

Arquitectura de Scripting en Unity



Comportamiento de los GameObjects

Scripting

El uso de scripts en Unity es esencial en todo juego ya que define la respuesta de los GameObjects a eventos dentro del juego o entradas del usuario.

Usos avanzados implican efectos gráficos, control de física o implementaciones de Inteligencia Artificial.

Scripting

El comportamiento de los GameObjects lo definen los Components que estén asociados a dicho GameObject.

Unity trabaja con 3 lenguajes de scripts:

- C#
- UnityScript
- Boo

Lenguajes

C#. Un lenguaje muy común y similar a Java o C++

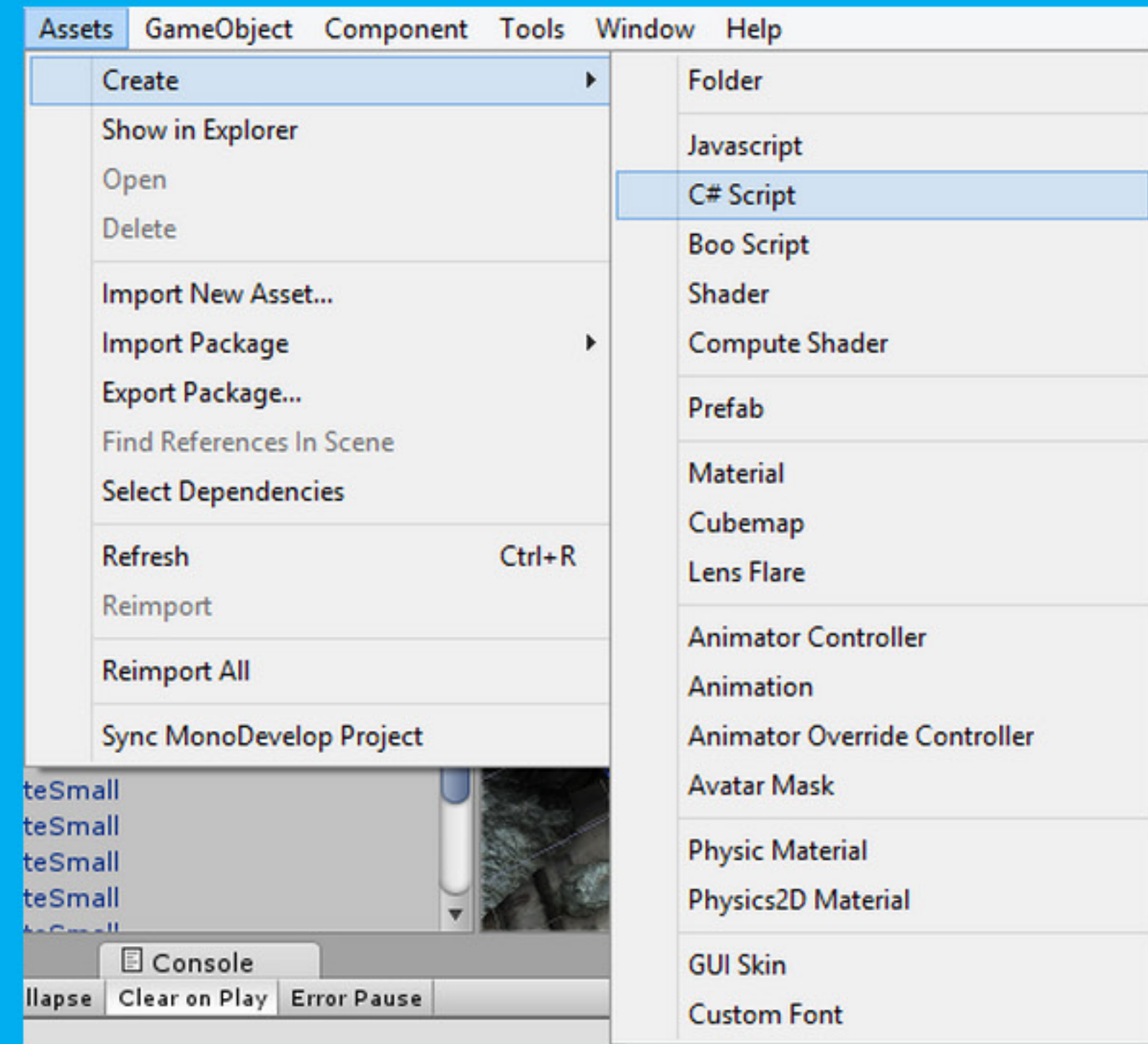
UnityScript. Es un lenguaje modelado a partir de JavaScript.

