được chia sẻ với Microsoft Expression Web. Ngoài ra ASP.NET MVC Framework hỗ trợ cho công nghệ MVC là tải xuống riêng biệt và dự án ASP.NET Dynamic Data có sẵn từ Microsoft.

Class designer: các lớp thiết kế được dùng để biên soạn và chỉnh sửa các lớp (bao gồm cả các thành viên và truy cập của chúng) sử dụng mô hình UML. Các lớp thiết kế có thể tạo ra mã phác thảo C# và VB.NET cho các lớp và cá phương thức. Nó cũng có thể tạo ra sơ đồ lớp từ các lớp viết tay.

Data designer: thiết kế dữ liệu có thể được sử dụng để chỉnh sửa đồ họa giản đồ cơ sở dữ liệu bao gồm các bảng, khóa chính, khóa ngoại và các rằng buộc. Nó cũng có thể được sử dụng để thiết kế các truy vấn từ các giao diện đồ họa.

Mapping designer: từ Visual Studio 2008 trở đi, thiết kế ánh xạ được dùng bởi Language Integrated Query để thiết kế các ánh xạ giữa các giản đồ cơ sở dữ liệu và các lớp để đóng gói dữ liệu. Các giải pháp mới từ cách tiếp cận ORM, ADO.NET Entity Framework sẽ thay thế và cải thiện các công nghệ cũ.

***2.5.3.4. Các công cụ khác***

Open Tabs Browser: được sử dụng để liệt kê tất cả thẻ đang mở và chuyển đổi giữa chúng. Được viện dẫn bằng cách sử dụng CTRL+TAB.

Properties Editor: được sử dụng để chỉnh sửa các thuộc tính trong một cửa sổ giao diện bên trong Visual Studio. Nó liệt kê tất cả các thuộc tính có sẵn (gồm chỉ đọc và những thuộc tính có thể được thiết lập) cho tất cả các đối tượng bao gồm các lớp, biểu mẫu, trang web và các hạng mục khác.

Object Browser: là một không gian tên và trình duyệt lớp thư viện cho Microsoft NET. Nó có thể được sử dụng để duyệt các không gian tên (được sắp xếp theo thứ bậc) trong Assembly (CLI). Các hệ thống phân cấp có thể hoặc không có thể phản ánh các tổ chức trong hệ thống tập tin.

Solution Explorer: theo cách nói trong Visual Studio, một giải pháp là một tập hợp các tập tin mã và các nguồn khác được sử dụng để xây dựng một ứng dụng. Các tập tin trong một giải pháp được sắp xếp theo thứ bậc, mà có thể có hoặc không thể phản ánh các tổ chức trong hệ thống tập tin. Solution Explorer được sử dụng để quản lý và duyệt các tập tin trong một giải pháp.

Team Explorer: được sử dụng để tích hợp các khả năng của Team Foundation Server, Revision Control System và là cơ sở cho môi trường CodePlex đối với dự án mã nguồn mở. Ngoài việc kiểm soát nguồn nó cung cấp khả năng xem và quản lý các công việc riêng lẻ (bao gồm cả lỗi, nhiệm vụ và các tài liệu khác) và để duyệt thống kê TFS. Nó được bao gồm như là một phần của một cài đặt TFS và cũng có sẵn để tải xuống cho Visual Studio. Team Explorer cũng có sẵn như là một môi trường độc lập duy nhất để truy cập các dịch vụ TFS.

Data Explorer: được sử dụng để quản lý cơ sở dữ liệu trên Microsoft SQL Server. Nó cho phép tạo ra và sửa đổi các bảng cơ sở dữ liệu (hoặc bằng cách ban hành các lệnh T-SQL hoặc bằng cách sử dụng các thiết kế dữ liệu). Nó cũng có thể được sử dụng để tạo các truy vấn và các thủ tục lưu trữ trong T-SQL hoặc trong Managed code thông qua SQL CLR. Có sẵn gỡ lỗi và hỗ trợ IntelliSense.

Server Explorer: công cụ được sử dụng để quản lý các kết nối cơ sở dữ liệu trên một máy tính truy cập được. Nó cũng được sử dụng để duyệt chạy Windows Services, quầy thực hiện, Windows Event Log và hàng đợi tin nhắn và sử dụng chúng như một nguồn dữ liệu.

Dotfuscator Software Services Community Edition: Visual Studio bao gồm một phiên bản light của sản phẩm PreEmptive Solutions' Dotfuscator cho mã gây rối và giảm kích thước ứng dụng. Khởi đầu với Visual Studio 2010, phiên bản này của Dotfuscator sẽ bao gồm khả năng Runtime Intelligence cho phép tác giả thu thập cách sử dụng của người dùng cuối, hiệu suất, tính ổn định và các thông tin từ các ứng dụng của họ chạy trong sản xuất.

Text Generation Framework: Visual Studio bao gồm một khung tạo văn bản đầy đủ được gọi là Text Template Transformation Toolkit T4 cho phép Visual Studio tạo ra tập tin văn bản từ các mẫu hoặc trong IDE hoặc thông qua mã.

ASP.NET Web Site Administration Tool: công cụ quản trị trang web ASP.NET cho phép cấu hình các trang web ASP.NET.

Visual Studio Tools for Office: Công cụ Visual Studio cho Office là một SDK và một add-in cho Visual Studio bao gồm các công cụ để phát triển cho các bộ Microsoft Office. Trước đây (với Visual Studio.NET 2003 và Visual Studio 2005) đó là một SKU riêng biệt mà chỉ hỗ trợ Visual C# Visual Basic.NET hoặc đã được đưa vào Team Suite. Với Visual Studio 2008, nó không còn là một SKU riêng biệt nhưng lại kèm trong các phiên bản chuyên nghiệp và cao hơn. Một thời gian chạy riêng biệt được yêu cầu khi triển khai các giải pháp VSTO.

***2.5.3.5. Khả năng mở rộng***

Visual Studio cho phép các nhà phát triển viết các phần mở rộng cho Visual Studio để mở rộng tính năng của nó. Những phần mở rộng "cắm vào" Visual Studio và mở rộng tính năng của nó. Các phần mở rộng đến ở dạng macro, add-in và các gói. Các macro đại diện cho các nhiệm vụ lặp đi lặp lại và hành động mà các nhà phát triển có thể ghi lại theo chương trình để tiết kiệm, các cài đặt. Các chế độ biệt lập của vỏ tạo ra một AppID mới, nơi các gói được cài đặt. Những thứ này được bắt đầu với một thực thi khác nhau. Nó nhằm mục đích cho sự phát triển của môi trường phát triển tùy chỉnh, hoặc cho một ngôn ngữ cụ thể hoặc một kịch bản cụ thể. Các chế độ tích hợp cài đặt các gói vào AppID của các phiên bản Professional / Standard / Team System, do đó các công cụ tích hợp vào các phiên bản. Visual Studio Shell là miễn phí tải về.

Sau khi phát hành Visual Studio 2008, Microsoft đã tạo ra Visual Studio Gallery. Nó phục vụ như vị trí trung tâm cho đăng tải thông tin về phần mở rộng cho Visual Studio. Phát triển cộng đồng cũng như phát triển thương mại có thể tải lên thông tin về các phần mở rộng của họ đến Visual Studio.NET 2002 thông qua Visual Studio 2010. Người sử dụng trang web có thể đánh giá và xem lại các phần mở rộng để giúp đánh giá chất lượng các phần mở rộng được đăng. RSS feed thông báo cho người dùng trên bản cập nhật tới trang web và các tính năng gắn thẻ cũng được lên kế hoạch.

**2.6. Vuforia Engine**

***2.6.1. Tổng quan Vuforia Engine***

Vuforia là một bộ công cụ phát triển phần mềm thực tế tăng cường (SDK) dành cho thiết bị di động cho phép tạo các ứng dụng thực tế tăng cường. Nó sử dụng công nghệ thị giác máy tính để nhận dạng và theo dõi hình ảnh phẳng và vật thể 3D trong thời gian thực. Khả năng đăng ký hình ảnh này cho phép các nhà phát triển định vị và định hướng các đối tượng ảo, chẳng hạn như mô hình 3D và các phương tiện khác, liên quan đến các đối tượng trong thế giới thực khi chúng được xem qua camera của thiết bị di động. Đối tượng ảo sau đó theo dõi vị trí và hướng của hình ảnh trong thời gian thực để góc nhìn của người xem về đối tượng tương ứng với phối cảnh trên mục tiêu. Do đó, có vẻ như đối tượng ảo là một phần của cảnh trong thế giới thực.

Vuforia SDK hỗ trợ nhiều loại mục tiêu 2D và 3D bao gồm Mục tiêu hình ảnh ‘không điểm đánh dấu’, Mục tiêu mô hình 3D và một dạng Điểm đánh dấu Fiducial có thể địa chỉ, được gọi là VuMark. Các tính năng bổ sung của SDK bao gồm bản địa hóa thiết bị 6 bậc tự do trong không gian, Phát hiện tắc cục bộ bằng cách sử dụng 'Các nút ảo', lựa chọn mục tiêu hình ảnh thời gian chạy và khả năng tạo và định cấu hình lại các tập hợp mục tiêu theo lập trình trong thời gian chạy.

Vuforia cung cấp giao diện lập trình ứng dụng (API) bằng ngôn ngữ C ++, Java, C#, Objective - C ++ và .NET thông qua phần mở rộng cho công cụ trò chơi Unity. Bằng cách này, SDK hỗ trợ cả phát triển gốc cho IOS, Android và UWP trong khi nó cũng cho phép phát triển các ứng dụng AR trong Unity có thể dễ dàng di chuyển sang cả hai nền tảng. Vuforia đã được mua lại bởi PTC Inc. vào tháng 11 năm 2015.

***2.6.2. Lịch sử phát triển***

***2.6.3. Tính năng của Vuforia Engine***

***2.6.3.1. Image Targets***

Đính kèm nội dung lên hình ảnh phẳng, chẳng hạn như giấy in và bao bì sản phẩm:

* Mục tiêu hình ảnh tức thì - Tạo Mục tiêu Hình ảnh trong thời gian chạy hoặc từ cơ sở dữ liệu.
* Các phương pháp hay nhất - Tìm hiểu cách tạo bố cục hình ảnh lý tưởng cho mục tiêu hình ảnh.
* Cơ sở dữ liệu thiết bị - Tìm hiểu cách tạo Cơ sở dữ liệu thiết bị trong Trình quản lý mục tiêu Vuforia.

***2.6.3.2. Object Tracking Methods***

Trong bài viết này, các phương pháp Theo dõi Đối tượng Vuforia; Mục tiêu Mô hình, Mục tiêu Đối tượng và Mục tiêu Mô hình được tạo từ quét 3D được giới thiệu và so sánh. Mỗi tính năng có lợi thế trong các trường hợp sử dụng nhất định và mỗi tính năng được tạo bằng các công cụ khác nhau. Chọn phương pháp phù hợp nhất cho trường hợp sử dụng của bạn.

Ba cách triển khai Theo dõi đối tượng mang những ưu điểm và hạn chế riêng và chúng được thiết kế để sử dụng trong các trường hợp sử dụng cụ thể.

* Model Targets: được tạo ra bằng cách quét một đối tượng vật lý. Chúng là một lựa chọn tốt cho đồ chơi và các sản phẩm khác có bề mặt chi tiết phong phú và hình dạng nhất quán.
* Object Targets: cho phép bạn nhận dạng các đối tượng theo hình dạng bằng các mô hình 3D có sẵn từ trước. Đặt nội dung AR trên nhiều mặt hàng như thiết bị công nghiệp, xe cộ, đồ chơi và đồ gia dụng.
* Model Targets from a 3D scan: là Model Targets được tạo từ quét 3D và đặc biệt hữu ích khi không có sẵn mô hình CAD.

***2.6.3.5. Cloud Recognition***

Cloud Recognition (nhận dạng điện toán đám mây) là một giải pháp Nhận dạng Hình ảnh cấp doanh nghiệp cho phép các nhà phát triển lưu trữ và quản lý Mục tiêu Hình ảnh trực tuyến. Nhận dạng đám mây khả dụng với các giấy phép Phát triển, Cơ bản + Đám mây và Pro. Mức sử dụng được xác định bằng tổng số lần nhận dạng hình ảnh hay còn gọi là "số lần lặp lại" mỗi tháng mà ứng dụng của bạn thực hiện và được tính khi mục tiêu phù hợp.

***2.6.3.6. VuMarks***

VuMarks là mã vạch thế hệ tiếp theo. Chúng cho phép tự do thiết kế tùy chỉnh và có ý thức về thương hiệu đồng thời mã hóa dữ liệu và hoạt động như một mục tiêu AR có thể theo dõi. Các thiết kế VuMark hoàn toàn có thể tùy chỉnh, vì vậy bạn có thể có một VuMark duy nhất cho mọi đối tượng độc đáo.

***2.6.3.7. Cylinder Targets***

Cylinder Targets cho phép bạn phát hiện và theo dõi các hình ảnh được bao bọc thành hình trụ và hình nón. Vuforia Engine có thể theo dõi các mặt và mặt phẳng trên và dưới của Mục tiêu xi lanh. Hình dạng hình trụ được sử dụng rộng rãi cho hàng tiêu dùng và thường có nhãn độc đáo, lý tưởng để tạo ra trải nghiệm thực tế tăng cường đi kèm.

