

**TFG**

---

**ANCIENT TEMPLE ANEXO I**

**GLOSARIO DE TÉRMINOS**

**Presentado por Cristina Ortega Redondo**

**Tutor: Francisco Martí Ferrer**

**Facultat de Belles Arts de San Carles**

**Grado en Bellas Artes**

**Curso 2013-2014**



**UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA**



**UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
FACULTAT DE BELLES ARTS DE SANT CARLES**

**Alpha:** En computación gráfica, la composición alfa o canal alfa es la que define la opacidad de un píxel en una imagen. El canal alfa actúa como una máscara de transparencia que permite, de forma virtual, componer (mezclar capas) imágenes o fondos opacos con imágenes con un cierto grado de transparencia. Así, en un formato de imagen que admita transparencia, a los tres canales de color básicos RGB que definen la cantidad de rojo, verde y azul respectivamente del píxel, se añade un cuarto canal, el alfa, que define el grado de opacidad de ese píxel.

**Assets:** Los assets son los elementos que serán introducidos al videojuego. Estos elementos incluyen modelos 3D, personajes, texturas, materiales, animaciones, scripts, sonidos, y algunos elementos específicos de cada motor. Cada motor trabaja de una manera distinta a otros lo cual puede aceptar "Assets" que otros motores no pueden manejar, sin embargo los ejemplos mencionados antes, son elementos que todos los motores de hoy en día usan.

**Avatar:** Es una representación gráfica, generalmente humana, que se asocia a un usuario para su identificación. Los avatares pueden ser fotografías, dibujos artísticos o representaciones tridimensionales.

**BSP Brush:** Es una herramienta codificada del software del motor, que no todos los motores utilizan. Son utilizados por motores de nueva generación como Unreal y CryEngine que los utilizan y manejan High-Order surfaces para deformar la superficie mediante vértices. También, se pueden crear BSPs con Brushes que crean figuras geométricas simples como: Cuadrado, Esfera, Cilindro, Cono, etc.

**Bug:** Es un error de software, comúnmente conocido como bug («bicho»). Un error o fallo en un programa de computador o sistema de software que desencadena un resultado indeseado.

**Concept art:** El arte de concepto o *concept art* es a grandes rasgos, el arte del proceso de desarrollo de juegos o películas a través del cual se define la estética final de la producción. Toma lugar en la fase de pre-producción y asegura la dirección que tomará el proyecto antes de la siguiente fase de inversión.

**Colisiones:** La Detección de colisiones es el método utilizado por algunos videojuegos para detectar si dos objetos (Sprites) han colisionado. La detección puede ser por área o píxel a píxel. Por Área: los objetos ocupan un área, rectangular o circular, cuando dos de estas áreas se superponen hay una colisión. Píxel a Píxel: Los objetos ocupan un área rectangular, pero tienen una máscara que define que píxeles son visibles. Primero se realiza una detección

de colisión de área, luego, si hubo colisión, se realiza una detección pixel a pixel entre los píxeles superpuestos de ambos objetos. Si existen dos píxeles superpuestos, y ambos son visibles, entonces hay una colisión.

**Editable poly:** Modificador que se aplica a una maya poligonal y permite la edición de vértices, caras, aristas y polígonos mediante numerosas herramientas.

**FBX:** Formato de archivo 3D independiente de la plataforma que proporciona acceso al contenido creado en cualquier paquete de software.

**Gameplay:** La jugabilidad es un término empleado en el diseño y análisis de juegos que describe la calidad del juego en términos de sus reglas de funcionamiento y de su diseño como juego. Se refiere a todas las experiencias de un jugador durante la interacción con sistemas de juegos. La definición estricta de jugabilidad sería aquello que hace el jugador. Otra definición posible, dada por el famoso diseñador de videojuegos Sid Meier, sería esta: La jugabilidad es una serie de decisiones interesantes.

**LOD (*level of detail, nivel de detalle*):** el sistema de nivel de detalle está relacionado con la complejidad geométrica de los modelos. Algunos sistemas necesitan que se hagan múltiples versiones del modelo, para que dependiendo de cuán cerca se este del modelo así será su cantidad de polígonos. Otros sistemas ajustan dinámicamente esta característica pero en este caso da más carga al CPU

**Mapeado ( *UV Mapping*):** Es el proceso por el cual se proyecta un mapa de textura en un objeto 3D. Las letras "U" y "V" denotan los ejes de la textura 2D porque "X", "Y" y "Z" ya se utilizan para referirse a los ejes del objeto 3D. El texturizado UV permite a los polígonos que componen un objeto 3D ser pintados con el color de una imagen. La imagen es llamada *UV texture map* pero es sólo una imagen común. El proceso de mapeado UV consiste en la asignación de los píxeles de la imagen a la superficie de asignaciones en el polígono, generalmente por "programación" copiar una pieza en forma de triángulo del mapa de imagen y pegarla en un triángulo en el objeto.

