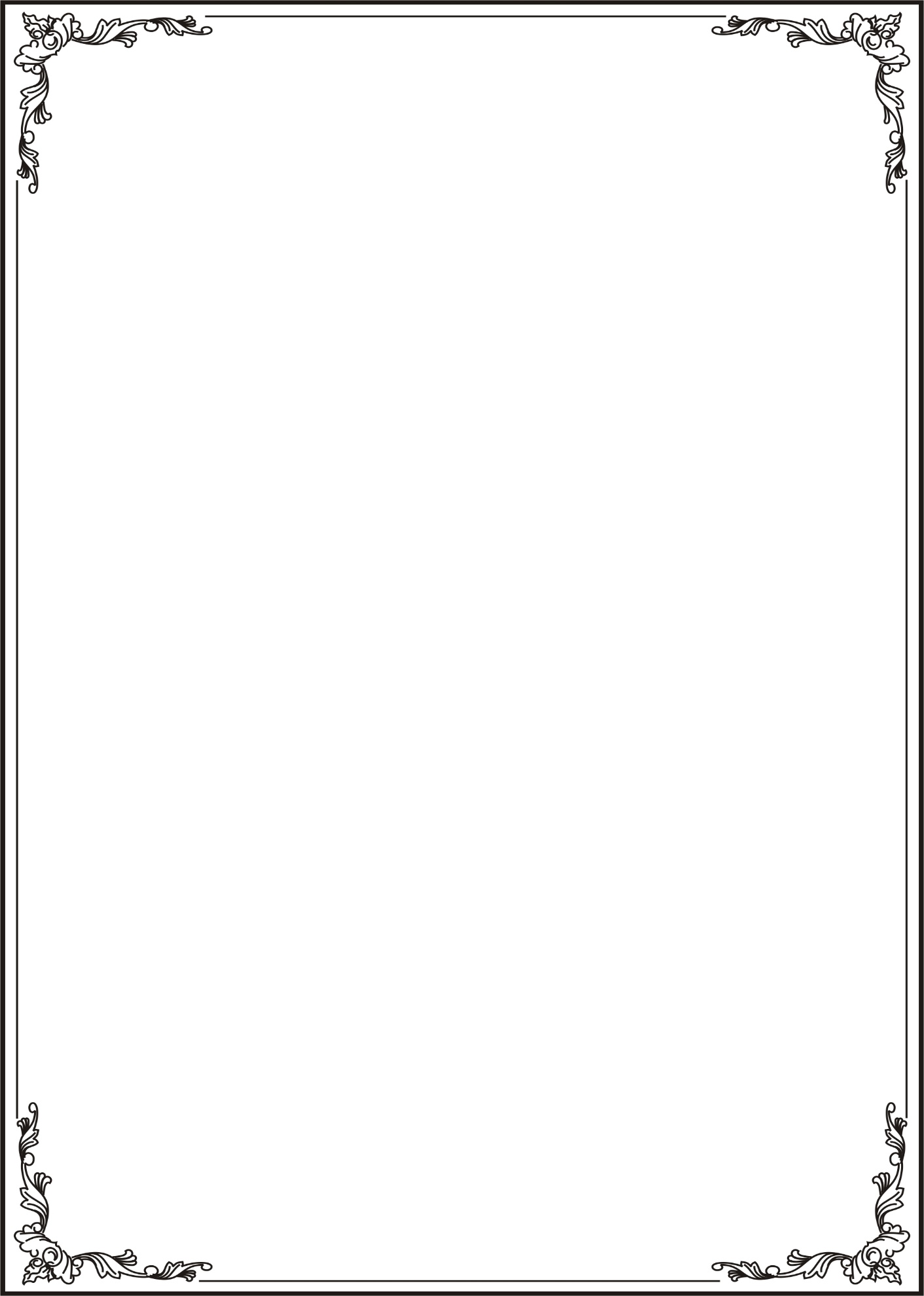
**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙞🙜🕮🙞🙜-----**



