

Bài 1: Khởi tạo dự án 2D, tạo Game, Object và Sprite

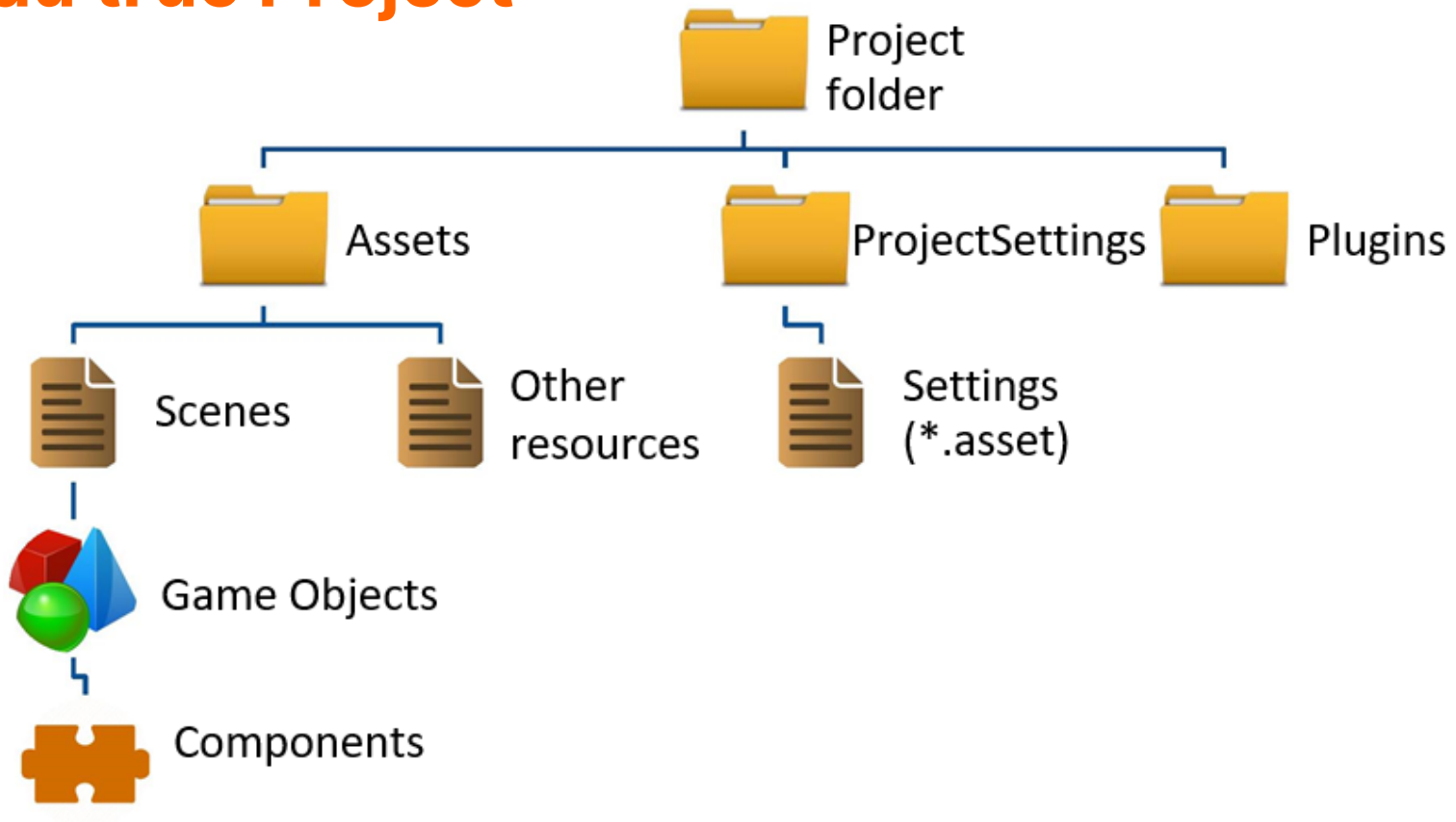


MỤC TIÊU

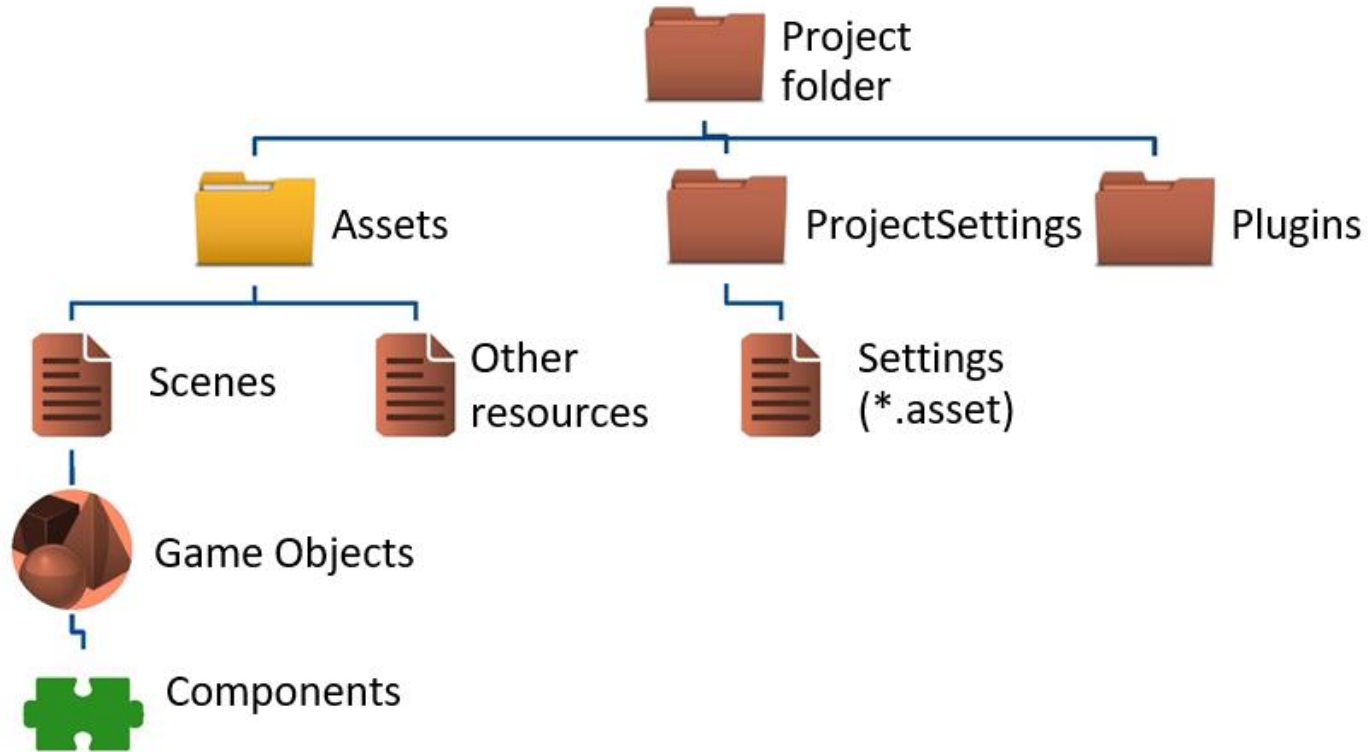
- . Cấu trúc Project
- . Khởi tạo và cấu hình dự án Game 2D




