Ejercicio Pre parcial

Andrés Hernández López 6000346

Gabriel Eduardo Ávila Buitrago

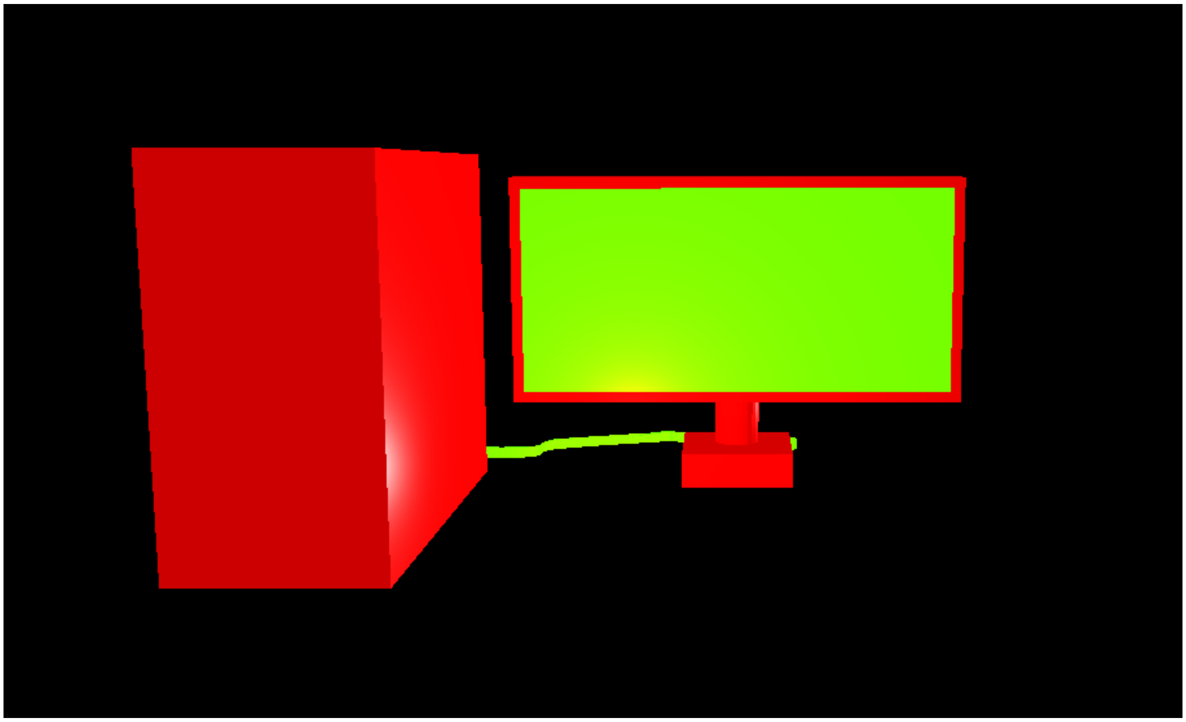
