



# UD2.1 Introducción a Unity

**2º CFGS**  
**Desarrollo de Aplicaciones Multiplataforma**  
**2024-25**

# 1.- Introducción

## ¿Qué es Unity?

Unity es un **motor de juegos** multiplataforma.

Su origen data del año 2005.

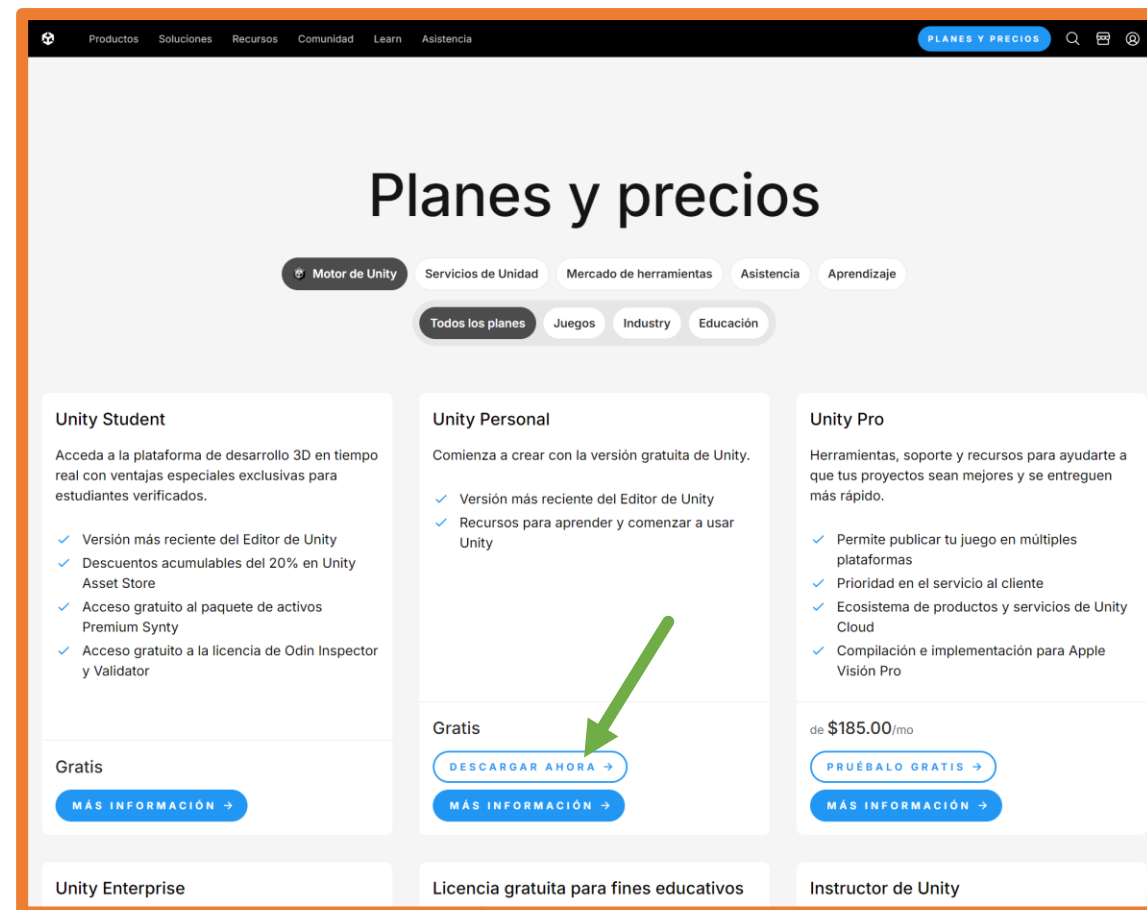
Permite el **desarrollo de juegos 2D y 3D**.

Actualmente Unity soporta el desarrollo para unas **20 plataformas** de las cuales las más populares son **PC, iOS y Android**.

Unity ofrece un **conjunto de herramientas** tan completo que minimiza el uso de herramientas externas para realizar los proyectos.

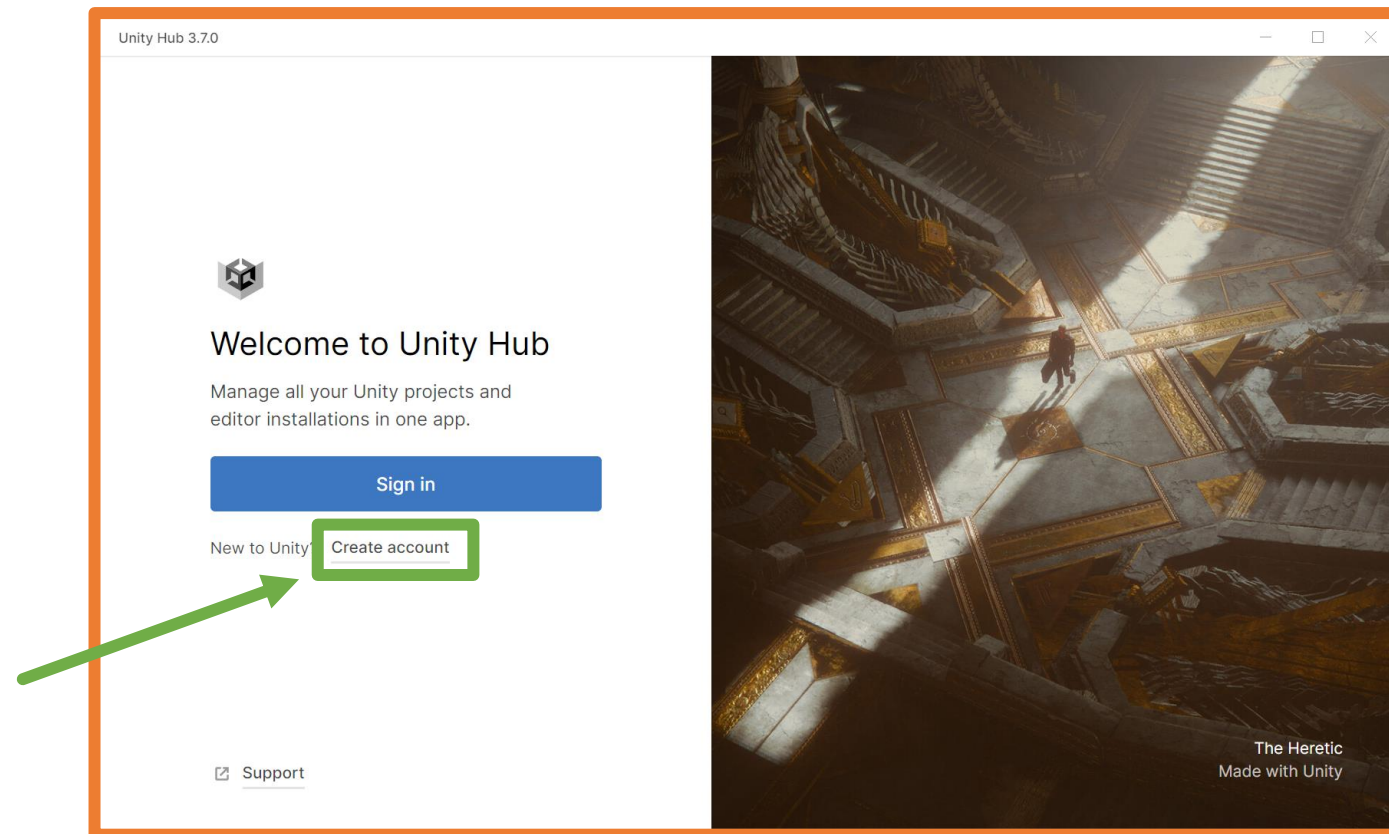
## 2.- Unity HUB

El programa Unity Hub permite administrar las versiones del editor Unity que se deseen utilizar.



## 2.- Unity HUB

Para poder utilizar Unity Hub se necesita disponer de una **Unity ID**.



## 2.- Unity HUB

### Unity ID

También se puede crear la Unity ID desde la web de Unity:

<https://id.unity.com/es/>



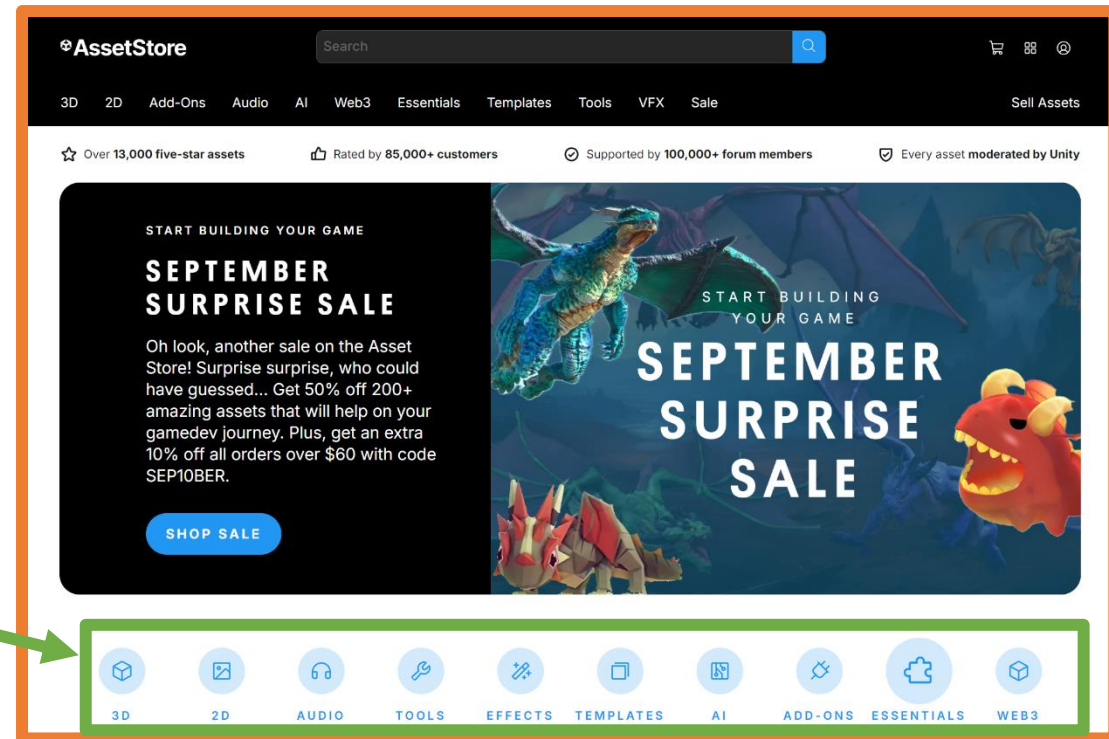
The image shows a screenshot of the Unity ID login page. The page has a light blue header with the Unity logo and 'Unity ID' text. Below the header, the main heading is 'Iniciar sesión en tu Unity ID'. Underneath, there is a line of text: 'Si no tienes una Unity ID, [crea una.](#)'. A blue arrow points from the right side of the slide to this link. Below this text are two input fields: 'Correo electrónico' and 'Contraseña'. At the bottom, there is a checkbox labeled 'Recuérdame' which is checked, and a green button labeled 'Iniciar sesión'. To the right of the button, there are links for '¿Olvidaste tu contraseña?' and 'Ayuda'.

## 2.- Unity HUB

### Unity ID

Al disponer de una Unity ID además de poder utilizar el motor de videojuegos (licencia de uso) también se puede usar la **Unity Asset Store** desde la cual se podrá descargar assets tanto gratuitos como de pago.

<https://assetstore.unity.com/>



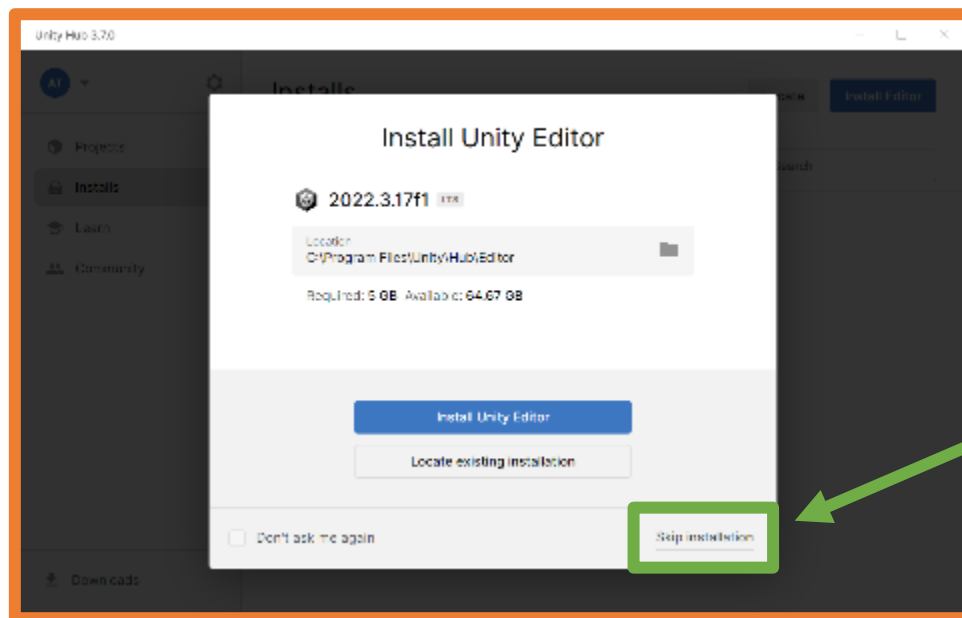
### 3.- Instalación del editor Unity

Para desarrollar videojuegos se necesita el **editor Unity**.

La primera vez que se accede a Unity Hub con la Unity ID se indica que se puede instalar un editor Unity.