***2.6.3.8. Multi Target***

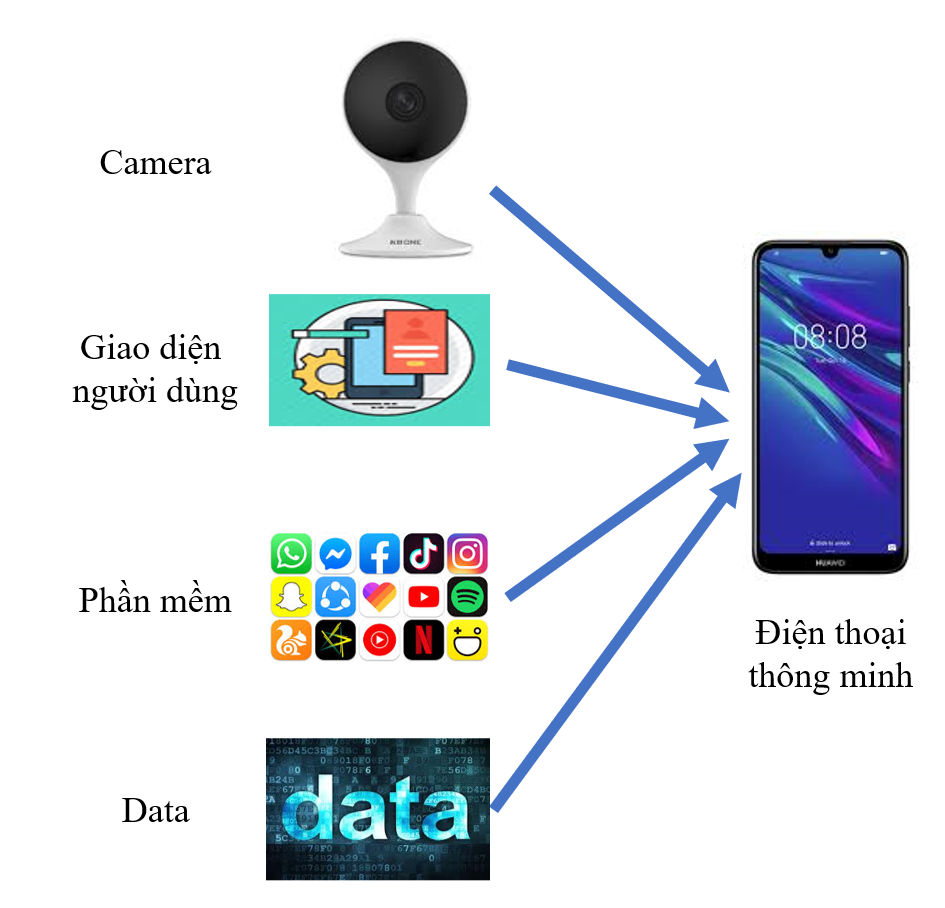
Multi Target là một tập hợp nhiều Image Target được kết hợp thành một sắp xếp hình học xác định, chẳng hạn như các hộp. Điều này cho phép theo dõi và phát hiện từ mọi phía và có thể phục vụ nhiều trường hợp sử dụng, chẳng hạn như tiếp thị, đóng gói và trong các ngữ cảnh hướng dẫn. Bắt đầu bằng cách tạo Multi Target của bạn trong Vuforia Target Manager và tải lên hình ảnh của bạn phù hợp với kích thước của Multi Target của bạn.

# **CHƯƠNG 3. XÂY DỰNG HỆ THỐNG THỰC TẾ ẢO TĂNG CƯỜNG HỖ TRỢ HIỂN THỊ MÔ HÌNH TRÊN DI ĐỘNG TRONG HỌC TẬP**

**3.1. Cấu trúc tổng quan của một hệ thống thực tế ảo tăng cường trên điện thoại**

Một hệ thống thực tế ảo tăng cường trên điện thoại thông minh bao gồm:

* Giao diện người dùng: Giúp người dùng tương tác và giao tiếp với hệ thống.
* Camera: Thu thập thông tin từ cảnh quan bên ngoài.
* Data: là nơi lưu trữ dữ liệu như mô hình 2D, mô hình 3D, …
* Phần mềm: giúp xử lý thông tin đưa vào từ camera và thực hiện mong muốn của người dùng.
* Điện thoại thông minh: Dùng để kết nối mọi thứ lại với nhau.



***Hình 3.1.*** Tổng quan cấu trúc hệ thống thực tế ảo tăng cường trên điện thoại

**3.2. Quy trình xây dựng hệ thống thực tế ảo tăng cường trên điện thoại**

Để giúp cho việc xây dựng hệ thống được xây dựng một cách dễ dàng và nhanh chóng, chúng tôi đã sử dụng công nghệ là Vuforia và phần mềm gồm có Solidworks, Blender, Unity, Visual Studio và PowerPoint.

Công nghệ Vuforia giúp kết nối camera của điện thoại với phần mềm. Lấy dữ liệu từ camera đưa vào và xử lý. Hiện này công nghệ này hỗ trợ Scan 3D trên IOS, Tracking Image, Tracking Plant, …. Chúng tôi đã sử dụng nó vào dự án này với chức năng là kết nối camera của điện thoại với phần mềm và Tracking Image.

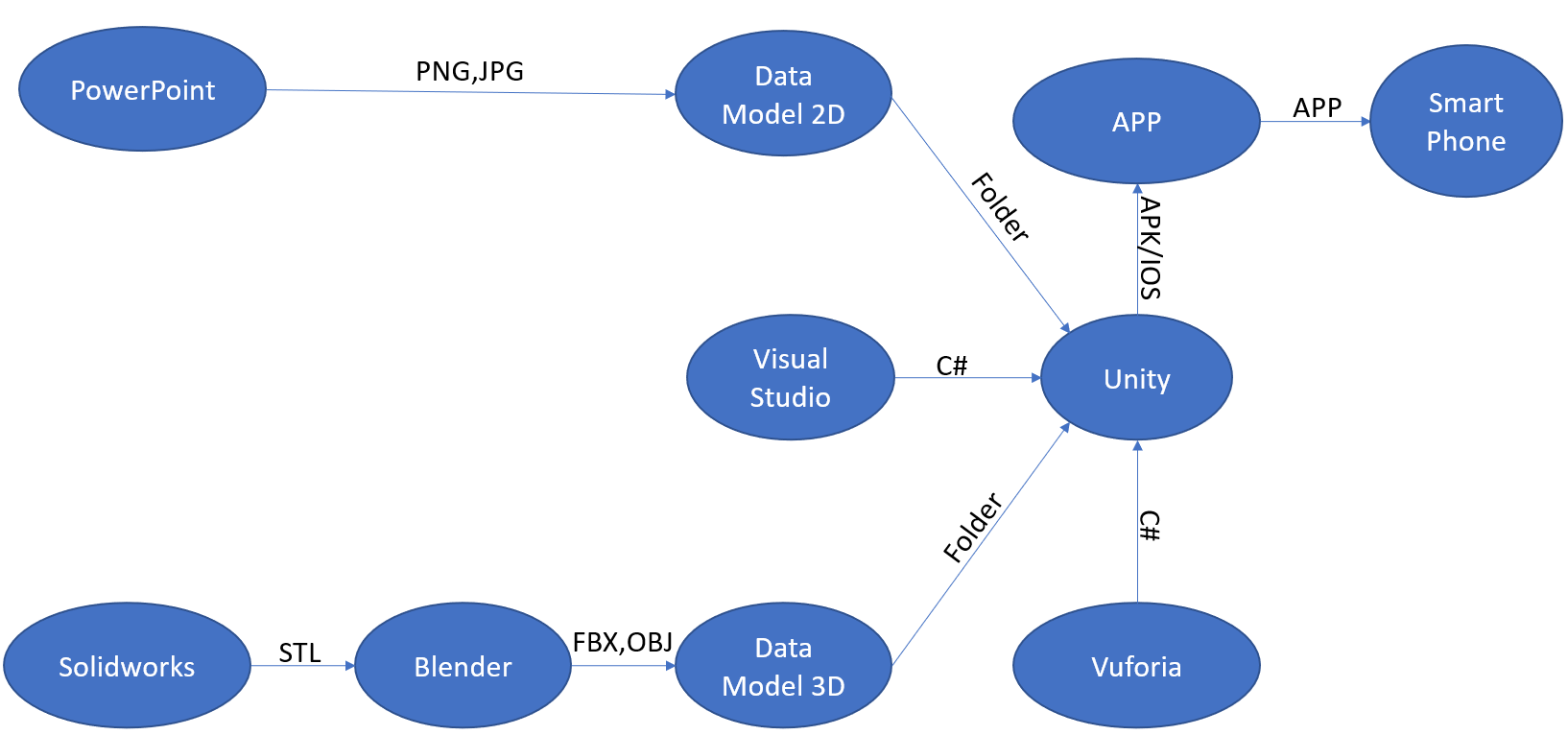
Solidworks là phần mềm hỗ trợ thiết kế, thi công và mô phỏng cơ khí. Phần mềm này khá là dễ sử dụng và tiện lợi. Chúng tôi đã sử dụng nó để tạo ra các chi tiết nhỏ lẻ vào ghép lại tạo thành chi tiết lớn. Khi chi tiết lớn được hoàn thành thì để đưa chúng vào Unity để tạo cơ sở dữ liệu 3D cho phần mềm thì có 2 giải pháp. Giải pháp thứ nhất là dùng công cụ hỗ trợ nhưng sẽ mất phí như Tlink, … Giải pháp thứ 2 là chúng tôi sẽ dùng phần mềm khác để chuyển định dạng của mô hình sang định dạng mà Unity đọc được. Chúng tôi đã chọn giải pháp thứ 2 vì nó tiết kiệm chi phí hơn.

Blender là phần mềm thiết kế 3D cho các nhân vật, cảnh quan, … trong các dự án game và phim. Chúng tôi chọn Blender vì nó miễn phí, dung lượng thấp (khoảng 300MB), dễ dàng sử dụng và cài đặt.

Unity là phần mềm chúng tôi chọn để kết nối mọi thứ lại với nhau và hỗ trợ xuất file thành nhiều dạng khác nhau như IOS, Android, … Để kết nối và xuất file thành IOS hay Android thì có rất nhiều phần mềm hỗ trợ nhưng hiện nay nổi bật hơn cả là Unreal Engine và Unity. Chúng tôi đã chọn Unity vì nó dùng C# dễ dàng học và sử dụng còn Engine sử dụng C++ khá khó khăn cho những ai không biết lập trình.

Visual Studio là phần mềm dùng để hỗ trợ lập trình. Phần mềm này tích hợp sẵn trong Unity bạn chỉ cần tích vào trong lúc cài đặt Unity là được. Phần mềm khá là dễ sử dụng và dễ học. Vì vậy chúng tôi chọn phần mềm này để lập trình cho hệ thống theo ngôn ngữ C#.

PowerPoint là phần mềm hỗ trợ thuyết trình nhưng cũng có thể dùng để thiết kế 2D vì nó khá dễ dàng sử dụng, có khá đầy đủ các tính năng để thiết kế 2D và nó còn giúp xuất file dưới dạng PNG và JPG. Vì vậy chúng tôi đã chọn phần mềm này để tạo ra dữ liệu 2D cho dự án.



***Hình 3.2.*** Sơ đồ quy trình xây dựng hệ thống thực tế ảo tăng cường trên điện thoại

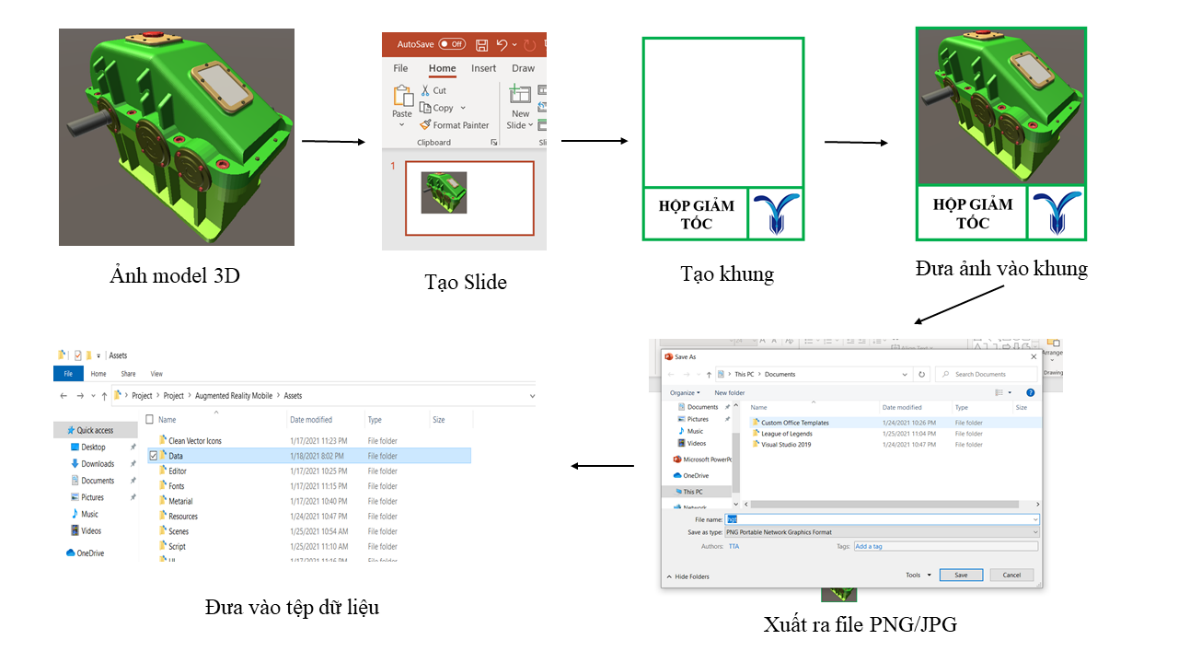
## **3.3. Tạo cơ sở dữ liệu 2D**

Để xây dựng mô hình 2D ta có thể sử dụng theo 2 cách : vẽ thủ công hoặc sử dụng phần mềm hỗ trợ. Với cách thứ nhất là vẽ thủ công sẽ yêu cầu có kỹ năng vẽ và mất khá nhiều thời gian còn cách thứ hai thì đơn giản mà không cần tốn quá nhiều công sức. Do vậy ở đây chúng tôi chọn cách thứ 2.

