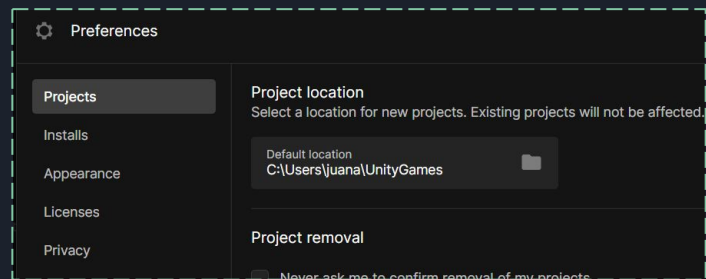


The background is a dark navy blue. On the left, there are two overlapping parallelogram shapes, one blue and one light green. In the bottom left, there is a circular inset showing a detailed, high-contrast image of a printed circuit board (PCB) with various electronic components. In the top right corner, there is a faint, stylized graphic of a circuit board layout.

# HLC

2° DAM/W

# Etiquetas



Crearemos una carpeta específica para almacenar nuestros proyectos (juegos).

Los proyectos de Unity suelen ocupar mucho espacio, incluso si solo has añadido un elemento simple como un cubo 3D, debido a varios factores relacionados con la estructura y los archivos del motor:

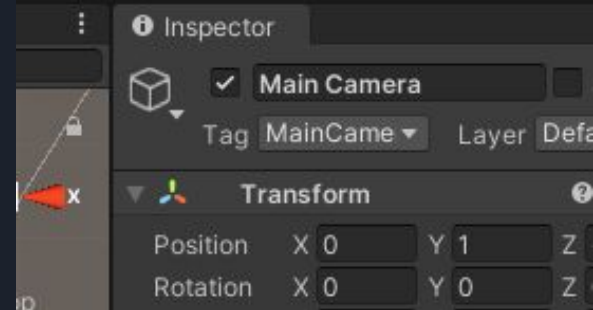
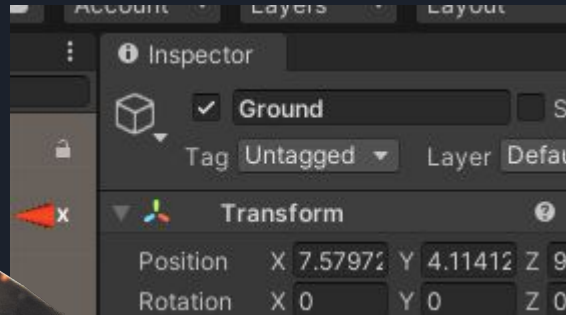
Unity genera una serie de carpetas y archivos necesarios para el funcionamiento del proyecto, independientemente de su complejidad:

- Biblioteca (Library): Contiene datos generados automáticamente, como precompilaciones de scripts, cachés de importación de recursos, y configuraciones específicas del proyecto. Esta carpeta puede volverse muy grande.
- Temp y Obj: Estas carpetas se utilizan para almacenar datos temporales y de compilación.
- Packages y ProjectSettings: Incluyen configuraciones y dependencias del proyecto.

# Etiquetas

Un **Tag** es una palabra de referencia que puede asignar a uno o más GameObjects. Por ejemplo, puede definir Tags de “Jugador” para los personajes controlados por el jugador y una Tag de “Enemigo” para los personajes no controlados por el jugador.

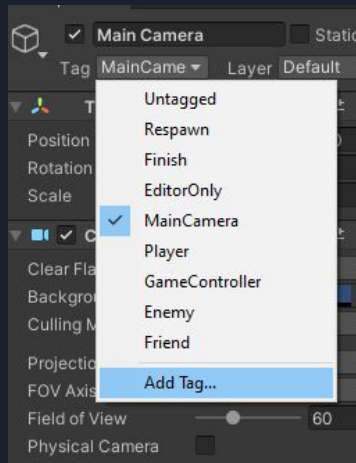
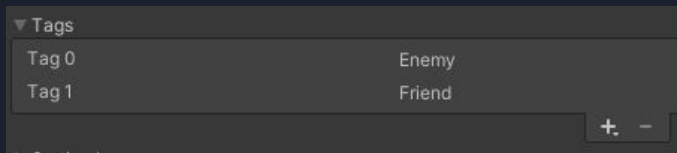
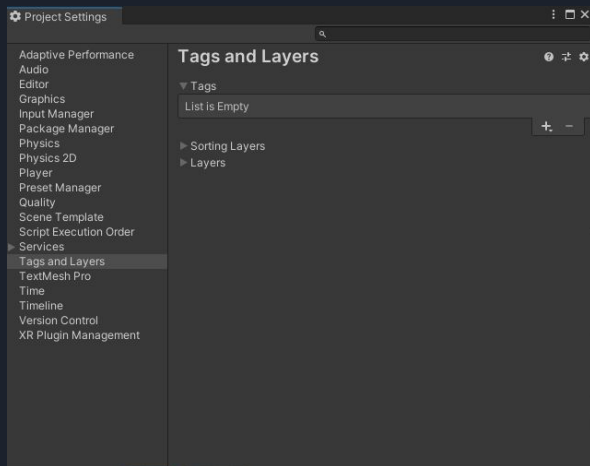
<https://docs.unity3d.com/es/2018.4/Manual/Tags.html#:~:text=Un%20Tag%20es%20una%20palabra.no%20controlados%20por%20el%20jugador.>



Podemos distinguir a los objetos en el código gracias a su etiqueta. También es necesario para instanciarlos.

# ¿Cómo crear tags?

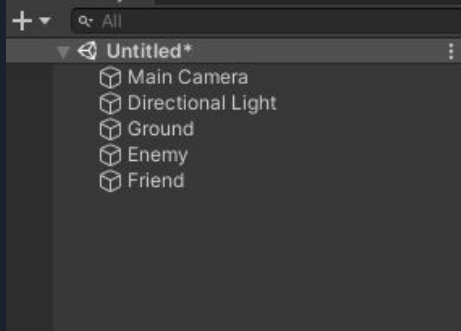
Para crear tags nos iremos a Edit>Project Settings > Tags and Layers. Crearemos dos: Enemy y Friend



También pueden  
añadirse desde aquí

# Aplicar etiquetas

Crearemos una escena como la que tenemos a continuación. Donde tendremos un cubo “aplastado” que hará de suelo, una esfera y un cubo. Estos dos últimos deben estar flotando arriba del suelo.



Les aplicaremos texturas. Las dimensiones (escala) del suelo pueden ser (10, 0.1, 10).

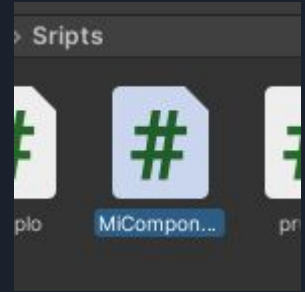
Los nombres de los objetos serán los que podemos ver a la izquierda. Igualmente, etiquetaremos (tag) a la esfera con la etiqueta Friend y al cubo con la etiqueta Enemy.

Le aplicaremos gravedad (Rigidbody) a estos dos objetos.



# Aplicar etiquetas

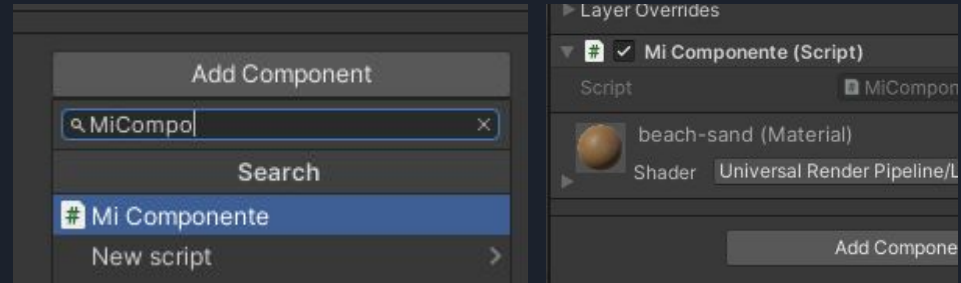
Antes de seguir, crearemos un script que nos será necesario para nuestro juego.



Añadiremos un nuevo Script llamado MiComponente.cs. Al abrirlo le añadiremos este método al código:

```
private void OnCollisionEnter(Collision collision){  
    if(collision.gameObject.CompareTag("Enemy")){  
        Destroy(collision.gameObject);  
        Debug.Log("Destruyendo enemigo");  
    }  
    if(collision.gameObject.CompareTag("Friend")){  
        Debug.Log("Hola, amigo!");  
    }  
}
```

Tras guardar, le aplicamos el componente al suelo.



# Aplicar etiquetas

Comprobamos que el enemigo desaparece y el amigo queda.

