

INTRODUCCIÓN A UNITY

JAIME3D.INFO



Tabla de contenido

1	ATAJOS DE TECLADO	2
2	IMPORTAR / EXPORTAR.....	3
2.1	Exportar desde 3DSTUDIO MAX :	3
2.2	Importar desde UNITY:.....	3
3	CREAR OBJETOS.....	4
4	PREFABRICADOS y uniones entre objetos	4
4.1	CREAR GAMEOBJECT	5
4.2	MODIFICAR PREFABRICADO	6
5	COLLIDER.....	6
6	IMPORT PACKAGE	7
7	CREAR MATERIALES.....	8
8	AÑADIR LUCES.....	8
9	SISTEMAS DE PARTICULAS.....	9
10	SIMULACIONES FISICAS	10
11	EL SCRIPT.....	11

1 ATAJOS DE TECLADO

W : Mover

E: Rotar

R: Escalar

Q: Pan

X : Cambia entre pivot local o Global.

Z : Cambia entre Pivote del objeto y centro del objeto.

V : Con la letra V pulsada podemos cambiar momentáneamente el pivote del objeto hacia uno de sus vértices y luego poder mover ese objeto a partir de ese vértice.

CTRL: Con la tecla Control pulsada activamos el snap. Tanto para posición como para rotación.

F: Focus in object, o lo que es lo mismo enfocamos la vista en el objeto seleccionado.

También podemos hacer lo mismo haciendo doble clic en los objetos de la escena que aparecen en la ventana Hierarchy.

ALT + LMB (botón izquierdo del ratón): Rotamos la vista alrededor del objeto seleccionado.

RMB: Rotamos la vista

ALT + RMB (botón derecho del ratón) : Zoom

CMB (botón central del ratón) : Zoom

RMB+W: Avanzamos la vista hacia adelante.

RMB+S: Tiramos la vista atrás.

RMB+A: Nos desplazamos hacia la izquierda.

RMB+D: Nos desplazamos hacia la derecha.

RMB+Q: Nos desplazamos hacia abajo.

RMB+E: Nos desplazamos hacia arriba.

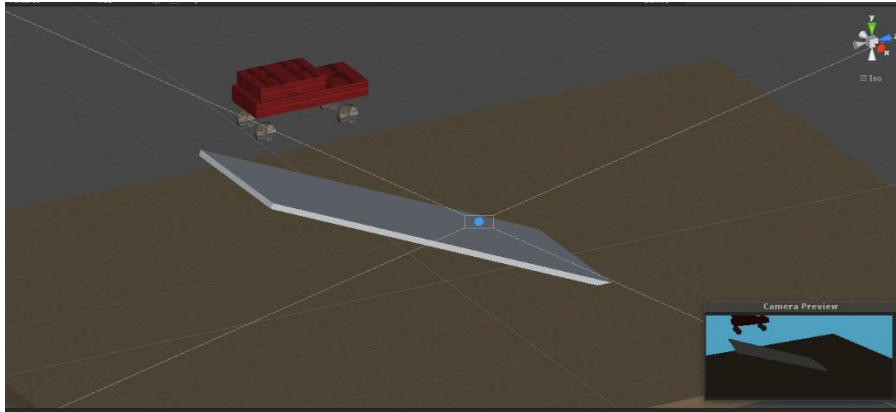
CTRL + D: Duplicar objetos.

Barra espaciadora: Modo pantalla completa sobre cualquier ventana.

CTRL+ Alt+F : Mueve el objeto seleccionado hasta el centro de la vista actual.

El centro de la vista podemos manipularla con el F(Focus in object) para poder alinear unos objetos con otros.

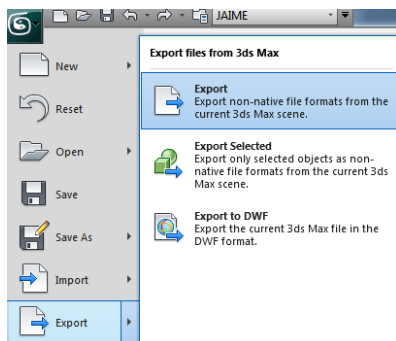
CTRL+Shift+F: Alinea cualquier objeto seleccionado a la vista actual. Nos puede servir para crear una vista que nos pueda ser interesante y alinear la cámara a esa vista para que cuando le demos al play para activar el juego, veamos exactamente lo mismo que veíamos en el viewport.



En la captura se aprecia cómo hemos alineado la cámara a la vista actual, de manera que veremos lo mismo en una y en otra.

2 IMPORTAR / EXPORTAR

2.1 Exportar desde 3DSTUDIO MAX :



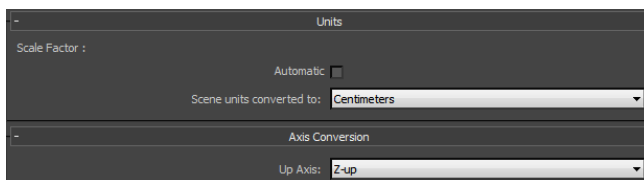
Con la escena u objeto cargado.