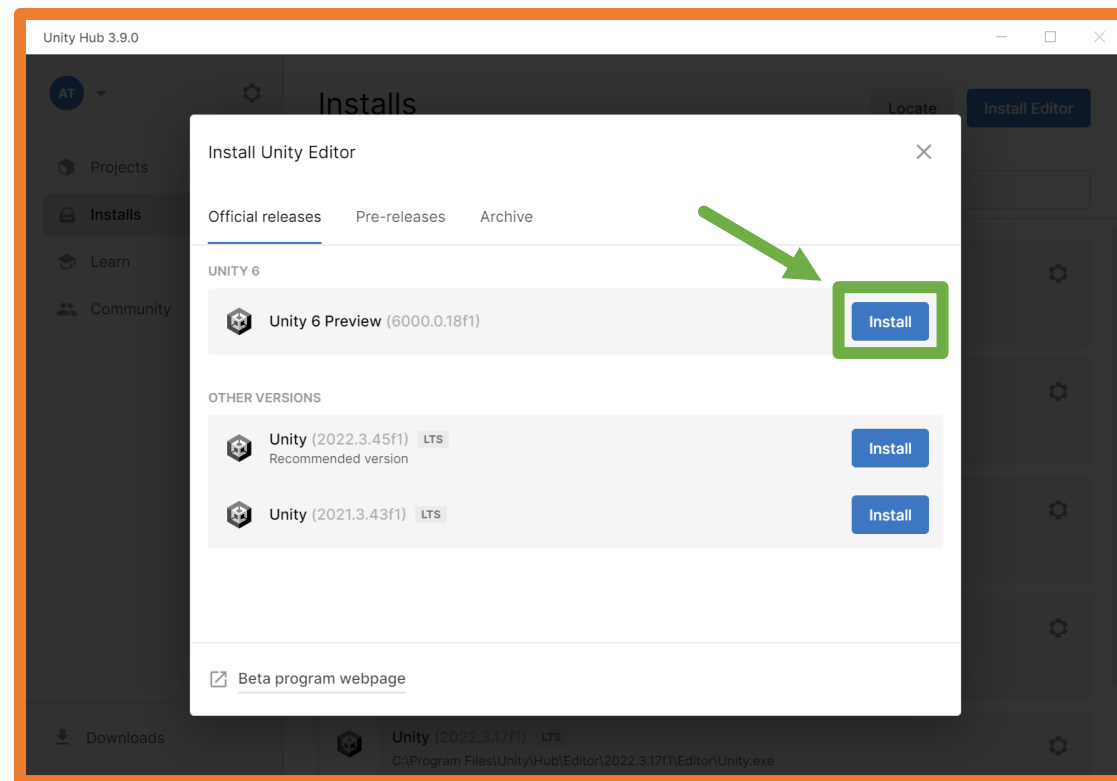
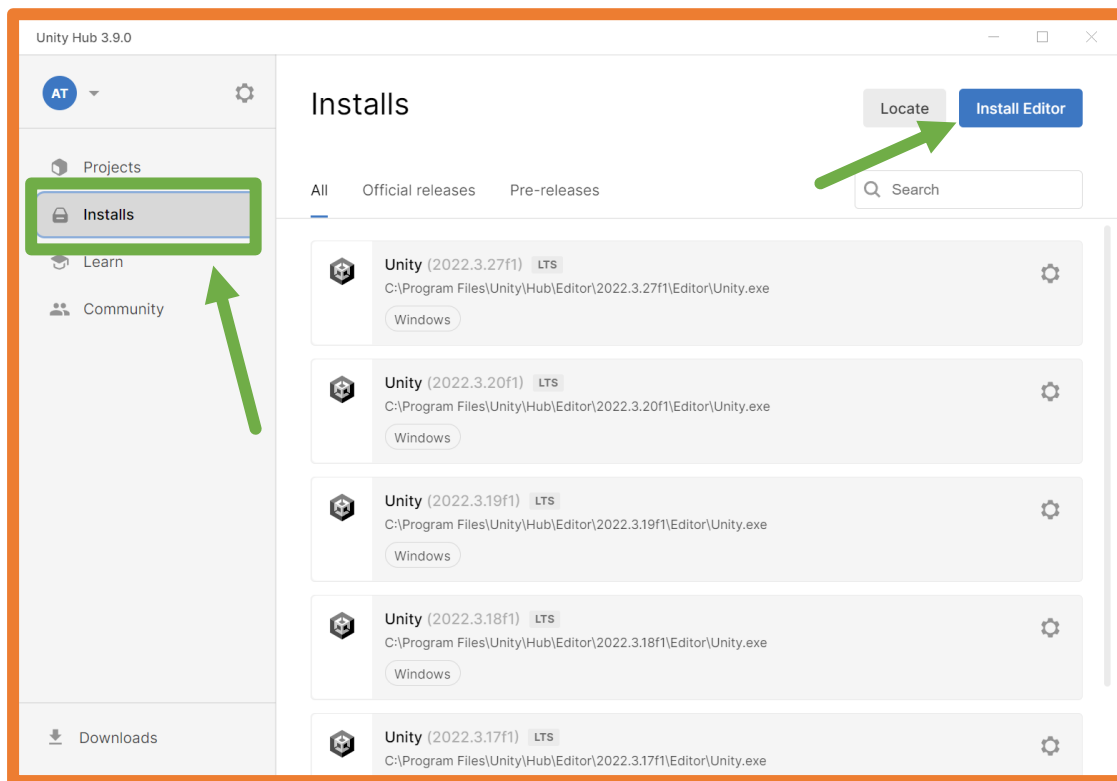
Por defecto, el editor Unity que se instalará será la última versión estable **LTS** (Long Term Support).

No se recomienda esta instalación por defecto ya que no permite configurar la instalación.



## 3.- Instalación del editor Unity

Desde el apartado **Installs** de **Unity Hub** se pueden instalar las diferentes versiones del editor Unity disponibles:

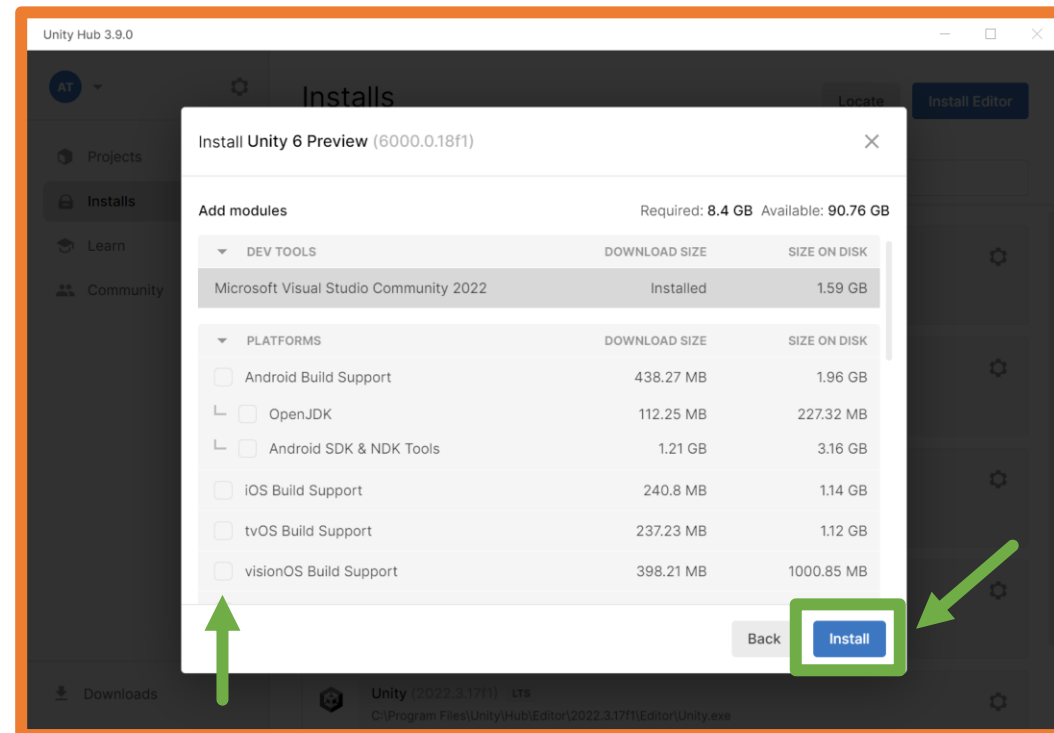




### 3.- Instalación del editor Unity

Se debe elegir qué configuración se quiere para el editor Unity que permitirá crear los ejecutables del juego para cada plataforma, añadir idiomas asiáticos y la documentación de Unity.

Para clase será suficiente con marcar **Linux Build Support (IL2CPP)**.



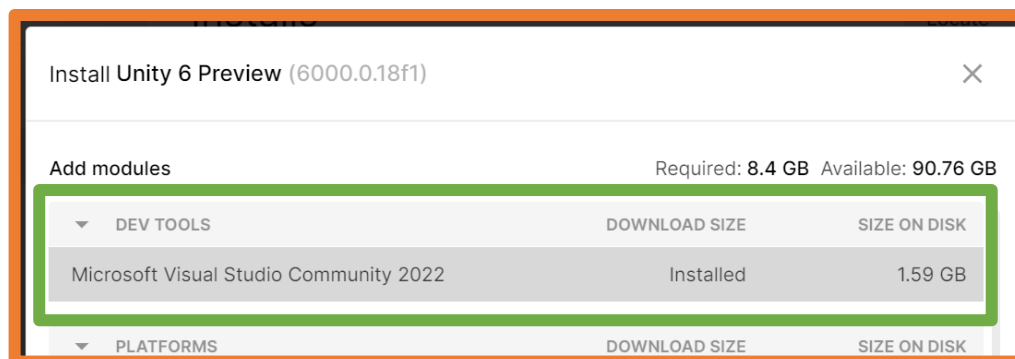
### 3.- Instalación del editor Unity

#### Editor de código

Unity necesita un editor de código para la programación, en Windows durante la instalación del editor Unity se incluye **Microsoft Visual Studio Community**. En clase detectará **Visual Studio Code** y lo configurará como editor por defecto.

Si no se tiene instalado ese editor y no se quiere instalar, se debe desmarcar.

En el caso de usar **Microsoft Visual Studio Community** o **Visual Studio Code** como editor por defecto se pueden instalar las extensiones para el lenguaje de programación **C#** que es el usado con Unity.

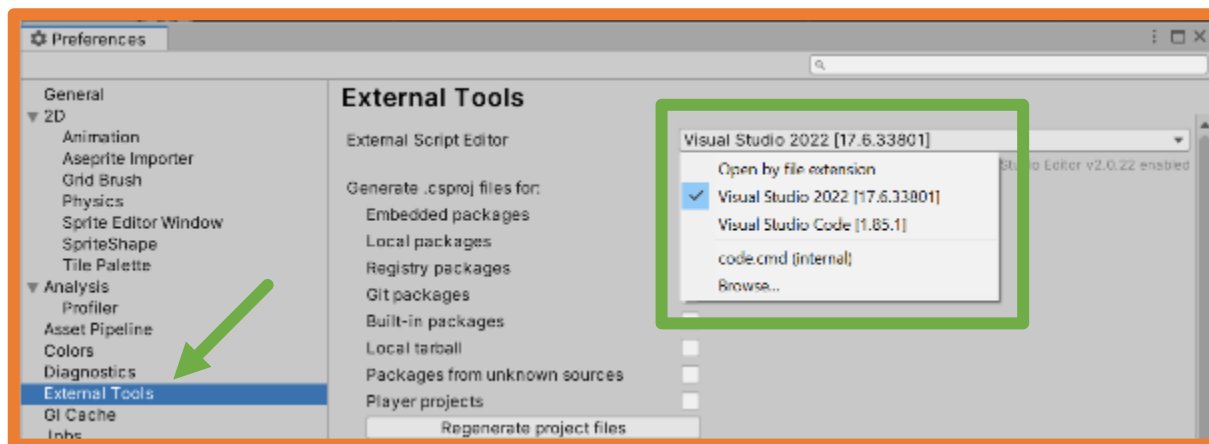
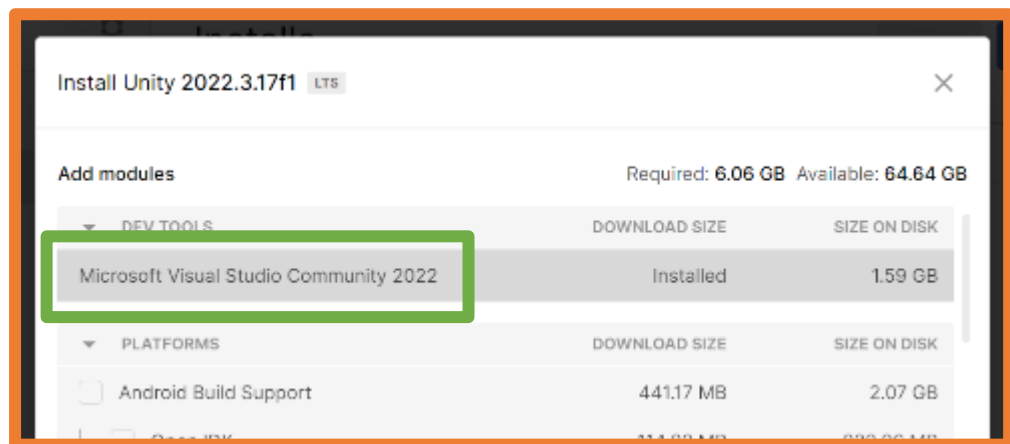


### 3.- Instalación del editor Unity

#### Editor de código

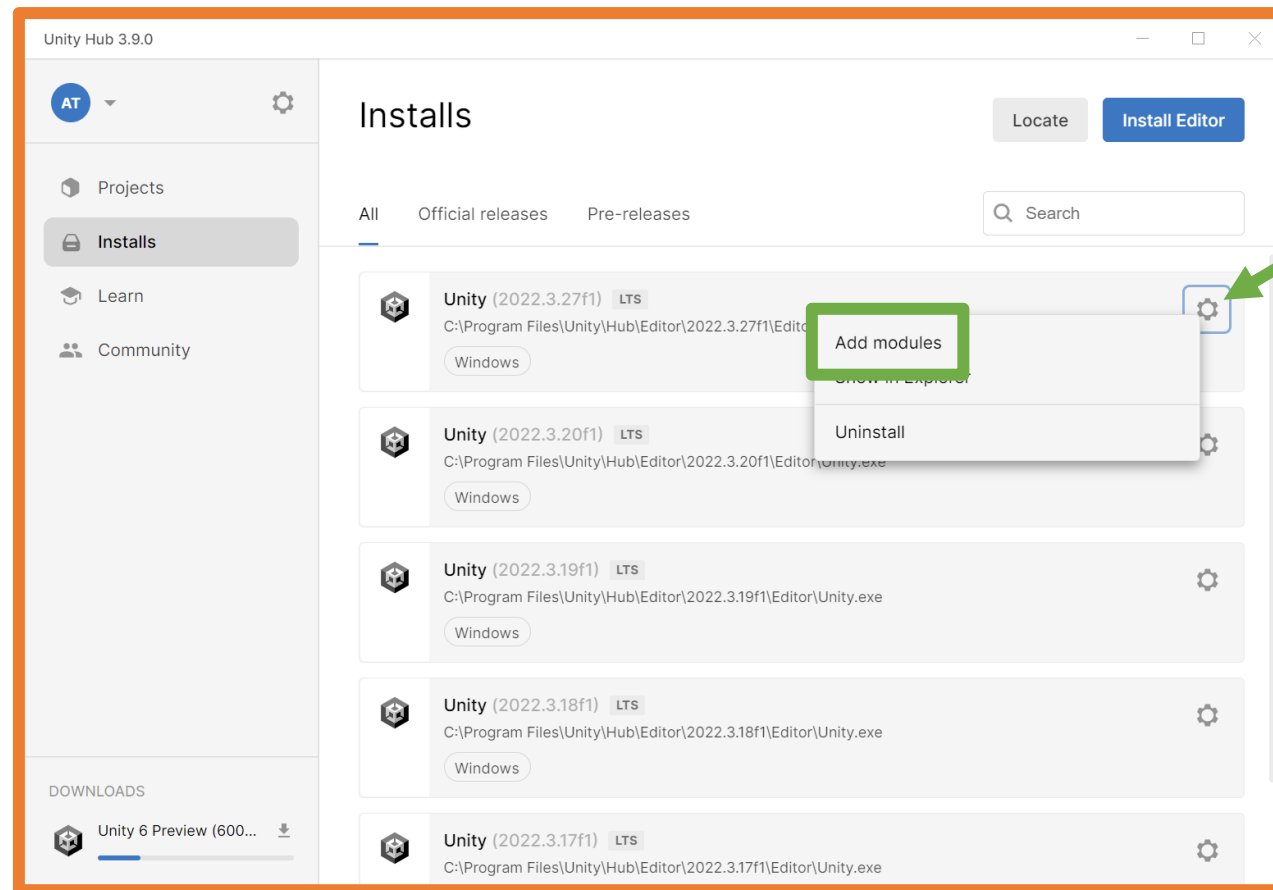
Si no se tiene instalado **Microsoft Visual Studio Community** y no se quiere instalar, se debe desmarcar.

