**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

Xin chân thành cảm ơn các thầy cô trường Đại học Trà Vinh, đặc biệt là các thầy cô khoa Kỹ thuật & Công nghệ đã tạo cơ hội cho chúng em được thực tập và giao lưu. Tôi có thể tránh được các vấn đề và nhầm lẫn trong môi trường làm việc trong tương lai. Đặc biệt xin cảm ơn thầy Trịnh Quốc Việt. Nhờ sự giúp đỡ và hướng dẫn tận tình của thầy từ đầu đến cuối đồ án, em đã hoàn thành đồ án đúng thời hạn và tích luỹ được một vốn kiến thức quý báu. Mặc dù đã cố gắng hết sức để hoàn thành đề tài này nhưng do hạn chế về thời gian cũng như kiến thức chuyên môn nên trong quá trình nghiên cứu, đánh giá và trình bày đề tài, em còn nhiều thiếu sót. Rất mong được sự quan tâm, góp ý của các thầy cô bộ môn để khóa luận được hoàn thiện và hoàn thiện hơn. Xin chân thành cảm ơn.

Mục Lục

[CHƯƠNG 1: TỔNG QUAN 10](#_Toc22761)

[1.1. Tổng quan về Unity 10](#_Toc1690)

[1.1.1. Khái niệm 10](#_Toc26709)

[1.1.2. Lịch sử hình thành và phát triển 10](#_Toc10349)

[1.1.3. Các tính năng cơ bản. 11](#_Toc1856)

[1.2. Tổng quan về game thể loại Endless runner. 11](#_Toc24742)

[1.2.1. Lối chơi. 11](#_Toc14662)

[1.2.2. Đồ họa. 11](#_Toc28059)

[1.2.3. Âm thanh. 11](#_Toc4075)

[1.2.4. Tổng quan về game được thiết kế trong đồ án. 11](#_Toc15680)

[1.3. Tổng quan về Visual Studio và C# 12](#_Toc16871)

[1.3.1. Visual Studio. 12](#_Toc29106)

[1.3.2. C#. 12](#_Toc15871)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 13](#_Toc18236)

[2.1. Các khái niệm trong Unity 13](#_Toc7560)

[2.1.1. Project 13](#_Toc20299)

[2.1.2. Assets (tài nguyên) 14](#_Toc15357)

[2.1.3. Scenes (cảnh) 15](#_Toc20520)

[2.1.4. Game Object (Đối tượng game) 15](#_Toc9299)

[2.1.5. Packages( Gói ) 15](#_Toc22395)

[2.1.6. Component ( Thành phần ) 16](#_Toc18935)

[2.1.7. Script ( Mã nguồn ) 16](#_Toc15837)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 21](#_Toc29325)

[3.1. Tạo và cấu hình project và cảnh Bắt đầu (Start). 21](#_Toc3077)

[3.1.1. Tạo project. 21](#_Toc7924)

[3.1.2. Tạo Game Object Canvas để chứa nút bắt đầu. 23](#_Toc30292)

[3.1.3. Tạo Game Object nút bắt đầu ( Start\_button) 27](#_Toc29977)

[3.1.4. Tạo Game Object EventSystem và BGroound 30](#_Toc2509)

[3.2. Cảnh Game Play (GamePlay Scenes) 30](#_Toc31126)

[3.2.1. Main Camera và EventSystem 31](#_Toc15803)

[3.2.2. ScriptPause 31](#_Toc7998)

[3.2.3. Background 31](#_Toc30608)

[3.2.4. ButtonANDPause 33](#_Toc9210)

[3.2.5. SpawnBot 36](#_Toc15048)

[3.2.6. SpawnTop 37](#_Toc11690)

[3.2.7. T-Rex (Nhân vật). 40](#_Toc19667)

[3.3. Cảnh thua (Die Scenes) 43](#_Toc28384)

[3.4. Cấu hình khi Build sang Apk 43](#_Toc5454)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 44](#_Toc6838)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 45](#_Toc1305)

[DANH MỤC TÀI LIỆU THAM KHẢO 46](#_Toc26636)

**DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

Hình 1.1: Visual Studio.

Hình 1.2: Minh họa quá trình biên dịch của 1 chương trình.

Hình 1.3: Project hiển thị trong Unity Hub.

Hình 1.4: Project hiển thị trong Untity Editor.

Hình 1.5: Minh họa về Asset.

Hình 1.6: Minh họa về cấp bậc của GameOject.

Hình 1.7: Minh họa về Component.

Hình 1.8: Sơ đồ lớp Monobehaviour

Hình 1.9: Initialization trong sơ đồ lớp Monobehaviour

Hình 1.10: Physics trong sơ đồ lớp Monobehaviour

Hình 2.1: Điều chỉnh Fixed Timestep trong Project Settings

Hình 2.2: Game Logic trong sơ đồ lớp Monobehaviour

Hình 2.3: Decommissioning trong sơ đồ lớp Monobehaviour

Hình 2.4: Lựa chọn templates khi tạo project trong Unity Hub

Hình 2.5: Main camera của cảnh Start trong Unity Editor

Hình 2.6: Các Component của Main Camera

Hình 2.7: Cấu hình Transform của Main Camera

Hình 2.8: Cấu hình Camera của Main Camera

Hình 2.9: Tạo Canvas cho cảnh.

Hình 2.10: Các Component của Canvas.

Hình 3.1: Cấu hình của Rect Transform.

Hình 3.2: Cấu hình của Canvas.

Hình 3.3: Cấu hình của Canvas Scaler.

Hình 3.4: Cấu hình của Graphic Raycaster.

Hình 3.5: Mã nguồn Clickk.

Hình 3.6: Cấu hình Audio Source.

Hình 3.7: Tạo nút bắt đầu .

Hình 3.8: Các Component của Start\_buuton .

Hình 3.9: Cấu hình Canvas Renderer và Image.

Hình 3.10: Cấu hình Button.

Hình 4.1: Sơ đồ điều khiển của Animator.

Hình 4.2: Các Component của EventSystem.

Hình 4.3: GameObject BGround.

Hình 4.4: Các Game Object chính trong cảnh.

Hình 4.5: Các Component của ScriptPause.

Hình 4.6: Các Game Object con của Background.

Hình 4.7: Các Game Object hình ảnh Background và Background0.

Hình 4.8: Script của Background và Background0

Hình 4.9: Inspector của Background.

Hình 4.10: Các Game Object con của ButtonANDPause

Hình 5.1: Script Point

Hình 5.2: Cấu hình và sự kiện OnClick() của ButtonJump.

Hình 5.3: Hàm được thực thi khi người chơi chạm vào ButtonJump.

Hình 5.4: Cấu hình Floating Joystick.

Hình 5.5: Component Button của Pause.

Hình 5.6:Mã của Script PauseMenu

Hình 5.7:SpawnBot và Component Spawn.

Hình 5.8:Mã của Script Spawn.

Hình 5.9:Mã của Script Collision\_Box.

Hình 5.10:Script Bay trong Inspector của SpawnTop.

Hình 6.1:Mã của Script Bay .

Hình 6.2:GameObject 0 và các Component.

Hình 6.3:Các hàm xử lý duy chuyển cho game object.

Hình 6.4:Hàm xử lý tăng máu cho game object.

Hình 6.5:Game Objcet T-Rex và các Component.

Hình 6.6:Mã của Script Player.

Hình 6.7:Các component của Gun 1

Hình 6.8: Mã của Script Gun.

Hình 6.9: Các component của bullet .

Hình 6.10: Mã của Script collision\_handling .

Hình 7.1: Mã của Script DestroyerBullet.

Hình 7.2: Các Game Object của cảnh Die.

Hình 7.3: Cửa sổ Project Settings.

Hình 7.4: Cảnh bắt đầu.

Hình 7.5: Cảnh vào chơi .

Hình 7.6: Cảnh thất bại .

**TÓM TẮT NIÊN LUẬN ĐỒ ÁN CƠ SỞ NGÀNH**

**Vấn đề nghiên cứu:**

* Thiết kế một trò chơi 2D, cụ thể là một trò chơi thể loại endless runner với đồ họa 2D bằng Unity Engine một nền tảng phát triển game phổ biến hiện nay.
* Với mục tiêu là tạo ra một trò chơi mang thú vị mang lại cho người chơi cảm giác sôi động và thú vị, đặc biệt với thể loại endless runner người chơi phải vượt qua các chướng ngại vật liên tục và chinh phục điểm số cao.

**Hướng tiếp cận:**

* Sử dụng Unity để phát triển trò chơi 2D với các tính năng đa dạng và và vô cùng đầy đủ cho phát triển trò chơi như: quản lý cảnh, âm thanh, đồ họa, chuyển động,…
* Sử dụng Visual Studio với ngôn ngữ C# để tạo và tích hợp các yếu tố game play như chuyển cảnh, nhảy, duy chuyển của các đối tượng trong game và nhiều yếu tố khác,...
* Tạo ra đa dạng các đối tượng trong game để tăng tính sinh động và thách thức.

**Cách giải quyết vấn đề:**

* Nghiên cứu các game thuộc thể loại endless runner đã phát hành hiện nay để hiểu rõ hơn về thể loại game này và các cách thức nó vận hành.
* Sử dụng các kỹ thuật lập trình Unity phù hợp như: Lập trình hướng đối tượng, lập trình 2D, vật lý,… với ngôn ngữ C# để tạo ra các xử lý và chức năng trong game một cách dễ dàng. Xây dựng một cấu trúc game linh hoạt, thuận tiện cho việc cập nhật các tính năng mới sau này.
* Sử dụng các công cụ thiết kế đồ họa 2D: Nhầm tạo ra các tài nguyên hình ảnh đẹp và chất lượng cho game một cách trao chuốt và tỉ mỉ.

**Kết quả đạt được:**

* Hoàn thành một trò chơi 2D với thể loại endless runner được phát triển trên nền tảng Unity. Trò chơi có giao diện thân thiện, lối chơi, âm thanh thú vị chạy trên nền tảng Android.
* Có được kinh nghiệm về kiến thức cũng như kỹ năng thiết kế game bằng Unity. Trải nghiệm được quá trinh xây dựng hoàn chỉnh một tựa game từ việc lập kế hoạch, thiết kế, học tập, thử nghiệm,..

**MỞ ĐẦU**

**Lý do chọn đề tài:**

* Unity là một trong những công cụ phát triển game phổ biến và mạnh mẽ nhất trên thế giới. Sự phổ biến này mang lại nhiều lợi ích, giúp tiếp cận công nghệ hàng đầu và phát triển kỹ năng cần thiết cho ngành công nghiệp game. Cộng đồng Unity Developer trên thế giới và trong nước phát triển vô cùng mạnh.
* Thị trường game ngày càng chú trọng vào cảm giác trải nghiệm người chơi, và game 2D vẫn giữ được sức hấp dẫn lớn. Lựa chọn đề tài này giúp học được cách thiết kế game 2D, nắm vững các nguyên lý của nó. Thể loại endless runner là một trong những thể loại game được yêu thích vì tính đơn giản và thách thức cao. Cơ hội thực hành và nắm bắt được các nguyên tắc quan trọng trong thiết kế gameplay.

**Mục đích:**

* Áp dụng kiến thức đã học từ các môn cơ sở ngành vào một dự án thực tế và ứng dụng nó trong việc phát triển một trò chơi 2D trên nền tảng Unity.
* Phát triển kỹ năng thực hành trong lĩnh vực thiết kế game, bao gồm cả lập trình, thiết kế đồ họa, và quản lý dự án.
* Hiểu rõ quy trình phát triển một sản phẩm game từ khâu lập kế hoạch, thiết kế, lập trình, đến kiểm thử và triển khai.
* Tạo cơ hội tìm hiểu, học tập và ứng dụng kỹ thuật phát triển game trên Unity, một trong những công cụ phổ biến nhất trong ngành công nghiệp game.

**Đối tượng và phạm vi nghiên cứu:**

* Unity là một công cụ phát triển game mạnh mẽ và phổ biến. Unity cung cấp nhiều tính năng và công cụ giúp bạn phát triển game 2D một cách dễ dàng và hiệu quả.
* Tập trung vào quy trình thiết kế và phát triển game 2D sử dụng Unity, từ việc xây dựng gameplay đến tối ưu hóa hiệu suất. Trò chơi 2D trong thể loại endless runner, với các yếu tố như điều khiển nhân vật, tạo môi trường game đa dạng và tạo ra sự thách thức liên tục. Thực hiện các biện pháp tối ưu hóa để đảm bảo trò chơi chạy mượt mà trên thiết bị Android.

# **CHƯƠNG 1: TỔNG QUAN**

## Tổng quan về Unity

### 1.1.1. Khái niệm

* [Unity](https://unity.com/) là một game engine đa nền tảng được phát triển bởi công ty Unity Technologies, chủ yếu được dùng để phát triển video game cho 21 nền tảng bao gồm máy tính, game consoles (như PlayStation) và điện thoại,…
* Hơn [50% số lượng game trên thị trường](https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown) được sản xuất bởi Unity. Một vài tựa game vô cùng nổi tiếng được tạo ra bởi Unity có thể được kể đến như Pokémon Go, Hearthstone, Ori And The Blind Forest, Monument Valley, Axie Infinity,… Độ “phủ sóng” của Unity rất rộng, có thể được áp dụng phổ biến trong nhiều dòng game khác nhau từ game “hạng nặng” Triple A (AAA) cho đến game giáo dục đơn giản cho con nít.

### 1.1.2. Lịch sử hình thành và phát triển

Giai đoạn đầu (2005-2010)

* Unity được thành lập bởi David Helgason, Joachim Ante, và Nicholas Francis vào năm 2004. Họ bắt đầu phát triển Unity như một dự án phụ trong khi vẫn làm việc tại công ty cũ của họ, Web3D.
* Vào năm 2005, Unity được phát hành lần đầu tiên với tên Unity3D. Phiên bản này chỉ hỗ trợ phát triển game 3D trên Windows.
* Trong giai đoạn này, Unity chủ yếu được sử dụng bởi các nhà phát triển game độc lập. Tuy nhiên, Unity đã bắt đầu thu hút sự chú ý của các nhà phát triển game lớn.

