**Apartado 1: descripción 🡪 historia del motor y descripción general**

Unreal Engine es un motor de juego de PC y consolas creados por la compañía Epic Games. En sus inicios se utilizó para implementar el shooter llamado “Unreal” en el 1998, posteriormente a soportado algunas entregas de juegos conocidos como Unreal Tournament, BioShok, Star Wars, Batman etc.

El entorno está escrito íntegramente en C++ y utiliza distintas librerías gráficas para abarcar cualquier temática del área de videojuegos. Su primer motor tiene versiones para Windows bajo DirectX, GNU/Linux y Macintosh bajo OpenGL, PlayStation2 y Dreamcast. A lo largo de su historia ha ido evolucionando, desarrollando el entorno para adaptarlo a OpenGL y DirectX9/10, posteriormente se desarrolló para poder programar para casi todas las consolas Xbox, Playstation 3, Game Cube y todos los sistemas de pc, Windows XP/Vista/7, Unix/Linux/Mac.

A continuación vamos a desarrollar brevemente la historia del entorno.

La primera versión de Unreal Engine se desarrolló en 1998, sus características principales integraban renderizado, detección de colisiones, IA, visibilidad, opciones para redes y manipulación de archivos de sistema en un motor bastante completo.

La siguiente versión apareció en 2002 con el desarrollo de “America’s army”. Se reestructuro todo el entorno, tanto a nivel del núcleo como el código completo y el motor de renderizado. Además, se incluyeron sistemas de física y soporte para las primeras consolas.

Dicha versión se actualizó a la 2.5 que mejoró el rendimiento y añadió física para vehículos y la edición del sistema de partículas. También se mejoró de cara a Xbox y se añadieron efectos de sonido EAX 3.0

En el 2006 apareció la tercera versión de Unreal Engine, preparado para DirectX 9-10, Xbox 360 y PlayStation 3. Para ello se añadió soporte para técnicas avanzadas de HDRR, mapping y sombras dinámicas. También se añadieron herramientas complementarias sobre las de anteriores versiones, modificando el sistema de física.

Esta versión sufrió varias adaptaciones y mejoras por distintos motivos. Se mejoró de cara a consolas como la Wii, se añadió el kit para desarrolladores amateur y se añadieron simuladores de construcción, conducción etc.

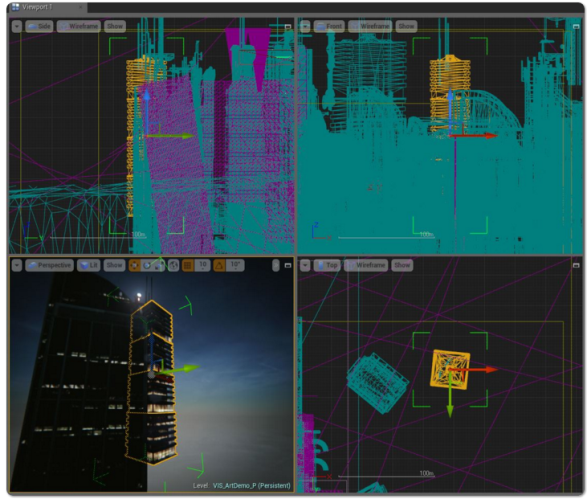
**Apartado 2: categorías:**

1. **DISEÑO**

La estructura del entorno es completamente idéntica a la versión del UE3, tiene un gestor de contenidos que permite realizar los diferentes niveles y tiene distintos visores para navegar por las distintas creaciones y programaciones dentro de un videojuego. Dado que luego hay categorías más concretas, en este apartado vamos a examinar aquellos referentes a las categorías de interfaz y materiales.

* 1. **Interfaz**

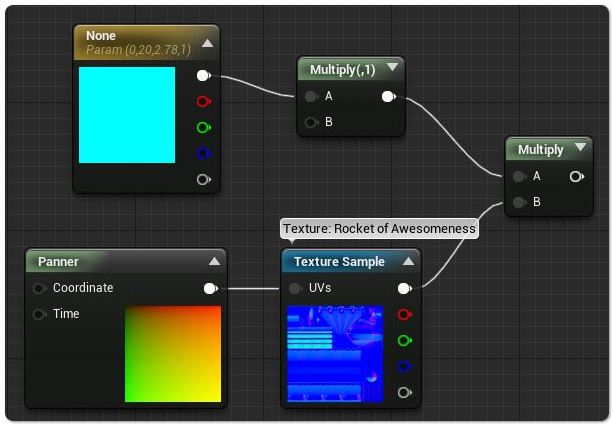
**Editor de niveles**, esta parte del entorno permite construir los niveles y vistas. En él se modifican principalmente la colocación, la transformación y la edición de las propiedades de los actores. Permite definir el mundo a través de objetos geométricos ya sean luces, mallas o personajes. Reduciéndolo al proceso más básico, esta construcción consiste en colocar los objetos en el mapa y programar su comportamiento e integración.



