

«Talento Tech»

Videojuegos

Clase 08





Clase N° 8 | Conceptos básicos

Temario:

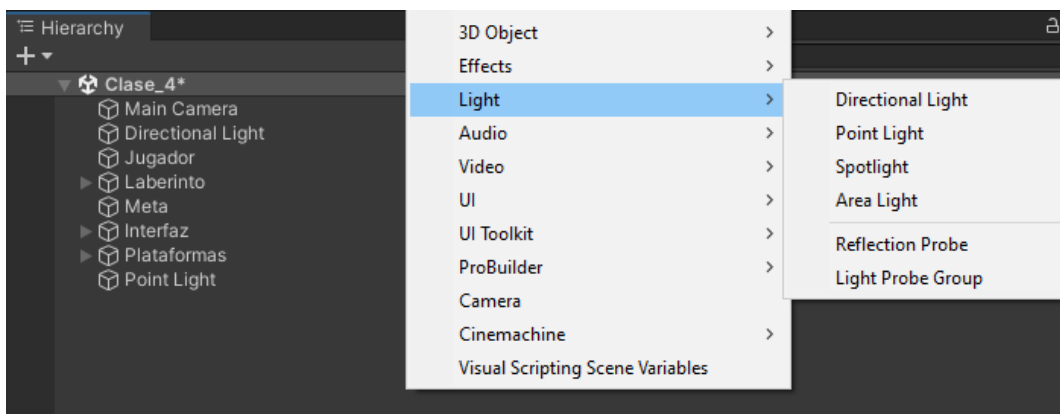
- Iluminación
- Sonido
- Skybox

Iluminación

Las luces en **Unity 3D** son elementos que iluminan escenas y objetos en un entorno 3D.

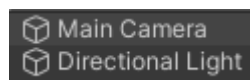
Para crear una luz, vamos a ir a la *Jerarquía*, **clik derecho** → **Light**

Y nos va a aparecer una lista de luces:

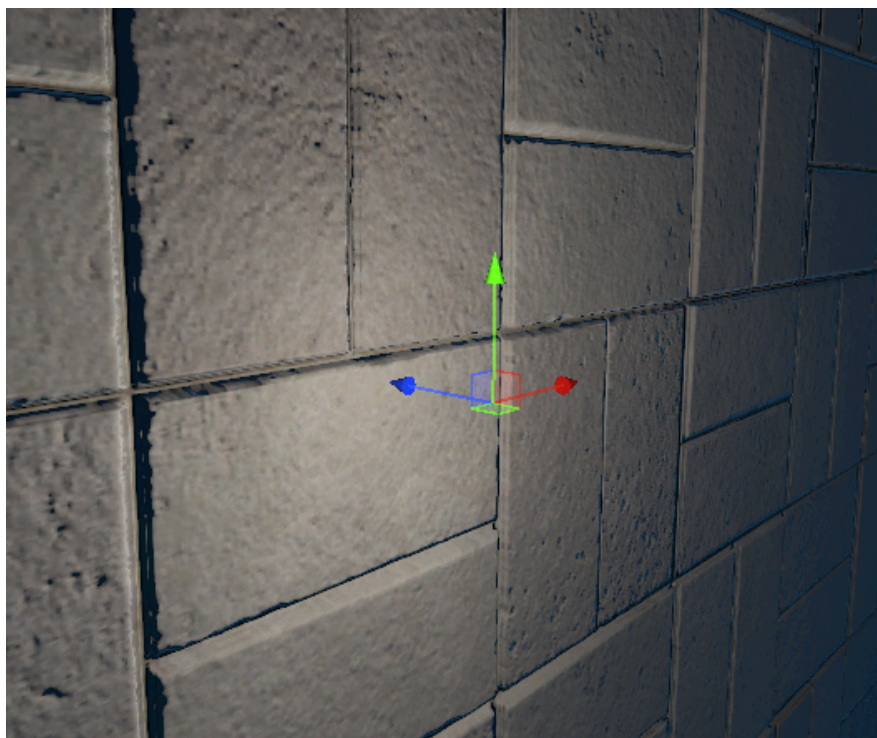


Vamos a explicar las más importantes, que son las siguientes:

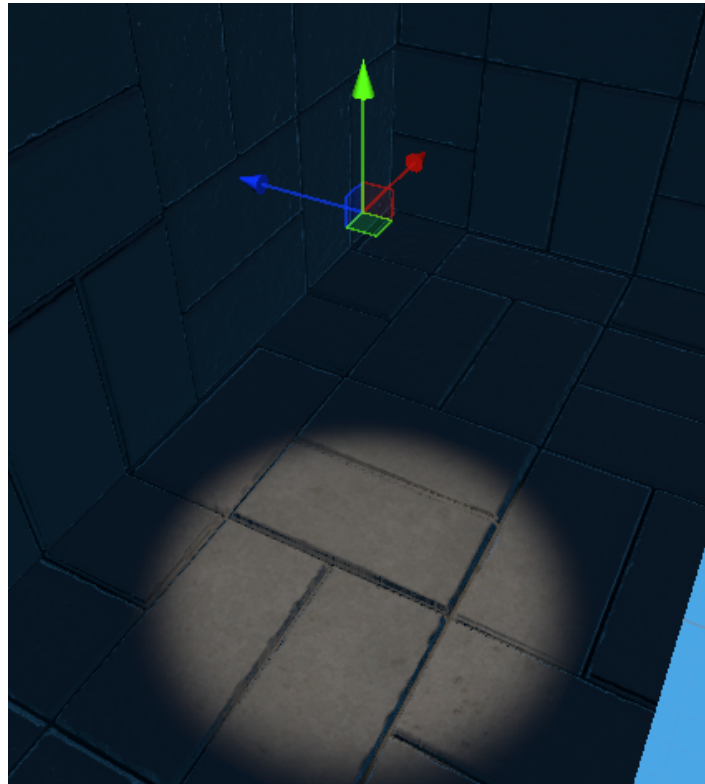
Directional Light: Emite luz en una dirección, funciona como el sol de la escena. Cuando creamos una escena nueva siempre vamos a ver una **Directional Light** y una **Main Camera**.



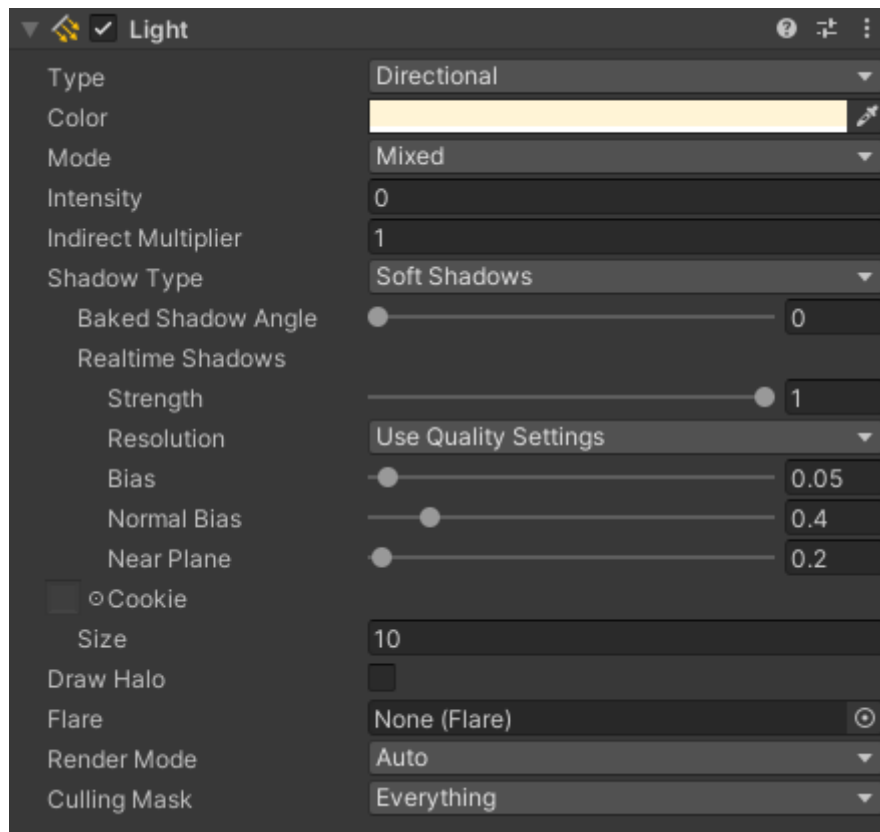
Point Light: Emite luz desde un punto en todas las direcciones. Tiene atenuación con la distancia y produce sombras. No es una luz que emite luz desde el centro, sino que ilumina alrededor desde el centro.



