

Proyecto Grupal

Suma y Resta de Vectores con cámaras

Diego Vargas y María Paula Velandia.

Ingeniería Multimedia

Universidad Militar Nueva Granada

u6000460@unimilitar.edu.co y u6000204@unimilitar.edu.co

Resumen- En este documento explicaremos que es una cámara, se darán a conocer los 4 conceptos importantes que se deben tener en cuenta al hacer la rotación de la cámara, cada concepto irá explicado y nos dará la funcionalidad e importancia de cada uno, se mostrará el proceso que se llevó a cabo y lo que finalmente quedó de acuerdo con el cronograma establecido.

I. ¿Qué es una cámara?

Una cámara en webgl significa que es la que deja ver a las personas un objeto desde diferentes ángulos y no es necesario mover tu objeto, por ejemplo, cuando uno toma una foto a un edificio lo que uno hace es mover la cámara para captar el ángulo en el cual uno quiere obtener la foto mas no rotamos ni movemos el edificio, de esto trata la cámara en webgl que nos ayuda a ver un objeto sin necesidad de tener que mover el objeto del plano solo moveremos la cámara en los diferentes ángulos para así poder ver los objetos desde diferentes perspectivas.

II. Rotación de Cámara.

Para poder realizar la rotación de cámaras tenemos que tener en cuenta los siguientes 4 conceptos:

1. **Frustum:** es la porción de un sólido el cuál se encuentra entre uno o dos planos paralelos, en los gráficos del computador el núcleo de visualización de este es la región tridimensional que esta es visible en la pantalla.

Cada sección plana es un piso o base del tronco, un tronco es circular si este tiene bases circulares esto es correcto si el eje es perpendicular a ambas bases o si no de lo contrario es oblicuo.

La altura de un tronco es la distancia perpendicular entre los planos a la base, los conos y las pirámides se pueden ver como casos generados por el Frustum.

2. **Eye:** El ojo es el vector que indicará la posición desde la cual se verá el objeto, este vector estará

3. conformado por 3 números los cuales serán x , y , z , una vez escogido estos valores podremos ver en qué posición definida nos quedó ubicado el Ojo.

4. **Center:** EL centro es el vector que especifica la ubicación del objeto que se quiere ver, en que plano este estará ubicado, con respecto a esta ubicación se tendrá en cuenta las rotaciones de los demás objetos creados ya que cada objeto tendrá un centro diferente, todo esto depende de la ubicación en el plano que se le vaya a dar a cada objeto.

5. **Up:** Este es la ubicación de la cámara, en donde se definirá la posición en la cual la cámara empezará a mirar el objeto.

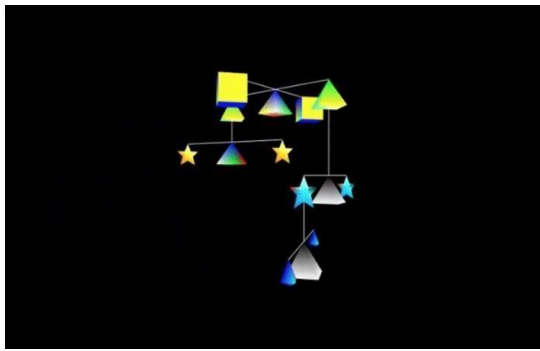
III. Cambio de Proyecciones.

Para especificar si una proyección es de tipo perspectiva se tiene que utilizar una librería que se llamaría *mat4.perspective* de lo contrario si la proyección es de tipo Ortogonal se tendría que llamar la librería *mat4.ortho* para la proyección ortogonal se tienen que mirar los bordes máximos y mínimos del campo visualizado en cada objeto.

