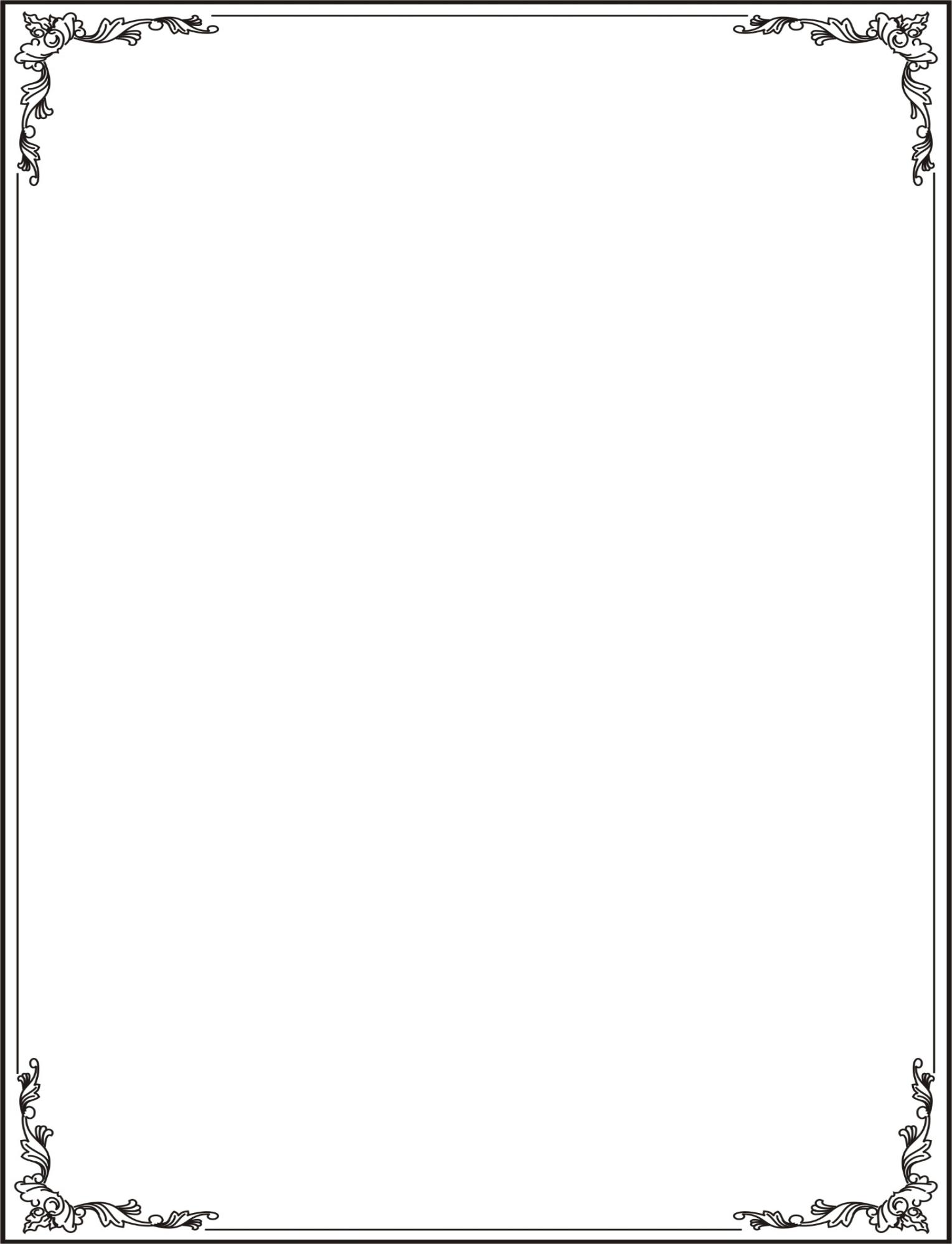
****

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**

**~~~~~~\*~~~~~**

****

**BÁO CÁO ĐỒ ÁN 2**

**ĐỀ TÀI**

Lập trình game vượt chướng ngại vật 3D bằng Unity

**Sinh viên thực hiện 1: Châu Văn Tấn**

**Class: 21TIN01-TT**

**Mssv : 214458**

**Sinh viên thực hiện 2: Trương Duy Tân**

**Class: 21TIN01-TT**

**Mssv : 214234**

**Giảng Viên Hướng Dẫn: Trần Văn Thiện**

**Cần Thơ, 02 Tháng 12 năm 2024**

*Đồ án cơ sở ngành - Trường đại Nam Cần Thơ*

**TRƯỜNG ĐH NAM CẦN THƠ CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Công nghệ thông tin Độc lập - Tự do -H ạnh phúc**

****

ĐỀ TÀI

ĐỒ ÁN 2 CƠ SỞ NGÀNH

**Ngày giao đề: 23/09/2024**

**Ngày nộp: 02/12/2024**

1. **Họ và tên sinh viên/ nhóm sinh viên:**

- Châu Văn Tấn MSSV: 214458 Lớp: 21Tin01-TT

- Trương Duy Tân MSSV:214234 Lớp: 21Tin01-TT

1. **Tên đề tài:**

Lập trình game vượt chướng ngại vật 3D với Unity và sử dụng Visual studio 2022

1. **Nội dung nhiệm vụ:**

- Tìm hiểu về ứng dụng làm game Unity3D, các phần mềm hỗ trợ để lập trình một game hoàn chỉnh

- Tìm hiểu sâu hơn về ngôn ngữ C# trên nền tảng Unity engine trong công nghệ làm game.

- Thiết kế ra game bắn súng 3D góc nhìn thứ nhất

1. **Kết quả tối thiểu phải có:**

- Thiết kế một sản phẩm game 3D hoàn chỉnh

- Trò chơi giao diện đơn giản, trực quan và dễ sử dụng

- Mô tả chi tiết về quy trình thiết kế và phát triển trò chơi, bao gồm các tài liệu hướng dẫn cài đặt và sử dụng.

- Báo cáo đánh giá về trải nghiệm của người dùng và những tiến trình có thể thực hiện trong tương lai

1. **Dự kiến kinh phí (Nếu có):**

….………………………………………………….…………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

….……………………………………………………………………………………

**LỜI NHÂN XÉT**

Nhận xét và đánh giá điểm của người hướng dẩn:

**-Ý thức thực hiện:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Nội dung thực hiện:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Hình thức trình bày:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

**-Tổng hợp kết quả:**

………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..………………………………………………………………………………………………………………..…………………………………………………………………………………………………………

Cần Thơ, ngày 02 tháng 12 năm 2024

Người hướng dẫn

*(Kí và ghi rõ họ tên)*

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Nam Cần Thơ đã tạo điều kiện tốt cho chúng em thực hiện đề tài này!

Cùng với sự giúp đỡ hướng dẫn tận tính của Thạc sĩ – Giảng viên – thầy Trần Văn Thiện chúng em đã phát triển nên Game Vượt Chướng Ngại Vật 3D phiên bản tương tự như game Subway Surfers. Hứa hẹn sẽ mang đến trải nghiệm mới lạ, giúp người chơi thư giãn, giải trí sau những giờ học tập lao động căng thẳng. Game còn là công cụ rèn luyện kỹ năng phản xạ, tính tập trung, khả năng tính toán suy nghĩ để có thể hoàn thành thử thách khó nhất của trò chơi.

Do thời gian còn hạn hẹp cũng như chưa có kinh nghiệm làm một game lớn, nền tảng Framework mới xuất hiện trong những năm gần đây, chương trình thành phẩm của chúng em có thể còn tồn tại lỗi không mong muốn. Rất mong nhận được góp ý của thầy cô để chúng em hoàn thiện hơn và rút kinh nghiệm cho những đồ án, dự án lớn hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn thầy cô và chúc thầy cô thật nhiều sức khỏe!

**MỤC LỤC**

[CHƯƠNG 1 TỔNG QUAN 1](#_Toc533551257)

[1.1. Lý do chọn đề tài 1](#_Toc533551258)

[1.2. Đối tượng nghiên cứu 1](#_Toc533551259)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 2](#_Toc533551260)

[2.1. Tổng quan về Unity Hub 2](#_Toc533551261)

[2.2. Các khái niệm trong Unity 4](#_Toc533551265)

[2.3. Cửa sổ làm việc 8](#_Toc533551270)

[2.4. Các phần mềm hỗ trợ khác 12](#_Toc533551271)

[2.5. Game và hướng phát triển 13](#_Toc533551274)

[CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME 15](#_Toc533551275)

[3.1. Ý tưởng 15](#_Toc533551276)

[3.2. Tạo bản đồ và nhân vật 16](#_Toc533551277)

[3.3. Xử lý di chuyển, va chạm, điểm số, kết thúc màn chơi 19](#_Toc533551280)

[3.4. Tiến trình của Game 20](#_Toc533551281)

[3.5. Hoạt họa và cảm nhận 21](#_Toc533551282)

[CHƯƠNG 4 CHƯƠNG TRÌNH DEMO 22](#_Toc533551283)

[4.1. Gameplay 22](#_Toc533551285)

[4.2. Các chức năng cơ bản 22](#_Toc533551286)

[CHƯƠNG 5 KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 24](#_Toc533551288)

[5.1 Kết luận 24](#_Toc533551289)

[5.2. Định hướng phát triển 24](#_Toc533551290)

[CHƯƠNG 6 TÀI LIỆU THAM KHẢO – PHỤ LỤC 25](#_Toc533551291)

[6.1. Tài liệu tham khảo 25](#_Toc533551292)

[6.2. Phụ lục 25](#_Toc533551293)

# CHƯƠNG 1 TỔNG QUAN

* 1. **Lý do chọn đề tài**

Xã hội ngày càng phát triển, nhu cầu giải trí của con người ngày càng tăng cao. Cùng với sự phát triển của công nghệ thông tin, điện tử, các thiết bị chơi game ngày càng phổ biến. Trong những năm gần đây ngành công nghệ game nổi lên và đem lại doanh thu khổng lồ. Có thể nói game không chỉ là một phương tiện giải trí cơ bản, nó còn là một lĩnh vực đáng quan tâm và theo đuổi. Thiết kế game là một công việc không dễ dàng đòi hỏi rất nhiều thời gian và công sức, tuy nhiên quá đó ta có thể biến những dòng code khô khan thành những trải nghiệm thú vị cho người chơi, đồng thời có thể học hỏi những kiến thức mới về nền tảng Unity Framework và củng cố ngôn ngữ C#

* 1. **Đối tượng nghiên cứu**

**1.2.1 Nền tảng Unity**

· Tính năng nổi bật

· Hỗ trợ đa nền tảng

· Thư viện và Asset Store

· Công cụ hỗ trợ

· Phát triển lối chơi vượt rào cản

· Ưu điểm khi sử dụng Unity

**1.2.2** **Người chơi (Player)**

· Đặc điểm: Lứa tuổi, sở thích, kỹ năng.