Si la malla es una esfera UV, por ejemplo, el modelador puede transformarla en una proyección equidistante. Una vez que el modelo se desenvuelve, el artista puede pintar una textura en cada triángulo individualmente, usando la malla sin envolver como una plantilla. Cuando se representa la escena, cada triángulo asignará a la textura apropiada de la "hoja de calco."

Un mapa UV o bien puede ser generado automáticamente por la aplicación de software, hecho manualmente por el artista, o alguna combinación de ambos. A menudo, un mapa UV se generará, y luego el artista ajustará y

optimizará el mismo para reducir al mínimo las costuras y solapamientos. Si el modelo es simétrico, el artista puede superponer triángulos opuestos para permitir que la pintura de ambos lados se realice simultáneamente.

El proceso de asignación de UV a su forma más simple requiere tres pasos: Desenvolver la malla, la creación de la textura, y la aplicación de la textura.

**Mapa de normales**(*Normal Map*): Se trata de una imagen 2D que se utiliza para modificar una superficie 3D. Es un mapa de relieve que utiliza la información por píxel en colores RGB para representar los tres ejes X,Y,Z. De este modo, se recrea el relieve de una malla detallada, aunque el observador al acercarse a dicho objeto pierde la sensación de relieve ya que se trata sólo de un efecto visual creado por los sombreadores por píxel.

**Mapa de relieve** (*Bump map*): Es un mapa 2D anterior a la creación de los mapas de normales en el que solo se representa la altura y la información por píxel se aplica en tonos de escala de grises.

**Modelado**: La creación de modelos digitales tridimensionales es la piedra angular del desarrollo de juegos. Los modelos construyen mundos virtuales en los cuales sucede el juego; cada animal, persona, edificio y objeto es un modelo tridimensional. Los modelos digitales son entidades virtuales: aunque parezcan tener profundidad, en realidad no la tienen, la sensación de profundidad se crea a partir del rendimiento del programa en 3D. Para ello, es necesario que el programa cree un modelo de espacio tridimensional con ese fin, se generan tres ejes perpendiculares entre sí y, mediante una referencia a los tres ejes, se puede situar un punto en el espacio.

Los modelos tridimensionales están compuestos de una malla (mesh) de puntos interconectados en un espacio virtual tridimensional. Esta malla está formada por triángulos o polígonos. El punto correspondiente a la esquina de un polígono es un vértice, y la línea situada entre los vértices es un lado. Un polígono a veces recibe el nombre de “cara”, pero normalmente este término se refiere a muchos polígonos que forman el lado de un objeto. La malla de polígonos forma una superficie virtual sobre la cual se puede situar un color o una imagen.

**Modelo de alta poligonización** (*High poly model*): Modelos de un realismo extremo y gran cantidad de detalles. La concentración de polígonos es muy alta por lo que aumenta de manera exponencial el tiempo de procesado.

**Modelo de baja poligonización** (*Low poly model*): Se trata de un modelo 3D que presenta poca cantidad de polígonos y por lo tanto el tiempo de procesado es muy bajo. Sin embargo, se sacrifican los detalles y la calidad del modelado.

**Motor de juego:** Un motor de videojuego es un sistema de software diseñado para la creación y desarrollo de videojuegos. Existen múltiples motores de juegos diseñados para trabajar en consolas de videojuegos y sistemas operativos. En general, la funcionalidad típica de estos motores es proveer un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, animación, inteligencia artificial, redes, streaming, scripting, administración de memoria, escenario gráfico, etc. El desarrollo de un videojuego se reduce considerablemente por reusar o adaptar un mismo motor de videojuego para crear diferentes juegos. Muchas veces los motores de juegos proveen herramientas de desarrollo visual, además de componentes de software reusables, provistas muchas veces en forma de IDE. Algunos motores solo proveen funcionalidades de 3D, en lugar de todas las funcionalidades mencionadas. Esto suelen llamarse motores gráficos (graphics engine) o motores de renderizado (rendering engine) o motores 3d (3D engine).