Cấu trúc Project



Cấu trúc Project - Assets



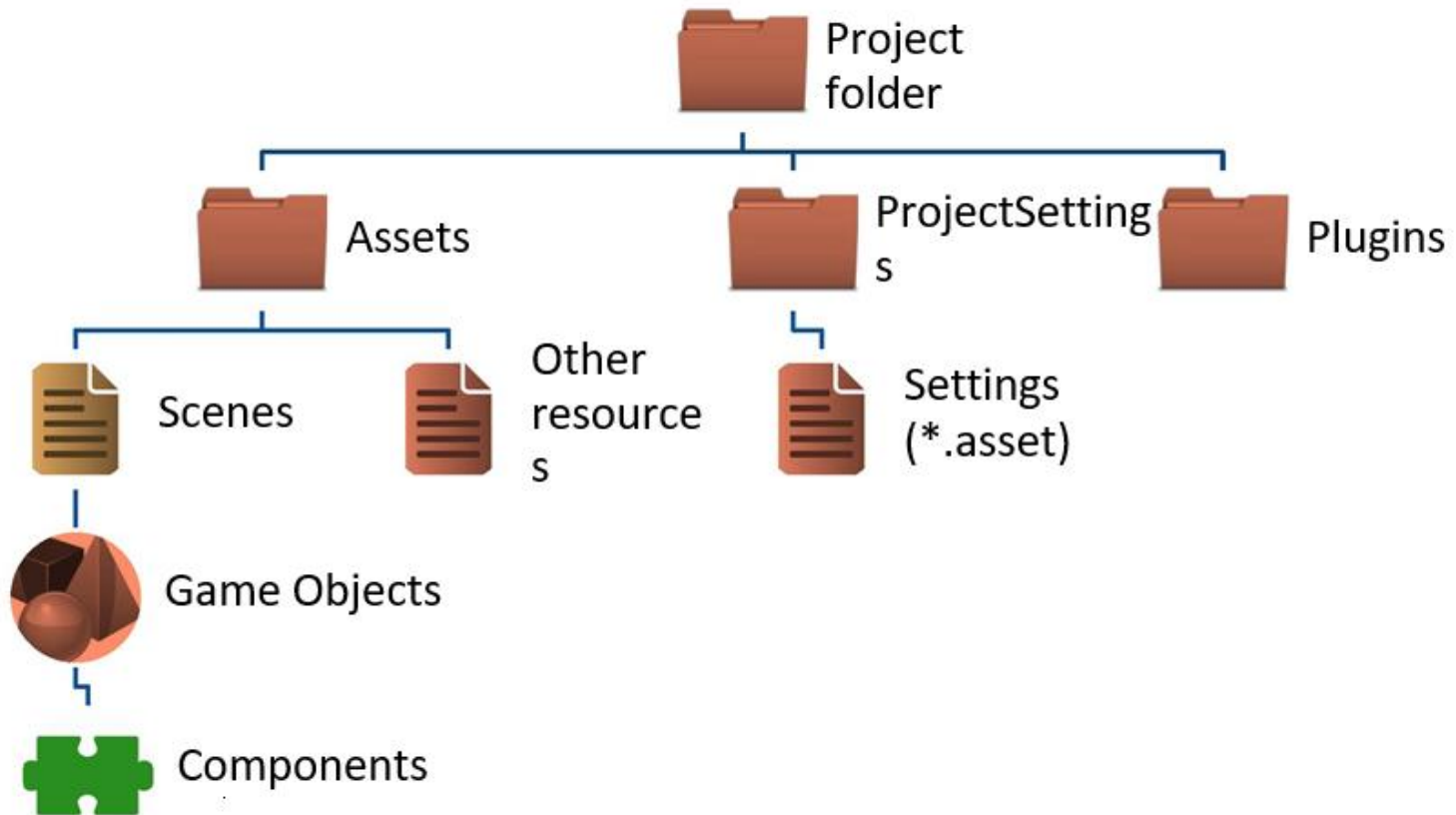
Cấu trúc Project - Assets

- . Asset là những tài nguyên xây dựng nên một dự án Unity như:
 - Tập tin hình ảnh;
 - Mô hình 3D;
 - Các tập tin âm thanh.
 - . Các tập tin mà chúng ta dùng để tạo nên trò chơi là tài sản (Assets).
 - . Tất cả các tập tin, thư mục của các dự án Unity đều được lưu trữ trong một thư mục có tên “Assets”.
- 