Spotlight: Es un tipo de luz de foco en forma de cono. Esta hace que la luz parta de un punto hacia un lugar en concreto.



Existen otros tipos de luces, pero con estas tres podemos hacer casi todo lo que deseemos en nuestros proyectos. En general, más allá de ciertas diferencias que hay entre los distintos tipos de luces que repasamos, estos comparten una serie de variables que vamos a modificar para conseguir el efecto deseado:



Type: Podemos elegir el tipo de luz antes mencionadas (***Directional, Point, Spotlight, etc***).

Color: Cambia el color de la luz emitida.

Intensity: Probablemente la variable más importante de las luces, podemos elegir que tan potente es la luz emitida.

Ahora vamos a implementar todo esto a nuestra escena, vamos a iluminar a nuestro antojo, hay que ponerse creativos.



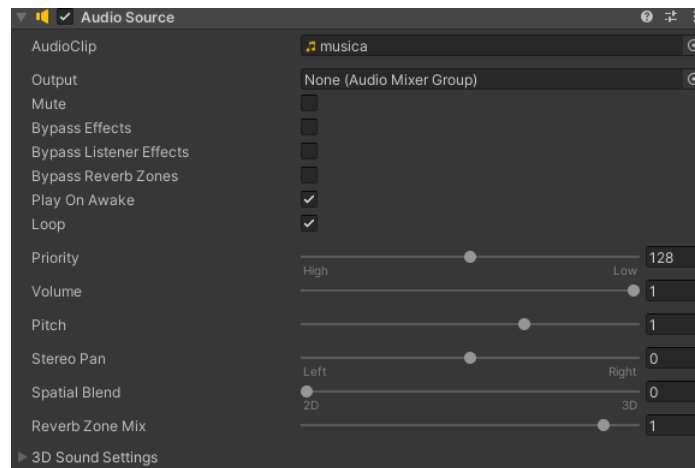
La música y el sonido en videojuegos

Los sonidos en los juegos a veces parecen estar en un segundo plano cuando se trata de diseño, muchas personas lo dan por sentado y no le prestan atención al diseño de sonido ni a la música, lo cual no es algo malo. Un buen diseño de sonido pasa desapercibido porque se siente acorde al mundo que nos propone el juego y sin darnos cuenta nos aporta inmersión y feedback en nuestras acciones.

