

Tutorial Threejs – Parte I , Introducción

Hola a todos.

Vamos a comenzar una serie de posts dedicados a Threejs , que espero que sea de vuestro agrado.

¿Que es Threejs?

[Threejs](#) es motor 3D ligero para javascript .

Funciona muy bien bajo [webGL](#) , aprovechando la aceleración de nuestra tarjeta gráfica , dando unos resultados espectaculares.

Podemos encontrar unos ejemplos de lo que se puede hacer en la [página de ejemplos oficial](#) de Threejs.

Ésta serie de artículos pretende explicar el funcionamiento de Threejs , que hacer , como empezar , como crear escenas , importar objetos , etc , etc.

Empezaremos explicando un poco que es necesario para empezar:

Un navegador con soporte webGL , actualmente los mejores resultados los da Google Chrome , aunque se puede utilizar Mozilla Firefox , aunque es más pesado y por lo tanto un poco más lento , no olvidemos que Chrome utiliza el motor webkit que utilizan también muchos smartphones , mientras que Mozilla utiliza el motor Gecko , basado en java .

Conocimientos de programación , javascript , y html5 .

Nociones de 3D , para no perderse , o mucha paciencia para absorber conceptos nuevos.

Descargar la librería [Threejs](#) con los ejemplos y herramientas

que iremos utilizando y explicando.

Que podemos hacer !

Con Threejs podemos hacer casi cualquier cosa que podemos hacer con un motor 3D , vamos a explicarlo para que sirva también como tutorial de iniciación al 3D.

Básicamente consiste en crear una escena , con objetos dentro que visualizaremos en la pantalla , en nuestro caso en un canvas de html5 , a través de cámaras e iluminados por luces .

La escena

Es la representación del mundo virtual tridimensional , donde se irán colocando los demás elementos , es algo así como un contenedor del espacio en 3D , que empieza estando vacío .

Este espacio está vacío , no tiene nada , iremos colocando por turno los objetos que deseamos visualizar , basándonos en que la escena se puede medir , con eje de coordenadas virtual a partir del cual iremos posicionando los objetos en las 3 dimensiones básicas .

Gracias a esta referencia dentro de la escena podemos rotar los objetos , desplazarlos , etc , etc.

La escena tiene un fondo que se puede personalizar , que sería un poco así lo que sería el horizonte , se puede dejar un color plano , o una imagen , depende de lo que se quiera hacer.

La cámara

La cámara nos permite visualizar la escena desde un punto de vista concreto .

Se comporta como un objeto más , se puede rotar , desplazar , pero además tiene algunas propiedades más , propias de una cámara .

Podemos tener varias cámaras y utilizar una u otra según

convenga.

Las luces

Las luces son objetos que irradian luz y nos permiten visualizar los objetos gracias a la reflexión de la luz en ellos.

Hay varios tipos de luz , de punto (Point), solar (sun), foco (Spot) , en otras aplicaciones podemos encontrar más tipos de luz.

La intensidad y el color de las luces son configurables y nos permiten una multitud de posibilidades y pueden provocar sombras sobre los objetos de la escena.

Cada objeto reacciona diferente a la luz según sus valores para la luz de difusión (color reflejado) , especular (color refractado) y emisión (color emitido por ejemplo una bombilla) , variando según el color e intensidad.

Para una iluminación correcta es necesario el uso de normales , que son el cálculo de la perpendicular de cada cara visible del objeto .

Para empezar , la mayoría de aplicaciones comienzan con un escena que contiene una cámara y una luz , en muchos casos incluye un objeto en modo de ejemplo que suele ser un cubo o una esfera .

Los objetos

Los objetos son una representación de arrays de vértices que forman líneas y caras.

Jeje que fácil no ,

Los vértices son puntos que ocupan unas coordenadas .

La unión de 2 vértices forma líneas.

La unión de varias líneas forman planos que se suelen llamar caras.

Las caras se suelen formar por 3 o 4 líneas que le dan forma , y obviamente una cara/plano de 4 líneas se puede representar como 2 caras/planos de 3 líneas , así que lo más normal es encontrarnos con modelos de objetos formados por arrays de vértices que representan estos triángulos .

En la mayoría de aplicaciones para modelar se especifica al exportar si se quieren crear triángulos (3 líneas) en aquellas caras que son romboides (4 líneas) , para su uso posterior.

Este array de vértices que comentamos no es más que la forma del objeto , hay que posicionar el objeto en la escena .

Los objetos están hechos con uno o varios materiales a los que hay que asignarle valores según tenga que reaccionar a la luz un color de difusión , otro de especular y el color de emisión si emite luz.

Cada material puede tener una textura , que puede ser una imagen , hasta en ocasiones un vídeo.

Las texturas dan el toque de calidad a los objetos , siendo unos de los temas más importantes del desarrollo en 3d ,por ejemplo podemos aplicar transparencias gracias a las texturas , o de un plano básico aplicarle una textura para que parezca una casa.

Los objetos pueden ser de formas básicas como un cubo o una esfera , o puede ser una figura compleja , pero en el caso de ser una figura compleja mejor crearla con un editor externo (no con código me refiero) , e importarlo como un modelo.

Los modelos

Los modelos 3D son objetos pre-diseñados que se utilizan para incorporar en la escena. Pueden incluir sus propias texturas , y estar formados por uno o varios objetos , a su vez con sus propias texturas.

Los modelos se comportan como una unidad , y se ubican a

partir de un punto que es su centro de coordenadas , que se suele ubicar abajo , pero puede variar según el creador.

Podemos diseñar objetos en aplicaciones como Blender , con la comodidad que ello aporta , y exportarlos para el uso en nuestras aplicaciones , Threejs nos aporta herramientas para convertir objetos en formato obj en objetos en formato compatible para Threejs en json.

Los modelos pueden tener animaciones programadas preparadas para nuestro uso , por ejemplo ficheros md2 con acciones como saltar , correr , cargar , etc .

Threejs nos aporta una herramienta para crear modelos en formato Threejs , muy útil y práctica.

Resumiendo un poco , ya sabemos que tenemos que crear una escena , añadirle una cámara , una luz , y los objetos y modelos que queramos.

Bueno hasta aquí la introducción , en el próximo artículo pasaremos a la acción creando la primera escena .

Saludos ,
Scuraki