

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO MÔN HỌC
CÔNG NGHỆ .NET**

Đề tài: TÌM HIỂU UNITY

Lớp: SE310.K21.PMCL

Giảng viên hướng dẫn:

GV. HUỲNH HỒ THỊ MỘNG TRINH

Sinh viên thực hiện:

NGUYỄN THÀNH LUÂN - 16520703

Tp. Hồ Chí Minh, 03/2020

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày tháng năm 2020

Người nhận xét
(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong thời gian thực hiện báo cáo môn học, em đã nhận được sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của các Thầy Cô, các anh chị khóa trên, gia đình và bạn bè.

Em xin gửi lời cảm ơn chân thành đến Cô Huỳnh Hồ Thị Mộng Trinh, giảng viên môn học Công nghệ .NET – Trường ĐH Công nghệ thông tin – ĐH Quốc gia Tp.Hồ Chí Minh đã giúp đỡ, giải đáp các thắc mắc của em và các bạn trong quá trình thực hiện báo cáo.

Em cũng xin chân thành cảm ơn các Thầy Cô giáo, các tác giả có các bài viết, bài giảng, trích dẫn,... được sử dụng trong bài báo cáo này.

Cuối cùng, em xin chân thành cảm ơn các anh chị khóa trên, gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ trong suốt quá trình thực hiện báo cáo đồ án môn học Công nghệ .NET này.

Bài báo cáo chắc chắn không thể tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của Cô và các bạn để bài báo cáo đồ án môn học này của em được hoàn thiện hơn.

Một lần nữa, em xin chân thành cảm ơn!

Tp.Hồ Chí Minh, ngày 30 tháng 03 năm 2020

Sinh viên thực hiện

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN.....	3
LỜI CẢM ƠN	4
Phần I – TỔNG QUAN VỀ UNITY	7
1. Unity là gì?	7
2. Quá trình phát triển của Unity.....	7
3. Một số thống kê về Unity.....	7
4. Ưu điểm của Unity	9
Phần II – CÀI ĐẶT VÀ KÍCH HOẠT.....	11
Phần III – LÀM QUEN VỚI UNITY ENGINE	23
1. Các thành phần trong Unity Editor.....	23
a) Cửa sổ Scene	23
b) Cửa sổ Hierarchy.....	24
c) Cửa sổ Game	24
d) Cửa sổ Project	24
e) Cửa sổ Inspector	25
f) Tool bar.....	25
g) Window Controls	26
2. Các khái niệm cơ bản trong Unity.....	28
a) GameObject	28
b) Component.....	28
c) Sprite.....	28
d) Animation	29
e) Key Frame.....	30
f) Prefabs	30
g) Sounds.....	30
h) Script.....	30
i) Scenes	30
j) Assets	30

k) Camera	31
l) Transform.....	32
Phần IV – SỬ DỤNG UNITY	33
1. Unity 3D – Tạo một đối tượng 3D có thể di chuyển được	33
2. Unity 2D.....	40
Phần V – UNITY VÀ MỘT SỐ GAME ENGINE TIÊU BIỂU KHÁC	42
1. Unity 3D.....	43
2. Unreal Engine	44
3. CryEngine	45
4. Kết luận	46
Phần VI – TỔNG KẾT	48
TÀI LIỆU THAM KHẢO VÀ TỔNG HỢP	49
DANH MỤC HÌNH ẢNH ĐƯỢC SỬ DỤNG	50

Phần I – TỔNG QUAN VỀ UNITY



Hình 1.1

1. Unity là gì?

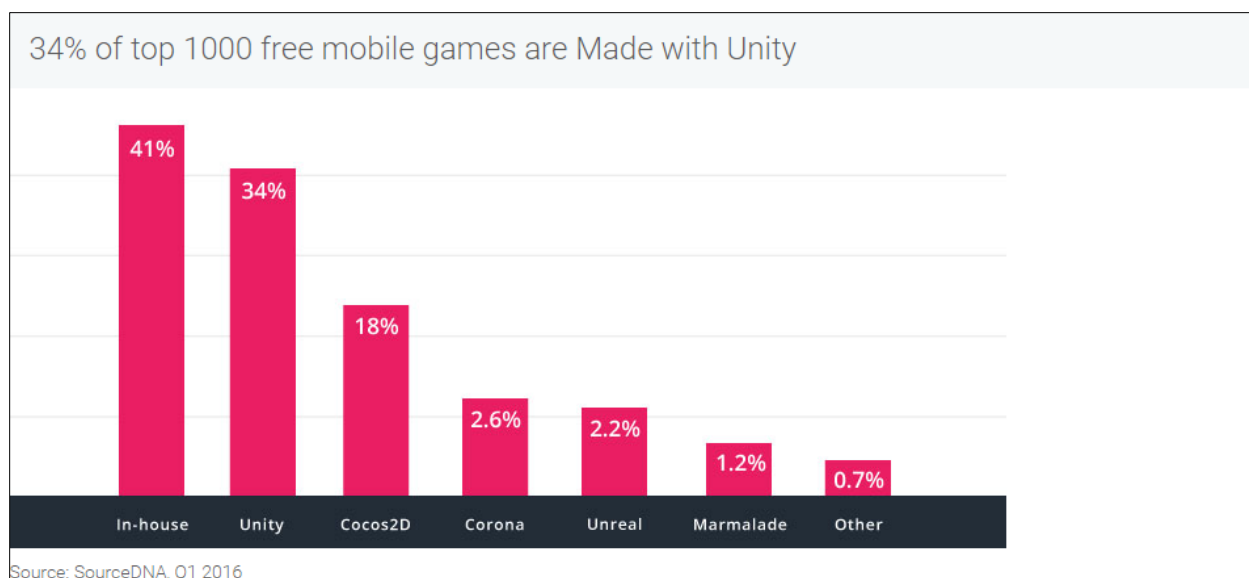
Unity là một “cross-platform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies, sử dụng .NET và ngôn ngữ lập trình C# (có thể sử dụng JavaScript hay Boo cho Unity 3D). Game engine này được sử dụng để phát triển game 2D/3D trên PC, consoles, thiết bị di động và trên websites.

2. 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 hơn 12 năm phát triển, nay 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 Excellence.

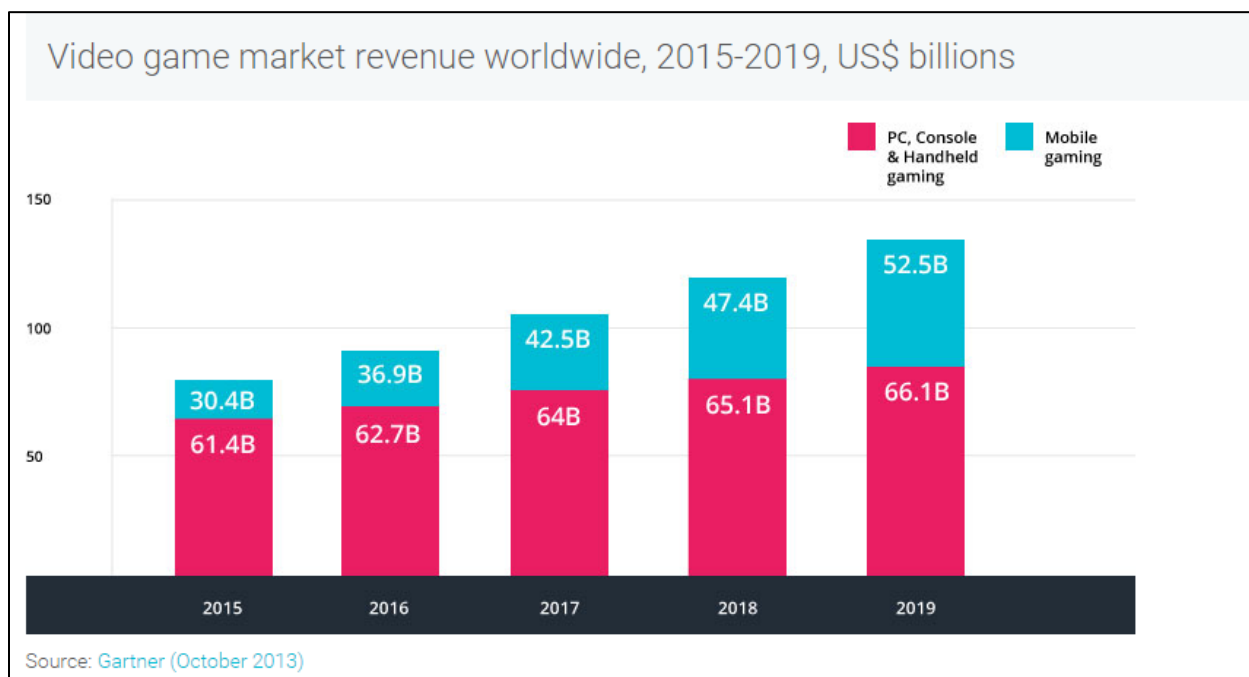
3. Một số thống kê về Unity

- Tính đến quý 3 năm 2016 đã có 5 tỉ lượt download game và ứng dụng được phát triển bởi Unity
- 2,4 tỉ thiết bị di động đã từng tải ít nhất 1 ứng dụng bởi Unity.
- Trong top 1000 game Mobiles miễn phí thì số lượng game tạo ra bởi Unity chiếm tới 34%



Hình 1.2

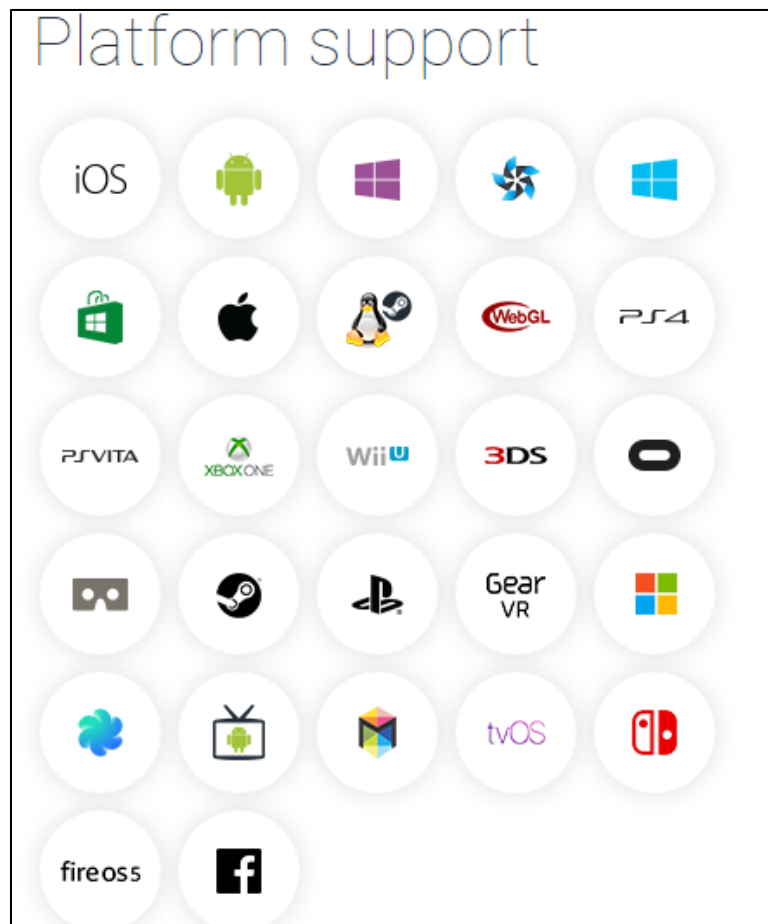
- Số lượng người dùng (gamer) của Unity đạt tới con số 770 triệu, trong khi đó số người thường xuyên sử dụng Twitter là 310 triệu người.
- Sự thay đổi trong cách thức chơi game của người chơi hay nói cách khác là xu hướng mọi người tập trung vào game trên di động nhiều hơn.



Hình 1.3

4. Ưu điểm của Unity

- 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 dòng 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....



Hình 1.4

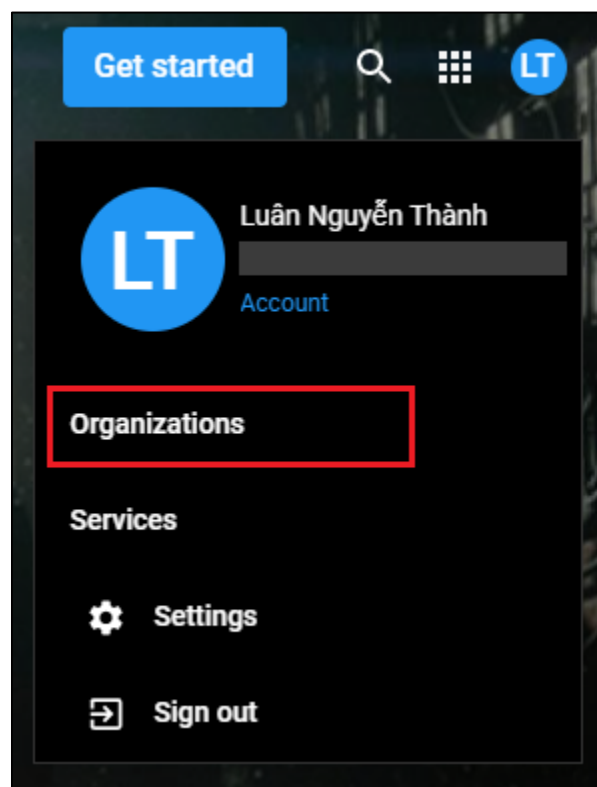
- Dễ sử dụng: Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá 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.