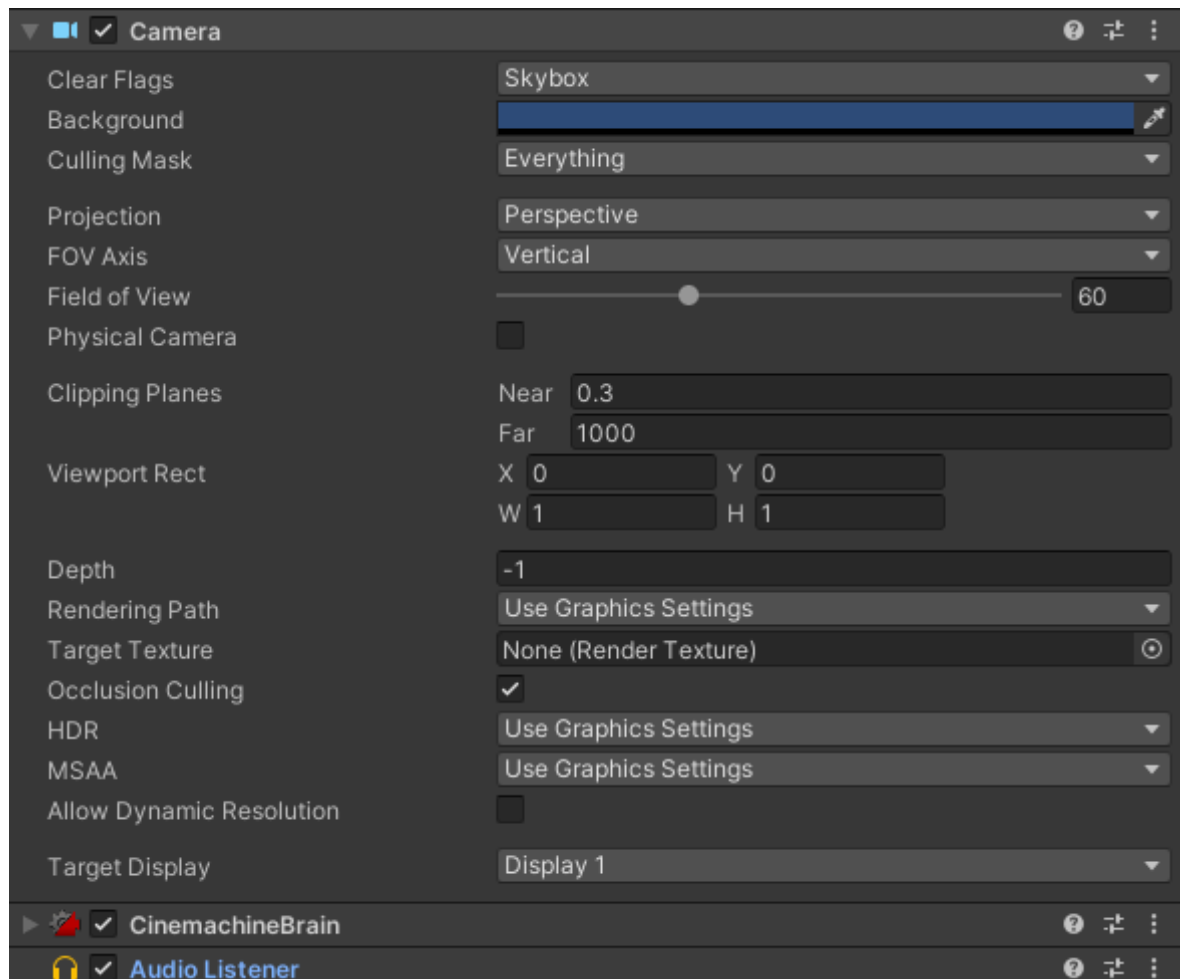
La idea de esta clase es que podamos agregarle unos efectos de sonido y además un poco de música para ambientar esta escena y darnos cuenta del poder de los efectos de sonido.

AudioSource y AudioListeners

Ya sea para implementar música o sonidos, en **Unity** vamos a utilizar un componente que se llama **AudioSource**, es decir, fuente de sonido; el cual posee muchas propiedades para facilitar la implementación de efectos de sonido.



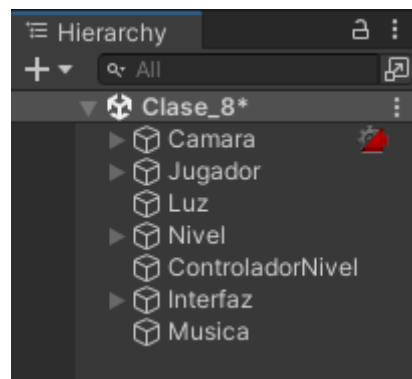
Además de un **AudioSource**, vamos a necesitar un **Listeners**, este es otro componente que va a escuchar el sonido, y le va a dar una espacialidad a lo que escuchamos, este componente no tiene ninguna propiedad, simplemente va a ser el que interprete el sonido del juego para que lo podamos escuchar cuando juguemos. En la mayoría de los casos, la cámara es el que posee el **AudioListener**, en la siguiente imagen se puede ver que la cámara del proyecto ya tiene uno de forma predeterminada:



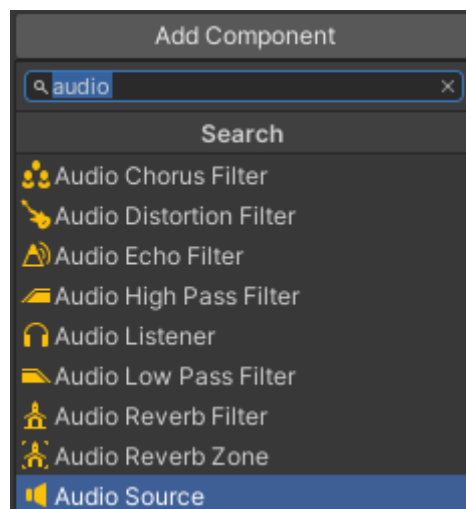
Implementando música

Como siempre aclaramos, existen infinitas formas de implementar recursos y funciones en Unity, si investigan por su cuenta van a aprender otras formas de hacer las cosas, en este curso vamos a aprender a hacerlo de una forma específica, pero no es la forma definitiva.