· Hành vi: Phản ứng cản trở, cách xử lý vấn đề.

**1.2.3 **Cơ chế gameplay****

· Hệ thống điều khiển: Chạm, vuốt, nhấn phím.

· Chướng vật: Di động (hàng rào, vật cản).

· Tăng dần khó khăn: Tốc độ vẫn giữ nguyên, nhưng cản trở phức tạp hơn.

**1.2.4 **Bối cảnh và thiết kế****

· Đồ họa 3D: Màu sắc, ánh sáng, môi trường chi tiết.

· Chủ đề: Trên con đường, thành phố, với bối cảnh bầu trời.

· Âm thanh: Hiệu ứng, nhạc nền thích hoạt động.

**1.2.5 **Challenge và mục tiêu****

· Thu thập vật phẩm: Xu.

**1.2.6 **Tương tác xã hội****

· Kết nối bạn bè: Chia sẻ thành tích.

**1.2.7 **Công nghệ****

· Công cụ phát triển: Unity, Unreal Engine.

· Tối ưu hóa: Hiệu suất trên các thiết bị khác nhau.

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

* 1. **Tổng quan về Unity engine**
     1. **Unity là gì?**

****

* **Phát hành lần đầu** : Năm 2005 bởi Unity Technologies.
* **Loại phần mềm** : Công cụ phát triển trò chơi (Game Engine).
* **Trình lập ngôn ngữ** : Chủ yếu sử dụng **C#** .
* **Hỗ trợ nền tảng đa phương** : Unity cho phép xuất bản trò chơi và ứng dụng trên 25 nền tảng, bao gồm Windows, macOS, Android, iOS, PlayStation, Xbox, Nintendo Switch và trình duyệt web.
* **Hỗ trợ 2D và 3D** : Unity cung cấp các công cụ mạnh mẽ để xây dựng đồ họa 2D (Sprite) và 3D (Mesh, Material).
* **Đường ống kết xuất** :
  + **Đường ống tích hợp** : Mặc định, đơn giản.
  + **URP (Universal Render Pipeline)** : Tối ưu hóa cho hiệu suất trò chơi cao.
  + **HDRP (High Definition Render Pipeline)** : Tạo hình ảnh chân thực, phù hợp cho đồ họa cao cấp.
* Hỗ trợ mô phỏng lực, va chạm, lực, và chuyển động của các đối tượng thông qua **Rigidbody** và **Collider** .
* **Animator** : Công cụ tạo hình ảnh nhân vật mượt mà.
* **Timeline** : Hỗ trợ hoạt động phức tạp.
* Tích hợp công cụ xử lý âm thanh 3D và hiệu ứng âm thanh.
* Hỗ trợ tạo trò chơi cho nhiều người chơi trực tuyến (nhiều người chơi) thông qua **Netcode** hoặc plugin thứ ba.
* **Sample 3D và 2D** : Nhân vật, môi trường.
* **Script và Plugin** : Tăng tốc độ phát triển lối chơi.
* **Trò chơi** : Temple Run, Monument Valley, Genshin Impact (di động).
* **AR/VR** : Ứng dụng giáo dục, y tế, giải trí.
* **Hình ảnh phim** : Sử dụng để xây dựng phim hoạt hình hoặc hiệu ứng.
* **Mô phỏng** : Huấn luyện lái xe, kiến ​​trúc và thiết kế sản phẩm.
* **Nền tảng** : Xuất bản trò chơi trên nhiều hệ điều hành.
* **Học dễ dàng** : Giao diện thân thiện, tài liệu phong phú.
* **Cộng đồng lớn** : Hỗ trợ nhanh chóng cho người dùng và chuyên gia.
* **linh tính hoạt cao** : Phù hợp cả người mới bắt đầu và đội ngũ phát triển chuyên nghiệp.
* Hiệu suất có thể giảm bớt một số công cụ khác (như Unreal Engine) trong siêu thực tế đồ họa.
* Yêu cầu mức độ ưu tiên cao để chạy mượt mà trên cấu hình thiết bị.

### **Quá trình phát triển của Unity:**

Ra mắt đầu tiên vào năm 2005 tại sự kiện Apple’s Worldwide Developer Conference bởi nhà sáng lập David Helgason, trải qua nhiều năm phát triển, Unity đã có version 5.5 hoàn thiện hơn về rất nhiều mặt. Tháng 5-2012 theo cuộc khảo sát Game Developer Megazine được công nhận là Game engine tốt nhất cho mobile. Năm 2014 Unity thắng giải “Best Engine” tại giải UK’s annual Develop Industry Exellence. Phiên bản mới nhất hiện nay là 2018.2 với nhiều tính năng nổi trội

### **Một số thống kê về Unity:**

· **5 tỷ lệ lượt tải xuống ứng dụng Unity:** Trò chơi và ứng dụng phát triển bằng Unity vẫn đạt được số lượng tải xuống trên toàn cầu.

· **Thiết bị di động:** Hơn 3 tỷ lệ thiết bị di động đã được tải xuống ít nhất một ứng dụng được phát triển bằng Unity.

· **Phần di động của trò chơi:** Trong 1.000 trò chơi di động miễn phí hàng đầu, hơn 38% được phát triển bằng Unity, có thể tạo ra sự phổ biến của nền tảng này

**· Hiện tại (năm 2024):** Unity có khoảng 4 tỷ lệ người dùng toàn cầu về thông tin trò chơi và ứng dụng được phát triển trên nền tảng này. Trong khi đó, Twitter (X) có khoảng 540 triệu người dùng hoạt động

*(nguồn: thống kê từ internet)*

### **Ưu điểm của Unity:**

· **Hỗ trợ đa nền tảng:** Xuất trò chơi trên 25 nền tảng: Windows, macOS, Android, iOS, PlayStation, Xbox, WebGL, VR/AR.

· **Dễ học và sử dụng:**  Giao diện thân thiện, Hỗ trợ phong phú tài liệu, cộng đồng.

· **Đồ họa 2D và 3D mạnh mẽ:** Hỗ trợ URP, HDRP để tối ưu hiệu suất và tạo hình ảnh chân thực.

· **Thư viện Asset Store:** Kho tài nguyên lớn: Mẫu 3D/2D, plugin, âm thanh, công cụ có sẵn.

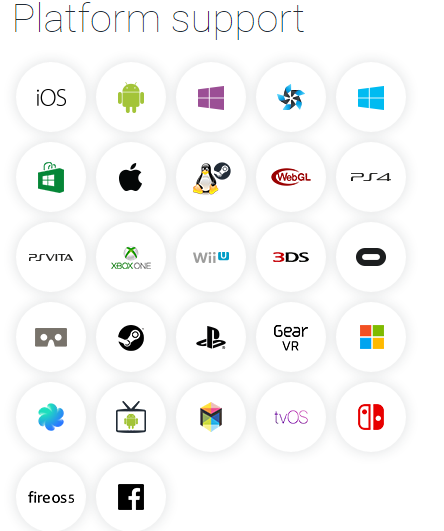
· **Tính linh hoạt:** Phù hợp từ dự án nhỏ đến game AAA., Hỗ trợ nhiều loại: game, AR/VR, phim, mô phỏng.

· **Công nghệ hiện đại:** Hệ vật lý NVIDIA PhysX, hệ thống Animation Mecanim, Hỗ trợ AI, kết nối mạng và nhiều người chơi.

· **Cộng đồng lớn và miễn phí:** Phiên bản miễn phí cho cá nhân, doanh nghiệp nhỏ., Diễn đàn, hội thảo, tài liệu trực tuyến phong phú.

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

Hỗ trợ đa nền tảng: Một trong các thế mạnh của Unity3D chính là khả năng hỗ trợ gần như toàn bộ các nền tảng hiện có bao gồm: PlayStation 3, Xbox 360, Wii U, iOS, Android, Windows, Blackberry 10, OS X, Linux, trình duyệt Web và cả Flash. Nói cách khác, chỉ với một gói engine, các studio có thể làm game cho bất kỳ hệ điều hành nào và dễ dàng convert chúng sang những hệ điều hành khác nhau. Đồng thời, đây cũng là giải pháp cho các game online đa nền tảng – có thể chơi đồng thời trên nhiều hệ điều hành, phần cứng khác nhau như Web, PC, Mobile, Tablet….



Dễ sử dụng: Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ.

Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại.

