Documentación Unreal engine

versión 4.26.2

abril del 2023

CTA

***Vicente Alexandre 6, Bajo. 18360 Huétor Tájar (Granada)***

**ÍNDICE**

[1- Introducción 2](#_Toc133480766)

[1.1 Uso y breve explicación 2](#_Toc133480767)

[1.2 Empezar el proyecto 2](#_Toc133480768)

[1.3 Ajustes 2](#_Toc133480769)

[2- Explicación entorno 2](#_Toc133480770)

[3- La iluminación 3](#_Toc133480771)

[3.1 Movilidad 4](#_Toc133480772)

[3.2 Mejoras e implementaciones 4](#_Toc133480773)

[3.3 Reflejos 5](#_Toc133480774)

[4- Quixel Bridge 5](#_Toc133480775)

[5- Cámaras y secuencias 6](#_Toc133480776)

[6- Terrenos 6](#_Toc133480777)

[7- Materiales 6](#_Toc133480778)

[8- Partículas 8](#_Toc133480779)

[9- Físicas 8](#_Toc133480780)

[10- GPU-Lightmass 9](#_Toc133480781)

[11- Raytrace 9](#_Toc133480782)

[12- Materiales en terreno 10](#_Toc133480783)

[13- Arquitectura 10](#_Toc133480784)

[14- Extra 11](#_Toc133480785)

[14.1 Agua 11](#_Toc133480786)

[14.2 Nubes Volumétricas 12](#_Toc133480787)

[14.3 Migrar Contenido 12](#_Toc133480788)

[15- Optimización 12](#_Toc133480789)

[16- Programación con blueprints 13](#_Toc133480790)

[17- Unreal Engine 5 14](#_Toc133480791)

# 1- Introducción

Unreal Engine es un motor gráfico que se creó para desarrollar juegos, pero ha ido avanzando con el tiempo y actualmente se usa para juegos, cine, arquitectura...

Para su uso necesitaremos descargar Epic Games e instalar Unreal Engine, también precisamos crear una cuenta de Epic, y en el apartado de Unreal Engine necesitaremos descargar una versión.

Para esta documentación nos hemos centrado en la versión 4.26 menos la parte de programación con blueprints que usaremos la 5. No se preocupen que no cambia mucho solo visualmente y un par de cosas que son fácilmente detectables ante cualquier usuario.

## 1.1 Uso y breve explicación

Abrimos el Unreal Engine una vez instalado, para poder empezar con nuestro proyecto también contamos con un Marketplace donde podemos obtener muestras gratis cada x tiempo para usarlas y también tenemos una pestaña de noticias del motor.

## 1.2 Empezar el proyecto

A la hora de empezar nos preguntarán la categoría del proyecto y una vez seleccionada te darán a escoger entre unas plantillas. (Esto ya va a nuestra decisión)

## 1.3 Ajustes

Aunque las configuraciones van al gusto de la persona una recomendación es cambiar el idioma de Unreal Engine a inglés para manejarse mejor con la información de Internet etc.

# 2- Explicación entorno

Hablaremos de actores, que hacen referencia a los elementos que usamos en Unreal.

**- Viewport** -> Escenario que vemos, nos podemos mover con ratón izq. (eje x) ambos click ambos ejes, la rueda apretada es lo mismo que ambos clicks, y evidentemente el click derecho (eje y), también te puedes mover con ASDW pulsando el click del ratón, Q y E te sirve para moverte en vertical, tenemos velocidad de cámara que se puede controlar con la rueda del ratón, la F te lleva al objeto seleccionado en el panel lateral y doble click también,

Contamos con un desplegable que nos deja marcar real time para ver los cambios o no (partículas …), mostrar estadísticas, fps, lo más interesante es el game view para que se oculten los elementos que no deberían verse sin necesidad de compilar, podemos guardar la posición actual para volver después de un solo click o asignarlas con ctrl+1… compactarlos etc.

Sumado a este desplegable tenemos varios botones que nos permiten cambiar la vista de las luces, exposición, mostrar contenido específico y los botones de control para la cámara y los actores que pongamos.