Universidad Militar Nueva Granada

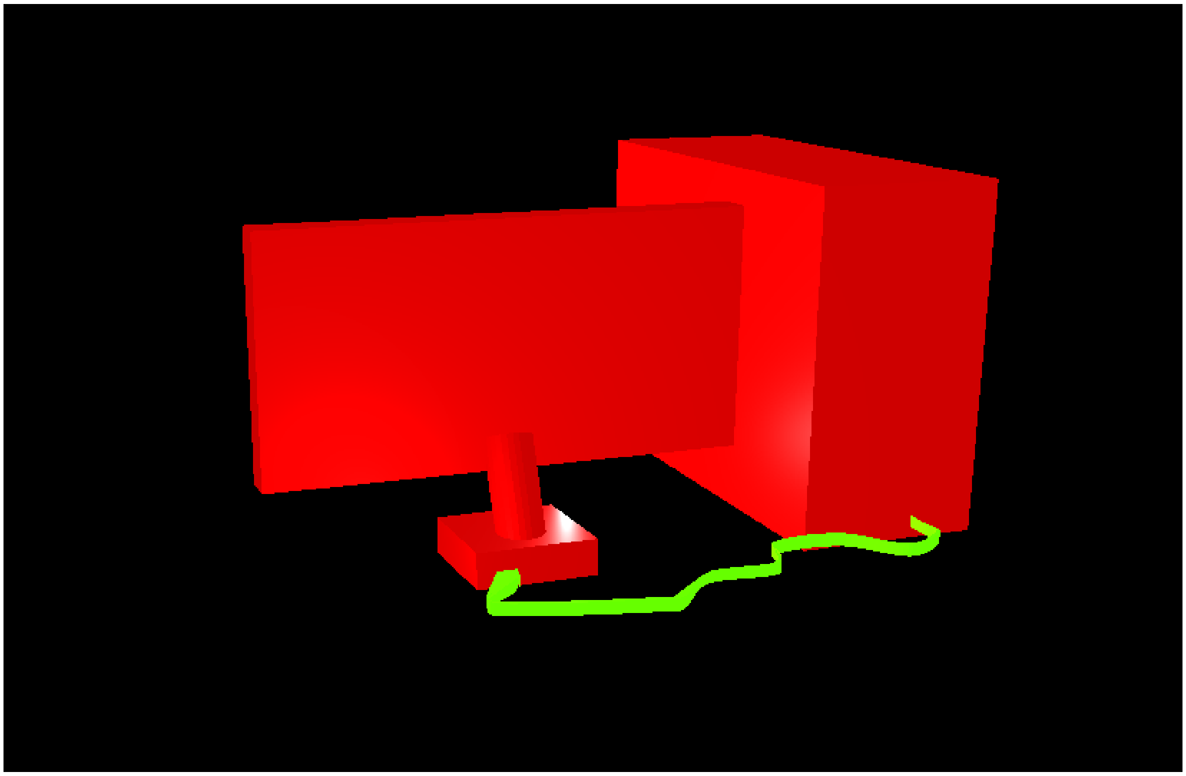
Cundinamarca, Cajicá

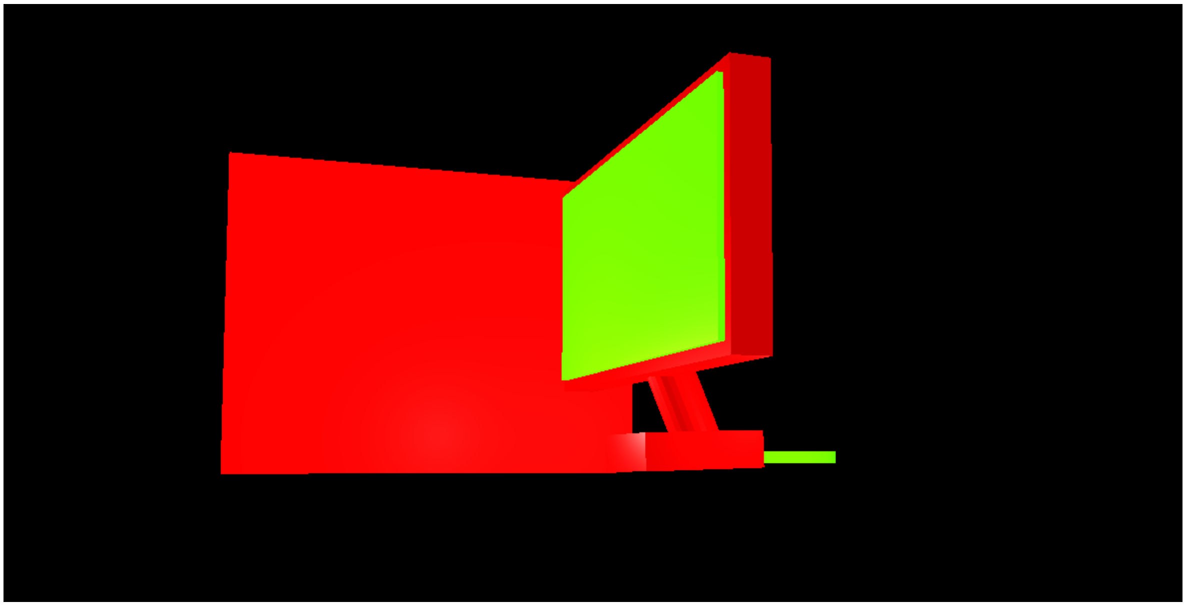
2020

Primer código:

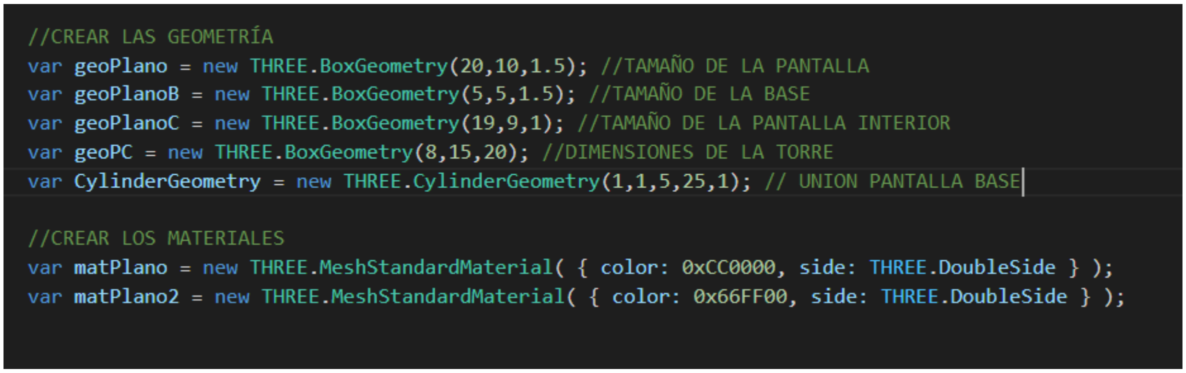
En este, se crearon los objetos y luego se unieron para poder formar la figura de una pantalla y una torre de computador de mesa.







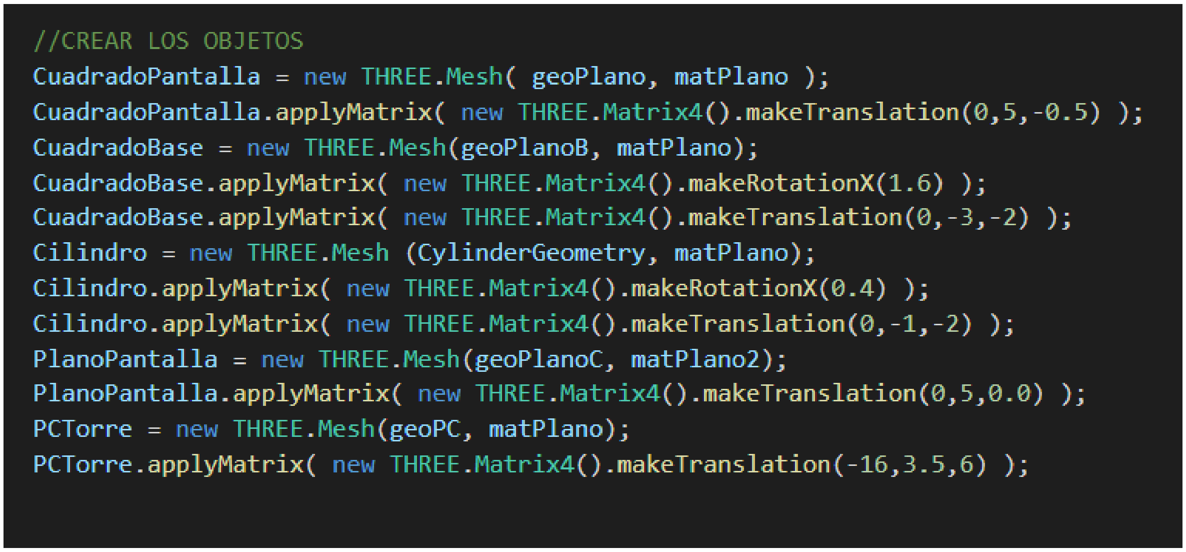
Para la resolución de este código, como primera instancia se crea las características de la ventana, y funciones necesarias, luego creamos los elementos de la escena. Luego creamos las geometrías y materiales.



```
//CREAR LAS GEOMETRÍA
var geoPlano = new THREE.BoxGeometry(20,10,1.5); //TAMAÑO DE LA PANTALLA
var geoPlanoB = new THREE.BoxGeometry(5,5,1.5); //TAMAÑO DE LA BASE
var geoPlanoC = new THREE.BoxGeometry(19,9,1); //TAMAÑO DE LA PANTALLA INTERIOR
var geoPC = new THREE.BoxGeometry(8,15,20); //DIMENSIONES DE LA TORRE
var CylinderGeometry = new THREE.CylinderGeometry(1,1,5,25,1); // UNION PANTALLA BASE

//CREAR LOS MATERIALES
var matPlano = new THREE.MeshStandardMaterial( { color: 0xCC0000, side: THREE.DoubleSide } );
var matPlano2 = new THREE.MeshStandardMaterial( { color: 0x66FF00, side: THREE.DoubleSide } );
```

