**ÍNDICE**

[1. IDENTIFICACIÓN Y OBJETO DEL PROYECTO 4](#_Toc119396588)

[2. AGENTES 4](#_Toc119396589)

[3. INFORMACIÓN PREVIA 5](#_Toc119396590)

[3.1. ANTECEDENTES Y CONDICIONANTES DE PARTIDA 5](#_Toc119396591)

[3.2. EMPLAZAMIENTO 6](#_Toc119396592)

[3.3. ENTORNO FÍSICO 7](#_Toc119396593)

[3.4. CUMPLIMIENTO DE LAS CONDICIONES URBANÍSTICAS 9](#_Toc119396594)

[4. DESCRIPCIÓN DEL PROYECTO 9](#_Toc119396595)

[4.1. DESCRIPCIÓN GENERAL DEL EDIFICIO 9](#_Toc119396596)

[4.2. PROGRAMA DE NECESIDADES 10](#_Toc119396597)

[4.3. USO CARACTERÍSTICO DEL EDIFICIO 10](#_Toc119396598)

[4.4. RELACIÓN CON EL ENTORNO 10](#_Toc119396599)

[4.5. DISPOSICIONES LEGALES Y NORMAS DE APLICACIÓN. 10](#_Toc119396600)

[4.6. DESCRIPCIÓN DE LA GEOMETRÍA DEL EDIFICIO, VOLUMEN, SUPERFICIES ÚTILES Y CONSTRUIDAS, ACCESOS Y EVACUACIÓN. 71](#_Toc119396601)

[4.7. DESCRIPCIÓN GENERAL DE LOS PARÁMETROS QUE DETERMINAN LAS PREVISIONES TÉCNICAS A CONSIDERAR EN EL PROYECTO. 71](#_Toc119396602)

[4.7.1. DECLARACIÓN RESPONSABLE SOBRE LAS CIRCUNSTANCIAS Y NORMATIVAS URBANÍSTICAS DE APLICACIÓN 73](#_Toc119396603)

[4.8. SOFTWARE UTILIZADO 74](#_Toc119396604)

[4.9. DEFINICIONES Y ABREVIATURAS 74](#_Toc119396605)

***1 Introducción ->***

Unreal Engine es un motor gráfico que se creó para desarrollar juegos, pero ha ido avanzando tanto con el tiempo que 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.

***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 No se->***