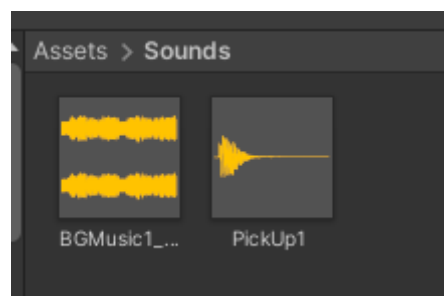
Lo primero que vamos a hacer es un objeto vacío y lo vamos a llamar “Música” (no utilizamos tildes para evitar problemas, ya que el lenguaje de Unity por defecto es el inglés).



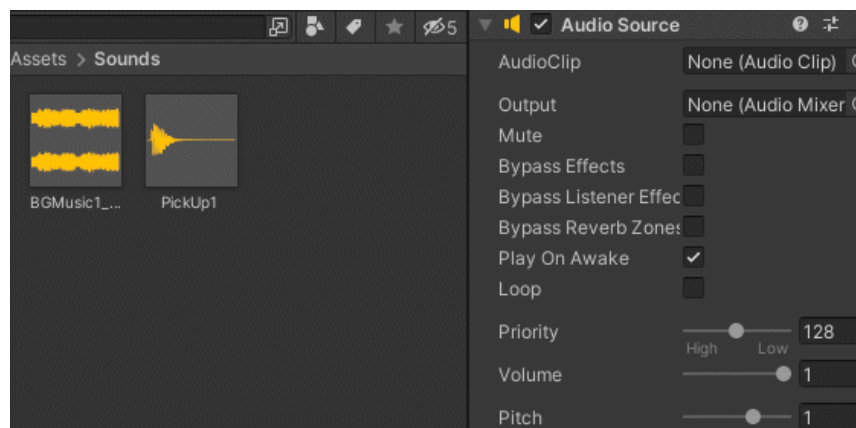
Dentro de este objeto vamos a agregar el **AudioSource**:



¡Excelente! Este objeto se va a encargar de loopear la música, es decir que cuando termine la pieza, la va a volver a repetir infinitamente. Vamos a descargar alguna musica de fondo desde la pagina freesound.org o podemos acceder a nuestra carpeta de [Recursos](#)



Vamos a subir el archivo de sonido arrastrándolo a **Unity** y luego lo pondremos en la variable **"AudioClip"** de nuestro **Audio Source**.



Es muy importante que la variable *Loop* sea verdadera, para que la música se repita infinitamente.

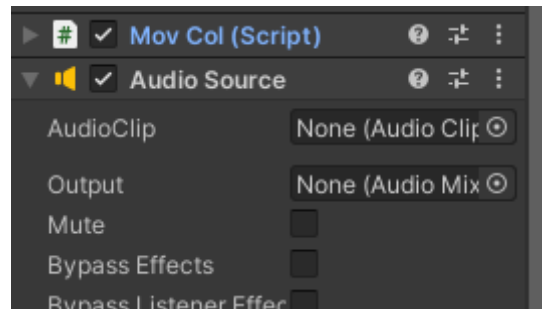


¡Nuestro juego ya tiene música, así de simple!

Los sonidos que elijamos son muy importantes, ya que buscamos generar distintas emociones según el juego. Acá, por ejemplo, creamos un ambiente competitivo utilizando un clásico sonido de Arcades.

Implementando efectos de sonido desde el código

Ahora que nuestro juego tiene música, tenemos que agregarle unos efectos de sonido para darle una sensación más responsiva al movimiento. Vamos a agregar un efecto de sonido cuando saltamos. Para esto, le añadiremos el componente **“Audio Source”** a nuestro personaje y modificaremos nuestro Script para que pueda reproducir audios desde él.