Giai đoạn phát triển (2010-2015)

* Trong giai đoạn này, Unity đã phát triển nhanh chóng. Unity đã bổ sung hỗ trợ cho nhiều nền tảng mới, bao gồm Mac, Linux, iOS, Android, và PlayStation 3.
* Unity cũng đã bổ sung nhiều tính năng mới, bao gồm hỗ trợ cho vật lý, AI, và đồ họa nâng cao.
* Trong giai đoạn này, Unity đã trở thành một công cụ phát triển game phổ biến. Unity được sử dụng để phát triển nhiều trò chơi nổi tiếng, bao gồm Temple Run, Zombies Run, và Ori and the Blind Forest.

Giai đoạn hiện tại (2015-hiện nay)

* Trong giai đoạn này, Unity tiếp tục phát triển và cải tiến. Unity đã bổ sung hỗ trợ cho nhiều nền tảng mới, bao gồm PlayStation 4, Xbox One, và Nintendo Switch.
* Unity cũng đã bổ sung nhiều tính năng mới, bao gồm hỗ trợ cho thực tế ảo và thực tế tăng cường.
* Trong giai đoạn này, Unity đã trở thành một công cụ phát triển game hàng đầu. Unity được sử dụng để phát triển nhiều trò chơi nổi tiếng, bao gồm Fortnite, Among Us, và Animal Crossing: New Horizons.

### 1.1.3. Các tính năng cơ bản.

* Tạo giao diện UI của Game như tạo drop bar, textbox,…
* Hiển thị mô hình 3D, 2D – hệ thống vật lý 2D, 3D
* Networking: Hỗ trợ tạo game chơi nhiều người cùng lúc
* Hỗ trợ nền tảng đặc biệt mới: AR – Augmented reality (Thực tế tăng cường), VR – Virtual reality (Thực tế ảo)
* AI – Hỗ trợ con bot trong màn hình game, hỗ trợ package tạo nên bot trong game
* Hỗ trợ hiển thị chữ bằng font đặc biệt
* …

## 1.2. Tổng quan về game thể loại Endless runner.

* Thể loại Endless runner là một thể loại game điện tử trong đó người chơi điều khiển nhân vật của họ chạy liên tục về phía trước, tránh các chướng ngại vật và kẻ thù. Thể loại này thường có nhịp độ nhanh và đòi hỏi sự phản xạ nhanh nhạy của người chơi.

### 1.2.1. Lối chơi.

* Lối chơi là yếu tố quan trọng nhất của bất kỳ trò chơi nào, đặc biệt là trò chơi Endless Runner. Lối chơi của trò chơi sẽ được thiết kế sao cho hấp dẫn và gây nghiện, đồng thời phù hợp với thể loại Endless Runner.

### 1.2.2. Đồ họa.

* Đồ họa là yếu tố quan trọng thứ hai của trò chơi. Đồ họa của trò chơi sẽ được thiết kế sao cho đẹp mắt và bắt mắt, đồng thời phù hợp với phong cách của trò chơi.

### 1.2.3. Âm thanh.

* Âm thanh là yếu tố quan trọng thứ ba của trò chơi. Âm thanh của trò chơi sẽ được thiết kế sao cho sống động và hấp dẫn, đồng thời phù hợp với lối chơi của trò chơi.

### 1.2.4. Tổng quan về game được thiết kế trong đồ án.

* Game được thiết kế với 3 cảnh : Bắt đầu, chơi , kết thúc.
* Khi vào game người chơi sẽ được đưa đến cảnh bắt đầu với 1 nút “ Play “ khi bấm vào nút người chơi sẽ được chuyển đến cảnh “ chơi“
* Ở cảnh chơi này người chơi điều khiển nhân vật nhảy lên để tránh các chướng ngại vật bên dưới đang duy chuyển tới, đồng thời người chơi cũng phải điều khiển thêm súng đã trang bị để bắn các chướng ngại vật đang bay đến bên trên. Nếu người chơi chạm vào bất khì chướng ngại vật nào thì sẽ thất bại.
* Khi thất bại người chơi sẽ chuyển đến cảnh “ kết thúc” với 2 lựa chọn là chơi lại hoặc rời trò chơi.

## 1.3. Tổng quan về Visual Studio và C#

### 1.3.1. Visual Studio.

* Visual studio là một phần mềm hỗ trợ đắc lực hỗ trợ công việc lập trình website. Công cụ này được tạo lên và thuộc quyền sở hữu của ông lớn công nghệ Microsoft. Năm 1997, phần mềm lập trình nay có tên mã Project Boston. Nhưng sau đó, Microsoft đã kết hợp các công cụ phát triển, đóng gói thành sản phẩm duy nhất.

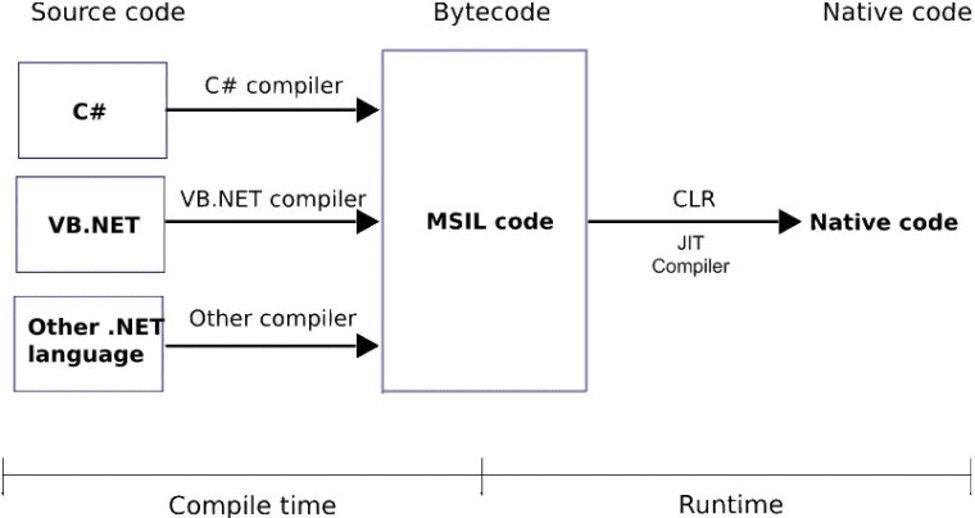


*Hình 1.1: Visual Studio*.

* Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sửa mã, trình thiết kế, gỡ lỗi. Tức là,có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm Visual Studio mà thôi. Không dừng lại ở đó, người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend vậy.

### 1.3.2. C#.

* C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
* Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.  
  Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).
* Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.

*Hình 1.2: Minh họa quá trình biên dịch của 1 chương trình*.

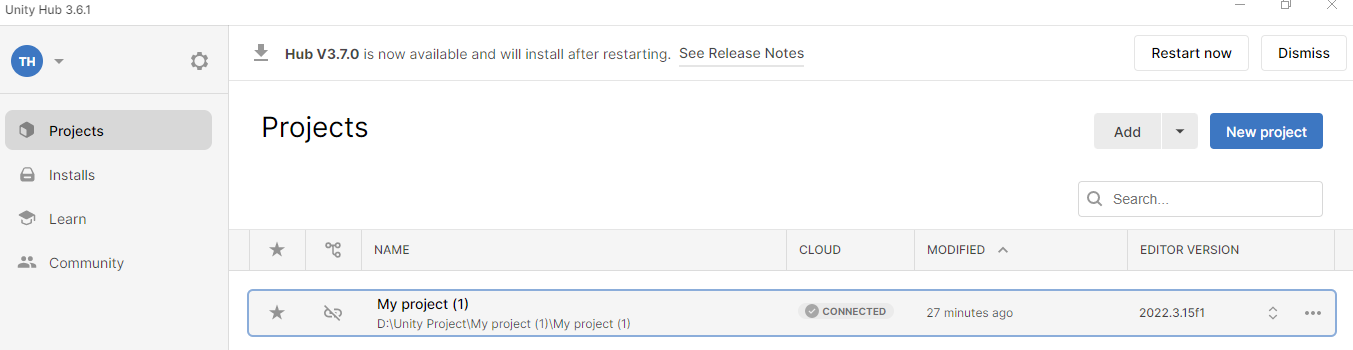
C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

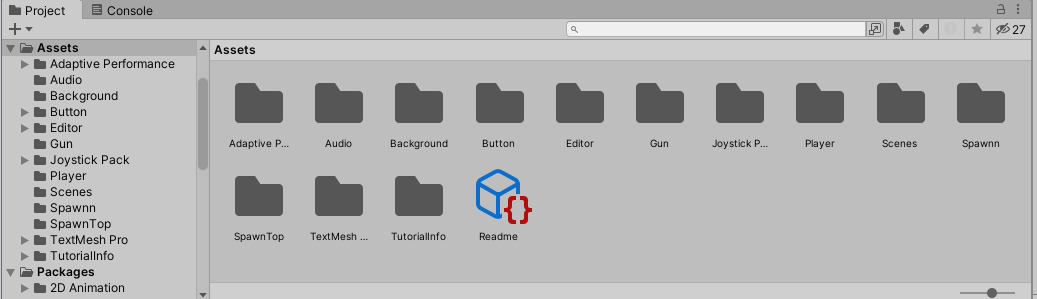
## 2.1. Các khái niệm trong Unity

### 2.1.1. Project

* Khái niệm Project bao trùm toàn bộ các thành phần làm nên một game: các models, assets, scripts, scenes,…. Projects được cấu trúc phân tầng tương tự với cấu trúc file và folder trong máy.



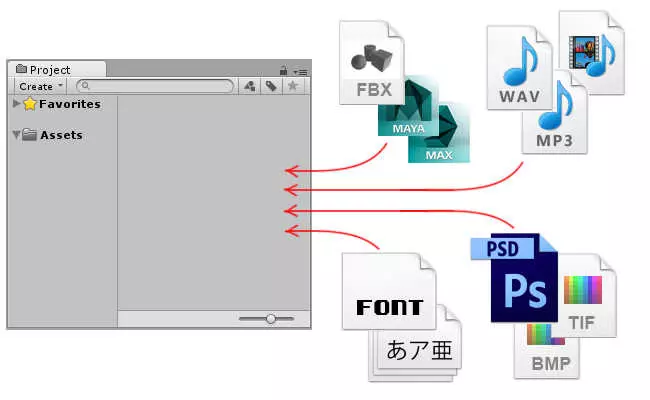
*Hình 1.3: Project hiển thị trong Unity Hub*.



*Hình 1.4: Project hiển thị trong Untity Editor.*

### 2.1.2. Assets (tài nguyên)

* Một asset là bất kỳ tài nguyên nào: âm thanh, texture (hình ảnh), model (vật thể 3D)… mà bạn cần để cấu thành một GameObject.



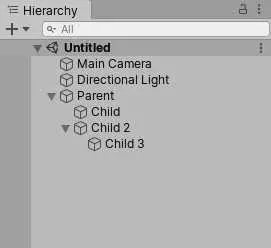
*Hình 1.5: Minh họa về Asset.*

### 2.1.3. Scenes (cảnh)

* Cảnh (Scenes) trong Unity là một tập hợp các đối tượng 3D, 2D, âm thanh, ánh sáng,... được bố trí trong một không gian ảo. Mỗi cảnh có thể là một màn hình, một cấp độ, một môi trường hoặc một khung cảnh trong trò chơi.
* Cảnh là một đơn vị cơ bản trong Unity, là nơi tạo và chỉnh sửa trò chơi. Có thể tạo nhiều cảnh trong một dự án Unity và chuyển đổi giữa chúng trong khi chơi trò chơi.

### 2.1.4. Game Object (Đối tượng game)

* Trong Unity, Game Object (Đối tượng game) là một khái niệm cơ bản và quan trọng trong việc xây dựng trò chơi. Một Game Object đại diện cho các thành phần cơ bản của môi trường trò chơi, như nhân vật, vật phẩm, ánh sáng, camera, terrain, và nhiều thứ khác có thể nhìn thấy hoặc không nhìn thấy trong Scenes. Điều này giúp tổ chức và quản lý các yếu tố khác nhau trong một trò chơi một cách có tổ chức.
* Mỗi Game Object có thể chứa các thành phần (Components) khác để định nghĩa và điều khiển các tính năng cụ thể. Ví dụ, một Game Object nhân vật có thể chứa các thành phần như Mesh Renderer (để hiển thị hình dạng của nhân vật), Collider (để xác định va chạm), và Script (để xử lý logic của nhân vật) và các thành phần khác để phù hợp với yêu cầu chức năng của Game Object.
* Tất cả các game objects (kể cả game objects rỗng) đều có vị trí và độ xoay trong thế giới game, nó có thể di chuyển, xoay và scale. Lưu ý rằng các yếu tố này sẽ chịu ảnh hưởng bởi cấp bậc như hình.



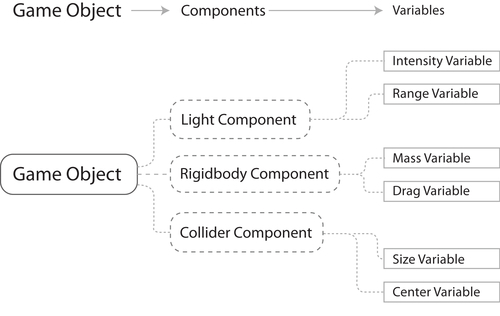
*Hình 1.6: Minh họa về cấp bậc của GameOject.*

### 2.1.5. Packages( Gói )

* Một package là tập hợp các game objects, assets và các file meta-data liên quan. Bạn có thể nghĩ package tương tự như một một package thư viện trong Java vậy.
* Chúng là các object liên quan với nhau: models, scripts, materials, etc,…:
* Một tập các shaders để hỗ trợ hiệu ứng hình ảnh.
* Các prefabs hiệu ứng particle systems.
* Các models xe cho game đua xe.
* Các models cây và vật cản,…
* Unity cung cấp một số “standard package” miễn phí trên [Asset Store](https://assetstore.unity.com/), bạn có thể import vào trong project đang phát triển.