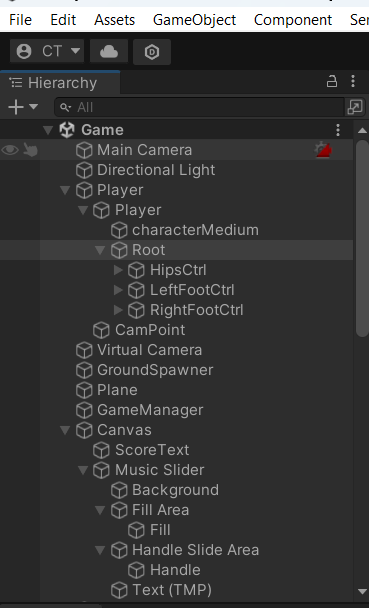
* 1. **Các khái niệm trong Unity**

### **Asset**

Đây là kho tài nguyên cho việc xây dựng game trong một project của Unity. Các tài nguyên này có thể là hình ảnh, âm thanh, hoặc một mô hình 3D có sẵn. Unity sẽ tham chiếu đến các tập tin chúng ta sẽ sử dụng để tạo ra các tài nguyên cho trò chơi. Đây là lý do tại sao trong bất kỳ thư mục chứa project sử dụng Unity thì tất cả các tập tin tài nguyên phải được lưu trữ trong một thư mục con tên là Assets.   
**2.2.2. Scene**

Trong Unity, chúng ta có thể xem Scenes là các màn chơi, cấp độ chơi riêng lẻ, hoặc các vùng của nội dung trò chơi.

Bằng cách xây dựng trò chơi với nhiều cảnh, chúng ta sẽ có thể phân phối thời gian tải và thử nghiệm các phần khác nhau của trò chơi riêng lẻ một cách nhanh chóng và chính xác.



### **Game Object**

Khi một tài nguyên được sử dụng trong một scene, khi đó chúng ta có thể coi tài nguyên này là một “Game Object” mới. Mỗi GameObject phải chứa ít nhất một thành phần, đó là thành phần “Transform”. Transform chứa các phép để biến đổi góc quay, tỷ lệ hay tịnh tiến của đối tượng.

* + 1. **Component**

Component là các thành phần trong một Game Object của Unity. Bằng cách đính kèm các thành phần vào cho một đối tượng, chúng ta có thể áp dụng ngay các phần mới của game engine vào đối tượng. Thông thường các thành phần này được Unity xây dựng sẵn như ánh sáng, camera, particle, hiệu ứng vật lý… 

### **Script**

Script là thành phần quan trọng nhất trong Unity, có thể xem scripts như là linh hồn của game. Chúng ta có thể viết kịch bản cho game bằng C#, Java Scripts hoặc Boo (một dẫn xuất của ngôn ngữ Python). Theo nhiều người đã sử dụng Unity thì code bằng C# sẽ giúp game chạy nhanh hơn và giúp kiểm soát code tốt hơn do tất cả các biến phải được khai báo rõ ràng. Mặt khác ngôn ngữ C# rất tiện dụng để lập trình, nên trong thành phẩm Game vượt chướng ngại vật 3D ở chương 4, chúng em dùng ngôn ngữ C# để viết kịch bản cho game. Mỗi file script C# là một class bắt buộc kế thừa từ lớp MonoBehaviour, có tên class phải trùng với tên file script.

Một đoạn script muốn thực thi được thì nó phải được gắn vào một đối tượng .

* + 1. **Prefab**

Prefab là một bản sao lưu các vật thể chúng ta đã tạo, bao gồm các kịch bản cho hành động (khởi tạo, di chuyển, hay hủy đối tượng). Ta có thể sử dụng đối tượng này nhiều lần trong trò chơi, và cũng có thể sử dụng lại cho project khác. Prefab cho phép chúng ta lưu trữ các đối tượng, toàn bộ thành phần bên trong và cấu hình hiện tại.

### **Material và Shader**

Shader là đoạn script qui định cách thức render của chất liệu trên bề mặt vật thể. Material sử dụng shader để làm chất liệu cho mô hình. Giữa materials và shaders có mối liên hệ với nhau. Shaders qui định các thuộc tính cần để shader làm việc. Còn material cho phép gán hình ảnh vào các thuộc tính đó từ Asset.

* + 1. **UnityAPI**

UnityAPI chứa rất nhiều lớp hỗ trợ lập trình game, trong đó có một số lớp quan trọng như :

**MonoBehaviour**: tất cả các script muốn gắn vào một đối tượng game bắt buộc phải kế thừa từ lớp này.

**GameObject**: lớp cha của tất cả các thực thể trong scene.

**Component**: lớp cha của tất cả các thành phần có thể gắn vào đối tượng.

**Transform**: giúp thay đổi vị trí, xoay, biến đổi tỉ lệ mô hình.

**Input**: hỗ trợ lập trình với chuột, cảm ứng đa điểm, cảm biến gia tốc.

**Camera**: giúp lập trình camera.

**Light**: giúp tạo ánh sáng trong game.

**Projector**: giúp chiếu texture lên bề mặt vật thể.

**ParticleEmitter**: hỗ trợ tạo các hiệu ứng particle đẹp mắt.

**Audio**: hỗ trợ lập trình với âm thanh.   
**Animation**: chạy chuyển động của mô hình nhân vật.

**Rigidbody**: giúp tạo hiệu ứng vật lý liên quan đến trọng lực như bóng nẩy, lăn, ..

**CharacterController**: giúp điều khiển nhân vật di chuyển theo độ cao địa hình.

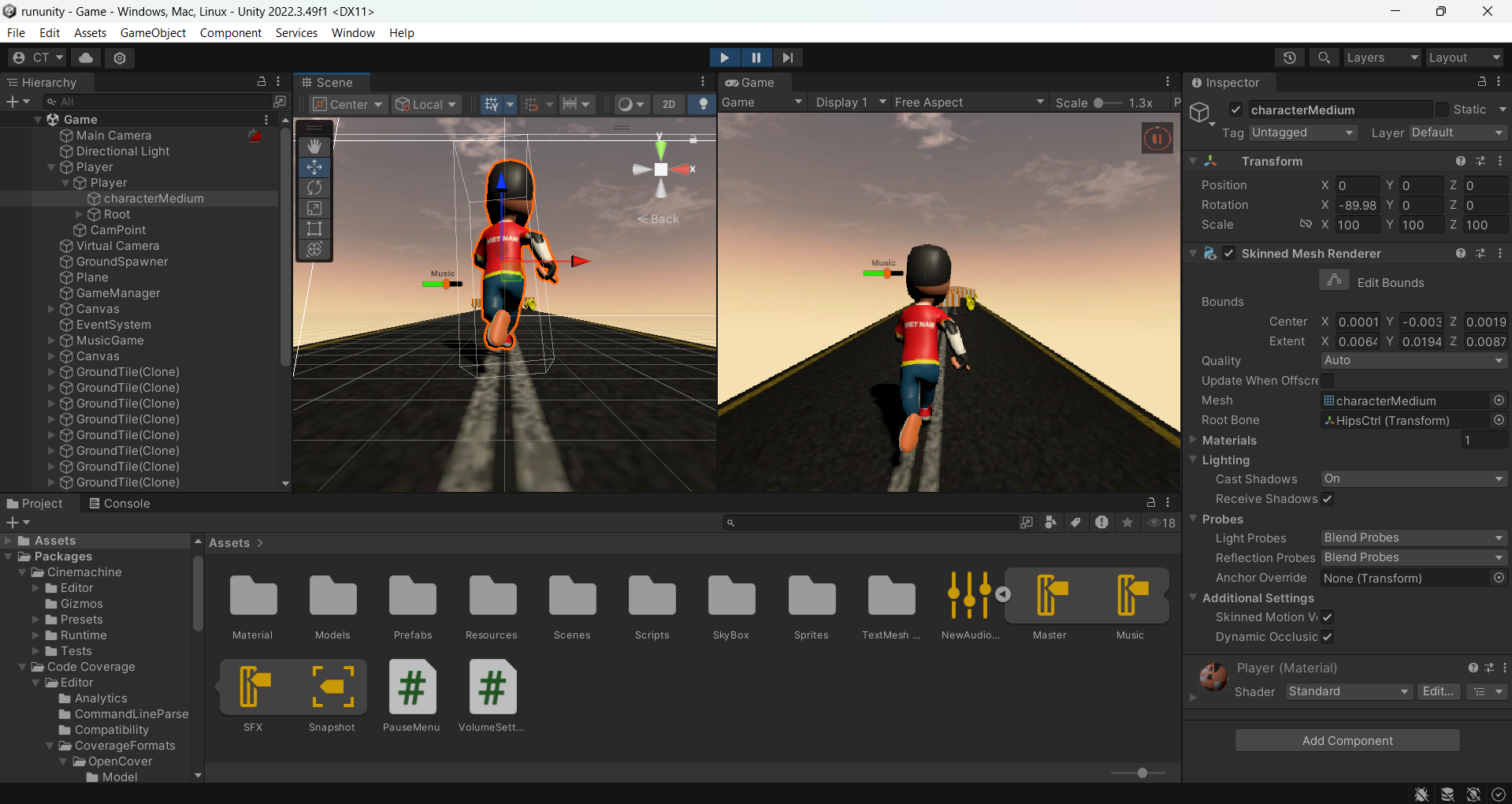
**Collider**: hỗ trợ lập trình va chạm giữa các vật thể.

**GUI**: giúp lập trình giao diện người dùng trên Unity.

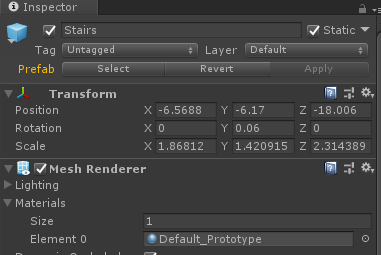
**Không gian 3D và vector trong không gian**

**Nhắc về hệ tọa độ không gian 3 chiều chắc hẳn chúng ta đều nhớ môn** hình học không gian hoặc đã từng sử dụng một công cụ dựng hình 3D nào đó. Hệ tọa độ không gian 3 chiều bao gồm 3 trục X, Y, Z hay hiểu đơn giản là chiều ngang (X) chiều cao (Y) và chiều sâu (Z). Chúng được kí hiệu theo cú pháp: (X, Y, Z)

Trong không gian 3 chiều, tồn tại 1 điểm gọi là Origin hoặc Word Zero. Đây là điểm có tọa độ (0,0,0). Tất cả tọa độ của các đối tượng tồn tại trong không gian đều có mối liên hệ với điểm này, cách tính này là theo Word Space. Tuy nhiên, để việc tính toán đơn giản, người ta sử dụng thêm một khái niệm gọi là Local Space nhằm xác định tọa độ của đối tượng này so với đối tượng khác. Mối quan hệ này gọi là parent – child.