**BÀI TẬP LỚN**

**MÔN HỌC: ĐỒ HỌA VÀ HIỆN THỰC ẢO**

**ĐỀ TÀI: MÔ PHỎNG NỘI THẤT VÀ KIẾN TRÚC PHÒNG HỌC CÁ NHÂN**

**Giảng viên hướng dẫn:** ThS. Hà Mạnh Toàn

**Sinh viên thực hiện: Nhóm 6**

Nguyễn Khắc Huân - 2055010106

Lê Đức Anh - 2055010110

Hà Nội, ngày 7 tháng 11 năm 2023

Mục lục

[**Lời mở đầu** 4](#_Toc150163832)

[**I. Giới thiệu** 4](#_Toc150163833)

[**1.1. Giới thiệu về môn học và công cụ** 4](#_Toc150163834)

[1.1.1. Giới thiệu về đồ họa và hiện thực ảo 4](#_Toc150163835)

[1.1.1.1. Đồ họa (Graphics) 4](#_Toc150163836)

[1.1.1.2. Hiện thực ảo 4](#_Toc150163837)

[1.1.2. Giới thiệu về Untiy 5](#_Toc150163838)

[1.1.2.1. Unity là gì? 5](#_Toc150163839)

[1.1.2.2. Lịch sử ra đời và phát triển 5](#_Toc150163840)

[1.1.2.3. Ưu điểm Unity 5](#_Toc150163841)

[1.1.3. Giới thiệu về Blender 6](#_Toc150163842)

[1.1.3.1. Blender là gì? 6](#_Toc150163843)

[1.1.3.2. Lịch sử ra đời và phát triển 6](#_Toc150163844)

[1.1.3.3 Ưu điểm của Blender 6](#_Toc150163845)

[1.1.4. Giới thiệu về Visual studio 2022 7](#_Toc150163846)

[**II. Phân tích và thiết kế** 9](#_Toc150163847)

[**2.1 Mô hình hóa** 9](#_Toc150163848)

[2.1.1. Thông tin không gian dựng 9](#_Toc150163849)

[2.1.1.1. Bản vẽ, đo đạc 9](#_Toc150163850)

[2.1.1.2. Các hiện vật cần dựng, hình ảnh và mô tả kích thước 9](#_Toc150163851)

[**2.2. Điều khiển** 13](#_Toc150163852)

[2.2.1. Điều khiển mở cửa 13](#_Toc150163853)

[2.2.1.1. Code 13](#_Toc150163854)

[2.2.1.2. Cách triển khai 14](#_Toc150163855)

[2.2.2. Bật, tắt đèn 15](#_Toc150163856)

[2.2.2.1. Code 15](#_Toc150163857)

[2.2.2.2. Cách triển khai 16](#_Toc150163858)

[2.2.3. Camera 16](#_Toc150163859)

[2.2.3.1. Code 16](#_Toc150163860)

[2.2.3.2. Cách triển khai 17](#_Toc150163861)

[**III. Dựng mô hình phòng học cá nhân** 18](#_Toc150163862)

[**3.1. Phòng học cá nhân** 18](#_Toc150163863)

[**3.1.1. Ảnh thật** 18](#_Toc150163864)

[3.1.2 Mô hình 19](#_Toc150163865)

[**3.2. Bàn ghế** 19](#_Toc150163866)

[3.2.1. Mô hình 19](#_Toc150163867)

[3.2.2. Ảnh thật 20](#_Toc150163868)

[**3.3. Cửa sổ** 21](#_Toc150163869)

[3.3.1. Ảnh thật 21](#_Toc150163870)

[3.3.2. Mô hình 22](#_Toc150163871)

[**3.4. Giường** 22](#_Toc150163872)

[3.4.1. Ảnh thật 22](#_Toc150163873)

[3.4.2. Mô hình 23](#_Toc150163874)

[**3.5. Cửa vào** 23](#_Toc150163875)

[3.5.1. Ảnh thật 23](#_Toc150163876)

[3.5.2. Mô hình 24](#_Toc150163877)

[**3.6. Đèn** 24](#_Toc150163878)

[3.6.1. Ảnh thật 24](#_Toc150163879)

[3.6.2 Mô hình 24](#_Toc150163880)

[**3.7. Gạch ốp và sàn nhà** 25](#_Toc150163881)

[3.7.1. Ảnh thật 25](#_Toc150163882)

[3.7.2. Mô hình 25](#_Toc150163883)