### 2.1.6. Component ( Thành phần )

* Component là những thành phần có chức năng riêng biệt, được gán vào một hoặc nhiều Game Object . Component bổ sung, cấu thành chức năng cho Game Object. Từ đó, giúp chúng ta xây dựng lên những Game Object theo từng yêu cầu của game.
* Mỗi script kế thừa từ lớp MonoBehaviour đều được coi là một component. Mỗi Game Object cơ bản đều được gán sẵn một component cơ bản là Transform.



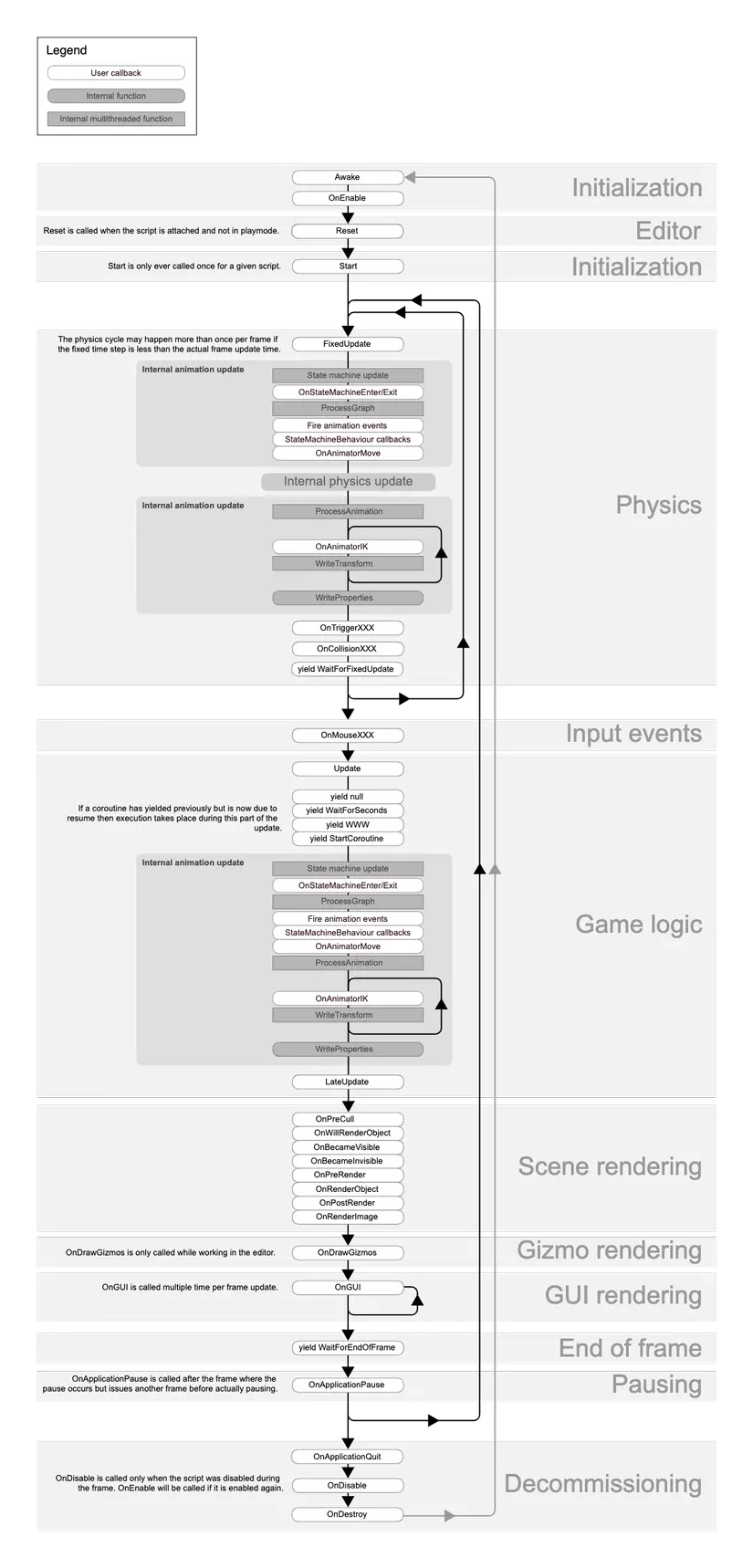
*Hình 1.7: Minh họa về Component.*

### 2.1.7. Script ( Mã nguồn )

* Script là mã nguồn (code) của chúng ta, có thể để định nghĩa một hành vi nào đó cho game objects, cũng có thể định nghĩa một chức năng nào đó, có rất nhiều loại classes, phụ thuộc và hành vi mà nó điều khiển.
* Tương tác trong game programming là loại event-driven, một script điển hình sẽ định nghĩa các functions dùng để phản ứng với các sự kiện trong game. Như hàm va chạm sẽ được gọi nếu 2 object va chạm với nhau, hàm tăng xu sẽ gọi nếu player nhặt được coin trong game,…
* Các functions của script có thể là: di chuyển một game object, tạo/ xóa các game objects, chạy function của game objects khác… Sau đó trả lại quyền điều khiển cho hệ thống Unity Engine

**2.2. Một số hàm cơ bản trong lớp MonoBehaviour.**

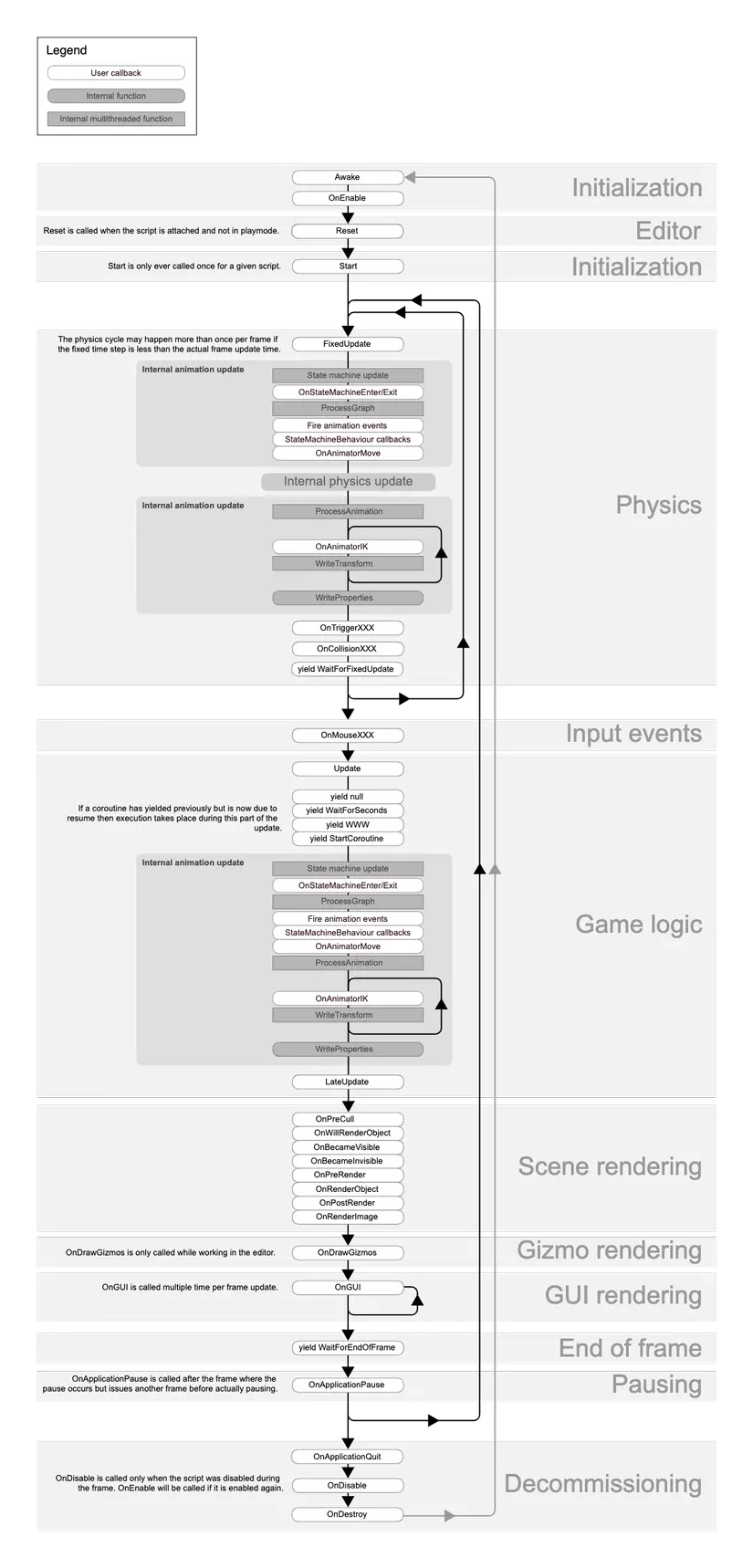
Trong Unity, MonoBehaviour là một lớp cơ bản và quan trọng trong việc lập trình các thành phần (components) của Game Object. MonoBehaviour được sử dụng để thêm các tính năng và hành vi đặc biệt vào Game Object, như là xử lý logic của trò chơi, điều khiển hoạt động và tương tác với người chơi.Mỗi script trong Unity muốn gắn vào một Game Object phải kế thừa từ lớp MonoBehaviour. Khi viết một script với MonoBehaviour, có thể triển khai các phương thức (methods) và thuộc tính (properties) đặc biệt mà Unity sẽ tự động gọi vào các thời điểm cụ thể trong quá trình thực thi của trò chơi.



*Hình 1.8: Sơ đồ lớp Monobehaviour*

**2.2.1. Initialization.**

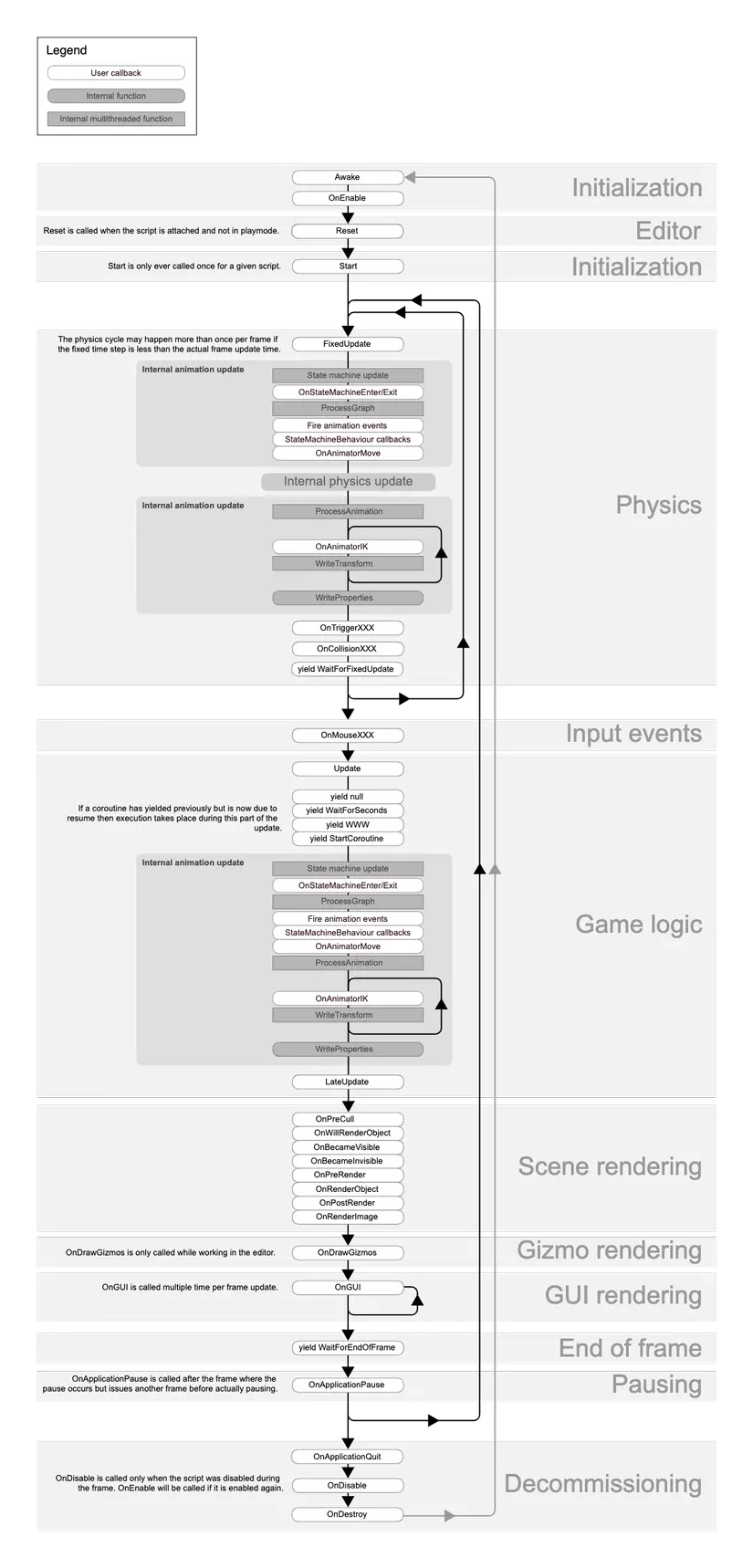
Tuy Awake, OnEnable và Start nằm cùng một phần gọi là Initialization nhưng công dụng của chúng khác nhau



*Hình 1.9: Initialization trong sơ đồ lớp Monobehaviour*

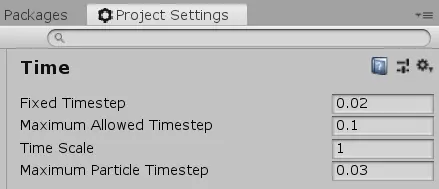
* **Awake():**
  + Các hàm Awake đều được gọi trước **tất cả** các hàm Start của các *MonoBehaviour* có trong Scene (được enable cùng lúc) hay gọi ngay sau khi gameObject đó được sinh ra bằng hàm Instantiate().
  + Awake chỉ được gọi một lần duy nhất trong vòng đời của một *MonoBehaviour*
* **OnEnable():**
  + Hàm này được gọi khi một gameObject được đổi trạng thái từ *deactive -> active* hoặc khi enable Component và được gọi lần đầu tiên ngay sau hàm Awake của nó.
* **Start():**
  + Hàm Start() được gọi sau OnEnable, trước khi các frames bắt đầu chạy hay trước các hàm Update. Cũng như Awake, nó chỉ được gọi một lần duy nhất.