File → Export → Export

O si queremos guardar un solo objeto, File → Export → Export Selected.

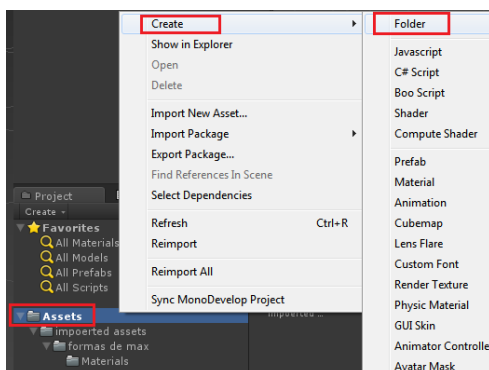
Se nos abrirá una ventana para nombrar al objeto y luego la opción que debemos escoger es .FBX

Se nos abrirá otra ventana de exportación en la que debemos elegir lo que se va a incluir en la escena u objeto exportado, como cámaras, luces , animación y el tipo de geometría.



Las unidades yo las suelo poner en centímetros, si trabajas en centímetros en Max. Damos a OK y ya hemos generado el archivo.

2.2 Importar desde UNITY:

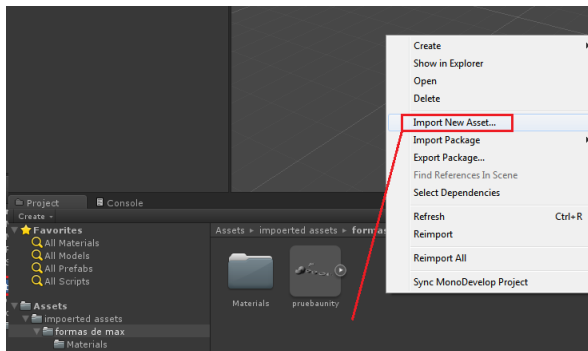


En la ventana de Project tenemos una carpeta por defecto llamada Assets.

Si clicamos botón derecho del ratón sobre ella podemos crear una carpeta nueva.

Crearemos la carpeta de importaciones o import assets o como la queramos llamar.

Dentro de esta crearemos otra con los objetos que traeremos de Max u otro programa 3D.



En la ventana de la derecha aparecen los objetos que contengan las carpetas seleccionadas de la izquierda.

Si clicamos sobre cualquier parte de esa ventana clic derecho del ratón, se nos abrirá un desplegable con la opción Import New Asset.

Aquí buscaremos el archivo .FBX generado por la aplicación 3D, en este caso 3D Studio

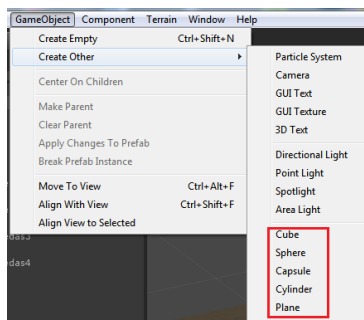
Max.

Una vez importado el objeto, lo podemos arrastrar a la escena.

Una vez en la escena y seleccionado, veremos a la derecha sus propiedades. En esta ventana de la derecha cambiaremos la posición numéricamente y su escala si lo vemos necesario.

3 CREAR OBJETOS

Es muy fácil crear objetos en Uniy. Aunque los objetos que nos permite crear el programa son muy básicos, nos van a permitir trastear lo suficiente con el programa como para hacernos una idea de las posibilidades que tiene.

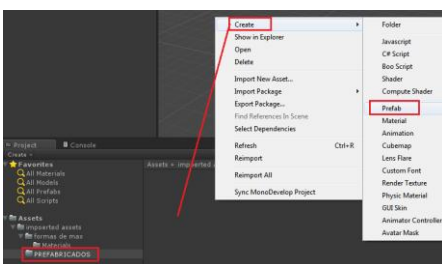


Las primitivas standard que podemos encontrar ,están dentro del desplegable GameObject→Create Other.

Estos objetos los podremos modificar a nuestro gusto moviéndolos rotándolos y escalándolos.

La imagen que aparece en el apartado de atajos de teclado de este manual, ha sido construida íntegramente con estos objetos básicos.

4 PREFABRICADOS y uniones entre objetos

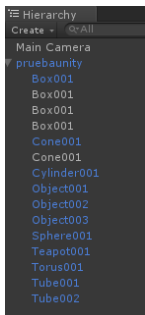


Los prefabricados es una manera de unir objetos de la escena para que luego puedan ser duplicados como un conjunto y no como objetos separados.