Cấu trúc Project - Assets

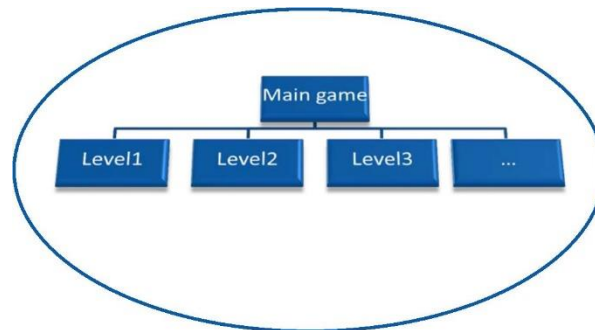


Cấu trúc Project - Scences

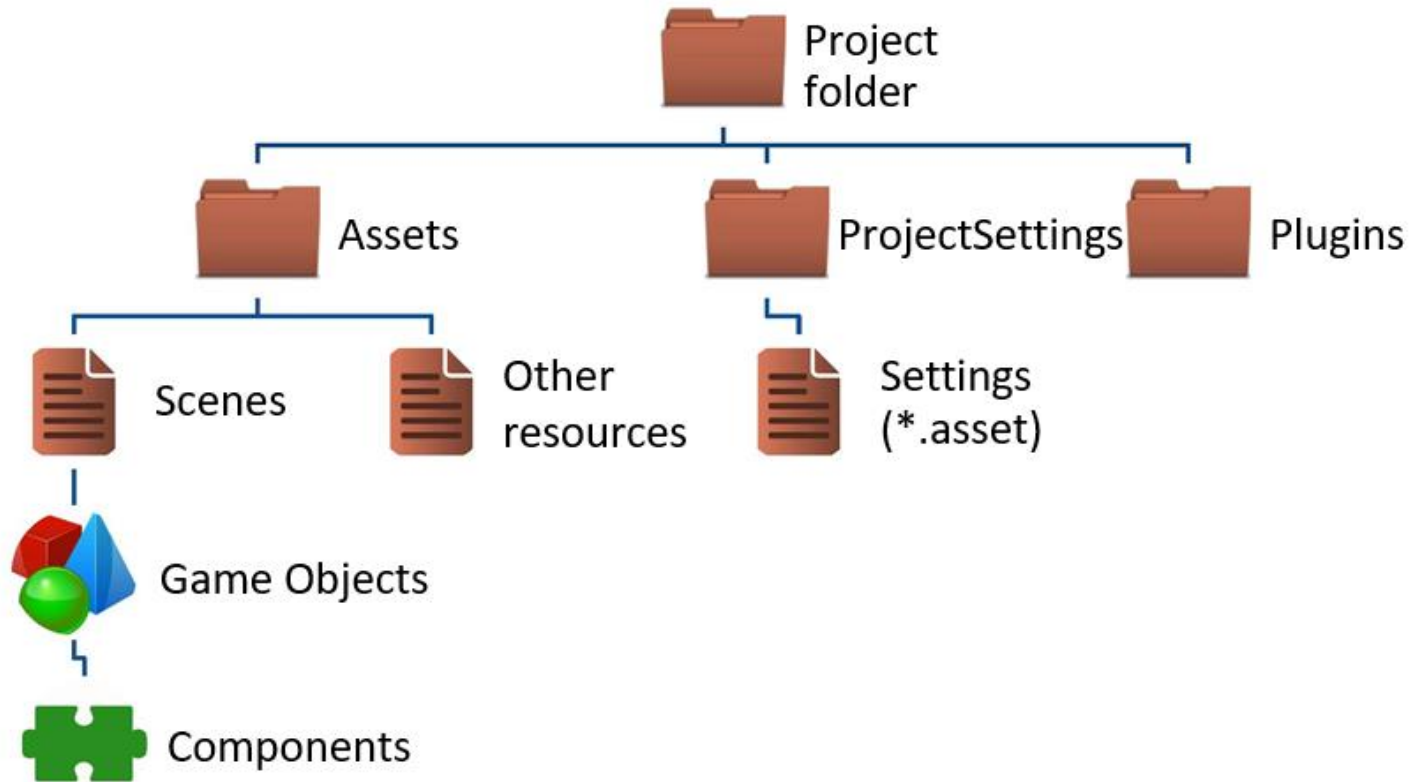


Cấu trúc Project – Scenes

- Trong Unity, scene- một cảnh (hay một phân đoạn) nghĩa là một màn chơi riêng biệt hoặc một khu vực hay thành phần có trong nội dung của trò chơi (ví dụ như Game menu).
- Bằng cách tạo nên nhiều Scene cho trò chơi, chúng ta có thể phân phối thời gian tải hoặc kiểm tra các thành phần khác nhau của trò chơi một cách riêng lẻ.
 - Mỗi scene là một level
 - Một game có thể có nhiều levels



Cấu trúc Project - GameObject



Cấu trúc Project – 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...
- . Khi Assets được sử dụng trong Scene, chúng trở thành GameObject – một thuật ngữ được sử dụng trong Unity (đặc biệt là trong mảng lập trình).
- . Game Object chứa ít nhất một thành phần là Transform.
- . Transform là thông tin về vị trí, góc xoay và tỉ lệ của đối tượng, tất cả được mô tả bởi bộ 3 số X, Y, Z trong hệ trục tọa độ. Thành phần này có thể được tùy biến lại trong quá trình lập trình nhằm thay đổi vị trí, góc quay và tỉ lệ của đối tượng (so với đối tượng gốc) qua các đoạn mã.