Primero nos aseguramos de que nuestro Script tenga definidas 2 variables muy importantes, el **AudioSource** y una del tipo **AudioClip** que contendrá nuestro sonido a reproducir.

```
private AudioSource aSource;
public AudioClip pickPotion;
```

Obtendremos nuestro audiosource usando el **GetComponent** en nuestra **función start()**.

```
private void Start() {
    aSource = GetComponent<AudioSource>();
}
```

Y ahora crearemos nuestra situación para reproducir el sonido. ¿En qué momento queremos que suene? En este caso, siguiendo la idea de la clase 4, haremos que se reproduzca cuando agarremos nuestro **pickUp** (la poción).

```
private void OnTriggerEnter(Collider other) {
```

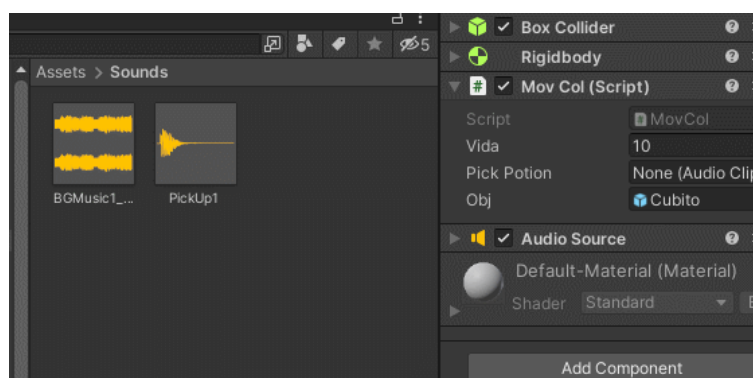
```

if (other.gameObject.CompareTag("PickUp")) {
    aSource.PlayOneShot(pickPotion);
    print("Agarre pota");
    Destroy(other.gameObject);
}
}
    
```

Como podrán ver, estamos usando `OnTriggerEnter`, que deben recordar tildar el **"IsTrigger"** de uno de los 2 objetos a chocar, y volvemos a usar nuestro Tag para diferenciar el objeto.

¡Ya tenemos nuestro código armado! Nos falta decidir qué sonido queremos escuchar. Para esto, podremos ingresar a la página <https://freesound.org/> o ingresar a nuestra carpeta de recursos.

Una vez descargado, lo pondremos en nuestra carpeta de sonidos dentro de Unity y lo colocaremos en nuestra variable usando el inspector

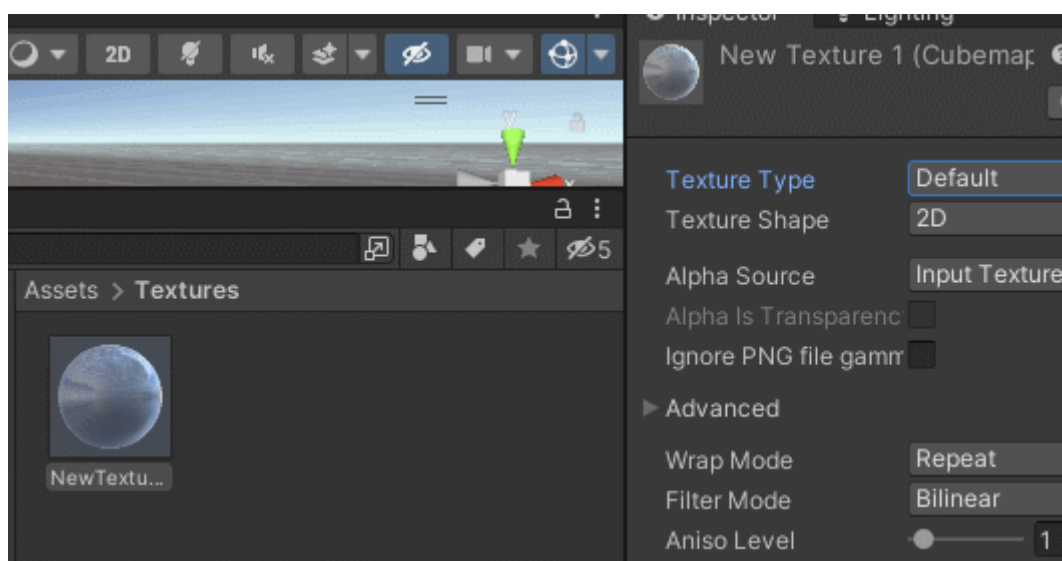


Y listo! Ya tendremos nuestro sonido reproduciéndose cada vez que agarramos un objeto con el **Tag "PickUp"**

