Usar Seen.js Para Crear Escenas 3D En JavaScript

Hace unos pocos años necesitábamos software de alta gama, para crear cualquier tipo de escena 3D. Ahora, tenemos un montón de maneras de crear escenas 3D más complejas y usarlas directamente en el navegador.

Pero si queremos tomar un camino bastante fácil, se puede utilizar una biblioteca, como la impresionante libreria Seen.js, la cual hace la vida un poco más fácil, para la creación de escenas 3D en JavaScript con sólo **unas pocas líneas de código**.

Usando CoffeeScript

SeenJS se construyó utilizando CoffeeScript, y todos los ejemplos que se muestran usan **CoffeeScript**. Aunque se puede usar JavaScript básico, en este paso a paso usaremos **CoffeeScript** para que sea más fácil de aprender.

Creando La Escena

En primer lugar vamos a crear el canvas que mostrara nuestra escena:

[html]<canvas id="canvas" width="500" height="500"> [/html]

Como siempre en una escena 3D, lo primero que vamos a necesitar es una figura. En nuestro caso, vamos a utilizar un icosaedro, pero se puede buscar en la fuente otras figuras para utilizar.

Para crear el icosaedro tenemos que llamar *Shapes* y entonces

la figura que estamos usando. Seguidamente, tenemos que modificar la escala hasta el tamaño que queremos:

[js]shape = seen.Shapes.icosahedron().scale(150)[/js]

Hemos creado una figura, pero aún no se ha añadido a la escena. Para hacer eso, tenemos que crear la escena:

[js]scene = new seen.Scene[/js]

Después de esto se crea una instancia, vamos a añadir nuestra figura de modelo a la escena:

[js]

model : seen.Models.default().add(shape)
viewport : seen.Viewports.center(500, 500)

[/js]

Como podemos ver, llamamos el modelo y luego la función **default()** que contiene un conjunto de 3 luces que iluminará nuestra escena correctamente. Después se añaden las luces que llamamos el **add()** y fijamos nuestra figura como el valor.

Después de que el modelo también fijamos el **viewport** y aquí le debemos simplemente ponemos el ancho de la ventana deseada, en nuestro caso serán las dimensiones del **canvas**.

Nuestra sencilla escena está muy cerca de ser finalizado, lo único que queda hacer es hacer, añadir la escena para que se pueda ver en nuestro navegador:

[js]context = seen.Context('canvas', scene).render()[/js]

Llamamos **seen.Context()**, que toma dos parámetros: el primero es el ID del archivo de canvas o SVG, donde se representará la escena, y la segunda es la propia escena. Por último, llamamos a la función **render()** y ahora se puede ver el icosaedro en su navegador.

Añadiendo Interactividad

Nuestro escenario está listo, pero falta algo, la interaccion con el usuario. Vamos a empezar a arreglar eso agregando un color al azar a la superficie:

[js]seen.Colors.randomSurfaces2(shape)[/js]

Esto debería ser añadido a la derecha por debajo de la línea de creación del icosaedro. Actualizar tu página ahora y verás que funciona bastante bien.

A continuación, vamos a añadir la interactividad. Queremos que el usuario sea capaz de girar la escena haciendo clic y arrastrando con el ratón. Para lo cual, tendremos que crear una instancia de un nuevo objeto Drag y pasar el ID al canvas de la misma:

[js]drag = new seen.Drag(‘canvas’, {inertia : true})[/js]

Después de esto, necesitamos una función que se ejecutará cuando el usuario arrastra la figura:

[js]

drag.on(‘drag.rotate’, (e) ->
xform = seen.Quaternion.xyToTransform(e.offsetRelative…)
shape.transform(xform)
context.render()
)

[/js]

Finalmente

Esto es todo lo que tiene que hacer para crear una escena 3D simple con un poco de interactividad **usando SeenJS**. La llegada de este tipo de biblioteca sólo demuestra lo mucho que la Web

ha cambiado en los últimos años. Espero que considere acoger esta nueva tecnología, porque es más fácil de usar que nunca!