Vamos a crear dentro de la carpeta que habíamos creado antes (la de Imported Assets u Objetos Importados), una carpeta de objetos Prefabricados (ya sabes, con clic derecho del ratón→ Create → Folder).

Dentro de esta carpeta haremos clic derecho del ratón sobre la ventana de la derecha y crearemos el prefabricado con Create→Prefrab.

Objetos sueltos pueden arrastrarse desde la ventana Hierarchy hacia el prefabricado que acabamos de crear, pero la idea es agrupar primero esos objetos para luego incluirlos en el prefabricado, ya que no vamos a poder arrastrarlos todos a la vez.

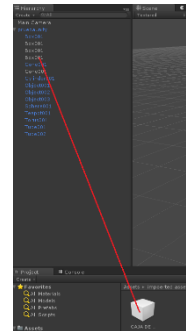


En la ventana Hierarchy podemos ver que los objetos están dentro de un objeto llamémosle padre, del que cuelgan todos los demás. Este objeto padre tiene el nombre de la escena que hemos importado en formato FBX (en este caso pruebaunity).

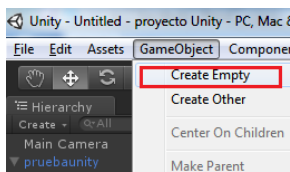
Todos estos objetos aparecen de color azul.

Si duplicamos alguno de ellos veremos que las copias pasan a ser de color blanco.

Si arrastramos unos sobre otros, se crean vínculos descendientes entre ellos, es decir, el objeto arrastrado pasa a ser hijo, y el objeto al cual hemos arrastrado el hijo, pasa a ser el padre.



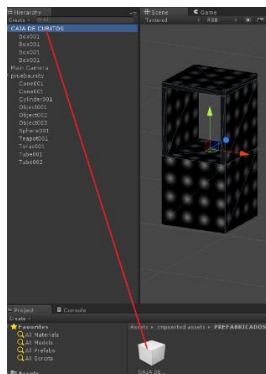
4.1 CREAR GAMEOBJECT



Una vez en este punto, lo que necesitamos para poder agrupar bien los objetos, es crear un gameobject, para eso vamos a la pestaña GameObject→Create Empty.

Lo renombramos como el prefabricado que hemos creado antes.

Este tipo de objeto no se representa en escena, digamos que es como un punto imaginario en el espacio, pero lo podemos mover, rotar y escalar. Lo he llamado Caja de Cubitos.



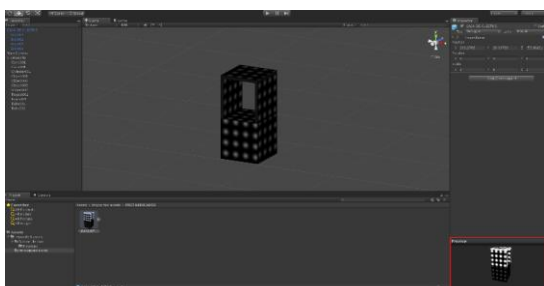
Debemos arrastrar los objetos que queremos que pertenezcan al prefabricado en el nuevo gameobject.

Si esos objetos estaban en azul es porque pertenecían a la escena importada, al arrastrarlos nos dirá si queremos destruir su vínculo de prefabricados, le diremos que sí.

Aclaración: Por defecto unity interpreta que cuando importamos una escena de 3DStudio Max entera, los objetos forman parte de un grupo.

He arrastrado las cajitas Box al gameobject. Ahora ya podemos arrastrar el gameobject al prefabricado de la ventana de Assets.

En resumen, los objetos se han arrastrado a un gameobject y este a su vez dentro de un prefabricado.



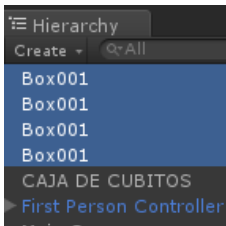
El icono del Prefabricado pasa a tener la apariencia de los objetos que tiene dentro.

Si lo seleccionamos nos aparece una previsualización a la derecha del todo, la cual podemos rotar.

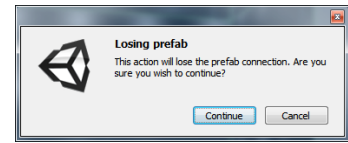
4.2 MODIFICAR PREFABRICADO

Para modificar un prefabricado deberemos temporalmente desvincular los objetos que tiene dentro.

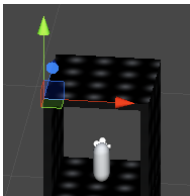
Arrastramos los objetos fuera de su contenedor



En ese momento aparece el mensaje de que perderemos la conexión existente en el prefabricado. Le decimos que nos parece muy bien y continúe.