Phần II – CÀI ĐẶT VÀ KÍCH HOẠT

Phần này sẽ hướng dẫn cách cài đặt và kích hoạt bản quyền *Unity Pro* thông qua [Github Student Developer Pack](#) do Microsoft tài trợ.

Bước 1:

- Truy cập vào [trang chủ Unity](#).
- Sign up và sử dụng tài khoản email bất kì để Login tài khoản.
- Chọn *Account* -> *Organizations*.



Hình 2.1

Bước 2:

“Unity Seat” là một khái niệm cho phép người dùng sử dụng các dịch vụ của Unity, mỗi một người sử dụng tương ứng với 1 seat.

- Trong mục *Add Seats*, chọn *New*.
- Nhập vào địa chỉ email đã dùng để login vào Unity.
- Nhấn vào [here](#).

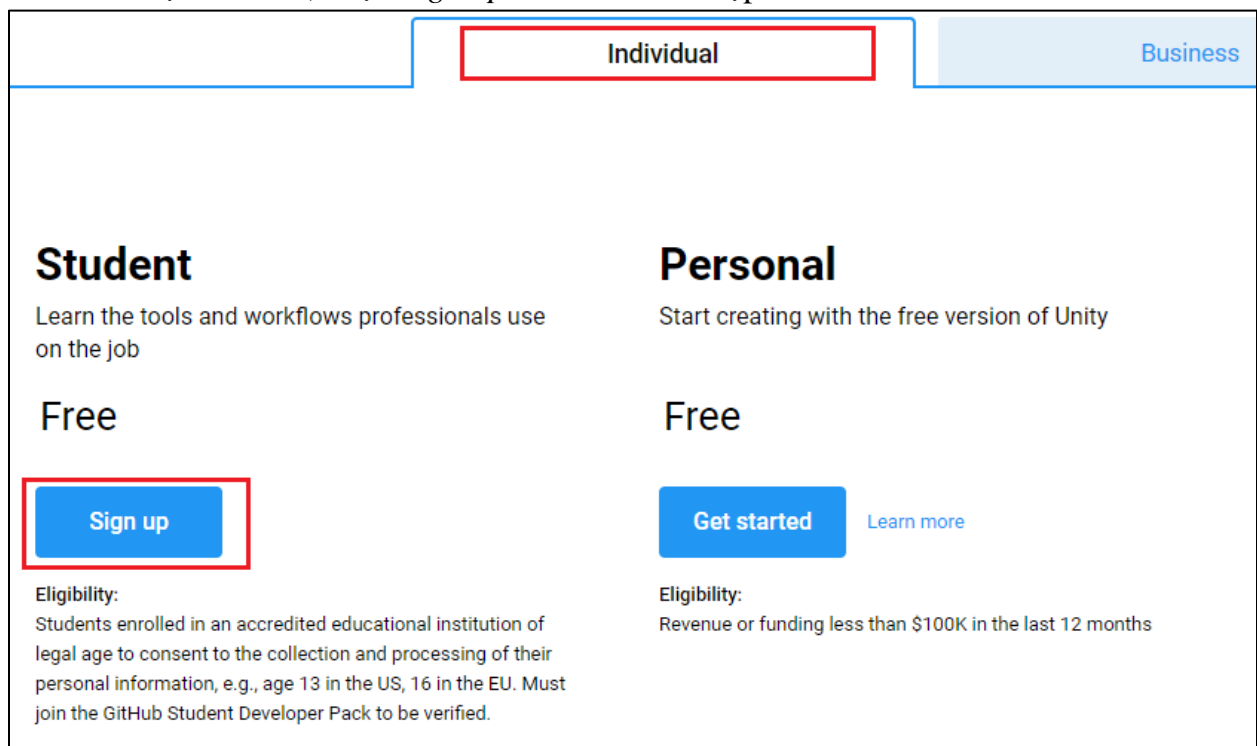


Hình 2.2

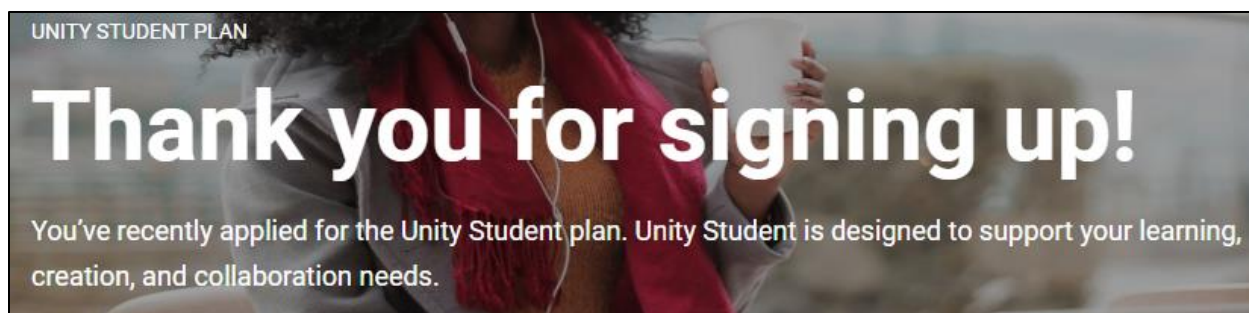
Bước 3:

Liên kết Unity với Github.

- Chọn tab *Individual*
- Ở mục *Student*, chọn *Sign up* và tiến hành nhập vào tài khoản Github để liên kết.



Hình 2.3



Hình 2.4

Bước 4:

Liên kết với GitHub Student Developer Pack để có toàn quyền truy cập vào các ích lợi của Student Plan.

- Sau khi đã Sign up thành công qua tài khoản Github ở bước 3, ở mục *What's next?* nhấn vào [Github Student Developer Pack](#)

What's next?

1. Download the Personal Edition of Unity and [start creating!](#) Saved projects in the Personal Edition will still be accessible after you update to the Unity Student plan.
2. Apply for the [GitHub Student Developer Pack](#) to access your Unity Student plan benefit.
3. Submit proof of enrollment in an accredited educational institution (GitHub will provide a timeframe for confirmation of your student status).
4. When you receive the GitHub email confirming you've joined the Student Developer Pack, return to the [Unity Student plan](#) page to activate your plan.

If you've already taken these steps and your student status has still not been verified, please troubleshoot any GitHub verification-related [here](#), or contact [GitHub](#) directly.

Hình 2.5

- Nhấn vào *Get Student Benefits*

Individuals

Students

Learn using real-world development tools

- ✓ **FREE** [GitHub Pro](#) while you are a student
- ✓ Valuable [GitHub Student Developer Pack](#) partner offers
- ✓ [GitHub Campus Expert training](#) for qualified applicants
- ✓ \$1000 first-time hackathon grant for [Major League Hacking](#) members

[Get student benefits](#)

Teachers

Teach your students with the industry-standard tools

- ✓ **FREE** [GitHub Teacher Toolbox](#) The best developer tools for teaching, free for academic use
- ✓ **FREE** [GitHub Team](#) for courses, coding clubs, and nonprofit research
- ✓ [GitHub Classroom](#) for managing assignments
- ✓ [GitHub Campus Advisor](#) training to master Git and GitHub

[Get teacher benefits](#)

Hình 2.6

- Điền thông tin xác thực tài khoản liên kết.
- Nếu như trước đó, tài khoản Github đã liên kết thành công với Student Packs thì sẽ không cần phải thực hiện lại bước này nữa.

Which best describes your academic status? [?](#)

☒ Student ☐ Faculty

The GitHub Student Developer Pack is only available to students aged 13 or older.

What e-mail address do you use for school?


⚡ Pro-tip: Selecting a school-issued email address gives you the best chance of a speedy review.

☒ @gm.uit.edu.vn **Ho Chi Minh City University of Information Technology**

+ Add an email address

Hmmm, we'll need a little more information.
Please upload proof of your academic status.

Use your camera to take a picture.



Please provide official, **dated** proof of academic affiliation like your:

- School ID card
- Academic transcript
- Or another affiliation/enrollment verification document

[Learn more about other types of proof](#)

Upload image

How do you plan to use GitHub?

Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.

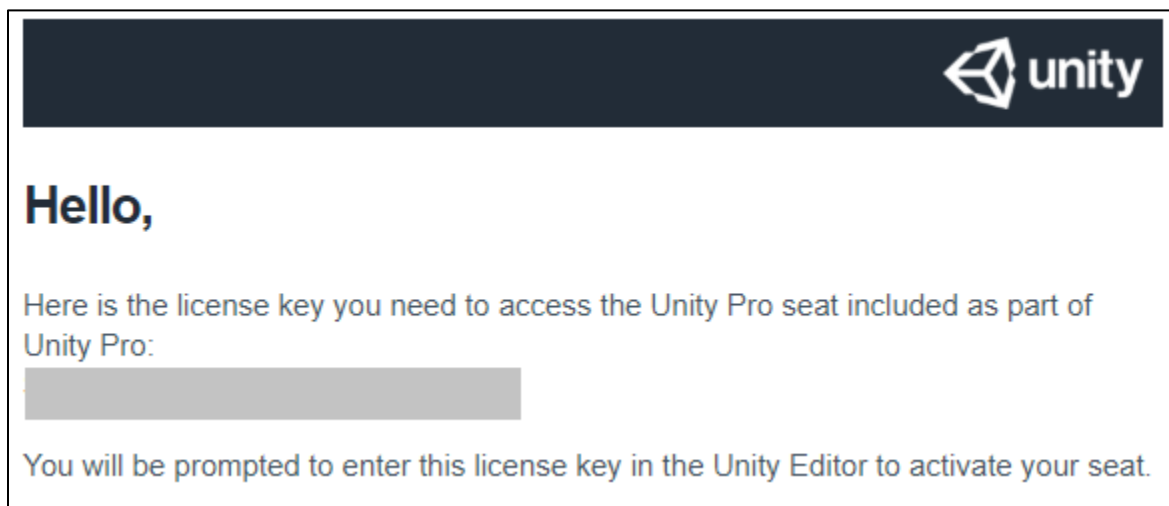
Submit your information

Hình 2.7

Bước 5:

Nhận key kích hoạt bản quyền Unity Pro.

Sau khi đợi xác thực tài khoản sinh viên xong (của mình là 2 ngày sau), một email kích hoạt bản quyền Unity Pro có dạng như sau sẽ được gửi về địa chỉ email đã dùng để đăng nhập lúc đầu.



Hình 2.8

Bước 6:

Unity Hub là phần mềm độc lập giúp quản lý danh sách các projects của Unity trên thiết bị, quản lý và cài đặt version cũng như các module của Unity Editor.

Download và cài đặt Unity Hub tại: <https://unity3d.com/get-unity/download>

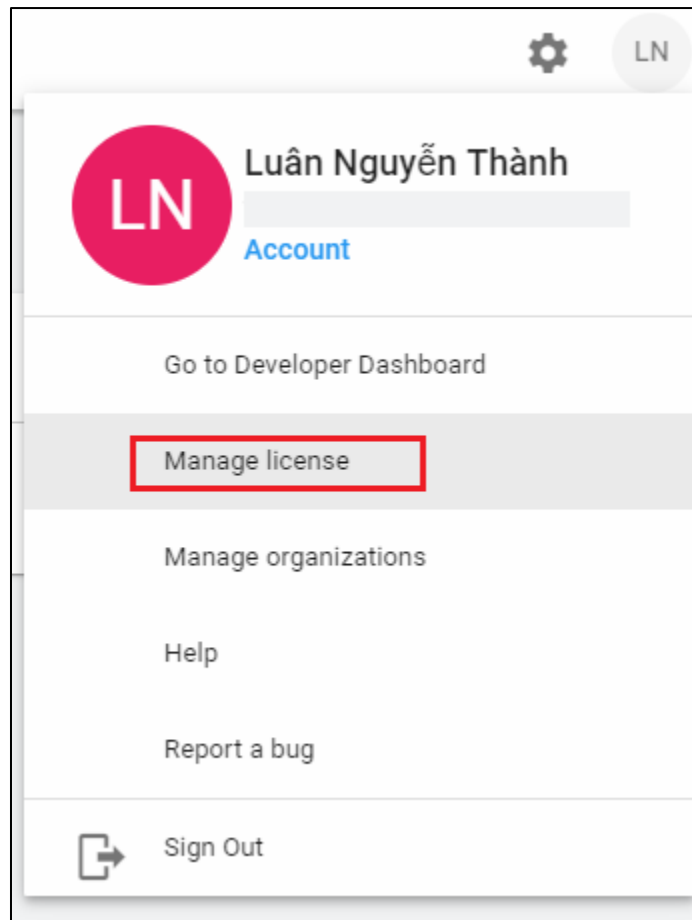


Hình 2.9

Bước 7:

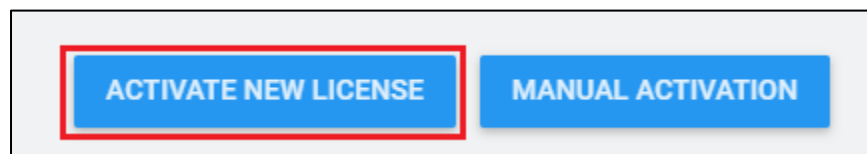
Đăng nhập và sử dụng key để kích hoạt Unity Pro.