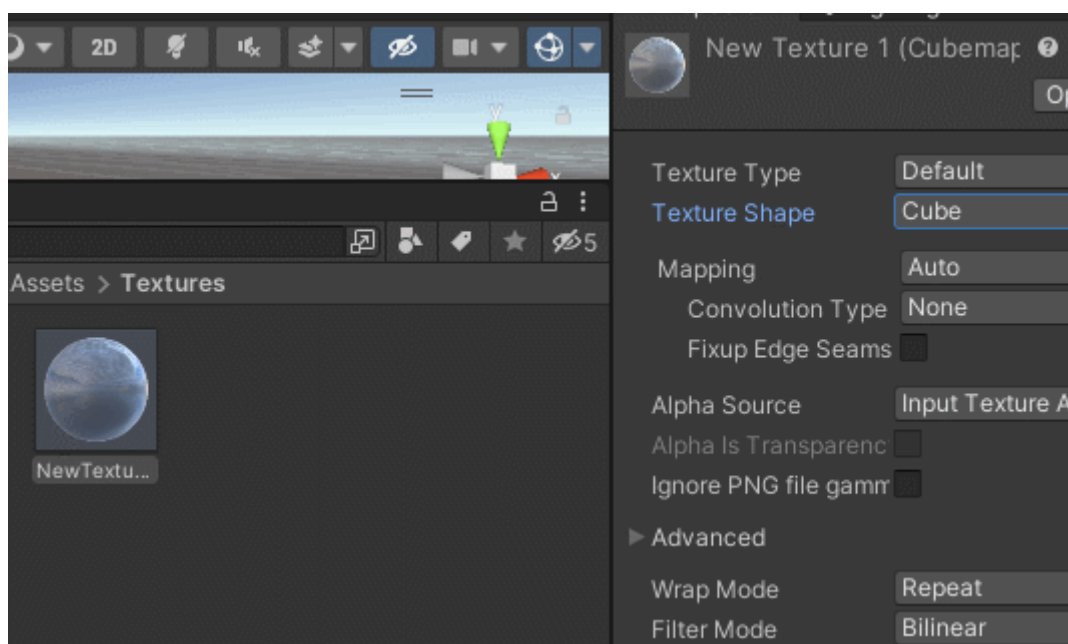
Skybox

Como tema final de nuestra clase, veremos cómo modificar el **Skybox**. Ese fondo en el horizonte que podemos ver todos en nuestro juego. La idea es colocar algo acorde a nuestro nivel y la temática que tratamos. Para eso, podemos ingresar a alguna de estas paginas <https://polyhaven.com/> <https://hdri-haven.com/>, o podemos ingresar nuevamente a nuestra “carpeta de recursos” y descargar el que estaremos utilizando: [Recursos](#)

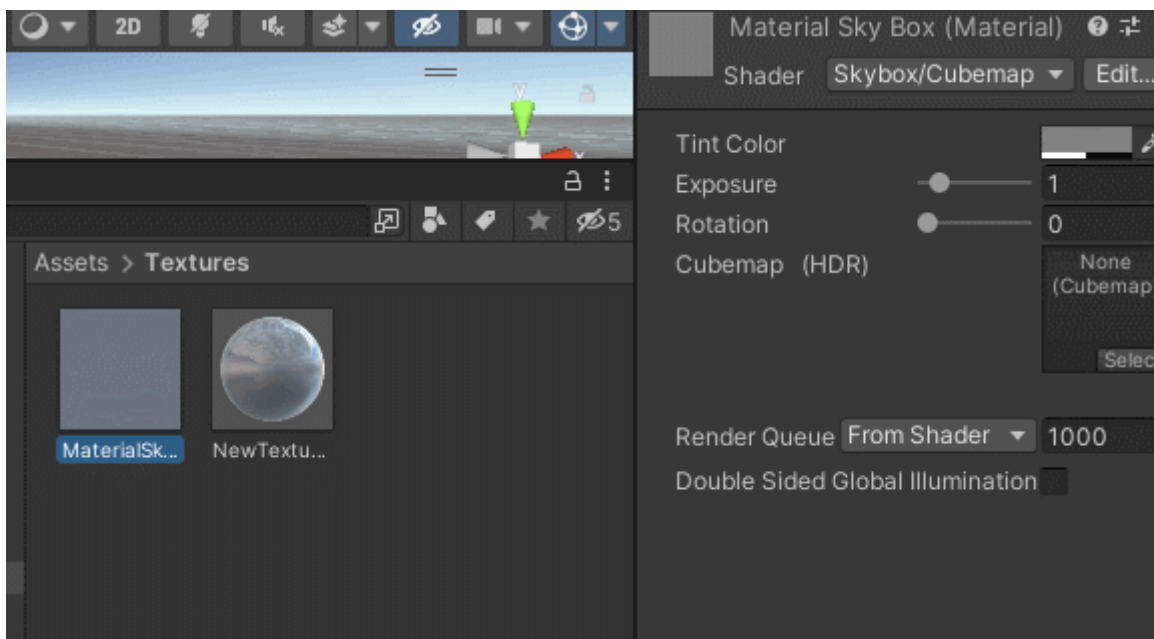
Empezaremos por subir la textura a nuestra carpeta “**Textures**” en **Unity**. Pasaremos a seleccionarla, iremos al inspector y arriba, donde dice texture shape, lo pasaremos de 2D a Cube.