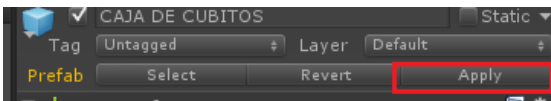


Aparecerán de color blanco y lógicamente fuera del gameobject, que yo llamé Caja de Cubitos, como se puede apreciar en la captura de la izquierda.



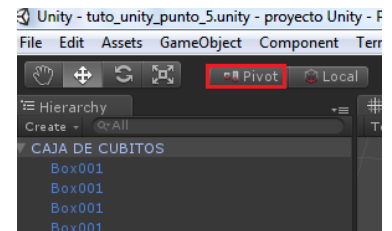
Vamos a seleccionar el Gameobject (caja de cubitos) y lo vamos a llevar a las coordenadas que nos vayan bien, yo lo que he hecho es colocarlo en una de las esquinas de la pared izquierda.

Ahora vamos a volver a vincular los objetos Box arrastrándolos de nuevo al gameobject.



Lo seleccionamos y vemos que siguen de color blanco. Si aplicamos cambios clicando

sobre Apply volveremos a tener los objetos y el gameobject de color azul. Eso querrá decir que el prefabricado queda cerrado de nuevo.

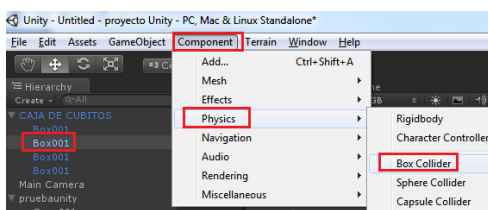


Para ver los cambios que hemos realizado iremos donde pone Center y lo pondremos en modo Pivot. El pivote del objeto está en la esquina donde lo he colocado antes.

5 COLLIDER

Vamos a añadir interacción a nuestro objeto.

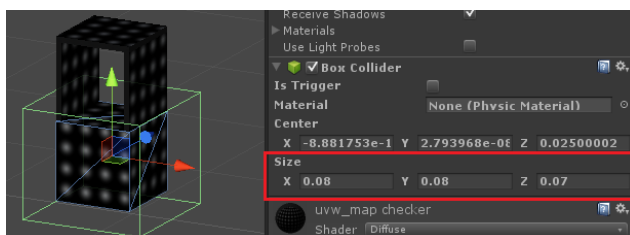
El Collider es colisión, es decir que box collider sería colisión con forma cuadrada.



Como la parte inferior de nuestro prefabricado es un cubo, he pensado que lo podemos usar de suelo, y los cubos aplanados de los lados se pueden usar de pared.

Seleccionaremos desde la ventana Hierarchy el cubo que hace de suelo y clicaremos en el desplegable

Component → Physics → Box Collider.



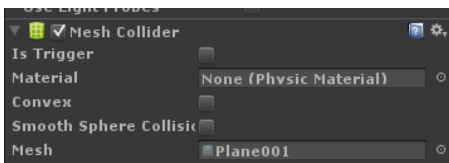
Se crea una caja de color verde alrededor del objeto, a la cual le podemos cambiar el tamaño.

Yo la dejaré con el tamaño del objeto porque no me interesa que interactúe más allá del espacio que ocupa.

Haremos lo mismo con las paredes.

Al agregar el Box Collider al objeto, cualquier otro objeto de la escena chocará contra él si quiere atravesarlo.

Si seleccionamos el objeto que contiene el collider, lo podremos ver en la ventana de la derecha.

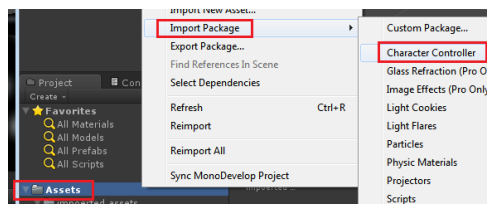


He creado un objeto y he decidido cambiar el Box Collider por el Mesh Collider.

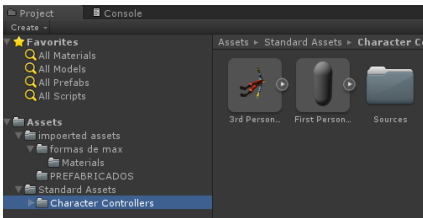
El Mesh Collider en lugar de tener una forma cuadrada, tiene la misma forma que el objeto original, de tal manera que la colisión será mucho mas realista.

Si clicamos botón derecho sobre el mesh collider veremos que podemos borrarlo con la opción Remove Component.

6 IMPORT PACKAGE



Vamos a incorporar un personaje que interactúe con paredes y suelo. Para eso hacemos clic derecho sobre Assets (dentro de la ventana de Project) → Import Package → Character Controller.



Por defecto sale una ristra de componentes a importar, simplemente le damos a import y nos quedamos tan panchos.