Se puede **utilizar cualquier otro editor de código** cambiando la configuración del editor Unity una vez esté instalado. En **Menú → Edit → Preferences**.



### 3.- Instalación del editor Unity

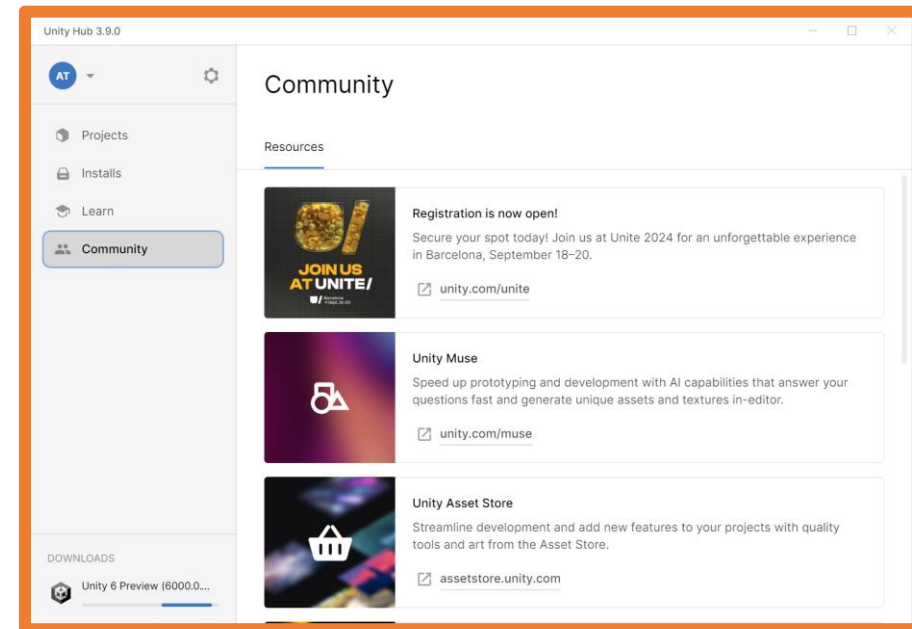
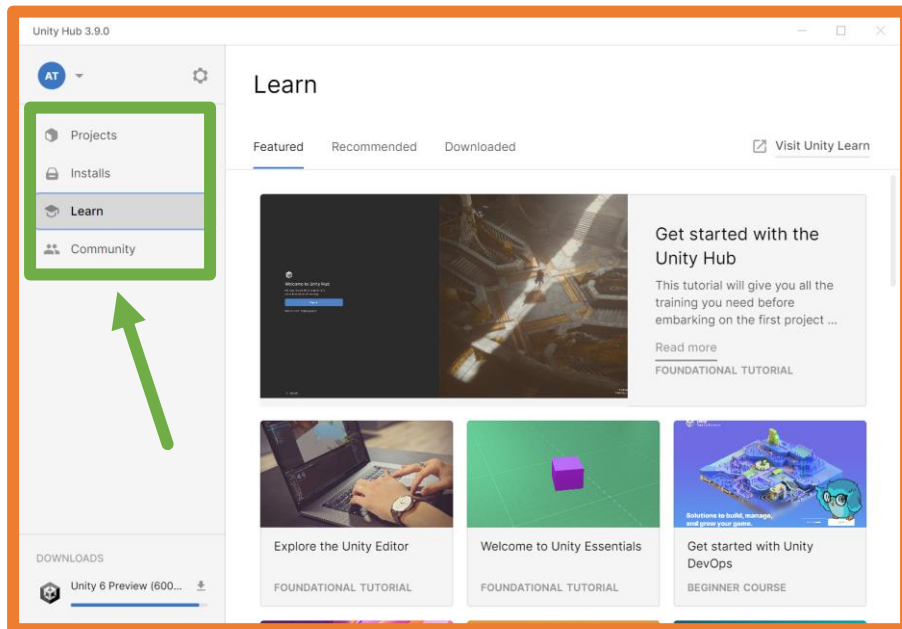
Si más adelante en el desarrollo se necesita, se pueden añadir/eliminar módulos al editor.



## 4.- Crear proyecto Unity

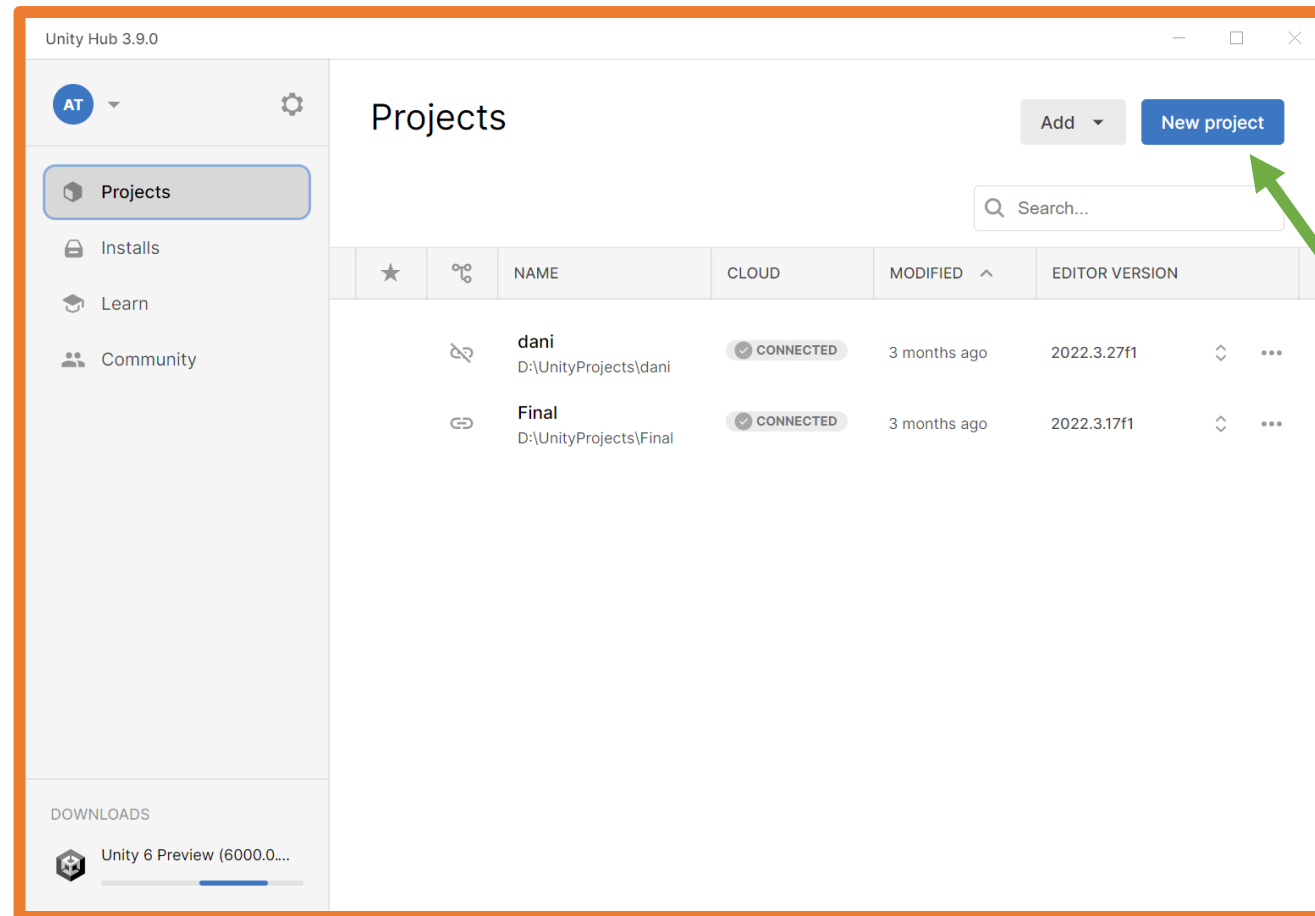
Unity HUB será el punto de inicio de cualquier proyecto Unity, en su ventana se pueden ver las secciones:

- **Projects:** gestor de proyectos.
- **Installs:** gestor de editores Unity instalados.
- **Learn:** apartado de autoaprendizaje
- **Community:** acceso a la comunidad con secciones como Assets Store o Foros.



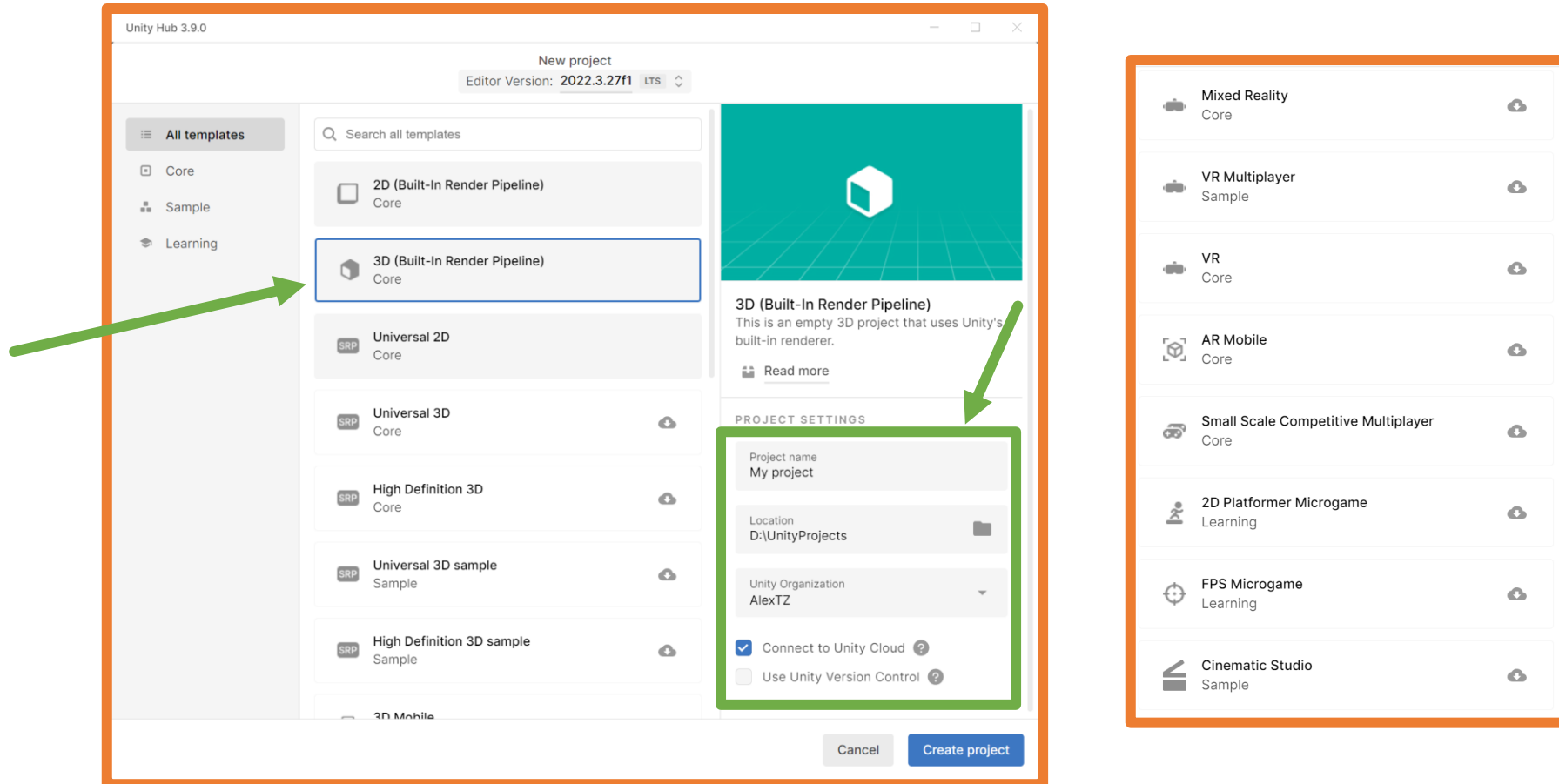
## 4.- Crear proyecto Unity

Desde **Unity Hub**, en el apartado **Projects** se pueden crear proyectos pulsando el botón **New project**.



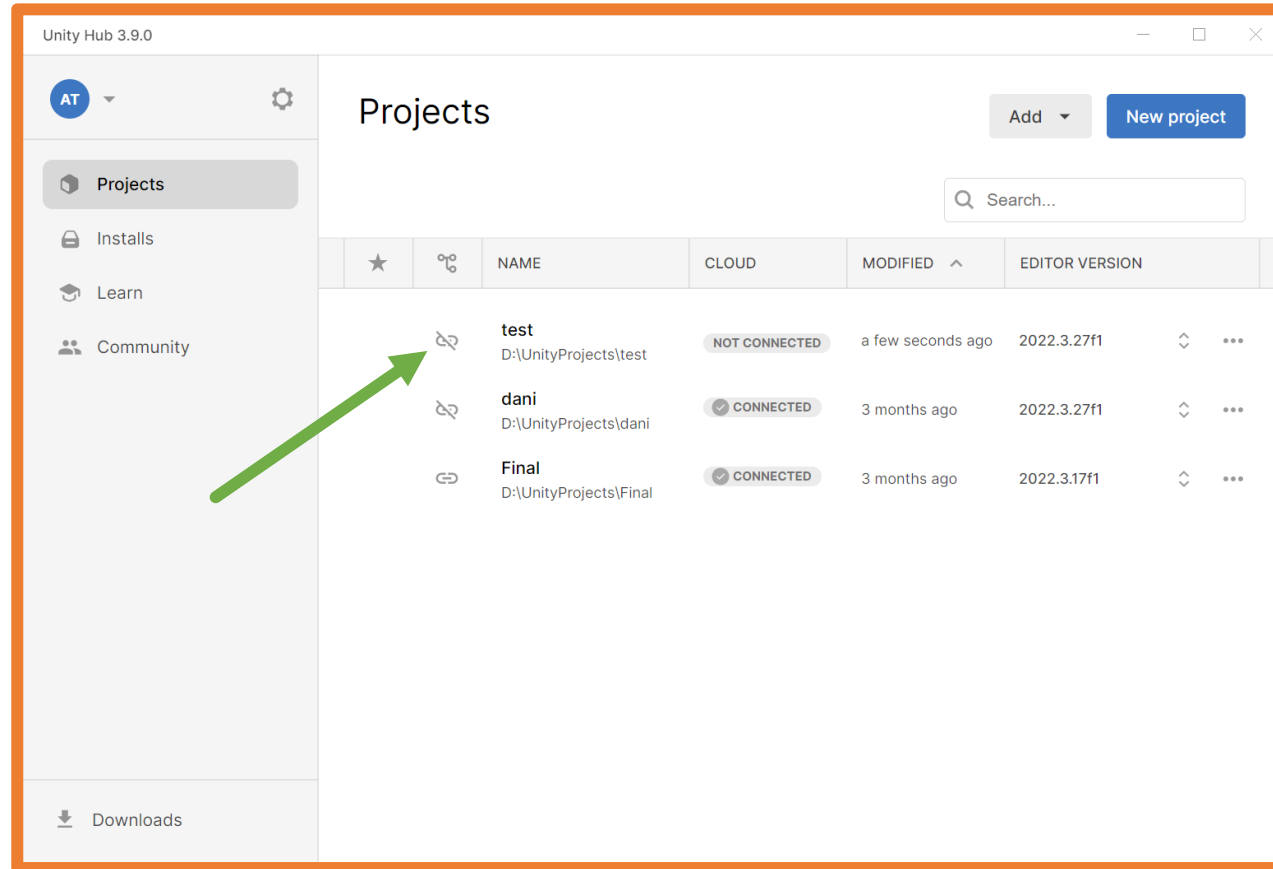
## 4.- Crear proyecto Unity

Al crear un proyecto se debe **elegir el tipo de proyecto** para que se carguen las librerías necesarias.