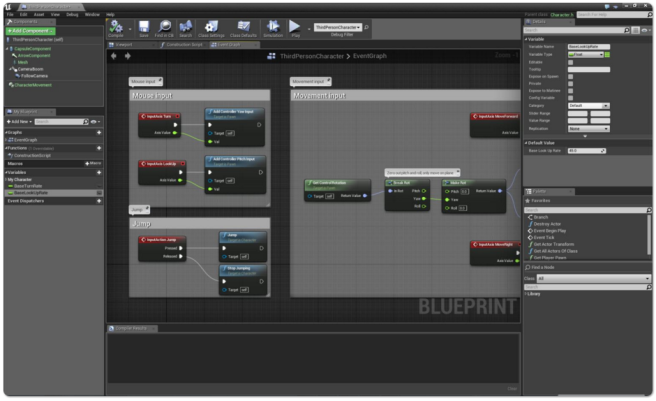


Además, en este gestor de contenidos se pueden crear carpetas y subcarpetas que permitan organizar el contenido del juego (texturas, modelos, sonidos, blueprints, animaciones, esqueletos, etc.). En estas carpetas se nos permite cambiar el nombre o mover cualquier asset de sitio y el engine recalculará las dependencias en base a la nueva ruta, haciendo mucho más fácil el desarrollo. También resulta indispensable de cara a optimizar la carga y descarga de contenido en las etapas finales del desarrollo y poder reorganizar todos los recursos la carpeta “prototipo”.

* 1. **Recursos predefinidos**

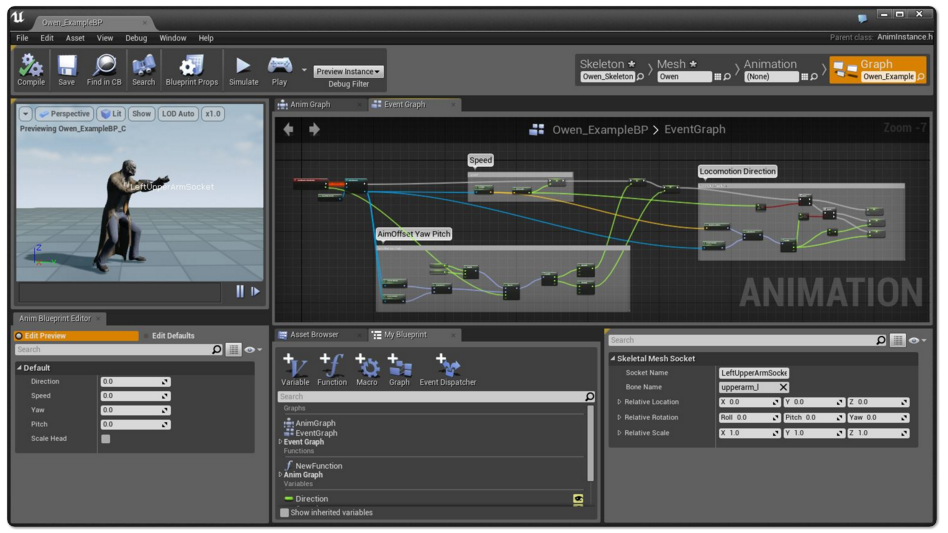
**Editor de materiales**, permite crear y editar los materiales que componen la escena del videojuego. Estos se definen como assets que pueden aplicarse a los objetos y terrenos, por ejemplo se puede crear el material “suciedad” y aplicarlo a los escenarios.

**Editor Blueprint**, es donde se puede trabajar y modificar planos especiales, estos se pueden utilizar para crear nuevos tipos de actores y eventos a nivel de guión, sin necesidad de escribirlos utilizando código C ++.

****

**Editor de comportamiento en árbol,** permite crear un script que programe la inteligencia artificial del videojuego mediante un sistema de nodos. Permite crear tareas, decorados, servicios etc.

**Editor de persona**, son las herramientas que permiten editar la animación de los personajes. Para ello se establecen distintas vistas, la de “esqueleto” para controlar el movimiento de sus huevos y articulaciones, la de “malla” para modificar los materiales que recubren el esqueleto. También tenemos la vista de “animación” para hacer los movimientos de los personajes y la vista “gráfica”, donde se editan los planos de animación y las máquinas de estado.



1. **CARACTERÍSTICAS GENERALES**
   1. **Multiplataforma**

**Capacidad de adaptar el mismo desarrollo a distintos entornos**

**Consolas 🡪 kit de desarrollo**

Unreal engine 4 está preparado para cualquier plataforma actual de videojuegos. Se puede desarrollar desde pc tanto en Windows como en Mac y los juegos que se creen en ella pueden ser generados para funcionar, utilizando el mismo código, en iOS, OS X, Playstation 4, Xbox One, Linux, Android y HTML5.

El sistema de blueprints es capaz de detectar sobre que dispositivos se está desarrollando el juego y adaptar los controles para tal, lo cual facilita el desarrollo.

Además, la implementación para iOS soporta las nuevas librerías gráficas para procesadores de 64 bits de Apple, con lo que el rendimiento en los nuevos dispositivos con pantalla retina sin implementar nada nuevo.

Todas las herramientas y complementos son libres de descarga y uso, solo es necesario crear un usuario en la web de Unreal Engine y descargar el entorno para el sistema operativo deseado. Además, si al usuario se le asocia una cuenta de GitHub, nos permitirá acceder a los repositorios privados para poder ver y descargar el código fuente completo en C++ tanto de la herramienta para Windows o Mac como la de los complementos que generan los binarios en los diferentes sistemas operativos.

* 1. **Licencia**

Actualmente la licencia de uso de Unreal Engine 4 es gratüita, es decir, cualquier usuario puede descargarse el entorno con todas las librerías y herramientas disponibles.

Sin embargo, cuando se publica un juego en la tienda, Unreal cobra un 5% de los ingresos brutos, tras los primeros 3000$ obtenidos por el desarrollador, al trimestre. Esto implica que cuando un desarrollador publica un juego con Unreal Engine 4, una vez haya obtenido 3000$, deberá pagar a la empresa que está detrás de unreal un 5% de los ingresos brutos trimestrales.

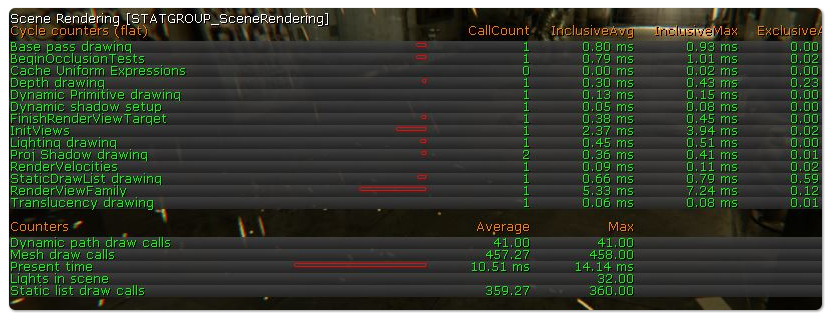
1. **RENDIMIENTO**
   1. **Optimización**

Cualquier videojuego tiene muchos objetosque se ejecutan de manera concurrente y que constan de múltiples características como material, iluminación, colisión, actualización etc. Es necesario optimizar y renderizar el juego lo máximo posible con el fin de que funcione correctamente. Para ello Unreal consta de diversas herramientas de las cuales explicaremos las más destacadas a continuación.

**3.1.1 CPU Profiling**

El rendimiento de la CPU puede verse comprometido cuando hay muchos objetos que tratar, por ello es necesario optimizar los hilos de procesamiento.

Además la CPU debe mandar los comandos de la GPU para tratar los gráficos. Para ayudar al desarrollador a disminuir todos estos posibles problemas Unreal contiene una consola de renderizado donde el desarrollador podrá ver todas las llamadas y características de la escena con el fin de optimizarlos.