Se añade una carpeta llamada Standard Assets y dentro de esta Character Controllers. Si la seleccionamos nos salen 2

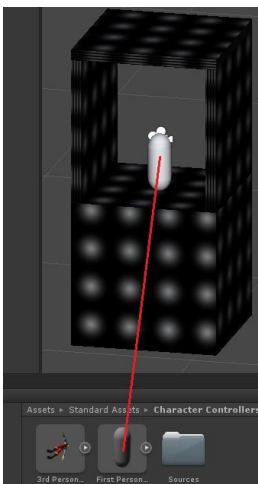
opciones, tercera persona y primera persona. En la tercera persona se ve el personaje que controlamos y en la primera persona digamos que nosotros seríamos el personaje y veríamos a través de una cámara.

Arrastramos el First Person encima del cubo que he creado para generar el suelo o lo que hayas creado para esto, en realidad vale cualquier objeto con un Collider.

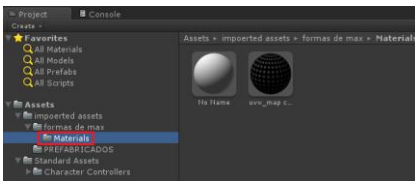
Le podemos dar al Play que hay en la pantalla de unity arriba, para que comience el juego.

Veremos que nosotros somos el objeto y lo controlamos con el ratón para rotar la vista y con el cursor para movernos. Chocamos con las paredes porque son un Collider. Si damos al play de nuevo salimos del juego.

Si seleccionamos Main camera podemos cambiar por ejemplo el fondo a negro.

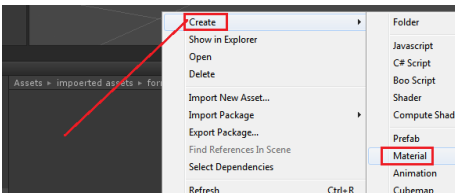


7 CREAR MATERIALES



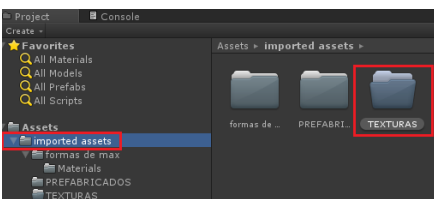
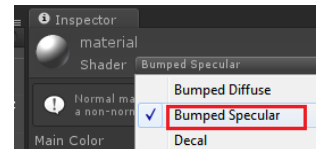
Por defecto cuando importamos una escena, aparece una carpeta de materiales. Yo en el ejemplo tengo 2. Uno gris y otro con puntitos blancos en fondo negro.

Vamos a borrarlos los 2 y dejar vacía la carpeta.



Clicamos botón derecho en la zona vacía y Create → Material.

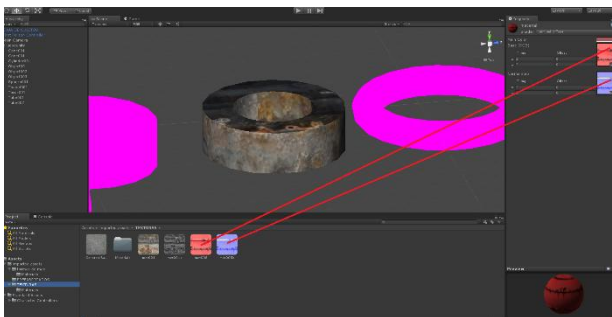
Con el nuevo material seleccionado, en la ventana de la derecha, donde pone Shader clicamos y lo desplegamos para elegir el Bumped Specular (por ejemplo).



En la ventana Project, dentro de la carpeta Assets, vamos a crear dentro de Imported Assets una nueva carpeta llamada texturas.

Entramos en la carpeta y Clic derecho en zona vacía → Import New Asset, elegimos la textura y aparecerá en unity.

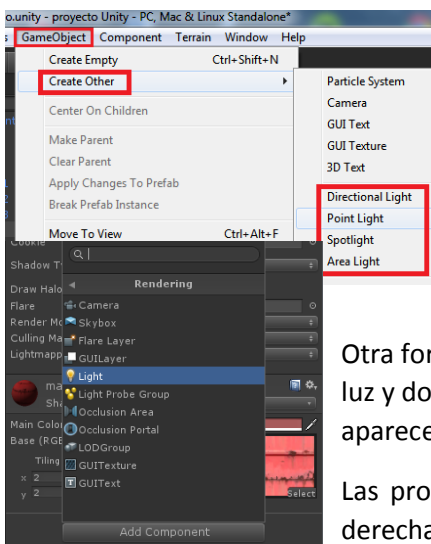
También podemos arrastrar los mapas a unity directamente.



Volvemos a nuestro material creado desde unity con el Shader Bumped Specular, al cual podemos arrastrar las texturas añadidas anteriormente.

Si arrastramos a donde pone normal map una textura bump, automáticamente nos va a generar un normal map a partir de esa textura.

