

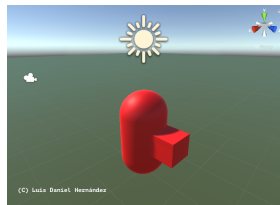
IA en Videojuegos

Primer Proyecto

L. Daniel Hernández¹

Departamento de Ingeniería de la Información y las
Comunicaciones
Universidad de Murcia

24 de enero de 2022





¹Erratas y sugerencias a ldaniel@um.es. Algunas imágenes pueden ser de internet. Si tuvieran derechos de autor/distribución conocidos, por favor, avisen.

Índice de Contenidos

- 1 Descargar e instalar Unity
- 2 Crear un proyecto nuevo a partir de una plantilla
- 3 Interface del Editor de Unity
- 4 Creación de una escena con un objeto
- 5 Moverse por la escena
- 6 Transformaciones básicas de un objeto
- 7 Crear y usar materiales
- 8 Jerarquía de objetos en una escena
- 9 Prefabs
- 10 Componentes Scripts

Descargar e instalar Unity

Descarga e instalación

- Descargar el instalador para tu sistema operativo desde
<https://unity3d.com/es/unity/qa/lts-releases>
<https://unity3d.com/es/get-unity/download/archive>
 Installing Unity
- Lo mínimo que se debe instalar es Unity, la documentación y el entorno de desarrollo Visual Studio.
 Installing Unity without the hub

**Crear un proyecto nuevo a partir de una
plantilla**

Creación de un proyecto

1. Abrir la aplicación y crear una cuenta.
2. Cuando salga una ventana emergente con el botón **New** hacer click sobre él.
3. Se mostrará otra ventana emergente. Entonces cumplimentar los campos.
 - Nombre de proyecto,
 - Ubicación donde se almacenará,
 - Seleccionar plantilla 3D, y
 - Pulsas el botón de creación.

Esperar a que se abra Unity (puede tardar un poco).



Project Templates

Interface del Editor de Unity

Interface

- Las ventanas principales que encontramos en la interface son: jerarquía, proyecto, escena, juego e inspector.



Unity's interface

- Existen más ventanas. Se pueden ver en el menú **Windows**.
- Las ventanas se pueden cambiar de sitio a gusto del usuario.



Customizing Your Workspace

- Hay un menú desplegable en la parte superior derecha de la ventana de inspección con el nombre **default** se pueden cargar algunas configuraciones por defecto.

El mismo botón ofrece la posibilidad de salvar la distribución que más le convenga al usuario e incluirla como una configuración más.

Creación de una escena con un objeto

La primera escena

Una escena está formada por GameObjects: cámaras, luces, objetos, personajes, paisajes, ...

- Seleccionar **File > New Scene**.
Si preguntase por salvar alguna escena previa seleccionar que no.
- Crear un objeto **3D Object > Quad**. Su creación se puede hacer desde:
 - El menú **GameObject**.
 - Pulsando botón derecho en la ventana de jerarquía.
- Seleccionar **File > Save Scenes** y guardar la escena con el nombre que se quiera.
- En la ventana del proyecto **Derecho > Create > Folder** y dar el nombre **Scenes**
- Arrastrar la escena a la carpeta.

Moverse por la escena

Moverse en una escena

- En la parte superior izquierda verá una mano. Haga clic sobre ella y después vaya a la ventana de la escena: deberá de ver una mano sobre la ventana. Opcionalmente puede pulsar en Q para activar la mano.
- Si ahora arrastrara el ratón sobre la ventana de la escena sin pulsa tecla alguna, pulsando la tecla Alt o pulsando la tecla control podrá: mover la escena, rotar la escena o hacer zoom sobre ella.
- Opcionalmente también puede pulsar el botón derecho del ratón y a continuación moverse con las teclas A, S, D, W, Q y E.
- El gizmo de escena nos muestra los ejes en los que nos movemos.



Scene View navigation

Transformaciones básicas de un objeto

Componente Transform I

- Visualice el Quad del paso previo: doble click sobre el Quad en la ventana de jerarquía (opcionalmente SHIFT+F).
- Junto a la mano aparece una serie de iconos que permiten transformar el objeto.

Se pueden activar haciendo click sobre ellos o usando las teclas W, E, R, T, Y. Permiten realizar las siguientes operaciones sobre el objeto: mover, rotar, escalar, transformación rectangular (más útil para 2d y UI) y transformación (mover+rotar+escalar).



Positioning GameObjects

- Seleccione cada uno de esos botones y juegue con los ejes y puntos.
- Observar que los ejes están coloreados con los mismos ejes que aparecen en el gizmo de escena (rojo = X, verde = Y, azul = Z).
- Verá que en la ventana de inspección aparece la componente Transform y que cuando edita el objeto cambian los valores de Position, Rotation, Scale.


