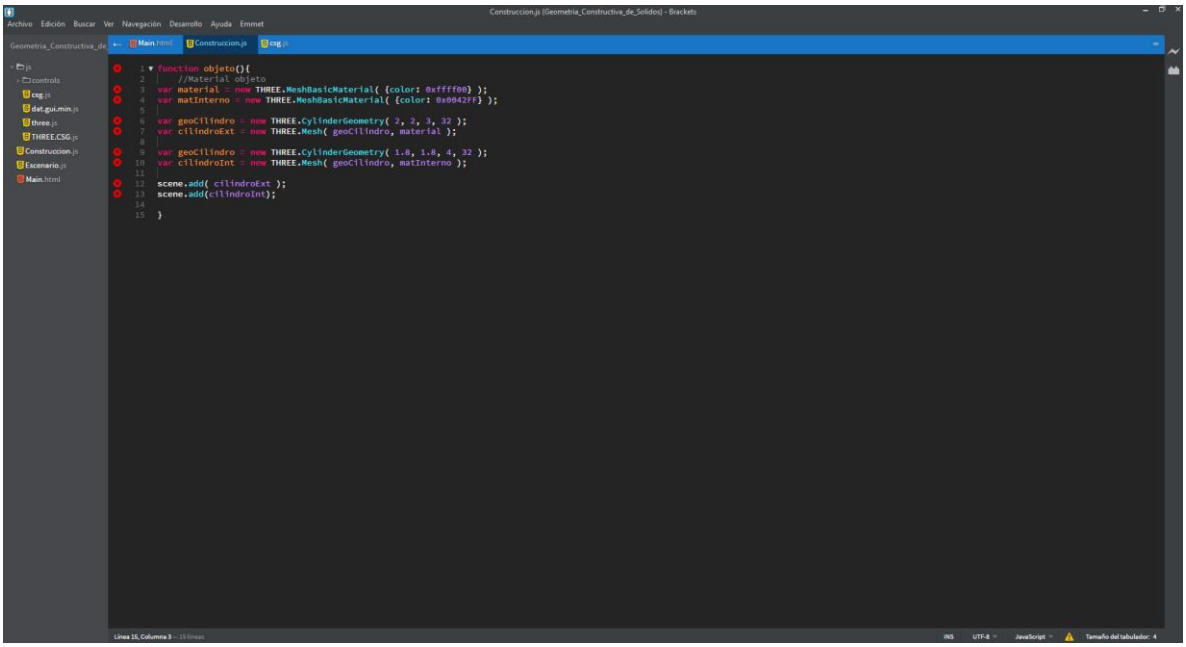
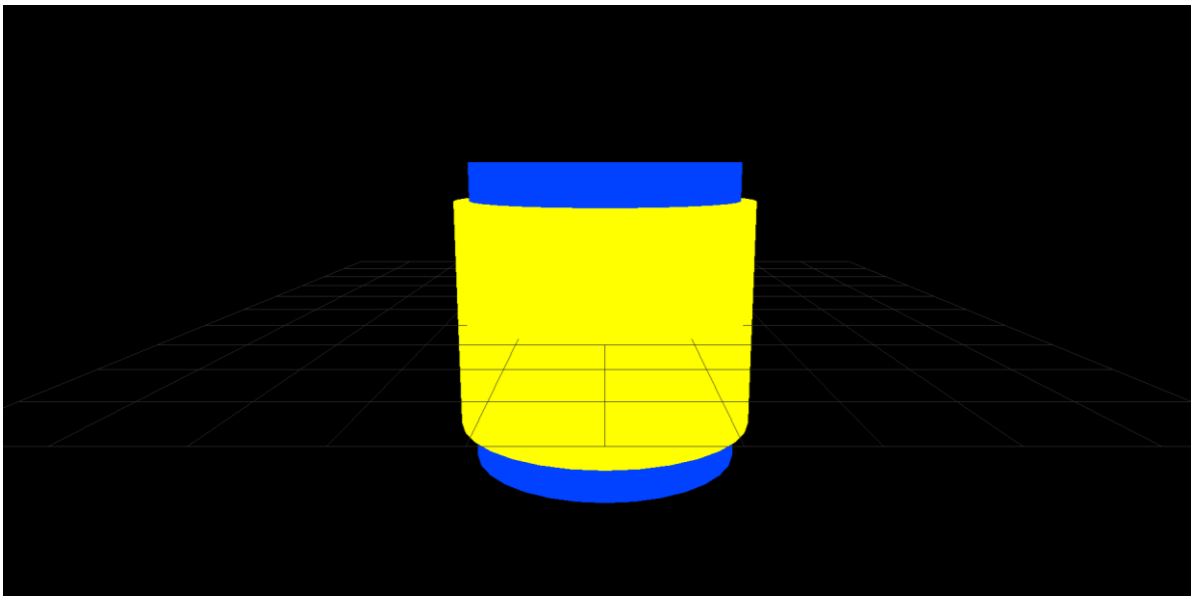
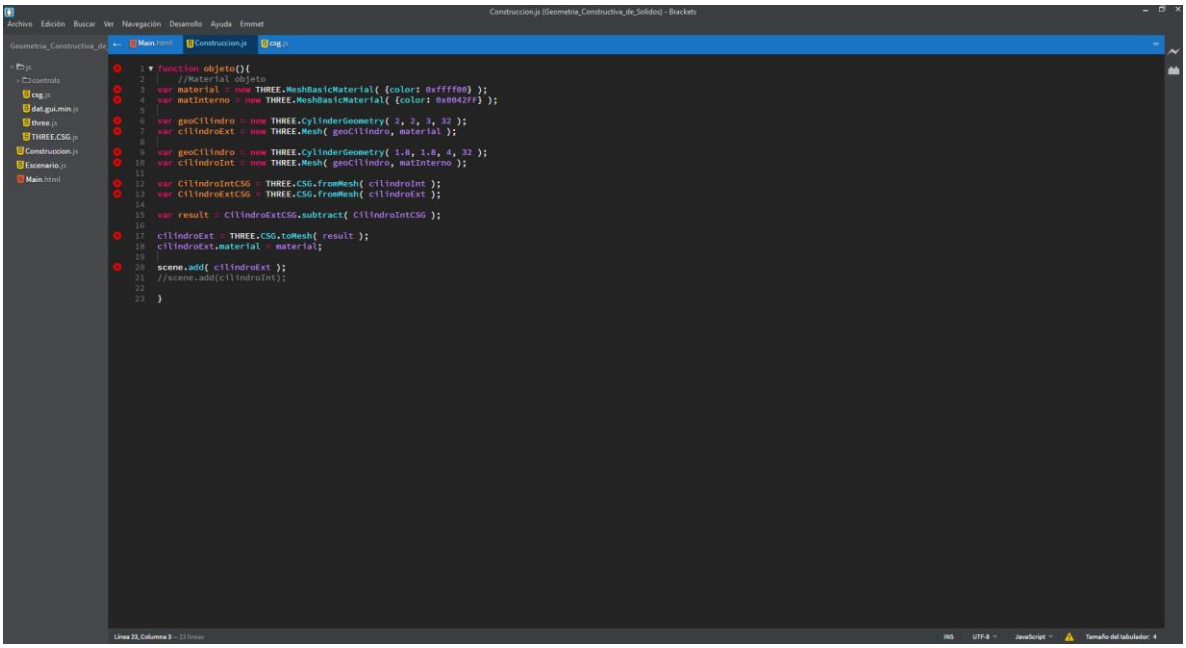
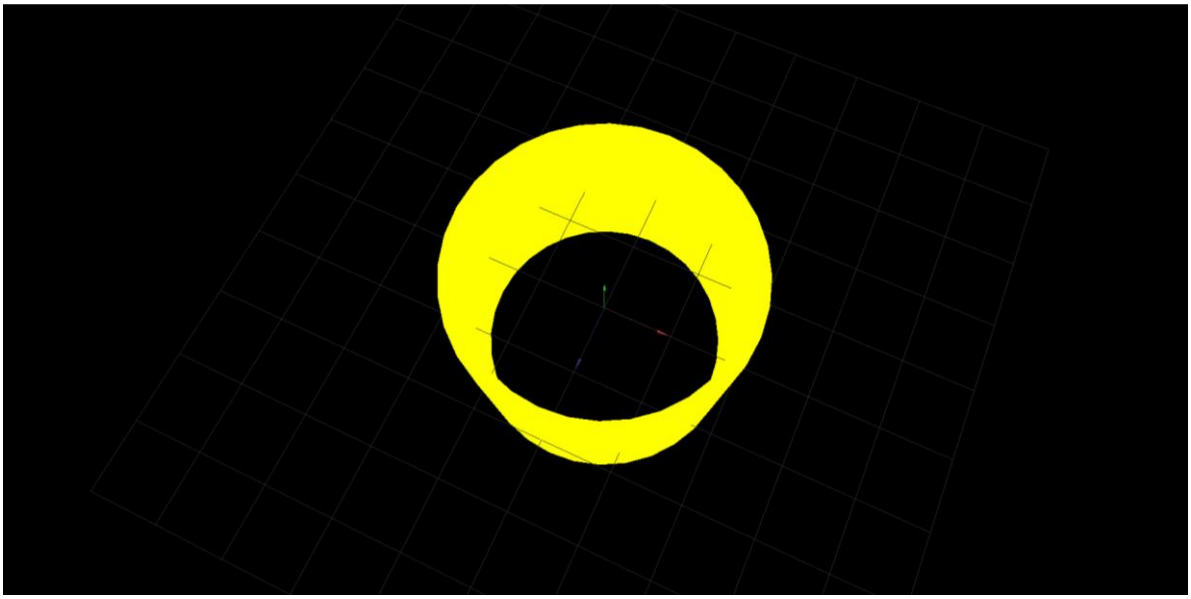
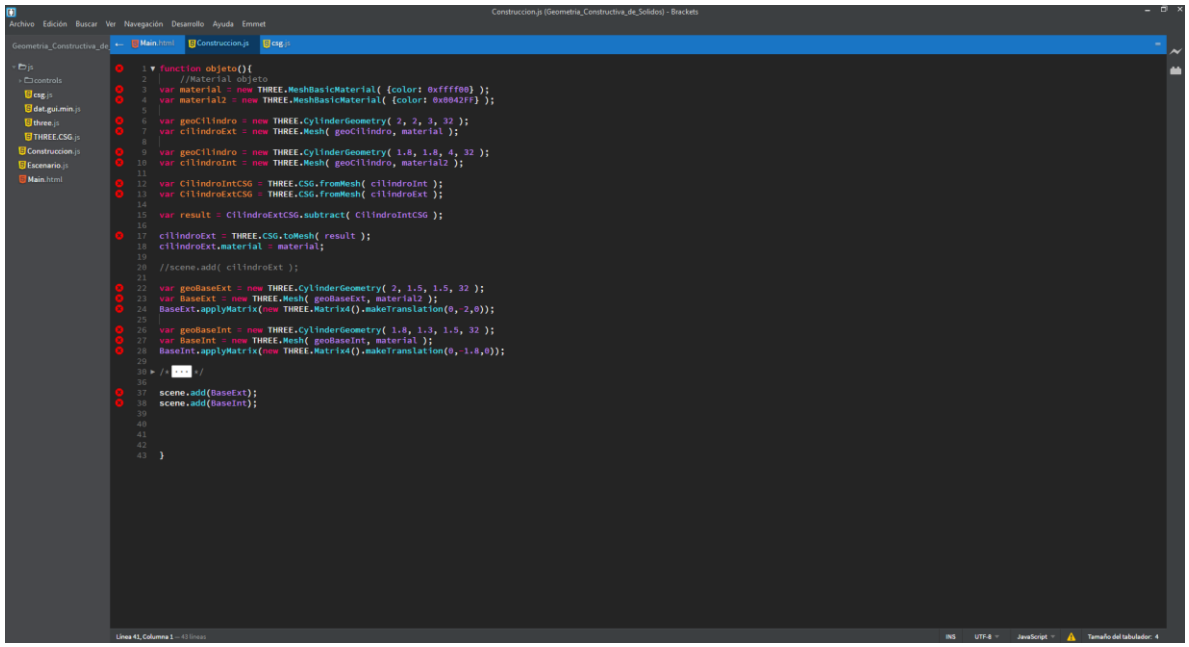
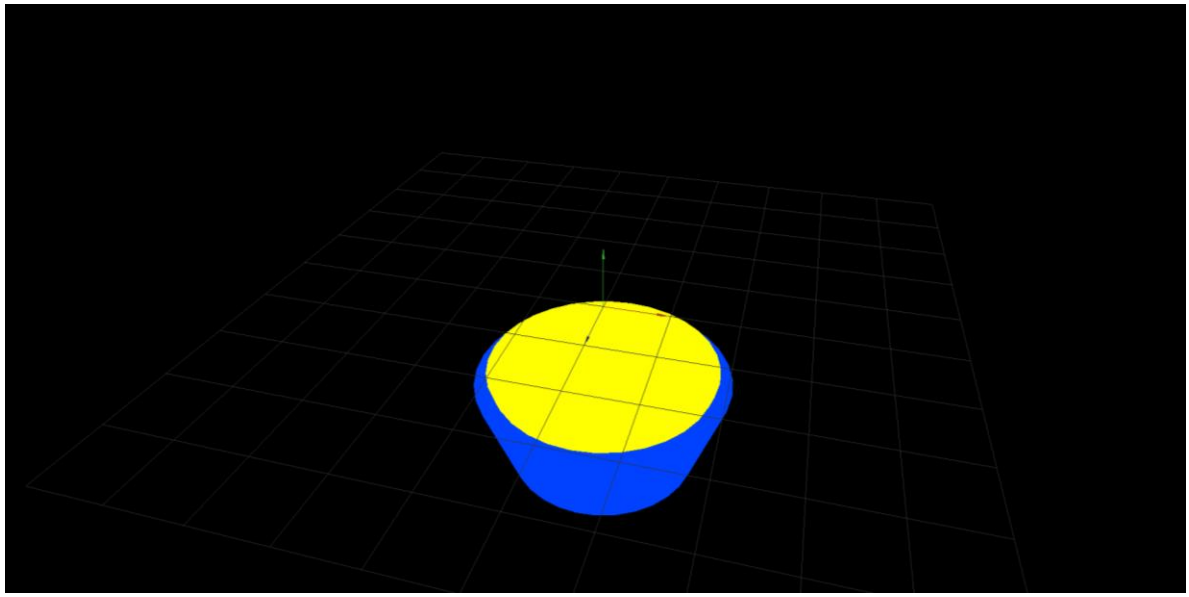

Construccion.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de
Main.html
Construccion.js