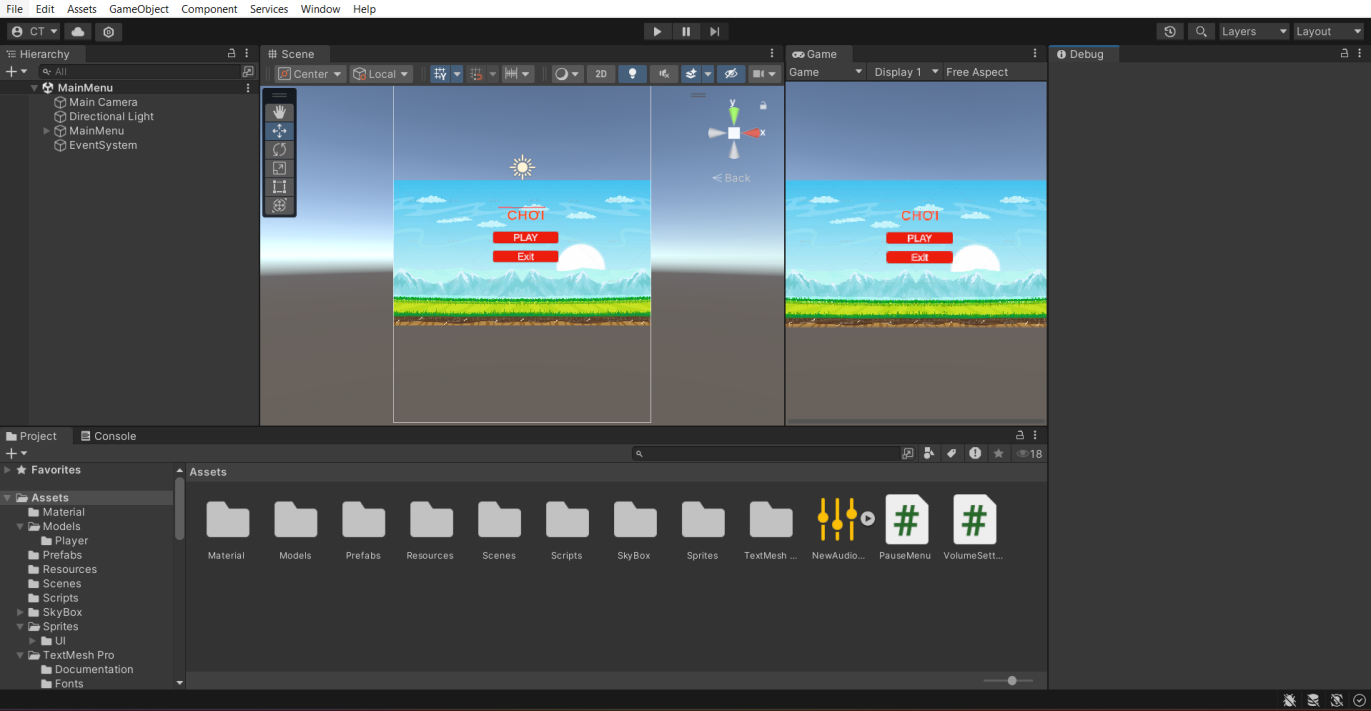


*Một nhân vật trong không gian 3 chiều*



*Tương ứng với Rotation X, Y, Z*

* 1. **Cửa sổ làm việc**

****

**2.3.1.**     **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 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

Để thêm assets vào project, 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.

**2.3.2. Hierarchy**

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. Chúng ta 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 .

**2.3.3. Parenting**

          Tức là thư mục chứa hay thư mục gốc, bất kỳ một game object 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.

**2.3.4. 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: dung cho việc thể hiện Scene view.

- Play/Pause/Step Buttons: dung cho view game, chỵ game ngay trong Editor để kiểm tra.

- Layer Drop-down kiểm soát đối tượng nào đang được thục hiện trong Scene view

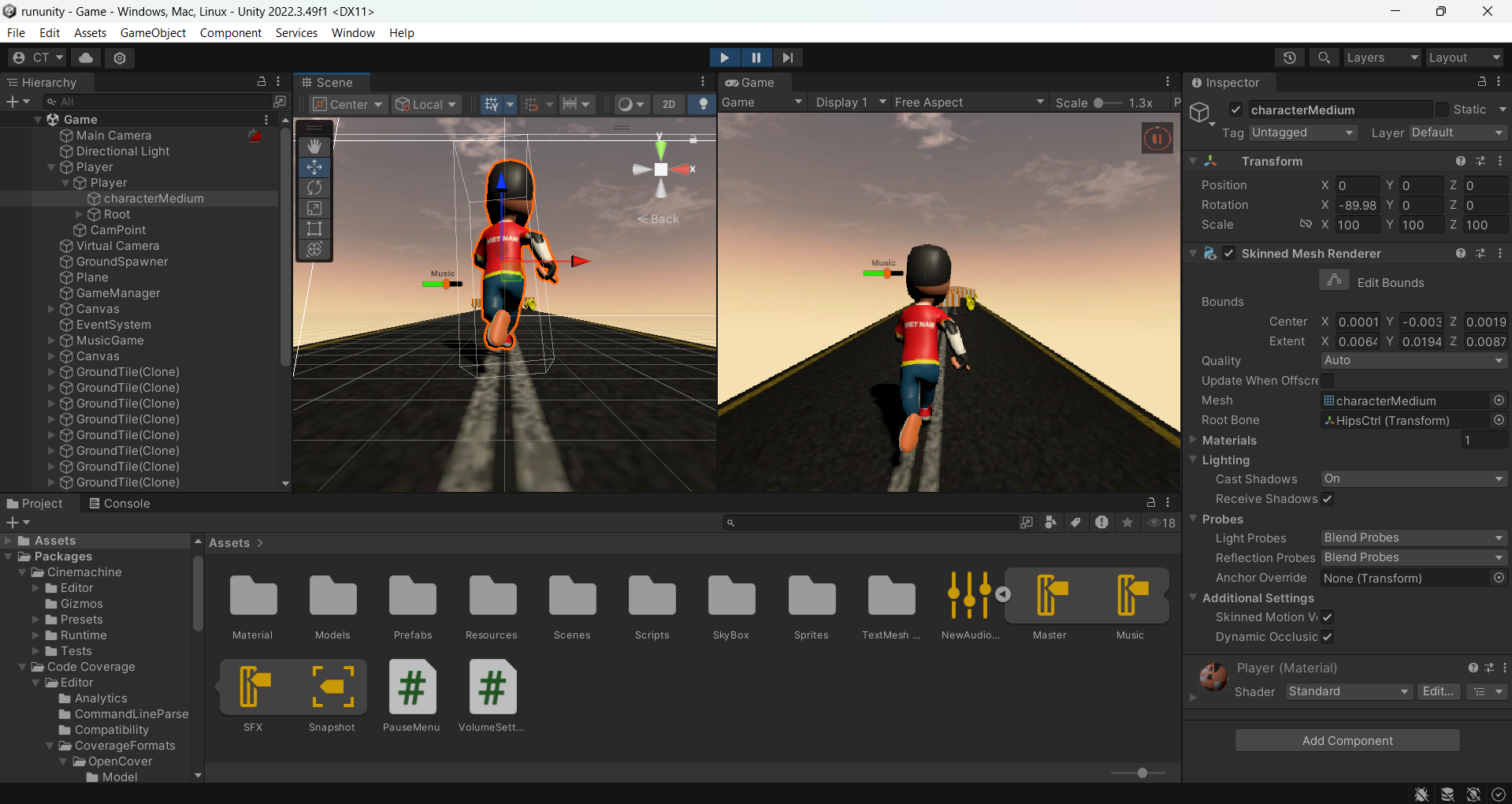
- Layout Drop-down kiểm soát sự sắp xếp cảu các Views.

**2.3.5. Scene View**

Là nơi Design Game , đối tượng Maneuvering và Importanting 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, click vào hình vuông ở giữa hay giữ phím Shift + Click để chuyển đổi chế độ nhìn.

Khi xây dựng một Game, chúng ta sẽ đặt rất nhiều đối tượng vào trong Game, khi đó ta có thể dử dụng các công cụ Transform Tools ở trong Toolbar để di chuyển, xoay, phóng to tu nhỏ từng đối tượng. Nếu họn 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.Ta có thể thay đổi đối tượng tùy ý, nếu muốn chính xác, có thể chỉnh chi tiết ở bảng Inspector.



*Các góc nhìn từ các hướng khác nhau*

**2.3.6. 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. Chúng ta sẽ cần ít nhất là một hoặc nhiều hơn số lượng các Camera để quết định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game.

**2.3.7. Play Mode**

Sử dụng những nút trên Toolbar để điều khiển Editor Play Mode, và xem trước Game 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.

**2.3.8. 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 v..v... 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 ta 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 chúng ta định nghĩa một giá trị là public cho một kiểu đối tượng (như GameObject hay Transform), ta 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 cảu Component để xuất hện context menu dành cho những thiết lập của Component.

Inspector cũng sẽ thể hiện mọi thông số Import Setting của assets đang làm việc.

* 1. **Các phần mềm hỗ trợ khác**

### **2.4.1. Photoshop**

Adobe Photoshop là phần mềm đồ họa có tác dụng chính là chỉnh sửa ảnh. Được phát triển và phát hành bởi Adobe năm 1988. Hiện nay Ps được xem là phần mềm đồ họa bitmap mạnh nhất trên thị trường, và được hầu hết những người làm thiết kế đồ họa cũng như các photography sử dụng. Photoshop cũng được đưa vào giáo trình giảng dạy chính thức của các trường đào tạo nghề cũng như các trường cao đẳng, đại học.