## 4.- Crear proyecto Unity

Una vez creado el proyecto, este estará disponible en Unity Hub.



Durante la creación **se abrirá el editor Unity** para ese proyecto.



# Práctica

## Actividad 1

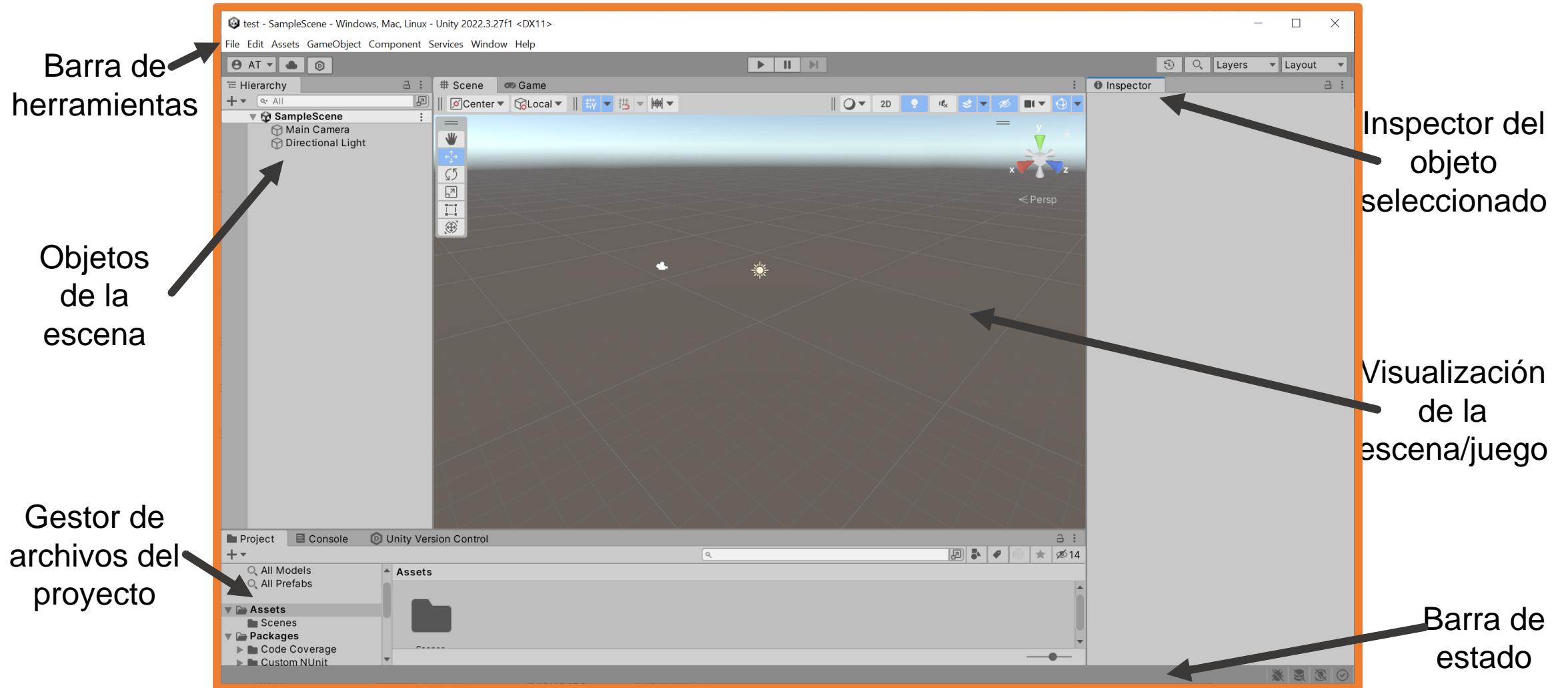
Instalación de un editor Unity.

## Actividad 2

Crea el proyecto **PruebaTuNombre**.

## 5.- Editor Unity

Por defecto la ventana del **editor Unity** tiene el siguiente aspecto:

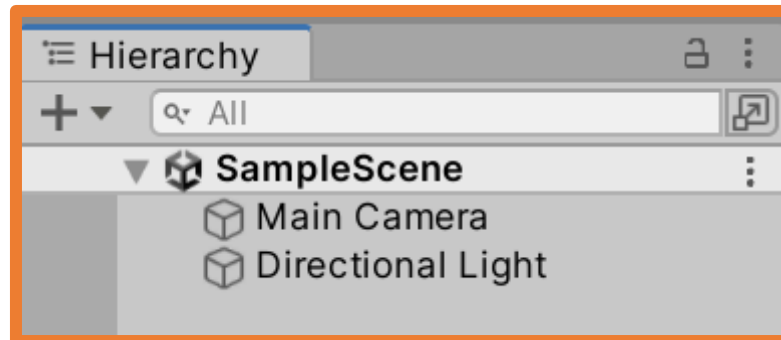


## 5.- Editor Unity

Unity hace uso de la **Programación Orientada a Objetos**, por esta razón todos los elementos del juego se incluyen en la jerarquía de objetos.

El **juego** ocupa la posición más alta de la jerarquía y dentro de él se ubican las **escenas** que podrían considerarse las pantallas del juego.

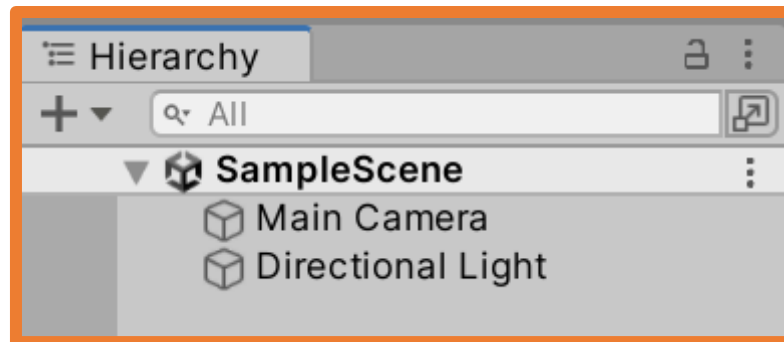
La **jerarquía** muestra todos los elementos que tiene una escena.



## 5.- Editor Unity

Dentro de cada escena se deben añadir todos los elementos que se necesiten, estos elementos son los **GameObject**, las escenas también siguen una estructura jerárquica pudiendo agruparse los **GameObjects** unos dentro de otros.

Por defecto un proyecto Unity contiene una **Escena** que a su vez contiene un **GameObject** de tipo **cámara** llamado **Main Camera** y un **GameObject** de tipo **luz** llamado **Directional Light**.

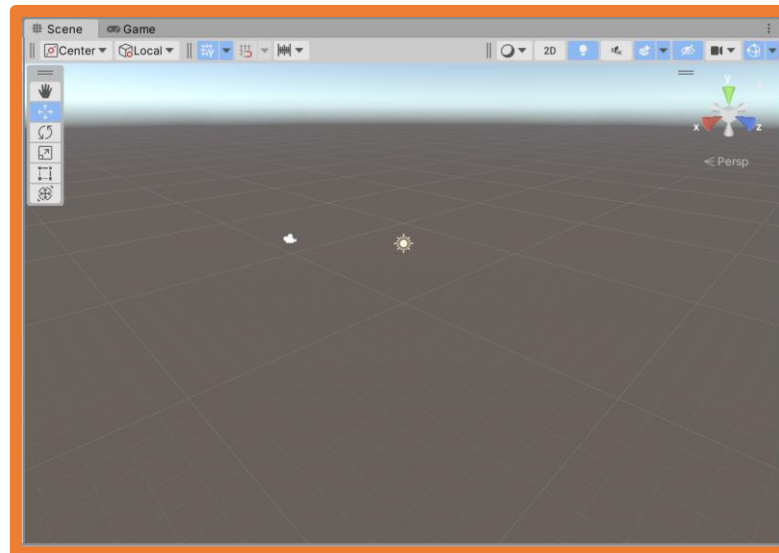


## 5.- Editor Unity

En la parte central del editor se pueden observar dos pestañas:

**Scene:** una **visualización de la escena** con una cámara virtual independiente a los elementos del juego, permite colocar todos los GameObjects de la escena en su lugar.

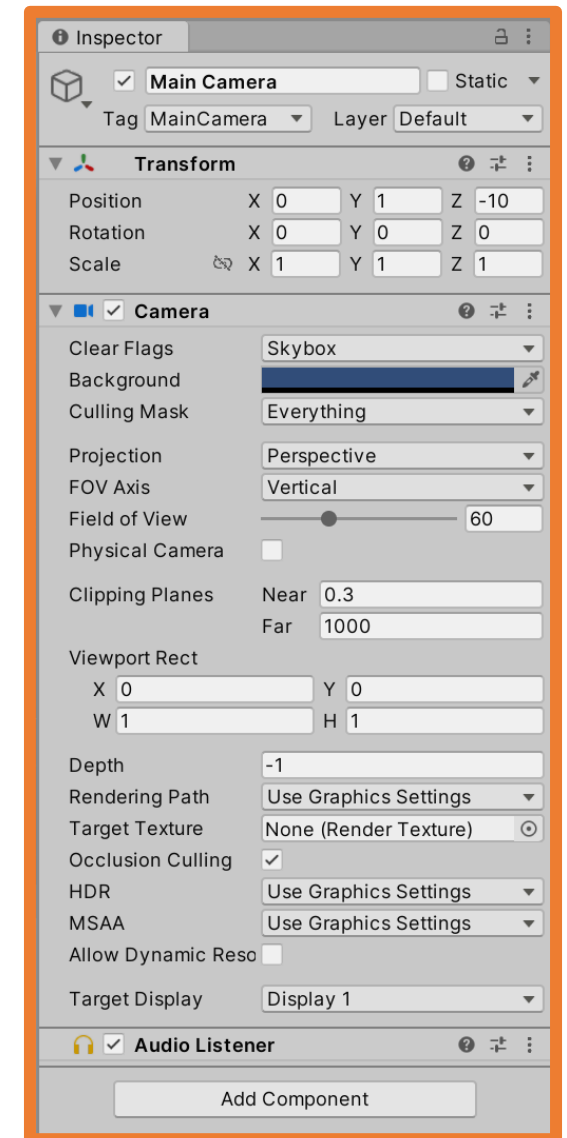
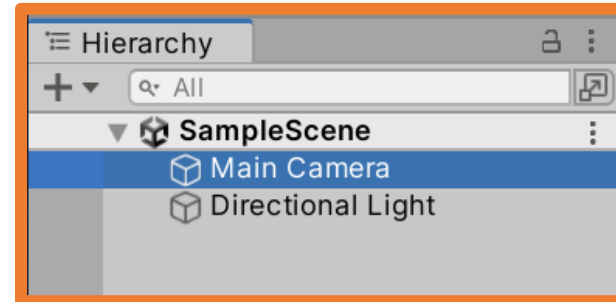
**Game:** muestra **la visualización desde la cámara configurada en el juego**, permite ver lo que se mostrará con la ejecución del juego.



## 5.- Editor Unity

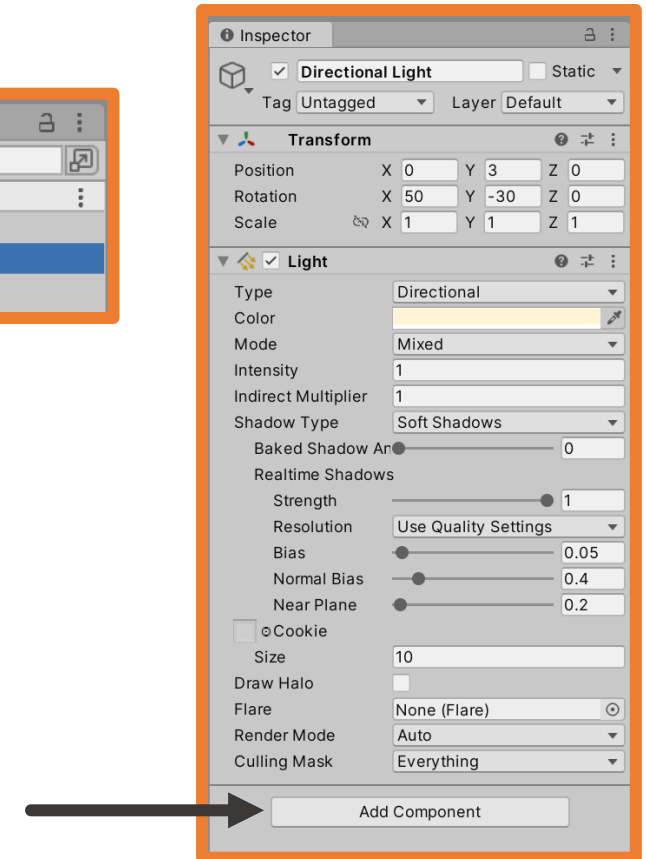
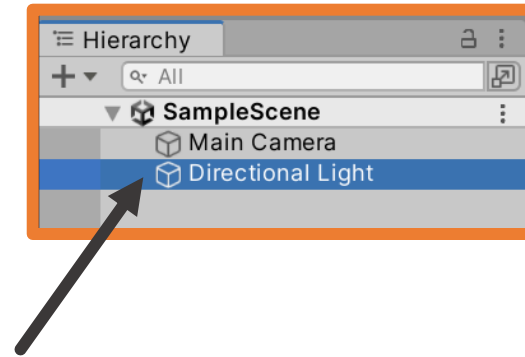
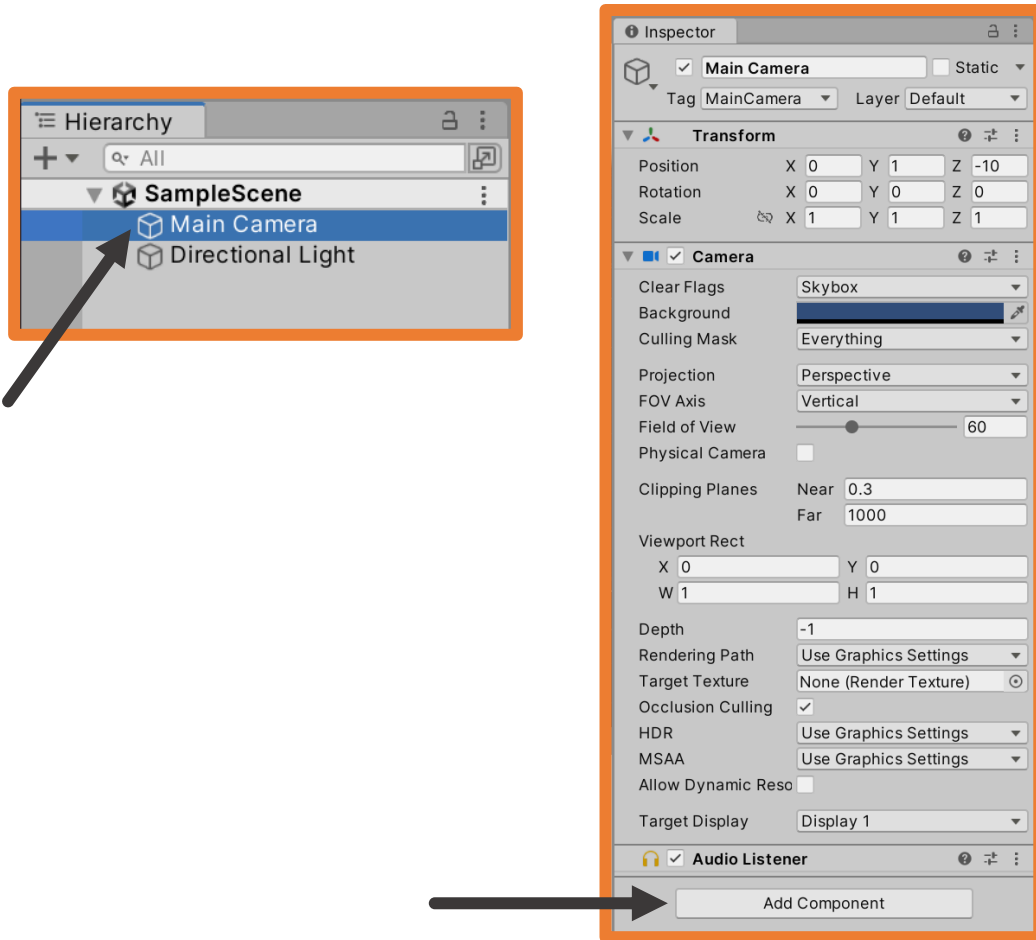
Cuando se selecciona un objeto de la escena, en el **Inspector** se pueden observar todas sus características.