[**3.8. Minh họa điều khiển** 26](#_Toc150163884)

[3.8.1. Mở cửa 26](#_Toc150163885)

[3.8.2. Bật đèn 27](#_Toc150163886)

[**IV. Kết luận và hướng phát triển** 28](#_Toc150163887)

[**4.1. Kết luận và hướng phát triển** 28](#_Toc150163888)

[**4.2. Danh mục tài liệu tham khảo** 28](#_Toc150163889)

**Lời mở đầu**

Trong thời đại ngày nay, chúng ta đang đứng trước cơ hội học tập đầy hứng thú và đổi mới – phòng học cá nhân dựa trên công nghệ Unity VR. Hôm nay, chúng ta sẽ cùng nhau khám phá một thế giới mà không chỉ có sách giáo trình và bảng đen, mà còn có môi trường ảo đa dạng và tương tác.

Dự án phòng học cá nhân bằng Unity VR là một dự án không chỉ đánh thức sự tò mò, mà còn thúc đẩy sáng tạo, khuyến khích mọi người tạo một phòng học cá nhân một cách độc đáo và thú vị.

Trong báo cáo này, chúng tôi xin chia sẻ với bạn đọc về quá trình phát triển dự án, từ việc nghiên cứu và thu thập dữ liệu cho đến thiết kế và triển khai ứng dụng phòng học cá nhân trên Unity VR.

Mục tiêu chính của dự án là mang đến cho khán giả một trải nghiệm thú vị và sâu sắc trong việc tương tác thông qua công nghệ VR. Chúng tôi hy vọng rằng dự án của chúng tôi sẽ giúp các bạn học sinh theo đuổi học tập và đam mê, khám phá những khía cạnh mới, và định hình môi trường học tập của riêng mình.

# **I. Giới thiệu**

## **1.1. Giới thiệu về môn học và công cụ**

### 1.1.1. Giới thiệu về đồ họa và hiện thực ảo

### 1.1.1.1. Đồ họa (Graphics)

- Là một lĩnh vực của công nghệ thông tin liên quan đến việc tạo ra, xử lý, và hiển thị hình ảnh và video trên máy tính hoặc các thiết bị điện tử. Đồ họa máy tính được sử dụng rộng rãi trong nhiều ứng dụng, bao gồm thiết kế đồ họa, phát triển trò chơi video, tạo phim hoạt hình và hiệu ứng đặc biệt trong phim, thiết kế sản phẩm, quảng cáo và nhiều lĩnh vực khác.

### 1.1.1.2. Hiện thực ảo

- Hiện thực ảo (Virtual Reality – VR) là một công nghệ giải trí và tương tác mà tạo ra một môi trường ảo hoàn toàn hoặc một cảm giác rất giống thực tế cho người dùng. Thông qua sử dụng thiết bị VR như tai nghe, gọng kính hoặc thiết bị đeo trên đầu, người dùng có thể đắm chìm hoàn toàn vào một thế giới tạo ra bởi máy tính và tương tác với nó bằng cách di chuyển và sử dụng các thiết bị điều khiển.

### 1.1.2. Giới thiệu về Untiy

### 1.1.2.1. Unity là gì?

**-** Unity là một game engine đa nền tảng được phát triển bởi Unity Technologies và được giới thiệu lần đầu vào năm 2005. Phần mềm này chủ yếu được sản xuất các video game trên 27 nền tảng, bao gồm: Game consoles, máy tính hay điện thoại,…

### 1.1.2.2. Lịch sử ra đời và phát triển

**-** Lịch sử ra đời:

* Unity được phát triển bời Unity technologies, một công ty có trụ sở tại San Francisco, California và được thành lập bời David Helgason, Nicholas Francis và Joachim Ante vào năm 2004.
* Unity lần đầu tiên được giới thiệu vào năm 2005 dưới phiên bản 1.0. Ban đầu, Unity được phát triển như một công cụ cho các dự án trò chơi nhỏ, nhưng sau đó đã mở rộng để hỗ trợ phát triển ứng dụng và trò chơi trên nhiều nền tảng.