- Đăng nhập vào Unity Hub (nên sử dụng cùng một địa chỉ email khi đăng nhập trên trang chủ Unity)
- Chọn *Account* -> *Manage license*



Hình 2.10

- Chọn *ACTIVATE NEW LICENSE*.



Hình 2.11

- Ở cửa sổ hiện ra, chọn *Unity Plus or Pro*, điền key kích hoạt đã nhận được ở bước 5 và ấn *DONE*.

New License Activation [X]

License Agreement

Please select one of the options below:

☐ Unity Personal

☒ **Unity Plus or Pro**
Serial Number *

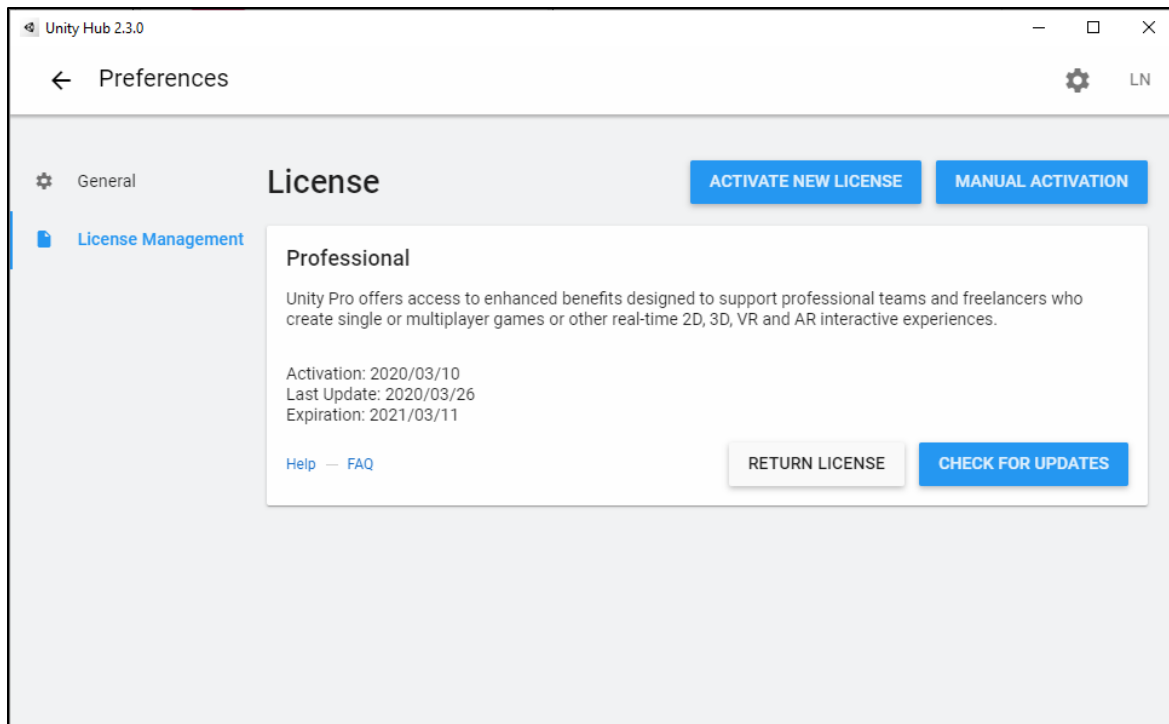
(Điền key kích hoạt đã được gửi về mail ở đây)

[Buy Professional Edition - Help - FAQ](#)

DONE

Hình 2.12

- Đây là kết quả sau khi kích hoạt.

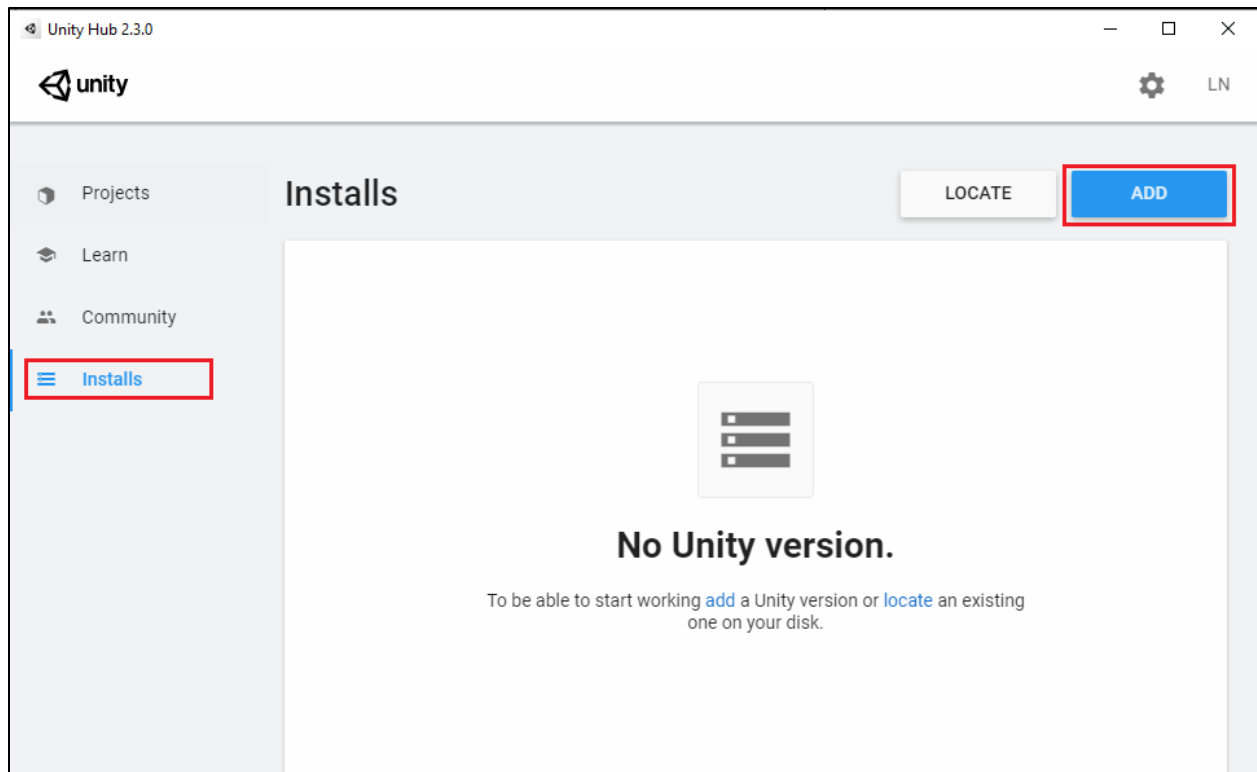


Hình 2.13

Bước 8:

Tải và cài đặt Unity Editor.

- Chọn mục *Installs* -> *ADD*.



Hình 2.14

- Chọn version và các modules cần thiết. Quá trình tải về sẽ tốn khá thời gian, nhưng mà tải xong rồi thì có thể sử dụng và vọc vạch Unity thoải mái, khỏi phải lo lắng gì nữa hết.

Tất nhiên là ngoại trừ cái License rồi, vì nó chỉ kéo dài trong 1 năm thôi =))

Add Unity Version

X

1

Select a version of Unity

2

Add modules to your install

Can't find the version you're looking for? Visit our [download archive](#) for access to [long-term support](#) and [patch releases](#), or join our [Open Beta program](#) releases.

Latest Official Releases

☒ Unity 2019.3.6f1

☐ Unity 2018.4.19f1 (LTS)

☐ Unity 2017.4.38f1 (LTS)

Latest Pre-Releases

☐ Unity 2020.1.0b2 (Beta)

CANCEL

BACK

NEXT

Hình 2.15

Add Unity Version

Add modules to Unity 2019.3.6f1 : total space available 164.3 GB - total space required 10.1 GB

Dev tools	Download Size	Install Size
<input type="checkbox"/> Microsoft Visual Studio Community 2019	1.4 GB	1.3 GB
Platforms		
<div> <div> </div> <div> <input checked="" type="checkbox"/> </div> <div> </div> </div> Android Build Support	238.1 MB	1.1 GB
<input checked="" type="checkbox"/> iOS Build Support	661.6 MB	2.7 GB
<input type="checkbox"/> tvOS Build Support	339.0 MB	1.5 GB
<input type="checkbox"/> Linux Build Support (Mono)	57.0 MB	261.1 MB
<input type="checkbox"/> Mac Build Support (Mono)	87.4 MB	489.7 MB

CANCEL

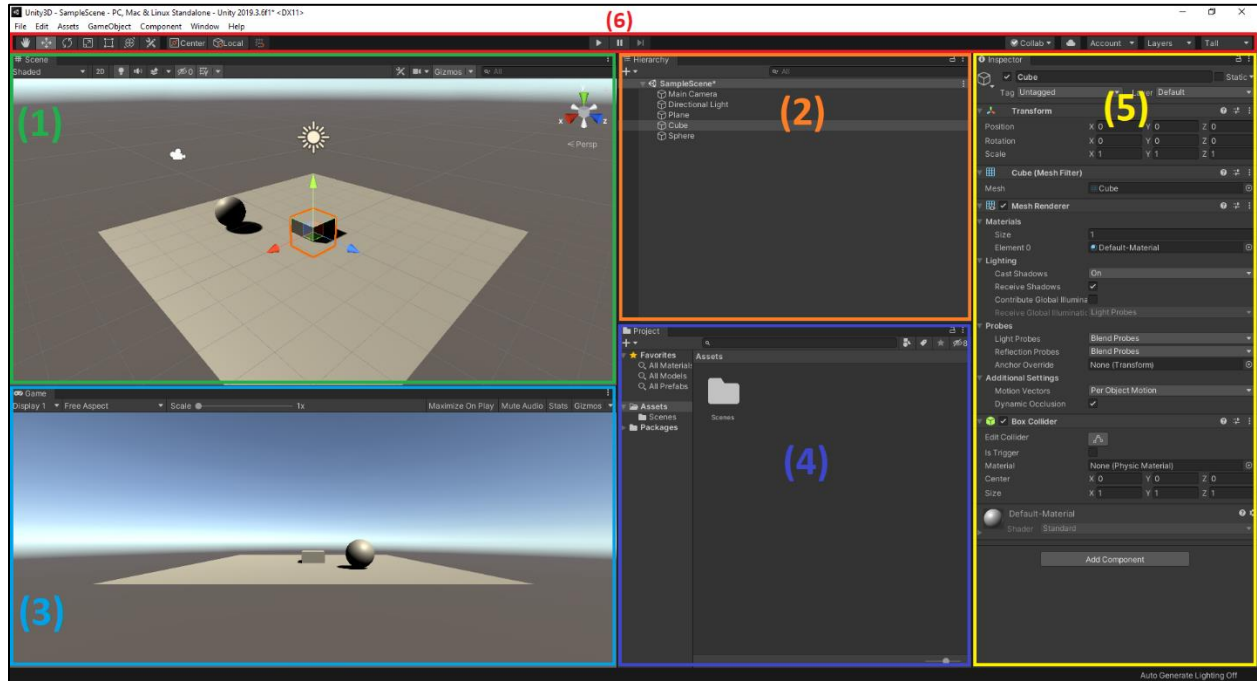
BACK

DONE

Hình 2.16

Phần III – LÀM QUEN VỚI UNITY ENGINE

1. Các thành phần trong Unity Editor



Hình 3.1

a) Cửa sổ Scene

- Phần này phần hiển thị các đối tượng trong scenes một cách trực quan, có thể lựa chọn các đối tượng, kéo thả, phóng to, thu nhỏ, xoay các đối tượng ...
- Phần này có thể thiết lập một số thông số như hiển thị ánh sáng, âm thanh, cách nhìn 2D hay 3D ... Khung nhìn Scene là nơi bố trí các Game Object như cây cối, cảnh quan, enemy, player, camera, ... trong game. Sự bố trí hoạt cảnh là một trong những chức năng quan trọng nhất của Unity.
- Gizmo:



Hình 3.2

Ở vị trí trên cùng bên phải của cửa sổ Scene, Gizmo cho phép bạn nhanh chóng thay đổi hướng của camera bằng việc nhấp chọn vào 1 trong 6 nhánh của nó, hoặc thay đổi giữa **Perspective mode** và **Isometric mode**.

b) Cửa sổ Hierarchy

- Tab hierarchy là nơi hiển thị các Game Object trong Scenes hiện hành. Khi các đối tượng được thêm hoặc xóa trong Scenes, tương ứng với các đối tượng đó trong cửa sổ Hierarchy.
- Tương tự trong tab Project, Hierarchy cũng có một thanh tìm kiếm giúp quản lý và thao tác với các Game Object hiệu quả hơn đặc biệt là với các dự án lớn.
- Bạn có thể tạo mới một đối tượng với nút xổ xuống Create và nhanh chóng focus tới nó bằng cách:
 - Chọn đối tượng trong cửa sổ Hierarchy
 - Di chuyển con trỏ chuột tới màn hình Scene
 - Ấn phím F.