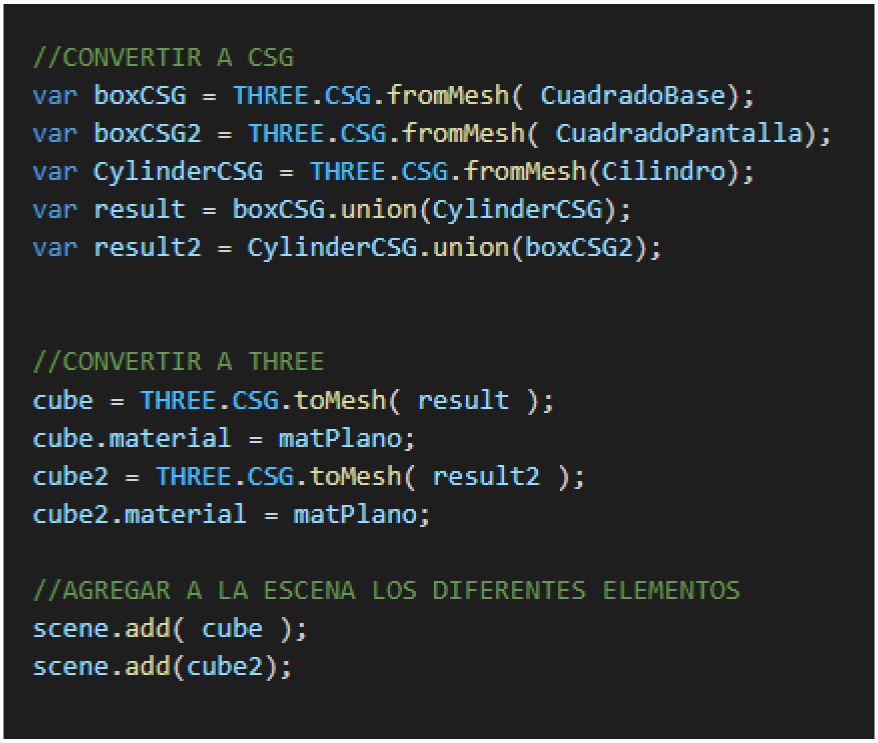
En el caso del material, podemos ver que se usó “MeshStandardMaterial” por lo cual previamente se creó la iluminación necesaria.

Luego se crean los objetos y se ubican en la posición deseada.