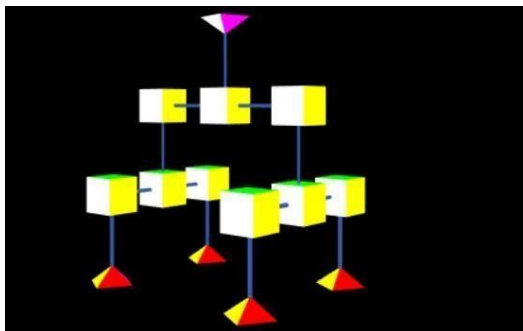
La especificación de cada figura se dará cuando se le dé la posición a cada uno de los objetos esta debe ser de más o menos de 5 unidades tanto en X como en Z y este nos garantizará un zoom en el plano ortogonal.

IV. Proyectos realizados

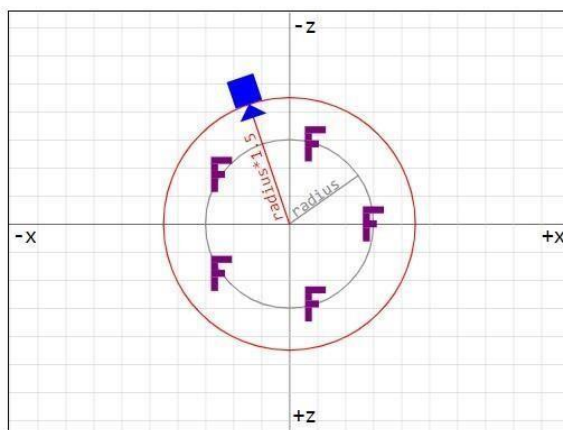
- Interactividad con WebGL: El proyecto fue publicado el día 30 de Mayo del año 2018. Este proyecto tenía como finalidad mostrar las diferentes vistas y transformaciones de la cámara sobre un objeto.



- **Cámara en WebGL:** El proyecto fue publicado el día 06 de Marzo del año 2018, este proyecto tenía como finalidad explicar en qué consiste la cámara y que conceptos se debían tener presentes antes de empezar a hacer la rotación de la cámara.



- **Cámaras en 3D:** El proyecto fue publicado en el año 2015, este proyecto tenía como finalidad explicar el concepto de cámara por medio de una letra en el cual la cámara se movía en forma circular hasta para así poder ver todos los ángulos de la letra y la creación de esta cámara la hacían por medio de matrices.



V. ¿Qué vamos a hacer?

Lo que queremos realizar en nuestro proyecto es primero que todo la suma y resta entre los mismos objetos en donde estas sumas y restas consisten en escalar, trasladar y rotar cada uno de los objetos creados.

- **Escalar:** Lo que trata esta función es crear una matriz la cual se aplicará al objeto y este lo que hará es una suma de la misma matriz para que el objeto quede 2 veces mayor al original

- **Trasladar:** La traslación consiste en que el objeto se le aplica una matriz en la cual se indica las unidades que se desea trasladar el objeto, estas unidades serán positivas y a la vez se decidirá en que eje se quiere trasladar el objeto.
- **Rotación:** La rotación puede ir en sentido positivo o en sentido negativo dependiendo de los valores que nosotros vayamos a darle a la matriz.

Por otro lado, queremos hacer que la cámara en perspectiva ya creada rote sobre cada objeto mostrado así cada objeto desde sus diferentes ángulos.

Continuando con la realización del proyecto y debido a lo que se estableció en el cronograma se empezaron a hacer los siguientes adelantos:

```
//animate();
function init(){
//INICIALIZACIÓN
    scene = new THREE.Scene();
    aspect = window.innerWidth / window.innerHeight;
    camera = new THREE.PerspectiveCamera( 75, aspect, 0.1, 1000);
    camera2 = new THREE.PerspectiveCamera( 45, aspect, 0.1, 1000);
    camera3 = new THREE.PerspectiveCamera( 45, aspect, 0.1, 1000);
    camera4 = new THREE.PerspectiveCamera( 45, aspect, 0.1, 1000);

    renderer = new THREE.WebGLRenderer();
    renderer.setSize( window.innerWidth, window.innerHeight );
    document.body.appendChild( renderer.domElement );
    controls = new THREE.OrbitControls( camera, renderer.domElement );
```

Este código lo que muestra la inicialización de las cámaras en perspectiva que se van a utilizar durante la realización del proyecto.