### 1.1.2.3. Ưu điểm Unity

**-** Ưu điểm của Unity

* Đa nền tảng: Unity cho phép bạn phát triển ứng dụng và trò chơi một lần và triển khai chúng trên nhiều nền tảng khác nhau, bao gồm iOS, Android, PC, Mac, console trò chơi, VR, AR và nhiều nền tảng khác. Điều này giúp tiết kiệm thời gian và nguồn lực phát triển.
* Hỗ trợ Đồ họa 2D và 3D: Unity hỗ trợ phát triển cả ứng dụng đồ họa 2D và 3D. Bạn có thể tạo cả trò chơi 2D nhỏ nhẹ và thế giới 3D phức tạp.
* Lập trình dễ dàng: Unity sử dụng ngôn ngữ lập trình C#, một ngôn ngữ phổ biến và dễ học, giúp các nhà phát triển nhanh chóng làm quen với môi trường phát triển.
* Cộng đồng lớn: Unity có một cộng đồng phát triển rộng lớn với nhiều tài liệu, tài nguyên và hỗ trợ từ cộng đồng, giúp bạn giải quyết vấn đề và học hỏi.
* Unity Asset Store: Unity Asset Store cung cấp một cửa hàng trực tuyến nơi bạn có thể tìm mua và bán các tài nguyên, từ mã nguồn mở đến mô hình 3D, giúp tăng tốc quá trình phát triển.

### 1.1.3. Giới thiệu về Blender

### 1.1.3.1. Blender là gì?

- Blender là một phần mềm mã nguồn mở (open source) được sử dụng để tạo và chỉnh sửa đồ họa 3D, hiệu ứng đặc biệt, phim hoạt hình và nhiều ứng dụng liên qua đến đồ họa 3D. Nó cung cấp một loạt các công cụ và tính năng mạnh mẽ để tạo ra nội dung đồ họa 3D chất lượng cao.

### 1.1.3.2. Lịch sử ra đời và phát triển

**-** Lịch sử ra đời:

* Blender được tạo ra bởi Ton Roosendaal, một nhà phát triển phần mềm Hà Lan, và phiên bản đầu tiên của Blender (phiên bản 1.0) được giới thiệu vào năm 1998.
* Trước khi có blender, Ton Roosendaal đã phát triển một phần mềm khác gọi là “Traces” cho việc ra đồ họa 3D. Tuy nhiên, sau khi phát triển kết thúc, Roosendaal quyết định tạo ra một phần mềm mới và tự do mã nguồn mở, dẫn đến việc ra đời của Blender.

### 1.1.3.3 Ưu điểm của Blender

- Mã nguồn mở (Open source): Blender là một phần mềm mã nguồn mở hoàn toàn miễn phí. Điều này có nghĩa là bất kỳ ai cũng có thể tải xuống, sử dụng, thay đổi và phân phối phiên bản của Blender theo nhu cầu của họ mà không cần thiết trả tiền.

- Công cụ đa dạng: Blender cung cấp một loạt công cụ đa dạng cho mô hình hóa, chế tác, kết xuất, hiệu ứng đặc biệt, hoạt hình và nhiều khía cạnh khác của đồ họa 3D. Bạn có thể tạo ra các dự án phức tạp mà không cần sử dụng nhiều phần mềm khác.

- Hỗ trợ đồ họa 2D và 3D: Blender không chỉ hỗ trợ đồ họa 3D mà còn cho phép bạn tạo và chỉnh sửa đồ họa 2D. Điều này làm cho nó phù hợp cho nhiều loại dự án đồ họa.

- Hiệu suất và hiệu ứng đặc biệt: Blender cung cấp nhiều công cụ để tọa ra hiệu ứng đặc biệt và đặc điểm đồ họa như ánh sáng, vật lý, phân loại và thậm chí cả phát triển trò chơi. Nó cũng có một hệ thống kết xuất mạnh mẽ cho việc tạo hình ảnh chất lượng cao.

- Cộng đồng lớn: Blender có một cộng đồng người dùng và nhà phát triển nhiệt tình và phong phú. Bạn có thể tìm kiếm hỗ trợ, tài liệu và hướng dẫn từ cộng đồng, giúp giải quyết vấn đề và học hỏi.

- Hỗ trợ đa nền tảng: Blender có phiên bản cho nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux. Điều này giúp người sử dụng Blender trên nhiều nền tảng khác nhau.