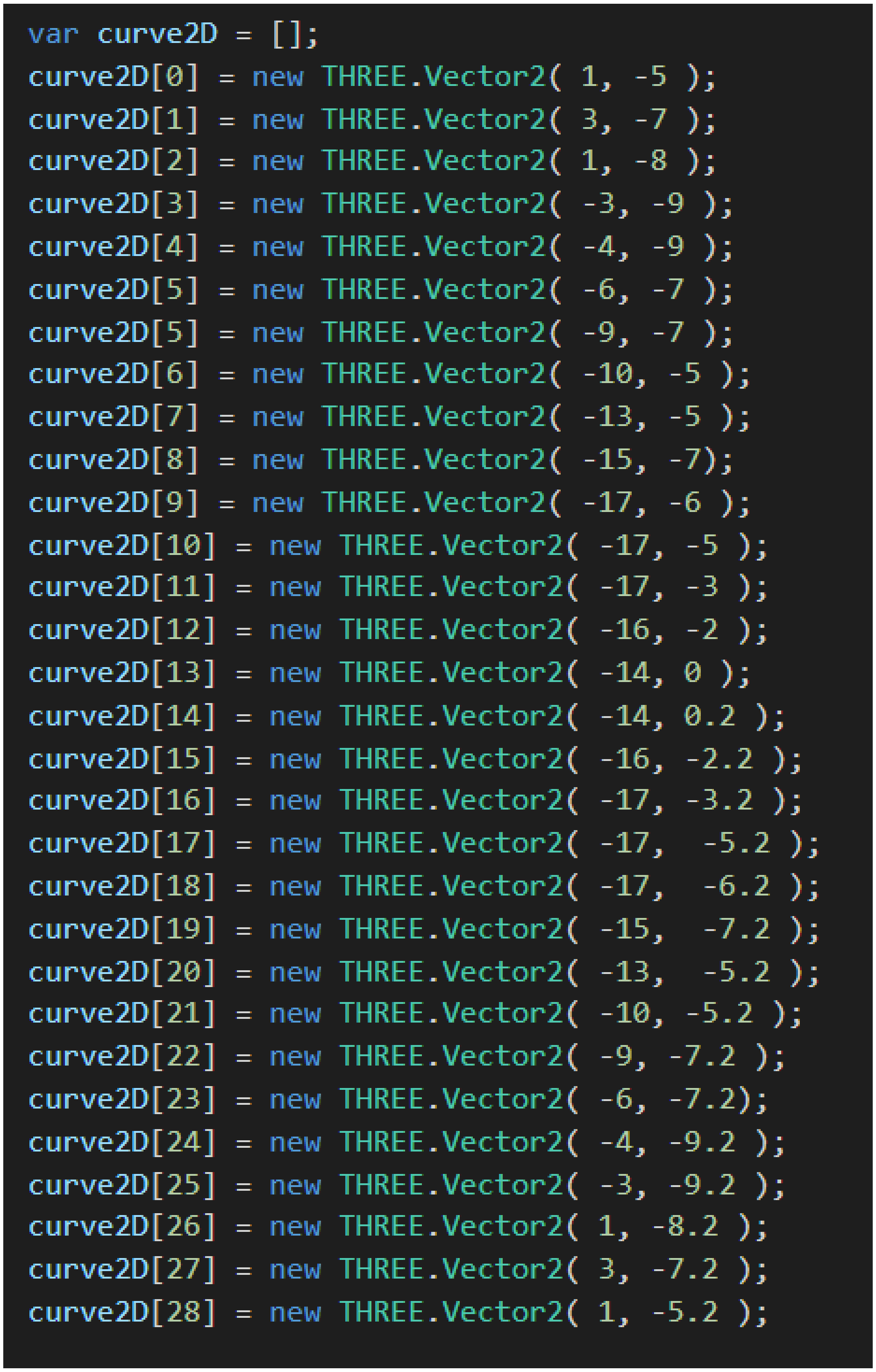
```
//CREAR LOS OBJETOS
CuadradoPantalla = new THREE.Mesh( geoPlano, matPlano );
CuadradoPantalla.applyMatrix( new THREE.Matrix4().makeTranslation(0,5,-0.5) );
CuadradoBase = new THREE.Mesh(geoPlanoB, matPlano);
CuadradoBase.applyMatrix( new THREE.Matrix4().makeRotationX(1.6) );
CuadradoBase.applyMatrix( new THREE.Matrix4().makeTranslation(0,-3,-2) );
Cilindro = new THREE.Mesh (CylinderGeometry, matPlano);
Cilindro.applyMatrix( new THREE.Matrix4().makeRotationX(0.4) );
Cilindro.applyMatrix( new THREE.Matrix4().makeTranslation(0,-1,-2) );
PlanoPantalla = new THREE.Mesh(geoPlanoC, matPlano2);
PlanoPantalla.applyMatrix( new THREE.Matrix4().makeTranslation(0,5,0.0) );
PCTorre = new THREE.Mesh(geoPC, matPlano);
PCTorre.applyMatrix( new THREE.Matrix4().makeTranslation(-16,3.5,6) );
```