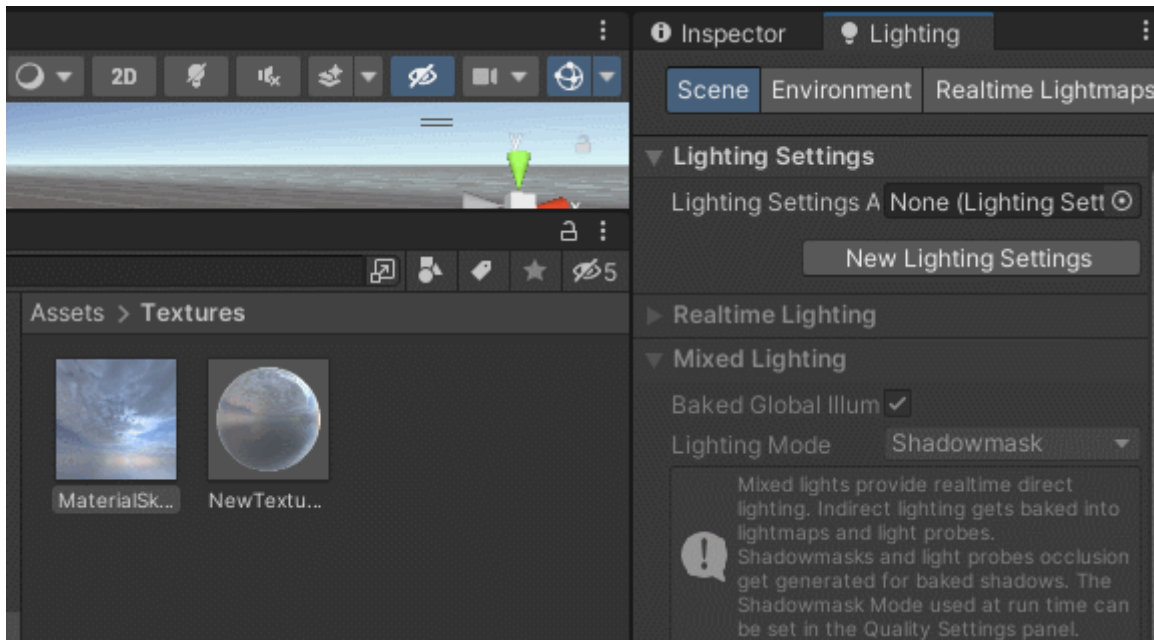
Seguiremos creando un material y en el inspector pasaremos el Shader de Standard a **Skybox** → **CubeMap**.



Una vez hecho esto, deberemos colocar nuestra textura, lo cual lo haremos arrastrándola.



Por último, iremos a la pestaña **“Lightning”**, que se encuentra en la solapa de **“Window”** → **“Rendering”** → **“Lighting”**, iremos a la solapa **“environment”** de la misma, y en la sección de **“Skybox Material”**, arrastraremos el material que hemos creado para esto.



¡Con esto, ya tenemos nuestro nuevo Skybox colocado!

Si todo salió bien en la clase de hoy, deberíamos de tener un nivel iluminado, musicalizado, con feedback en alguna acción y con un bello Skybox.



Desafío N° 8:

Agregale sonido al salto del jugador. Ya le agregamos sonido a los pickUps ¿Cómo se te ocurre que podrías hacerlo utilizando el código visto en clase?

Acordate de seleccionar otro sonido y añadirlo como variable en el código.

No te olvides de **guardar** todos los cambios.

Tomá una **captura de pantalla** del código.

Subilas al espacio correspondiente del Desafío 8.





Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad