

PROGRAMACIÓN 3D

Máster en Programación de Videojuegos

TEMA 10: Unreal.



CENTRO UNIVERSITARIO
DE TECNOLOGÍA Y ARTE DIGITAL

Juan Mira Núñez

Índice

- Introducción
- Render
- Iluminación
- Sombreado
- Shaders
- Billboards
- NormalMap
- Cubemap

Introducción

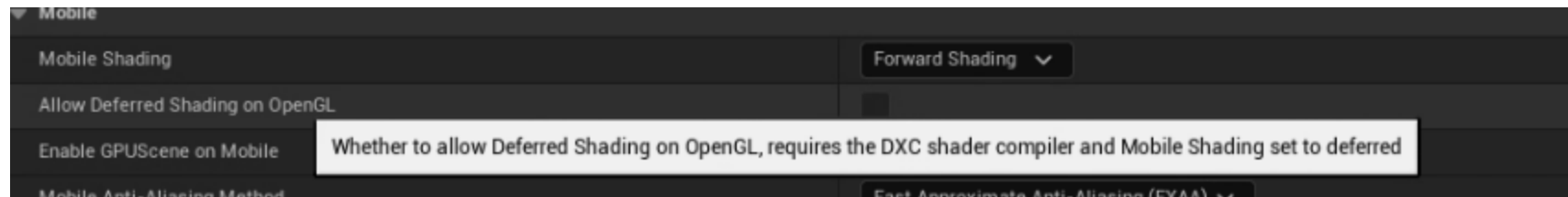
El objetivo principal de este tema es comparar los contenidos que hemos ido viendo a lo largo de la asignatura comparando como serían ciertos contenidos en Unreal Engine.

La mayoría de los módulos que hemos visto en la asignatura suelen ser procesos simples o que requieren poca implementación para tenerlo en unreal ya que suele estar gestionado por el motor.

Así que veremos algunos puntos que cabe destacar del temario dado y conocer sus versiones en unreal y cómo usarlas correctamente.

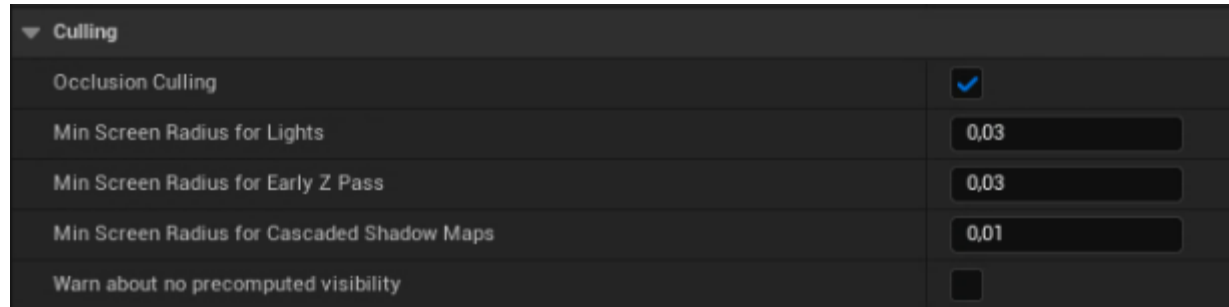
Render

El render normal que utiliza unreal es el Forward shading, pero también tiene soporte con OpenGL si utilizáramos el Deferred shading



Culling

Al igual que gestionamos en OpenGL con el culling a la hora de dibujar geometría también podemos gestionar culling como hacíamos con el Depth test y el scissor test