Hiện nay có rất nhiều phần mềm hỗ trợ như paint, CS6, …. Nhưng ở đây để nhanh chóng, dễ sử dụng và thuận tiện chúng tôi sử dụng PowerPoint. Phần mềm PowerPoint có tính năng giúp người dùng xuất file dưới dạng PNG hoặc JPG và phần mềm khá là dễ sử dụng và quen thuộc với chúng ta.

Để tạo ra các mô hình 2D ta làm theo các bước sau:

* Bước 1: chụp một tấm hình của mô hình 3D.
* Bước 2: tạo một Slide trong PowerPoint.
* Bước 3: thiết kế khung cho mô hình 2D.
* Bước 4: chèn ảnh đã chụp ở bước 1 vào trong khung vừa thiết kế ở bước 3.
* Bước 5: xuất Slide dưới dạng PNG hoặc JPG và lưu vào 1 file để tạo data cho hệ thống.



***Hình 3.3.*** Các bước tạo dữ liệu 2D

## **3.4. Tạo cơ sở dữ liệu 3D**

Để xây dựng mô hình 3D trong lĩnh vực ô tô chúng tôi sử dụng phần mềm Solidworks để vẽ các chi tiết rồi lắp ghép lại với nhau tạo thành một mô hình hoàn chỉnh. Làm từng chi tiết riêng lẽ sẽ giúp cho hệ thống có một dữ liệu lớn và linh hoạt hơn.

Khi đã vẽ xong các chi tiết của một mô hình lớn thì chúng tôi bắt đầu ghép chúng lại với nhau. Để vẽ các chi tiết chúng tôi sử dụng Part của Solidworks còn lắp ghép thì chúng tôi sử dụng Assembly của Solidworks.

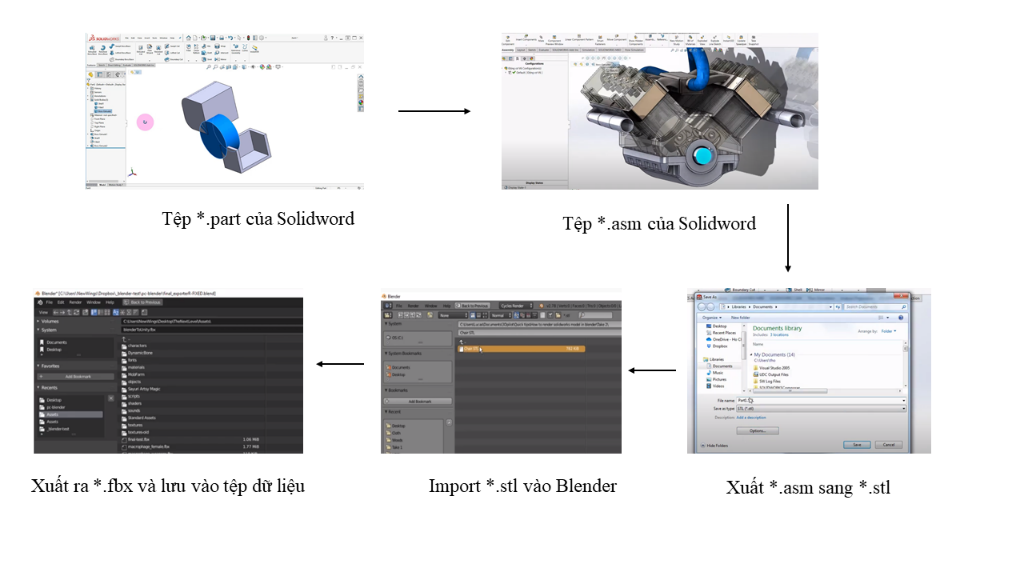
Sau khi lắp ghép xong thì chúng tôi sẽ xuất file dưới dạng STL thì chúng tôi bắt đầu đưa các file STL vừa tạo thành vào trong Blender bằng lệnh Import trong Blender.

Sau khi đưa vào trong Blender chúng tôi xuất file dưới dạng FBX hoặc OBJ. FBX sẽ tạo thành các chi tiết rời rạc còn OBJ sẽ kết nối các chi tiết rời rạc đó thành một khối. Tức là file FBX sẽ có những file OBJ con nằm bên trong còn file OBJ chỉ có mình nó.

Sau khi đã có file FBX hoặc OBJ thì chúng ta đưa vào Unity vào cách kéo đối tượng từ ngoài vào trong Unity hoặc sao chép hoặc di chuyển chúng vào thư mục nơi chứa dự án.

Khi đưa vào Unity thì các mô hình 3D sẽ bị mất mày. Để tô màu cho mô hình 3D ta có thể tô màu tại Blender hoặc Unity. Ở Blender thì chúng ta có 2 cách là dùng Texture Image hoặc tô màu theo lưới. Nếu dùng Texture Image thì cần có một bức ảnh với màu sắc khớp với từng điểm trên mô hình 3D. Việc này khá là khó khăn và tốn thời gian. Nếu tô màu theo lưới thì rất mất thời gian và các mô hình gồm nhiều chi tiết sẽ rất khó tô và dễ gây lem màu. Do vậy chúng tôi đã chọn là tô màu trên Unity với file định dạng FBX.

Tô màu trên Unity thì chúng tôi dùng Material để tô màu từng chi tiết nên sẽ không bị lem màu và màu sắc cũng có thể dễ dàng thay đổi linh hoạt nhưng file dưới dạng OBJ sẽ không tô được theo ý muốn được vì mỗi OBJ chỉ được tô một màu.



***Hình 3.4.*** Các bước tạo dữ liệu 3D

**3.5. Kết nối các thành phần của hệ thống dựa trên Unity và xuất file APK/IOS**

Chúng tôi đã đề cập sử dụng Unity để kết nối mới thứ lại với nhau ở trên. Để làm rõ ràng hơn thì chúng tôi sẽ chia làm các bước là đưa dữ liệu 2D và 3D vào Unity, viết các lệnh trên Visual Studio và đưa vào Unity, thêm công nghệ Vuforia vào Unity, xây dựng giao diện người dùng và xuất thành tệp dưới dạng APK cho Android (IOS tương tự).

Để đưa dữ liệu 2D và 3D vào Unity ta chỉ cần tạo 1 tệp data rồi đưa các file vào đó. Sau đó di chuyển tệp data vừa tạo vào nơi chứa dự án Unity.

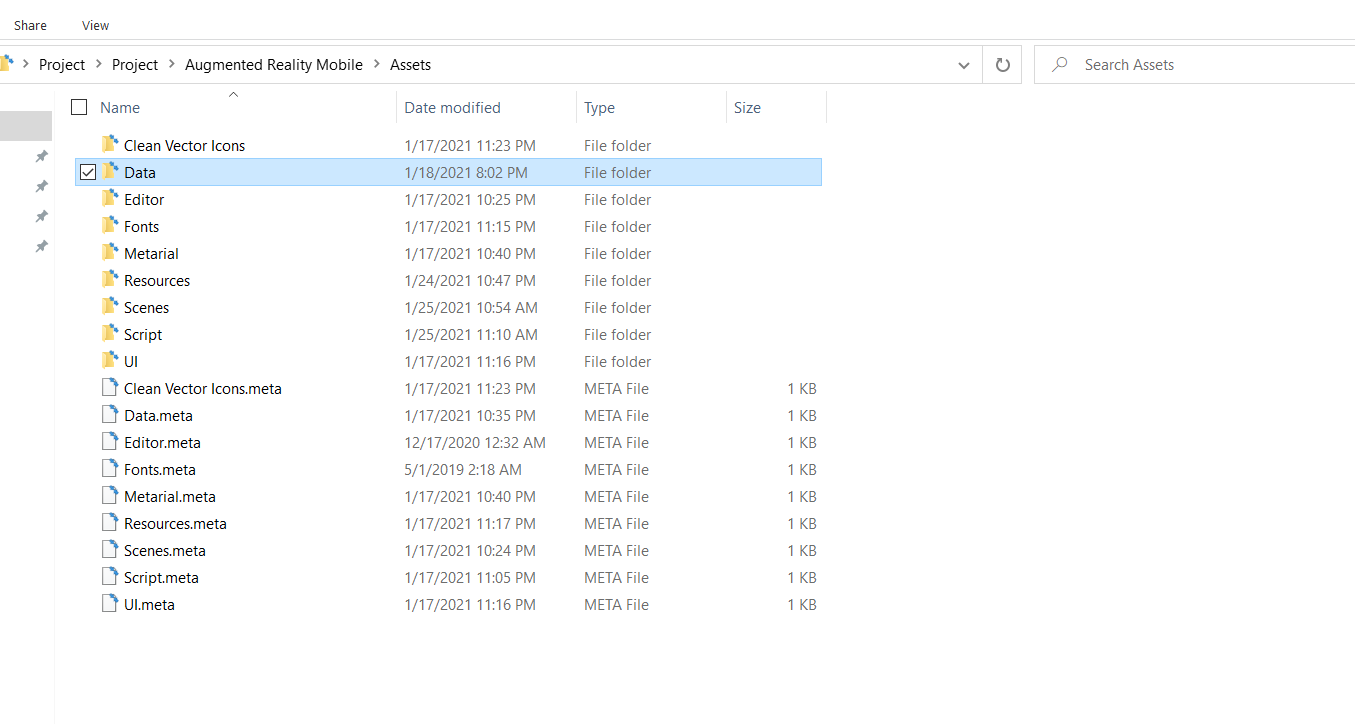
Trong Unity ta đưa con trỏ chuột vào phần project nhấn chuột phải chọn create sau đó chọn C# Script. Đặt tên cho tệp vừa tạo rồi enter để mở file. Trong đó chúng ta bắt đầu viết code, sau khi viết xong ta save lại là được.

Muốn thêm công nghệ Vuforia vào Unity chúng ta chỉ cần lên trang chủ của Vuforia Engine đăng nhập và tải về. Trong quá trình chờ tải về hãy tạo một dự án phát triển trên trang chủ Vuforia Engine và sao chép đoạn key của dự án để kích hoạt camera AR của Vuforia. Khi tải về xong chúng ta chỉ cần kéo thả tệp tải về vào phần Project của Unity thì sẽ hiện ra một bảng và chúng ta chỉ việc Import là xong. Vậy là đã thêm Vuforia vào Unity. Khi sử dụng Camera AR bạn phải thêm key đã sao chép từ trước vào thì nó mới hoạt động được.

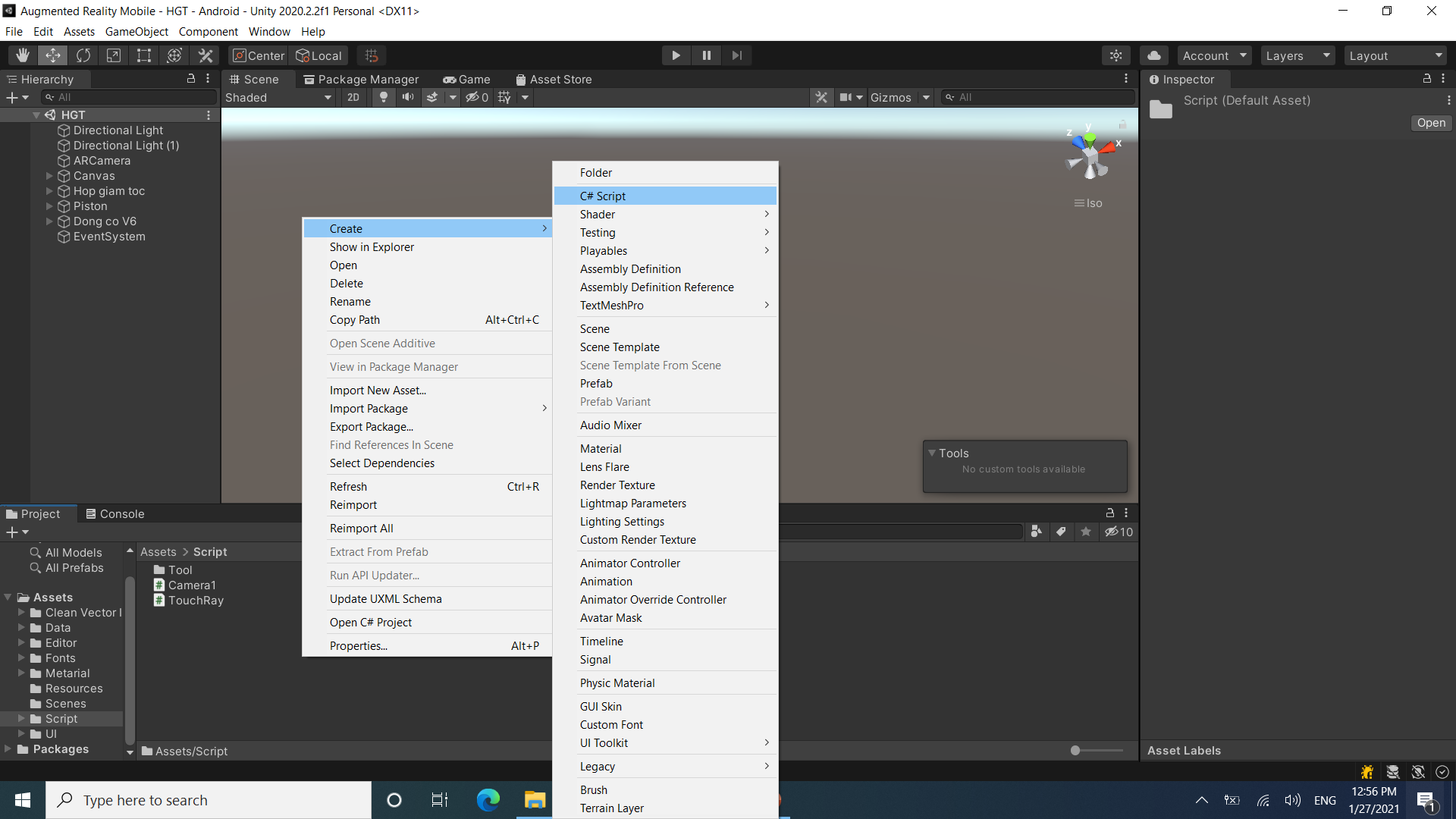
Để xây dựng giao diện người chúng tôi đã sử dụng công cụ UI trong Unity. Nó cung cấp khá đầy đủ để thiết lập một giao diện như ý muốn. Chúng ta chỉ cần đưa con trỏ chuột vào phần Hierarchy rồi nhấn chuột phải thì công cụ UI sẽ hiện ra.

Sau khi hoàn thành xong dự án thì chúng tôi sẽ vào file chọn build settings. Sau đó chọn Android. Chúng tôi cài đặt ETC2 fallback là 16-bit để các điện thoại đều sử dụng được. Sau đó chúng tôi vào Player Settings để thiết lập thông số bên trong như tên công ty, tên phần mềm, phiên bản phần mềm, biểu tượng phần mềm, định dạng màn hình ngang dọc, hệ màu và nhiều tính năng khác.

Khi đã thiết lập xong chúng tôi quay lại bảng Build Settings chọn Build. Unity sẽ bắt đầu chuyển thành dạng APK, đợi một lúc khi hệ thống build xong thì bạn chỉ cần đưa file APK vừa tạo xong vào điện thoại và cài đặt.



***Hình 3.5.*** Đưa dữ liệu vào trong nơi chứa các tệp dự án

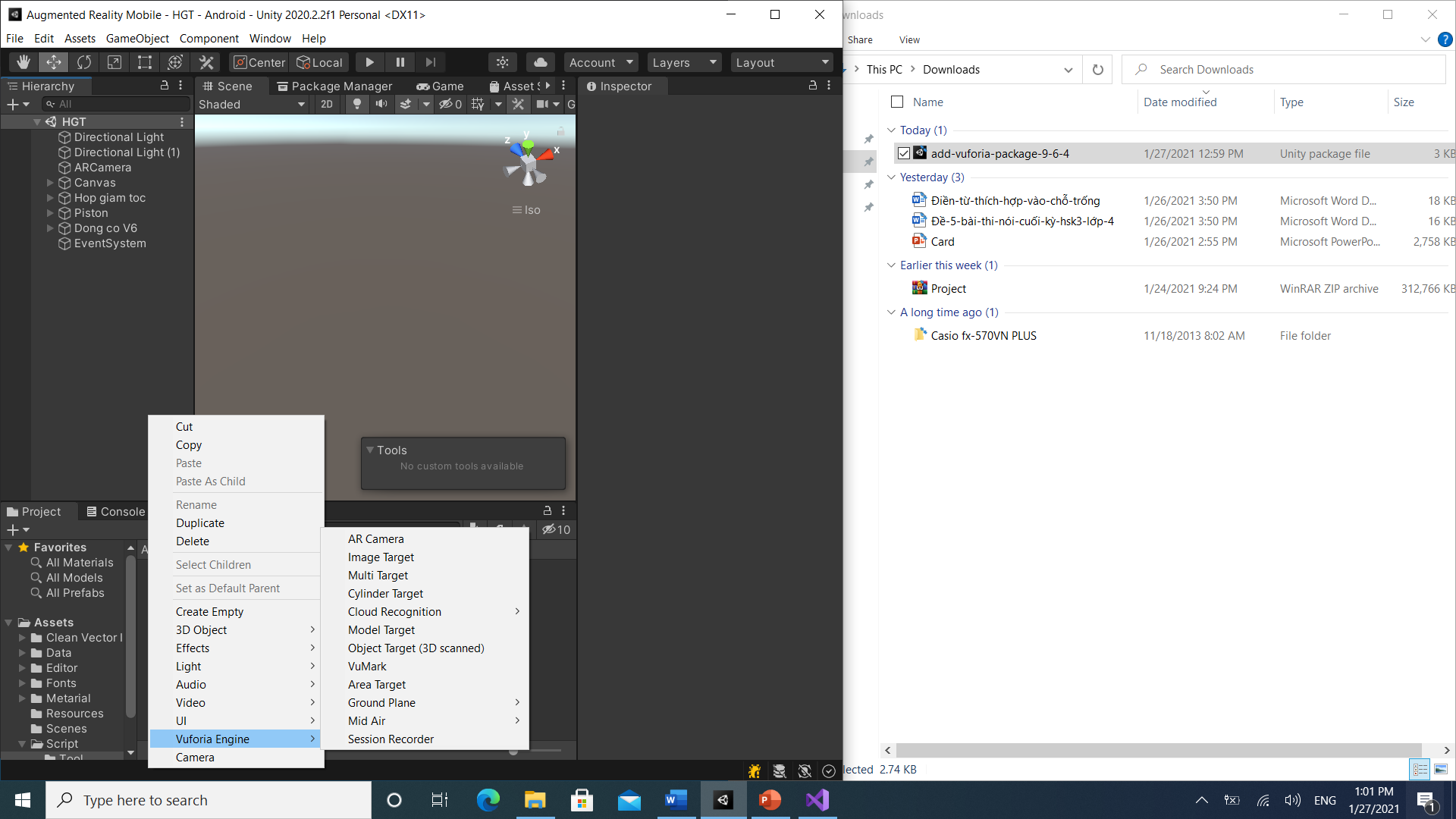


***Hình 3.6.*** Tạo tệp C# để lập trình cho hệ thống

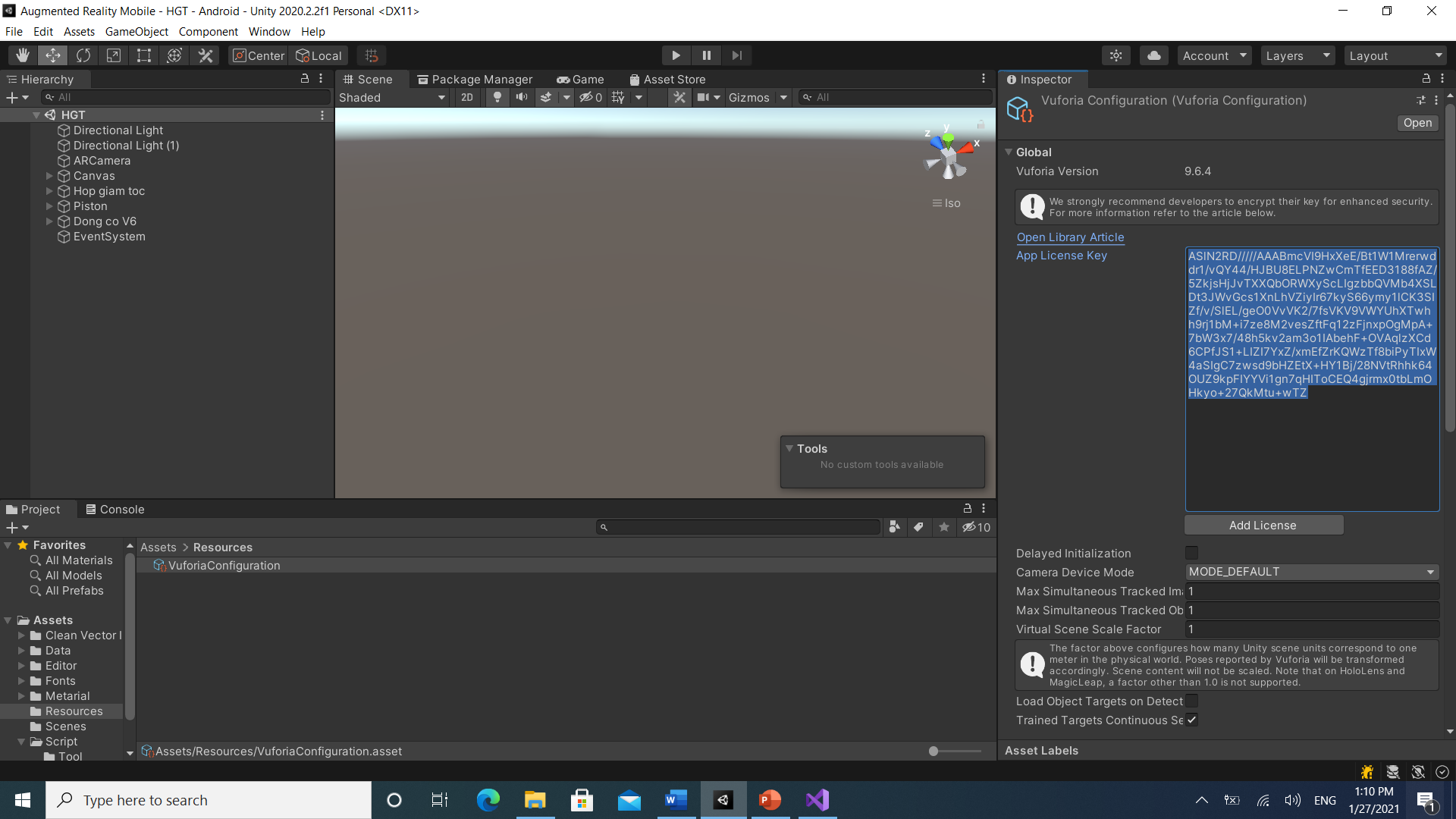
A screenshot of a computer screen

Description automatically generated

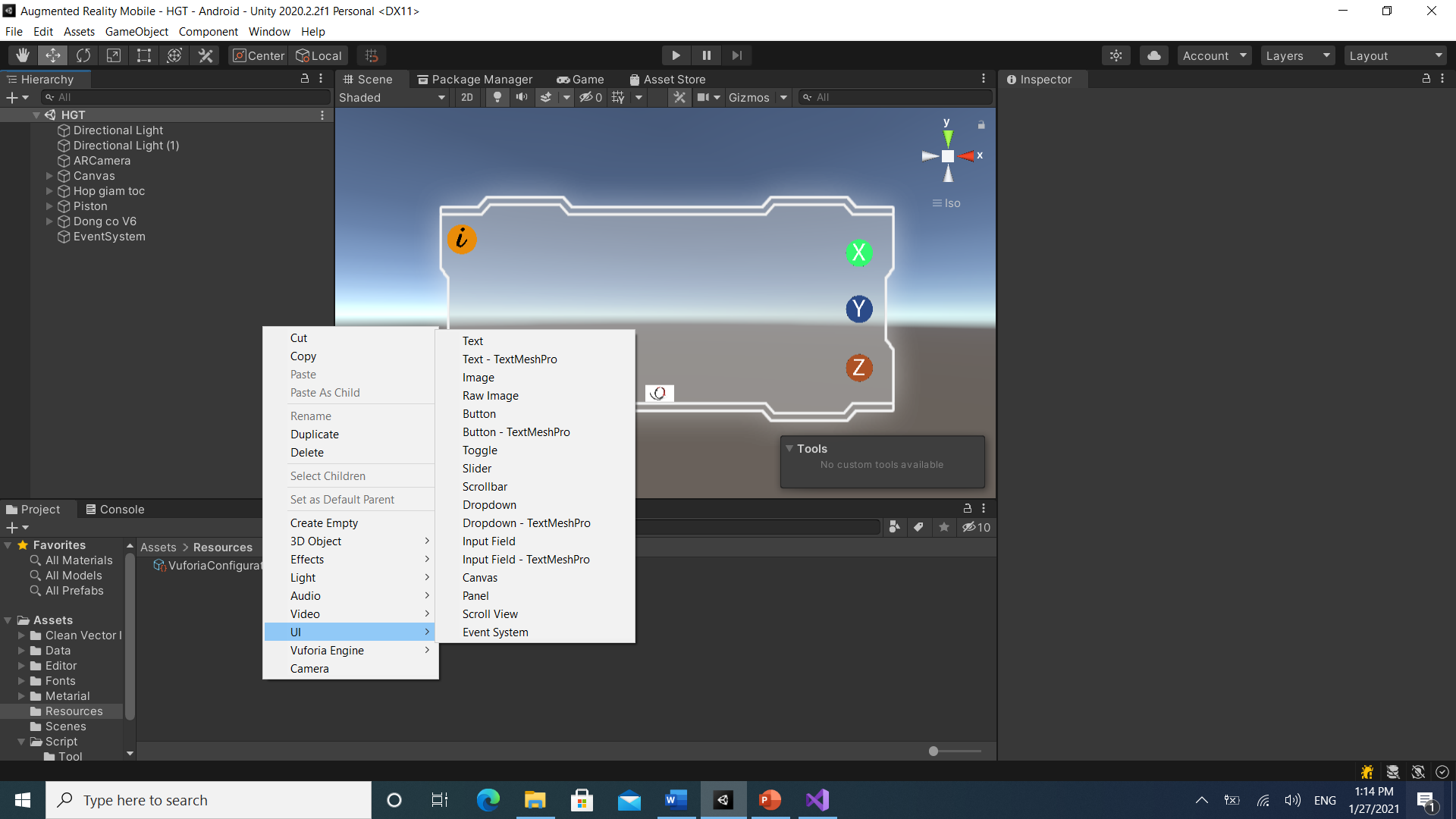
***Hình 3.7.*** Lập trình các tính năng cho hệ thống