**- World Outliner ->** Pestaña lateral donde podemos seleccionar los actores con los que vamos trabajando ordenarlos en carpetas, buscarlos en assets y un sinfín de opciones más.

Una de las opciones más útiles es que puedes arrastras un objeto encima de otro se van a enlazar y por lo tanto se van a guardar la distancia y se moverán conjuntamente si lo vuelves a hacer pues lo desencadenas.

Y también podemos “agrupar” que te crea un objeto grupo con todo lo seleccionado (se pueden ir seleccionando con shift o control) y de este grupo podemos modificar los elementos de forma individual o grupal.

Una vez hablado de la parte de arriba también tenemos una parte abajo la cuál tiene todas las características modificables de los actores seleccionados y la principal área de trabajo.

**- Content Browser ->** Es la carpeta donde guardamos todas las cosas que usamos en el proyecto o que vamos importando en él.

Es importante hacer jerarquía de carpetas principalmente para que no sea un caos encontrar algo, de formas hay varias opciones de filtrado, también se pueden personalizar las carpetas, coloreándolas, renombrándolas… se puede filtrar por texturas, blueprints, materiales, etc.

La parte potente es que podemos tener abiertos varios entornos desde **window > content browser** y también tenemos varias opciones de vista de los elementos con el botón de abajo a la derecha por si queremos más información de estos o una vista previa.

**- Place Actors ->** Siendo esta parte una de las principales ya que desde aquí vamos buscando y añadiendo los actores principales a nuestro proyecto.

# 3- La iluminación

Existen 5 tipos de luces y casi todas ellas tienen las mismas propiedades, aunque todas tienen cosas diferentes:

**- Directional Light ->** Simula la luz del sol, da igual donde se ponga que la luz siempre será la misma a no ser que se rote su actor, cambia si se modifica su inclinación.

**- Point Light ->** Similar a la luz de una bombilla emite sombras y luz 360º.

**- Spot Light ->** Simula la luz de un foco.

**- Rect Light ->** Es una luz que parte de un plano.

**- Sky Light ->** Genera una luz de ambiente, para su uso es necesario tener también una **“atmospheric fog”** que es un entorno.

## 3.1 Movilidad

Existen 3 tipos de movilidad de las luces:

**- Static ->** Se debe calcular y no es dinámica osease no se puede cambiar ni mover, consume muy poco y da muy buena calidad, aunque tarda en calcularse.

**- Stationary ->** Son una mezcla de los otros dos tipos, genera sombras suaves y genera los rebotes de la luz y las sombras son dinámicas, osease, si movemos otro elemento que tenga influencia con esta luz generará sombra en tiempo real.

**- Movable ->** No se debe calcular, consume más y no tiene rebotes es luz plana. Se suele usar para eventos puntuales.

## 3.2 Mejoras e implementaciones

Para mejorar las luces en un área usamos el **“Lightmass Importance Volume”** todo aquello dentro de este espacio se va a calcular con la mejor de las posibilidades en términos de luces, reflejos y sombras.

Para forzar la entrada de luz por un lugar, usamos los **“Lightmass Portal”**

Importante para generar los lightmass, que son las capas que se pintan sobre los actores y simulan la luz, se debe buildear el proyecto y debe de existir en esos actores una capa UV que esté destinada a la luz, esto se puede ver haciendo doble click sobre el actor/elemento en el content browser, una vez en la nueva ventana pulsamos sobre UV y una de esas capas debe tener el **“Destination Lightmass Index”** en 1.

A la hora de importar un elemento al proyecto se puede seleccionar una casilla que te crea esta capa UV de manera automática, debemos buscar el **“Generate Lightmap UV”** y dejarlo marcado.

Debemos tener en cuenta que para modificar las características de Lightmass debemos acceder a la pestaña de **“World Settings”** si no está activa la podemos mostrar desde Window > World Settings. Hay que tener cuidado con lo que tocamos ya que incrementa el tiempo de render una barbaridad.

Si queremos que x o x’s elemento/s tenga mejor calidad también podemos cambiar su lightmass resolution para esto podemos cambiarlo en la vista al abrir dicho elemento o forzar un cambio al seleccionarlo y navegando por sus propiedades activamos el **“Override Light Map Res”** y podemos cambiar su valor.