8 AÑADIR LUCES



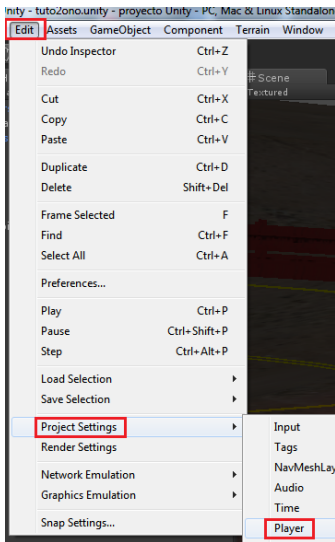
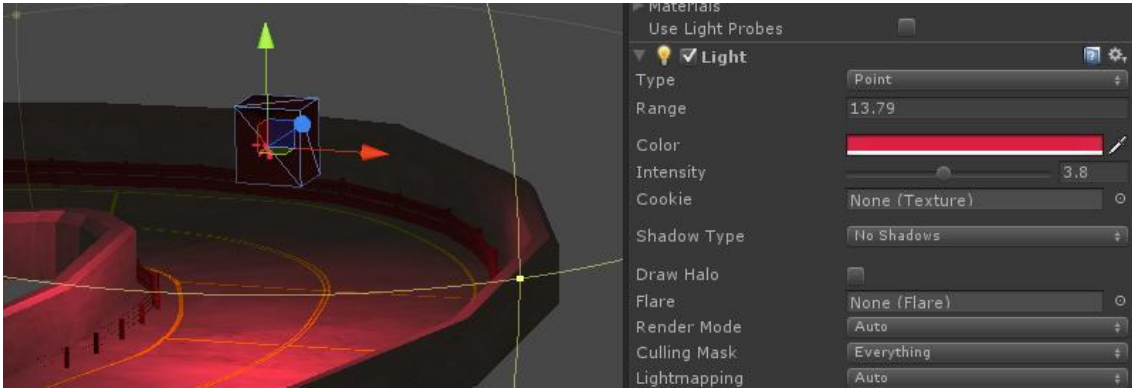
Una de las formas que tenemos de crear luces es irnos a la pestaña GameObject → Create Other.

En ese menú desplegable nos aparecen todas las luces, desde Directional Light hasta Area Light.

Si clicamos sobre Point Light por ejemplo se nos crea una luz puntual en la escena con sus propiedades en la ventana de la derecha.

Otra forma de crear luces es, irnos a un objeto que queramos que genere luz y donde pone Add Component → Rendering → Light. Automáticamente aparece una luz vinculada a ese objeto.

Las propiedades de esa luz se ven como siempre en la ventana de la derecha.



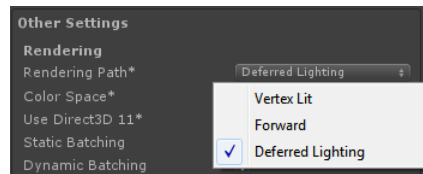
Range es el tamaño de la esfera que rodea la luz y que indica hasta donde llega su radio de acción.

Color, el color de la luz, e intensity la intensidad con la que brilla.

Shadow Type: Para activar las sombras en un Point Light, debemos configurar lo siguiente.

Nos vamos a Edit → Project Settings → Player.

En la ventana de la derecha, en Other Settings → Rendering Path, debemos elegir Rendering Path → Deferred Lighting.

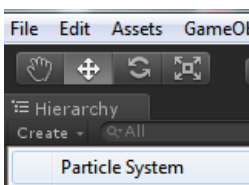


Ahora ya podremos ver sombra con las luces Point. En teoría deberíamos

verlo en la escena directamente, pero si no fuese así, ten por seguro que cuando le des al play para activar tu juego, las sombras aparecerán en escena.

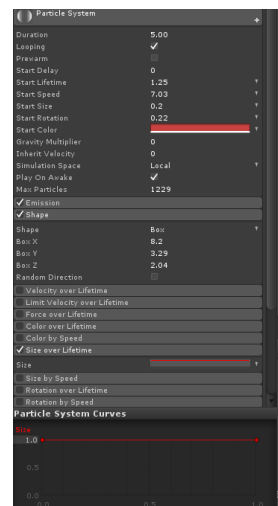
Briconsejo: Para crear la luz de por ejemplo una lámpara, lo ideal es crear la luz justo sobre la bombilla y hacer un prefabricado de la lámpara y la bombilla, los dos objetos juntos, para poderlos luego multiplicar si es necesario.

9 SISTEMAS DE PARTICULAS



Para añadir un sistema de partículas, lo único que hay que hacer es clicar sobre Create (debajo de Hierarchy) → Particle System.

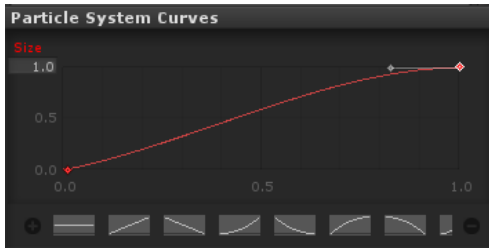
A la derecha se despliega la ventana con todas las opciones del sistema de partículas.



Si desplegamos Emission, el valor Rate es el número de partículas que se van a representar en escena.

Si desplegamos Shape, podemos cambiar la forma a partir de la cual se emiten las partículas y cambiar sus dimensiones en los 3 ejes de coordenadas. Random direction cambia la dirección de las partículas de una manera aleatoria.

Sobre la pestaña Emission existen las opciones generales. Entre las cuales tenemos la velocidad (start speed), el tamaño (star size), la vida media de las partículas (start lifetime), el color (start color), la gravedad y por ejemplo el Prewarm que sirve para cuando paramos la simulación y la volvemos a reanudar, vuelva a empezar desde donde lo dejamos y no vuelva a empezar de 0.



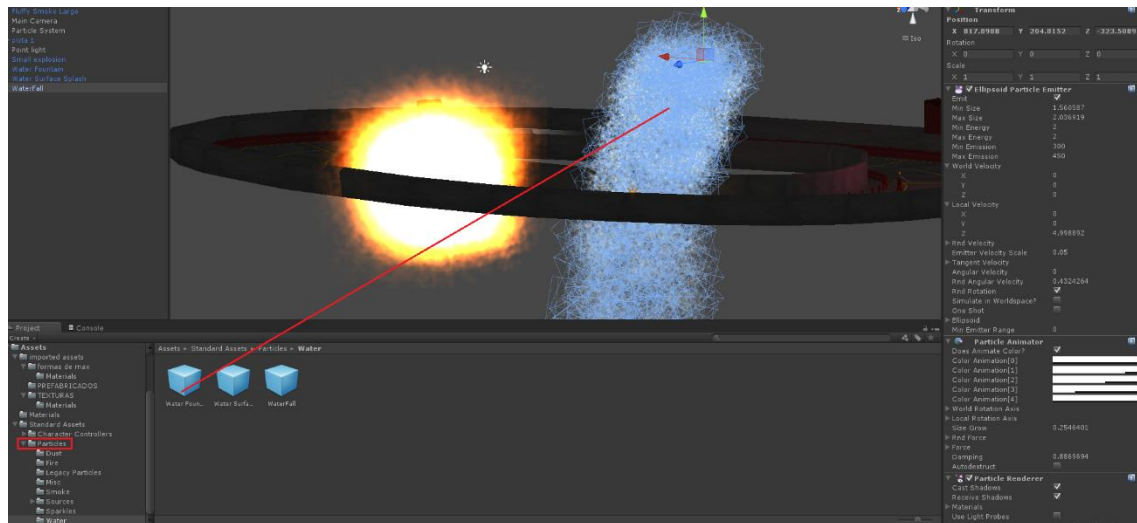
La pestaña Size Over Lifetime muestra una curva para poder modificar el tamaño de las partículas de manera que empiecen siendo pequeñas y acaben siendo mas grandes, o al revés.



La pestaña Renderer muestra el material asignado a las partículas.

Si clicamos en la pelotica que hay a la derecha de Default-Particle aparecen los materiales que tenemos en el proyecto y que podemos asignar a las partículas.

Para incorporar a la escena sistemas de partículas ya definidos como por ejemplo fuego, humo agua y algunos efectos mas, solo hay que hacer clic derecho sobre Standard Assets o Assets o la



carpeta donde queramos ponerlos dentro del apartado Project , y desplegar la carpeta. A continuación arrastraremos los prefabricados sobre la escena y retocaremos sus valores para que se asemeje al efecto que estamos buscando.

10 SIMULACIONES FISICAS

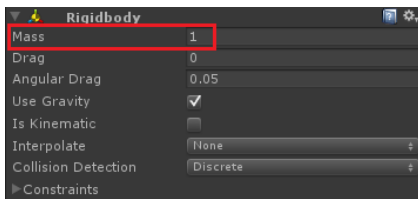
Antes hemos visto como funcionan los Collider. Son objetos que no se pueden atravesar.

Se pueden aplicar a estos objetos colliders en forma de cubo o de esfera, o tambien que se adapten a la forma del objeto con el Mesh Collider.

Ahora vamos a aplicarles simulacion fisica de manera que interactuen con el escenario y personajes.

Vamos a convertir un Collider en un Rigidbody. Para esto tenemos 2 opciones.

La primera sería irnos al desplegable de arriba Component→Physics→RigidBody, teniendo el objeto seleccionado, dentro de las propiedades el objeto , en el Inspector, darle al botón Add Component por el que accedemos al mismo menú.