Luego, para hacer las operaciones necesarias, debemos convertir los objetos a CSG y posteriormente se convierten en objetos THREE.JSS



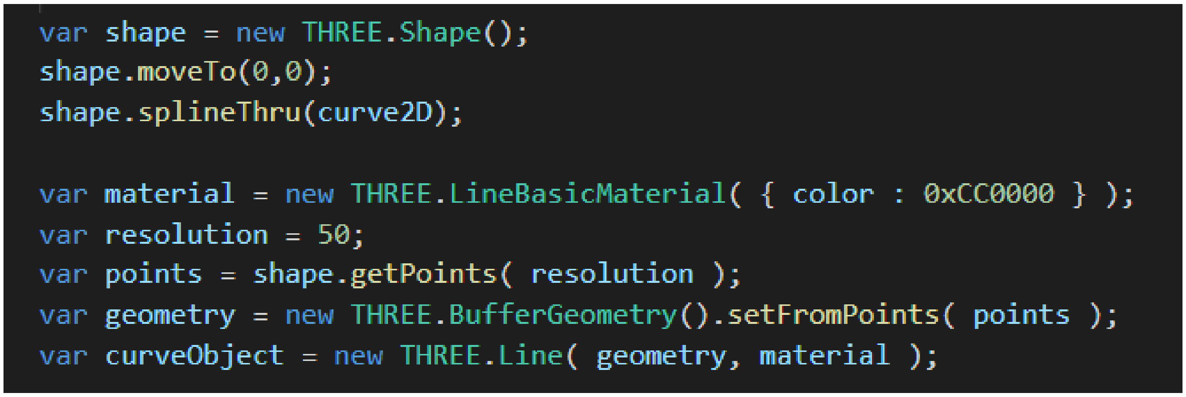
```
//CONVERTIR A CSG
var boxCSG = THREE.CSG.fromMesh( CuadradoBase);
var boxCSG2 = THREE.CSG.fromMesh( CuadradoPantalla);
var CylinderCSG = THREE.CSG.fromMesh(Cilindro);
var result = boxCSG.union(CylinderCSG);
var result2 = CylinderCSG.union(boxCSG2);


//CONVERTIR A THREE
cube = THREE.CSG.toMesh( result );
cube.material = matPlano;
cube2 = THREE.CSG.toMesh( result2 );
cube2.material = matPlano;

//AGREGAR A LA ESCENA LOS DIFERENTES ELEMENTOS
scene.add( cube );
scene.add(cube2);
```

Para el solido de revolucion y la extrusion , se crearon los puntos por donde va a pasar el solid de revolucion.