Điều này sẽ định lại vị trí của camera và focus tới đối tượng đã chọn.

c) Cửa sổ Game

- Đây là màn hình demo game, là góc nhìn từ camera trong game. Các bạn có thể chơi thử game đã tạo ở cửa sổ này.
- Thanh công cụ trong cửa sổ game cung cấp các tùy chỉnh về độ phân giải màn hình, thông số (stats), gizmos, tùy chọn bật tắt các component...

d) Cửa sổ Project

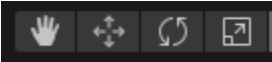


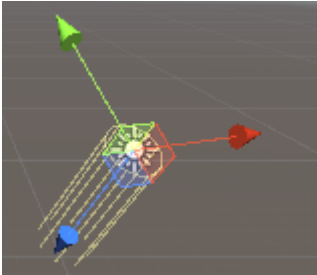

- Đây là cửa sổ explorer của Unity, hiển thị thông tin của tất cả các tài nguyên (Assets) trong game của bạn.
- Cột bên trái hiển thị assets và các mục yêu thích dưới dạng cây thư mục tương tự như Windows Explorer. Khi click vào một nhánh trên cây thư mục thì toàn bộ nội dung của nhánh đó sẽ được hiển thị ở khung bên phải. Ta có thể tạo ra các thư mục mới bằng cách Right click -> Create -> Folder hoặc nhấn vào nút Create ở góc trên bên trái cửa sổ Project và chọn Folder. Các tài nguyên trong game cũng có thể được tạo ra bằng cách này.
- Phía trên cây thư mục là mục Favorites, giúp chúng ta truy cập nhanh vào những tài nguyên thường sử dụng. Chúng ta có thể đưa các tài nguyên vào Favorites bằng thao tác kéo thả.
- Đường dẫn của thư mục tài nguyên hiện tại. Chúng ta có thể dễ dàng tiếp cận các thư mục con hoặc thư mục gốc bằng cách click chuột vào mũi tên hoặc tên thư mục.

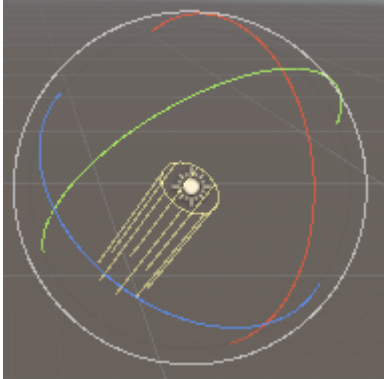

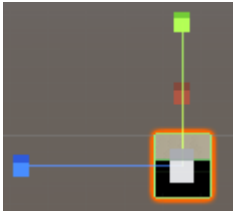
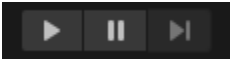



e) Cửa sổ Inspector

- Cửa sổ Inspector hiển thị chi tiết các thông tin về Game Object đang làm việc, kể cả những component được đính kèm và thuộc tính của nó. Bạn có thể điều chỉnh, thiết lập mọi thông số và chức năng của Game Object thông qua cửa sổ Inspector.
- Mọi thuộc tính thể hiện trong Inspector đều có thể dễ dàng tùy chỉnh trực tiếp mà không cần thông qua một kịch bản định trước. Tuy nhiên Scripting API cung cấp một số lượng nhiều và đầy đủ hơn do giao diện Inspector là có giới hạn.
- Các thiết lập của từng component được đặt trong menu. Các bạn có thể click chuột phải, hoặc chọn icon hình bánh răng nhỏ để xuất hiện menu.
- Ngoài ra Inspector cũng thể hiện mọi thông số Import Setting của asset đang làm việc như hiển thị mã nguồn của Script, các thông số animation, ...

f) Tool bar

Mình sẽ nói qua một số công cụ nổi bật và hay dùng nhất nằm trên thanh Tool bar.

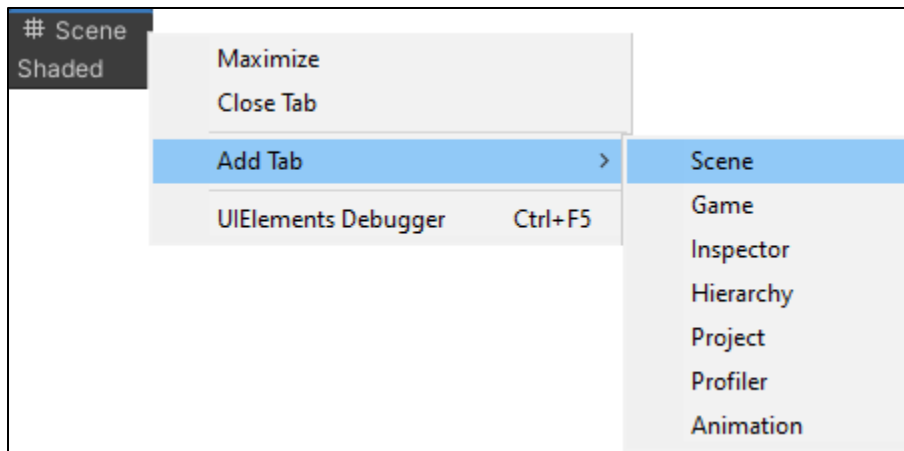
Nhóm công cụ	Công cụ	Chức năng
 <p>Nhóm công cụ di chuyển, thao tác màn hình, đối tượng trong Scenes.</p>	 <p>Tương ứng với phím Q</p>	Giúp dịch chuyển vị trí màn hình Scene
	 <p>Tương ứng với phím W</p>	<p>Dùng để lựa chọn các đối tượng, dịch chuyển theo 3 hướng: x (màu đỏ), y (xanh lam), z (xanh da trời). Muốn đối tượng dịch chuyển theo 3 chiều khác nhau chỉ việc click vào mũi tên và kéo theo chiều đó.</p>  <p>Hình 3.3</p>
		Dùng để xoay các đối tượng theo các hướng khác nhau.

	Tương ứng với phím E	 <p>Hình 3.4</p>
	 <p>Tương ứng với phím R</p>	<p>Dùng để kéo đối tượng to hơn hay nhỏ hơn so với kích thước</p>  <p>Hình 3.5</p>
 <p>Nhóm công cụ Debug trong Games</p>		Cho phép chạy demo game (Play)
		Cho phép dừng game tại một frame nào đó (Pause)
		Cho phép chạy từng frame (Step)

g) Window Controls

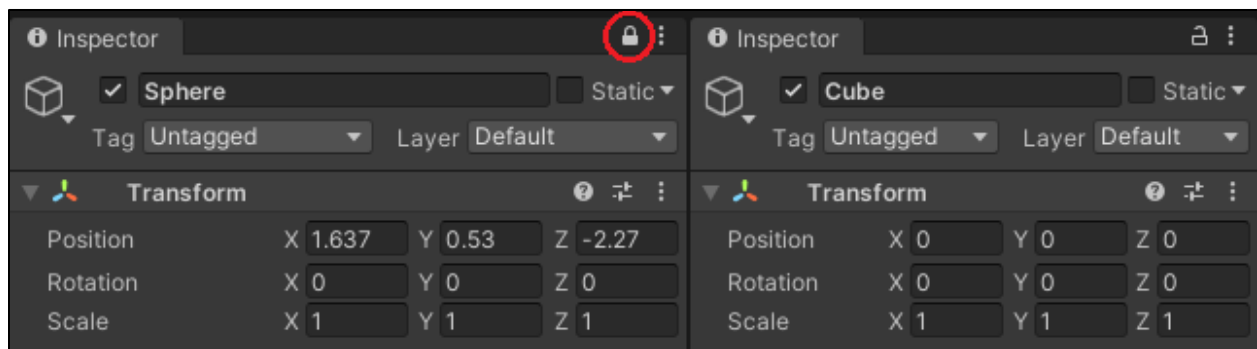
Ngoài ra còn có các cửa sổ khác của Unity như Sprite Editor, Animation, Animator, Console... và các bạn có thể tự tùy biến cho mình một bố cục sao cho thuận tiện với mình nhất.

Danh sách, số lượng cửa sổ được hiển thị không bị giới hạn trong Unity. Bạn có thể thêm mới một cửa sổ bằng cách click chuột phải vào đầu mỗi tab và chọn Add Tab.



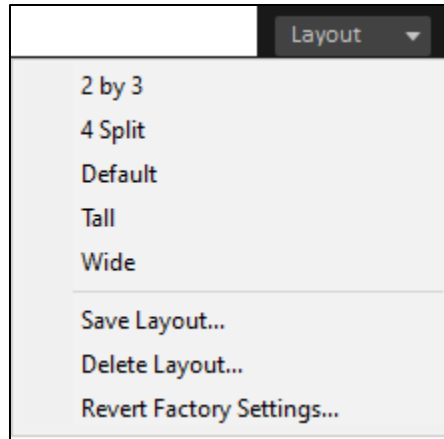
Hình 3.6

Điều này cực kì hữu dụng, bởi vì các cửa sổ Project và Inspector cung cấp một chức năng là Lock (click chuột phải để chọn, tương tự như Add Tab) cho phép cố định đối tượng đang chọn, do đó bạn có thể thao tác chỉnh sửa 2 đối tượng khác nhau cùng một lúc!



Hình 3.7

Nếu như bạn không thích định dạng cửa sổ mặc định của Unity, bạn có thể thay đổi nó bất cứ khi nào chỉ bằng việc kéo thả các cửa sổ. Có một số cách bố trí cửa sổ có thể hữu dụng trong menu drop-down Layout ở tận cùng bên phải của thanh Toolbar.



Hình 3.8

2. Các khái niệm cơ bản trong Unity

a) GameObject

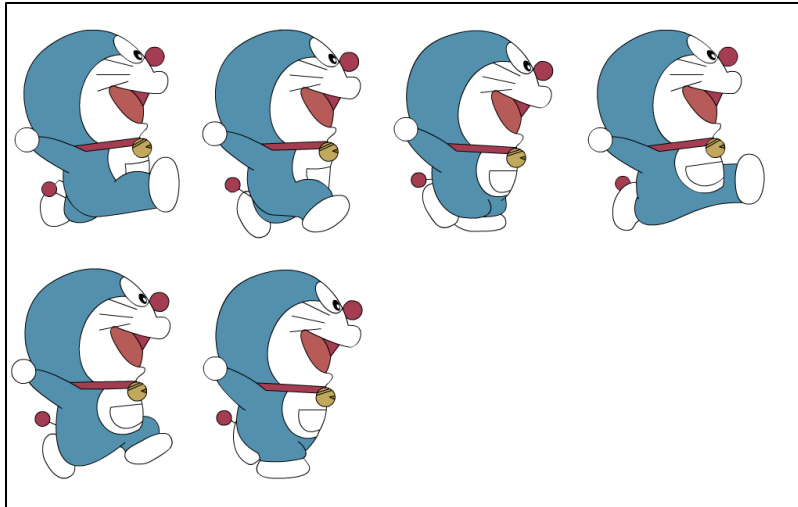
Một đối tượng cụ thể trong game gọi là một game object, có thể là nhân vật, đồ vật nào đó. Ví dụ: cây cối, xe cộ, nhà cửa, người...

b) Component

Một GameObject sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động (animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... mỗi thứ như vậy gọi là một component của GameObject.

c) Sprite

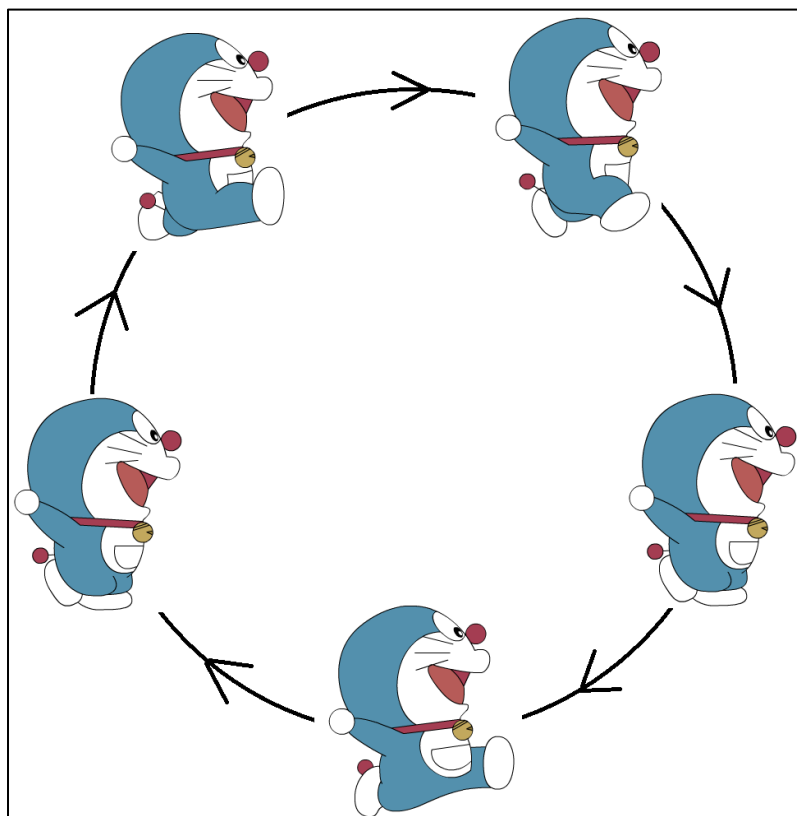
Là một hình ảnh 2D của một game object có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó.