Hay un campo en el **“World Settings”** llamado **“Compress Lightmaps”** se suele usar en arquitecturay me parecía importante citarlo, si lo desmarcamos va a tardar mucho más ya que las texturas son 4 veces más pequeñas evitando el ruido lumínico de algunos muros.

Se pueden implementar IES para modificar un elemento con esa nueva textura/forma, de igual forma se pueden cambiar las proyecciones de las luces.

## 3.3 Reflejos

Los reflejos funcionan de manera especial y es que solo reflejan aquello que en la pantalla se ve, asique para crear un reflejo realista debemos hacer capturas del entorno, para ello podemos usar una **“Sphere Reflection Capture”** pero debemos ser especialmente cuidadosos con su rango de efecto y no poner muchas juntas ya que los reflejos se superpondrán y esto consume nuestro ordenador y no es irónico.

Desde Project Settings podemos cambiar la configuración de los reflejos y aumentar su calidad, pero cuidado con la memoria.

Para planos (espejos de pared…) que tengan que tener una buena calidad podemos usar un **“Planar Reflection”** en vez de la esfera. Para usarlos debemos activar su uso desde el Project Settings. Para mejorar el consumo al usar un plano debemos desactivar las casillas de capturar cada frame y capturar en movimiento.

[Esto es para que el ordenador no explote].

# 4- Quixel Bridge

Nos podemos descargar la aplicación siguiendo los pasos que nos dan en la página de Quixel Bridge y al ser usuarios de Epic y trabajar con Unreal disponemos de un montón de contenido gratuito para su uso, es tan sencillo como descargar y una vez descargado importar cada elemento y automáticamente te lo añade al proyecto que tenga abierto.

Una vez importado puedes usarlo como quieras, hay que tener en cuenta una cosilla, a la hora de descargar hay que ver dese las opciones de descarga que vamos a traernos LODs algunas veces vienen todos marcados cuando con uno tenemos de sobra.

# 5- Cámaras y secuencias

Hay dos tipos de cámaras las normales y las de cine, las cámaras son de las cosas más importantes, estas cámaras tienen de por sí mismas un postprocesado donde podemos jugar con las escenas todo lo que queramos, sumado a la serie de opciones y características que ya disponen, (la de cine más que la normal como es obvio).

Se pueden colocar varias y tener varios procesados diferentes para cada una de ellas y se pueden tomar las screenshots de la cámara dando a elegir si queremos solo la imagen o también sus componentes como buffers.

Para crear una secuencia debemos añadir un nivel desde **“cinematics > add level sequence”** seleccionamos donde lo vamos a guardar y te crea una vista para gestionar la secuencia. Ahora procedemos a crear una cámara y trackearla, una vez tengamos la cámara esto se convierte en programación de animaciones y transiciones y manejo de programa de edición, cosas que debemos ir probando y probando de nuevo hasta conseguir lo que queremos como queremos.

Las partículas con movimiento tienen un pequeño gran secreto -> debemos acceder a su material, luego una vez abierto el material seleccionamos el parent, y si escribimos **“dof”** en el buscador lo deshabilitamos, así los cambios en la cámara como el enfoque… se aplicarán también a las partículas.

Las secuencias se pueden exportar a video o imágenes de los frames.

# 6- Terrenos

Los podemos crear desde **“Moldes > Landscape”** o **shift+F2**, lo más importante de saber principalmente es que preferimos tener más componentes, aunque ocupen el mismo tamaño.

Podemos moldear los terrenos como queramos y disponemos de varias herramientas con las que trabajar, debemos tener cuidado con la dureza del “pincel”, esta hace que las herramientas funcionen de diferente forma.

# 7- Materiales

A modo curiosidad -> el motor se encarga de comprimir las texturas asique no las tenemos que pasar comprimidas

Y aunque unas pesan más que otras finalmente para el motor pesarán lo mismo.



Las texturas van en base 2 -> 256, 518, 1024, 2048, 4096, 8192 ...

Es bueno saber que hay una opción desde las texturas para escalarlas a otro tamaño menor reduciendo su peso considerablemente.