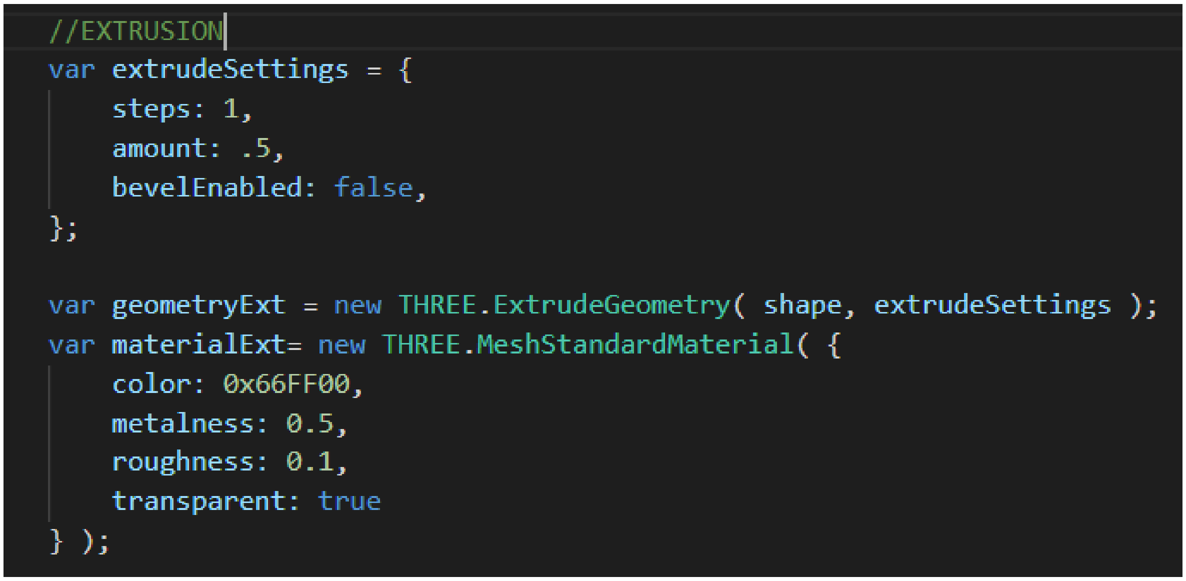
```
var curve2D = [];
curve2D[0] = new THREE.Vector2( 1, -5 );
curve2D[1] = new THREE.Vector2( 3, -7 );
curve2D[2] = new THREE.Vector2( 1, -8 );
curve2D[3] = new THREE.Vector2( -3, -9 );
curve2D[4] = new THREE.Vector2( -4, -9 );
curve2D[5] = new THREE.Vector2( -6, -7 );
curve2D[5] = new THREE.Vector2( -9, -7 );
curve2D[6] = new THREE.Vector2( -10, -5 );
curve2D[7] = new THREE.Vector2( -13, -5 );
curve2D[8] = new THREE.Vector2( -15, -7);
curve2D[9] = new THREE.Vector2( -17, -6 );
curve2D[10] = new THREE.Vector2( -17, -5 );
curve2D[11] = new THREE.Vector2( -17, -3 );
curve2D[12] = new THREE.Vector2( -16, -2 );
curve2D[13] = new THREE.Vector2( -14, 0 );
curve2D[14] = new THREE.Vector2( -14, 0.2 );
curve2D[15] = new THREE.Vector2( -16, -2.2 );
curve2D[16] = new THREE.Vector2( -17, -3.2 );
curve2D[17] = new THREE.Vector2( -17,  -5.2 );
curve2D[18] = new THREE.Vector2( -17,  -6.2 );
curve2D[19] = new THREE.Vector2( -15,  -7.2 );
curve2D[20] = new THREE.Vector2( -13,  -5.2 );
curve2D[21] = new THREE.Vector2( -10, -5.2 );
curve2D[22] = new THREE.Vector2( -9, -7.2 );
curve2D[23] = new THREE.Vector2( -6, -7.2);
curve2D[24] = new THREE.Vector2( -4, -9.2 );
curve2D[25] = new THREE.Vector2( -3, -9.2 );
curve2D[26] = new THREE.Vector2( 1, -8.2 );
curve2D[27] = new THREE.Vector2( 3, -7.2 );
curve2D[28] = new THREE.Vector2( 1, -5.2 );
```

Se le indica al programa de que convierta dicha línea en un sólido.



```
var shape = new THREE.Shape();
shape.moveTo(0,0);
shape.splineThru(curve2D);

var material = new THREE.LineBasicMaterial( { color : 0xCC0000 } );
var resolution = 50;
var points = shape.getPoints( resolution );
var geometry = new THREE.BufferGeometry().setFromPoints( points );
var curveObject = new THREE.Line( geometry, material );
```

Luego se aplica la extrusión, con las características que se deseen (transparencia, rugosidad, etc).



```
//EXTRUSION
var extrudeSettings = {
    steps: 1,
    amount: .5,
    bevelEnabled: false,
};

var geometryExt = new THREE.ExtrudeGeometry( shape, extrudeSettings );
var materialExt= new THREE.MeshStandardMaterial( {
    color: 0x66FF00,
    metalness: 0.5,
    roughness: 0.1,
    transparent: true
} );
```

Finalmente se añade todo lo necesario a la escena.