Al agregar el RigidBody al objeto, si este está apoyado sobre alguna parte del escenario, posiblemente no pasará nada, pero si este está elevado , al darle al Play para comenzar el juego, el objetó se precipitará por efecto de la gravedad y caerá sobre otro objeto que sea un collider, rebotando según el caso. Logicamente la opción Use Gravity debe estar

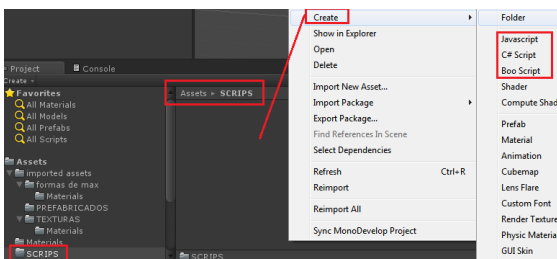
seleccionada para que pase esto.

Un objeto RigidBody nunca debe interseccionar con otros objetos Collider, ya que chocarian desde dentro y posiblemente desaparecerian.

11 EL SCRIPT

El script se puede decir que son las líneas de comandos que hacen interactuar los elementos en unity a través de un lenguaje de programación.

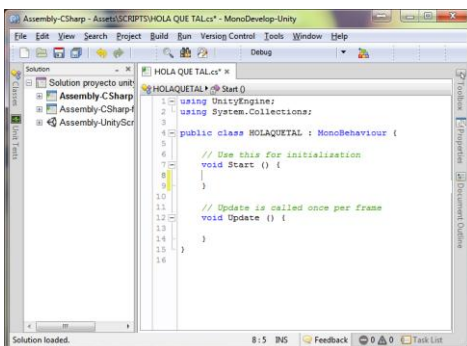
Es la introducción de código en unity a través de Javascript,C# y BooScript, según el lenguaje que más te convenga o sepas para programar.



Para empezar lo que podemos hacer es crear una carpeta dentro de Project→Assets , llamada Scripts.

Dentro de esta, hacemos CLIC DERECHO→ Create y elegimos un lenguaje.

Voy a elegir por ejemplo C# Script.



Nos aparece un icono llamado NewBehavior. Cambiaremos su nombre. Vamos a crear un mensaje en unity, y por eso llamaremos al archivo creado, HOLA QUE TAL. Por poner un ejemplo.

Si hacemos doble clic sobre el, se nos abre la ventana para

introducir código.

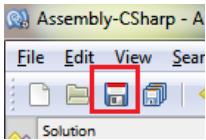
En esta ventana nueva, vemos en verde dos mensajes, el primero use esto para inicialización.

Esto quiere decir que lo que escribas despues del void start aparecerá solo una vez al iniciar el juego.

A continuación el segundo mensaje es Update is called once per frame, es decir, que se actualizará en el juego en cada frame.

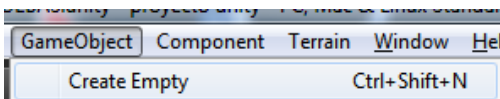
Vamos a crear un mensaje que se repita cada frame, para eso vamos después del void update y escribimos : `Debug.Log("HOLA QUE TAL");`

```
// Update is called once per frame
void Update () {
    Debug.Log ("HOLA QUE TAL");
}
```

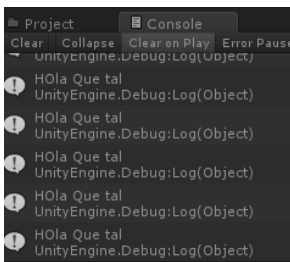


Le damos al disquette de grabar.

Para que surta efecto , debemos arrastrar el script a un objeto de la escena y para



eso crearemos un GameObject yendo a GameObject→Create Empty. Lo podemos llamar ...



Mensaje de la maquina, por ejemplo.

Arrastramos el script dentro del GameObject, y ahora si ,podemos darle al Play para activar nuestro juego.

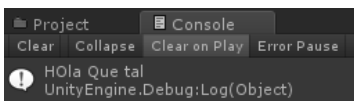
Abajo a la izquierda, al lado de Project, tenemos la pestaña Console que es la que nos devuelve el mensaje que hemos creado.

Si no aparece el mensaje una y otra vez, es posible que hayamos cambiado algo sin darnos cuenta. Podemos volver a empezar creando un script y un gameobject nuevos.

Aparecen muchos mensajes, porque al ponerlo después del void update, aparece un mensaje por cada frame. Paramos el juego y en Console le damos a Clear, para borrar los mensajes.

```
// Use this for initialization
void Start () {
    Debug.Log ("HOLA Que tal");
}
```

Ahora lo que deberíamos hacer es ,cortar-pegar el debug.log del mensaje después del Void Start.



Le damos a grabar, y ahora solo aparece el mensaje que habíamos creado ,una sola vez al iniciar el juego.