Componente Transform II

- Introducir los siguientes valores:
 - Position=(0, 0, 0), origen de coordenadas del escenario.
 - Rotation=(90, 0, 0), rotando 90 grados sobre el eje X.
 - Scale=(100, 100, 1), incrementando por 10 sus lados en los ejes X y en Y.

Estos valores son siempre relativos a la componente Transform del padre (ver jerarquías).

Crear y usar materiales

Introducción a los materiales

- Seleccionar *Assets > Create > Material* desde el menú principal o desde el menú contextual de la vista del proyecto.
- Cambiar el nombre a hierba.
- Al hacer click se mostrará en la ventana de una inspección lo siguiente:
 **Creating and Using Materials**
- Hacer click sobre la caja de color junto al gotero y cambiar el color a la tonalidad de verde que prefiera. Verá el cambio de color en la esfera que está en la ventana de inspección.
- Crear la carpeta Materials y arrastrar el nuevo material a dicha carpeta.
- Arrastar el material desde la ventana del proyecto hasta (sobre) el objeto Quad de la ventana jerárquica. Acaba de usar el nuevo material en un objeto de la escena.

Jerarquía de objetos en una escena

Jerarquías de objetos

- Si se tienen dos GameObjects se puede arrastar uno sobre el otro en la ventana de jerarquía. El objeto hijo tendrá un Transform relativo al origen de coordenadas del padre, y el padre tendrá un Transform relativo al origen de coordenadas de la escena.
- Crear un objeto vacío con posición (0, 0, 0) y darle el nombre de Player.
- Crear un objeto vacío, hijo del anterior con posición (0, 0, 0) y llamarle Render.
- Crear un cilindro o una cápsula, hijo del anterior con posición (0, 1, 0).
- Crear un cubo, hijo del anterior con posición (0, 0, 0.5) y escala (0.5, 0.5, 1).
- Seleccione el objeto Player y juegue con los botones de posición, rotación y escala.

Observar que a pesar de cambiar la componente transform de Player, los valores Transform de los objetos hijos no se alteran (pues son relativas al padre).



The Hierarchy window

Prefabs

Prefabs

Los Prefabs en Unity puede entenderse como objetos del juego prefabricados. Los pasos usuales son:

1. Crear un objeto del juego, con la jerarquía que se considere. Normalmente al objeto `root` se le asigna posición `(0, 0, 0)` en la escena.
2. Arrastrar el objeto de la jerarquía a la ventana del proyecto. Con esta acción se acaba de crear el Prefab.
3. Dar nombre al prefab y guardarlo en la carpeta Prefabs.
4. Para usar un prefab basta arrastrarlo de la ventana del proyecto a la ventana jerárquica. Cada vez que se haga esta acción se tendrá un clone del prefab en la escena.
 - Un prefab en la escena no modifica el prefab origen. Modificar el prefab origen no modifica los clones previos que se tuvieran en la escena.



Prefabs



Nested Prefabs

Crear dos prefabs: el suelo (el quad de la escena) y el player (el objeto Player de la escena).

Componentes Scripts

Crear y añadir un script a un objeto

Un Script es una componente opcional en un objeto del juego. Permite darle el comportamiento que se desee mediante programación.

- Seleccionar *Assets > Create > C#Script* del menú principal, renombrar y guardar en un carpeta nueva llamada Scripts.
- Hacer doble click para abrir Visual Studio, programar y salvar.
- Arrastrar el Script al objeto del juego para el que queramos ese comportamiento.
- Si se desea, se puede hacer un prefab del objeto con dicho script.



Creating and Using Scripts

Si un Script se añade como componente a un objeto y contiene variables globales, serán editables desde el **inspector** pudiendose cambiar sus valores como cualquier otro parámetro de cualquier otra componente.



Variables and the Inspector

Un ejemplo de Script

El siguiente código permite controlar un objeto con el teclado.

```
1 public class Controller : MonoBehaviour {
2     public float velocity = 2f; // Visible en el inspector.
3
4     void Update() // Update is called once per frame
5     {
6         // Leer el teclado
7         Vector3 newDirection = new Vector3(
8             Input.GetAxis("Horizontal"),
9             0,
10            Input.GetAxis("Vertical"));
11
12        // Mirar en la dirección del vector leído.
13        transform.LookAt(transform.position + newDirection);
14
15        // Avanzar de acuerdo a la velocidad establecida
16        transform.position +=
17            newDirection * velocity * Time.deltaTime;
18    }
19 }
```

Añadir el script Controller al objeto Player la escena ... y a ¡ jugar !