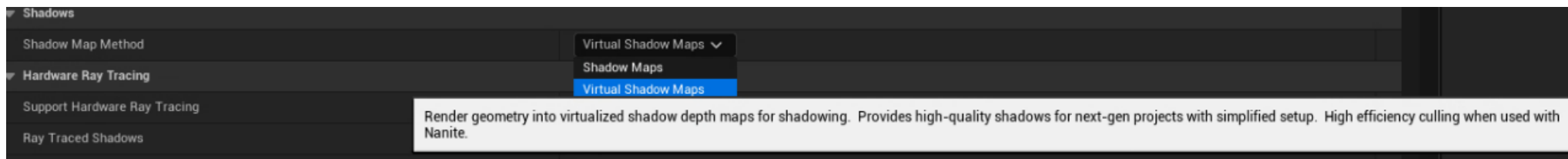
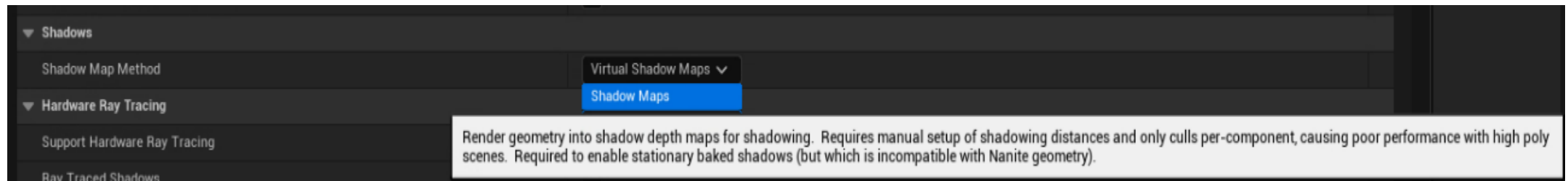


Iluminación

▼ Global Illumination	
Dynamic Global Illumination Method	Lumen ▼
▼ Reflections	
Reflection Method	Lumen ▼
Reflection Capture Resolution	128
Reduce lightmap mixing on smooth surfaces	<input checked="" type="checkbox"/>
Support global clip plane for Planar Reflections	<input type="checkbox"/>
▼ Lumen	
Use Hardware Ray Tracing when available	<input type="checkbox"/>
Ray Lighting Mode	Surface Cache ▼
High Quality Translucency Reflections	<input type="checkbox"/>
Software Ray Tracing Mode	Detail Tracing ▼
Ray Traced Translucent Refractions	<input checked="" type="checkbox"/>

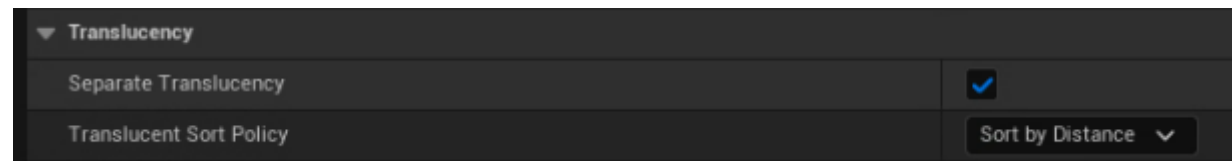
Shadow map

Podemos utilizar tanto un shadow map como un virtual shadow map para ahorrarnos el proceso de configuración enlazado con Nanite es muy eficiente