Photoshop giúp ta tạo ra các bản vẽ nhân vật, cắt ghép các chi tiết, tạo chữ, tạo nền trong suốt

**2.4.2**. **After Effects**

**Adobe After Effects** là phần mềm đồ họa tổng hợp được phát triển bởi hãng Adobe Systems. Phần mềm này được ứng dụng nhiều trong  đồ họa động và kỹ xảo chuyên nghiệp. Chức năng chính của After Effects là tạo ra được những chuyển động đồ họa ấn tượng, đẹp mắt. Đây là một công cụ tuyệt vời cho ngành thiết kế đồ họa, kỹ xảo trong quảng cáo hay game hoặc truyền hình. After Effects tương thích ngược với nhiều phần mềm khác của Adobe vì vậy ta có thể lấy sản phẩm tạo ra từ Photoshopđể xử lý tiếp, hoàn thiện hiệu ứng nhân vật trong game

### **2.4.3. Autodesk 3ds Max**

· **Tạo mô hình 3D** : Thiết kế các vật thể, vật cản, nhân vật hoặc môi trường trong trò chơi.

· **Mô hình hóa ưu tiên tối ưu** : Giảm số lượng đa giác (đa giác) để đảm bảo hiệu suất khi nhập vào Unity.

· **UV Mapping and Texturing** : Gắn vật liệu và họa tiết (kết cấu) chi tiết cho các đối tượng.

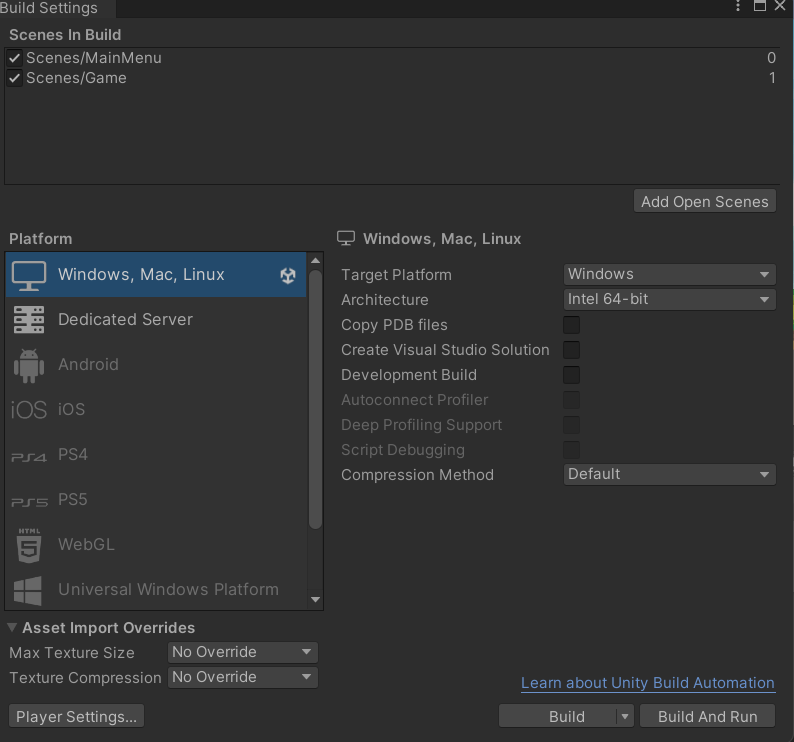
· **Rigging và Animation** : Tạo chuyển động cho vật thể hoặc vật thể động (ví dụ: cản vật chuyển động).

· **Xuất sang Unity** : Hỗ trợ định dạng FBX, dễ dàng import vào Unity để tích hợp vào game.

**2.4.4. Visual Studio**

**Visual Studio 2022** là một môi trường hợp tác phát triển (IDE) được Microsoft phát triển, hỗ trợ lập trình viên xây dựng, chỉnh sửa, kiểm tra và gỡ lỗi phần mềm. Đây là công cụ phổ biến dành cho các nhà phát triển ở nhiều lĩnh vực, từ ứng dụng máytínhđểbàn,web,diđộngđếnđámmâydịchvụ. 

**2.4.5. Hỗ trợ Build (xuất game) đa nền tảng:**



* 1. **Game và hướng phát triển**
     1. **Các thể loại game nổi bật trong những năm gần đây:**

## **1. Game nhập vai trực tuyến (MMORPG)**

MMORPG được phát triển bởi Xenon Interactive và WeaselDev, sử dụng Unity làm nền tảng. Trò chơi này mang phong cách pixel-art, cung cấp trải nghiệm trải nghiệm lưu trữ trong thế giới mở rộng với các tính năng như chiến đấu, chế độ tạo, nhiệm vụ PvE và PvP, cùng các lớp nhân vật như pháp sư, chiến binh. Trò chơi được hỗ trợ đa nền tảng (PC và di động) và được đánh giá cao về cách tạo lối chơi và giao diện đơn giản.

Một số cập nhật đáng chú ý bao gồm cải tiến hệ thống AI, mạng tối ưu hóa, bổ sung chế độ chiến đấu PvP và tạo giao diện PC thân thiện hơn. Elysium Online vẫn đang được phát triển và hoàn thiện để cải thiện trải nghiệm người chơ. có thể kể đến như: **Elysium Online**

## 2. Game nhập vai (RPG Role-Playing Game)

Người chơi đóng vai 1 hoặc một số nhân vật sống trong một thế giới tưởng tượng, tham gia các sự kiện, giao tiếp với các nhân vật phụ, tham gia cuộc chiến với quá trình xây dựng nhân vật thông qua các việc tăng cấp độ (level). Dòng game nhập vai được xây dựng kết hợp với nhiều bối cảnh và cốt truyện thú vị từ cổ trang đến hiện đại để tạo nên sự đa dạng trong thị trường game. Những cái tên vàng trong làng game nhập vai Việt Nam gồm có: **Hollow Knight**

## 3. Game chiến thuật (SLG - SimuLation Game)

Đây là loại game thể loại trò chơi yêu cầu người chơi phải áp dụng chiến lược để quản lý tài nguyên, điều hành quân đội hoặc các yếu tố khác trong một thế giới ảo. Mục tiêu của các trò chơi SLG là để người chơi suy nghĩ chiến lược và đưa ra quyết định thông minh trong các tình huống phức tạp phức tạp, từ việc xây dựng căn cứ, chiến binh quân lính, cho đến các trận chiến chiến thuật, cụ thể như game**: Age of Empires**

## 4.Game MOBA

Game MOBA hành động chiến lược thời gian thực (MOBA - Multiplayer Online Battle Arena - Đấu trường người chơi trực tuyến) là game hành động chiến lược thời gian thực (Action Real Time Strategy - ARTS), người chơi điều khiển nhân vật Hero hoặc Anh Hùng. Mục tiêu chính là phá hủy nhà chính của đối phương và giành chiến thắng. Game MOBA sở hữu lối chơi trên 2 phe đối lập, giao tranh ở bất cứ đâu trên bản đồ rộng lớn và nhiệm vụ chung của 2 bên là phải phá hủy căn cứ đối phương, farm, giết quái, đối thủ.... Các vị tướng có các kỹ năng và lối chơi khác nhau, người chơi cần tìm hiểu để có lựa chọn phù hợp với chiến thuật của mình. Trên nền tảng PC thì DOTA 2 và Liên Minh Huyền Thoại là hai ông lớn bậc nhất của làng game MOBA thế giới. Những tựa game MOBA trên nền tảng Mobile đã bắt đầu phát triển một cách mạnh mẽ với những cái tên tiêu biểu như: **Battalion 1944**

## **Game thể thao (Sports)**

Thể loại game sports mô tả một môn thể thao trong thế giới đồ họa tưởng tượng để người chơi được hòa mình vào trải nghiệm vận động thể dục thể thao. Môn thể thao thường được yêu thích và đang xuất hiện ở nhiều tựa game thể thao là bóng đá. Game thường tổ chức theo cơ cấu mùa giải, và cho phép người chơi đảm nhận các vị trí huấn luyện viên, ông bầu… bên cạnh các tính năng trao đổi, chuyển nhượng cầu thủ, quản lý ngân sách, như game: **Football Manager Mobile**

**2.5.2. Hướng phát triển:**

Mỗi thể loại game đều có những điểm hấp dẫn riêng và phù hợp với sở thích của từng nhóm đối tượng. Để mở rộng phạm vi người chơi chúng em phát triển game bắn súng 3D nhập vai FPS góc nhình thứ nhất, có yếu tố kinh dị kết hợp chiến thuật, sử dụng MAP (bản đồ) thế giới mở. Hiện tại chúng em xây dựng phiên bản game cho người dùng chạy hệ điệu hành Windows dưới dạng file thực thi (.exe). Nhớ tính đa nền tảng của Unity, trong thời gian sắp tới game sẽ được xây dựng bản (.apk) hoặc phát hành trên CH Play cho người dùng android, có mặt trên Appstore cho người dùng IOS

# CHƯƠNG 3 MÔ TẢ THIẾT KẾ GAME

- Phần mềm: Unity

- Nền tảng: Windows 11

* Ngôn ngữ lập trình : C#
* Công cụ hỗ trợ Visual Studio 2022

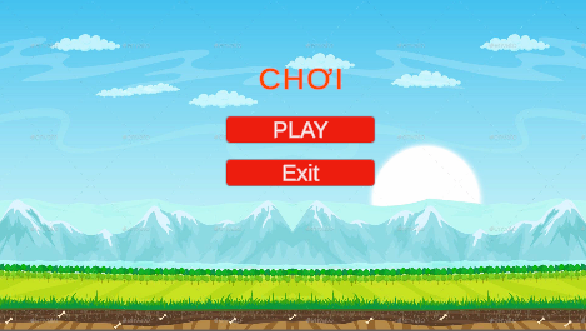
## 3.1 Ý tưởng thực hiện

Thế giới đã trải qua một thảm họa thiên nhiên khủng, tạo thành phố trở nên hoang tàn, đổ nát. Các khu vực đô thị giờ chỉ còn lại những đổ nát, đổ nát, đổ nát, cây cối bị phá hủy, và các con đường bị chặn bởi vật cản. Người chơi vào vai một nhân vật (có thể là một người sống sót sau thảm họa) và phải vượt qua những khu vực đầy rẫy nguy hiểm để tìm kiếm sự sống, người cứu người hoặc chỉ đơn giản là thoát khỏi thành phố đang bị hủy bỏ.