### 1.1.4. Giới thiệu về Visual studio 2022

**-** Visual Studio 2022 là một phiên bản mới của môi trường phát triển tích hợp (IDE) phổ biến của Microsoft, được thiết kế để hỗ trợ phát triển ứng dụng phức tạp trên nhiều nền tảng. Đây là một số điểm quan trọng về Visual Studio 2022:

* Hỗ trợ đa nền tảng: Visual Studio 2022 cho phép phát triển ứng dụng trên nhiều nền tảng, bao gồm Windows, macOS, Linux và nhiều nền tảng web. Điều này giúp bạn tạo ứng dụng có khả năng chạy trên nhiều thiết bị và hệ điều hành.
* Mã nguồn mở và công cụ mở rộng: Visual Studio 2022 có khả năng mở rộng thông qua các tiện ích mở rộng (extensions) và hỗ trợ cho nhiều ngôn ngữ lập trình và khung làm việc. Nó cung cấp hỗ trợ mã nguồn mở, cho phép bạn sử dụng các thư viện và công cụ mã nguồn mở để phát triển ứng dụng.
* Hiệu suất tối ưu hóa: Phiên bản mới này của Visual Studio được tối ưu hóa về hiệu suất, giúp giảm thời gian biên dịch và chạy ứng dụng. Điều này cung cấp trải nghiệm phát triển nhanh hơn và tốt hơn.
* Hỗ trợ ngôn ngữ lập trình đa dạng: Visual Studio 2022 hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C++, C#, Visual Basic, Python, JavaScript và nhiều ngôn ngữ khác.
* Cộng đồng mạnh mẽ: Visual Studio có một cộng đồng phát triển lớn, cung cấp nhiều tài liệu, hướng dẫn và tiện ích mở rộng từ cộng đồng, giúp bạn giải quyết vấn đề nâng cao kỹ năng của mình.

**1.1.5. Giới thiệu về C#**

**-** C# là một ngôn ngữ lập trình được phát triển bởi Microsoft vào cuối những năm 1990. Nó là một trong những ngôn ngữ lập trinh phổ biến và mạnh mẽ được sử dụng rộng rãi trong phát triển ứng dụng và phần mềm

* Ngôn ngữ lập trình ứng dụng đa năng: C# là một ngôn ngữ đa dạng và đa năng, phù hợp cho việc phát triển nhiều loại ứng dụng, bao gồm desktop, ứng dụng di động, ứng dụng web, ứng dụng trò chơi, ứng dụng IoT và nhiều loại ứng dụng khác.
* Phát triển trên nền tảng Microsoft: C# là một phần của nền tảng phát triển Microsoft, vì vậy nó được sử dụng phổ biến trong phát triển ứng dụng Windows.
* Thuận tiện và đơn giản: C# có cú pháp rõ ràng và dễ đọc, làm cho việc phát triển và bảo trì mã nguồn mở trở lên dễ dàng. Nó cũng cung cấp nhiều tính năng thuận tiện như quản lý bộ nhớ tự động (garbage collection) và hỗ trợ của Visual Studio, một môi trường phát triển tích hợp mạnh mẽ.
* Hỗ trợ đa nền tảng: Với sự phát triển của .NET Core và sau này là .NET 5 và .NET 6, C# đã mở rộng khả năng hỗ trợ nền tảng, cho phép phát triển ứng dụng chạy trên nhiều hệ điều hành và môi trường.
* Cộng đồng lớn và thư viện mở rộng: C# có một cộng đồng lập trình đông đảo và đa đạng, cung cấp nhiều thư viện và khung làm việc mở rộng.