**2.2.2. Physics**



*Hình 1.10: Physics trong sơ đồ lớp Monobehaviour*

* **FixedUpdate**: Hàm này được gọi không phụ thuộc vào vòng lặp chính của game mà gọi theo vòng lặp của vật lý trong game, thường nó được gọi cố định vào mỗi 0.02s theo mặc định của project setting, con số này gọi là Fixed Timestep hay fixedDeltaTime.

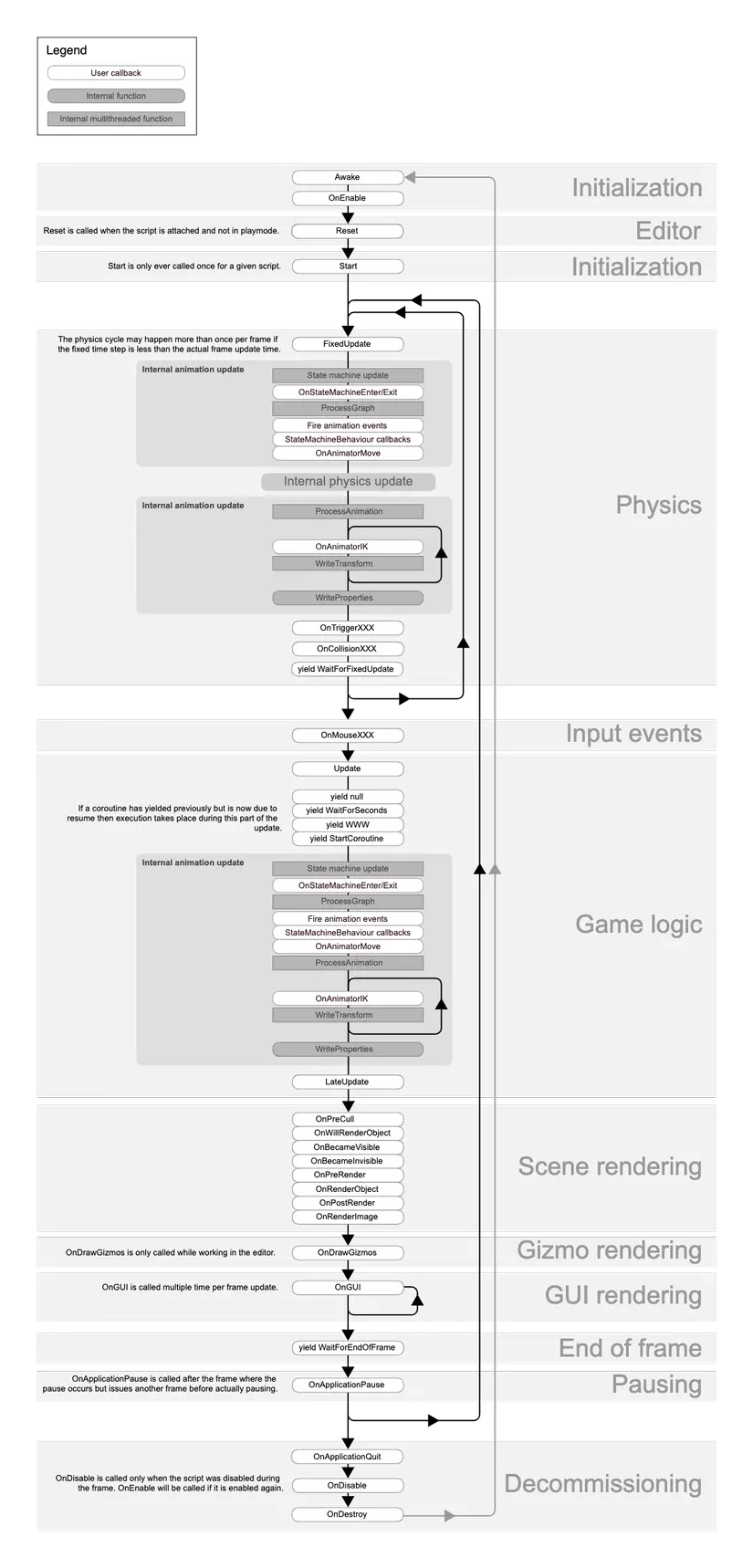


*Hình 2.1: Điều chỉnh Fixed Timestep trong Project Settings*

* **Internal Physics Update:** Đây là vùng xử lý chính của Physics System trong unity được xử lý riêng mà chúng ta không can thiệp, với một số chức năng sẵn có như áp trọng lực, trigger các hàm va chạm, di chuyển các vật thể sử dụng vật lý (rigidbody, joints,…)
* Vùng này được thực thi sau FixedUpdate, vì vậy các xử lý liên quan tới vật lý thì chúng ta nên đặt ở trong FixedUpdate thay vì Update để Physics System của Unity có thể cập nhật vị trí và xét va chạm kịp thời.

**2.2.3. Game Logic**

MonoBehaviour được gắn vào các GameObject thì StateMachineBehaviour sẽ được gắn vào các animation state và cũng có các hàm tương tự để xử lý



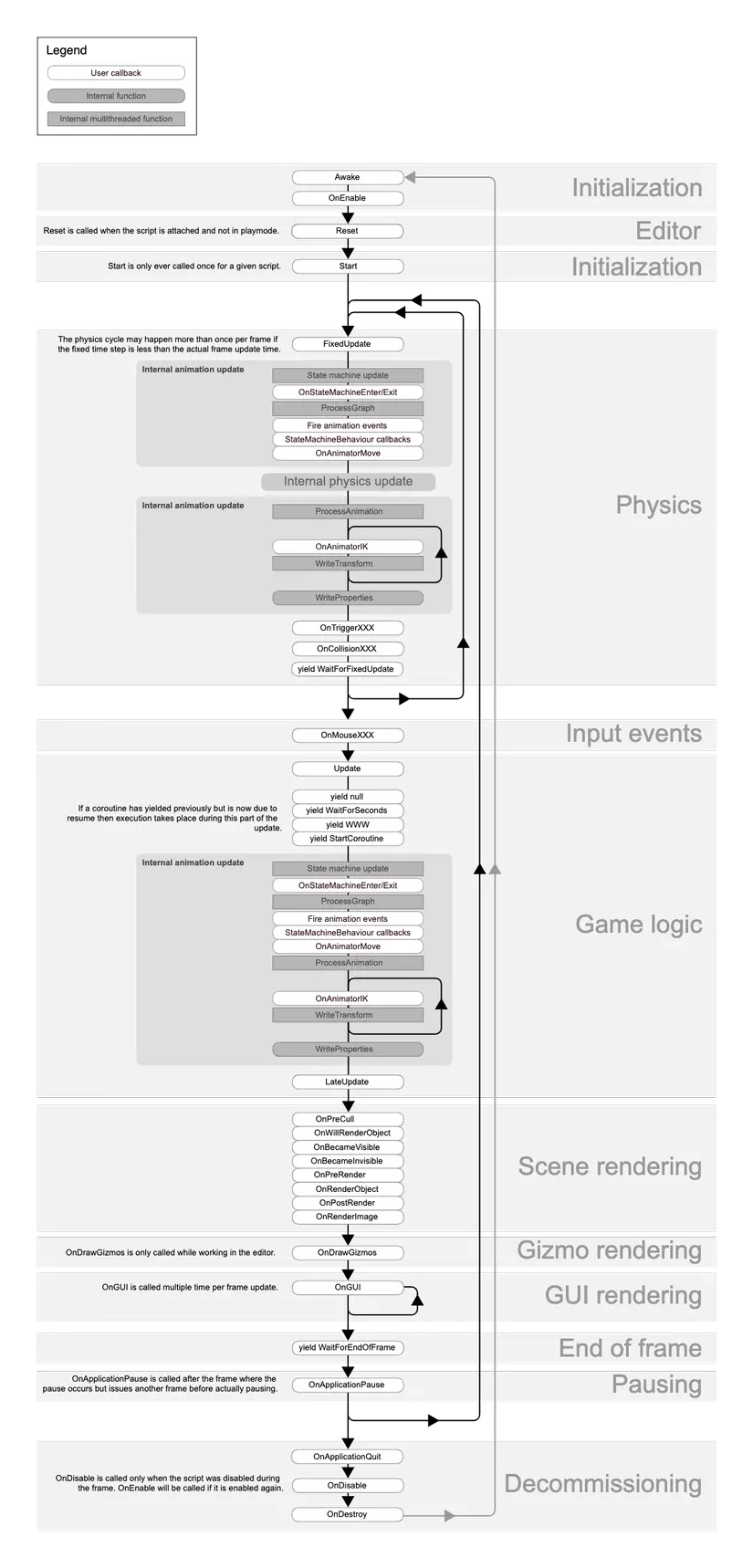
*Hình 2.2: Game Logic trong sơ đồ lớp Monobehaviour*

* **Update():** chắc hẳn hàm này đã quá quen thuộc khi mới bắt đầu sử dụng Unity engine, Update được gọi mỗi lần vào mỗi frame và được xem như hàm xử lý chính của vòng lặp game.
* **LateUpdate():** LateUpdate() được gọi như Update, mỗi frame một lần, điểm khác biệt duy nhất là nó được gọi sau khi các hàm Update của tất cả các MonoBehaviour đã được thực thi xong.

Một chức năng kinh điển của hàm này là khi sử dụng camera follow theo player, sau khi vị trí cũng như rotation của player đã được tính toán hoàn tất trong Update, thì ta sẽ LateUpdate vị trí và góc xoay của camera theo player, đảm bảo được độ chính xác cao.

**2.2.4. Decommissioning**

Decommissioning trong lớp MonoBehaviour của Unity đề cập đến quá trình tắt và dọn dẹp các thành phần và tài nguyên liên quan đến một GameObject cụ thể, chuẩn bị cho việc nó bị phá hủy và loại bỏ khỏi bộ nhớ.



*Hình 2.3: Decommissioning trong sơ đồ lớp Monobehaviour*

Đây là một bước quan trọng trong việc quản lý bộ nhớ và hiệu suất trong các trò chơi Unity, và giúp đảm bảo mọi tài nguyên được sử dụng bởi GameObject được giải phóng đúng cách để tránh rò rỉ bộ nhớ.

* **OnApplicationQuit()**: cái tên cũng nói lên được công dụng của hàm này rồi, nó sẽ được gọi trước khi thoát game (hoặc tắt Play mode trong Unity Editor)
* **OnDisable():** hàm này tương ứng với OnEnable mà mình đã đề cập ở trên, có thể được gọi nhiều lần nếu GameObject active-deactive nhiều lần (hoặc enable-disable Component gắn vào GameObject)
* **OnDestroy():** cuối cùng là OnDestroy, hàm này sẽ được gọi ở cuối frame, đồng nghĩa với việc khi xóa một GameObject trong scene (bằng script hoặc các sự kiện như thoát game, chuyển scene,…) hay sử dụng hàm **Destroy(gameObject)**, nó sẽ không xóa ngay lập tức mà phải đến cuối frame đó nó mới được dọn dẹp.

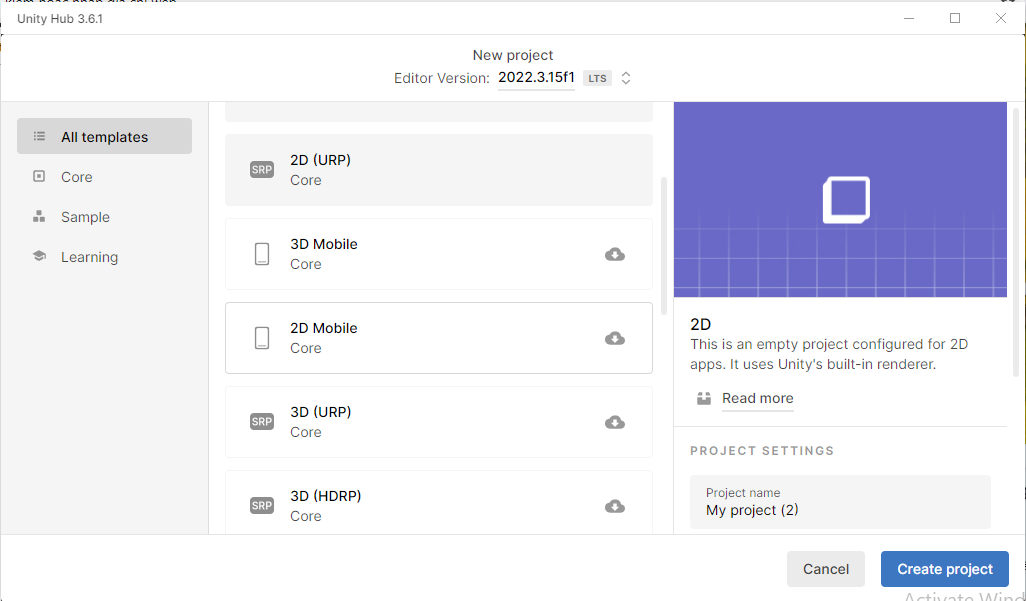
Để dọn dẹp một GameObject ngay lập tức mà không cần delay tới cuối frame, có thể sử dụng DestroyImmediate, tuy nhiên Unity khuyến cáo là chỉ nên sử dụng khi debug và không nên sử dụng trong game.

# CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

## 3.1. Tạo và cấu hình project và cảnh Bắt đầu (Start).

### 3.1.1. Tạo project.

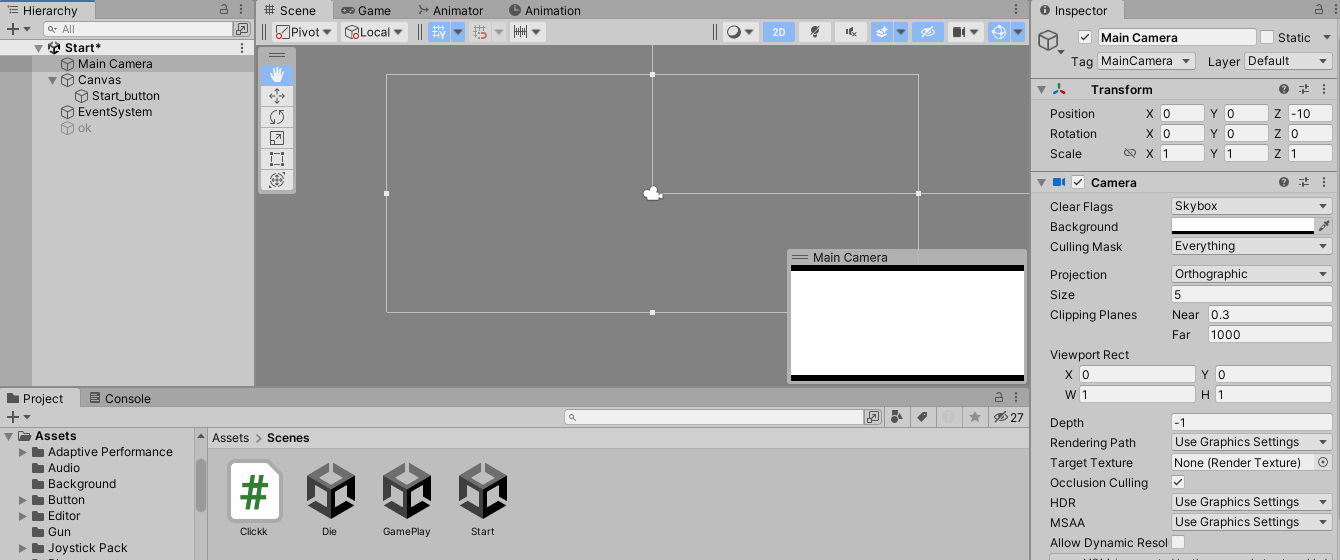
Bấm vào New project và chọn Templates cho project là 2D mobile



*Hình 2.4: Lựa chọn templates khi tạo project trong Unity Hub*

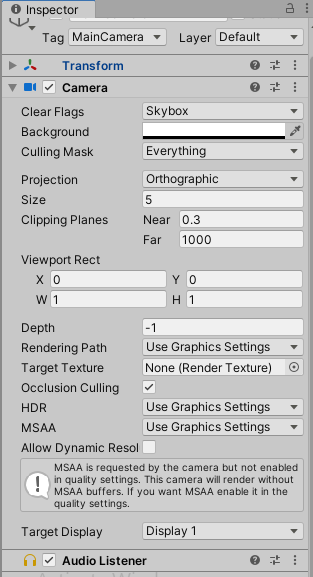
Sau khi tạo xong project và được chuyển đến Unity Editor thì sẽ được tạo sẵn cảnh (cảnh được tạo và có sẵn trong cảnh 1 (GameObject) tên là Main Camera

* Main Camera là một đối tượng (GameObject) đặc biệt được sử dụng để đại diện cho máy ảnh chính trong môi trường 3D, 2D. Camera này thường được sử dụng để xác định góc nhìn của người chơi trong trò chơi và để hiển thị các đối tượng trên màn hình.



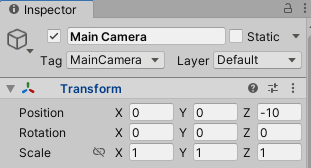
*Hình 2.5: Main camera của cảnh Start trong Unity Editor*

* Game Object Main Camera có các Component sau: Transform, Camera, Audio Listener



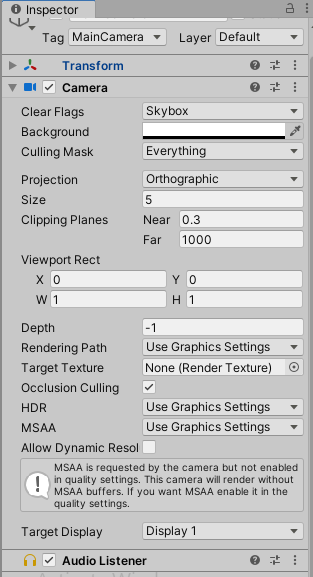
*Hình 2.6: Các Component của Main Camera*

* + **Transform** ( Component quan trong của mỗi Game Object): đại diện cho thông tin về vị trí, quay và tỷ lệ của một đối tượng trong không gian 3D hoặc 2D. Mọi đối tượng trong Unity đều có một Transform và nó là một phần quan trọng của cấu trúc dữ liệu của đối tượng.



*Hình 2.7: Cấu hình Transform của Main Camera*

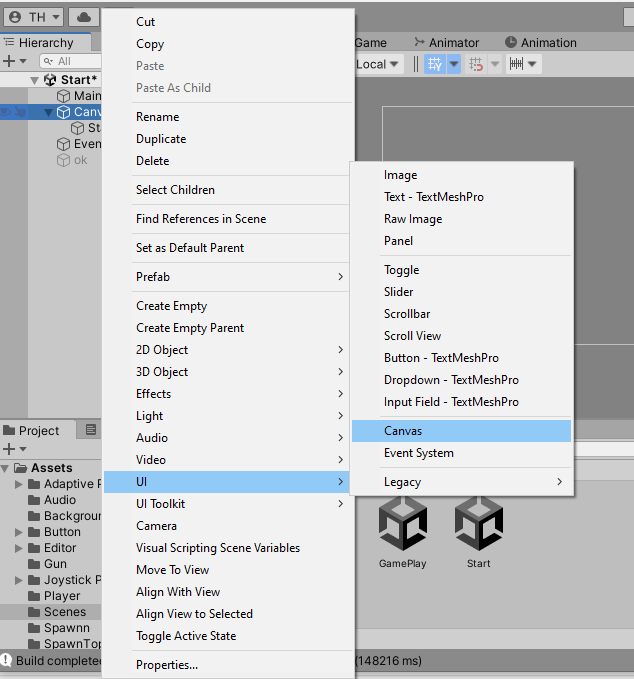
* + **Camera** : chịu trách nhiệm cho cấu hình các thuộc tính cụ thể liên quan đến cách Camera hoạt động trong môi trường 3D.



*Hình 2.8: Cấu hình Camera của Main Camera*

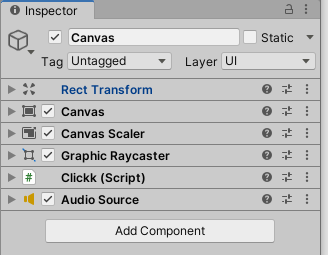
* + **Audio Listener:** cho phép camera hoặc đối tượng khác nghe âm thanh trong môi trường 3D, 2D.

### 3.1.2. Tạo Game Object Canvas để chứa nút bắt đầu.



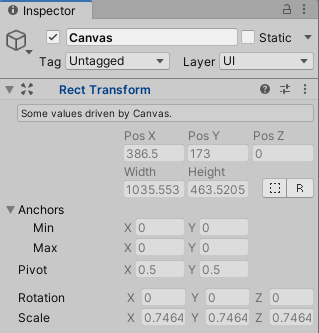
*Hình 2.9: Tạo Canvas cho cảnh.*

Các Component của Game Object Canvas: Rect Transform, Canvas, Canvas Scaler, Graphic Raycaster, Script ( Clickk), Audio Source. Trong đó ngoại trừ 2 Component là Script ( Clickk) và Audio Source được tạo và thêm vào còn lại thì được thêm mặc định khi tạo Canvas.



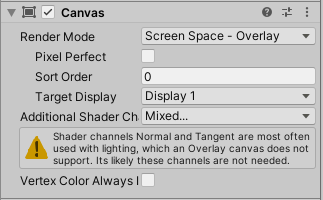
*Hình 2.10: Các Component của Canvas.*

* **Rect Transform:** để đặt và điều chỉnh vị trí và kích thước của các UI element.



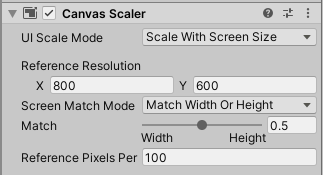
*Hình 3.1: Cấu hình của Rect Transform.*

* **Canvas :** là một container chứa tất cả các UI elementt trong một cảnh. Nó xác định vị trí của UI elementt trên màn hình và cách chúng tương tác với đối tượng khác trong scene. Sử dụng Canvas để đặt UI elementt và quyết định cách chúng xuất hiện và tương tác.



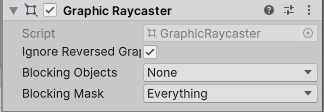
*Hình 3.2: Cấu hình của Canvas.*

* **Canvas Scaler:** giúp điều chỉnh kích thước của **Canvas** và các UI elementt trong đó để chúng có thể tự động điều chỉnh với kích thước màn hình. Có thể giúp duy trì tỷ lệ khung hình và kích thước chính xác của UI elementt khi kích thước màn hình thay đổi. Ở đây đặt Match : 0.5 để tỉ lệ của nút Bắt đầu (UI elementt) có tỉ lệ phù hợp với màn hình điện thoại.



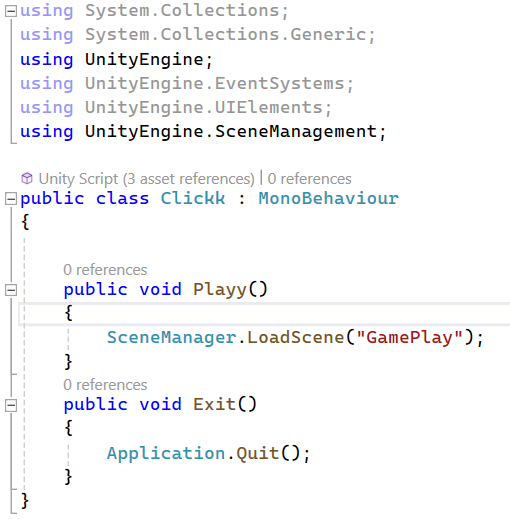
*Hình 3.3: Cấu hình của Canvas Scaler.*

* **Graphic Raycaster:** xác định UI elementt mà người chơi đang tương tác với, bằng cách sử dụng raycasting từ vị trí chuột hoặc điểm chạm.



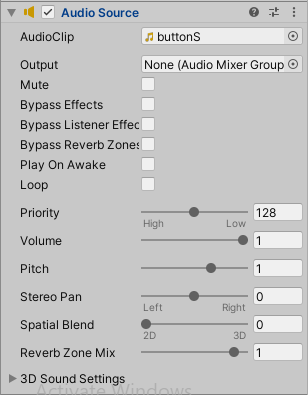
*Hình 3.4: Cấu hình của Graphic Raycaster.*

* **Script (Clickk):** Thêm Component vào trong Canvas để có thể khéo thả Script này vào sự kiện On Click của nút bắt đầu ( Button Start).



*Hình 3.5: Mã nguồn Clickk.*

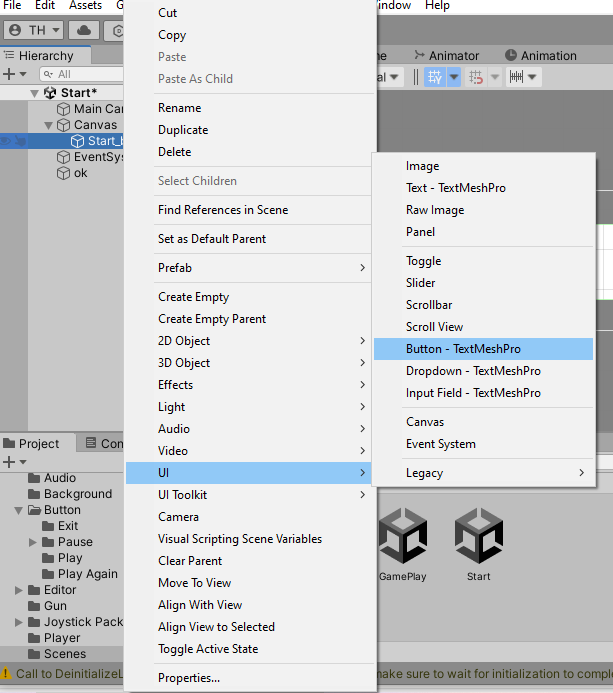
* **Audio Source:** là một component quan trọng được sử dụng để phát lại âm thanh trong trò chơi. Component này được thêm vào để dùng như script bên trên , dùng để gọi âm thanh trong On Click (). Mục AudioClip cho phép thả file Audio muốn sử dụng vào trong , ở đây là file âm thanh mỗi khi nhấn nút.



*Hình 3.6: Cấu hình Audio Source.*

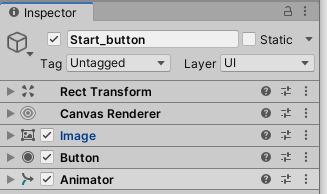
### 3.1.3. Tạo Game Object nút bắt đầu ( Start\_button)

Game Object Start\_button được đặt trong Canvas, với quan hệ Start\_button là con và Canvas là cha.



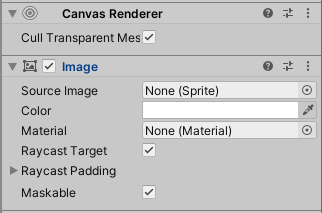
*Hình 3.7: Tạo nút bắt đầu .*

* Game Object Start\_button có các component sau: Rect Transform, Canvas Renderer, Image, Button, Animator. Với Animator là Component được thêm vào.