## 3.2. Tạo menu và nhân vật

**3.2.1. Menu:**

Đây là giao diện khi vào game. Play (Vào chơi) và Exit (Thoát game)



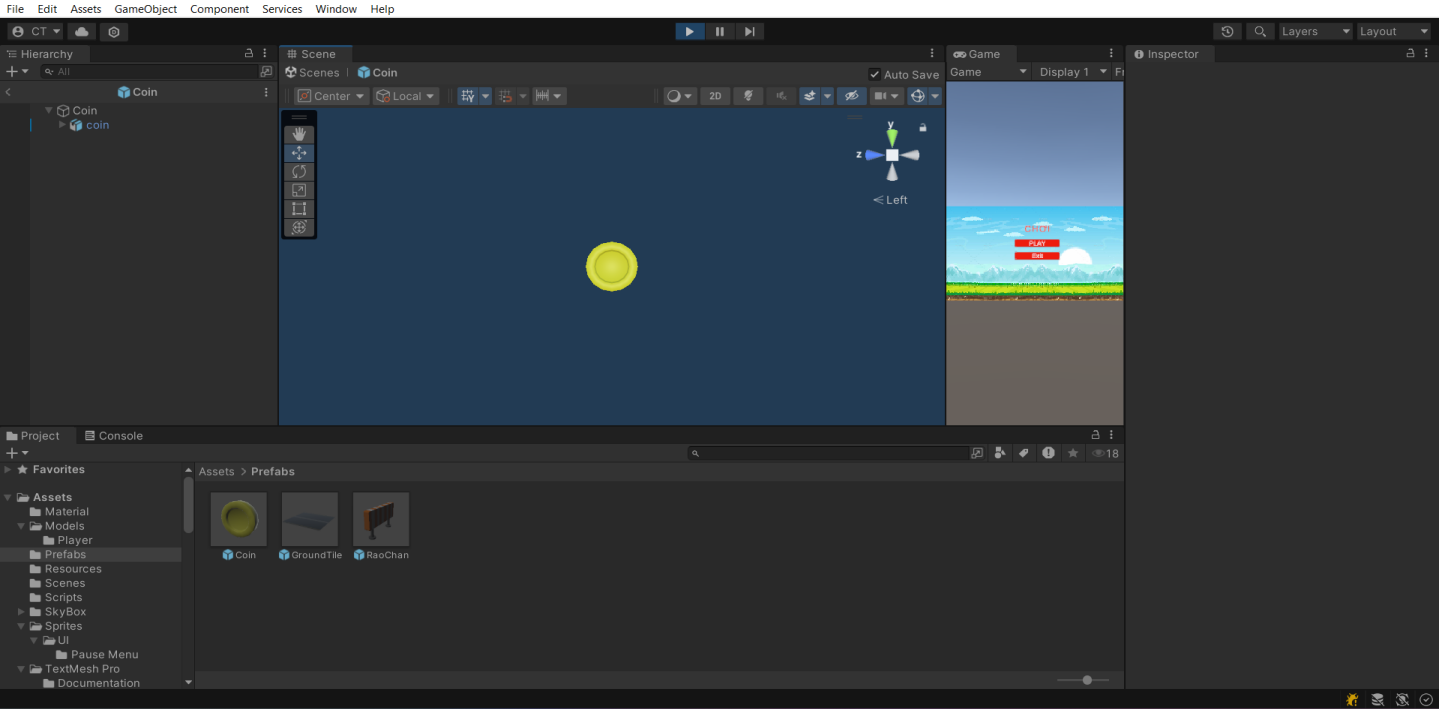
**3.2.2. Nhân Vật, Vật phẩm và các chi tiết khác:**

1. **Player** (Nhân Vật):

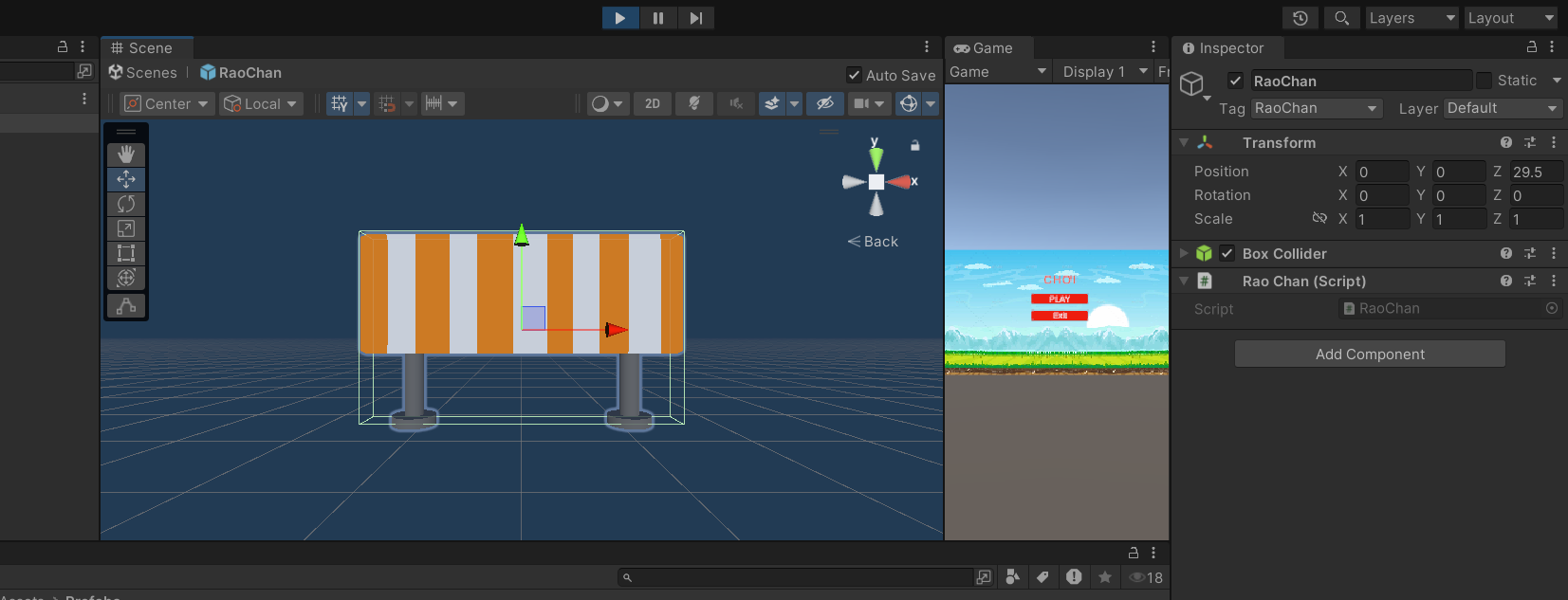


* Di chuyển và chạy nhảy để tránh chướng ngại vật

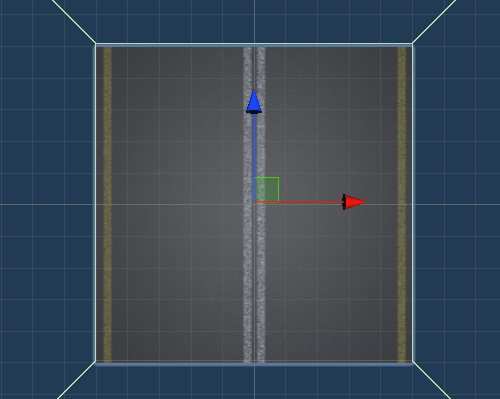
1. **Coin (**đồng xu**):**



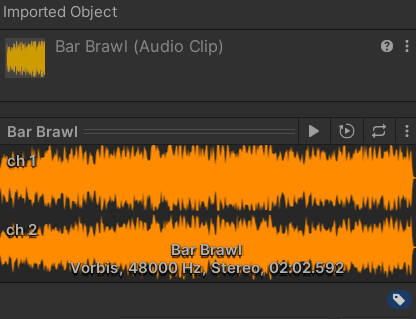
**3. Bao gồm RaoChan** (vật cản)



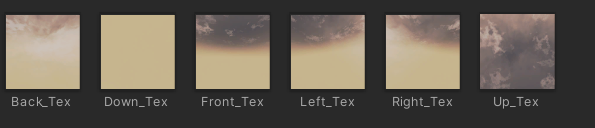
**4. GroundTile** (Gạch lát nền di chuyển)



1. **Music** (Nhạc nền):