# **II. Phân tích và thiết kế**

## **A black and white drawing of a piano Description automatically generated2.1 Mô hình hóa**

### 2.1.1. Thông tin không gian dựng

### 2.1.1.1. Bản vẽ, đo đạc

*Hình 1. Bản vẽ phòng học*

### 

### 2.1.1.2. Các hiện vật cần dựng, hình ảnh và mô tả kích thước

Sau khi khảo sát chúng em có kích thước những đồ vật trong phòng như sau

A door with a sign and arrows

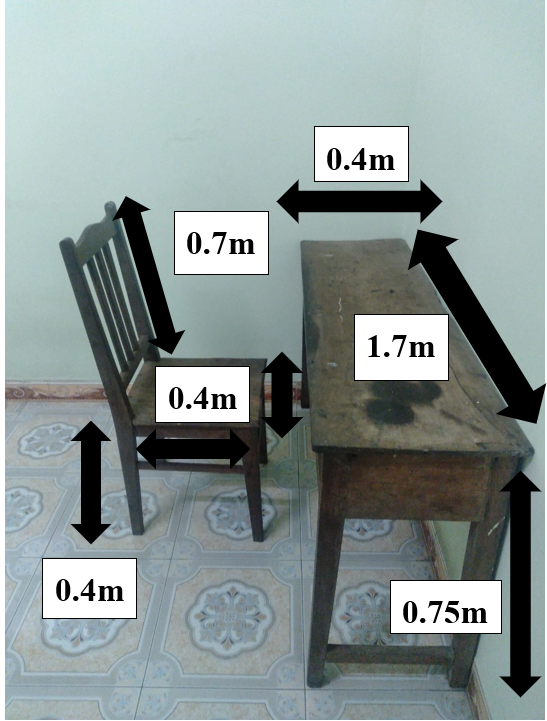
Description automatically generated with medium confidence

Kích thước của cửa

A window with a bar and a window with a window

Description automatically generated with medium confidence

Kích thước của cửa sổ



Kích thước của bàn ghế

A bed with a mattress and a mat

Description automatically generated with medium confidence

Kích thước của giường

- Cách dựng:

* Thực hành trên phần mềm Blender
* Trong project này, Một số đồ vật như điều hòa và camera là chúng em lấy ở Asset Store
* Các vật thể còn lại em dùng các khồi Cube (Khồi hộp) để dựng bằng cách thay đổi các kích thước của 3 chiều x, y, z sau đó thay đổi vị trí của vật thể vừa tạo rồi liên kết vật thể khác để tạo thành ra 1 vật thể hoàn chỉnh.

## **2.2. Điều khiển**

### 2.2.1. Điều khiển mở cửa

### 2.2.1.1. Code

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MoveDoor : MonoBehaviour

{

public float maxAngle = 90f;

public float speed = 5f;

private bool isOpen = false;

private float initialAngle;

void Start()

{

initialAngle = transform.eulerAngles.y;

}

void Update()

{

if (Input.GetKeyDown(KeyCode.F))

{

isOpen = !isOpen;

}

if (isOpen)

{

float targetAngle = initialAngle - maxAngle;

float newAngle = Mathf.LerpAngle(transform.eulerAngles.y, targetAngle, speed \* Time.deltaTime);

transform.eulerAngles = new Vector3(transform.eulerAngles.x, newAngle, transform.eulerAngles.z);

}

else

{

float newAngle = Mathf.LerpAngle(transform.eulerAngles.y, initialAngle, speed \* Time.deltaTime);

transform.eulerAngles = new Vector3(transform.eulerAngles.x, newAngle, transform.eulerAngles.z);

}

}

}

### 2.2.1.2. Cách triển khai

- Trước tiên chúng em khai báo biến maxAngle, speed, isOpen, initalAngle

* Biến maxAngle: Xác định góc quay tối đa mà cánh cửa có thể mở
* Biến speed: khai báo tốc độ mở của cửa
* Biến isOpen: kiểm tra trạng thái mở hoặc đóng của cửa
* Biến intitalAngle: Lưu góc ban đầu của cửa

- Phương thức void Start(): góc ban đầu của cánh cửa được thiết lập bằng góc hiện tại của đối tượng.

- Phương thức void Update(): Thực hiện logic cho việc đóng mở cửa

### 2.2.2. Bật, tắt đèn

### 2.2.2.1. Code

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class OpenLight : MonoBehaviour

{

private Light myLight;

// Start is called before the first frame update.

void Start()

{

// Lấy Light component trên đối tượng mà script này được gắn vào.

myLight = GetComponent<Light>();

}

// Update is called once per frame.

void Update()

{

// Kiểm tra nếu người dùng nhấn phím B.

if (Input.GetKeyDown(KeyCode.B))

{

// Đảm bảo rằng myLight không null.

if (myLight != null)

{

// Chuyển đổi trạng thái bật/tắt của đèn.

myLight.enabled = !myLight.enabled;

}

}

}

}

### 2.2.2.2. Cách triển khai

- Trước tiên khởi tạo một biến là myLight với kiểu dữ liệu là Light

- Hàm void Start(): Tìm kiếm và lưu thành phần ‘Light’ của đối tượng vào biến ‘myLight’.