csg.js
js
controls
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
Construccion.js
Escenario.js
Main.html
1 function objeto(){
2 //Material objeto
3 var material = new THREE.MeshBasicMaterial({color: 0xffff00});
4 var matInterno = new THREE.MeshBasicMaterial({color: 0x0042FF});
5
6 var geoCilindro = new THREE.CylinderGeometry(2, 2, 3, 32);
7 var cilindroExt = new THREE.Mesh(geoCilindro, material);
8
9 var geoCilindro = new THREE.CylinderGeometry(1.8, 1.8, 4, 32);
10 var cilindroInt = new THREE.Mesh(geoCilindro, matInterno);
11
12 scene.add(cilindroExt);
13 scene.add(cilindroInt);
14
15 }
Línea 15, Columna 3 — 15 líneas
INS
UTF-8
JavaScript
Tamaño del tabulador: 4

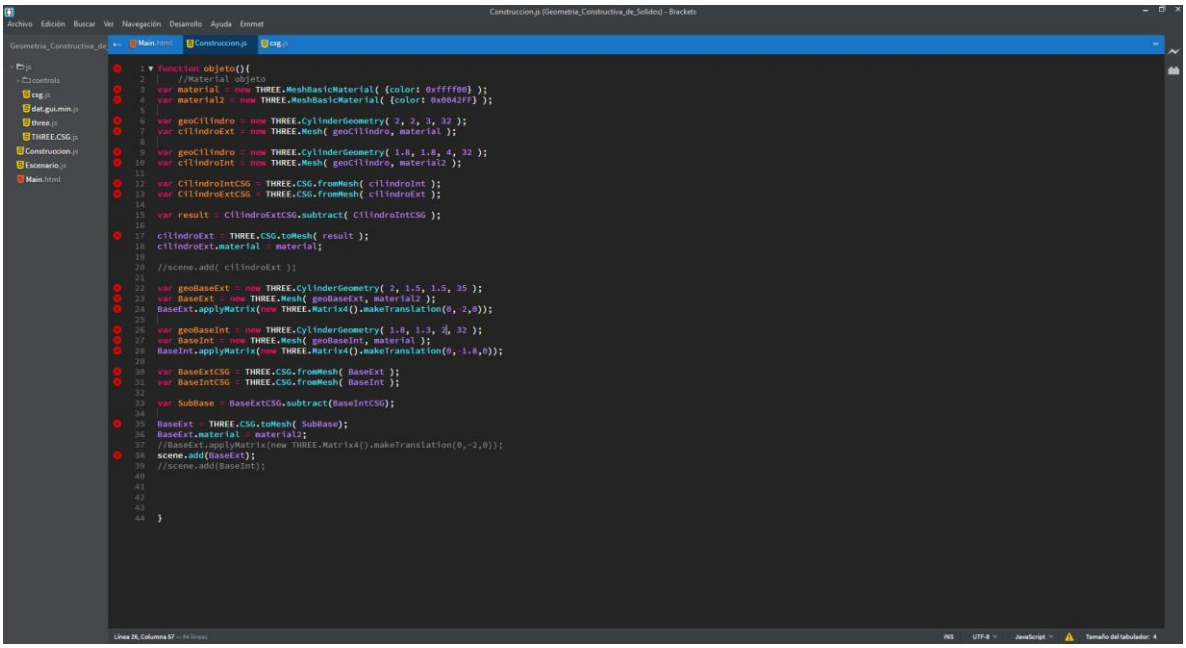


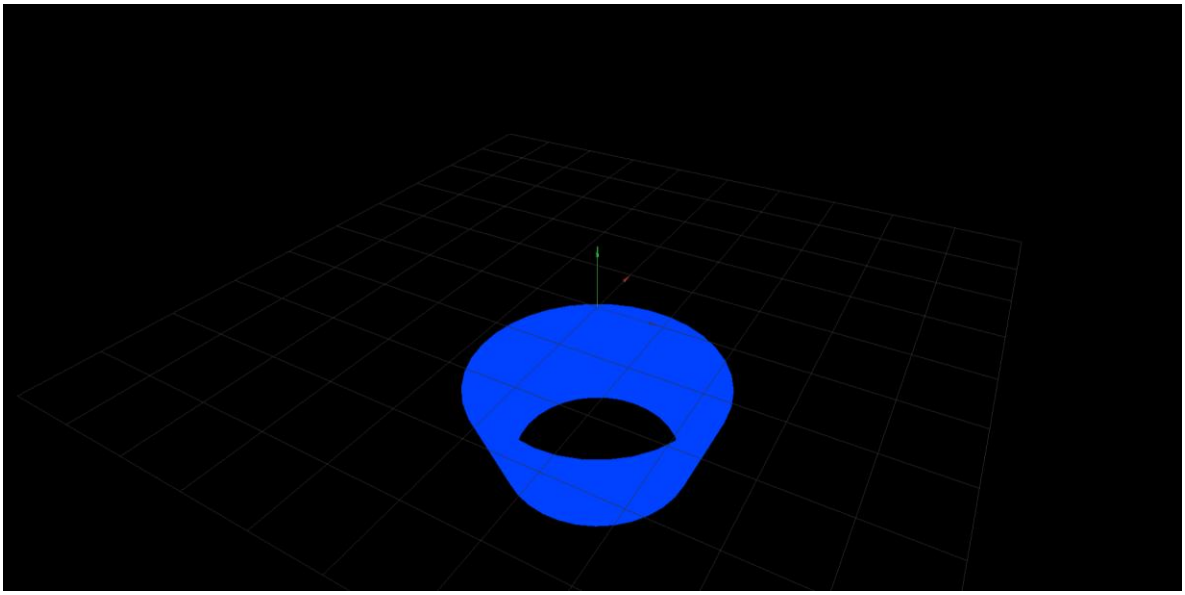

Construccion.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de
Main.html
Construccion.js
csg.js
js
controls
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
Construccion.js
Escenario.js
Main.html
1 function objeto(){
2 //Material objeto