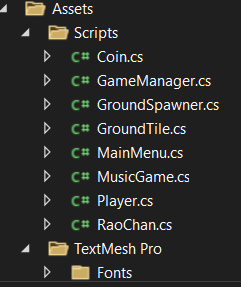
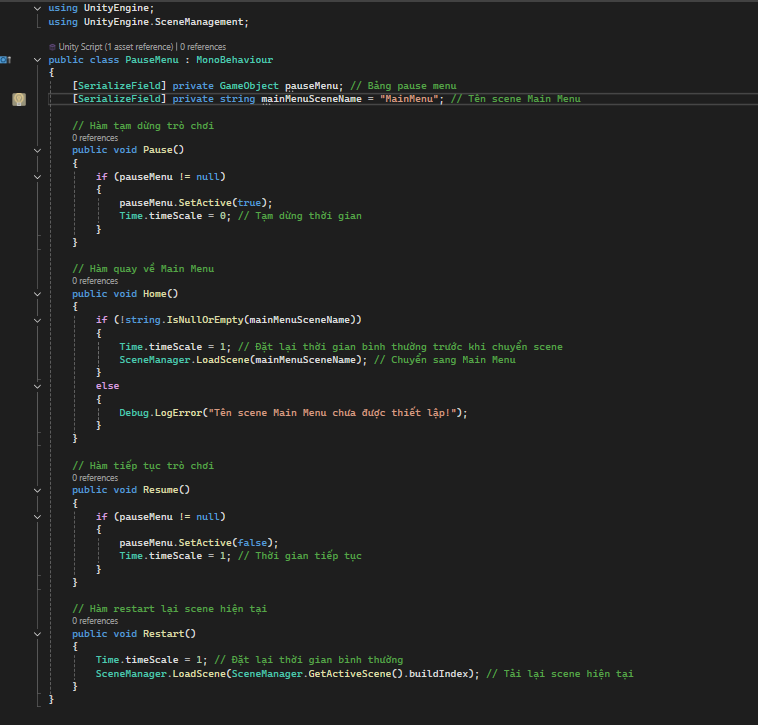


1. **Background** (Phong cảnh ):



1. **Hỗ trợ vận hành Game:**

**Visual studio 2022**



## 3.3. Xử lý di chuyển, va chạm, điểm số, kết thúc màn chơi

Khi phát triển một trò chơi vượt chướng ngại vật 3D trong Unity, có một số yếu tố quan trọng cần xử lý để tạo ra một trải nghiệm mượt mà cho người chơi, bao gồm di chuyển nhân vật, va chạm, tính điểm và kết thúc màn hình.

**Di chuyển nhân vật:** Người chơi điều khiển nhân vật trong không gian 3D bằng các phím điều khiển. Để thực hiện điều này, bạn có thể sử dụng một hệ thống vật lý như **Rigidbody** hoặc **CharacterController** . Việc chuyển nhân vật thường được thực hiện bằng cách đọc đầu vào từ bàn phím (hoặc gamepad) và dịch chuyển nhân vật theo hướng mà người chơi yêu cầu. Trong trò chơi vượt chướng ngại vật, nhân vật sẽ có thể nhảy qua các vật cản, điều này được thực hiện khi người chơi nhấn nút nhảy (thông thường là Space), và nhân vật sẽ được đưa lên một khoảng cách nhất định.

**Và:** Khi nhân vật và phải vật cản, game sẽ xử lý và chạm để quyết định kết quả. Nếu nhân vật và phải ngăn cản vật chất, ngoại trừ tường, đá hay vật cản vật, trò chơi có thể tạo ra nhân vật mất mạng hoặc kết thúc màn chơi. This va chạm được phát hiện qua **Collider** thông tin và được xử lý trong các hàm kiểm tra và chạm như OnCollisionEnter. Có thể dẫn đến việc nhân vật quay lại điểm phát hiện hoặc trò chơi kết thúc nếu không thể tiếp tục.

**Tính điểm:** Điểm số trong game vượt chướng ngại vật có thể được tính theo nhiều cách, ví dụ như qua thời gian người chơi hoàn thành màn chơi hoặc số lượng coin và vật cản vượt qua.

**Kết thúc màn hình:** Màn hình sẽ kết thúc khi người chơi hoàn thành mục tiêu của màn hình (ví dụ: đến đích hoặc vượt qua một số vật cản nhất định). Nếu nhân vật quá nhiều vật cản hoặc không thể vượt qua các thử thách, trò chơi sẽ kết thúc và màn hình trò chơi kết thúc sẽ xuất hiện. Người chơi có thể được yêu cầu bắt đầu lại hoặc chuyển sang màn chơi tiếp theo nếu có.

Thông qua việc kết hợp các yếu tố di chuyển, va chạm, tính điểm và kết thúc màn chơi, một trò chơi vượt chướng ngại vật 3D sẽ mang lại trải nghiệm đầy thử thách và hấp dẫn cho người chơi.

## 3.4. Tiến trình của Game

### **1.Giai đoạn lên ý tưởng và thiết kế ban đầu**

* **Mục tiêu** : Xác định mục tiêu của trò chơi, cốt truyện, thể loại và cơ chế chơi trò chơi cơ bản. Trong trường hợp trò chơi bị cản trở, mục tiêu chính có thể là vật giúp vượt qua vật cản trong môi trường 3D.
* **Thiết kế nhân vật và môi trường** : Lên ý tưởng về các nhân vật (người chơi) và môi trường (các loại vật cản, địa hình).

**2. **Trò chơi phát triển giai đoạn (Lập trình và nội dung thiết kế)****

* **Lập trình di chuyển và điều khiển vật vật** : Bước đầu tiên quan trọng trong quá trình cài đặt trò chơi là tạo ra hệ thống di chuyển cho nhân vật, cho phép người chơi chuyển qua các khu vực và nhảy qua vật cản.
* **Thiết kế ngăn chặn vật** : Các chướng ngại vật sẽ là yếu tố chính tạo thử thách cho trò chơi. Những vật cản này có thể bao gồm các bức tường bẫy nguy hiểm.
* **Tạo môi trường trò chơi (Thiết kế cấp độ)** : Xây dựng các cấp độ với các thử nghiệm khác nhau. Mỗi cấp độ có thể bao gồm các môi trường khác nhau và các loại vật cản phù hợp.
* **Xử lý** : Lập trình hệ thống và các vật cản và vật cản. Khi nhân vật va chạm với một vật thể nguy hiểm (chẳng hạn như bẫy hoặc vật cản), trò chơi cần phải xử lý vấn đề này, có thể dẫn đến mất mạng hoặc trò chơi kết thúc.
* **Tính điểm và hệ thống thăng tiến** : Tạo ra hệ thống tính điểm, có thể tính toán dựa trên thời gian hoàn thành màn hình hoặc số lượng vật cản vượt qua. Hệ thống này sẽ giúp người chơi theo dõi tiến trình của mình.

### ***3.* ***Thử nghiệm*****

* **Kiểm tra lối chơi** : Sau khi các tính năng chính đã được thiết lập, cần kiểm tra lối chơi để đảm bảo các cơ chế chuyển đổi, va chạm và tính năng hoạt động chính xác như mong đợi. Đảm bảo rằng người chơi có thể vượt qua các rào cản hợp lý và không gặp lỗi không đáng có.
* **Cải thiện giao diện người dùng (UI)** : Thiết kế giao diện người dùng rõ ràng, bao gồm các yếu tố như điểm số, thanh sức khỏe và các biểu tượng cho các tính năng như nhảy, di chuyển. Cần chắc chắn rằng người chơi có thể dễ dàng theo dõi tiến trình và điều khiển vật thể.

### 4. **Giai đoạn hoàn thiện và xuất bản**

* **Tạo âm thanh và hiệu ứng** : Âm thanh là một yếu tố quan trọng trong trò chơi trải nghiệm nâng cao trải nghiệm. Âm thanh như bước chân, tiếng nhảy, tiếng va chạm, và nhạc nền sẽ giúp tạo bầu không khí cho game. Các hiệu ứng hình ảnh như độ sáng, độ bóng và hiệu ứng đặc biệt khi vượt qua vật cản hoặc khi vật vật va chạm cũng cần được bổ sung vào.
* **Cài đặt màn hình kết thúc** : Cần có một màn hình kết thúc (trò chơi kết thúc) hiển thị khi người chơi vượt qua, bao điểm, thời gian hoàn thành và tùy chọn để chơi lại hoặc quay về menu chính. Màn hình chiến thắng (hoàn thành cấp độ) cũng cần được thiết kế nếu người chơi hoàn thành một cấp độ.
* **Kiểm tra lỗi và sửa lỗi** : Tiến hành kiểm tra tất cả các phần của trò chơi một lần nữa để phát hiện và sửa các lỗi có thể xảy ra trong quá trình chơi. Điều này giúp đảm bảo trò chơi hoạt động ổn định và không phải gặp vấn đề kỹ thuật.

### **5. **Xuất bản và tiếp theo giai đoạn****

* **Xuất bản trò chơi** : Sau khi trò chơi đã hoàn thiện, trò chơi có thể được xuất bản trên các nền tảng khác như Steam, Google Play, App Store hoặc các nền tảng trò chơi khác. Cần phải đảm bảo trò chơi được tối ưu hóa cho từng nền tảng và đáp ứng các yêu cầu về phần cứng, hệ điều hành, vv

### 6. **Hỗ trợ giai đoạn sau khi ra mắt**

* **Cập nhật và sửa lỗi** : Sau khi game ra mắt, có thể sẽ có vấn đề phát sinh mà người chơi gặp phải. Các bản vá lỗi (bản vá) cần được phát hiện để sửa các lỗi và cải thiện trò chơi.
* **Cải tiến nội dung bổ sung** : Để duy trì tâm trí của người chơi, bạn có thể bổ sung các cấp độ mới, vật cản mới, tính năng mới hoặc các chế độ chơi đặc biệt trong bản cập nhật org sau này.

## 3.5. Hoạt họa và cảm nhận

### **1.Hoạt hình nhân vật (Hoạt hình nhân vật)**

* **Di chuyển mượt mà** : Các chuyển động như chạy, nhảy, trượt phải được hoạt họa theo cách mượt mà và tự nhiên để người chơi có thể cảm nhận được hoạt động linh hoạt của nhân vật. Ví dụ: nhân vật sẽ hoạt động khác biệt khi tăng tốc, nhảy cản vật hoặc thực hiện các hành động đặc biệt.
* **Hoạt họa khi va chạm** : Khi nhân vật va vào vật vật, có thể có hoạt họa phản ứng như ngã, Cánh ra hoặc rung lắc để tạo ra sự sống động và phản phản trực quan.
* **Hoạt động họa tiết khi hoàn thành màn chơi** : Mỗi lần hoàn thành màn chơi hoặc chiến thắng, nhân vật có thể có các hoạt hình ăn mừng như nhảy lên hoặc tạo ra các cuộc bầu cử chỉ vui mừng.