Crear un material es simple, dando click derecho en el **“Content Browser”** luego damos un nombre para el material y procedemos al doble click para abrir el editor del material con sus nodos. Y de aquí pasamos a dar valores y unir nodos probando y probando hasta conseguir el material que queramos.

Finalmente lo guardamos y ya dispondremos de este material para usarlos donde queramos.

Recomendaciones -> Activar las opciones **“Connectors” / ”Live Preview” / ”Live Nodes” / ”Live Update”.**

Ojo 👀 algunas de estas características de los nodos solo aceptan unos valores predeterminados, cuidado con los valores que les damos.

Vamos a resumir lo más básico:

* **“Base Color”** es el color que le vamos a dar al material.
* **“Metalic”** es un parámetro (número entre 0 y 1).
* **“Roughness”** es un parámetro (número entre 0 y 1 ambos incluidos), podemos pasarle una textura.
* Y unas cuantas más como **“Specular”**, **“Emissive Color”**, **“Normal”**, **“Tangent”**, **“Ambient Occlusion” …**

**Atajos importantes:**

* Podemos dar al **1** (no al numérico) y click para crear una constante de 1
* Podemos dar al **2** (no al numérico) y click para crear una constante de 2
* Podemos dar al **3** (no al numérico) y click para crear una constante de 3
* Podemos dar al **4** (no al numérico) y click para crear una constante de 4
* Podemos dar a la **“l”** y click para crear un **“lerp”** en base a un alpha (normalmente de alpha no se puede pasar un RGB recomendado pasar solamente un canal) mezcla dos valores.
* Podemos dar a la **“m”** y click para crear un **“multiply”** multiplicación de toda la vida.
* Y podemos seguir así hasta la saciedad…
* No cuenta como atajo, pero podemos arrastrar desde el **“Content Browser”** hacía la ventana del material texturas y automáticamente te la pondrá como un nodo.

**Normales ->** Las normales que aplican el relieve al material pueden llegar a ser bastante fuertes y hay dos maneras de controlarlas:

* Manual -> Se modifica el rojo y el verde sin tocar el azul y luego se juntan con el nodo **“append”**
* God version -> Se usa el nodo **“FlattenNormal”** y solo debemos jugar con un valor entre 0 y 1

El resto de explicaciones se hayan en un proyecto dentro de los materiales creados viendo sus nodos todo está comentado y es altamente entendible, el proyecto se llama **“Materiales”.**

A partir de un material podemos crear instancias del mismo, pero debemos de conectarlo con el material maestro y dentro de este material maestro, convertir en parámetros las texturas y campos que ponemos (se suelen poner contenidos que vienen por defecto en el material maestro y ya luego se añaden en la interfaz los valores que se deseen). Se pueden dar grupos y prioridad a cada parámetro para organizar la vista de recogida de datos.

# 8- Partículas

Podemos crear partículas al igual que creamos materiales, recomendable crearse una carpeta para estas, click derecho sobre esta carpeta del **“Browser Content”** y creamos la partícula con **“Particle System”** y de nuevo tenemos nuestro editordisponible para configurarla, lo principal es saber que necesitamos un material al cual le podemos decir que coja los colores que le damos en el editor a lo largo de la vida de la partícula. Esto es gracias a un nodo en el material llamado **“Particle Color”** puesto en **“Emissive Color”**.

Por defecto las partículas solo se ven cuando el emisor se ve en pantalla si queremos que se vean más a menudo debemos ir al menú superior de la partícula y en **“Bounds”** darle un tamaño más grande para ampliar el rango de visión del emisor.

Se puede usar **“Niagara”** que debemos activar, para ello vamos a **Editar > Plugins > Buscamos niagara en el buscador** y lo activamos. Es un sistema más moderno para personalizar la partícula similar a los nodos, pero a fin de cuentas y a efectos simples ambos dan resultados similares.

Para crear una partícula debemos dar click derecho en el **“Content Browser” > FX** y ya seleccionamos según nuestras conveniencias.

# 9- Físicas

Para empezar nuestro apartado de físicas debemos asegurarnos de que en **Project Settings > Constants > Default Gravity Z = -980.0** ya que es la fuerza del planeta tierra.