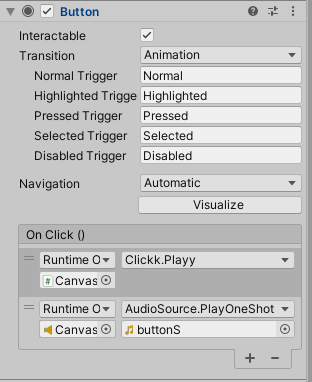
*Hình 3.8: Các Component của Start\_buuton .*

* + **Canvas Renderer và Image :** Canvas Renderer là một Component được sử dụng để hiển thị đối tượng trên Canvas. Nó chịu trách nhiệm vẽ đối tượng UI lên màn hình. Còn Image được sử dụng để hiển thị hình ảnh trên UI. Bạn có thể sử dụng nó để hiển thị biểu tượng



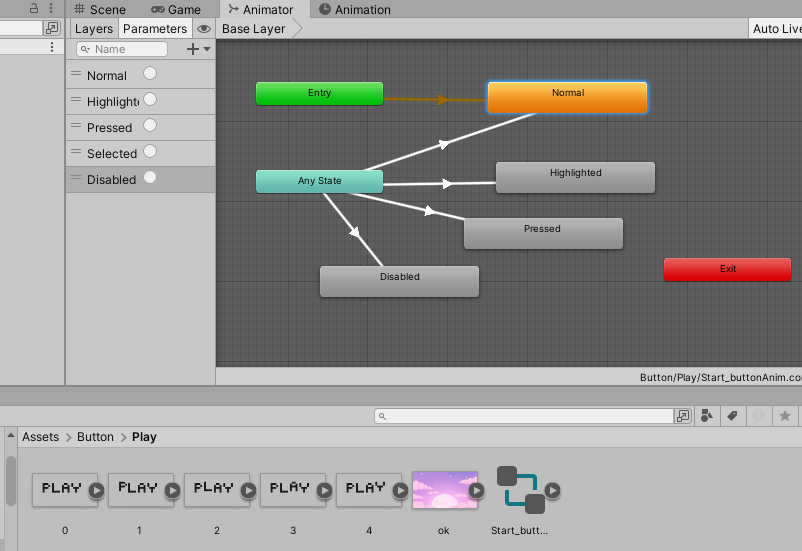
*Hình 3.9: Cấu hình Canvas Renderer và Image.*

* + **Button:** sử dụng để tạo nút tương tác trong UI, có thêm các sự kiện tương tác như nhấn, thả, và click.Trong sự kiện OnClick () được thêm Script và Audio từ Canvas để lấy hàm Playy() và Audio buttonS



*Hình 3.10: Cấu hình Button.*

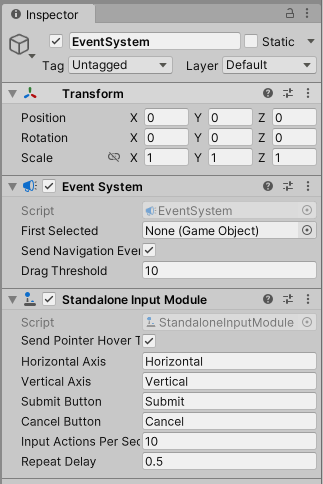
* + **Animator:** được sử dụng để tạo và điều khiển các hoạt ảnh (animations)



*Hình 4.1: Sơ đồ điều khiển của Animator.*

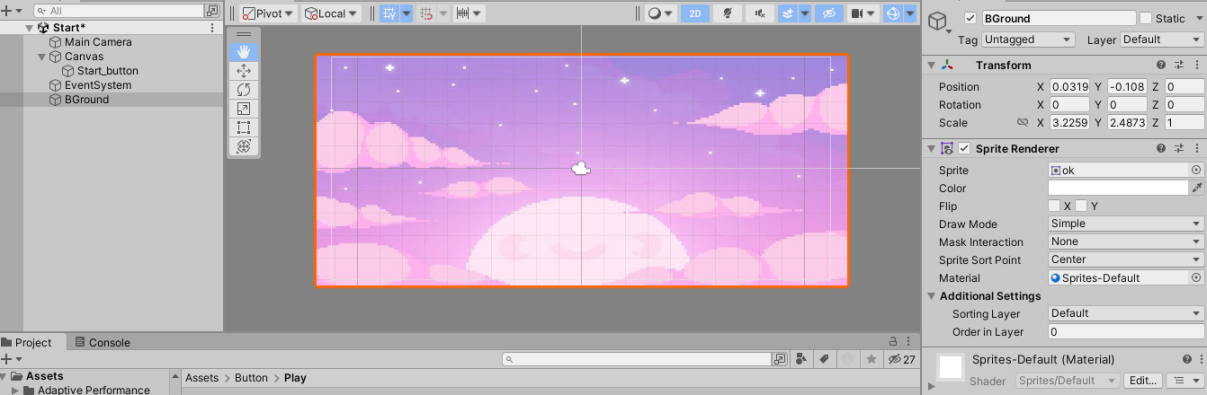
### 3.1.4. Tạo Game Object EventSystem và BGroound

* EventSystem quản lý và truyền đạt các sự kiện UI. EventSystem giúp xác định đối tượng UI nào đang được tương tác và chịu trách nhiệm gửi các sự kiện tương ứng với hành động người dùng, chẳng hạn như nhấn nút chuột, nhấn phím, chạm màn hình, và nhiều hành động khác. Khi tạo sẽ có các Component mặc định sau:



*Hình 4.2: Các Component của EventSystem.*

* **BGround:**  là hình ảnh dùng làm nền cho cảnh .



*Hình 4.3: GameObject BGround.*

Khi người chơi nhần vào nút PLAY (Start\_button) trên màn hình sẽ được dẫn đến cảnh GamePlay.

## 3.2. Cảnh Game Play (GamePlay Scenes)

Nơi ngươi chơi thực hiện các thao tác với trò chơi. Cảnh này gồm các Game Object chính sau :



*Hình 4.4: Các Game Object chính trong cảnh.*

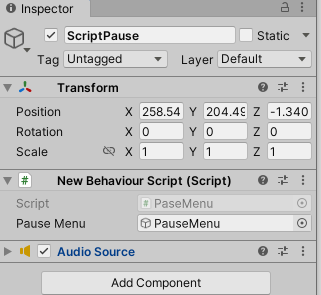
Bây giờ sẽ là giới thiệu từng Game Object có trong cảnh

### 3.2.1. Main Camera và EventSystem

NgoàiMain camera được thêm thành phần Audio Source để chứa nhạc nền cho game thì phần còn lại của 2 Game Object này giống như trên cảnh Start.

### 3.2.2. ScriptPause

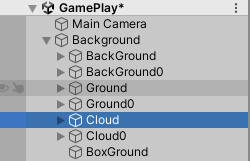
Là Game Object rỗng chứa Script PauseMenu và biến gán Game Objcet PauseMenu.



*Hình 4.5: Các Component của ScriptPause.*

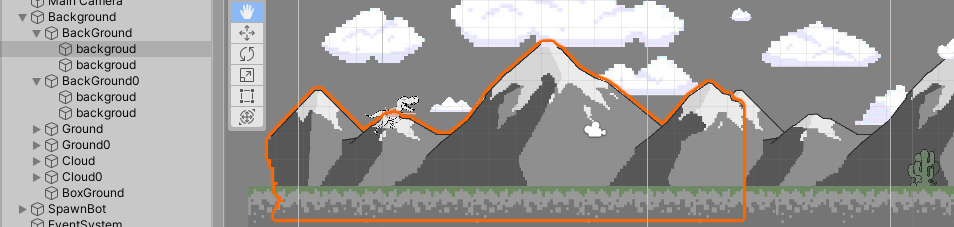
### 3.2.3. Background

Trong Game Object cha sẽ chứa 7 Game Object con khác ảnh, riêng Game Object BoxGround là Game Objcet rỗng chứa BoxColider để làm nền đất cho game thì các Game Object còn lại sẽ chứa các hình ảnh để làm ảnh nền (Back ground) và hình ảnh làm nền đất



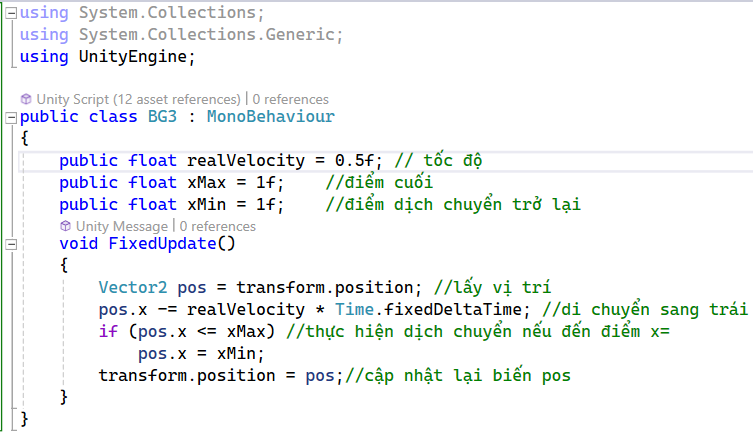
*Hình 4.6: Các Game Object con của Background.*

* **BackGround và BackGround0:** Chứa hình ảnh để làm nền cho game, các hình ảnh này sẽ được sắp xếp để tạo ra chuyển động hợp lí đem lại cảm giác nhân vật đang chạy tới.



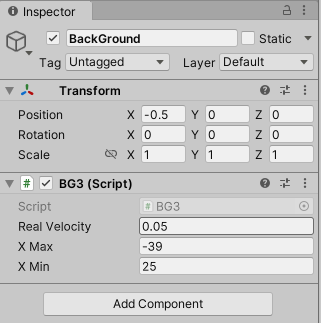
*Hình 4.7: Các Game Object hình ảnh Background và Background0.*

* + Để thực hiện chuyển động thì BackGround và BackGround0 sẽ được thêm vào Script để nó chạy qua trái và dịch chuyển trở lại để tạo cảm giác liên tục.



*Hình 4.8: Script của Background và Background0*

* + Các biến được khai báo **Public** sẽ được xuất hiện trên khung InSpector bên trong Component của Script BG3 để có thể tùy chỉnh các thông số này.

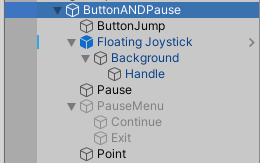


*Hình 4.9: Inspector của Background.*

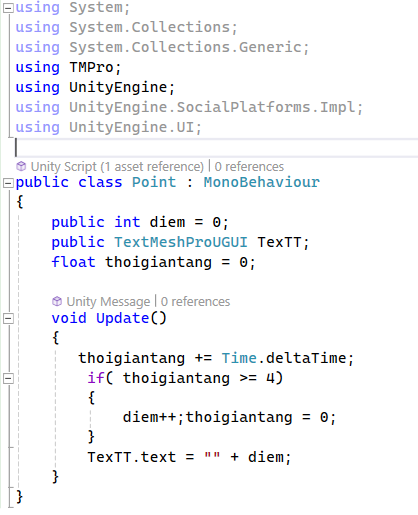
* **Ground, Ground0, Clould và Cloud0:** 4 Game Object này cũng tương tự như 2 Game Object ở trên nhưng được bố trí và các thông số trong Script BG3 sẽ được hiểu chỉnh cho phù hợp.

### 3.2.4. ButtonANDPause

Đây là một Canvas được đặt tên là “ButtonANDPause” vì nó chứa và quản lý các Button của trò chơi và Button của menu Pause (tạm dừng) và một Text (Point) để hiển thị điểm. Trong Component chứa Script Point để tính điểm và hiển thị qua Text (Point).

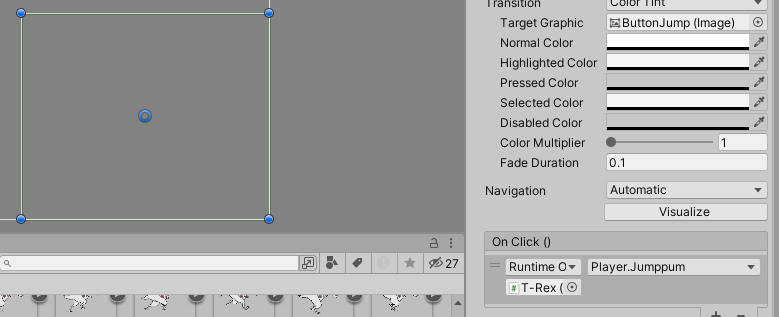


*Hình 4.10: Các Game Object con của ButtonANDPause.*



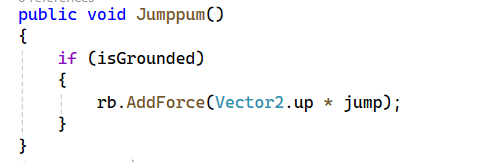
*Hình 5.1: Script Point*

* **ButtonJump:** Là một button được xủ lý màu trong suốt và bố trí bên phải của Canvas ButtonANDPause,



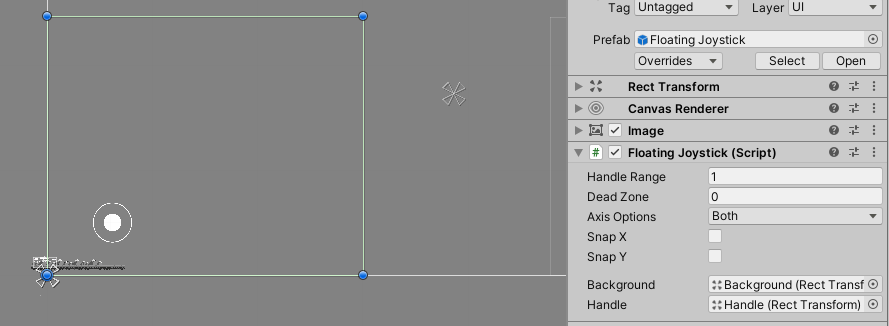
*Hình 5.2: Cấu hình và sự kiện OnClick() của ButtonJump.*

* Khi người chơi chạm vào button thì gọi hàm nhảy (Jumppum) trong Script của nhân vật mà người chơi điều khiển ( T-Rex).



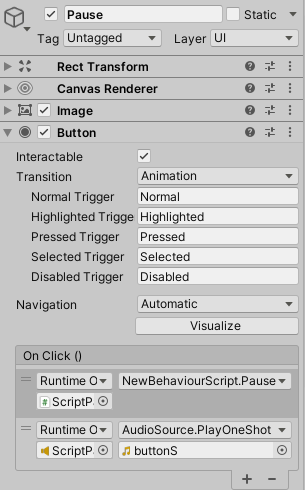
*Hình 5.3: Hàm được thực thi khi người chơi chạm vào ButtonJump.*

* **Floating Joystick:** Nút để điều khiển vũ khí (Gun) của nhân vật để bắn các chướng ngại vật.Chứ bên trong là các hình ảnh hiển thị Background và Handle. Đây là một Package miễn phí và phổ biến cho game trên thiết bị di động trên Asset Store của Unity.