- Hàm void Update(): kiểm tra xem người dùng có nhấn ‘B’ từ bàn phím hay ko. Nếu có thì kiểm tra ‘myLight’ đã được gán cho đối tượng ‘Light’ hay ko. Nếu ‘myLight’ khác null (tức là nó được gán thành công) tức là đèn đang bật sẽ tắt và ngược lại.

### 2.2.3. Camera

### 2.2.3.1. Code

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CamMove : MonoBehaviour

{

public float moveSpeed = 0.009f;

public float turnSpeed = 0.003f;

private float alpha = 0f;

void Start()

{

}

void Update()

{

// Khởi tạo một Vector3 mới với giá trị ban đầu là zero.

Vector3 v = Vector3.zero;

if (Input.GetKey(KeyCode.W))

{

v.z += moveSpeed;

}

if (Input.GetKey(KeyCode.S))

{

v.z -= moveSpeed;

}

if (Input.GetKey(KeyCode.A))

{

alpha += turnSpeed;

}

if (Input.GetKey(KeyCode.D))

{

alpha -= turnSpeed;

}

// Lấy giá trị đầu vào của trục Y từ chuột và nhân với 0.25.

float mouseY = Input.GetAxis("Mouse Y") \* 0.25f;

v.y += mouseY; // Cộng giá trị đó vào trục y của vector v.

// Di chuyển camera theo vector v mà không thay đổi vị trí của nó trong không gian.

transform.Translate(v);

// Đặt hướng nhìn của camera theo góc alpha, giữ nguyên trên trục y và z.

transform.forward = new Vector3(Mathf.Cos(alpha), 0, Mathf.Sin(alpha));

}

}

### 2.2.3.2. Cách triển khai

- Khai báo một biến kiểu float 'alpha' để lưu góc quay và khởi tạo nó là 0.

- Hàm void Update()

* Khởi tạo Vector3 ‘v’: Một vector mới được tạo ra để xác định và vận tốc di chuyển của đối tượng.
* Di chuyển theo trục Z: Nếu người dùng nhấn phím W, camera sẽ di chuyển lên phía trước (+z). Nếu nhấn phím S, camera sẽ di chuyển lùi lại (-z).
* Quay camera (xoay alpha): Khi nhấn phím A, camera sẽ quay sang trái (+alpha). Nếu nhấn phím D, camera sẽ quay sang phải (-alpha).
* Điều chỉnh hướng di chuyển của chuột: Lấy độ nghiêng chuột bên trục Y (lên xuống) và thêm vào hướng di chuyển của camera.
* Áp dụng di chuyển: Sử dụng hàm Transform.Translate() di chuyển camera theo vector ‘v’ đã được xác định.
* Đặt hướng nhìn: Hàm cuối cùng sử dụng giá trị ‘alpha’ để thiết lập hướng nhìn mới của đối tượng thông qua thuộc tính ‘transform.forward’, làm camera hướng về góc ‘alpha’ đang chỉ đến.