La mayoría de actores/objetos tienen unas físicas establecidas que podemos modificar, pero están desactivadas, para activarlas debemos ir al apartado de **Physics** y marcar el **“Simulate Physics”**.

Para crear físicas vamos de nuevo al **“Content Browser” > click derecho > “Physics” > “Physical Material”**.

Ahora podemos agregar este material, desde las opciones de los actores buscando la opción de override o accediendo con doble click al actor y arrastrando el material recién creado a **“Phys Material”**.

**Actores derivados de las físicas:**

* **Constraint ->** Funciona como pegamento entre elementos (debemos seleccionar los elementos que une, son padre e hijo y ponerlo físicamente entre ambos)
* **Thruster ->** Propulsor (como un cohete), se debe activar el Auto Active y tenemos que unirlo con el actor, además de poner la fuerza con relación a la masa del actor.
* **Radial Force ->** Simula un “agujero negro” que aplica una fuerza en un campo.

# 10- GPU-Lightmass

Esta parte depende de tu gráfica por temas de compatibilidad y precisa de mucho cuidado a la hora de su uso ya que equivocarnos en algo y darle al build puede hacer que se nos caliente de más el ordenador. También debemos saber que esto está a partir de la versión 4.26 las anteriores necesitan una ayudita externa que debemos buscar por Internet si queremos usarlas (es un lío su uso ya que debemos buscar los archivos de configuración en la carpeta y editar un txt cada vez que queramos cambiar algún parámetro).

Aun así, para usarlo y poder toquetear varias cosas debemos, primero activar el **“Ray Trace”** al crear el proyecto, después en **“Project Settings”** buscamos **“Ray Tracing”** para comprobar si está activo, seguimos con la pestaña de **“Plugins”,** ahora, seleccionamos **“Built-In”** en la barra lateral y buscamos **“GPU-Lightmass”** y activamos.

Ahora ya desde el build de siempre para compilar el proyecto podemos seleccionar el **“gpu-lightmass”**. Y podremos manejar una nueva ventana de configuración que ya trae un build desde donde compilaremos ahora.

Para poder observar los cambios y cálculos debemos asegurarnos de que en “Project Settings” tenemos activado el

**“Enable virtual texture support”** y **“Enable virtual texture lightmass”**.

# 11- Raytrace

Esta es posiblemente la explicación más simple de todas a no ser que tengas un pc de la nasa no uses Raytrace salvo para detalles puntuales y mínimos.

Es una herramienta potentísima ya que nos permite hacer que las luces sean móviles y se calcule la luz en tiempo real cosa que para videos/películas está muy bien, pero de nuevo consume mucho y hay que usarlo con mucho cuidado. Básicamente calcula todo recogiendo información de los rayos que lanza en diversas direcciones, podemos jugar con el número de rayos que mandamos para aumentar la calidad de sombras reflejos y luces pero es probable que quememos algún ordenador si no sabemos lo que hacemos.

Debemos activarlos al crear el proyecto o buscándolos en las **“Project Settings”**, básicamente se usa desde las opciones de los actores/objetos no tenemos un menú especial ni nada, solamente lo vamos buscando en las diferentes pestañas como **“Raytracing”** o **“Light”**. Así mismo, debemos buscar **“rh”** para ver que tenemos activado o en su defecto, activar, el **Direct 11 & 12** y en el desplegable poner **DirectX12**.

Donde sí podemos ver más cosas es desde el **volumen de post procesado “Post Process Volume”** y su mayor uso es para el **“Ambient Occlusion”**, que no consume tanto.

Se usa para cosas muy específicas porque consume mucho y ralentiza bastante.

A la hora de usarlo para la refracción de un material es el valor de **“Specular”**.

# 12- Materiales en terreno

Para usar los materiales en terrenos para hacer un **“Landscape”** debemos crear un **“Material Function”** a esta altura supongo que no tenemos problema con ello, usaremos **“MakeMaterialAttributes”** y para usar estos materiales debemos crear un material base, ¿qué es un material base?, bueno pues es un material normal pero que se le añaden los nodos **“Layer Blend” y** **“BreakMaterialAttributes”**.

