

Nociones básicas de javascript para three.js

Joseba Makazaga

UPV/EHU

A la hora de trabajar con three.js necesitaremos definir puntos, vectores y matrices. Los puntos y los vectores se representan mediante **vector3**¹ o **vector4**². Las matrices pueden ser **Matrix3**³ o **Matrix4**⁴.

Hay que tener en cuenta que en gráficos se trabaja en coordenadas homogéneas, por lo que los puntos y vectores tienen 4 coordenadas, la cuarta en los puntos es un 1 mientras que en los vectores es 0. Los 4 campos del vector4 son x, y, z, w . Para definir un punto o un vector tenemos las siguientes declaraciones de variables (sin argumentos x, y, z se inicializa a 0, mientras que w se inicializa a 1):

```
1 var mivector = new THREE.Vector4( 0, 1, 0, 0 );
2 var mipunto = new THREE.Vector4();
3 var miotropunto = new THREE.Vector4(3, 4, 5, 1);
```

Las matrices las podemos crear mediante una llamada a su constructora: **Matrix4()** que inicializará la matriz con la matriz identidad. Los valores de la matriz se guardan en el campo o **property** llamado **elements**, pero, aunque para asignarle valores hay que pasarle los mismos en formato **row-major**, internamente se guarda de forma traspuesta o **column-major**.

```
1 var mimatriz = new Matrix4();
2
3 mimatriz.set(Math.cos(a), -Math.sin(a), 0, 0,
4              Math.sin(a),  Math.cos(a), 0, 0,
5              0,          0,          1, 0,
6              0,          0,          0, 1);
```

Una vez definidas las variables podemos realizar operaciones como pueden ser multiplicaciones, sumas, productos vectoriales... Por ejemplo, a un vector le puedo sumar otro vector, y dejar el resultado en el primero de ellos, o

¹<https://threejs.org/docs/api/en/math/Vector3.html>

²<https://threejs.org/docs/api/en/math/Vector4.html>

³<https://threejs.org/docs/api/en/math/Matrix3.html>

⁴<https://threejs.org/docs/api/en/math/Matrix4.html>

puedo asignarle a un vector el resultado del producto vectorial de otros dos vectores:

```
1 var v1 = new THREE.Vector3( 0, 1, 0 );
2 var v2 = new THREE.Vector3( 1, 0, 0 );
3 var v3 = new THREE.Vector3();
4 var minumero;
5
6 v3.addVectors(v1,v2);    // v3 = v1 + v2
7 v3.add(v1);              // v3 = v3 + v1
8 v3.normalize();          // v3 = v3/||v3|| (misma direccion
    pero longitud 1)
9 v3.copy(v1);             // v3 = v1
10 v3 = v1.clone();         // v3 = v1 (pero genera un Vector3
    nuevo, no hace falta new previo)
11 v3.cross(v1);            // v3 = v3 ^ v1 (producto vectorial)
12 v3.crossVectors(v1,v2); // v3 = v1 ^ v2 (producto vectorial)
13 minumero = v1.x*v2.x + v1.y*v2.y + v1.z*v2.z; // producto
    escalar accediendo a los campos
14 minumero = v1.dot(v2);   // producto escalar.
```

Además de las operaciones entre vectores podemos multiplicar matrices a los puntos, o podemos multiplicar matrices:

```
1 var v1 = new THREE.Vector3( 0, 1, 0 );
2 var v2 = new THREE.Vector4();
3 var M1 = new Matrix3();
4 var M2 = new Matrix4();
5 var M3 = new Matrix4();
6
7 M1.set(Math.cos(a), -Math.sin(a), 0,
8         Math.sin(a),  Math.cos(a), 0,
9         0,             0,           1);
10 M2.set(1, 0, 0, -2,
11        0, 1, 0, -4,
12        0, 0, 1, -2,
13        0, 0, 0,  1); // M3 = matriz de traslacion
14 M2.makeTranslation(-2,-4,-2) // M3 = matriz de translacion (
    misma que la anterior)
15 M3.makeRotationX(alfa); // M3 = matriz de rotacion respecto
    a eje X, alfa grados
16 M3.multiply(M2);        // M3 = M3 x M2 (multiplicacion
    matricial)
17 M3.premultiply(M2)      // M3 = M2 x M3 (multiplicacion por la
    izquierda)
```

Para multiplicar matrices a los puntos hay que tener en cuenta si trabajamos con coordenadas homogéneas o cartesianas. Podemos pasar de unas a otras expandiendo los puntos con un 1 y los vectores con 0:

```

1 v1.applyMatrix3(M1);    // v1 = M1 x v1  (matriz3x3 por
    vector3)
2 v2.set(v1.x, v1.y, v1.z, 1);  // expansion de punto v1 a
    homogeneas
3 v2.apllyMatrix4(M3);    // v2 = M3 x v2  multiplicacion de
    matriz4x4 y vector4
4 v1.set(v2.x/v2.w, v2.y/v2.w, v2.z/v2.w); paso de homogeneas a
    cartesianas: solo puntos!
5 v1.applyMatrix4(M3)     // equivalente a las tres anteriores
6 v2.set(v1.x, v1.y, v1.z, 0); // expansion de vector v1 a
    homogeneas

```