**3.1.2 GPU Profiling**

La GPU tiene muchas unidades trabajando en paralelo por lo que se destina cada una a una parte de la estructura distinta. Normalmente los cuellos de botella se producen en el tratamiento del escalado de los elementos y la gestión de las sombras entre otros. Con la consola de GPU se permite al desarrollador ver cual provoca más costes y, por tanto, supone un cuello de botella para el juego ya que atasca la ejecución de otros elementos. Una vez detectados los problemas se pueden modificar los gráficos para conseguir disminuir estos costes de recursos y, por tanto, intentar eliminar los atascos.



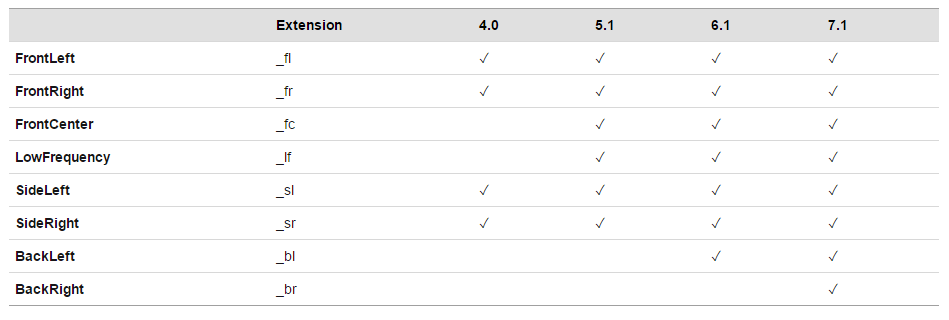
**UTILIDAD**

* 1. **Compatibilidad con programas externos**

Hablamos de compatibilidad en el sentido de poder importar archivos de programas externos al entorno de Unreal para poder utilizarlos como componentes dentro del videojuego. Se pueden importar distintos tipos de archivos que vamos a explicar a continuación.

En cuanto a los modelos 3D se debe importar en formato fbx, dicho formato lo exportan la mayoría de sistemas de modelado 3D por lo que no resulta complicado.

Para los archivos de audio se pueden importar pequeños archivos de sonido de 16 bits sin comprimir en cualquier frecuencia de muestreo, aunque se recomiendan frecuencias de muestreo de 44100 Hz o 22050 Hz. También se pueden importar sonidos base con distintas características, según la siguiente tabla:



* 1. **Servicios**

En Unreal Engine se pueden configurar distintos tipos de herramientas de análisis según el tipo de videojuego o las estadísticas que se quieran obtener.

**Blueprint analytics plugin**

Esta API está integrada en el entorno de Unreal engine pero debe ser habilitada como complemento, permite ejecutar análisis sobre el juego durante el desarrollo. El plug-in traduce las llamadas y los envía al proveedor de análisis por defecto que se haya registrado para el proyecto. Una vez iniciado el análisis para una sesión de usuario se pueden registrar los eventos que se producen y ver su frencuencia, característica, atributos etc.

Para utilizar la API anterior debe escogerse un proveedor de los cuatro ofrecidos por Unreal, se diferencian en el tipo de plataforma sobre la que se va a ejecutar el juego:

* Apsalar, es el analizador para IOS. Comenzó como un servicio de análisis gratuito que permite reportar publicidad a Facebook, aunque recientemente se ha pasado a tener una cuota. Lo más destacado es que se puede medir el retorno de la inversión para los anuncios para móviles de Facebook, algo poco común. Para utilizarlo, el desarrollador se debe registrar, obtener una clave de aplicación y la contraseña que la identifica de forma exclusiva. Después descargar las bibliotecas que se compilan en el plugin Apsalar y colocarlas en el directorio habilitado.
* Flurry analytics provider, es otro analizador gratuito para IOS que se utiliza para comparar los datos con otros proveedores.
* File logging analytics provider, se utiliza para escribir las llamadas de la API de análisis en el disco en formato JSON, permitiendo depurar el proceso de análisis. Se escribe los datos en archivos propios en la carpeta Analytics permitiendo comparar los datos con los producidos por los eventos de consola.
* Multicast analytics provider, permite utilizar varios proveedores a la vez sin necesidad de ejecutarlos manualmente. Esto permite tener todas las cosas buenas de cada uno de los proveedores.