Arrastraremos los **“Material Function”** creados anteriormente en este material base, los añadiremos al “**Layer Blend”** y posteriormente uniremos los parámetros que usemos.

Es recomendable cambiar el nombre de cada material que metamos en el **“Layer Blend”**.

Este material base lo aplicamos al landscape y en **“Modes”** seleccionamos landscape para abrir el editor con su pincel etc. Aquí deben aparecer como Layers los materiales que agregamos anteriormente, para usarlos le damos en cada material al icono del **+** y crearemos una capa normal y le damos un nombre.

Así podremos pintar de diferentes formas con estos materiales generando terreno rápidamente.

Para poner elementos en el paisaje podemos ir de nuevo a **“Modes”** y seleccionar **“Foliage”**, aquí podremos arrastrar los actores/objetos para usarlos cuando y como queremos, se sigue usando el pincel con diferentes parámetros como el de densidad…

Consejo: Si pulsamos Shift borramos los elementos que tengamos seleccionados y que estén dentro del pincel.

# 13- Arquitectura

Para los proyectos de arquitectura se recomienda tener todos los elementos en el programa 3D en pequeñas partes ya que a la hora de calcular con Unreal es mejor tener varias partes pequeñas con una resolución más pequeña que una muy grande con una resolución enorme para que se vea medianamente bien siendo la segunda opción menos eficiente.

Lo más importante de esta parte para nosotros va a ser el cargar los FBX, y ¿cómo se hace esto?, fácil.

Arrastramos los archivos dentro de una carpeta en nuestro proyecto de Unreal en el **“Content Browser”** o click derecho importar.

Usualmente cuando importamos quitamos la opción de generar colisiones y la de generar materiales y activamos la opción de generar UV Lightmaps, adicionalmente al arrastrar todos los elementos al área de trabajo les damos las coordenadas (0,0,0) para que todo se vaya colocando en su posición.

Podemos importar al proyecto usando un plugin llamado **“Datasmith Importer”** pero debemos activarlo, para ello desde **Settings** nos vamos a plugin y en “**Built In”** lo buscamos, activamos. Al reinicia debería aparecer un nuevo botón que nos permite importar los **“.udatasmith”**, aunque es preferible trabajar con FBX.

Una vez tenemos todo lo necesario importado, para estos proyectos se suelen traer los materiales maestros de otros proyectos, esto se puede hacer siempre y cuando la versión sea la misma, pero si cambia la versión del proyecto no podremos exportarlos y debemos crearlo de nuevo a mano.

Se puede decir que ahora nos enfocaremos en crear luces, vegetación, materiales etc, esto es como siempre hasta dejar el proyecto como nos resulte atractivo, adicionalmente podemos configurar las colisiones con los objetos por si queremos crear un entorno donde permitamos el movimiento y exploración de cada rincón.

Algo que debemos saber es como exportar nuestro proyecto para hacerlo un ejecutable, antes de hacer esto vamos a contemplar varias opciones para el empaquetado, en **“Project Settings”** podemos cambiar tanto como queramos, pero nos centraremos en **“Maps & Modes”** donde podemos establecer los niveles por defecto.

Llegamos a la parte importante **“Packaging”** aquí podemos modificar las opciones de empaquetado básico como el “Shipping” para indicar si es algo terminado para mayor calidad y el **“Full Rebuild”** para que se calcule todo al abrir.

Se puede hacer una programación básica de blueprints para que al dar al escape o la tecla q se cierre, esto es arrastrando 3 nodos que se hayan por su nombre, no hay pérdida, la otra forma y por defecto, es hacer alt+f4.

# 14- Extra

Ahora veremos algunas explicaciones extra, para usar en nuestros proyectos:

## 14.1 Agua

Vamos a usar el plugin del agua, activamos el plugin del agua **Plugins > Water** y **Plugin > LandMass**.

Lo principal es crear un Landscape, y marcamos la opción de **“Edit Layers”**, ahora podremos buscar water en el buscador de **“Place Actors”** y podremos ver las diferentes opciones que trae el water. Estos actores se pueden ampliar y reducir pinchando en las bolitas blancas que los delimitan y con alt pulsado creamos otras bolitas y ampliamos o achicamos.