**Oculus Rift:** Es un head-mounted display de realidad virtual que está siendo desarrollado por *Oculus VR*. Durante su periodo como compañía independiente. Oculus Rift es un dispositivo de realidad virtual con un amplio campo de visión y con un grado de inmersión nunca antes visto.

**Plataforma:** En informática, determinado software y/o hardware con el cual una aplicación es compatible y permite ejecutarla. Una plataforma es, por ejemplo, un sistema operativo, un gran software que sirve como base para ejecutar determinadas aplicaciones compatibles con este. También son plataformas la arquitectura de hardware, los lenguajes de programación y sus librerías en tiempo de ejecución, las consolas de videojuegos, etc. Existen programas multiplataforma, que permiten ejecutarse en diversas plataformas. También existen emuladores, programas que permiten ejecutar desde una plataforma programas de otra emulando su funcionamiento.

**Poligonización:** Cantidad de polígonos que conforman un modelo 3D. Cuantos más polígonos se utilicen para construir un modelo, más real y homogéneo parecerá. Sin embargo, cuanto mayor sea el número de polígonos, mayor será el número de cálculos que tiene que realizar el programa de modelado y más adelante el motor de juego para crear la imagen en 3D. Incluso con los mejores ordenadores y consolas actuales, se suele asignar con mucho cuidado el número de polígonos desde el momento del diseño; así, se consigue que el juego se ejecute a velocidad continua y sin interrupciones debidas a los tiempos de cálculo necesarios para representar las imágenes en pantalla.

**Prefab :** Un prefab es creado para instanciar de forma sencilla un modelo formado por varias piezas o piezas y física conjuntamente. De este modo,

podemos mover este objeto como un único grupo. Una de sus características más destacadas es que cada prefab, referencia al modelo original pero pueden ser modificados individualmente o modificar uno de ellos y actualizar el resto para aplicar al conjunto esa modificación.

**Props:** Los props son objetos de utillería o atrezzo que se utilizan en el set de grabación de una película, o en este caso en el escenario de un juego.

**Renderizado (Render):** Para ver un modelo en 3D el usuario adopta un punto de vista. Una vez que un modelo ha sido construido, dotado de textura e iluminado, se puede generar una imagen de dicho modelo desde ese punto de vista, el render.

**Sistema de partículas:** Un sistema de partículas es una función con la que se puede animar una gran cantidad de objetos. Este tipo de sistemas se utilizan por ejemplo para simular el comportamiento del humo, del fuego o de una explosión. El nombre proviene porque se basa en computar la animación basándose en los sistemas de partículas de la mecánica clásica.

En los sistemas de partículas hay un emisor que expulsa partículas, cuyo movimiento puede influenciarse por medio de numerosos parámetros. Entre ellos cabe destacar: Velocidad de expulsión, esperanza de vida, amortiguamiento, cantidad de partículas en el sistema, influencias de la fuerza sobre las partículas... Las partículas son inicialmente sólo elementos lógicos a los que se debe otorgar propiedades gráficas para que sean visibles. Por ejemplo, cada partícula se puede sustituir por un objeto geométrico, gracias a lo cual es posible mostrar enjambres o campos de asteroides. Con la adición del material correspondiente se pueden utilizar sistemas de partículas para mostrar humo, niebla o fuego.

**Textura procedimental:** Son aquellas texturas que se generan mediante un algoritmo de programación. Estas texturas parecen naturales, pero se han creado íntegramente por ordenador gracias a unos cálculos fractales que simulan el aspecto de unos materiales ricos en detalles y repetitivos como la madera, el agua, el metal...

**Tostado de normales (Baking normals):** Es un proceso que trata de la creación de un modelo de baja poligonización y un modelo de alta poligonización a partir del mismo. A través de este método de proyección, se extraen los mapas de normales del modelo de alta poligonización y se aplican al modelo de baja poligonización. De este modo, se consigue que un modelo que ocupa menos recursos de memoria y es menos pesado tenga una cantidad de detalles similares al modelo de alta poligonización. Por tanto, se trata de un proceso de optimización.

**Wireframe:** es un algoritmo de renderización del que resulta una imagen semitransparente, de la cual solo se dibujan las aristas de la malla que constituye al objeto. De ahí su nombre. Casi nunca se emplea en la representación final de una imagen, pero sí en su edición, debido a la escasa potencia de cálculo necesaria (comparada con otros métodos). Para conseguir una imagen en wireframe solo tenemos que tener en cuenta las posiciones de los puntos en el espacio tridimensional y las uniones entre ellos para formar los polígonos. Habitualmente estas imágenes no tienen en cuenta la presencia de luces en la escena.