# 

# **III. Dựng mô hình phòng học cá nhân**

**3.1. Phòng học cá nhân**

**3.1.1. Ảnh thật**



### 3.1.2 Mô hình

**A room with a window and a desk

Description automatically generated with medium confidence**

## **3.2. Bàn ghế**

### 3.2.1. Mô hình

****

### 3.2.2. Ảnh thật

A chair and desk in a room

Description automatically generated

## **3.3. Cửa sổ**

### 3.3.1. Ảnh thật

****

### 3.3.2. Mô hình

A window with a view of the sky

Description automatically generated

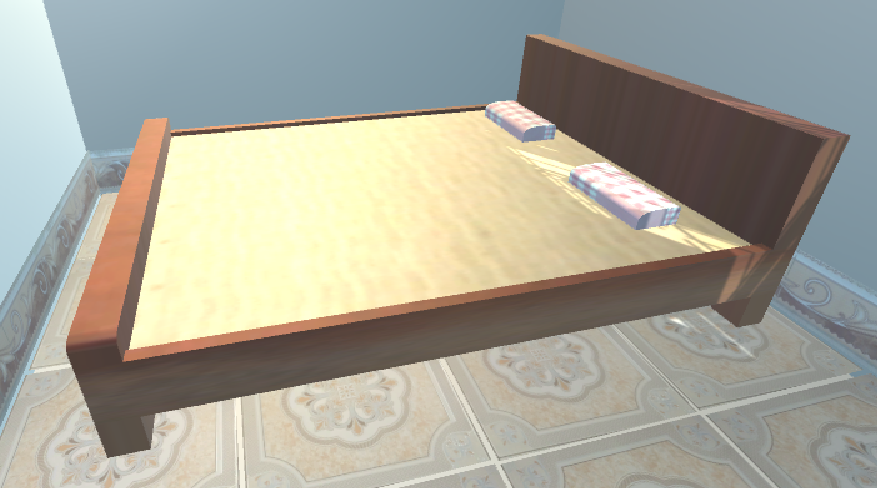
## **3.4. Giường**

### 3.4.1. Ảnh thật

A bed with a pink pillow on it

Description automatically generated

### 3.4.2. Mô hình



## **3.5. Cửa vào**

### 3.5.1. Ảnh thật

A door in a room

Description automatically generated

### 3.5.2. Mô hình

A door with a black frame

Description automatically generated

## **3.6. Đèn**

### 3.6.1. Ảnh thật

A light on a wall

Description automatically generated

### 3.6.2 Mô hình

A blurry image of a white rectangular object

Description automatically generated

## **3.7. Gạch ốp và sàn nhà**

### 3.7.1. Ảnh thật



### 3.7.2. Mô hình

A room with a tile floor and a window

Description automatically generated

## **3.8. Minh họa điều khiển**

### 3.8.1. Mở cửa

Ảnh cửa lúc chưa mở

A door with a black frame

Description automatically generated

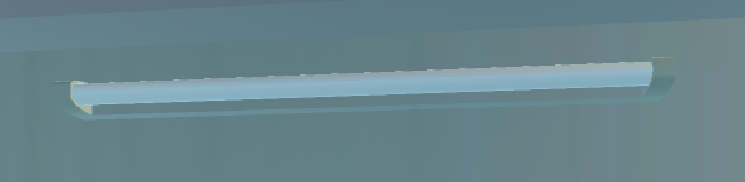
Ảnh lúc mở cửa

A door with a white frame

Description automatically generated with medium confidence

### 3.8.2. Bật đèn

Ảnh lúc chưa bật đèn



Ảnh sau khi bật đèn

A white background with blue lines

Description automatically generated

# **IV. Kết luận và hướng phát triển**

## **4.1. Kết luận và hướng phát triển**

- Trong quá trình học tập và nghiên cứu đề tài phòng học cá nhân, chúng em nhận thấy bản thân còn nhiều điểm hạn chế và thiếu sót. Dựa vào những gì mà nhóm chúng em đã làm thì em đánh giá là sản phẩm của nhóm em giống khoảng 80%.

- Hướng phát triển:

* Chỉnh sửa đồ vật sao cho giống thật nhất với từng chi tiết.
* Thêm một số chức năng cho dự án.
* Thêm một vài chi tiết hoặc vật ở bên ngoài cho sinh động

Phân công nhiệm vụ

|  |  |  |
| --- | --- | --- |
| Tên | Công việc | Tỷ lệ phần trăm |
| Nguyễn Khắc Huân | Tinh chỉnh mô hình phòng và đồ vật hoàn chỉnh từ Blender sang Unity, code. | 55% |
| Lê Đức Anh | Dựng phần thô của phòng và đồ vật từ các khối Cube trong Blender, làm word và powerpoint | 45% |

## **4.2. Danh mục tài liệu tham khảo**

- <https://www.youtube.com/@BlenderVNTraining>

- <https://chat.openai.com/>

- **https://assetstore.unity.com**