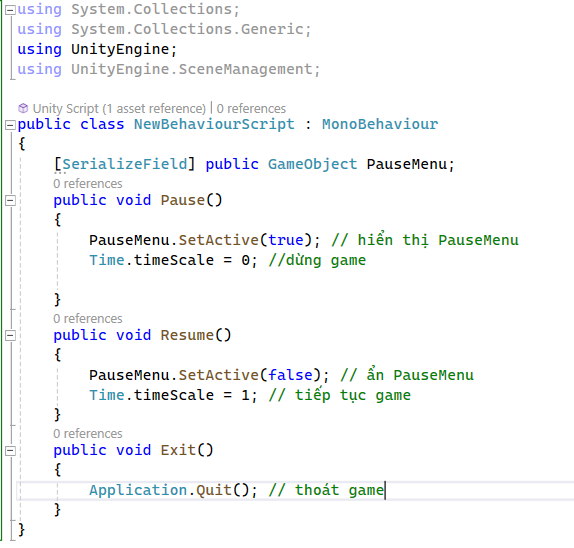


*Hình 5.4: Cấu hình Floating Joystick.*

* **Pause:** Là một button với chức năng dừng game: Trạng thái bình thường, trạng thái khi được nhấn vào. Khi nhấn vào sẽ gọi hàm Pause() được gán từ game object ScrpitPause trong On Click()



*Hình 5.5: Component Button của Pause.*

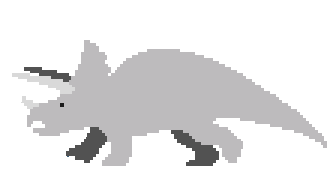


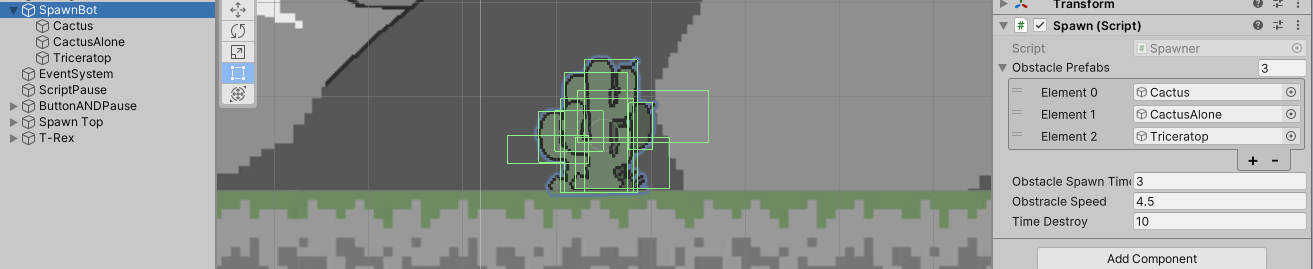
*Hình 5.6:Mã của Script PauseMenu .*

* **PauseMenu:** Chứa bên trong là 2 nút Continue và

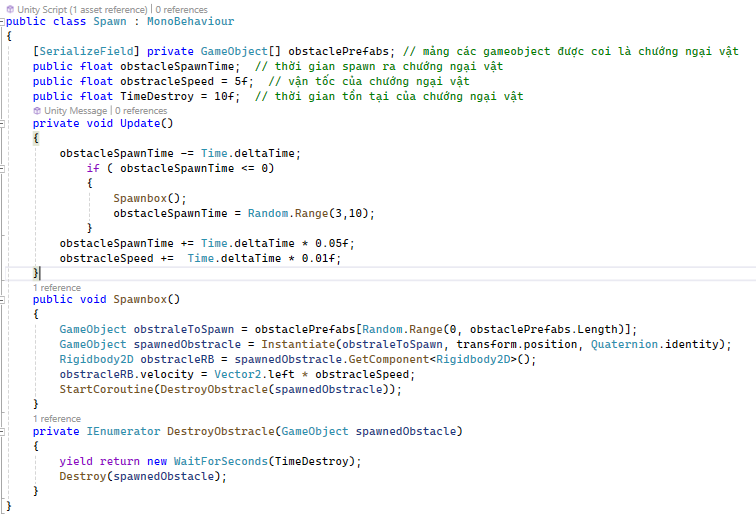
Exit . Với button Continue sẽ gọi hàm Resume() ,button Exit sẽ gọi hàm Exit() trong Script PauseMenu tương tự như button Pause.

### 3.2.5. SpawnBot

Game Object đảm nhiệm việc sinh sản ra 3 chướng ngại vật có tên lần lượt là: Cactus , CactusAlone , Triceratop  . Ba chướng ngại vật này được sinh sản ngẫu nhiên 1 trong 3 và thời gian ngẫu nhiên từ 3 đến 10 giây thông qua Script Spawn .

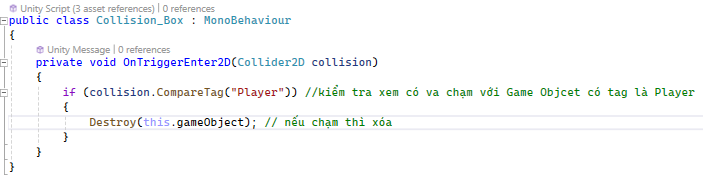


*Hình 5.7:SpawnBot và Component Spawn.*



*Hình 5.8:Mã của Script Spawn.*

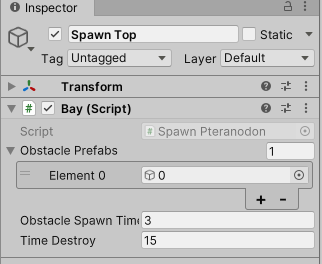
* Các chướng ngại vật sẽ được thêm các Component Boxcolider được bật Trigger và có Tag là “Box” để xử lý các va chạm với nhân vật. Thêm thành phần Rigchibody 2D để có thể duy chuyển sang trái khi được sinh sản. Cuối cùng là 1 Script để xử lý va chạm tên là “Collision\_Box”.



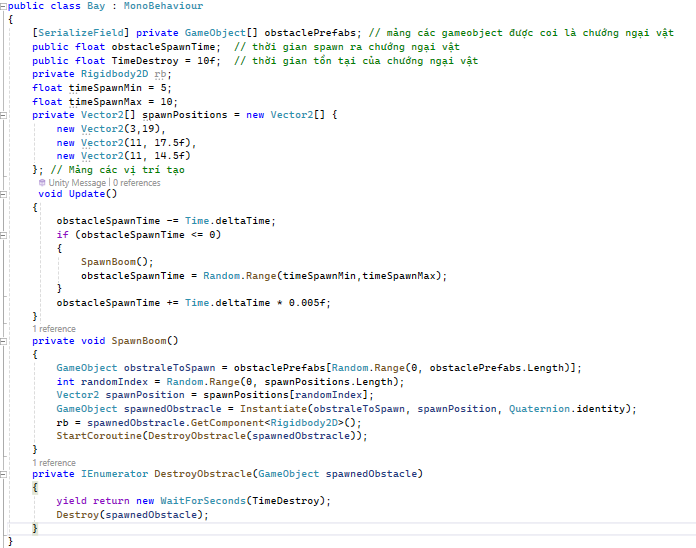
*Hình 5.9:Mã của Script Collision\_Box.*

### 3.2.6. SpawnTop

Thực hiện sinh sản chướng ngại vật tên là “0” ngẫu nhiên ở 3 vị trí ngoài Camera bằng Script Bay script này hoạt động cũng gần tương tự như script Spawn bên trên.

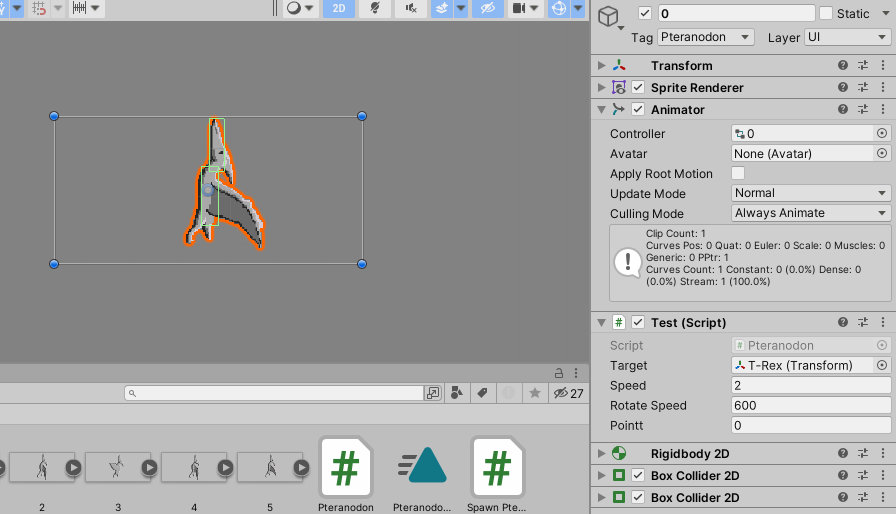


*Hình 5.10:Script Bay trong Inspector của SpawnTop.*

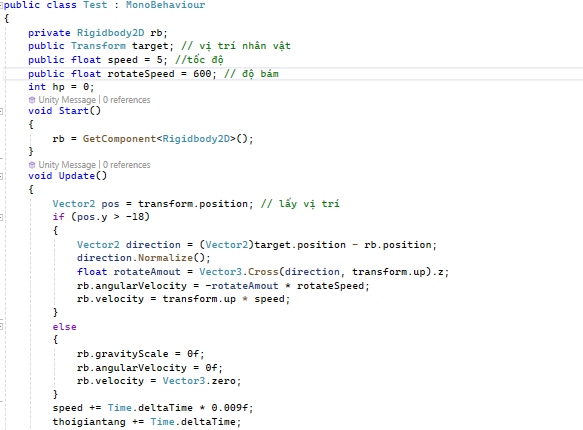


*Hình 6.1:Mã của Script Bay .*

* Chướng ngại vật “0”: với cơ chế là khi được sinh sản ra thì sẽ bay đến vị trí của nhân vật. Nếu chạm đạn từ nhân vật bắn ra với số lần quy định thì sẽ bị xóa, Script tên “Test” sẽ xử lý chức năng này. Tag của game object này là Pteranodon.



*Hình 6.2:GameObject 0 và các Component.*



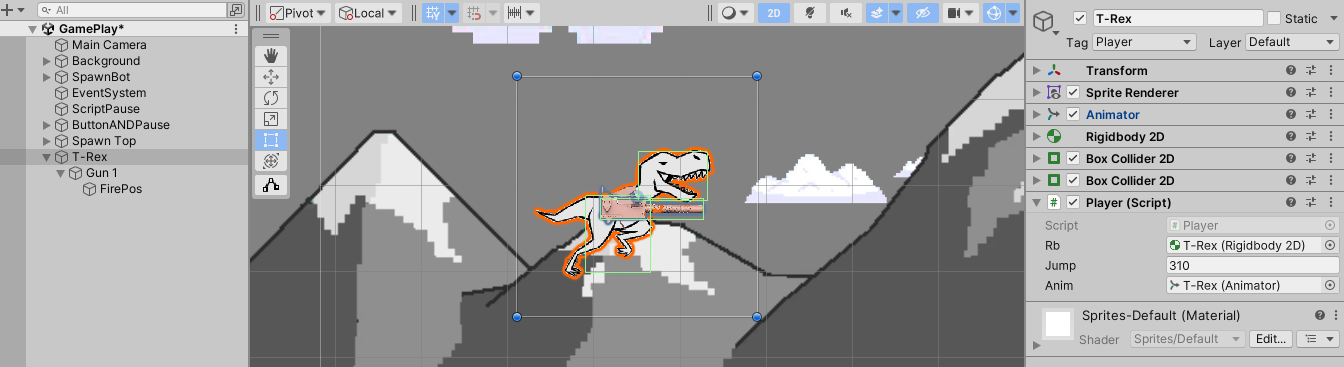
*Hình 6.3:Các hàm xử lý duy chuyển cho game object.*



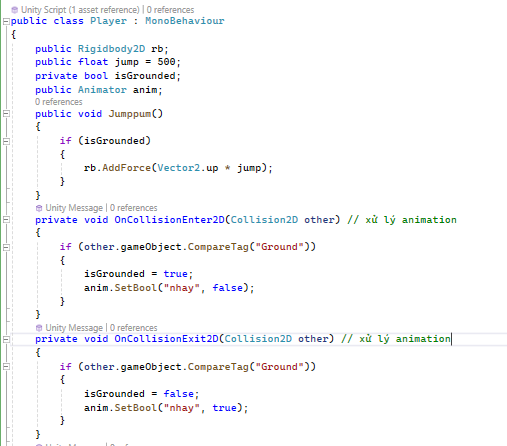
*Hình 6.4:Hàm xử lý tăng máu cho game object.*

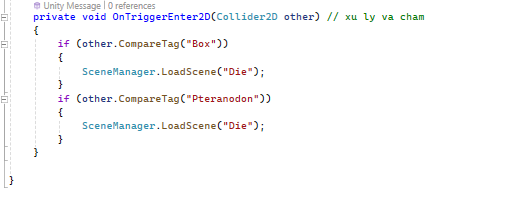
### 3.2.7. T-Rex (Nhân vật).

* Game Object có Script Player đảm nhiệm việc xử lý các va chạm và hàm nhảy cho nhân vật (hàm nhảy được thực thi bằng ButtonJump) để né các chướng ngại vật bên dưới . Chọn tag là Player để xử lý va chạm.