### 2. **Môi trường Hoạt họa (Hoạt hình môi trường)**

* **Chướng vật cản** : Nếu game có các chướng ngại vật, hoạt họa môi trường sẽ giúp tạo ra sự căng thẳng và sắp thức cho người chơi.
* **Hiệu ứng đặc biệt** : Các yếu tố trong môi trường như có thể thay đổi và di chuyển, tạo ra một không gian sống động và thay đổi liên tục.

### 3. **Hiệu ứng âm thanh (Hiệu ứng âm thanh)**

* **Âm thanh khi chuyển** : Âm thanh bước chân, tiếng nhảy hoặc tiếng trượt sẽ giúp người chơi cảm nhận rõ ràng từng hành động của nhân vật.
* **Âm thanh va chạm** : Khi nhân vật va vào vật cản, một âm thanh mạnh mẽ và rõ ràng sẽ tạo ra sự phản hồi cho người chơi, đồng thời tăng tính chân thực.
* **Âm thanh khi chiến thắng** : Âm nhạc hoặc âm thanh chiến thắng sẽ tạo cảm giác hưng phấn mỗi khi người chơi hoàn thành một cấp độ hoặc vượt qua thử nghiệm.

### 4. **Cảm nhận tốc độ và chuyển động**

* **Tốc độ di chuyển** : Tạo cảm giác tốc độ thực tế khi nhân vật di chuyển nhanh hoặc nhảy qua các vật cản. Sự thay đổi tốc độ khi nhân vật chạy nhanh hoặc giảm tốc độ sẽ được thực hiện thông qua các hình ảnh hiệu ứng như mờ cảnh vật.

### 5. **Cảm nhận không gian và chiều sâu**

* **Môi trường 3D** : Tạo ra cảm giác chiều sâu trong không gian 3D, nơi người chơi có thể thấy rõ các vật cản từ nhiều góc độ khác nhau.
* **Chuyển động camera** : Thay đổi góc nhìn của camera để tạo ra cảm giác Người chơi đang chuyển qua một môi trường rộng lớn, ví dụ như khi vật chạy nhanh, camera có thể thay đổi góc nhìn để làm nổi bật cảm giác Chuyển động.

### 6. **Người dùng giao diện (UI)**

* **Chức năng UI động** : Các yếu tố UI như điểm số, cách sống, thời gian cần được thay đổi và hiển thị một sinh động. Ví dụ: khi người chơi mất mạng, mạng thanh sẽ biến mất dần dần.
* **Giao diện người dùng phản hồi** : Giao diện cần phản hồi nhanh chóng hành động của người chơi. Ví dụ: khi người chơi nhấn nút, nhấp nháy.

### 7. **Cảm giác thử nghiệm và thành phần**

* **Cảm giác vượt qua thử thách** : Khi người chơi vượt qua vật cản, trò chơi có thể sử dụng âm thanh và hoạt họa ăn mừng.
* **Cảm giác giác thất bại** : Khi người chơi va chạm vật cản hoặc mất mạng, game sẽ có họa tiết và âm thanh phản hồi mạnh mẽ như cảm giác giác thu cuộc.

# CHƯƠNG 4 CHƯƠNG TRÌNH DEMO

## 4.1. Gameplay

### 1. **Mục tiêu của trò chơi**

* **Mục tiêu chính** : Người chơi điều khiển vật vượt qua vật cản trong môi trường 3D để đạt được điểm số cao nhất hoặc hoàn thành một mức độ nhất định.
* **Cấp độ** : Game được chia thành các cấp độ (cấp độ) khác nhau, mỗi cấp độ sẽ có độ khó tăng dần, với các vật cản ngày càng phức tạp hơn.
* **Điểm số** : Người chơi kiếm điểm khi vượt qua các vật cản hoặc hoàn thành các mục tiêu như nhặt xu trong trò chơi.

### 2. **Nhân vật điều khiển**

* **Di chuyển** : Người chơi có thể điều khiển vật vật di chuyển về các hướng khác nhau (trái, phải, nhảy lên) để tránh vật cản.
* **Skip** : Nhân vật có thể nhảy để vượt qua các vật cản, tường hoặc vật cản.

### 3. **Chướng ngại vật và thử thách**

* **Vật cản** : Các vật cản có thể là tường tường, bẫy.
* **Độ khó** : Mỗi cấp độ có thể tăng độ khó bằng cách thay đổi vị trí, tốc độ hoặc số lượng vật chất.
* **Khả năng vượt qua** : Người chơi phải nhảy qua các vật cản, trượt qua các khu vực nguy hiểm hoặc thậm chí phải hoàn thành một loạt các thử thách liên tiếp mà không gặp phải lỗi.

### 4. **Điểm số và phần thưởng**

* **Cộng điểm** : Người chơi nhận điểm khi vượt qua vật cản, hoàn thành thử nghiệm hoặc hoàn thành một cấp độ.
* **Phần thưởng** : Trò chơi có thể cung cấp phần thưởng cho người chơi đạt được các thành tích, như hỗ trợ sản phẩm, nâng cấp nhân vật hoặc mở khóa cấp độ mới.

### 5. **Mất mạng và game over**

* **Mất mạng** : Khi nhân vật va chạm với một vật cản trò chơi sẽ kết thúc.
* **Game over** : Khi người chơi thua cuộc, màn hình game over sẽ hiển thị, cho phép người chơi bắt đầu lại hoặc quay lại menu chính.

### 6. **Chế độ chơi**

* **Chế độ đơn** : Người chơi cố gắng hoàn thành cấp độ một mình, vượt qua chướng ngại vật để đạt được điểm số cao nhất.

## 4.2. Các chức năng cơ bản:

· **Di chuyển vật** : Điều khiển vật chuyển và nhảy qua chướng ngại vật.

· **Va và tương tác** : Xử lý va chạm giữa vật và vật cản, quyết định kết quả khi va phải (mất mạng, game over).

· **Điểm số** : Tính điểm dựa trên thời gian hoàn thành hoặc số lượng vật cản vượt qua, hiển thị trên giao diện.

· **Kết thúc màn chơi** : Hiển thị trò chơi trên màn hình kết thúc hoặc chiến thắng khi hoàn thành màn chơi hoặc thất bại.

· **Hiệu ứng âm thanh và hình ảnh** : Thêm âm thanh và hiệu ứng hình ảnh để tăng trải nghiệm cho người chơi.

· **Tạo và quản lý vật cản** : Sinh ra các vật cản trong game, thay đổi độ khó qua cấp độ.

· **Người dùng giao diện (UI)** : Hiển thị các thông tin quan trọng như số, mạng sống và các tùy chọn trong trò chơi.

· **Lưu trữ tiến trình** : Cho phép người chơi lưu và tải lại trò chơi để tiếp tục chơi.

# CHƯƠNG 5

# KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## 5.1 Kết luận

**Kết luận** , phát triển một game vượt chướng ngại vật 3D trong Unity là một quy trình phức tạp, bao gồm nhiều giai đoạn từ ý tưởng ban đầu cho đến khi hoàn thiện và xuất bản. Trong suốt quá trình này, các yếu tố như thiết kế di chuyển nhân vật, va chạm, tính điểm và kết thúc màn chơi đều đóng vai trò quan trọng để tạo nên một trò chơi vị trí thú vị và thử thách cho người chơi.

Việc hiểu rõ quy trình phát triển trò chơi sẽ giúp bạn chủ động hơn trong việc lập kế hoạch và phát triển từng bước. Các công thức trong quy trình thiết lập, thiết kế môi trường và tối ưu hóa sẽ yêu cầu sự sáng tạo và kỹ năng, nhưng với sự chăm chỉ và hiển thị, bạn có thể tạo ra một trò chơi hấp dẫn, mang lại niềm vui cho người chơi và phần nâng cao giải trí trải nghiệm.

Cuối cùng, để duy trì sức hút cho game, việc liên tục cải tiến, cập nhật và lắng nghe phản hồi từ cộng đồng người chơi là rất quan trọng, giúp game không chỉ hoàn thiện trong từng chi tiết mà còn phát triển lâu dài .

# CHƯƠNG 6 TÀI LIỆU THAM KHẢO – PHỤ LỤC

## 6.1. Tài liệu tham khảo

* Lập trình game 2D - Trần Doãn Hiển (Youtube)
* Lập trình Unity căn bản – HowKteam
* Lập trình Unity nâng cao – Kim Long
* After effect cơ bản – Nhiều nguồn
* Learning C# Programming with Unity 3D - Alex Okita
* Tài liệu lập trình C# - Đại học Nam Cần Thơ , J2team and WebSite to Internet
* Lập trình hướng đối tượng – Cô Bùi Thị Diễm Trinh
* Unity Learn Tutorials
* Lịch sử ra đời Unity – Nhiều nguồn
* Thống kê sử dụng unity – Nhiều nguồn
* Khóa học các công cụ của Adobe – DayNhauHoc, HowKteam,KhoaPham,CodeDao, U dev, nguồn trên Tiktok,….

## 6.2 Phụ lục

· **Probuilder:** Plugin hỗ trợ xây dựng nhanh các đối tượng 3D cơ bản, giúp thiết kế môi trường và chướng ngại vật trong Unity.

· **Obstacle (Chướng ngại vật):** Các đối tượng trong game cản trở người chơi, như: tường, hố sâu, bẫy, hoặc vật thể di chuyển.

· **Character Controller:** Công cụ Unity giúp lập trình hành vi nhân vật, như di chuyển, nhảy, né tránh chướng ngại vật.

· **Unity Asset Store:** Kho tài nguyên hỗ trợ thiết kế nhanh các vật thể, nhân vật, hoặc hiệu ứng cho game.