Desde el código se podrá acceder a los componentes de cada objeto.



## 5.- Editor Unity

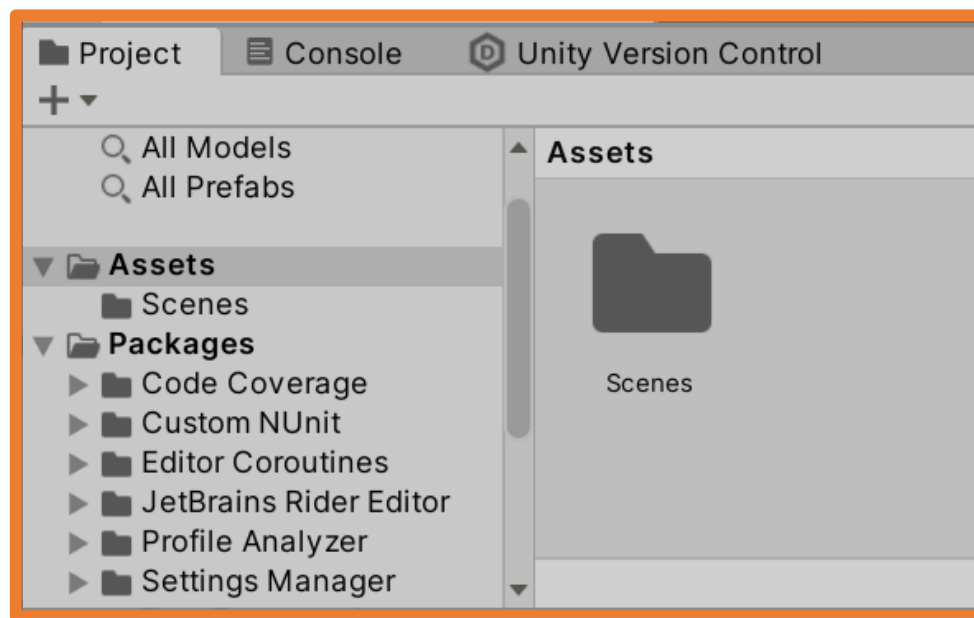
Cada tipo de objeto tendrá unos componentes por defecto, y además, se les puede añadir componentes si se necesita.



## 5.- Editor Unity

En el **Gestor de archivos** permite visualizar y realizar todas las acciones que se requieran sobre los archivos del proyecto.

Se recomienda que todas las carpetas y archivos que se quieran crear se creen desde el gestor de archivos del editor.





## 5.- Editor Unity

En la documentación oficial se puede encontrar toda la información de la interfaz de Unity: <https://docs.unity3d.com/2023.3/Documentation/Manual/UsingTheEditor.html>

Unity Documentation

Manual Scripting API

Search manual...

Version: Unity 6 Preview (6000.0) Language: English

Unity Manual

- Unity 6 Preview User Manual
- What's new in Unity
- Install and upgrade
- Packages and feature sets
- Get started with Unity
- The Unity Editor
  - Unity's interface**
    - The Project window
    - The Scene view
    - The Game view
    - Device Simulator
    - The Hierarchy window
    - The Inspector window
    - Editing properties
    - The Toolbar
    - The status bar
    - The Background Tasks window
    - Console Window
    - Additional windows
    - Undo
    - Search in the Editor
    - Customizing your workspace


The Unity Editor / Unity's interface

←

→

### Unity's interface

This section provides a detailed tour of the most common editor windows, and how to make full use of them.



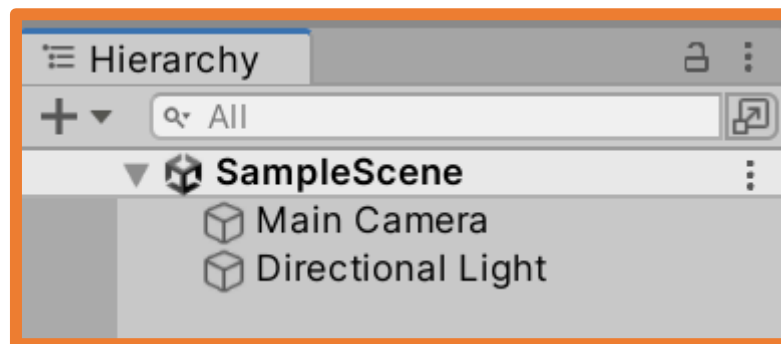
The screenshot shows the Unity Editor interface with several windows labeled with letters: A points to the Scene View, B to the Hierarchy window, C to the Project window, D to the Inspector window, E to the Toolbar, F to the Console window, G to the Background Tasks window, and H to the Additional Windows section. The interface includes a top navigation bar, a left sidebar with the Unity Manual, and a main content area with the title 'Unity's interface' and a description.

The most common windows in their default positions

## 5.- Editor Unity

Cuando se crea un proyecto, Unity crea una **Escena** y le añade dos **GameObjects**:

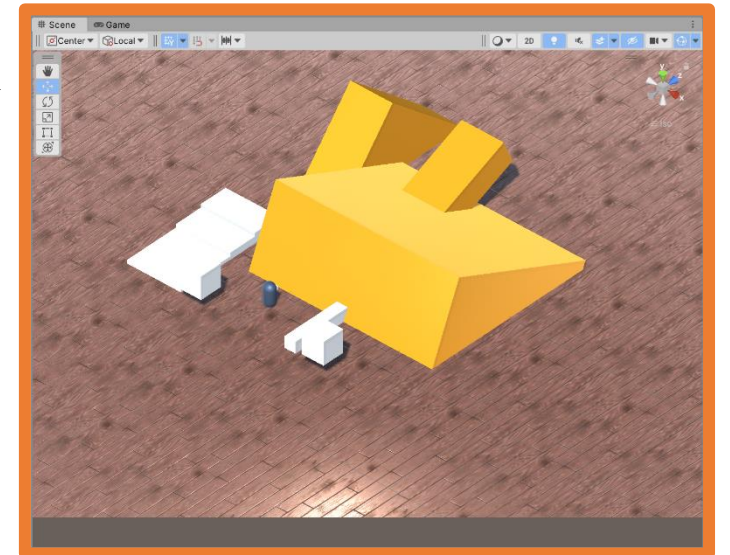
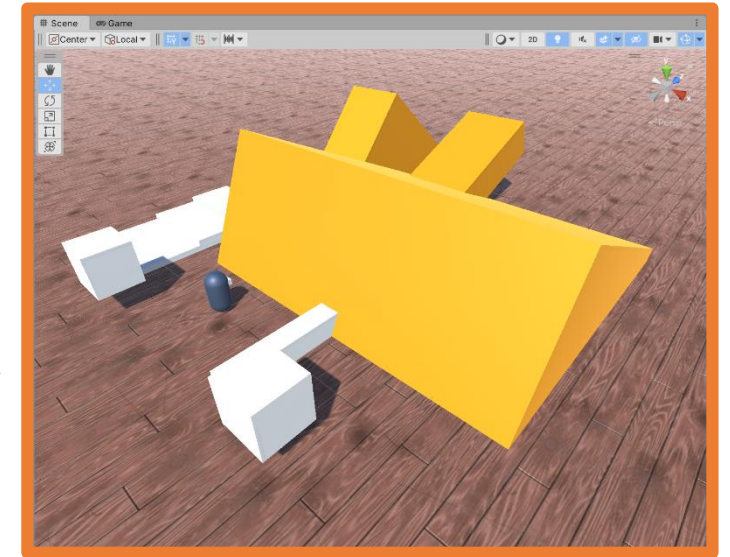
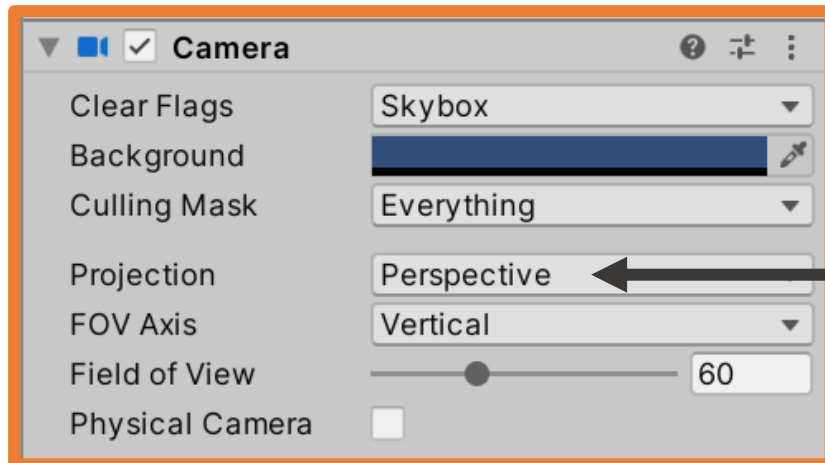
- **Main Camera:** es una cámara virtual que enfoca a lo que se mostrará cuando se ejecute el juego.
- **Directional Light:** luz que ilumina toda la escena con rayos de luz paralelos que simulan la luz del sol.



## 6.- Cámaras

En Unity las cámaras pueden ser de dos tipos dependiendo de cómo se rendericen los :

- **Perspectiva:** los elementos más alejados de la cámara se visualizarán más pequeños y con perspectiva (juegos 3D).
- **Ortográfica:** es una cámara isométrica donde no existe profundidad (juegos con visualización 2D).

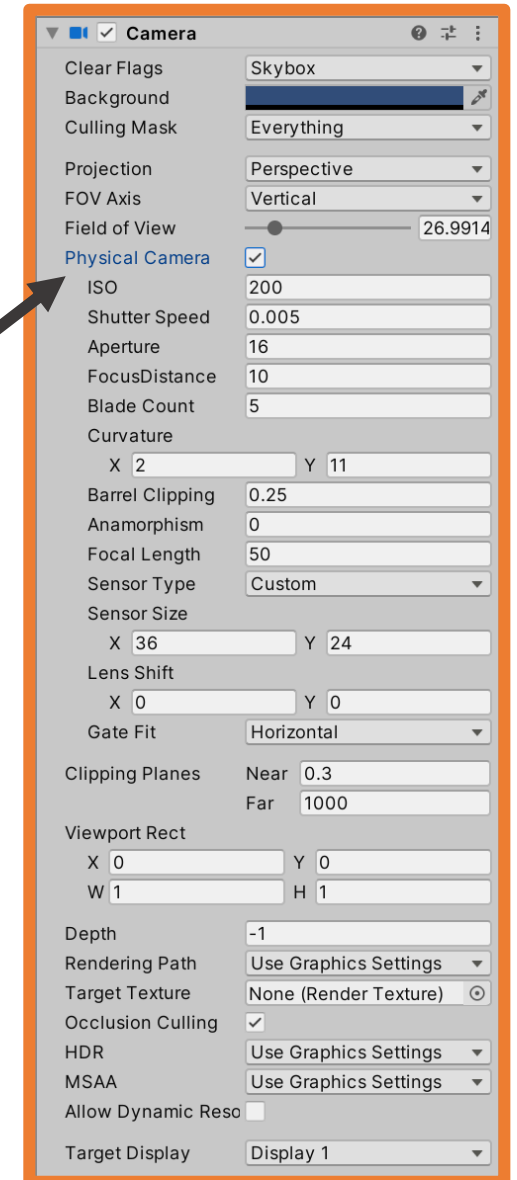


## 6.- Cámaras

Una cámara se puede marcar con la opción **Physical Camera** lo que habilitará una serie de opciones que permitirán realizar una configuración realista de la cámara:

- ISO
- Velocidad de obturación
- Apertura de diafragma
- ...

Con estos ajustes se pueden conseguir resultados muy realistas pero para configurarlos correctamente se necesitan conocimientos avanzados de fotografía.