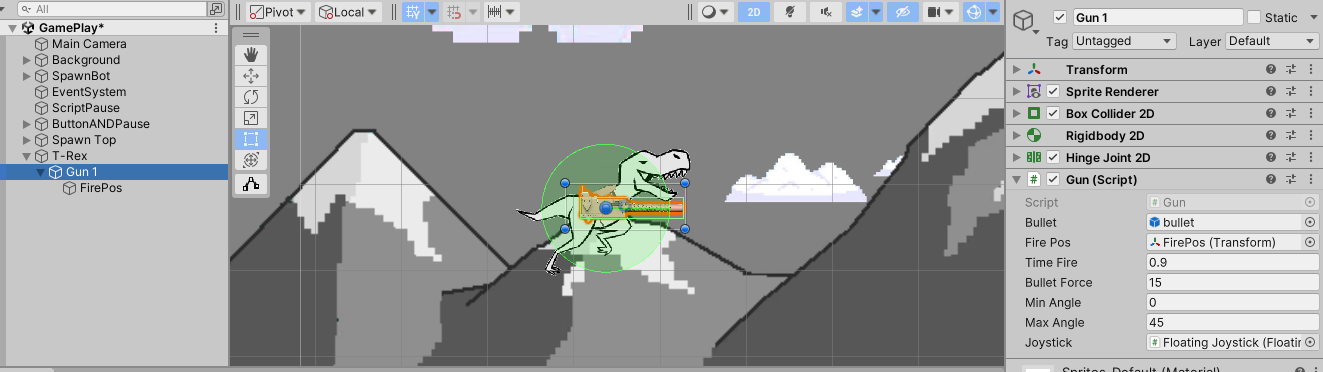
*Hình 6.5:Game Objcet T-Rex và các Component.*





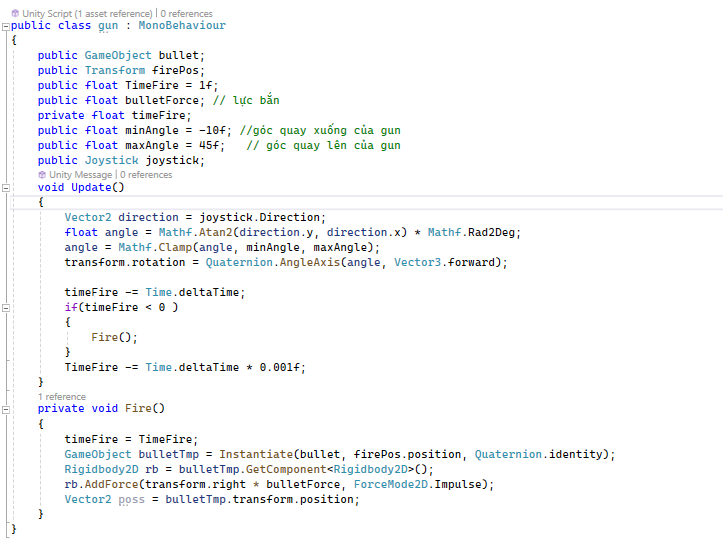
*Hình 6.6:Mã của Script Player.*

* **Gun 1 và FirePos:** nằm bên trong game object Player với cơ chế dùng để cho người chơi điều khiển Gun1 (súng)bằng Floating Joytick để bắn các chướng ngại vật “0” bay đến. Gun 1 được gắn với Player bằng Component Hinge Joint 2D. FirePos là một game object rỗng được gắn vào đầu của Gun 1 làm nơi sinh sản ra đạn (Bullet).



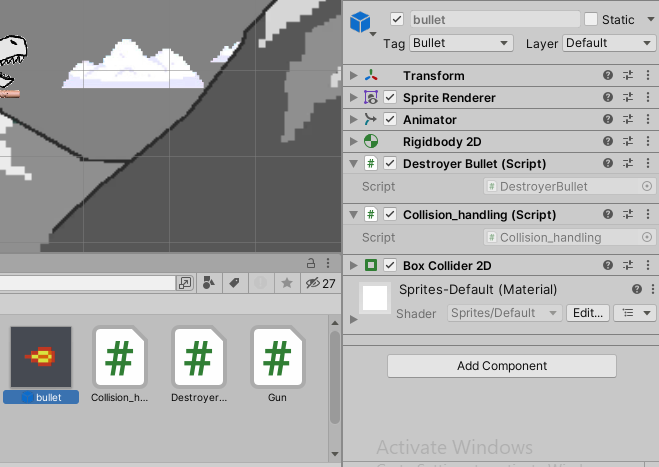
*Hình 6.7:Các component của Gun 1.*

* Script Gun chịu trách nhiệm cho việc sinh sản ra các component Bullet và cơ chế điều khiển.

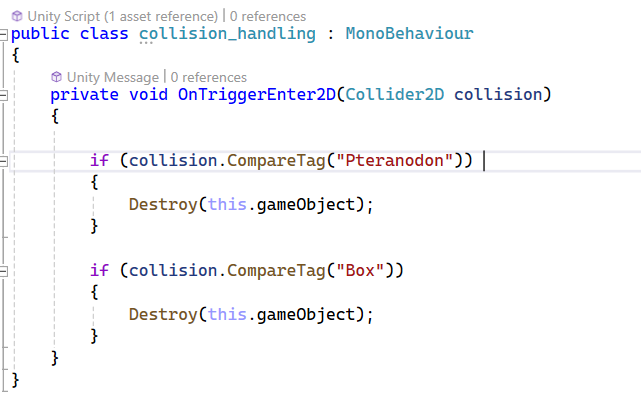


*Hình 6.8: Mã của Script Gun.*

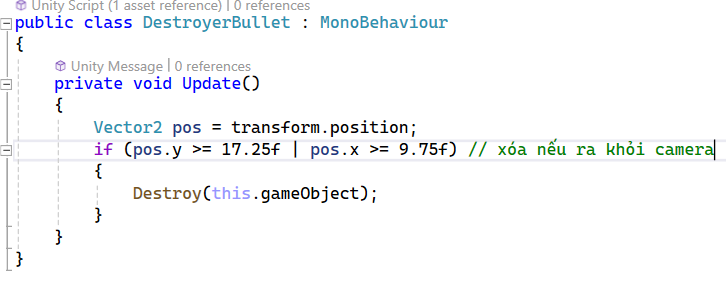
* **Bullet** : sau khi đưa lên khung cửa sổ Hierarchy để tùy chính cho Game Object sau đó kéo nó xuống lại khung Assets trở thành một Package để không hiển thị trong Camera . Bullet được kéo thả vào Script Gun để sinh sản.



*Hình 6.9: Các component của bullet .*



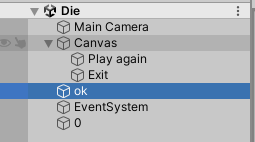
*Hình 6.10: Mã của Script collision\_handling .*



*Hình 7.1: Mã của Script DestroyerBullet.*

## 3.3. Cảnh thua (Die Scenes)

Khi người chơi thất bại thì sẽ được chuyển đến cảnh này. Ở cảnh này có 2 nút với 2 lựa chọn chơi lại hoặc thoát trò chơi .

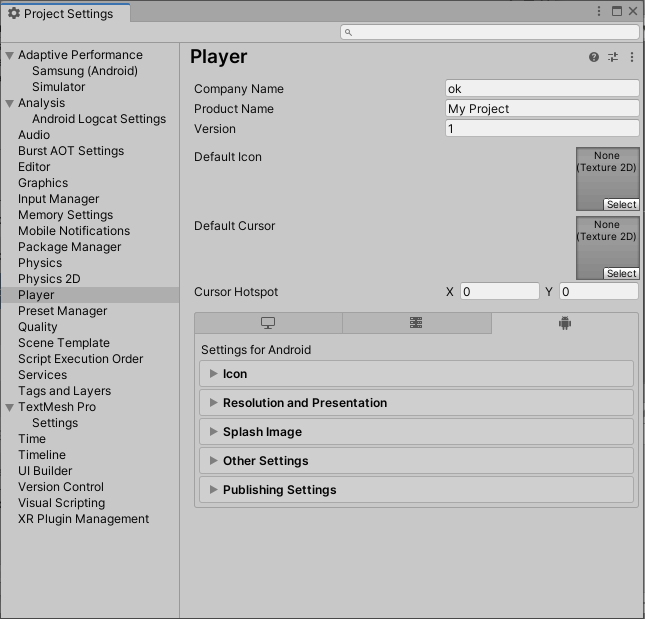


*Hình 7.2: Các Game Object của cảnh Die.*

Với button Play again thì sẽ load lại cảnh Game Play ( giống như button Play ở cảnh Start). Button Exit gọi hàm Exit trong Script PauseMenu.

## 3.4. Cấu hình khi Build sang Apk

Vào File > Build Settings : Ở đây chọn Android và tiếp tục vào Player Settings bên dưới góc trái ở đây điền các thông tin và thêm các Icon cho game nếu muốn.



*Hình 7.3: Cửa sổ Project Settings.*

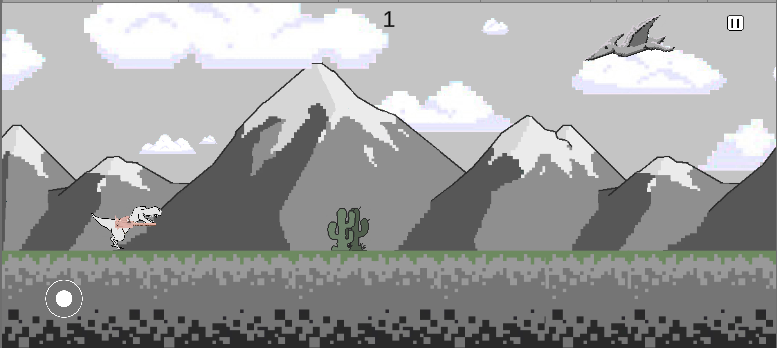
* Chọn **Landscape Left** trong mục **Resolutin and Presentation** để khi vào game trên điện thoại sẽ chuyển sang màng hình ngang

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

* Tạo thành công game thể loại Endless Runner và build thành công sang điện thoại cơ bản với 3 cảnh hoàn chỉnh như sau:



*Hình 7.4: Cảnh bắt đầu.*



*Hình 7.8: Cảnh vào chơi .*



*Hình 7.6: Cảnh thất bại .*

*Link Git bài nộp:* [*https://github.com/TTruonghuy/csn-da21ttb-tranvantruonghuy-game2dunity-csharp.git*](https://github.com/TTruonghuy/csn-da21ttb-tranvantruonghuy-game2dunity-csharp.git)

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Kết luận**

* Thiết kế hoàn chỉnh một game chạy trên nền tảng apk.
* Tiếp thu được nhiều kinh nghiệm từ quá trình thiết kế.
* Nâng cao các kỹ năng lập trình game, thiết kế đồ họa, âm thanh.

**Hướng phát triển**

* Nâng cao chất lượng về mặt hình ảnh
* Tích hợp thêm các chức năng như: Tùy chỉnh cài đặt, nhiệm vụ, đổi điểm,..
* Thêm nhiều skin và tính năng lựa chọn skin nhân vật

.

# DANH MỤC TÀI LIỆU THAM KHẢO

<https://viblo.asia/>

<https://giniwebseo.vn/>

[Wikipedia](https://www.wikipedia.org/)

<https://bard.google.com/>

<https://youtu.be/z1yRrRQo4aE?si=qYbxxWWAB5jWKFqN>

<https://youtu.be/NqrJHj9xlqY?si=_gP4So_wuqRQcvr7>

<https://youtu.be/uMAWn3GXU9g?si=YW1P42vSWtGmDE2c>

<https://youtu.be/TD5Rp__T668?si=YGXLDJwFCdRtJdzY>

<https://youtu.be/-bkmPm_Besk?si=FLbb3jq5BE4iOEFP>

PHỤ LỤC

[CHƯƠNG 1: TỔNG QUAN 10](#_Toc10888)

[1.1. Tổng quan về Unity 10](#_Toc2795)

[1.1.1. Khái niệm 10](#_Toc29593)

[1.1.2. Lịch sử hình thành và phát triển 10](#_Toc12990)

[1.1.3. Các tính năng cơ bản. 11](#_Toc1459)

[1.2. Tổng quan về game thể loại Endless runner. 11](#_Toc10196)

[1.2.1. Lối chơi. 11](#_Toc19426)

[1.2.2. Đồ họa. 11](#_Toc28185)

[1.2.3. Âm thanh. 11](#_Toc4928)

[1.2.4. Tổng quan về game được thiết kế trong đồ án. 11](#_Toc7062)

[1.3. Tổng quan về Visual Studio và C# 12](#_Toc2473)

[1.3.1. Visual Studio. 12](#_Toc7903)

[1.3.2. C#. 12](#_Toc8926)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 13](#_Toc26159)

[2.1. Các khái niệm trong Unity 13](#_Toc8131)

[2.1.1. Project 13](#_Toc21537)

[2.1.2. Assets (tài nguyên) 14](#_Toc4264)

[2.1.3. Scenes (cảnh) 15](#_Toc5986)

[2.1.4. Game Object (Đối tượng game) 15](#_Toc12984)

[2.1.5. Packages( Gói ) 15](#_Toc25816)

[2.1.6. Component ( Thành phần ) 16](#_Toc18948)

[2.1.7. Script ( Mã nguồn ) 16](#_Toc20490)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 21](#_Toc25798)

[3.1. Tạo và cấu hình project và cảnh Bắt đầu (Start). 21](#_Toc7126)

[3.1.1. Tạo project. 21](#_Toc6603)

[3.1.2. Tạo Game Object Canvas để chứa nút bắt đầu. 23](#_Toc28079)

[3.1.3. Tạo Game Object nút bắt đầu ( Start\_button) 27](#_Toc9001)

[3.1.4. Tạo Game Object EventSystem và BGroound 30](#_Toc24572)

[3.2. Cảnh Game Play (GamePlay Scenes) 30](#_Toc32683)

[3.2.1. Main Camera và EventSystem 31](#_Toc5207)

[3.2.2. ScriptPause 31](#_Toc22682)

[3.2.3. Background 31](#_Toc11438)

[3.2.4. ButtonANDPause 33](#_Toc5834)

[3.2.5. SpawnBot 36](#_Toc2718)

[3.2.6. SpawnTop 37](#_Toc10518)

[3.2.7. T-Rex (Nhân vật). 40](#_Toc28292)

[3.3. Cảnh thua (Die Scenes) 43](#_Toc20396)

[3.4. Cấu hình khi Build sang Apk 43](#_Toc7177)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 44](#_Toc6880)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 45](#_Toc6210)

[DANH MỤC TÀI LIỆU THAM KHẢO 46](#_Toc343)