Cada una trae unas propiedades diferentes y otras compartidas, por ejemplo, si usamos el ocean tenemos una propiedad customizable de las olas, el material, la profundidad, la curva, el filo de las olas, material, etc.

## 14.2 Nubes Volumétricas

Al igual que con el agua debemos activar un plugin, este es llamado **“Volumetrics”**, ahora en nuestro Content Browser debemos marcar las opciones de vista de **“Show Engine Content”** y **“Show Plugin Content”**.

Debemos buscar **“Volumetric Cloud”** en los actores de efectos visuales, con tan solo arrastrar y tenemos unas nubes establecidas, dentro de las propiedades de este actor podemos modificar algunas propiedades, pero no tantas como queremos, para ello buscamos en el **“Content Browser”** **>** **“Volumetrics Content” >** “**Tools” > “CloudCompositing” > “Blueprints”** Estos dos blueprints que tenemos aquí los arrastramos a nuestro entorno y los metemos en una carpeta para tenerlo en un sitio que sepamos acceder rápidamente y también metemos el **“Volumetric Cloud”**.

Ahora vamos con un paso importante, de nuevo en nuestro “Content Browser” buscamos la carpeta **“Volumetrics Content” >** “**Content” > “Sky” > “Materials”** de aquí vamos a establecer como material del **“Volumetric Cloud”** el material **“M\_VolumetricsCloud\_03\_Profile\_Billowy”** se puede y es recomendable crear una instancia de este material y así podemos modificar las propiedades de este material más fácilmente.

En nuestro **“BP\_CloudMask\_Object”** (el icono grande) podremos escalarlo y nos daremos cuenta que ha generado una nube volumétrica, podemos modificar sus parámetros para generar finalmente las nubes que queramos usualmente se modifican los parámetros del **“Noise”**.

Seguidamente en nuestro actor **“Sky Atmosphere”** (se suele usar este en vez del **“Atmospheric Fog”**).

## 14.3 Migrar Contenido

Para migrar contenido es tan simple como seleccionar el elemento hacer **“Ctrl+B”** para tenerlo localizado en el “Content Browser” seguidamente **click derecho > “Asset Actions” > “Migrate”**.

Migramos a la carpeta **Content** del otro proyecto.

Se puede migrar todo, incluso niveles enteros. También se puede copiar y pegar desde el explorador de Windows, esto no lo copia todo, pero puede servir.

Siempre se puede migrar de atrás hacia adelante para tener en cuenta las versiones. Nunca de una versión posterior a una anterior.

# 15- Optimización

Para optimizar el proyecto podemos aplicar unos LODs a los objetos, por ejemplo, tenemos una silla, entramos en sus detalles haciendo doble click desde el **“Content Browser”**, y en la pestaña de Lods vamos a darle un número de Lods para que los genere automáticamente (cuanto más alto el número más cambios implementa y más genera), supongamos que le damos un 3 a este número a nuestra silla, y aplicamos los cambios, pues el motor habrá generado 3 LODs que reducen la cantidad de triángulos en función a la proximidad del objeto, cuanto más cerca más calidad y cuanto más lejos menos calidad y menor gasto de recursos.

Este método suele funcionar tan bien que los cambios casi siempre son inapreciables, pero se reduce el consumo muchísimo, podemos modificar los parámetros para forzar menor, mayor calidad, la distancia de aplicación etc.

Se puede crear un LOD manualmente, pero tardas más en configurarlo que en generarlo por el motor.

**“Cull Distance Volume”** este actor es básicamente un cubo que aplicamos en la zona deseada y podemos generar en sus detalles/ajustes tantos “Cull Distance” como queramos y ¿cómo funciona?, pues esto toma dos valores, el tamaño del objeto y la distancia, así los objetos dentro del área que cumplan con lo expuesto desaparecerán o se mostrarán dependiendo de la distancia y su tamaño como hemos explicado anteriormente.

# 16- Programación con blueprints

Los blueprints son un sistema de programación visual por nodos, con los que podemos agregar funcionalidad a nuestro proyecto. Usaremos la versión 5 de Unreal Engine en esta parte (puede ser que en otras versiones algunas cosas varíen un poco).