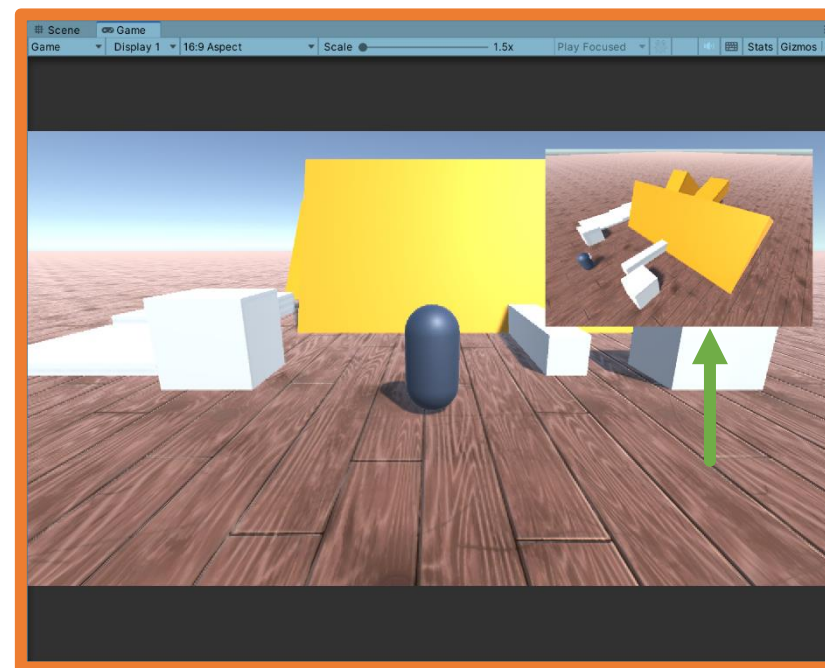
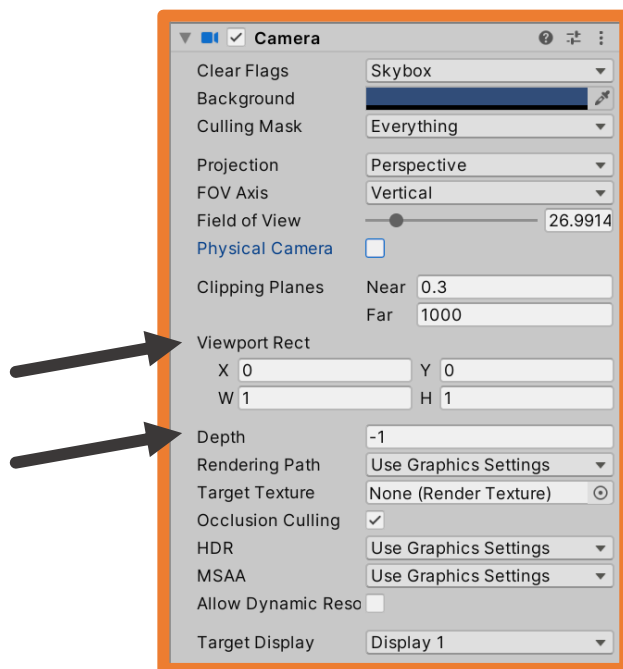


## 6.- Cámaras

En una escena puede haber tantas cámaras como se necesite.

Cada cámara puede llevar asociado un **Display** (monitor) donde se visualizará.

Si se configuran **varias cámaras en el mismo Display**, con las opciones de **View Rect** (recorte) y **Depth** (profundidad, a mayor valor la cámara estará por encima de las otras en el renderizado) se pueden visualizar las dos cámaras a la vez en el juego (por ejemplo para un minimapa o un retrovisor)



## 6.- Cámaras

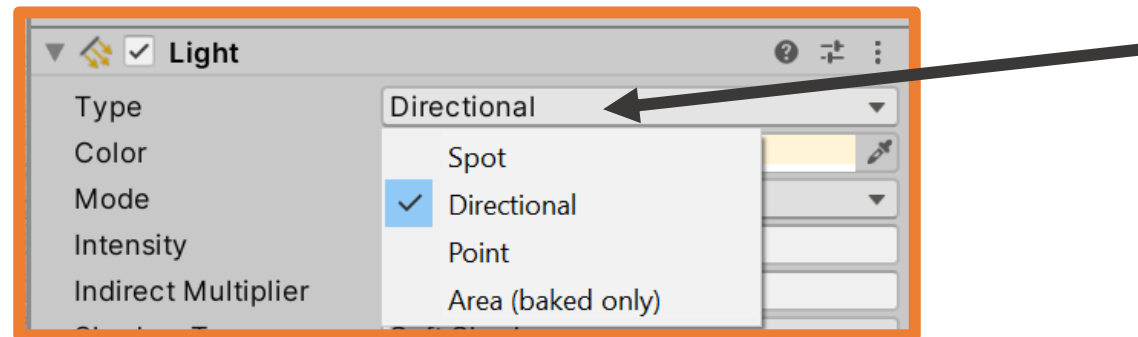
Tener varias cámaras en la escena también permite realizar acciones como:

- Cambiar la cámara cuando el jugador llega a un punto.
- Crear espejos.
- Mostrar en un monitor de seguridad lo que ocurre en otro lugar de la escena.
- ...

## 7.- Luces

En Unity hay disponibles los siguientes tipos de luces:

- **Spot:** emite la luz en forma de cono. Se pueden usar para linternas o farolas.
- **Directional:** emite rayos de luz paralelos simulando la luz del sol. Se usan en exteriores.
- **Point:** emite luz en todas las direcciones. Se pueden usar para fuegos, velas...
- **Area:** emiten luz como si se tratara de un panel luminoso. Son más difíciles de configurar y se pueden usar para la iluminación de interiores como los paneles del techo de este aula.



Se pueden añadir tantas luces como se quiera a la escena.

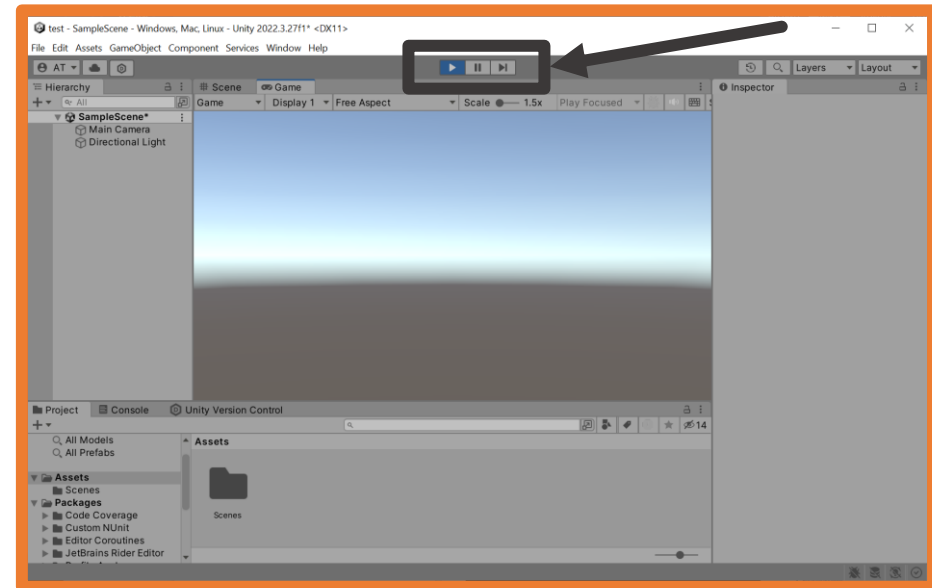
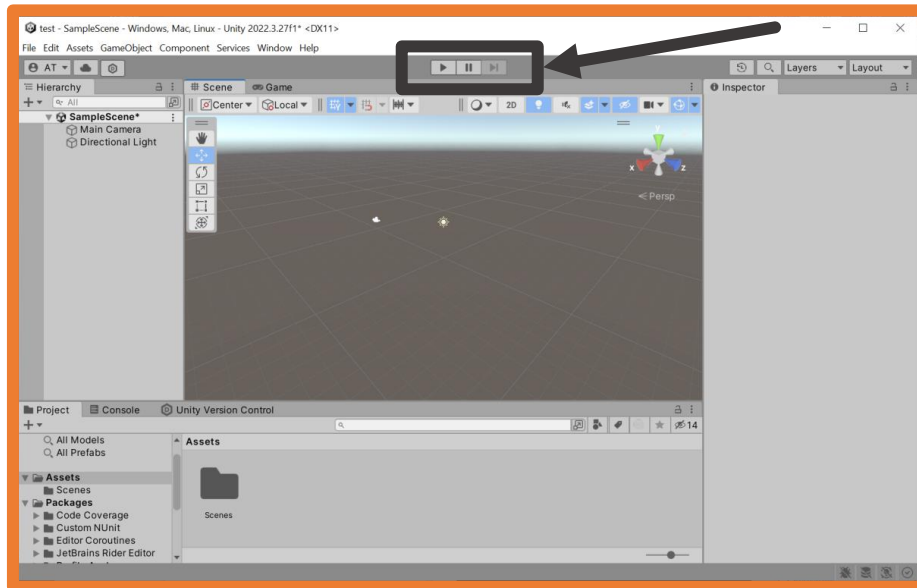


## 8.- Ejecución del juego

Cuando se **ejecuta el juego** (botón "Play" ▶ de la barra superior) la sección de visualización cambia a la vista de la cámara y toda la interfaz se oscurece.

Al ejecutar el juego se puede trabajar con el proyecto para poder ver cómo reacciona según diferentes valores, pero **los cambios solo se mantienen durante la ejecución del juego**.

Al no haber casi diferencia de estilo entre los dos estados a veces se nos olvida que se está ejecutando el juego y se pierden los cambios.

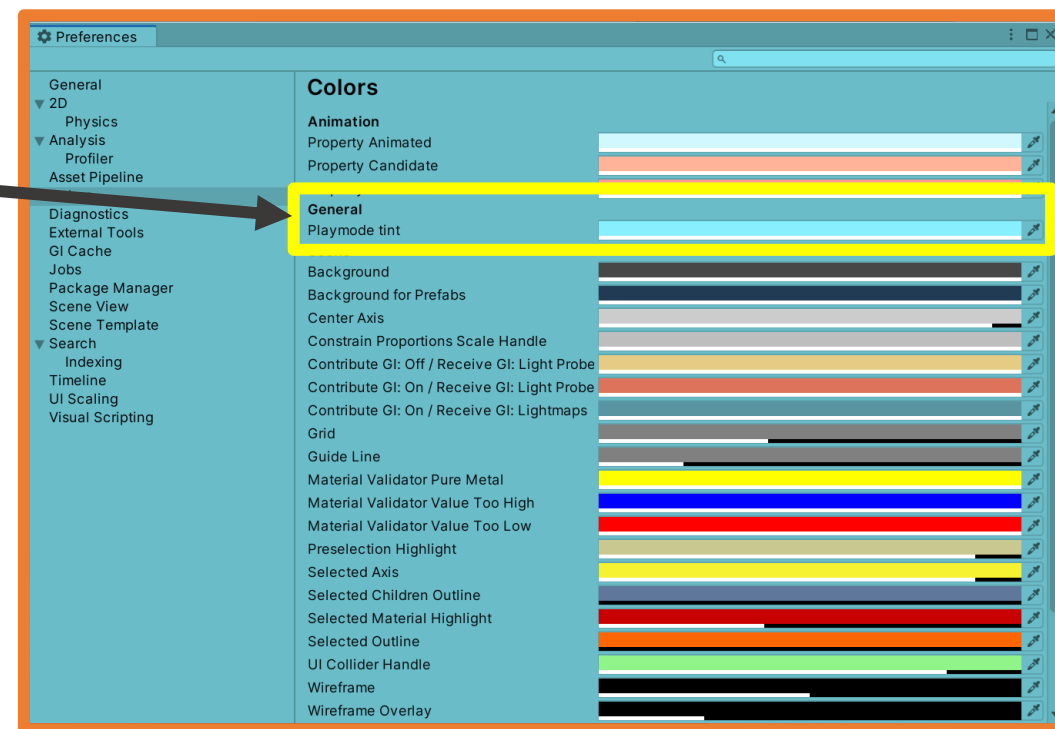
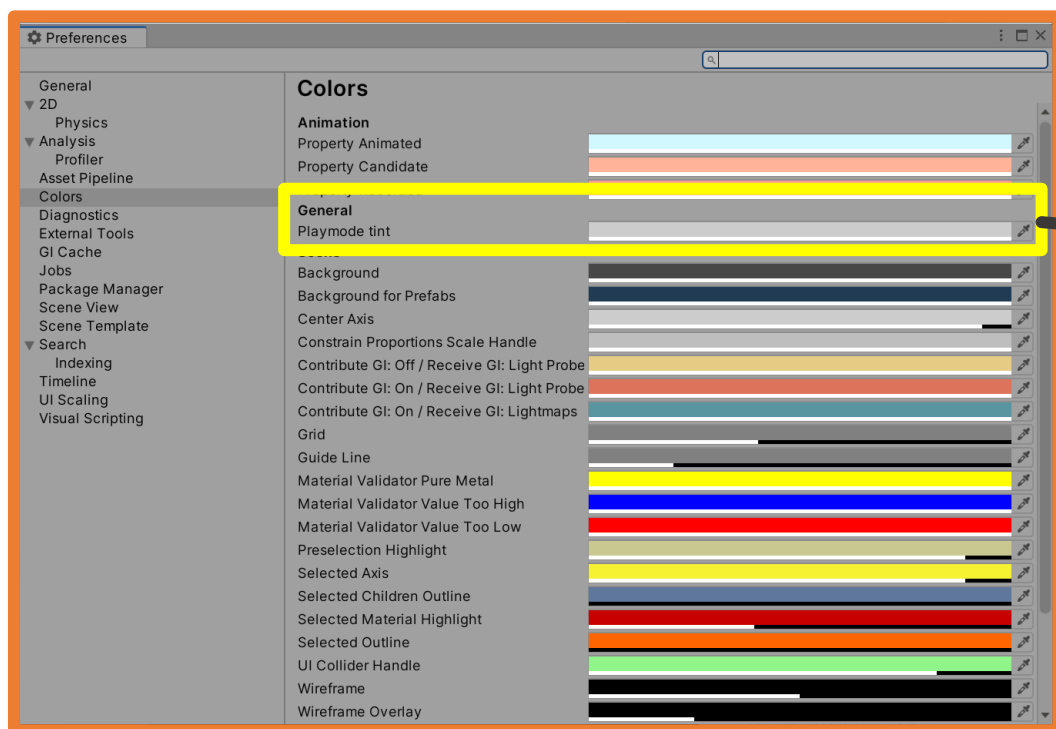