Cấu trúc Project - GameObject

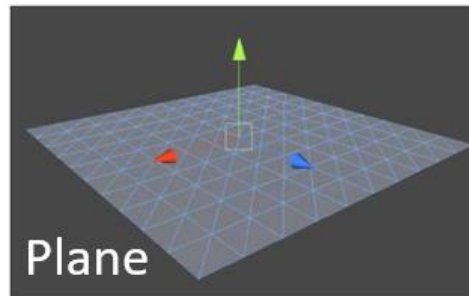
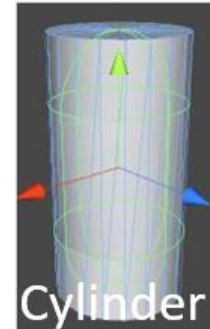
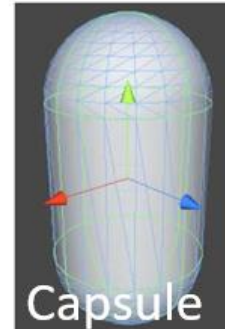
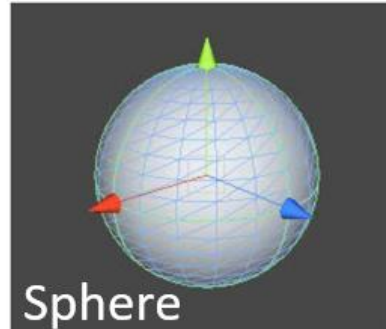
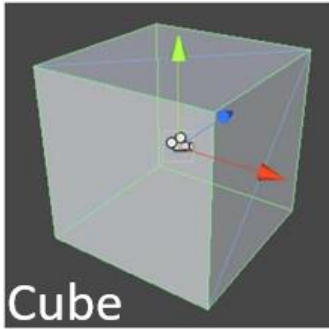
- GameObject là một lớp cơ bản trong Unity
- Mỗi GameObject chứa nhiều thành phần(components)
- Mỗi component chứa nhiều properties
- Có thể thay đổi giá trị property để điều khiển một object.
- Có thể thêm/sửa/xóa một component trong mỗi GameObject



Cấu trúc Project - GameObject

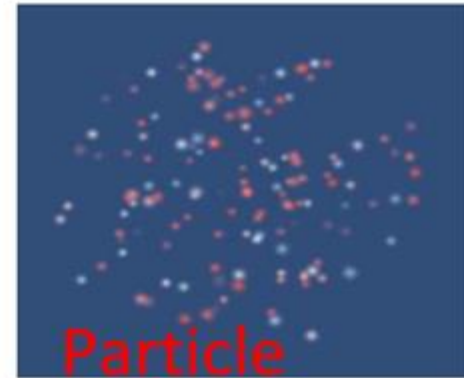
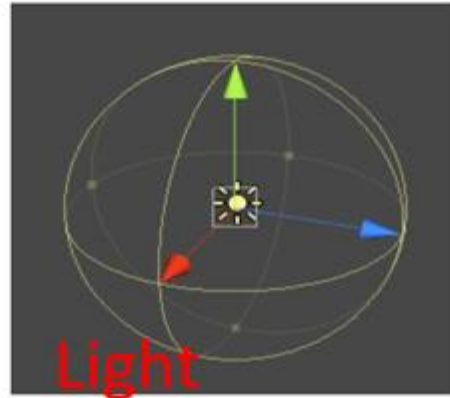
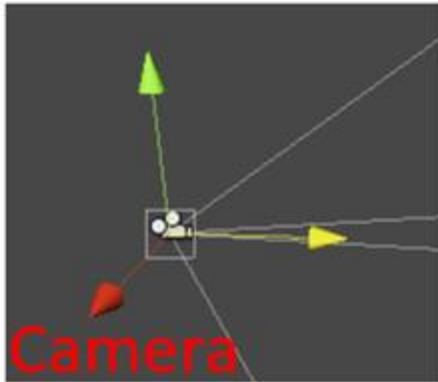
Ví dụ về GameObject:

- Object cơ bản: Cube, Sphere, Capsule, Cylinder, Plane



Cấu trúc Project – GameObject

- Ví dụ về GameObject:
 - Object đặc biệt: Empty Object, Camera, Light, Particle

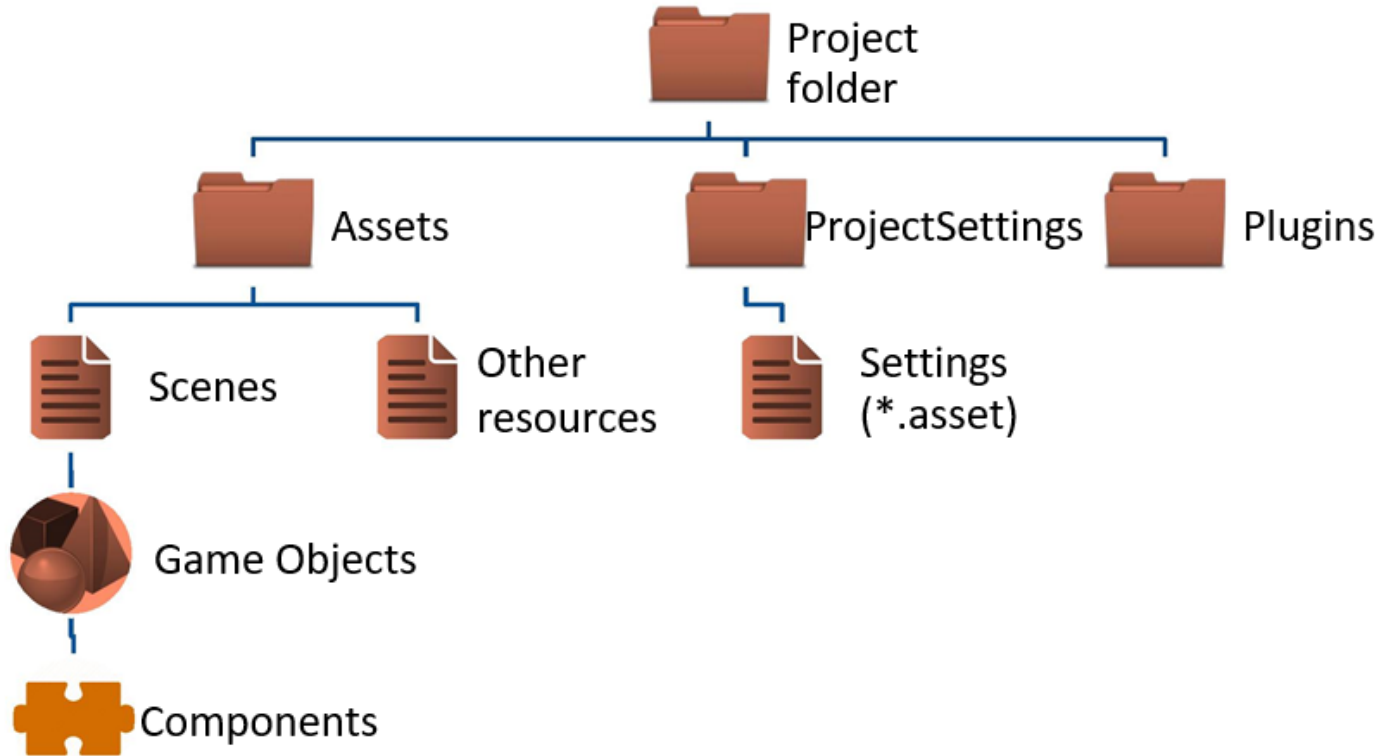


Cấu trúc Project- GameObject

- Ví dụ về GameObject:
 - Object mô hình: table, house, man, room,...



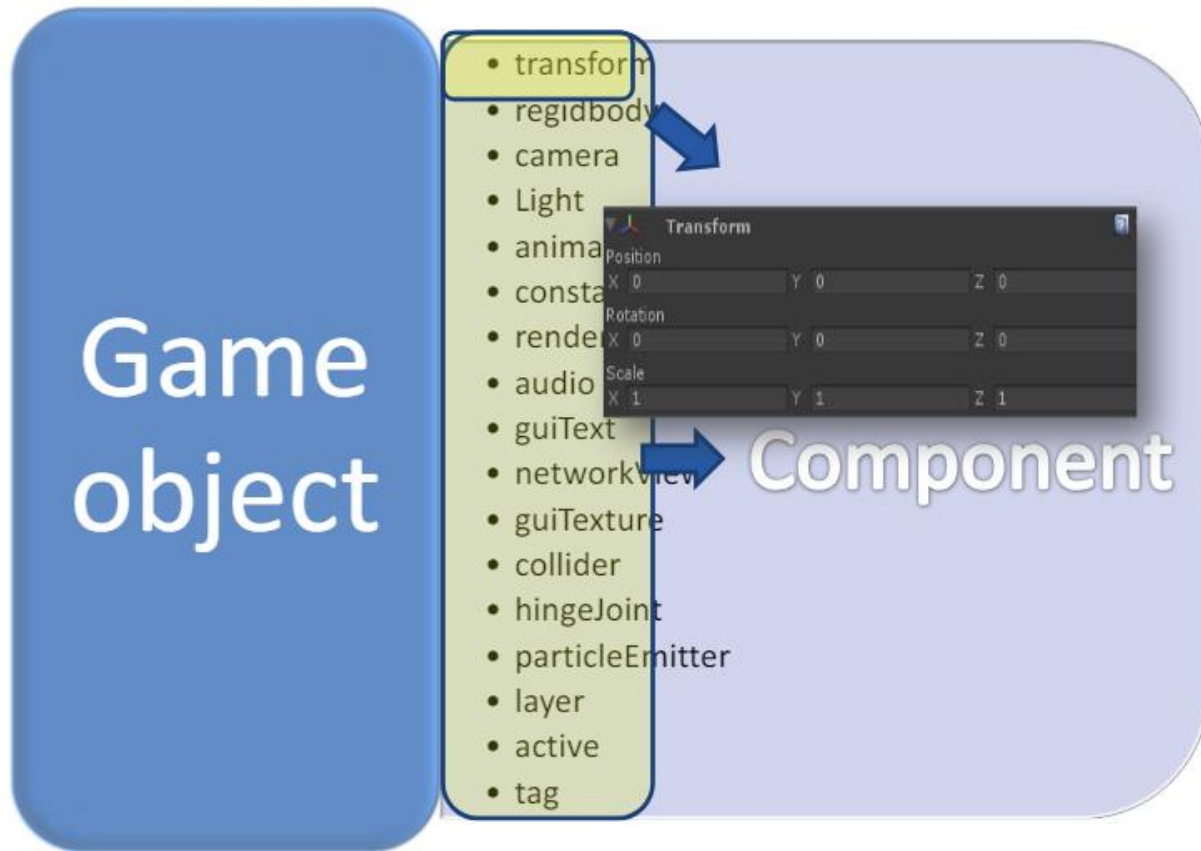
Cấu trúc Project- Components



Cấu trúc Project- Components

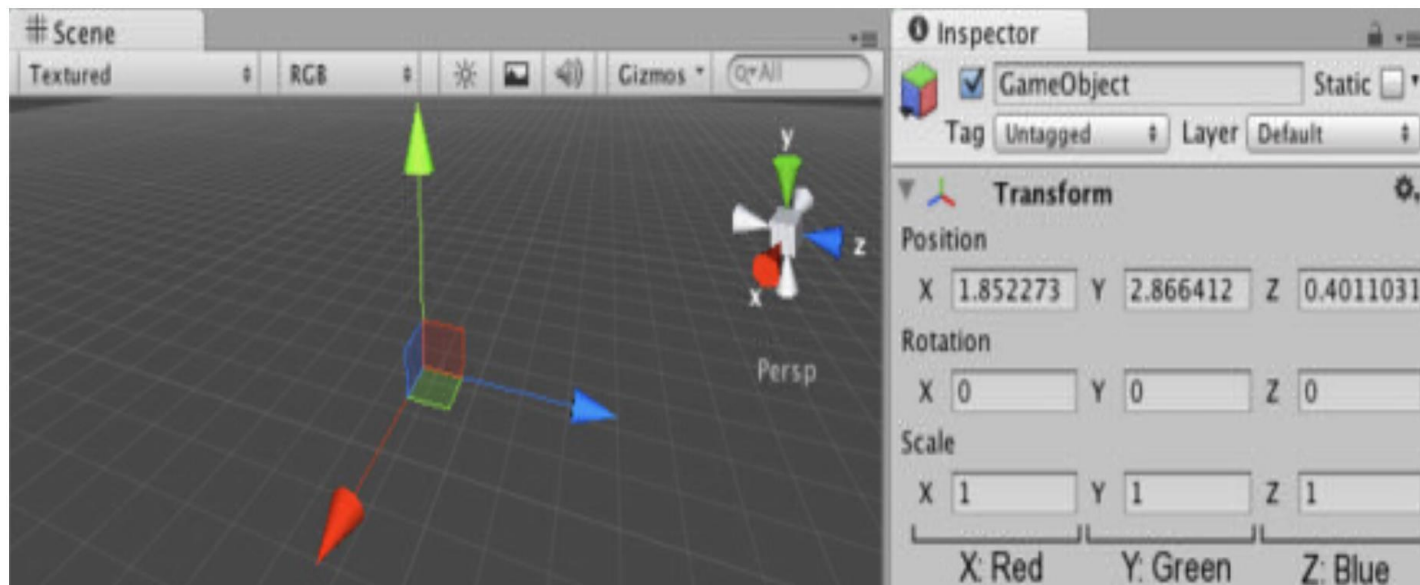
- . Một Game Object sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động(Animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... Mỗi thứ như vậy gọi là một component của GameObject.
- . Bằng cách “gắn” (attach) chúng vào trong Game Object, chúng ta có thể ngay lập tức áp dụng tác động của chúng lên đối tượng.
- . Những Components phổ biến trong quá trình phát triển trò chơi đều được Unity hỗ trợ sẵn.
- . Để tạo nên các yếu tố tương tác trong trò chơi, chúng ta sẽ sử dụng Script (mã kịch bản), chúng cũng được xem như là một Components trong Unity.

Cấu trúc Project - Components

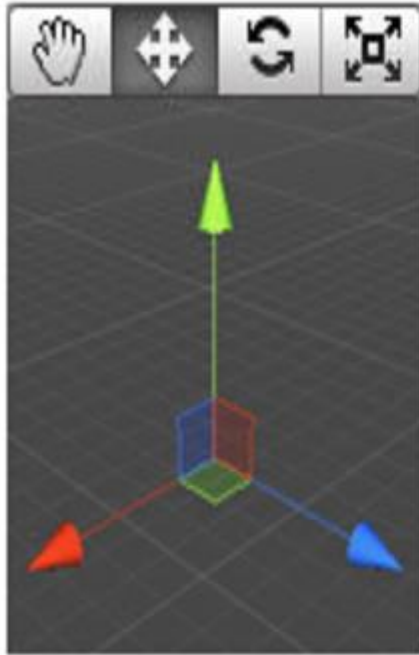


Cấu trúc Project- Transform Components

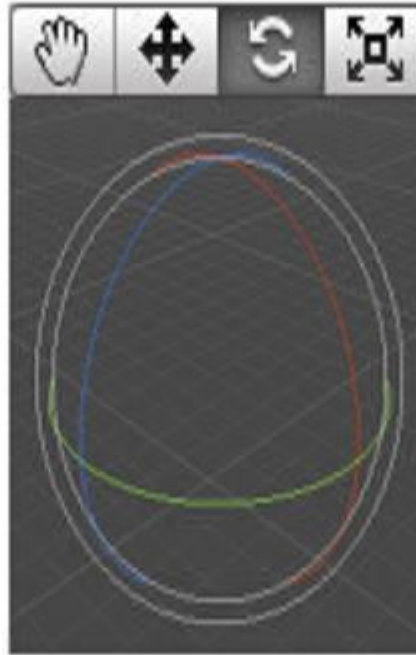
- Position: in X, Y and Z coordinates
- Rotation: around X, Y, and Z axes, measured in degrees
- Scale: along X, Y, and Z axes value "1" the original size.



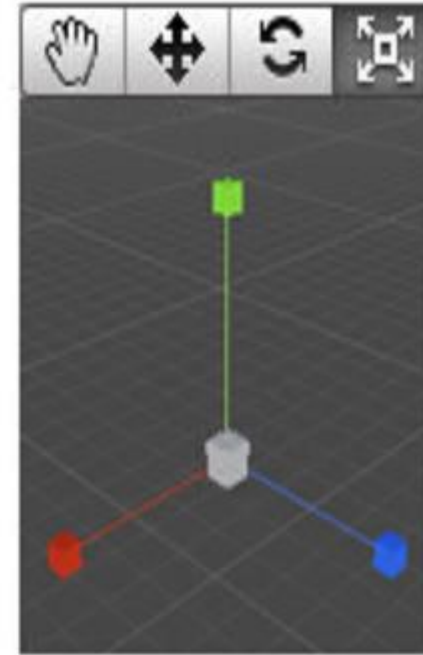
Cấu trúc Project- Object Transform



Translate (W)



Rotate (E)



Scale (R)

Cấu trúc Project- Script

- . Script được Unity xem như một Component.
- . Unity cung cấp cho chúng ta khả năng viết Script bằng cả 3 ngôn ngữ là JavaScript, C# và Boo (một dẫn xuất của ngôn ngữ Python).
- . Unity đã xây dựng sẵn một tập hợp đa dạng các Class, Function, ... để có thể ứng dụng trong quá trình lập trình.
- . Để viết Script, chúng ta sẽ làm việc với một trình biên tập Script độc lập của Unity, hoặc với chương trình Mono Developer được tích hợp và đồng bộ với Unity.
- . Mono Developer là một IDE khá tốt để lập trình khi cung cấp đầy đủ các chức năng của một IDE hoàn chỉnh tương tự Visual Studio. Mã nguồn viết trên Mono Developer sẽ được cập nhật và lưu trữ trong dự án Unity.

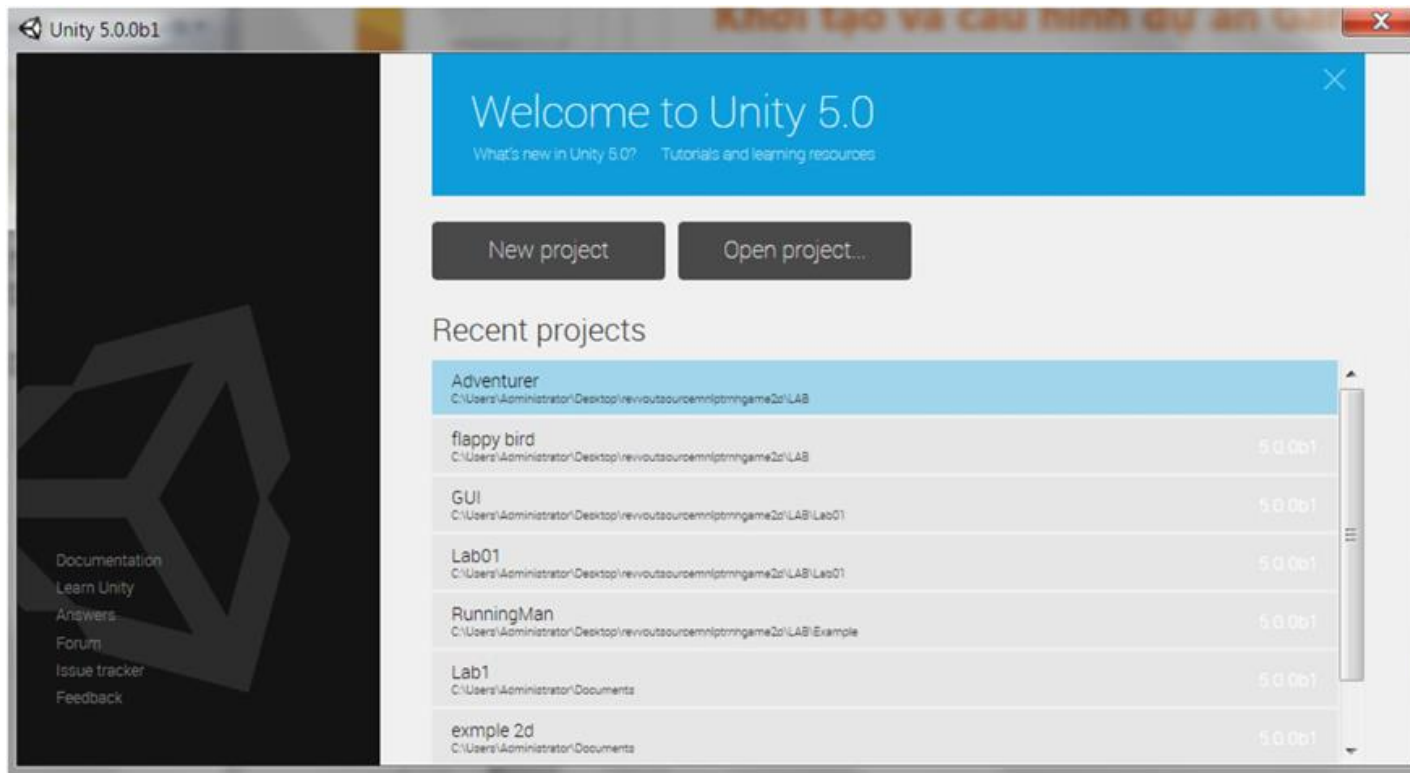
Cấu trúc Project- Prefabs

- . Là một khái niệm trong Unity dùng để sử dụng lại các đối tượng giống nhau có trong game bằng cách 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.

Khởi tạo và cấu hình dự án Game 2D

Tạo dự án

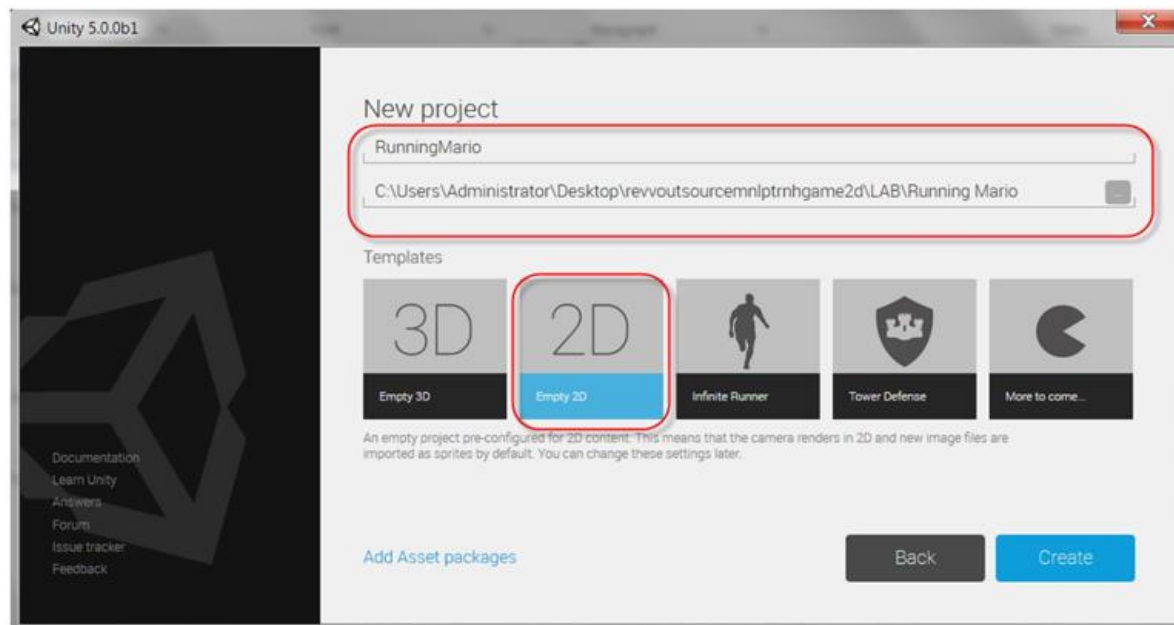
- . Bật Unity lên, vào menu File/New Project



Khởi tạo và cấu hình dự án Game 2D

Tạo dự án

- Chọn loại project là 2D sau đó gõ tên project xong nhấn Create Project.



Khởi tạo và cấu hình dự án Game 2D

Tạo dự án

- Chọn loại project là 2D sau đó gõ tên project xong nhấn Create Project. Kết quả như sau:



Khởi tạo và cấu hình dự án Game 2D

Cấu hình dự án 2D


- Sau khi tạo dự án xong, ở Hierarchy sẽ có một đối tượng là Main Camera. Chúng ta cần điều chỉnh kiểu hiển thị Scene là 2D



- Ở cửa sổ Hierarchy chọn Main Camera. Ở cửa sổ Inspector sẽ hiển thị các thông số của camera, ta sẽ chọn lại giá trị Projection hay phép chiếu là phép chiếu vuông góc thay cho phép chiếu phối cảnh.

Khởi tạo và cấu hình dự án Game 2D

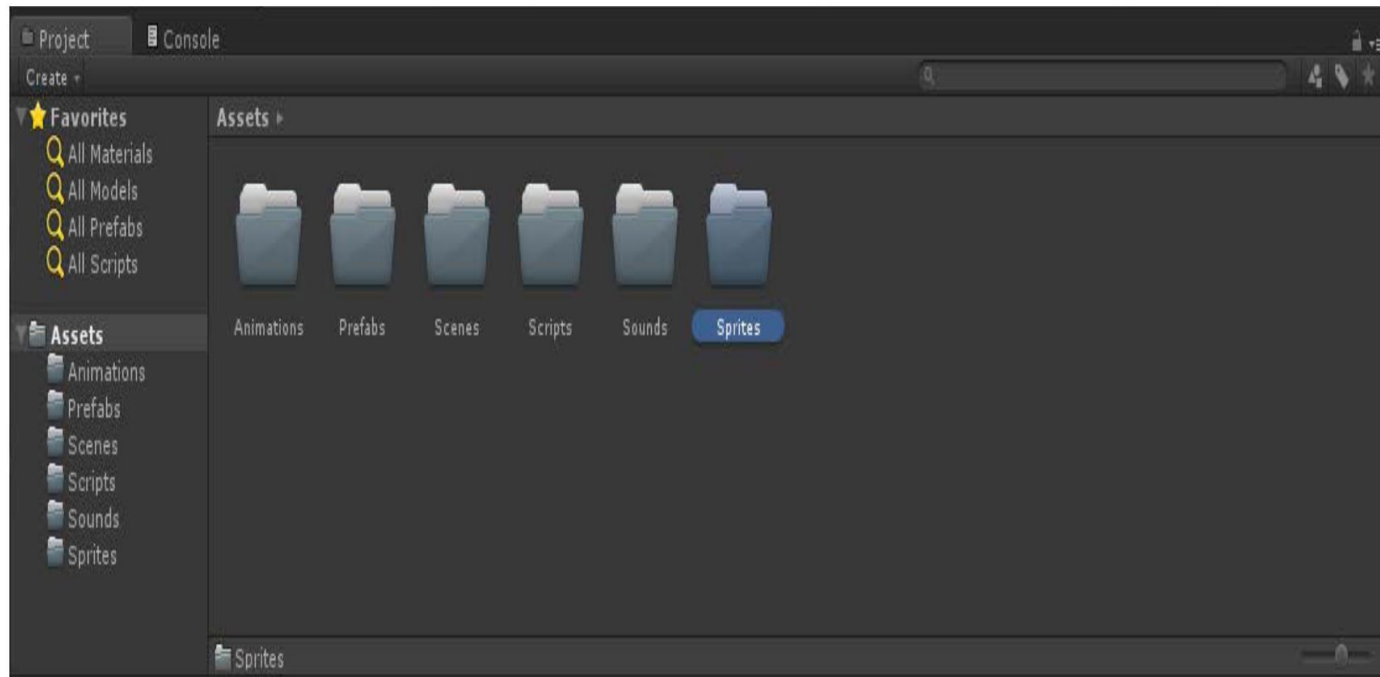
Cấu hình dự án 2D

- . Tiếp theo, ở cửa sổ Project, ta sẽ tạo sẵn các thư mục để chứa tất cả các tài nguyên có sử dụng trong game sau này lần lượt là: Animations, Prefabs, Scripts, Sprites, Sounds, Scenes.
 - . R-Click vào thư mục Assets ở cửa sổ Project, ta chọn Create, chọn Folder.
- 

Khởi tạo và cấu hình dự án Game 2D

Cấu hình dự án 2D

- Kết quả như hình sau:



THANKS FOR
WATCHING!