Transparencia

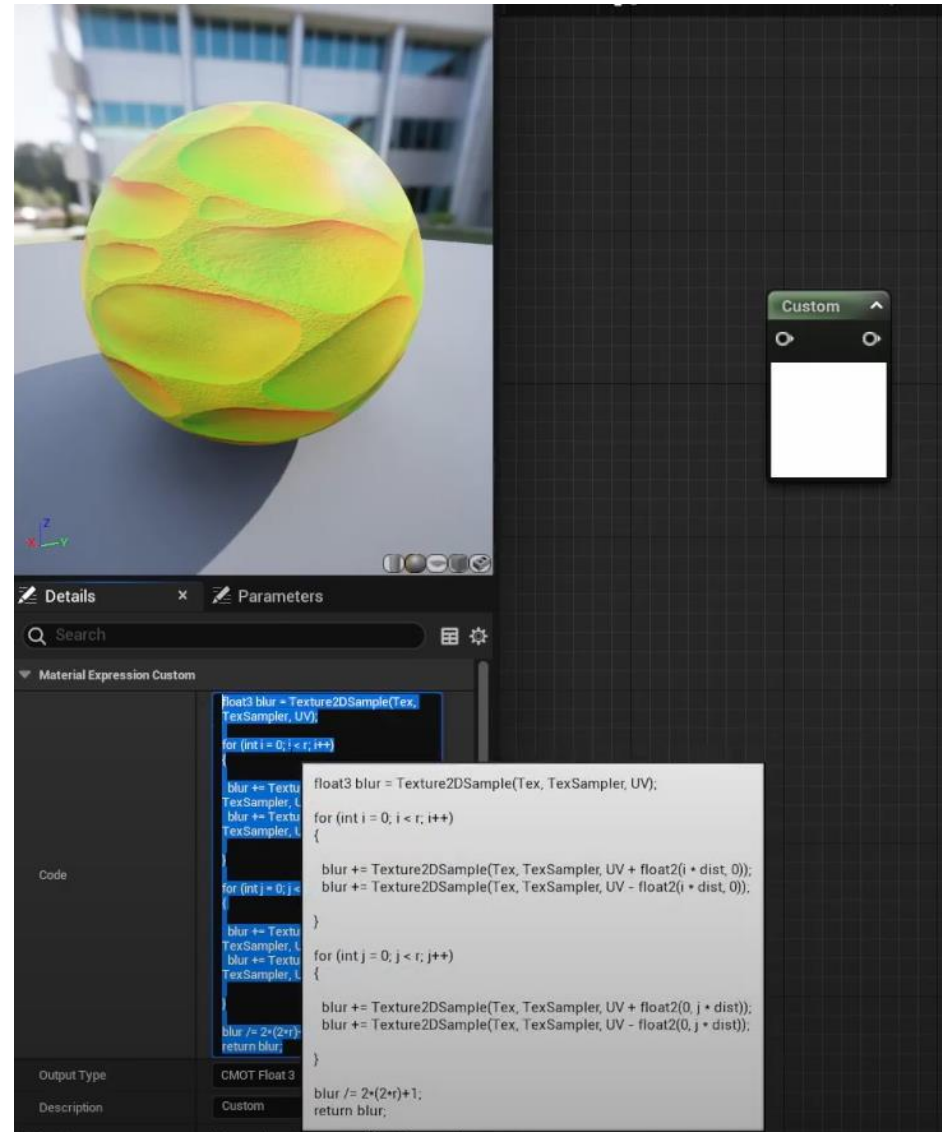
Al igual que gestionamos nosotros desde OpenGL la transparencia en un primer pase de render en el que calculamos el factor de profundidad para cada objeto iluminado, se utiliza la misma técnica.



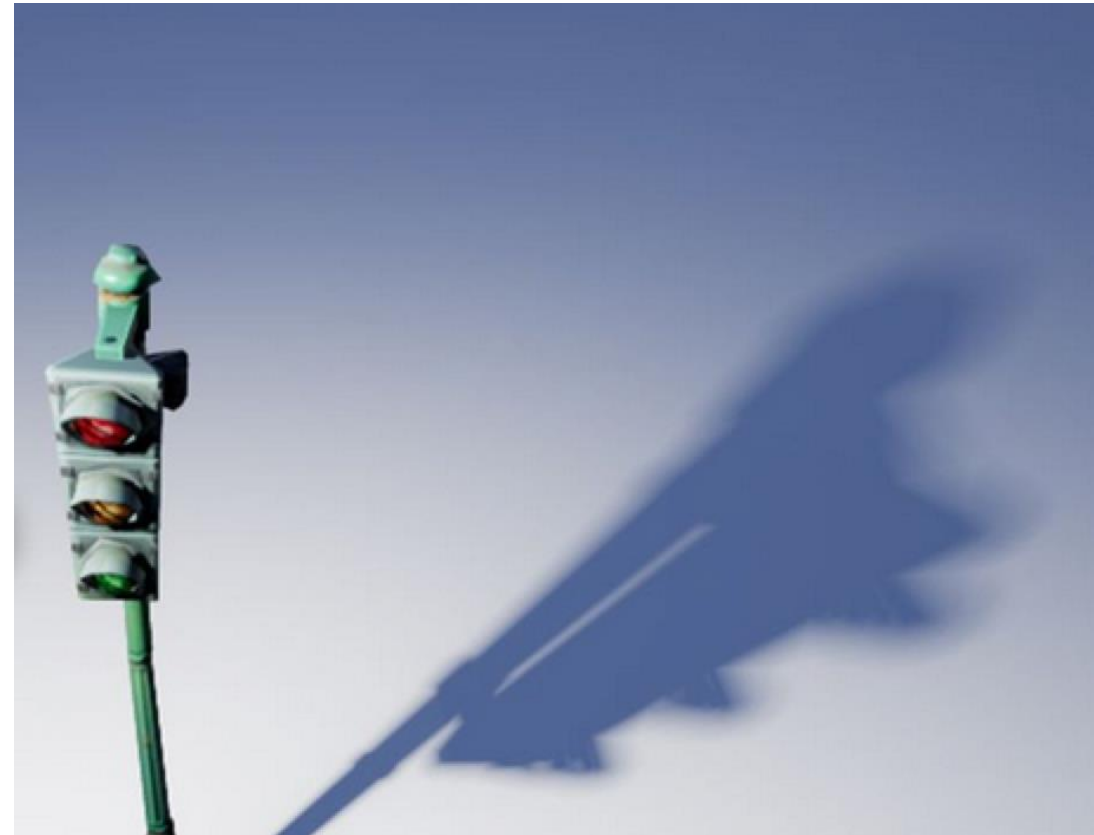
Allow translucency to be rendered to a separate render targeted and composited after depth of field. Prevents translucency from appearing out of focus.

