Informe y documentación del código Computación grafica

Morales V, David 6000408

Universidad Militar Nueva Granada

I. INTRODUCCIÓN

Este documento pretende ser un seguimiento documentado de la clase de computación Grafica, y de sus actividades del día jueves 8 de octubre de 2020.

El objetivo es crear un modelo 3D objeto que vemos cotidiana mente, para ello se utilizara JavaScript principal mente apoyado en 2 librerías, Three.js y CSG.js.

Se realizar el modelo 3D de una lampara (fig. 1), a continuación, se muestra el resultado y procedimiento paso a paso.

II. OBJETIVÓ

El primer objetivo planteado en la clase es realizar un modelo 3d del siguiente objeto.



Fig. 1 Objeto a modelar

III. PROCEDIMIENTO

- Establecer un objeto para replicar su forma en 3D a través de computación grafica.
- Establecer las geometrías básicas que logren replicar la forma del objeto.
 - Cilindros
 - Toroide
 - Esferas
 - Plano
- Crear las geometrías con sus propiedades

```
//CREAR LAS GEOMETRÍAS

var sphereGeometry = new THREE.SphereGeometry(0.1, 32, 32);
var sphereGeometry2 = new THREE.SphereGeometry(0.2, 32, 32);
var sphereGeometry3 = new THREE.SphereGeometry(0.3, 32, 32);
var torusGeometry = new THREE.TorusGeometry(0.4, 0.08, 32, 32);
var planeGeometry = new THREE.PlaneGeometry(0.65, 0.65, 1);
var cylinderGeometry = new THREE.CylinderGeometry(0.04, 0.04, 1, 10);
var cylinderGeometry2 = new THREE.CylinderGeometry(0.04, 0.04, 0.5, 10);
```

· Crear los materiales

```
//CMERR LOS MATERIALES
var material1 = new THREE.MeshStandardNaterial( { color: colorR, metalness: 0.5, roughness: 0.1,
var material2 = material1.clone();
material2.color = colorR;
material2.color = 0.3;
var material3 = material3.clone();
material3.color = colorB;
var material3 = material3.clone();
material3.color = colorB;
var material4 = material3.clone();
material4.color = new THREE dWTy of CR33CCCC );
var material5 = new THREE dWTy of CR33CCCC );
var material5 = new THREE dWTy of CR33CCCC );
var material5 = new THREE dWTy of CR33CCCC );
var material5 = new THREE dwShStandardNaterial( { color: colorR, metalness: 0.5, roughness: 0.1 } );
```

 Se hacen las transformaciones necesarias a las mallas de Threejs

```
//TRASLADAR LAS MALLAS A SU LUGAR ADECUADO

cilindroUno.translateY(0.5);
cilindroDos.translateY(1);

sphere.translateY(1.2);
sphere.translateZ(0.2);

sphere2.translateZ(0.2);

sphere2.translateZ(0.4);

sphere3.translateZ(0.4);

sphere3.translateZ(0.5);

sphere.rotateX(Math.PI /2);
cilindroDos.rotateX(Math.PI / 4.0);

torus1.rotateX( Math.PI/2.0 );
plao1.rotateX( -Math.PI/2 );
```

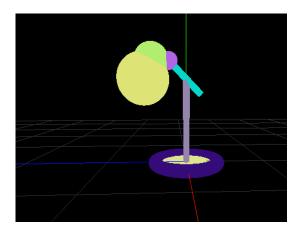
• Creamos los objetos CSG

```
//CONVERTIR A CSG

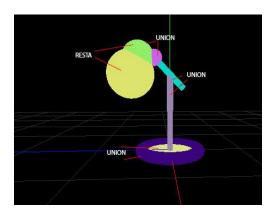
var sphereCSG = THREE.CSG.fromMesh( sphere );
var sphere2CSG = THREE.CSG.fromMesh( sphere2 );
var sphere3CSG = THREE.CSG.fromMesh( sphere3 );

var torus1CSG = THREE.CSG.fromMesh( torus1 );
var plao1CSG = THREE.CSG.fromMesh( plao1 );
var cilindroUnoCSG = THREE.CSG.fromMesh( cilindroUno );
var cilindroDosCSG = THREE.CSG.fromMesh( cilindroDos );
```

 Recreamos el objeto tridimensional con las mallas de Threejs



• Realizamos las operaciones entre los objetos CSG



```
// OPERACIONES

var result1 = sphereCSG.union(sphere2CSG); // UNION DE ESFERA

var result2 = plaotCSG.union( torusICSG); // BASE, TOROIDE Y PLANO

var result3 = result1.subtract( sphere2CSG); // RSTA 3 ESFERAS, LE DA FORMA AL FOCO

var result4 = cllindroUnoCSG.union( cllindroDosCSG); // UNIE LA BASE Y EL FOCO

var unionTotal= result2.union(result3).union(result4); // SE UNEN TODAS LAS PARTES
```