//ELEMENTOS COMUNES

```
var size = 10;
var arrowSize = 1;
var divisions = size;
var origin = new THREE.Vector3( 0, 0, 0 );
var x = new THREE.Vector3( 1, 0, 0 );
var y = new THREE.Vector3( 0, 1, 0 );
var z = new THREE.Vector3( 0, 0, 1 );
var color = new THREE.Color( 0x333333 );
var colorR = new THREE.Color( 0xAA3333 );
var colorG = new THREE.Color( 0x33AA33 );
var colorB = new THREE.Color( 0x333366 );
```

En esta parte del código se crearon los elementos básicos y colores que se llegaron a utilizar en el proyecto.

Luego creamos las diferentes geometrías que íbamos a utilizar

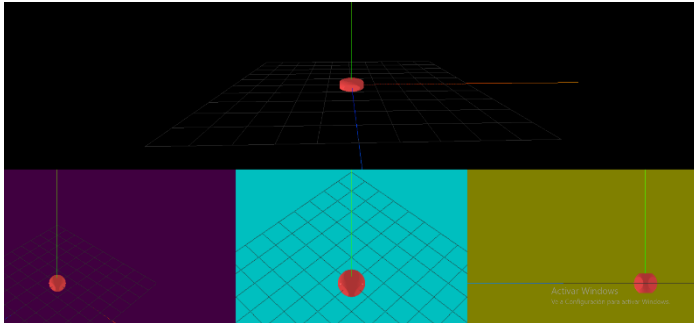
```
var boxGeometry = new THREE.BoxGeometry( 1, 1, 1 );
var sphereGeometry = new THREE.SphereGeometry( .6, 32, 32 );
var geocilindro = new THREE.CylinderGeometry( .3, .3, 2, 32 );
```

Enseguida se hicieron las operaciones que queríamos hacer como eran la sumas, restas de los diferentes vectores y geometrías que queríamos como lo fue un cubo, una esfera y un cilindro.

y por último este fue el resultado final que se obtuvo de todo el proyecto

del código y se hará el documento final en donde se mostrarán los resultados finales esperados.

DÍA: 04/12/2019: ENTREGA DEL PROYECTO.



En esta imagen se puede ver como se empezó la unión de dos objetos creados y nos muestra desde las diferentes cámaras como se ve el objeto con sus respectivas operaciones realizadas.

REFERENCIAS

- [1] <https://medium.com/@az.martinez4799/c%C3%A1mara-en-webgl-366be536d37c>.
- [2] <http://drmayor.blogspot.com/2018/03/5-vistas-de-camara-en-webgl.html>.
- [3] <http://macedoniamagazine.frodrig.com/opengl6.htm>

CRONOGRAMA

DÍA: 18/09/2019: El día 18 de septiembre se hará un avance del documento

DÍA: 25/09/2019: El día 25 de septiembre se realizará la entrega del segundo documento en el cual va especificado los conceptos con los cuales se realizará el proyecto.

DÍA: 02/10/2019: El día 02 de octubre se empezará la implementación del código de la creación de la cámara.

DÍA: 09/10/2019: El día 03 de octubre se empezarán las transformaciones dichas para cada uno de los objetos creados.

DÍA: 16/10/2019: El día 16 de octubre se empezará a explicar por medio de un documento cada pedazo del código hasta el momento implementado.

DÍA: 23/10/2019: El día 23 de octubre se retomará el avance de la creación de la cámara y empezaremos a añadir las cosas faltantes para así dejar ya lista la cámara creada.

DÍA: 30/10/2019: El día 30 de octubre retomaremos el código de transformaciones y lo modificaremos para cada uno de los objetos creados.

DÍA: 06/11/2019: El día 06 de noviembre se empezará la implementación de la traslación de la cámara.

DÍA: 20/11/2019: El día 20 de noviembre se empezará a mirar los últimos detalles de cada uno de los códigos anteriormente implementados.

DÍA: 27/11/2019: El día 27 de noviembre se hará lo final