Shader

El editor de materiales nos permite introducir código en HLSL como cálculos para nuestros materiales, algo parecido a tener tu propio shader para un material que tú mismo puedes gestionar.

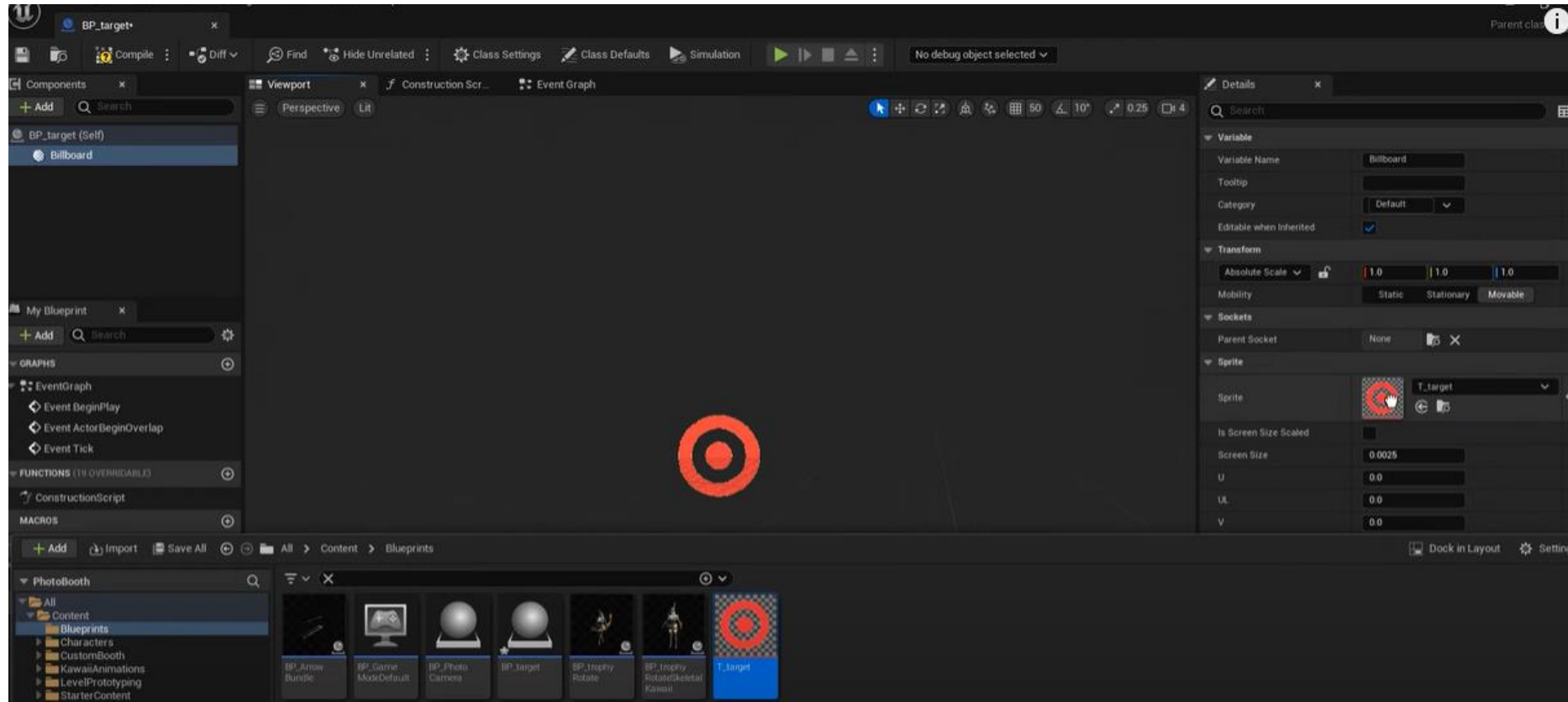