Hình 3.9

d) Animation

Là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau.



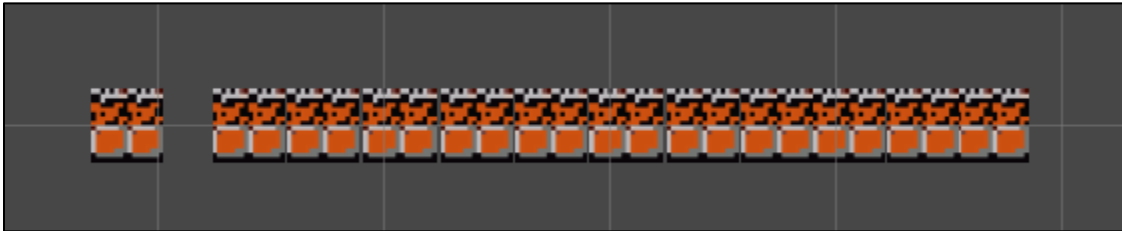
Hình 3.10

e) Key Frame

Key Frame hay Frame là một trạng thái của một animation. Có thể được tạo nên từ 1 sprite hay nhiều sprite khác nhau.

f) Prefabs

Là một khái niệm trong Unity, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay từ một đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiền trong game Mario đều có xử lý giống nhau, nên ta chỉ việc tạo ra một đối tượng ban đầu, các đồng tiền còn lại sẽ sử dụng prefabs. Hoặc khi ta lát gạch cho một cái nền nhà, các viên gạch cũng được sử dụng là prefabs.



Hình 3.11

g) Sounds

Âm thanh trong game.

h) Script

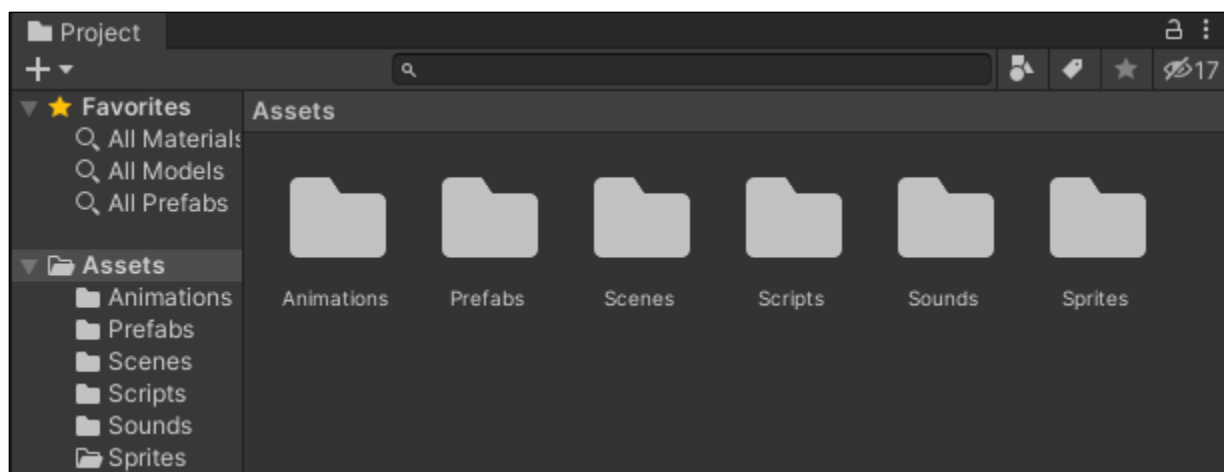
Script là tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Trong Unity có thể dùng C#, Java Script, BOO để lập trình Script.

i) Scenes

Quản lý tất cả các đối tượng trong một màn chơi của game.

j) Assets

Bao gồm tất cả những gì phục vụ cho dự án game như sprite, animation, sound, script, scenes... và được tổ chức dưới dạng cây thư mục như Windows Explorer.

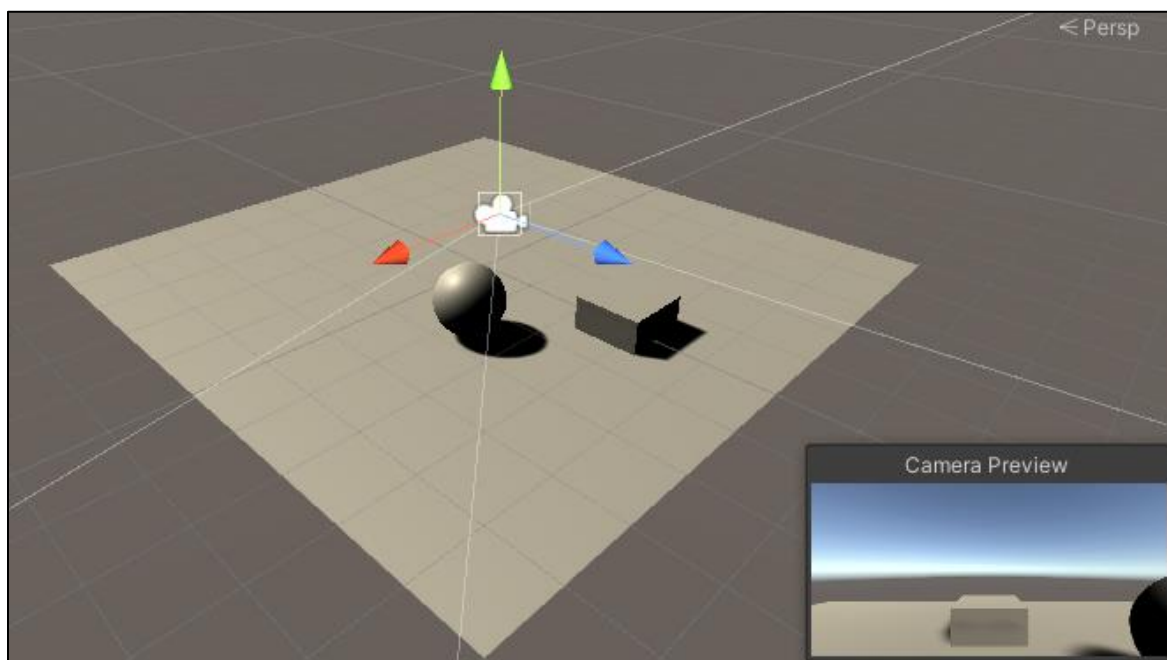


Hình 3.12

k) Camera

Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game.

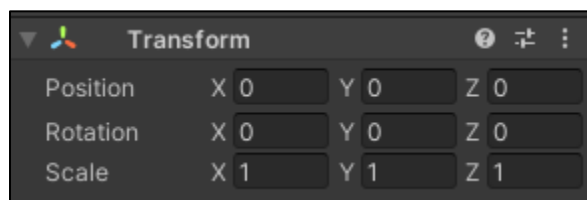
Theo mặc định có 2 loại camera: Editor Camera và Main Camera. Editor Camera cho phép chúng ta nhìn các đối tượng trong cửa sổ Scene và Main Camera là cái mà chúng ta sử dụng xuyên suốt quá trình chơi thử game.



Hình 3.13

1) Transform

Là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng.



Hình 3.14

Phần IV – SỬ DỤNG UNITY

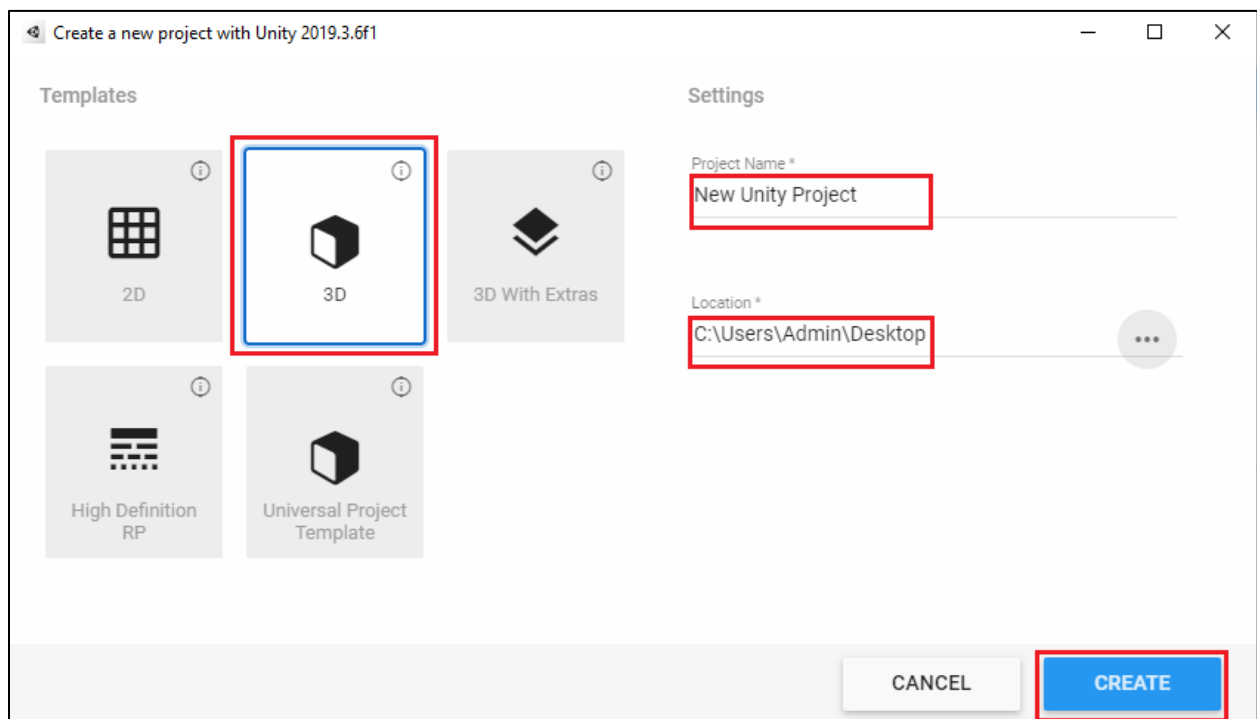
1. Unity 3D – Tạo một đối tượng 3D có thể di chuyển được

Mình sẽ hướng dẫn các bạn một bài khá đơn giản và cơ bản trong việc học Unity, đó là điều khiển một trái bóng di chuyển các hướng và nhảy lên bằng bàn phím.

Bước 1:

Tạo project Unity 3D

- Mở Unity Hub, ở thẻ *Projects*, chọn *NEW*
- Chọn Templates là *3D*, điền các thông tin *Project Name* và *Location* và ấn chọn *CREATE*

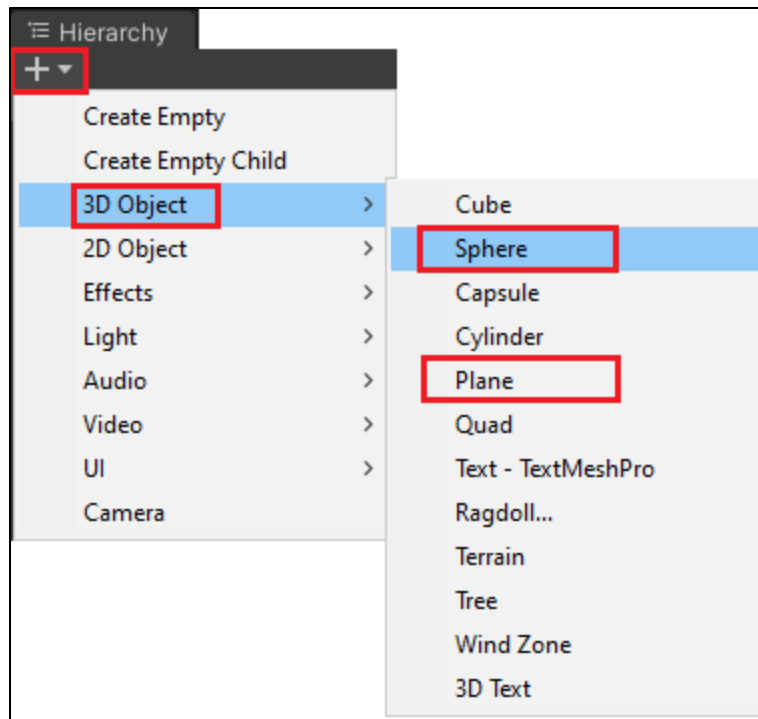


Hình 4.1

Bước 2:

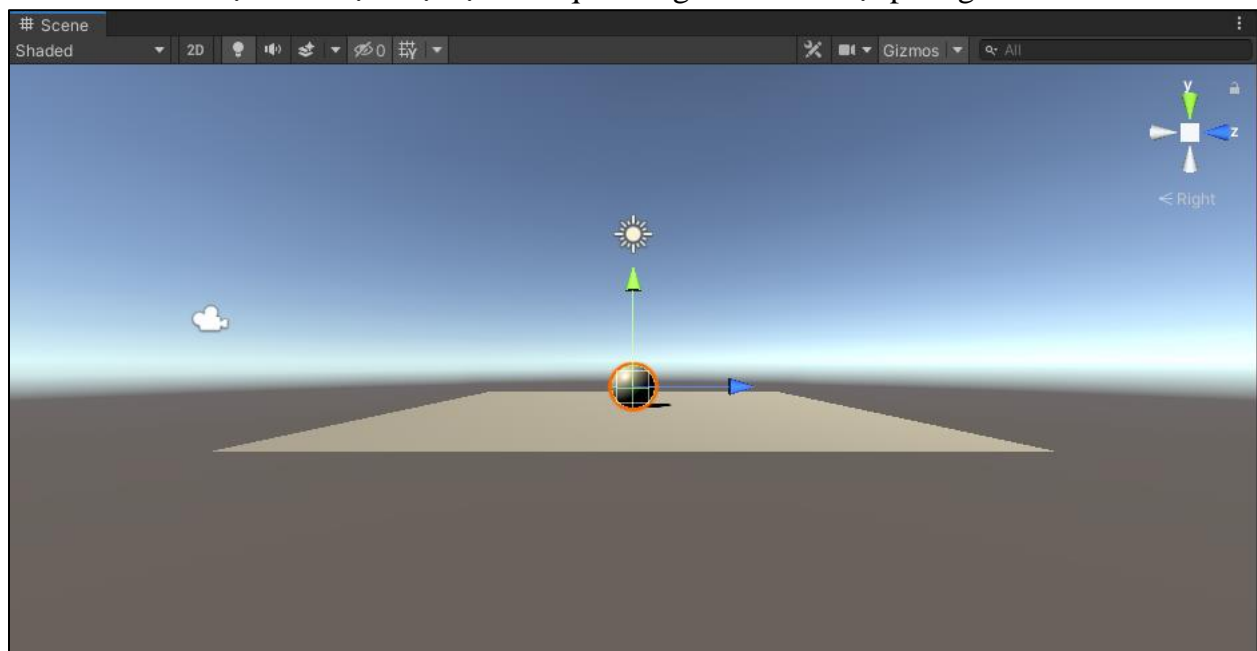
Tạo ra 1 quả bóng và 1 mặt phẳng để quả bóng có thể di chuyển trên đó

- Ở cửa sổ Hierarchy, ấn chọn vào nút xỏ *Create* có hình dấu +, chọn mục *3D Object* để thêm 2 đối tượng là *Sphere* và *Plane*



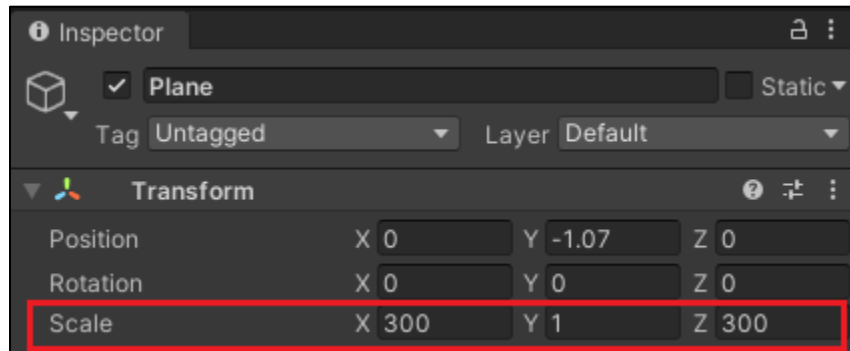
Hình 4.2

- Các bạn kéo trục tọa độ Y để quả bóng nằm trên mặt phẳng



Hình 4.3

- Sau đó kéo chiều dài và chiều rộng của mặt phẳng ra lớn hơn để quả bóng có thể di chuyển thoải mái bằng cách chỉnh thông số *Scale* của Plane trong cửa sổ Inspector.

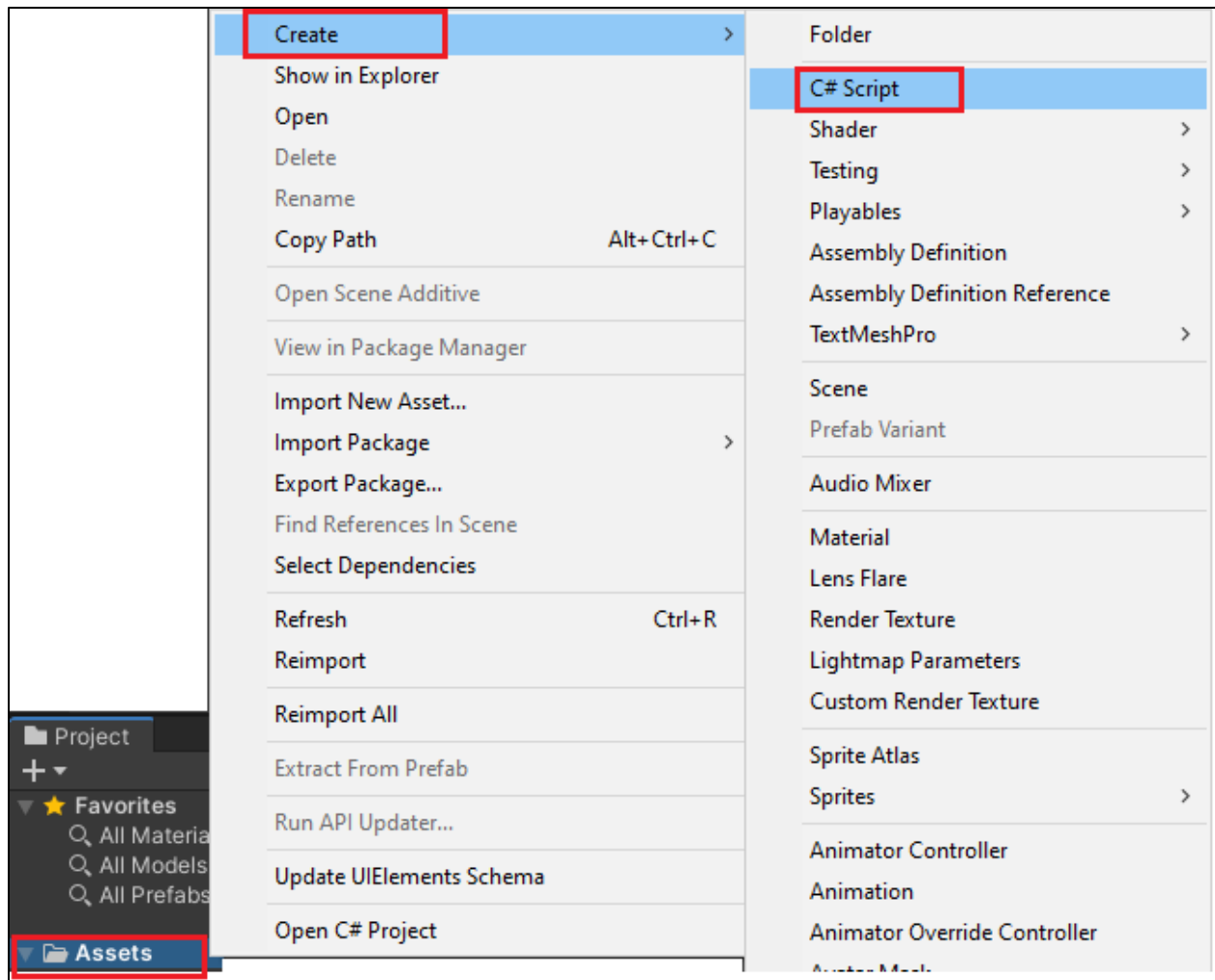


Hình 4.4

Bước 3:

Viết script để điều khiển quả bóng

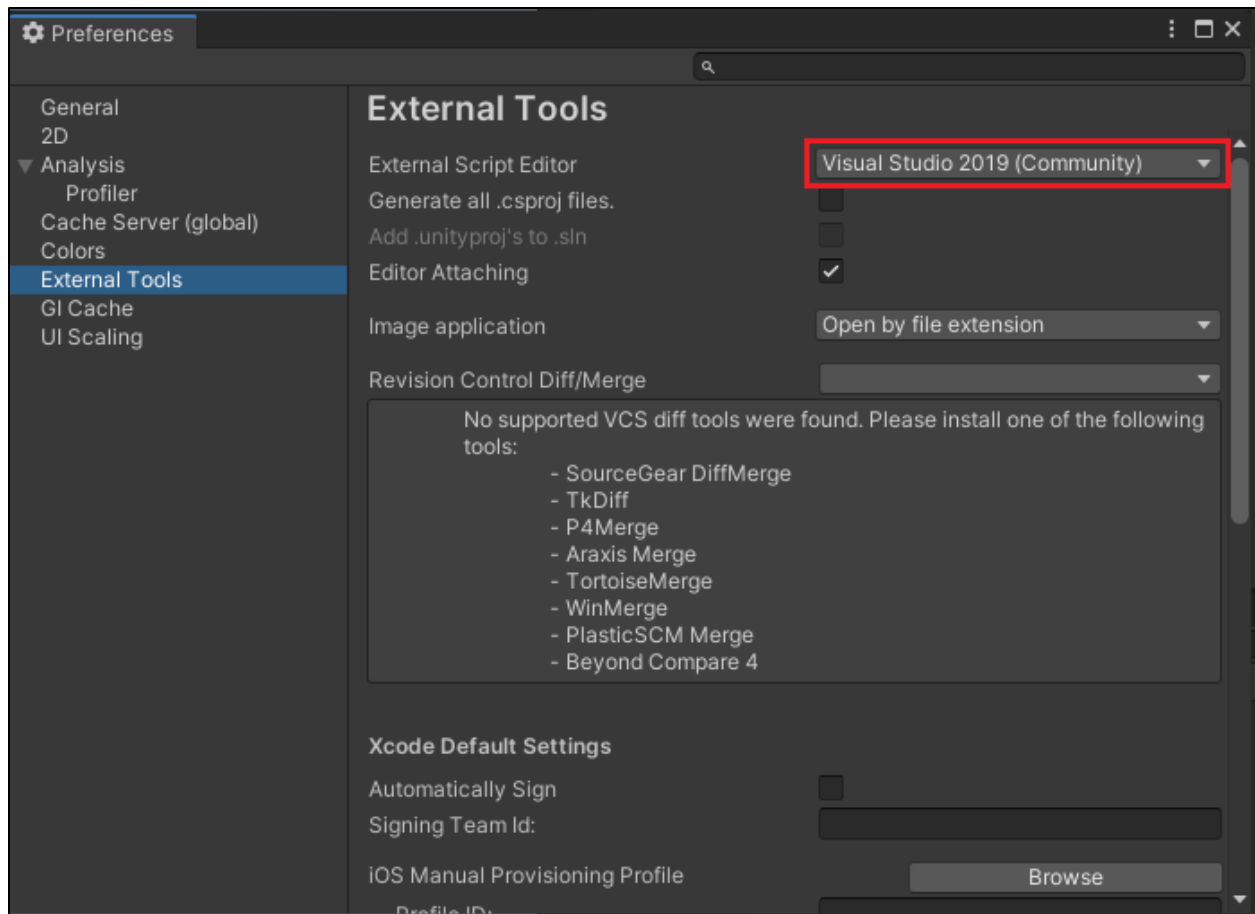
- Trong cửa sổ Project, chuột phải vào Assets -> Create -> C# Script (mình nghĩ là nhiều bạn đã quen với C#, các bạn cũng có thể chọn JavaScript hoặc Boo)



Hình 4.5

Một script mới sẽ được tạo ra, các bạn đặt tên cho nó. Mình đặt tên là “move”. Sau đó các bạn nhấp đúp vào script, cửa sổ editor sẽ được mở ra để chúng ta code trong đó.

Note: Dành cho các bạn chưa setup hoặc muốn thay đổi môi trường script editor. Các bạn vào menu *Edit -> Preferences*. Ở cửa sổ hiện ra, chọn thẻ *External Tools*, ở mục *External Script Editor* các bạn chọn trình editor mà mình muốn.