Boo. Un lenguaje .NET similar a Python.

Creando Scripts

Los scripts son Assets que se crean dentro de Unity.

Se puede crear un script desde el menú Assets.



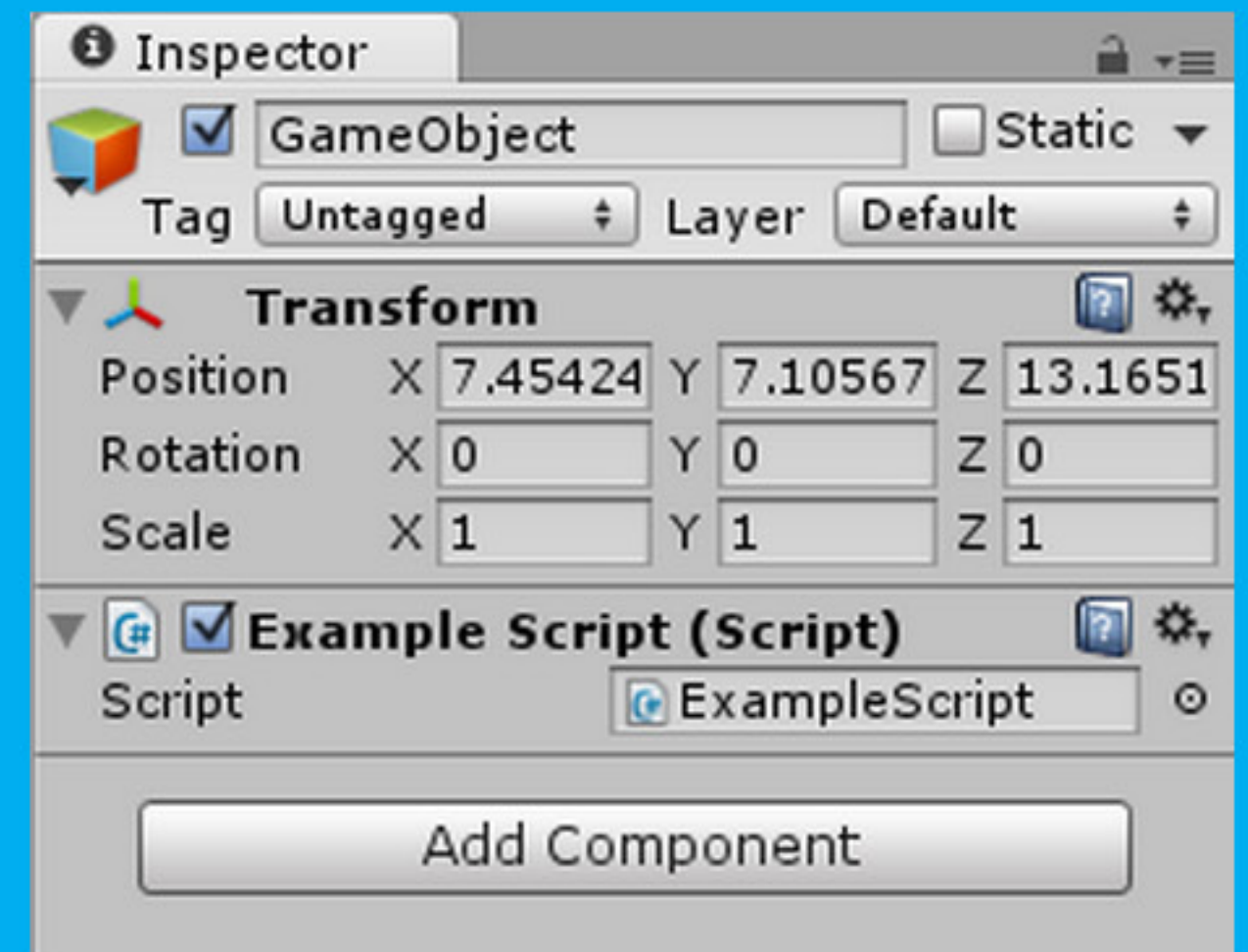
Arquitectura de un Script

Todo nuevo script en Unity contiene dos funciones por default:

- Start(). Es una función que se ejecuta antes de iniciar el comportamiento de un GameObject.
- Update(). Esta función se ejecuta en cada frame del juego.