Shadow map

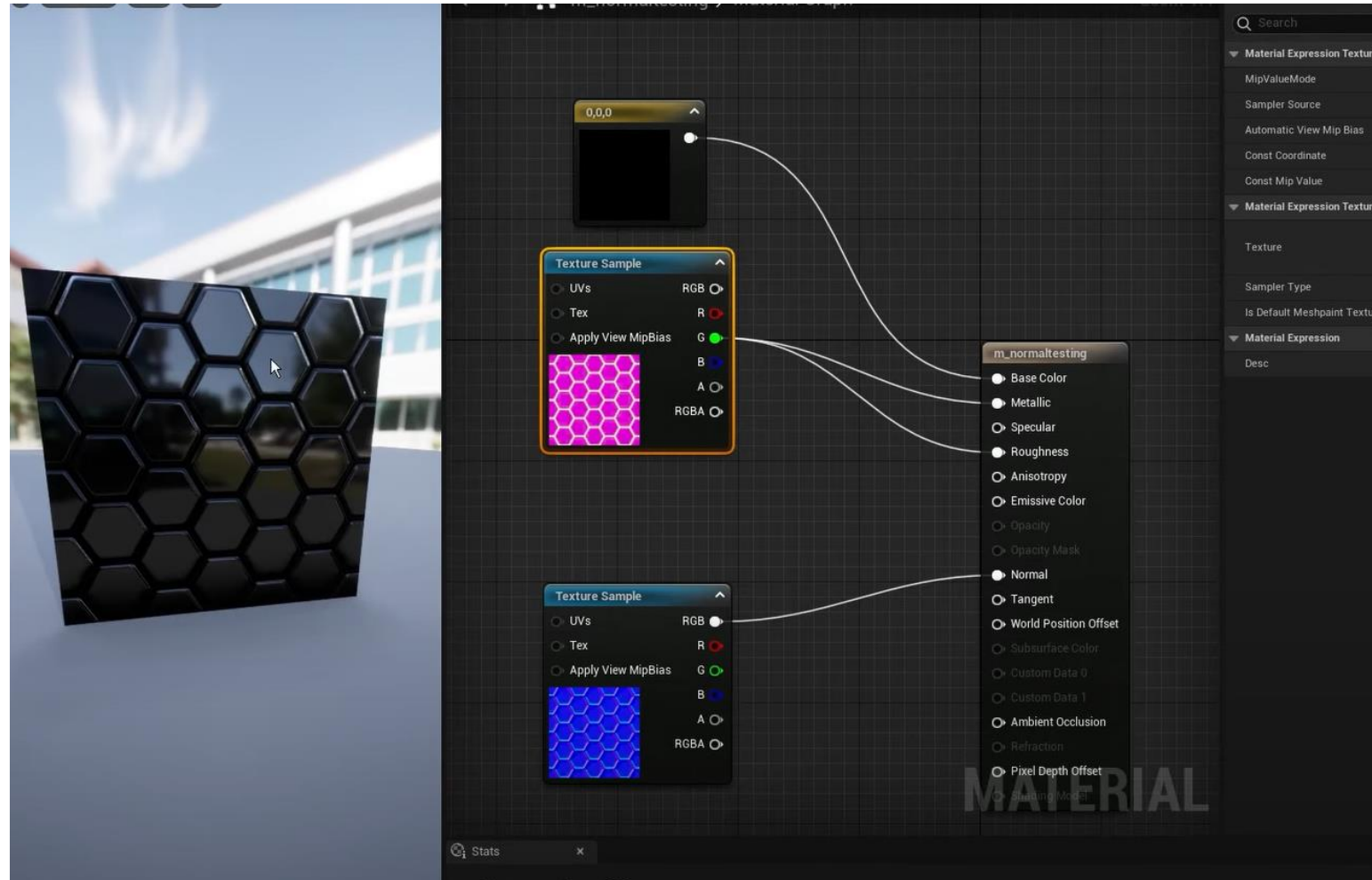


Ray traced distance field shadows

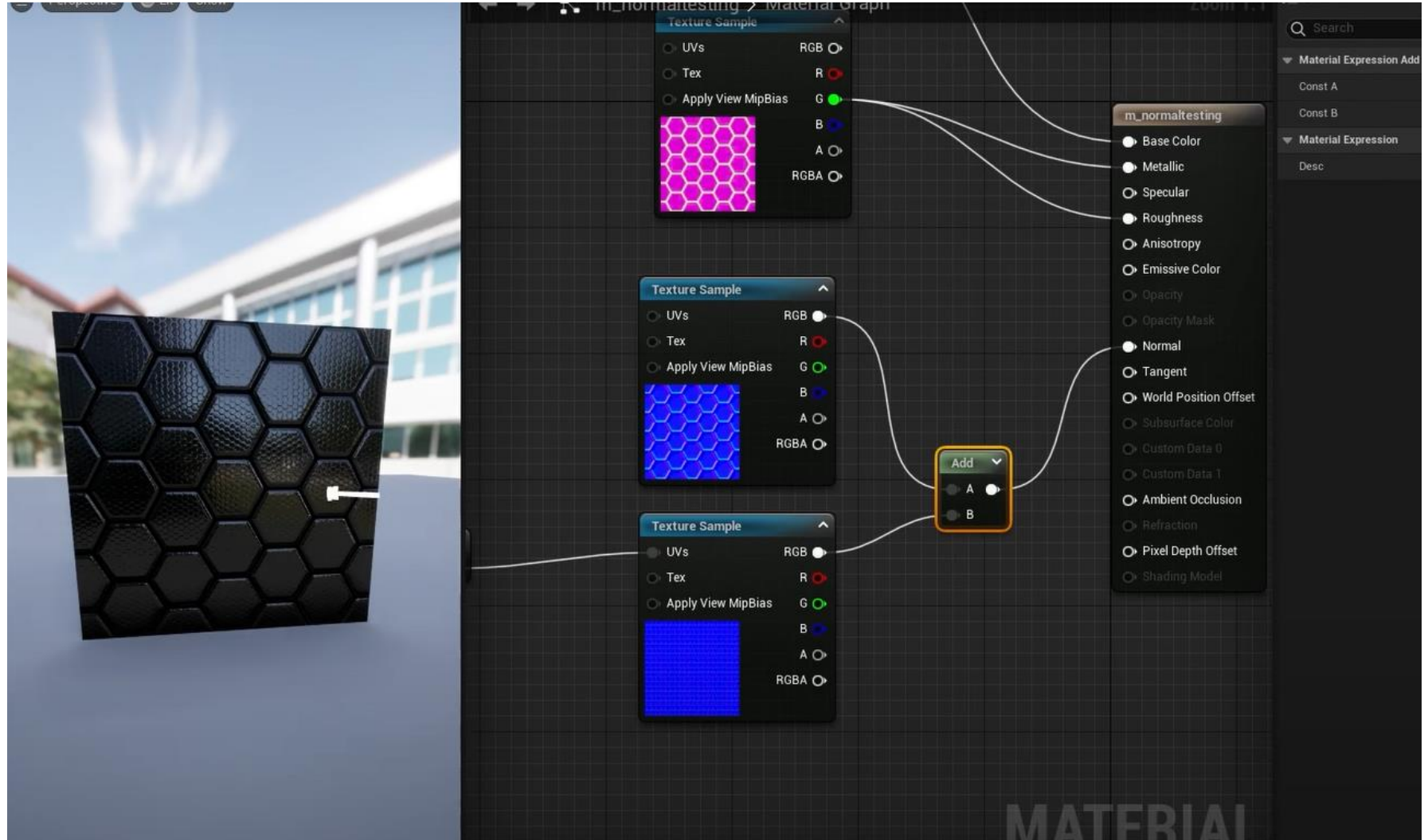