Hình 4.6

- Code của chúng ta sẽ như sau:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class move : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

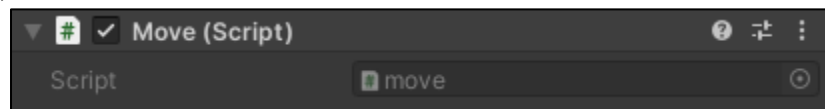
    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.Space))
        {
            GetComponent<Rigidbody>().AddForce(0, 5, 0);
        }
        if (Input.GetKey(KeyCode.UpArrow))
    }
}
```

```

    {
        GetComponent<Rigidbody>().AddForce(0, 0, 3);
    }
    if (Input.GetKey(KeyCode.DownArrow))
    {
        GetComponent<Rigidbody>().AddForce(0, 0, -5);
    }
    if (Input.GetKey(KeyCode.LeftArrow))
    {
        GetComponent<Rigidbody>().AddForce(-5, 0, 0);
    }
    if (Input.GetKey(KeyCode.RightArrow))
    {
        GetComponent<Rigidbody>().AddForce(5, 0, 0);
    }
}
}

```

Câu lệnh AddForce giúp chúng ta đẩy quả bóng đi với 1 lực theo 1 vector nào đó. Sau khi code xong các bạn ấn Ctrl + S để lưu lại rồi quay lại cửa sổ Unity, dùng chuột kéo file script move.cs vào đối tượng Sphere ở cửa sổ Scenes. Bây giờ thì đối tượng Sphere đã được điều khiển bằng script move (kiểm tra ở bên cửa sổ Inspector của Sphere)

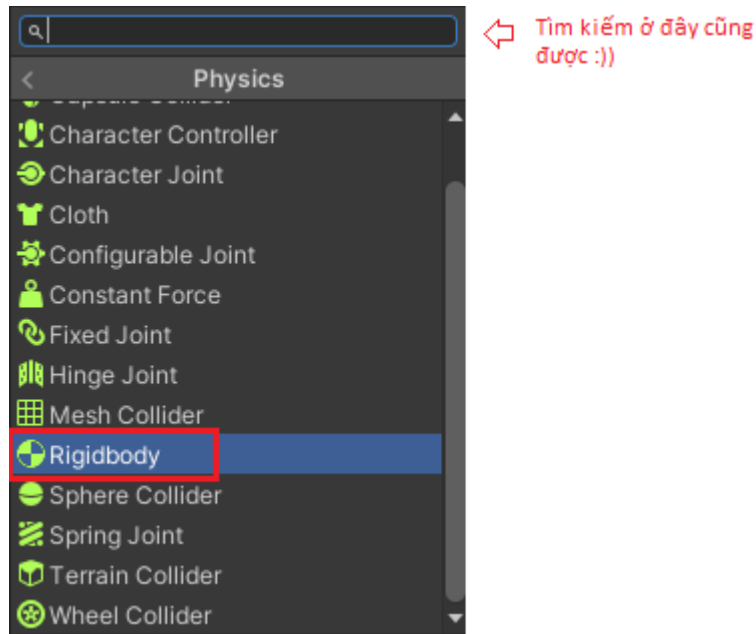


Hình 4.7

Bước 4:

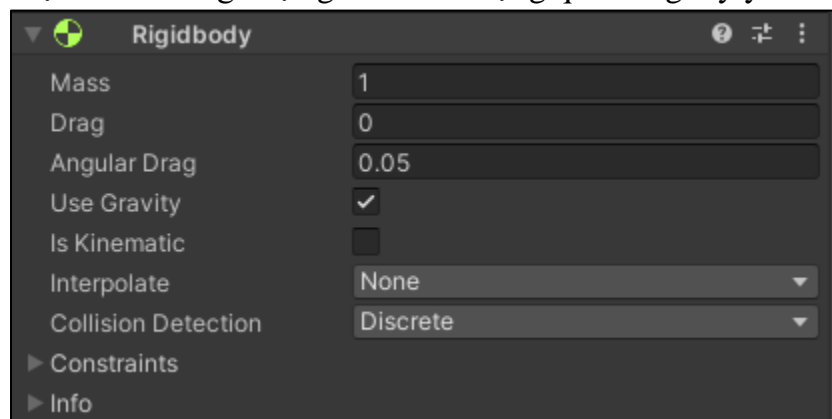
Thêm khối lượng cho quả bóng.

Để quả bóng có thể nhận lực tác động vào nó thì nó phải có khối lượng. Để tạo khối lượng cho quả bóng, ở cửa sổ Inspector của đối tượng Sphere, chọn *Add Component* -> *Physics* -> *Rigidbody*.



Hình 4.8

Bây giờ các bạn có thể tăng hoặc giảm khối lượng quả bóng tùy ý.



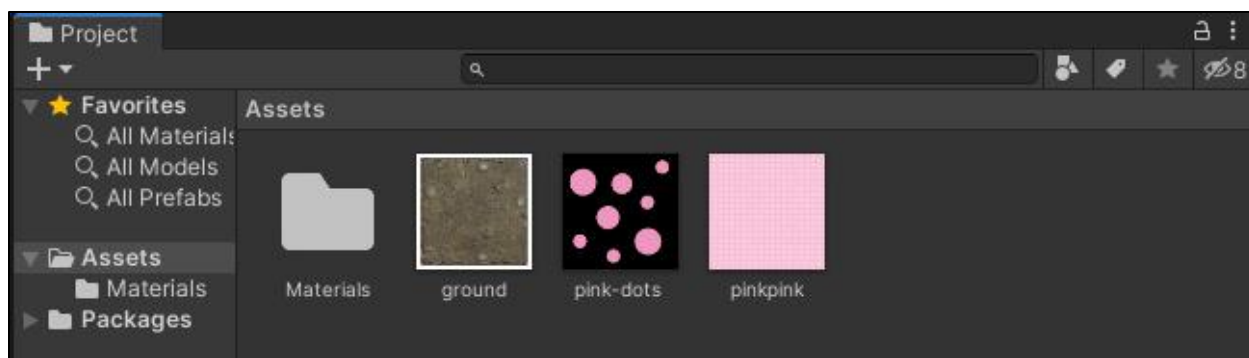
Hình 4.9

Vậy là xong! Bây giờ các bạn thử ấn Play và sử dụng các phím đã cài đặt trong script move.cs để di chuyển quả bóng.

Bước 5:

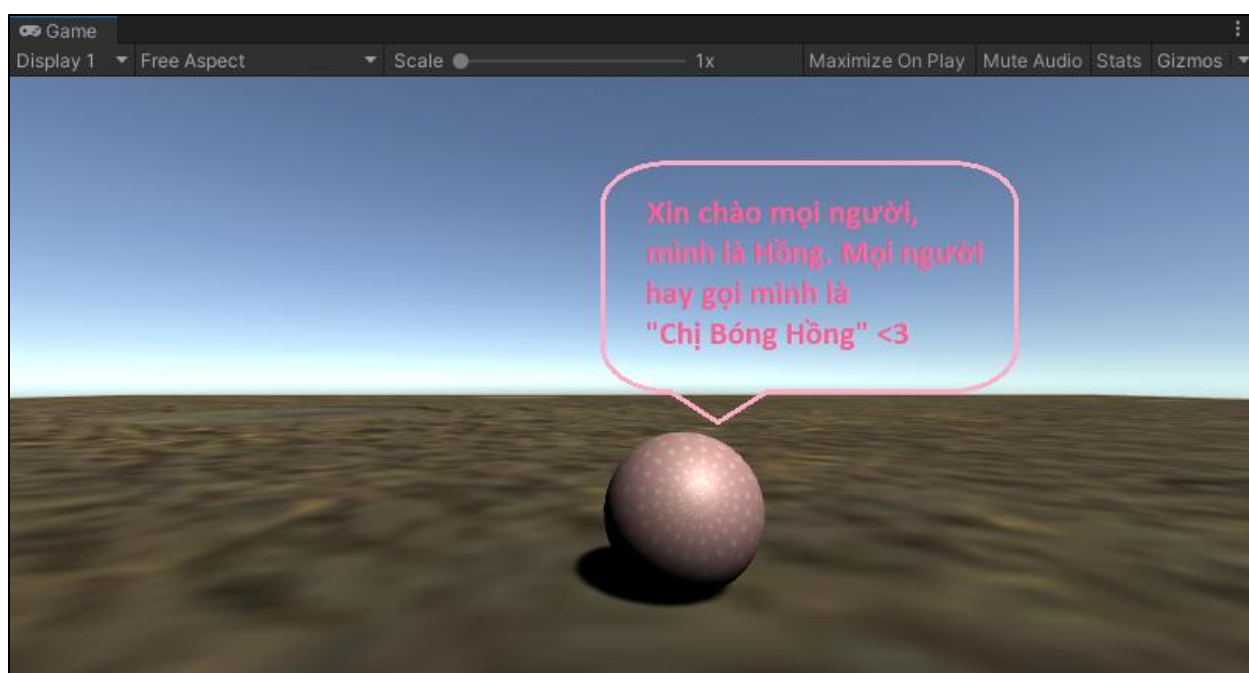
Để quả bóng chuyển động mượt mà hơn, hãy tô điểm thêm cho nó màu hồng, vì màu hồng trông rất nữ tính!

Các bạn có thể tải texture bất kỳ ở trên mạng về và kéo nó vào ô Assets của cửa sổ Project.



Hình 4.10

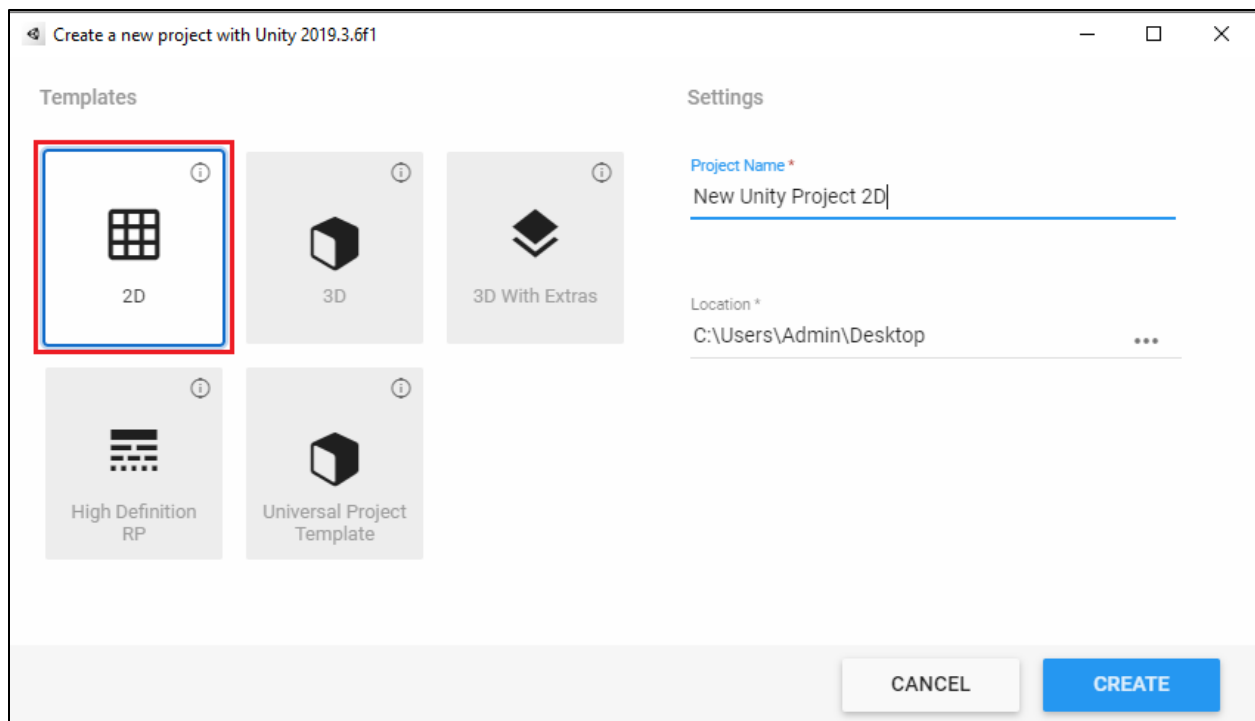
Sau đó các bạn dùng chuột kéo file đó thả vào quả bóng, vậy là quả bóng đã trông sống động hơn rồi!



Hình 4.11

2. Unity 2D

Hướng dẫn tạo và cấu hình project Unity 2D, cách tạo các đối tượng cơ bản, animation/animators, prefabs, xử lý vật lý và va chạm... các bạn có thể tham khảo tại series hướng dẫn làm game bằng Unity 2D tại [đây](#). Các bài viết được trình bày rất chi tiết và dễ hiểu, nên mình thiết nghĩ là dẫn nguồn vào thì sẽ tốt hơn là tổng hợp lại ;)



Hình 4.12

Phần V – UNITY VÀ MỘT SỐ GAME ENGINE TIÊU BIỂU KHÁC

Giống như “Car Engine”, “Game Engine” (GE) chính là bộ máy của một game và giúp game đó hoạt động. Việc lựa chọn một GE thích hợp là tối quan trọng mà đối với một game developer cần phải đặt lên hàng đầu để hướng tới hiệu suất tốt nhất cho sản phẩm. Câu hỏi đầu tiên và đầu đầu nhất một game dev cần phải giải quyết khi tạo ra một trò chơi đó chính là chọn engine nào sẽ mang lại trải nghiệm người dùng tốt nhất? Một GE không chỉ giúp tạo ra các game arcade cơ bản như Ping, Tetris, Snake, nó còn tạo ra nhưng game mang tính tiến hoá hay nhảy vọt của công nghệ như GTA hay Assassin’s Creed. Mặc dù hiện tại có khá nhiều GE trên thị trường, tuy nhiên khi đã là một game developer thì chắc hẳn phải nghe qua 3 loại GE nổi tiếng nhất: Unreal Engine, CryEngine và Unity3D.

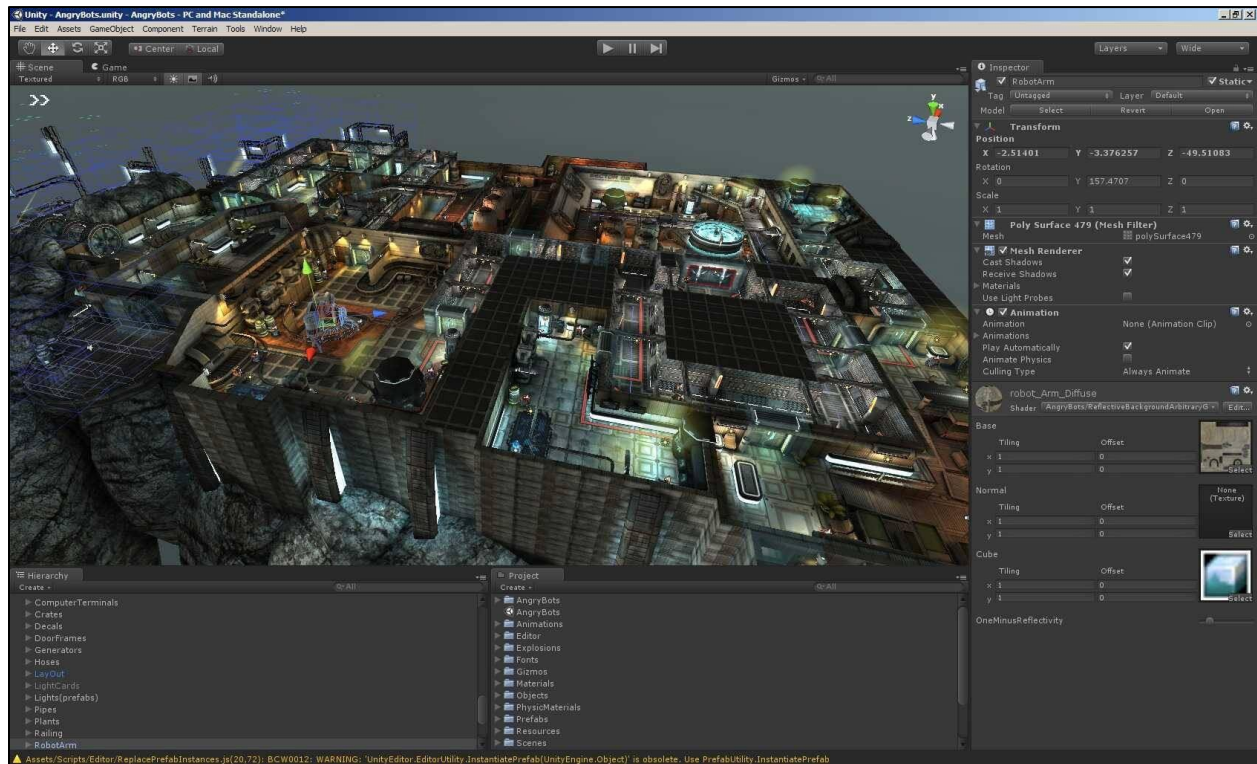


Hình 5.1

1. Unity 3D

Unity 3D được xem là 1 trong những GE tốt nhất trong thời điểm hiện tại với khả năng cung cấp đa dạng các công cụ và tính năng dễ dàng truy cập, sử dụng kể cả khi bạn không phải là người am hiểu công nghệ.

Tính năng đáng giá nhất của GE này chính là việc tích hợp đa nền tảng, cho phép game developers có thể thay đổi qua bất cứ platform nào như Android, Blackberry, iOS hay Windows. Trên thực tế, Unity hỗ trợ tới 25 nền tảng, nhiều hơn bất cứ một GE đang hiện hữu trong thời điểm hiện tại.



Hình 5.2

Cộng đồng Unity có một số lượng đông đảo có thể giúp bạn ngay lập tức. Một tính năng khác giúp Unity trở thành “Best 3D Game Engine in 2018” chính là ở việc nó hỗ trợ rất nhiều định dạng file 3D của các ứng dụng hàng đầu như là 3D Max, Blender, CINEMA, Maya, Softimage... Ngoài ra, người lập trình viên có thể truy cập và sử dụng hơn 15,000 free + paid 3D model, audio, animations, editor extensions, materials...

Unity 3D sử dụng C# hoặc JavaScript, là những ngôn ngữ được ưa chuộng hơn C++, và điều hoàn toàn ít khó khăn hơn khi mà chuyển đổi từ Java sang C# so với việc chuyển

từ Java sang C++. Tuy nhiên, Unity 3D lại có một giao diện gọn gàng và nhanh chóng, đủ nhẹ để có thể chạy trên cả Windows XP Service Pack 2 (SP2)!

2. Unreal Engine

Là nền tảng phát triển game mới nhất được phát hành bởi tập đoàn chuyên về làm game và phần mềm lớn nhất nước Mỹ - Epic Games Companies – vào khoảng đầu năm 2014, Unreal Engine 4 được kế thừa từ bộ Unreal Development Kit, hay còn được biết tới như là UDK trong thế giới gaming.

Unreal Engine 4 (UE4) ra đời với một khả năng đồ họa đáng kinh ngạc, cho phép mang lại những trải nghiệm game ở một tầm cao mới. Với hệ thống xử lý hạt mới (Particle System, xem thêm tại [đây](#)), Unreal Engine có thể xử lý hàng triệu hạt trong một phân cảnh tại một thời điểm.



Hình 5.3

Hơn thế nữa, UE4 còn hoàn toàn miễn phí, bạn chỉ phải trả 5% số tiền kiếm được cho các sản phẩm sử dụng engine này. Có nghĩa là, Epic Games sẽ có được 5% toàn bộ số tiền bạn có thể thu lại từ game, mua bán đồ trong game, quảng cáo trong game hoặc tính tiền trên mỗi user... Tuy nhiên nhà sáng lập lại cho phép các lập trình viên sử dụng phiên bản full version một cách miễn phí trong trường hợp tổng doanh thu từ game là trên 3000\$/1 quý!