• convertimos a mallas de Threejs nuevamente

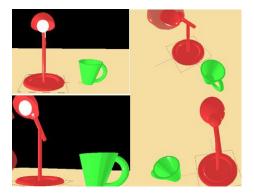
```
//CONVERTIR A MALLA DE THREE JS NUEVAMENTE
var C2T1 = THREE.CSG.toMesh( result1 );
var C2T2 = THREE.CSG.toMesh( result2 );
var C2T3 = THREE.CSG.toMesh( result3 );
var C2T4 = THREE.CSG.toMesh( result4 );
var C2Total = THREE.CSG.toMesh( unionTotal );

C2T1.material = material5;
C2T2.material = material5;
C2T3.material = material5;
C2Total.material = material5;
```

Se añade la unión de figuras a la escena

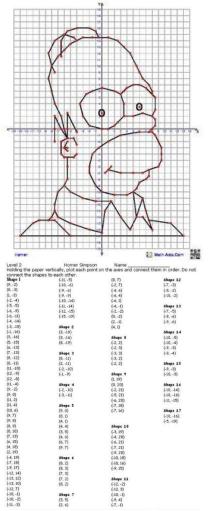
```
scene.add( C2Total );
```

Se realizo el mismo proceso para representar una taza de café y se obtuvo esto:



El siguiente reto fue logar representar otro objeto 3D curvas y el uso de Extrude, para esto fu necesario realizar curvas en 2D de una figura, para este caso se decidio representar el rostro de Homero Simson.

Este trabajo esta basado en la siguiente informacion obtenida de la red.



El primer paso fue fijar cada uno de los puntos por donde pasarían las diferentes curvas, un trabajo extenso, que se evidencia en el archivo Homero simson.html.

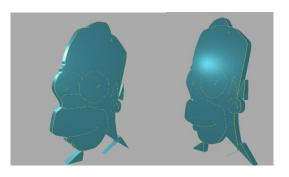
 Una ves fijados los puntos y trazadas las curvas se puede observar lo siguiente



 Para llegar hasta este punto se utilizó la propiedad Shape de Threejs para poder trazar las curvas

```
var shape = new THREE.Shape();
shape.splineThru(curve2D);
var material = new THREE.LineBasicMaterial( { color : '#EED71E' } );
var resolution = 50;
var points = shape.getPoints( resolution );
var geometry = new THREE.BufferGeometry().setFromPoints( points );
var curveObject = new THREE.Line( geometry, material );
scene.add( curveObject );
```

• La intención es llagar a este punto



Para lograr esto se hace uso de la geometria .ExtrudeGeometry, que nos brinda Threejs.

```
var geometryExt = new THREE.ExtrudeGeometry( shape, extrudeSettings );
var geometryExt2 = new THREE.ExtrudeGeometry( shape2, extrudeSettings );
var geometryExt3 = new THREE.ExtrudeGeometry( shape3, extrudeSettings );
var geometryExt4 = new THREE.ExtrudeGeometry( shape4, extrudeSettings );
var geometryExt5 = new THREE.ExtrudeGeometry( shape5, extrudeSettings );
```

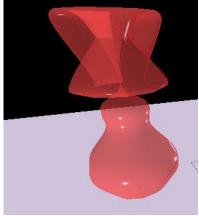
La cual recibe como parametro un objeto con propiedades como steps, amount...entre otros

Finalmete solo se crea una malla con la geometria mencionada anteriormente y se muestra en la escena.

```
var mesh = new THREE.Mesh( geometryExt, materialExt );
```

scene.add(mesh);

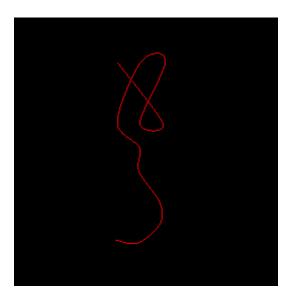
Por ultimo se realizo la siguiente actividad



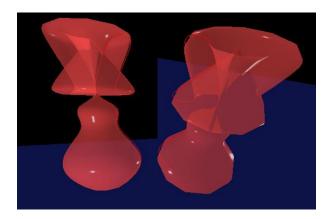
La representacion de una lampara mediante un solido en revolucion.

Se repite el proceso nuevamente, se fijan un par de puntos para que posterior mente una curva los siga.

Se trazo esta curva:



Que al realizar un giro con respeto al eje Y, arroja esto:



Para este caso, la geometría de Threejs que nos permite realizar esto es LatheGeometry.

```
var geometry = new THREE.LatheGeometry(points, 10);
var material = new THREE.MeshStandardMaterial({
    color: 0xAA3333,
    metalness: 0.5,
    roughness: 0.1,
    opacity: 0.75,
    transparent: true
});
var lathe = new THREE.Mesh(geometry, material);
```

Por ultimo se crea una malla y se añade a la ecena

Referencias

- Homer Simpson Coordinate Graphing Picture https://co.pinterest.com/pin/846324954944441006/
- Homer Simpson | Coordinate plane graphing https://co.pinterest.com/pin/503840277052754237/
- https://threejs.org/
- https://www.geogebra.org/graphing?lang=es