Podemos abrir un level blueprint que afecta a todo el nivel, o a objetos singulares, o podemos hacer click derecho en el content browser y creamos un blueprint básico, al dar doble click entramos en el editor del blueprint desde donde podemos crear variables, actores …

Desde **“event graph”** podemos acceder a los nodos, desde aquí programaremos lo que necesitemos de este blueprint.

Las variables, espacios en memoria que almacenan un valor, podemos crear las variables que precisemos con el tipo de dato más adecuado a lo que necesitemos. Las variables tienen get (traer el valor) y set (establecer el valor).

Si dejamos marcado el ojito de las variables las hacemos públicas y las podremos cambiar desde las características del blueprint.

Para tener variables en diferentes blueprints debemos usar una instancia.

**Crear nuestros propios nodos de activación al hacer x cosa:**

Desde **“Project Settings” > “Engine” > “Input”** podemos configurar un **“action mapping”** escribimos un nombre identificativo y asignamos tecla, botón de mando o lo que queramos, una vez realizado lo anterior, desde el diagrama del blueprint, podemos hacer click derecho y buscar por el nombre que le dimos, crearemos un nodo y ya lo podemos usar libremente.

**Nodos interesantes:**

**-Timeline** (hacemos que “x” acción transcurra en un “y” tiempo y podemos hacer los cambios que queramos en la gráfica, esta gráfica se saca creando una float track “f\*”)

**-Lerp** (va ligado al Timeline para sacar el valor “alpha” en cada momento)

**-Simulate Physics** (simula las físicas del objeto)

**-Branch** (dividimos lo que pasa si se cumple una condición o no)

**-FlipFlop** (dividimos la ejecución en dos partes, para que cada vez que le demos a la tecla haga una u otra cosa)

**-Add** (incrementar valor, puede ser cualquier tipo de variable lo que se incrementa menos las booleanas)

**-Get all actors of class** (podemos seleccionar el tipo de actor/blueprint y nos trae todos lo que hay en el nivel)

**-Foreach** (por cada elemento de un array te devuelve una serie de valores)

**Array** -> El concepto del array es importante saberlo para ahorrar tiempo de nuestra vida, un array es una tabla con todo lo que nosotros pongamos ordenados en posiciones que comienzan en la 0 hasta el infinito…

# 17- Unreal Engine 5

Los principales cambios que vamos a citar son, **la interfaz** ha cambiado pero la navegación por los menús y características siguen siendo los mismos, el principal cambio es que el tan famoso **Content Browser** por defecto está oculto y para mostrarlo debemos hacer **“ctrl+espacio”**, podemos darle al **“lock in layout”** para dejarlo fijo.

Si por algún motivo estamos muy tristes por que ya no es como el anterior, no vamos a tener que llorar, ya que, los desarrolladores han dejado una opción desde **Window > Load Layout > UE4 Classic Layout**, así tendremos todo tal cuál estaba antes sin ningún tipo de problema.

Ya nos viene por defecto **Quixel Bridge** desde el motor gráfico instalado para que solo tengamos que ir a Content y abrirlo. Podemos arrastrar y nos lo importa y mete automático super rápido.

Otra cosa nueva es **Nanite**, convierte geometría según la distancia del objeto a la cámara, esto reduce la carga y permite incluir un montón de elementos sin que se ralentice mucho.

¿Como se establece la geometría con Nanite?, bien pues desde nuestro querido Content Browser de nuevo seleccionamos la geometría que queramos y click derecho, aparecerá Nanite y lo activamos. Ya está de nuevo disponemos de nuestros frames estables ya que el motor solo se encarga de cambiar la geometría para tener menos polígonos.

Ojo que si queremos podemos importar directamente desde Quixel Bridge como Nanite, el único problema es que resulta más pesado.

Con respecto al **“sky light”** hay una opción de real time que permite hacer cambios a elementos como el **Sky Atmosphere** y que se cambie como su nombre indica en tiempo real.

Debemos activar desde el **Project Settings > “Global Illumination” > Lumen**.