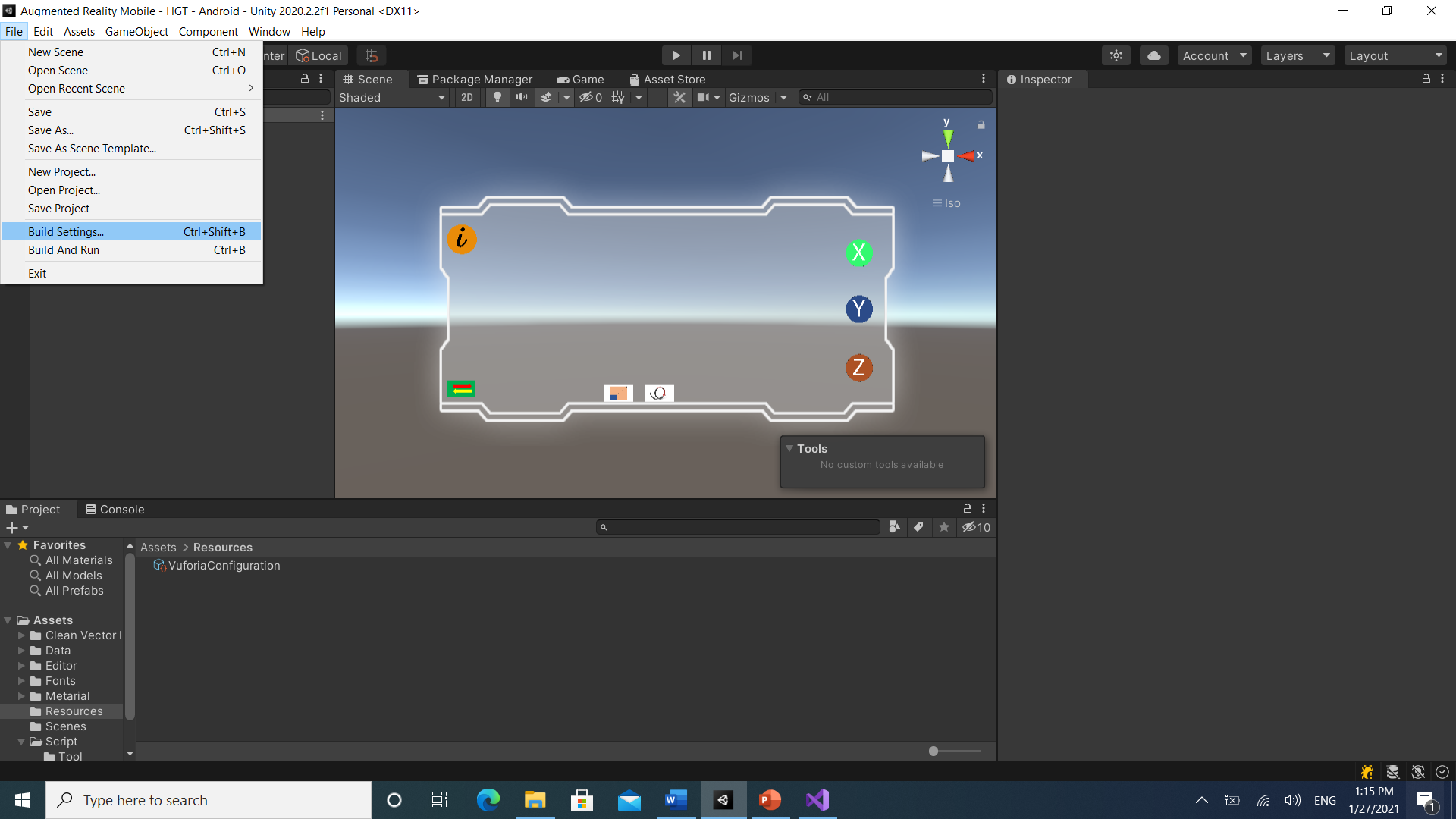
***Hình 3.8.*** Thêm Vuforia Engine vào Unity



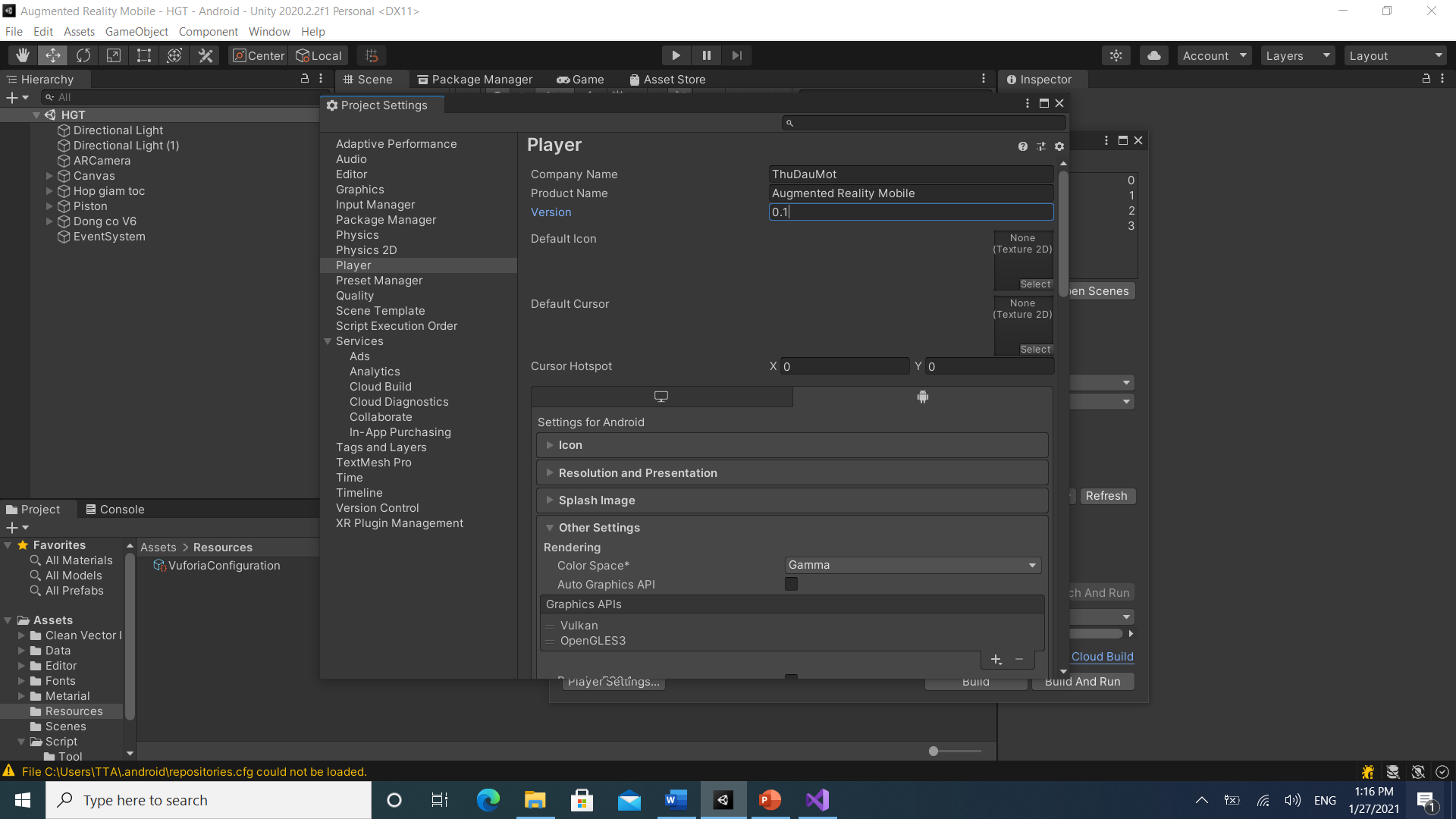
***Hình 3.9.*** Thêm key vào AR Camera



***Hình 3.10.*** Công cụ hỗ trợ thiết kế giao diện cho Unity



***Hình 3.11.*** Thiết lập cho Unity



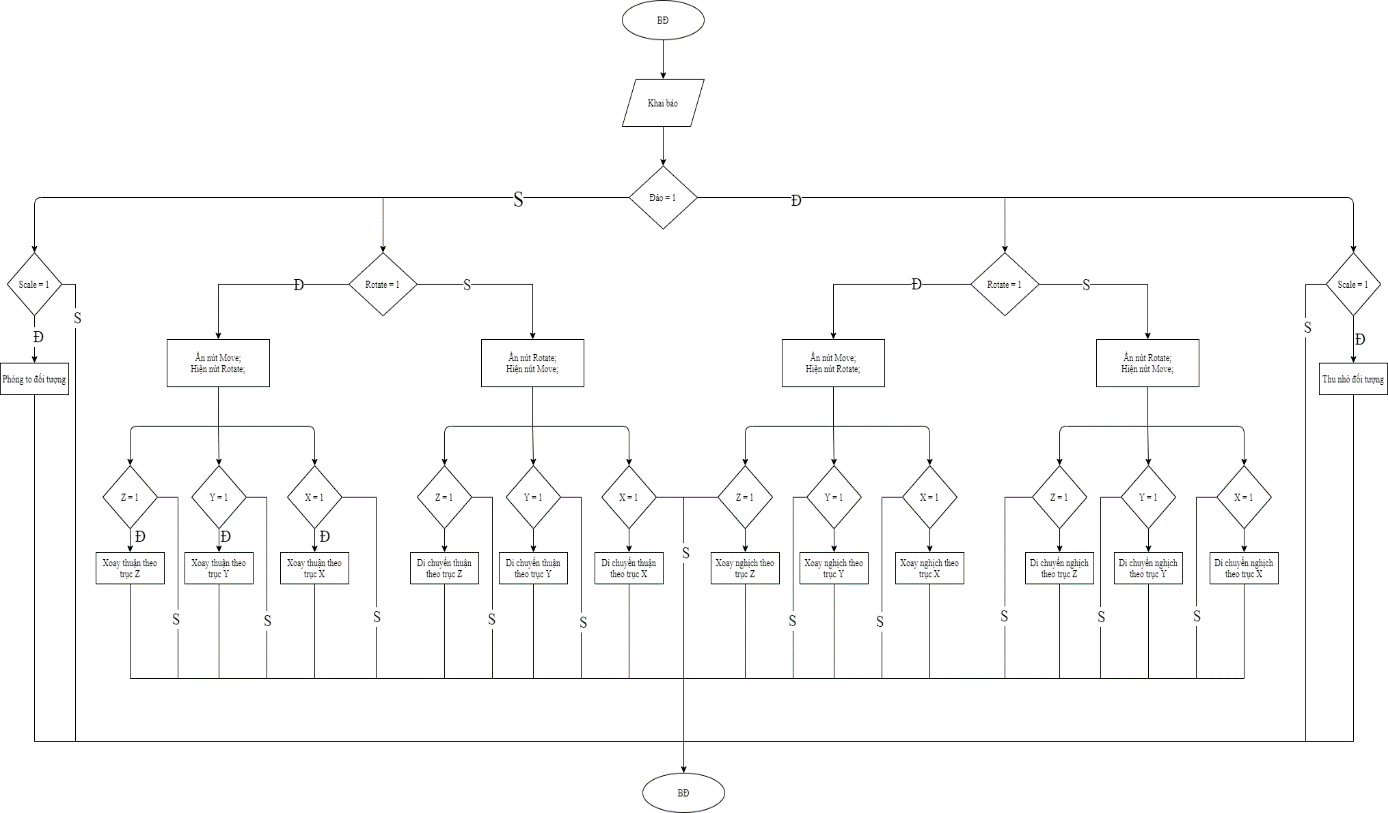
***Hình 3.12.*** Thiết lập các thông số và thông tin cho hệ thống

# **CHƯƠNG 4: LẬP TRÌNH TÍNH NĂNG VÀ TẠO GIAO DIỆN**

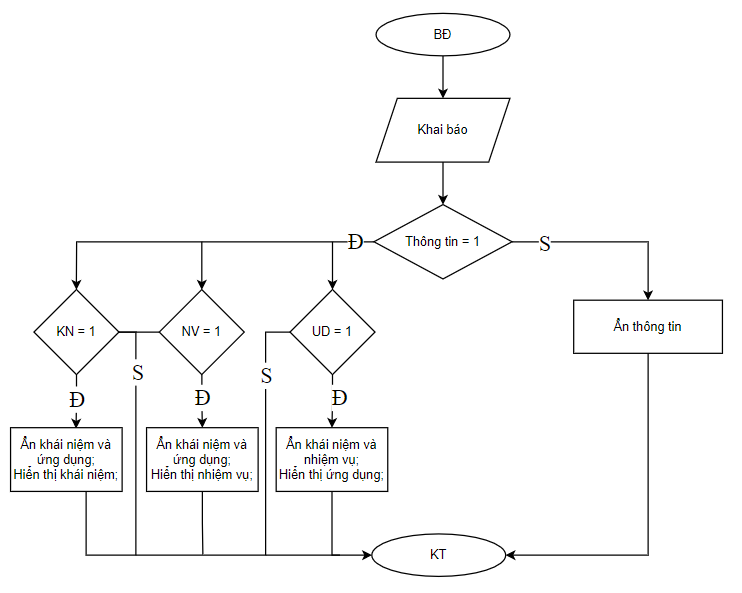
## **4.1. Lập trình các tính năng**

Các tính năng chính bao gồm: ẩn hiện, xoay, di chuyển đối tượng. Để giảm các nút nhấn và tăng diện tích hiển thị mô hình và giảm dung lượng hệ thống nên ta sử dụng các phím đảo như đảo chiều xoay, di chuyển và đảo trạng thái xoay, di chuyển. Để ẩn hiện các đối tượng ta sử dụng lệnh SetActive(). Để xoay đối tượng ta sử dụng lệnh transform.Rotate(). Để di chuyển ta sử dụng lệnh transform.Translate().

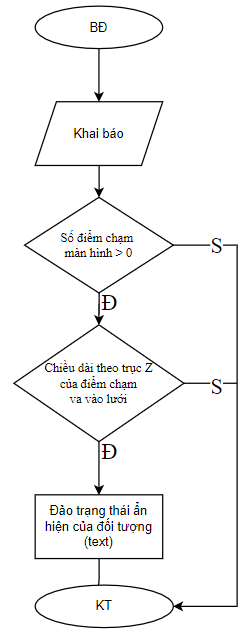
Để tăng tính tương tác giữa người dùng và hệ thống em đã thêm tính năng chạm chi tiết của mô hình. Khi người dùng nhấn vào vị trí các chi tiết trên màn hình thì sẽ hiện ra bảng hiển thị tên chi tiết đó là gì. Để làm được như vậy em đã dùng tính năng RayCastHit. Tính năng này giúp tạo một tia có tọa độ (x,y) và được chiếu theo trục z. Ở đối tượng em sẽ tạo lưới bằng Collider. Lưới này sẽ giúp em phát hiện được tia đó chiếu vào chi tiết nào của mô hình 3D. Từ đó em sẽ hiển thị ra tên của chi tiết đó. Có thể tìm hiểu rõ hơn các lệnh bằng các truy cập vào tài liệu của Unity. Code có thể tham khảo ở phần phụ lục.



***Sơ đồ 4.1.*** Lưu đồ giải thuật các tính năng cơ bản



***Sơ đồ 4.2.*** Lưu đồ giải thuật các tính năng ẩn hiện thông tin



***Sơ đồ 4.3.*** Lưu đồ giải thuật tính năng tương tác cao (chạm đối tượng)

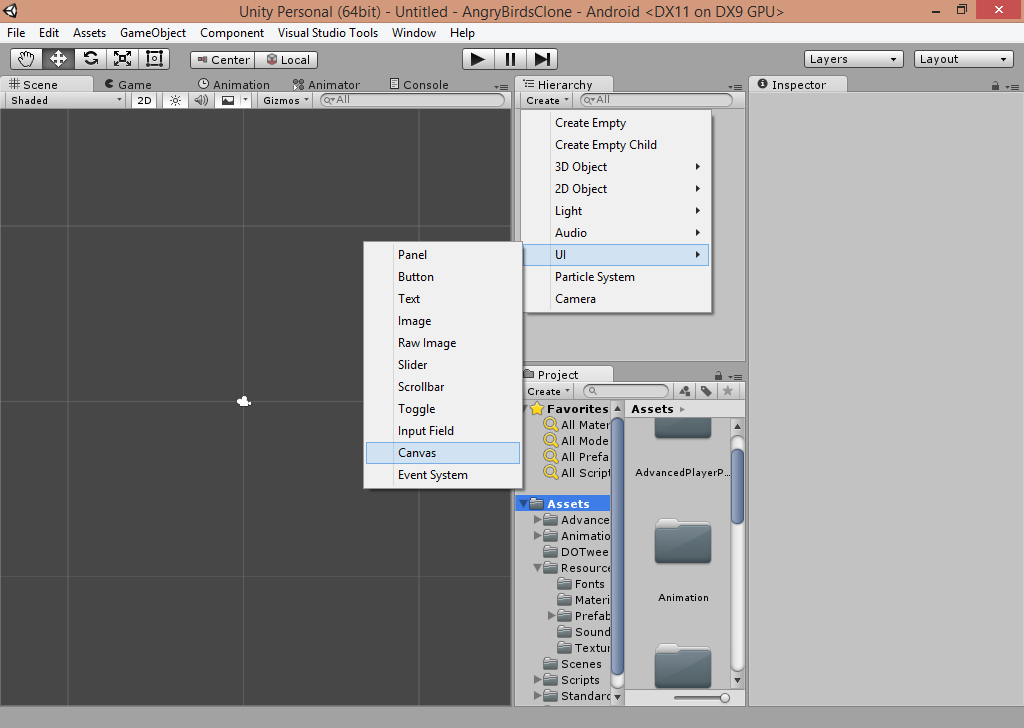
**4.2. Tạo giao diện**

Thiết kế giao diện người dùng đơn giản là sử dụng các assets có sẵn (hình ảnh, font chữ, các hiệu ứng, ...), sắp xếp chúng theo 1 bố cục được Designer thiết kế, các assets này có thể tìm thấy trên các website, Assets Store của Unity, hoặc do chính các Designer, Artist trong dự án thiết kế. Các thành phần cơ bản trong thiết kế UI bao gồm Canvas, Text, Image, Button, …

**4.2.1. Canvas**

Canvas là thành phần chính không thể thiếu trong thiết kế UI, các thành phần UI khác khi được khởi tạo bắt buộc phải nằm trong 1 Canvas. Khi khởi tạo 1 thành phần UI (Text, Image, ...), Unity sẽ tự động tạo ra 1 Canvas nếu chưa tồn tại trong Scene.

Để khởi tạo 1 Canvas, trong cửa sổ Hierarchy, chọn Create → UI → Canvas. Các đối tượng UI khác cũng được khởi tạo tương tự.



***Hình 4.1.*** Lấy Canvas trong UI của Unity

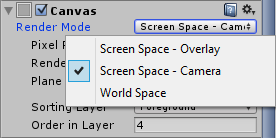
Các đối tượng con của 1 Canvas sẽ được render theo thứ tự từ trên xuống trong cửa sổ Hierarchy, đối tượng nào ở trên sẽ được render trước và có thể bị che khuất bởi đối tượng phía dưới.

Các chức năng quan trọng của Canvas:

**Render mode**

Có 3 tuỳ chọn hiển thị Canvas:

* Screen Space – Overlay: Canvas sẽ được vẽ lên layer cao nhất của màn hình và nằm trên mọi GameObject khác. Canvas với render mode này hoàn toàn không phụ thuộc vào camera.
* Screen Space – Camera: đối với mode này, cần chỉ định 1 camera cho Canvas, nó sẽ được render theo camera, nếu như không có camera được chỉ định thì Canvas và các thành phần bên trong sẽ không được render.
* World Space: với tuỳ chọn này, đối tượng Canvas sẽ được xem như 1 GameObject thông thường, tuỳ chọn này sử dụng event camera thay vì render camera. Ngoài các chức năng như render camera, event camera còn có thêm chức năng bắt sự kiện, dựa trên thứ tự render, toạ độ z, ... của các đối tượng UI.



***Hình 4.2****.* Render Mode của Canvas

Đối với các tùy chọn render theo Screen Space, Unity cung cấp tính năng Pixel Perfect, tăng khả năng hiển thị sắc nét và khử vết mờ.

**Rect Transform**

Tương tự như thành phần Transform trong các GameObject khác, Rect Transform được sử dụng để xác định kích thước, vị trí của giao diện.

Đối với các tuỳ chọn render mode Screen Space - Overlay và Screen Space - Camera, thành phần Rect Transform sẽ được khoá lại và không thể tuỳ chỉnh. Canvas sẽ điều chỉnh các thông số 1 cách tự động để phù hợp với độ phân giải màn hình game.

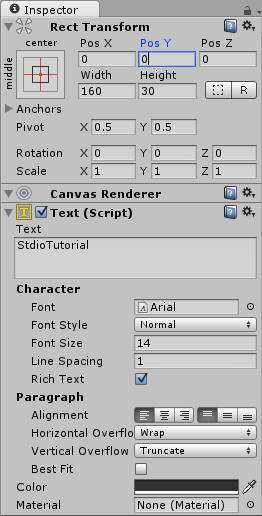
**Graphic Raycast**

Hỗ trợ bắt sự kiện, khi nhận được tín hiệu (mouse click, touch, ...), 1 tia nhìn tại vị trí tương tác sẽ được tạo ra, bằng cách này dễ dàng xác định được đối tượng người chơi muốn tương tác, thông qua tọa độ z của đối tượng.

**4.2.2. Text**

UI Text được sử dụng để hiển thị các thông tin trên màn hình như Score, Lives, Times, ... 1 số game sử dụng các texture riêng để hiển thị thông tin thay cho text, tuy nhiên text vẫn là lựa chọn phổ biến hơn vì tính đơn giản và dễ thao tác.

Sau khi khởi tạo 1 đối tượng Text, cửa sổ Inspector có dạng như sau:

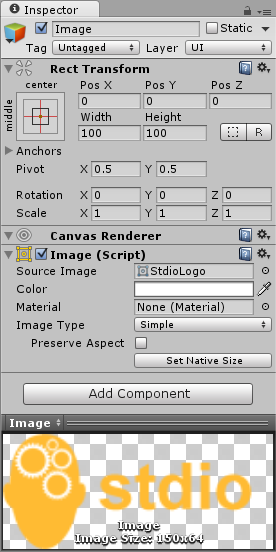


***Hình 4.3.*** Inspector của Text

* Rect Transform: quản lý vị trí, kích thước, góc quay, ... của Text.
* Text: lưu trữ chuỗi ký tự cần hiển thị ra màn hình. Unity hỗ trợ 1 số chức năng để tuỳ biến chuỗi như font, kiểu chữ (thường, in đậm, in nghiêng, ...), cỡ chữ, màu chữ, ... Tuỳ chọn Best Fit sẽ tự động điều chỉnh kích thước font chữ phù hợp với kích thước được quy định trong Rect Transform.
  + 1. **Image**

UI Image được sử dụng rất phổ biến, như thiết kế background, các button, title, ...

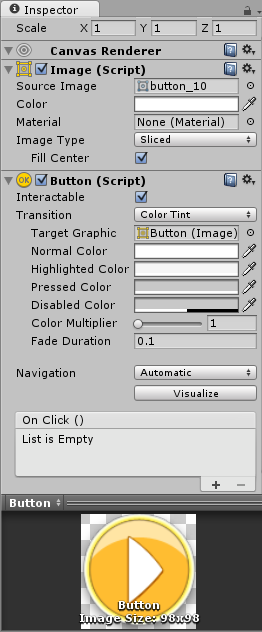
Chọn hình ảnh và kéo thả vào khung Source Image, ngoài ra còn có 1 số tuỳ chọn thay đổi màu sắc, chất liệu, ...



***Hình 4.4.*** Inspector của Image

* + 1. **Button**

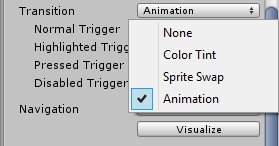
UI Button là 1 thành phần quan trọng, giúp người chơi tương tác với game. 1 số button quen thuộc có thể thấy trong nhiều game là Play, Pause, Resume, Replay, ...



***Hình 4.5.*** Inspector của Button

Unity cung cấp khá nhiều tính năng giúp hiển thị và thao tác với button được dễ dàng và đẹp hơn.

Về hiển thị, có 4 tùy chọn hiển thị trong phần Transition:

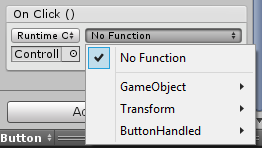


***Hình 4.6.*** Transition của Button

* None: sử dụng hình ảnh button được truyền vào thành phần Image. Với tùy chọn này, hình hiển thị button sẽ không thay đổi mỗi khi người chơi thao tác.
* Color Tint, Sprite Swap và Animation có các tùy chọn hiển thị riêng cho từng trạng thái của button. Với Color Tint, button sẽ thay đổi màu sắc theo từng trạng thái, Sprite Swap có hỗ trợ thay đổi sprite theo trạng thái, Animation hỗ trợ thêm các đặc tính như scale, transform, … Tùy vào yêu cầu của game mà lựa chọn tùy chọn phù hợp.

Về phần tương tác, Unity cung cấp sẵn event On Click. Khi click vào button, tất cả các hành động được thiết lập sẵn trong event On Click sẽ được thực hiện, do đó có thể thực hiện đồng thời nhiều hành động tùy theo nhu cầu.

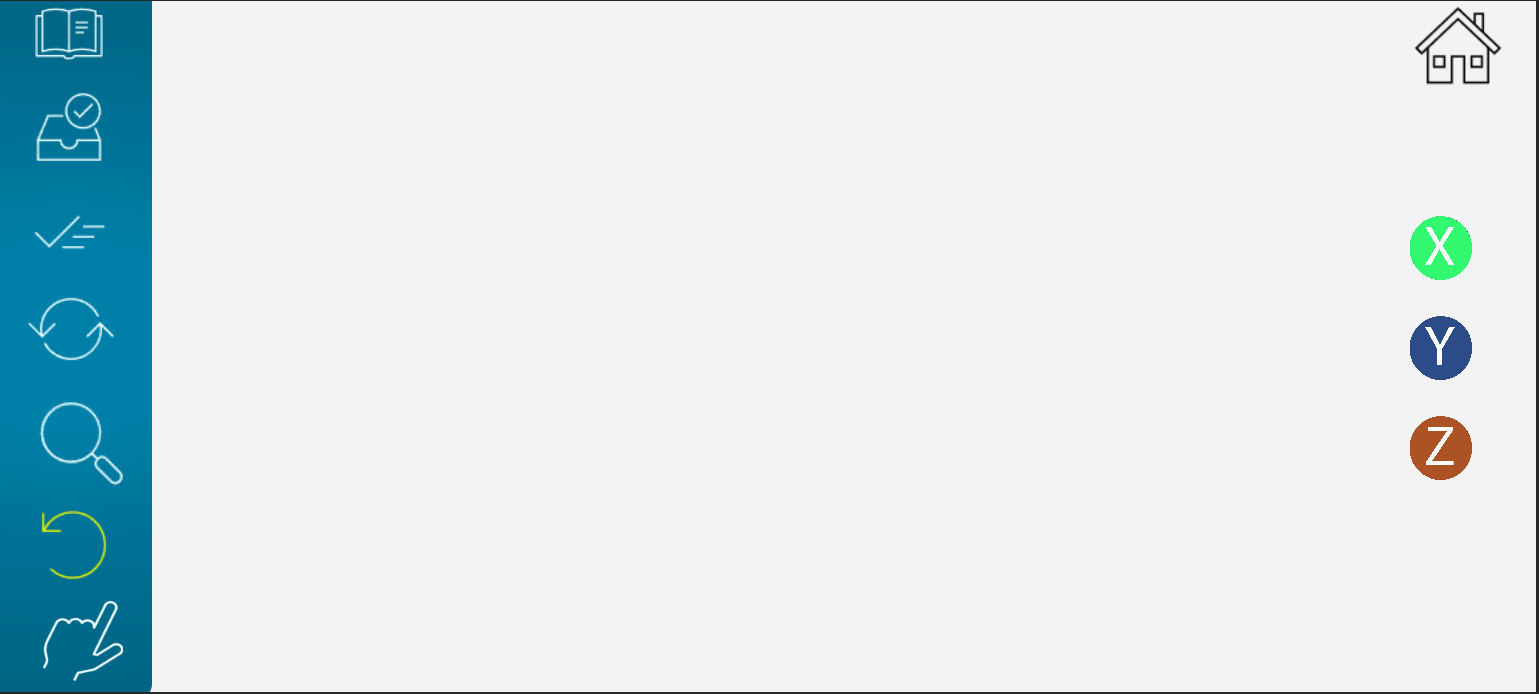
Các đối tượng được thêm vào có thể là 1 GameObject trong cửa sổ Hierarchy hoặc chính button đó. Khi đó event sẽ tự động nhận được các thành phần của GameObject và hiển thị trong menu như trong hình dưới.



***Hình 4.7.*** On Click của Button

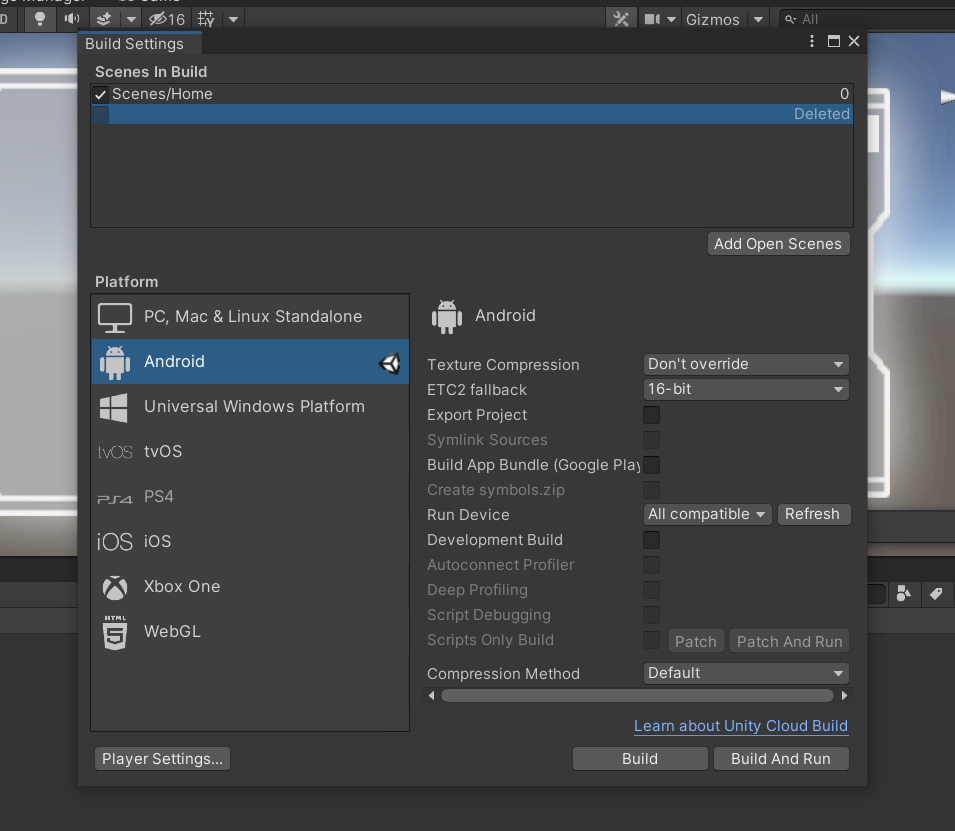
*\* Nếu cần gọi 1 hàm trong script, script đó phải được gắn vào 1 GameObject trong cửa sổ Hierarchy và hàm này cần có phạm vi truy cập là public.*

Ngoài ra, còn hàng loạt event khác cho button được cung cấp trong thành phần Event Trigger (Add Component -> Event -> Event Trigger). Cách sử dụng tương tự như với event On Click.

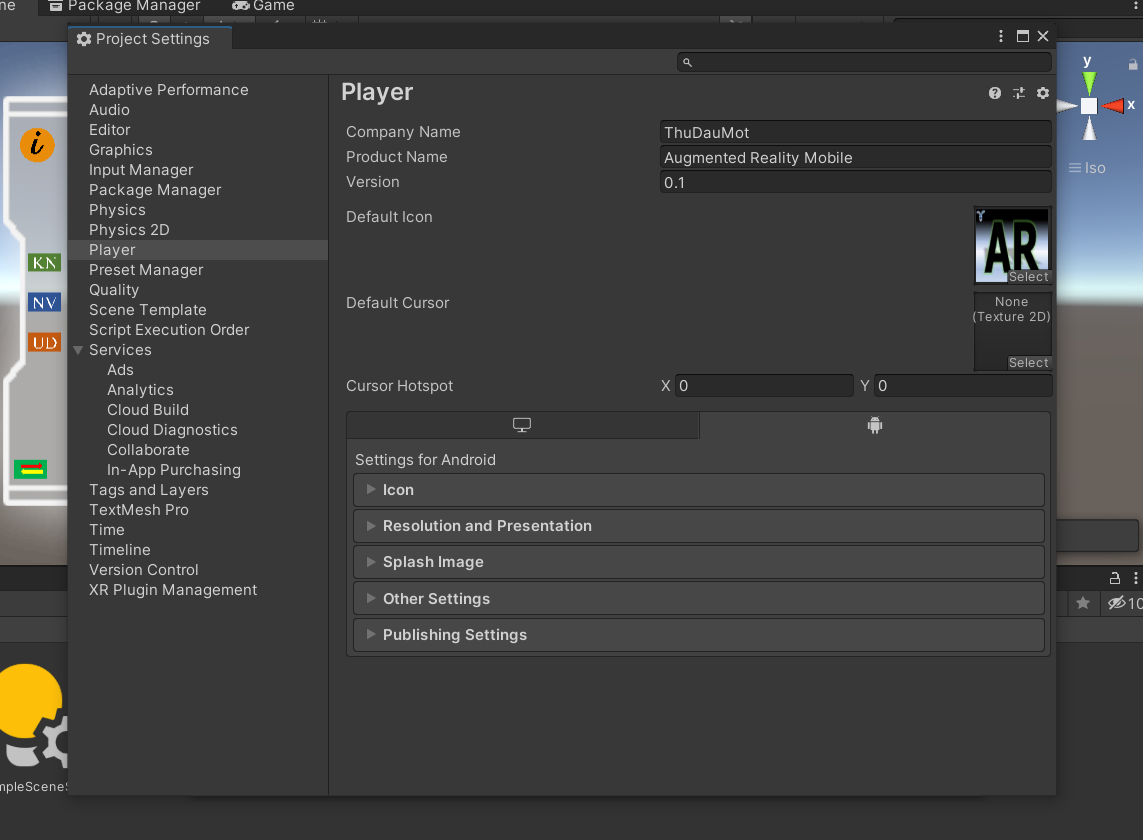
****

***Hình 4.8.*** Giao diện của hệ thống

* 1. **Build hệ thống thành file apk**

Sau khi đã hoàn thành hệ thống ta nhấn tổ hợp phím Ctrl + Shift + B thì hộp thoại build settings sẽ hiện ra và sau đó ta chọn player settings và thiết lập rồi nhấn Build.

***Hình 4.9.*** Hộp thoại build settings

****

***Hình 4.10.*** Hộp thoại player

**CHƯƠNG 5: TỔNG KẾT**

***Bảng 5.1.*** Thông tin về hệ thống

| **Hệ điều hành** | Androi 8.0 trở lên |
| --- | --- |
| **Kích thước màn hình yêu cầu** | 720x1520 pixel |
| **Dung lượng hệ thống (APK)** | 102 MB |
| **Số mô hình** | 8 mô hình |
| **Các tính năng cơ bản** | Xoay đối tượng, ẩn đối tượng, phóng to thu nhỏ đối tượng và di chuyển đối tượng |
| **Tính năng tương tác cao** | Chạm đối tượng |

8 mô hình bao gồm: Động cơ khí, Động cơ hướng tâm, động cơ V6, động cơ V2, động cơ V8, động cơ stirling, hộp giảm tốc, pis-tông.

**Tổng kết các vấn đề gặp phải:**

* Chọn phần mềm và công nghệ: Phần mềm và công nghệ phải dễ học và sử dụng (có nhiều tài liệu và video hướng dẫn), có khả năng phát triển hệ thống.
* Chuyển mô hình 3D từ Solidwork vào Unity: Giải pháp phải tiết kiệm, hiệu quả, nhanh chóng, dễ thực hiện và có lợi ích cho việc phát triển hệ thống về sau.
* Mất màu khi đưa mô hình 3D vào Unity và giải pháp tô màu: Giải pháp phải dễ dàng thực hiện, nhanh chóng, hiệu quả và có lợi cho việc phát triển hệ thống về sau.
* Tính năng tương tác nâng cao (chạm chi tiết mô hình 3D): Độ chính xác phải tương đối.

**Tổng lí do chọn các phần mềm và công nghệ:**

* Chọn phần mềm Unity mang tính trọng yếu nhất vì nó là phần mềm giúp liên kết thành phần của hệ thống lại và xuất file dưới dạng APK. Lý do chọn Unity vì hiện nay chỉ có 2 phần mềm nổi trội là Unity và Unreal Engine. Vì Unity viết bằng ngôn ngữ C# (một ngôn ngữ khá dễ học và được ứng dụng nhiều) và Unreal Engine được viết bằng C++ (một ngôn ngữ khá khó cho người không biết lập trình). Vì vậy chọn phần mềm Unity.
* Chọn phần mềm Solidworks vì là phần mềm khá dễ sử dụng và em đã tiếp xúc với nó từ trước.
* Chọn phần mềm Blender vì nó khá nhẹ và miễn phí khác với các phần mềm tương tự như Maya, … (mất phí và dung lượng khác nặng).
* Chọn phần mềm PowerPoint để thiết kế 2D vì phần mềm này là phần mềm hầu như ai cũng biết sử dụng và luôn có sẵn trong các máy tính và laptop.
* Chọn phần mềm Visual Studio vì nó được tích hợp sẵn để lập trình C# trong Unity và hầu như ai học công nghệ thông tin đều biết sử dụng.
* Công nghệ Vuforia là công nghệ cung cấp các công cụ trích đặc điểm ảnh từ camera và so sánh với hình ảnh trong data. Công nghệ này rất phổ biến, dễ dàng sử dụng và miễn phí.

**Tổng hợp lý do chọn phải pháp:**

* Chuyển đổi mô hình 3D từ Solidwork sang Unity có rất nhiều giải pháp nhưng chủ yếu chỉ có 2 loại là sử dụng công nghệ thứ 3 mất phí hoặc sử dụng phần mềm để chuyển đổi thành file .Obj hoặc .Fbx để đưa vào Unity. Vì vậy em chọn giải pháp thứ 2, tuy mất nhiều thời gian hơn nhưng tiết kiệm được chi phí.
* Tiếp theo sẽ chọn dạng xuất ra là .Obj hay .Fbx. Em chọn .Fbx vì dạng .Obj thì các chi tiết sẽ liên kết lại thành 1 chi tiết và các đặc tính trên các chi tiết (màu sắc, …) đều được đồng nhất nên việc thay đổi màu sắc và tương tác sẽ rất mất thời gian và công sức. Nhưng với .Fbx thì các chi tiết chỉ được gộp lại thành 1 đổi tượng nhưng chúng vẫn mang các đặc tính riêng biệt và không tác động lẫn nhau nên việc thay đổi màu sắc, … rất là đơn giản và phát triển về sau sẽ không gặp khó khăn.

**Mất màu mô hình 3D khi đưa vào Unity và giải pháp tô màu:**

* Khi đữa vào Unity sẽ gặp một số trường hợp mất màu do các đối tượng Metarial bị mất khi xuất thành dạng khác. Để khác phục em sẽ phải tô màu lại cho các mô hình 3D này.
* Có 2 giải pháp là tô màu trên Unity và tô màu trên Blender. Nếu tô màu trên Blender thì có 3 cách là sử dụng Image Texter, tô màu vào mặt phủ 2D và tô màu từng chi tiết bằng Metarial. Còn trên Unity thì chỉ có tô màu bằng Metarial.
* Nếu sử dụng Image Texter thì ta cần 1 bức ảnh 2D có màu sắc phù hợp với mặt phủ 2D của mô hình. Điều này là rất khó vì sẽ không có bức ảnh 2D đó. Nếu tô màu vào mặt phủ 2D thì dễ bị nhem vào các chi tiết khác nếu là mô hình 3D phức tạp. Còn tô màu bằng Metarial sẽ dễ dàng và mô hình 3D sẽ đẹp hợn tô màu bằng Metarial Unity nhưng thay đổi màu sắc vật liệu bằng code C# trong Unity. Vì vậy em lựa chọn tô màu bằng Metaral trên Unity.

Khi xây dựng tính năng tương tác cao thì độ chính xác sẽ vô cùng thấp do các lệnh tương tác chính làm lệch góc tọa độ của camera vì do góc tọa độ lệch nên trục z cũng lệch và dẫn đên tia sẽ sai hướng. Để không xảy ra hiện tượng lệch góc tọa độ của camera thì cần chính thông số trong Canvas. Ta vào thuộc tính của Canvas chọn Canvas sau đó chỉnh Render Mode thành Screen Space – Camera.

**Kết luận:**

* Do thời gian, năng lực và nhân lực có hạn nên số mô hình của hệ thống còn hạn chế, chưa có nhiều tính năng như animation, kéo thả các chi tiết, thay đổi màu chi tiết, …..
* Tương lai sẽ phát triển và nâng cấp hệ thống với các tính năng mới như aimation, kéo thả và thay đổi màu chi tiết, thêm nhiều mô hình mới cho nhiều môn học và kết hợp thêm VR để tăng tính đắm chìm cho hệ thống.

# **TÀI LIỆU THAM KHẢO**

[1] Kesim, Mehmet, and Yasin Ozarslan. "Augmented reality in education: current technologies and the potential for education." Procedia-social and behavioral sciences 47 (2012): 297-302.

[2] Wu, Hsin-Kai, et al. "Current status, opportunities and challenges of augmented reality in education." Computers & education 62 (2013): 41-49.

[3] Chen, Peng, et al. "A review of using Augmented Reality in Education from 2011 to 2016." Innovations in smart learning. Springer, Singapore, 2017. 13-18.

[4] Lee, Kangdon. "Augmented reality in education and training." TechTrends 56.2 (2012): 13-21.

[5] Wang, Minjuan, et al. "Augmented reality in education and training: pedagogical approaches and illustrative case studies." Journal of ambient intelligence and humanized computing 9.5 (2018): 1391-1402.

[6] Paavilainen, Janne, et al. "The Pokémon GO experience: A location-based augmented reality mobile game goes mainstream." Proceedings of the 2017 CHI conference on human factors in computing systems. 2017.

[7] Chatzopoulos, Dimitris, et al. "Mobile augmented reality survey: From where we are to where we go." Ieee Access 5 (2017): 6917-6950.

[8] Khan, Tasneem, Kevin Johnston, and Jacques Ophoff. "The impact of an augmented reality application on learning motivation of students." Advances in Human-Computer Interaction 2019 (2019).

[9] Pasaréti, Otília, et al. "Augmented Reality in education." INFODIDACT 2011 Informatika Szakmódszertani Konferencia (2011).

[10] Mota, José Miguel, et al. "Augmented reality mobile app development for all." Computers & Electrical Engineering 65 (2018): 250-260.

**PHỤ LỤC**

**I. MỤC TIÊU CẦN ĐẠT ĐƯỢC TRONG THỜI GIAN THỰC HIỆN ĐỀ TÀI**

Xây dựng được một hệ thống thực tế ảo tăng cường hỗ trợ hiển thị mô hình trong học tập có tính năng tương tác cao.

**II. KẾ HOẠCH THỰC HIỆN ĐỀ TÀI**

**Bảng 1.pl.** Kế hoạch thực hiện đề tài

| **Thời gian**  **(bắt đầu-kết thúc)** | **Các nội dung, công việc**  **thực hiện** | **Sản phẩm** | **Người thực hiện** |
| --- | --- | --- | --- |
| **11/2020-12/2020** | Tìm hiểu tài liệu liên quan đến đề tài | Báo cáo | Trần Tuấn Anh |
| **12/2020-01/2021** | Nghiên cứu phần mềm Unity, thiết kế 3D, Vuforia | Tài liệu | Trần Tuấn Anh |
| **01/2021-04/2021** | Thiết kế giao diện và tạo kho dữ liệu | Thiết bị, App | Trần Tuấn Anh |
| **04/2021-06/2021** | Thi công | Thiết bị, App | Trần Tuấn Anh |
| **06/2021-08/2021** | Tổng hợp kết quả đạt được, công bố kết quả và viết báo cáo nghiệm thu đề tài. | Thiết bị, App, báo cáo đề tài | Trần Tuấn Anh |

**III. CODE CỦA HỆ THỐNG**

* Code của các công cụ chính

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ToolMain : MonoBehaviour

{

private bool dao = false;

private bool Move;

private bool Rotate;

private bool Scale;

private bool information = false;

private bool X;

private bool Y;

private bool Z;

public float MoveSpeed = 50f;

public float Scale1 = 0.1f;

public float RotateSpeed = 50f;

public GameObject Model;

public GameObject Rotate0;

public GameObject Move0;

public GameObject Daochieu0;

public GameObject Daochieu1;

public GameObject thongtin0;

bool click = false;

bool click1 = false;

bool click2 = false;

private void Update()

{

if (dao)

{

if (Move)

{

if (X)

{

Model.transform.Translate(Vector3.up \* Time.deltaTime \* MoveSpeed);

}

if (Y)

{

Model.transform.Translate(Vector3.right \* Time.deltaTime \* MoveSpeed);

}

if (Z)

{

Model.transform.Translate(Vector3.forward \* Time.deltaTime \* MoveSpeed);

}

}

if (Rotate)

{

if (X)

{

Model.transform.Rotate(Vector3.up, Time.deltaTime \* RotateSpeed);

}

if (Y)

{

Model.transform.Rotate(Vector3.right, Time.deltaTime \* RotateSpeed);

}

if (Z)

{

Model.transform.Rotate(Vector3.forward, Time.deltaTime \* RotateSpeed);

}

}

if (Scale)

{

Model.transform.localScale -= new Vector3(Scale1, Scale1, Scale1);

}

}

else

{

if (Move)

{

if (X)

{

Model.transform.Translate(Vector3.down \* Time.deltaTime \* MoveSpeed);

}

if (Y)

{

Model.transform.Translate(Vector3.left \* Time.deltaTime \* MoveSpeed);

}

if (Z)

{

Model.transform.Translate(Vector3.back \* Time.deltaTime \* MoveSpeed);

}

}

if (Rotate)

{

if (X)

{

Model.transform.Rotate(Vector3.down, Time.deltaTime \* RotateSpeed);

}

if (Y)

{

Model.transform.Rotate(Vector3.left, Time.deltaTime \* RotateSpeed);

}

if (Z)

{

Model.transform.Rotate(Vector3.back, Time.deltaTime \* RotateSpeed);

}

}

if (Scale)

{

Model.transform.localScale += new Vector3(Scale1, Scale1, Scale1);

}

}

}

public void OnPressMoveX()

{

X = true;

}

public void OnRealessMoveX()

{

X = false;

}

public void OnPressMoveY()

{

Y = true;

}

public void OnRealessMoveY()

{

Y = false;

}

public void OnPressMoveZ()

{

Z = true;

}

public void OnRealessMoveZ()

{

Z = false;

}

public void OnPressScale()

{

Scale = true;

}

public void OnRealessScale()

{

Scale = false;

}

public void RMChange()

{

if (click == false)

{

click = false;

Move = true;

Rotate = false;

Rotate0.SetActive(false);

Move0.SetActive(true);

}

}

public void MRChange1()

{

if (click == false)

{

click = false;

Rotate = true;

Move = false;

Rotate0.SetActive(true);

Move0.SetActive(false);

}

}

public void daochieuvip0()

{

if (click1 == false)

{

click1 = false;

Daochieu0.SetActive(false);

Daochieu1.SetActive(true);

dao = false;

}

}

public void daochieuvip1()

{

if (click1 == false)

{

click1 = false;

Daochieu0.SetActive(true);

Daochieu1.SetActive(false);

dao = true;

}

}

public void thongtinvip()

{

if (information == false)

{

information = false;

click2 = !click2;

if (click2)

{

thongtin0.SetActive(true);

}

else

{

thongtin0.SetActive(false);

}

}

}

}

* Code tương tác cao (chạm đối tượng)

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class touch2 : MonoBehaviour

{

public GameObject Obj;

public Collider coll;

bool a;

void Start()

{

coll = GetComponent<Collider>();

}

void Update()

{

if (Input.touchCount > 0)

{

Touch touch = Input.GetTouch(0);

RaycastHit hit;

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

if (coll.Raycast(ray, out hit, 100.0f))

{

Debug.Log(hit.transform.name);

a = !a;

Obj.SetActive(a);

}

}

}

}

* Code năng cao chất lượng đầu vào camera

using UnityEngine;

using System.Collections;

using Vuforia;

public class Camera1 : MonoBehaviour

{

private bool mVuforiaStarted = false;

void Start()

{

VuforiaARController vuforia = VuforiaARController.Instance;

if (vuforia != null)

vuforia.RegisterVuforiaStartedCallback(StartAfterVuforia);

}

private void StartAfterVuforia()

{

mVuforiaStarted = true;

SetAutofocus();

}

void OnApplicationPause(bool pause)

{

if (!pause)

{

// App resumed

if (mVuforiaStarted)

{

// App resumed and vuforia already started

// but lets start it again...

SetAutofocus(); // This is done because some android devices lose the auto focus after resume

// this was a bug in vuforia 4 and 5. I haven't checked 6, but the code is harmless anyway

}

}

}

private void SetAutofocus()

{

if (CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode.FOCUS\_MODE\_CONTINUOUSAUTO))

{

Debug.Log("Autofocus set");

}

else

{

// never actually seen a device that doesn't support this, but just in case

Debug.Log("this device doesn't support auto focus");

}

}

}

* Code chạm đối tượng dùng RaycastHit

using UnityEngine;

using System.Collections;

public class touch2 : MonoBehaviour

{

public GameObject Obj;

public Collider coll;

bool a;

void Start()

{

coll = GetComponent<Collider>();

}

void Update()

{

if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began)

{

Touch touch = Input.GetTouch(0);

RaycastHit hit;

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

if (coll.Raycast(ray, out hit, 100.0f))

{

Debug.Log(hit.transform.name);

a = !a;

Obj.SetActive(a);

}

}

}

}

***Link tải ứng dụng***