Usando Scripts

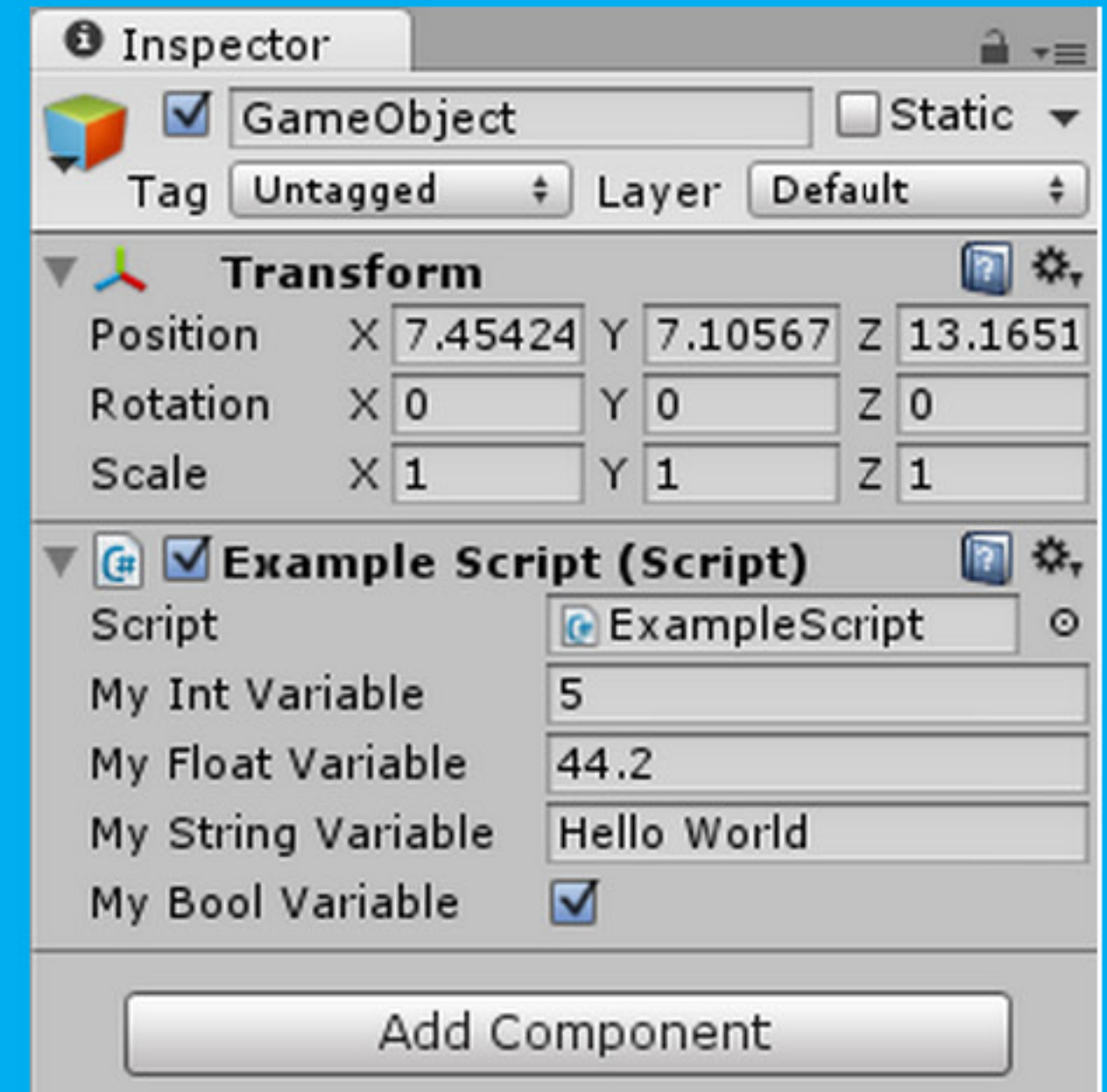
Para agregar Scripts podemos arrastrarlo hacia un GameObject en la ventana Jerarquía o en la Escena o al Inspector con el GameObject seleccionado.



Usando Scripts

En el Script aparecerán las variables que sean públicas dentro del script para que podamos modificarlas desde el Editor de Unity.

Podemos modificar los valores de las variables durante una corrida del juego para depurar.



Usando Scripts

También en el Inspector podemos ver cómo cambian los valores de las variables cuando probamos el juego.

Podemos habilitar o deshabilitar el script por completo si queremos ver cómo se comporta nuestro GameObject sin algún Component.