Một trong những thay đổi đáng kể ở UE4 đó chính là ngôn ngữ kịch bản. Các nền tảng Unreal từ trước tới nay đều chạy trên UnrealScript. Thế nhưng từ UE4 toàn bộ đã được thay thế bằng C++, Kismet (script editor cũ) đã được thay thế bằng một hệ thống Blueprint trực quan hơn. Một điểm đáng lưu ý nữa là nếu bạn muốn phát triển game cho các thế hệ máy game đời cũ thì bạn sẽ không thể thực hiện được với UE4. Khả năng đồ họa tuyệt vời tất nhiên phải đánh đổi với khả năng hỗ trợ trên các nền tảng màn hình công nghệ cũ!

3. CryEngine

CryEngine là một nền tảng game cực kỳ mạnh, được thiết kế bởi công ty Crytek, những người đã phát triển game Far Cry huyền thoại đầu tiên. Những tính năng đồ họa của CryEngine vượt xa Unity và mạnh ngang ngửa khả năng mà UE4 có thể thực hiện. Cho dù là vậy, bạn cũng sẽ phải mất một chút thời gian để nghiên cứu học hỏi trước khi có thể sử dụng nó thành thạo, và sẽ càng khó hơn nếu bạn chưa có một tí kinh nghiệm nào làm việc với các nền tảng game khác trước đó.

CryEngine hỗ trợ Công nghệ thực tế ảo (Virtual Reality) và có hiệu ứng hình ảnh đáng kinh ngạc, bao gồm hệ thống kết xuất đám mây và sương mù giúp tái hiện lại các hình ảnh 3D chân thực và sống động nhất. Hơn nữa, phần tuyệt vời nhất khi lựa chọn CryEngine đó là việc nó không yêu cầu người dùng phải trả các chi phí bản quyền trong suốt quá trình làm game. Tuy nhiên, bạn phải trả một khoản phí cố định là 9.90\$ / tháng để có thể truy cập vào CryEngine(??? Nguồn: Google). Bên cạnh đó, CryEngine có một diễn đàn Q&A có tên là CryEngine Answers, có thể giải thích mọi câu hỏi và hoài nghi của bạn, mang lại trải nghiệm sử dụng engine tốt hơn.



Hình 5.4

4. Kết luận

Để có thể có một cái nhìn tổng quan và so sánh Unreal, Unity vs CryEngine, chúng ta phải quan tâm đến những tính năng nổi bật nhất mà những engine này có thể mang lại.

Khi so sánh hiệu suất của Unity với Unreal, chúng ta chắc chắn rằng Unity là nền tảng tốt hơn cho việc lập trình ứng dụng mobile và game 2D/3D, trong khi đó Unreal sẽ thích hợp hơn cho việc tạo ra những game có đồ họa và khả năng ánh sáng, quang học cao hơn.

Ở mặt khác, CryEngine cũng nắm giữ khả năng tạo ra games có đồ họa ấn tượng. Bên cạnh đó, khi so sánh CryEngine và Unreal trên các cân về sự cung cấp các tính năng theo mức chi phí phải trả, thì CryEngine hoàn toàn đánh bại Unreal với hình thức chi trả hiệu quả của mình. Tuy nhiên, đối với một beginner, trên cơ sở sự dễ dàng và tối giản, thì Unity hay Unreal rõ ràng là đáng để thử qua trước so với CryEngine, mặc cho những tính năng vượt trội mà engine này có được!

Mỗi một engine trong 3 loại trên đều có những đặc điểm và tính năng ưu việt riêng. Để có thể lựa chọn ra đâu mới là engine tốt nhất là điều khó có thể làm được. Do đó, việc lựa

chọn engine nào phải dựa trên project bạn đang thực hiện. Bạn phải nắm rõ được môi trường hiện thực project, chẳng hạn như, kinh phí bản quyền, nền tảng game, số chiều không gian (2D hay 3D) .v.v.

Phần VI – TỔNG KẾT

Unity Engine thực sự rất mạnh và lại vô cùng đơn giản để tiếp cận cho người mới bắt đầu, như phần đã trình bày trên đây có thể thấy thao tác chủ yếu là kéo thả và cấu hình, còn phần code thì khá ngắn và dễ hiểu (tất nhiên để làm game phức tạp thì cần code nhiều hơn :D). Hi vọng thông qua báo cáo lần này các bạn sẽ có hứng thú hơn với việc làm game nói chung và sử dụng Unity nói riêng, và *hi vọng tất cả đều qua môn... (ba chấm)!*

Cảm ơn mọi người đã đọc <3

TÀI LIỆU THAM KHẢO VÀ TỔNG HỢP

- ✚ CryEngine vs Unreal Engine vs Unity – The best game engine ([source](#))
- ✚ So sánh Unity 5 và Unreal Engine 4 ([source](#))
- ✚ Bài viết giới thiệu Unity Engine ([source](#))
- ✚ Bài viết Tailieu_unity3d của 3DvietPro ([source](#))
- ✚ Series bài viết về làm game Unity 2D ([source](#))
- ✚ Building Levels in Unity – Volodymyr Gerasimov (book)
- ✚ Unity Tutorials ([source](#))

DANH MỤC HÌNH ẢNH ĐƯỢC SỬ DỤNG

Hình 1.1	Unity Logo
Hình 1.2	Sơ đồ thống kê top 1000 games mobile
Hình 1.3	Doanh số thị trường game toàn cầu, tính theo tỉ dollar
Hình 1.4	Các nền tảng được hỗ trợ của Unity
Hình 2.1	Truy cập vào Account -> Organization trên trang chủ Unity
Hình 2.2	Thêm một “Unity seat”
Hình 2.3	Link Unity với Github
Hình 2.4	Kết quả link Unity với Github thành công
Hình 2.5	Liên kết với Github Student Developer Pack
Hình 2.6	Chọn role của benefits
Hình 2.7	Điền thông tin xác thực của tài khoản Github Student
Hình 2.8	Email chứa key kích hoạt Unity Pro
Hình 2.9	Chọn tải về Unity Hub
Hình 2.10	Truy cập vào Account -> Manage License trên Unity Hub
Hình 2.11	Chọn thêm mới một License
Hình 2.12	Điền thông tin kích hoạt Unity Pro
Hình 2.13	Màn hình Manage License với bản Unity Pro vừa được kích hoạt
Hình 2.14	Chọn tải về Unity Editor
Hình 2.15	Chọn version của Unity Editor
Hình 2.16	Chọn các modules cần cài đặt đi kèm với Unity Editor
Hình 3.1	Unity Editor và vị trí các cửa sổ làm việc
Hình 3.2	Trục tọa độ Gizmo
Hình 3.3	Trục tọa độ để di chuyển object
Hình 3.4	Trục tọa độ dùng để xoay object
Hình 3.5	Trục tọa độ dùng để thay đổi kích thước object
Hình 3.6	Thêm một cửa sổ Scene mới
Hình 3.7	Thao tác với 2 cửa sổ Inspector cùng một lúc với chức năng Lock

Hình 3.8	Một số Layout của Unity
Hình 3.9	Doraemon sprites
Hình 3.10	Vòng lặp cho animation của Doraemon
Hình 3.11	Các ô gạch chính là prefabs
Hình 3.12	Tổ chức thư mục Assets
Hình 3.13	Camera và tầm nhìn của camera trong Unity
Hình 3.14	Bảng thông số Transform trong cửa sổ Inspector
Hình 4.1	Tạo project Unity 3D
Hình 4.2	Thêm mới 2 đối tượng Sphere và Plane
Hình 4.3	Thay đổi vị trí của Sphere cho nằm trên Plane
Hình 4.4	Mở rộng phạm vi của Plane
Hình 4.5	Tạo mới C# script
Hình 4.6	Thay đổi môi trường Script Editor
Hình 4.7	Kết quả sau khi thêm script move.cs vào object Sphere
Hình 4.8	Thêm component Physics/Rigidbody cho Sphere
Hình 4.9	Danh sách các thuộc tính của component Rigidbody trong Inspector
Hình 4.10	Một số texture trong Assets
Hình 4.11	Kết quả sau khi thêm texture cho đối tượng
Hình 4.12	Tạo project Unity 2D
Hình 5.1	CryEngine, Unreal Engine 4 vs Unity 5
Hình 5.2	Hình minh hoạ Unity 3D
Hình 5.3	Hình minh hoạ Unreal Engine
Hình 5.4	Hình minh hoạ CryEngine