Billboard



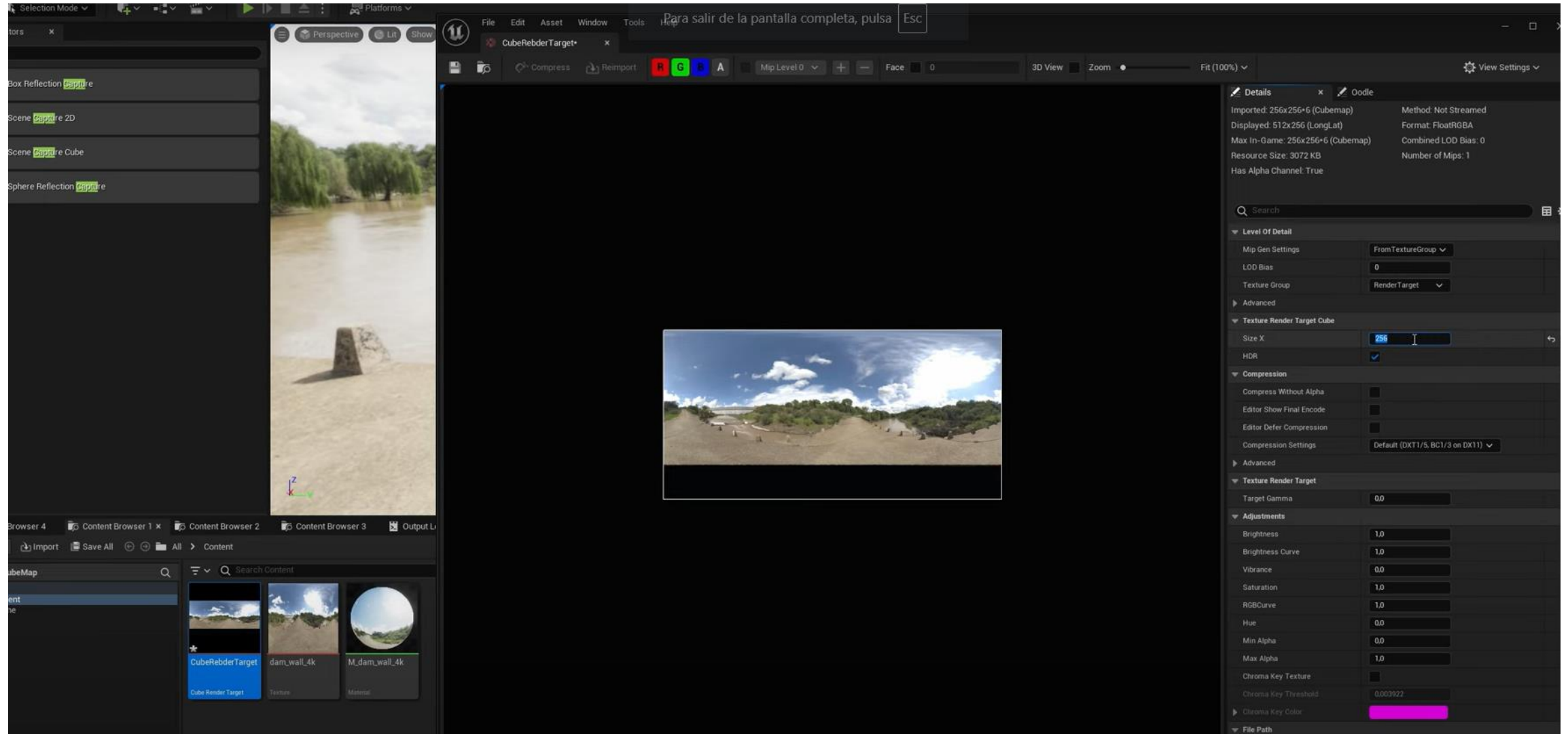
Nomal map



Normal map



Cubemap



¿Dudas?