## 8.- Ejecución del juego

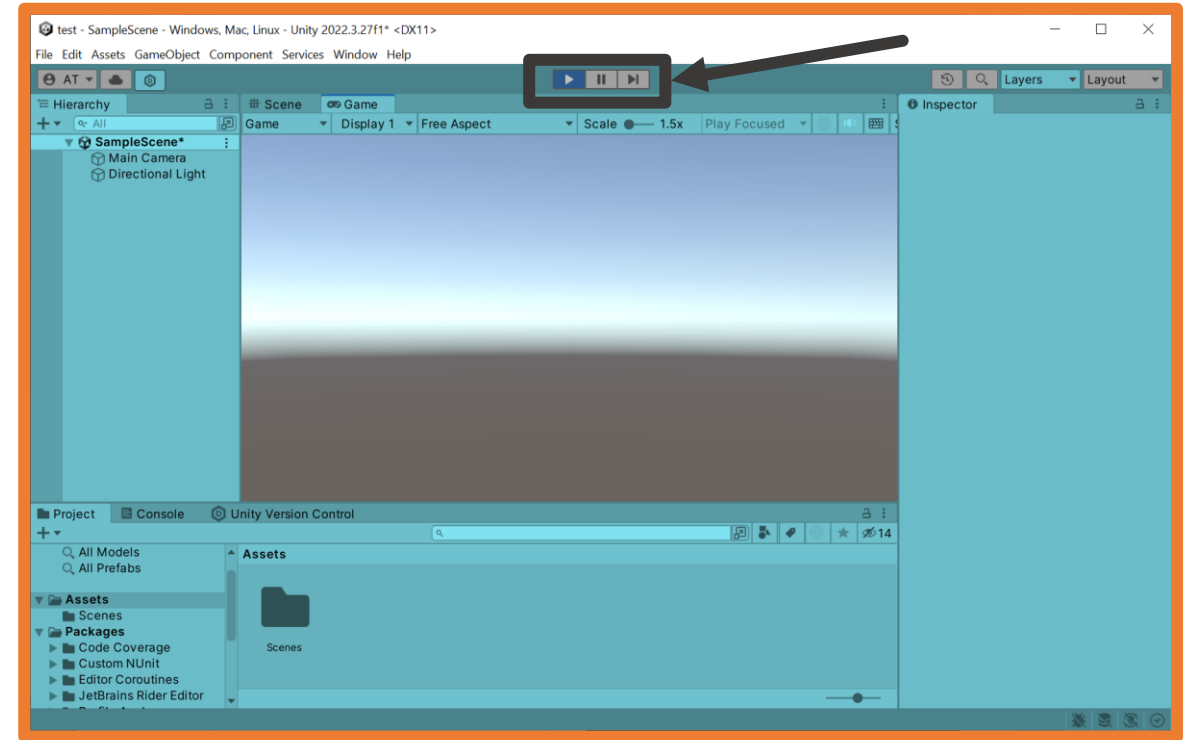
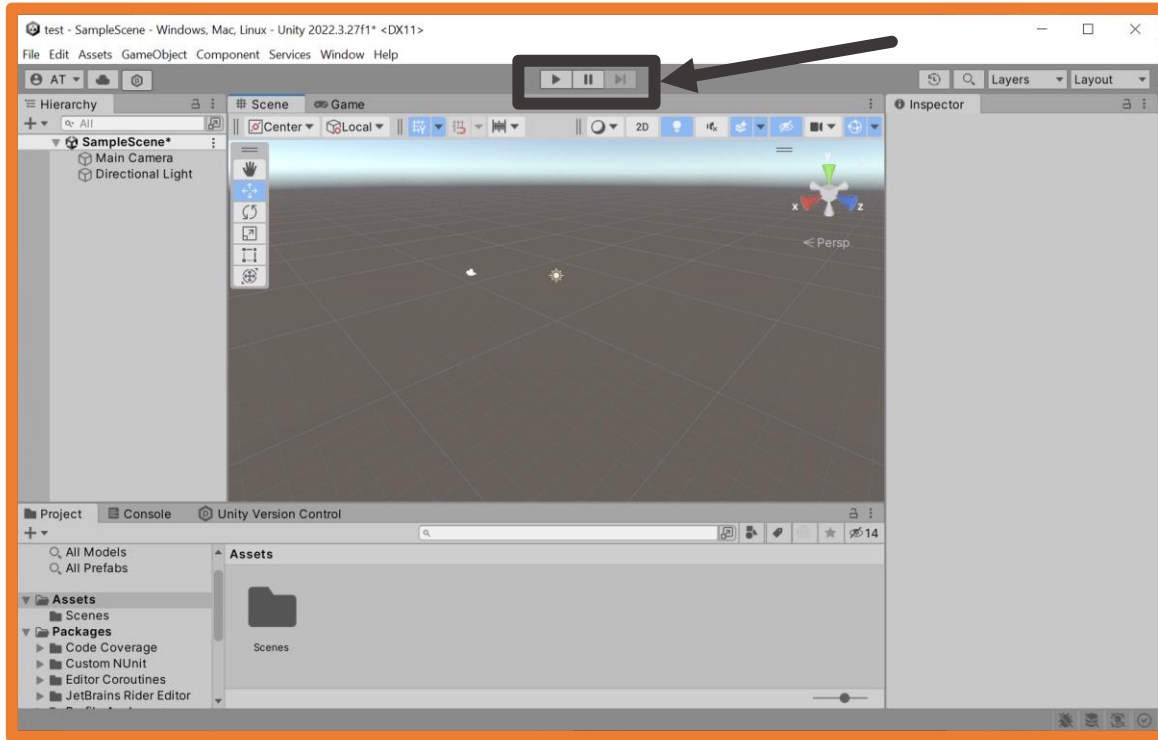
Desde el **Menú → Edit → Preferences** en la sección **Colors** se puede decidir qué color va a superponer a la interfaz cuando se ejecute el juego: **Playmode tint**.

Interesa poner un modo que aporte mucho contraste para así saber siempre si se está ejecutando o no.



## 8.- Ejecución del juego

Una vez cambiado el color se aprecia mucho mejor que se está ejecutando el juego.

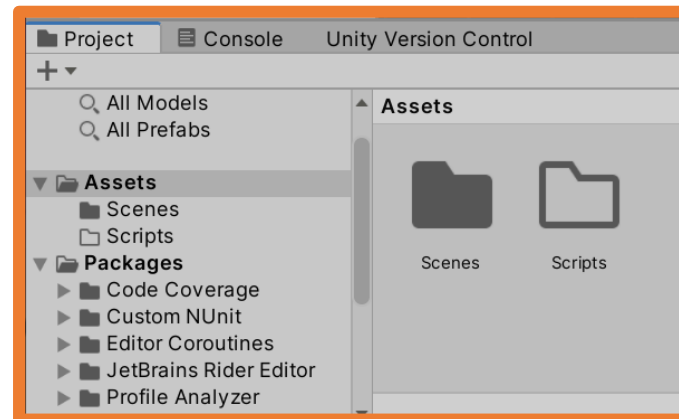


## 9.- Añadiendo código ejecutable al proyecto

Para añadir **código ejecutable** al proyecto se deben crear los archivos que permitan contener dicho código, a estos archivos se les llama **scripts**.

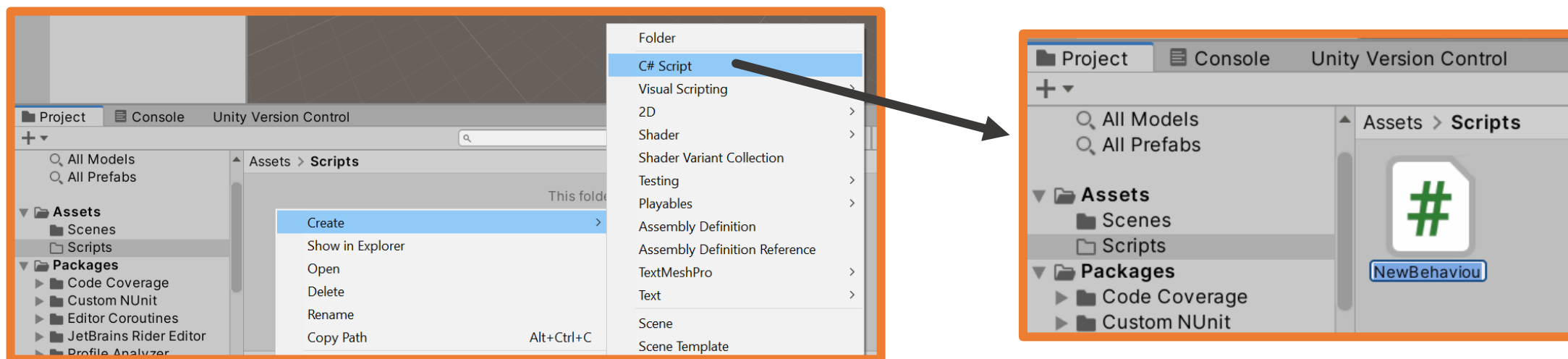
Existen varias maneras de añadir scripts al proyecto pero la más adecuada es mediante el gestor de archivos.

Además, es importante mantener una buena organización de los archivos por lo que lo más habitual es crear dentro de la carpeta **Assets** una carpeta llamada **Scripts** y en ella crear todos los Scripts necesarios.



## 9.- Añadiendo código ejecutable al proyecto

En la carpeta **Scripts** con el botón derecho sobre ella o sobre la parte derecha en el gestor (la que representa la carpeta), se debe seleccionar **Create** → **C# Script**

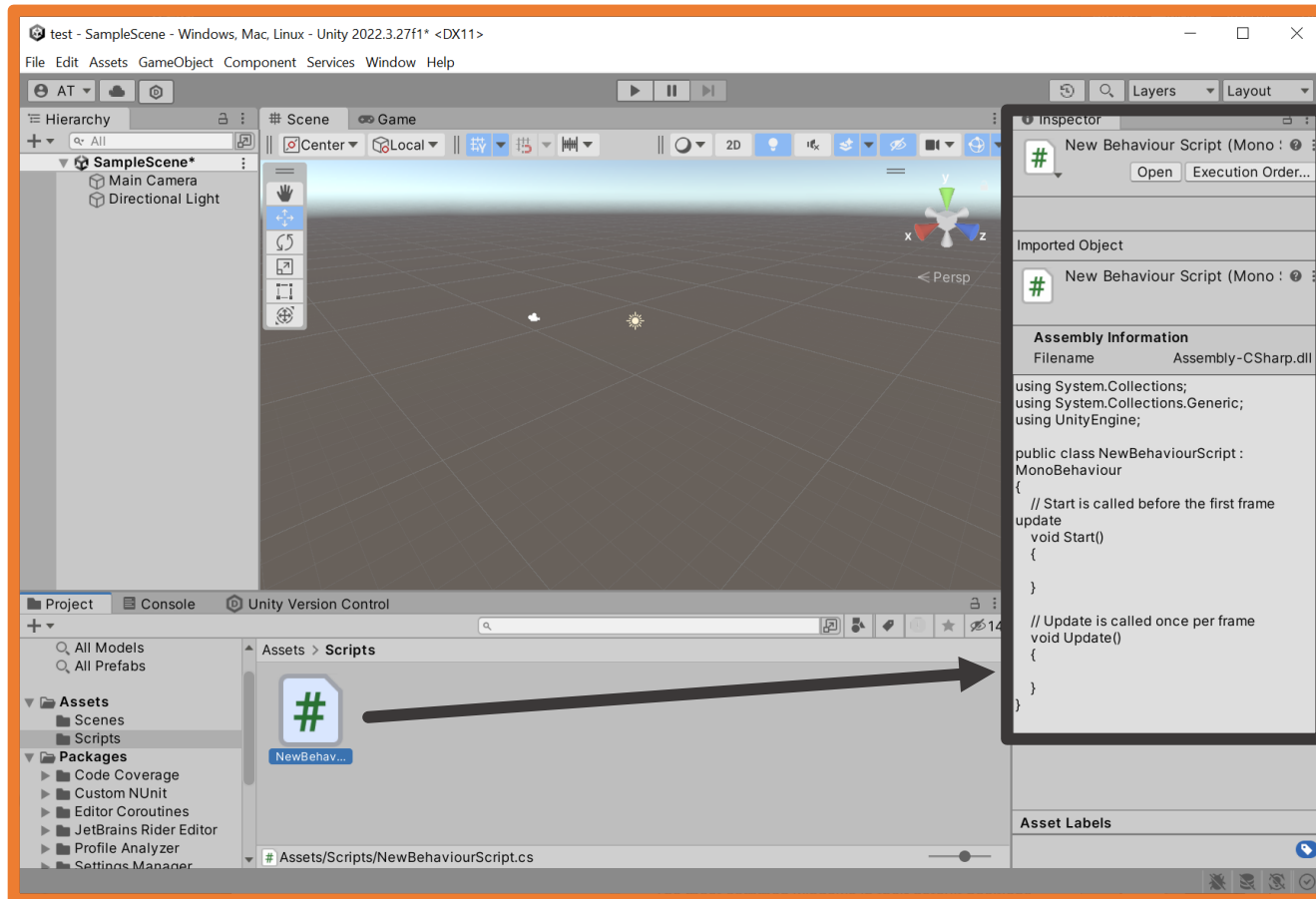


En ese momento se debe escribir el nombre del archivo, siguiendo las convenciones de C# que son similares a las de Java: **UpperCamelCase** con la primera en mayúscula.

El script debe tener un nombre que identifique correctamente las acciones que realiza.

## 9.- Añadiendo código ejecutable al proyecto

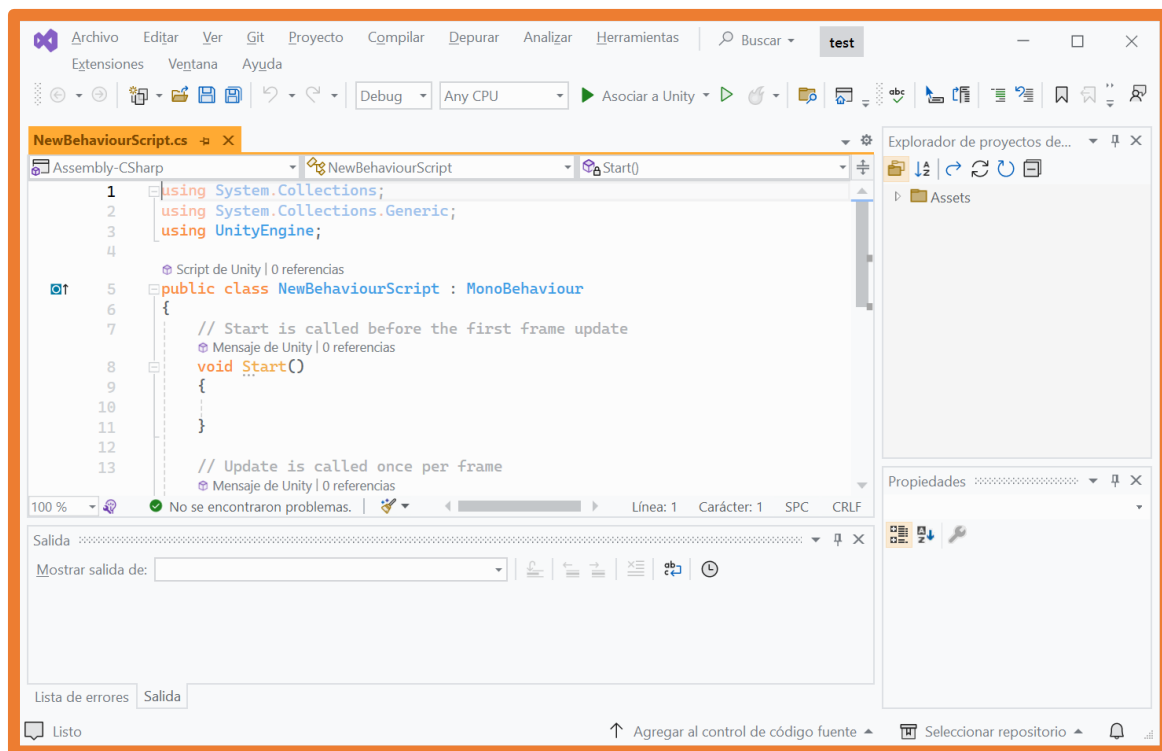
Cuando se selecciona un Script en el Gestor de archivos, en el Inspector se puede ver su código aunque no se puede editar desde ahí.



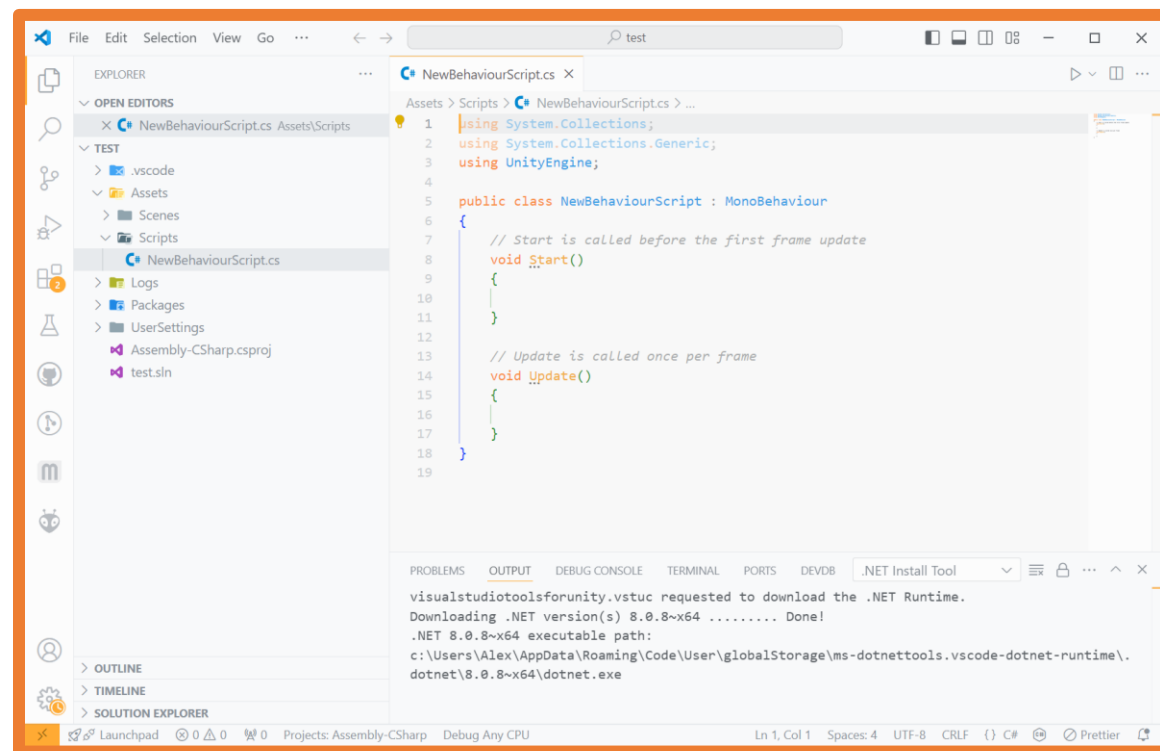
## 9.- Añadiendo código ejecutable al proyecto

Al hacer **doble clic** sobre el Script se abrirá el editor de código configurado.

Microsoft Visual Studio Community



Visual Studio Code



## 9.- Añadiendo código ejecutable al proyecto

Un Script de Unity por defecto tiene la siguiente estructura:



```
C# NewBehaviourScript.cs X
Assets > Scripts > C# NewBehaviourScript.cs > ...
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NewBehaviourScript : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18 }
19
```

The image shows a code editor window for a C# script named 'NewBehaviourScript.cs'. The script is a Unity MonoBehaviour. It contains two methods: 'Start()' and 'Update()'. The 'Start()' method is commented as 'Start is called before the first frame update' and the 'Update()' method is commented as 'Update is called once per frame'. Two black arrows point to the 'Start()' and 'Update()' method signatures respectively.

La cantidad de **frames** de un juego es variable por lo que el método **Update** no siempre se ejecuta la misma cantidad de veces.

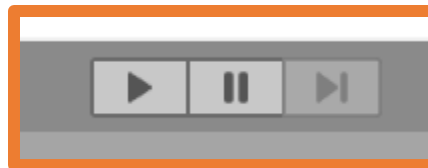
## 9.- Añadiendo código ejecutable al proyecto

El siguiente código hará que cuando se ejecute el script aparezcan por consola dichos mensajes.

```
public class HelloThere : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        Debug.Log("Start -> Hello there!");
    }

    // Update is called once per frame
    void Update()
    {
        Debug.Log("Update -> Hello there!");
    }
}
```

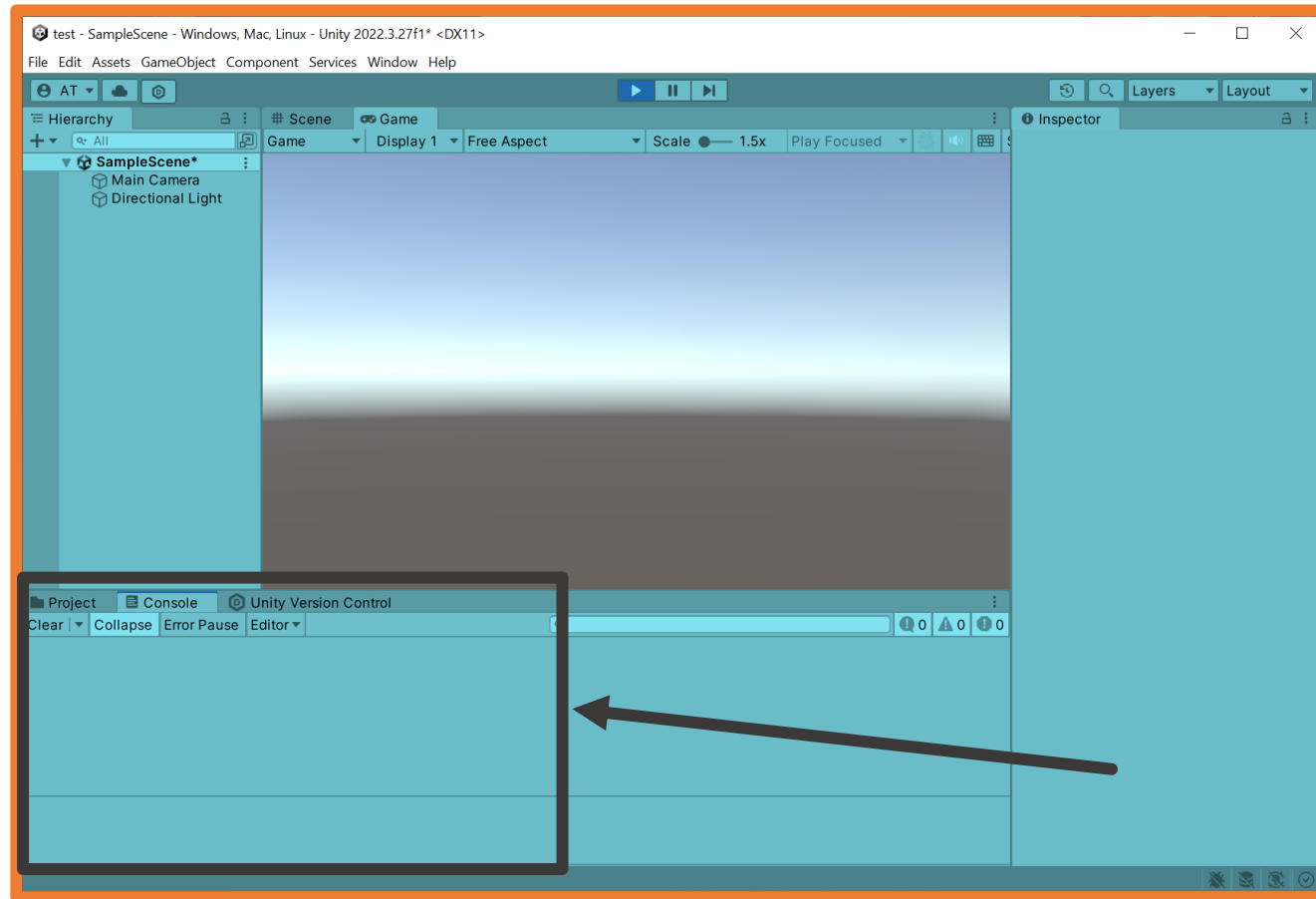
Para ejecutar el script se pulsa el botón "Play"  en la parte superior de Unity.





## 9.- Añadiendo código ejecutable al proyecto

Como se puede observar una vez seleccionada la pestaña **Console** no aparece ningún mensaje.



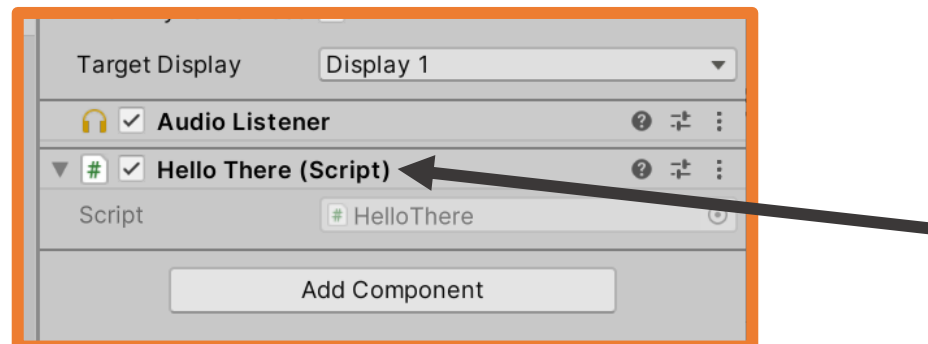
## 9.- Añadiendo código ejecutable al proyecto

Esto se debe a que para que un script se ejecute debe estar asociado a un GameObject.

Se puede asociar un script a un GameObject arrastrando el script desde el gestor de tareas al GameObject de diferentes maneras:

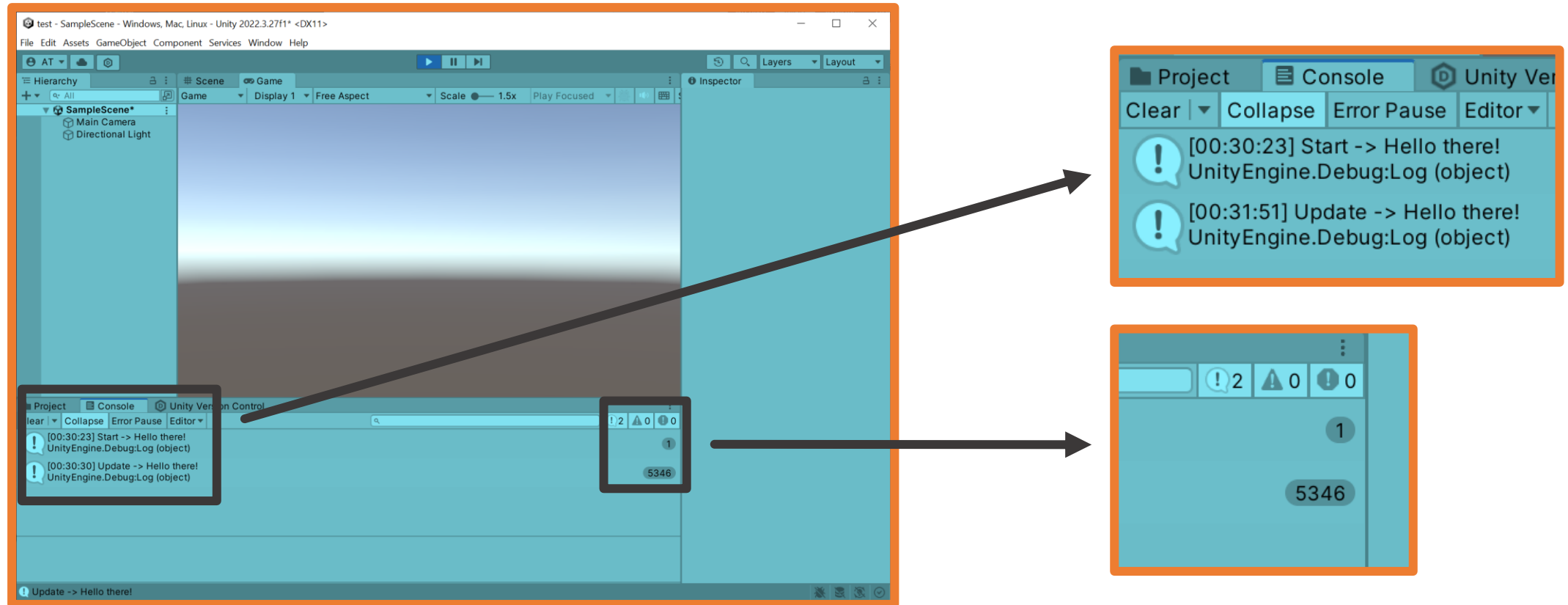
- Arrastrando el script hasta el GameObject en la jerarquía.
- Arrastrando el script hasta el GameObject en la zona de visualización.
- Seleccionando el GameObject y arrastrando el script hasta el Inspector.

Una vez se ha asociado un script a un GameObject, este se convierte en un componente del GameObject y aparece en el Inspector.



## 9.- Añadiendo código ejecutable al proyecto

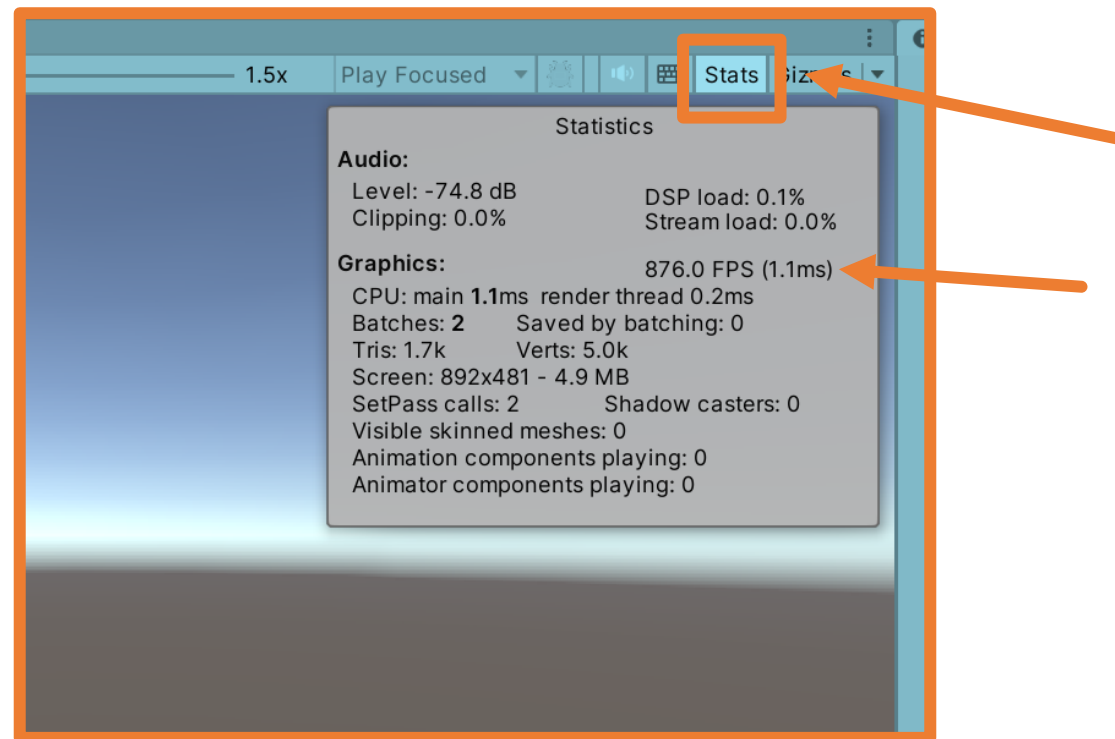
Ahora, al ejecutar el juego sí que se ejecutará el script creado.



Es importante marcar la opción **Collapse** que unificará en una línea los mensajes con el mismo contenido e irá aumentando el número de veces.

## 9.- Añadiendo código ejecutable al proyecto

También puede ser interesante activar la opción **Stats** en la visualización para obtener información como puede ser los **FPS** (frames por segundo) a los que se está ejecutando el juego.



# Práctica

## Actividad 3

Hello there!