3 var material = new THREE.MeshBasicMaterial({color: 0xffff00});
4 var matInterno = new THREE.MeshBasicMaterial({color: 0x0042FF});
5
6 var geoCilindro = new THREE.CylinderGeometry(2, 2, 3, 32);
7 var cilindroExt = new THREE.Mesh(geoCilindro, material);
8
9 var geoCilindro = new THREE.CylinderGeometry(1.8, 1.8, 4, 32);
10 var cilindroInt = new THREE.Mesh(geoCilindro, matInterno);
11
12 var CilindroIntCSG = THREE.CSG.fromMesh(cilindroInt);
13 var CilindroExtCSG = THREE.CSG.fromMesh(cilindroExt);
14
15 var result = CilindroExtCSG.subtract(CilindroIntCSG);
16
17 cilindroExt = THREE.CSG.toMesh(result);
18 cilindroExt.material = material;
19
20 scene.add(cilindroExt);
21 //scene.add(cilindroInt);
22
23 }
Línea 23, Columna 3 — 23 líneas
INS
UTF-8
JavaScript
Tamaño del tabulador: 4

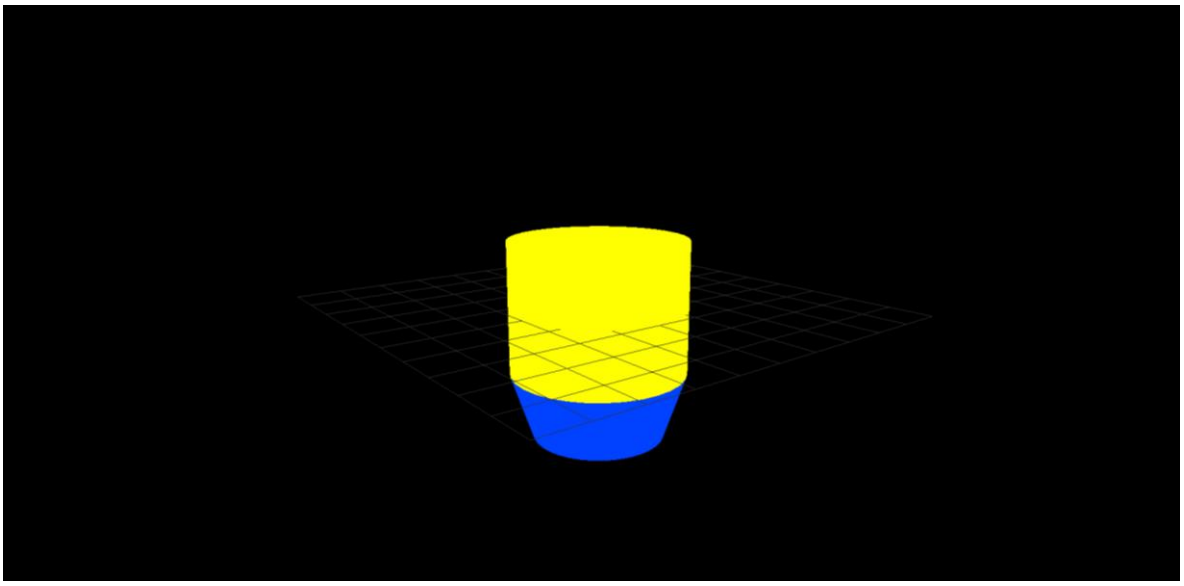


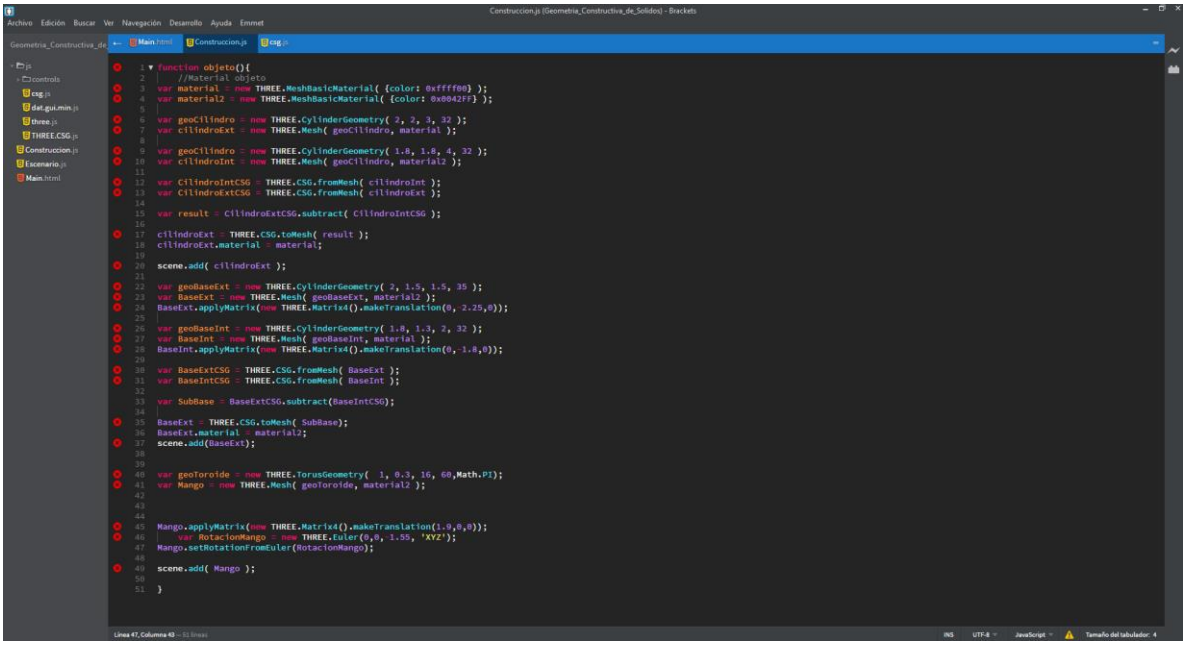

Construccion.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Main.html
Construccion.js
csg.js
function objeto(){
//Material objeto
var material = new THREE.MeshBasicMaterial({color: 0xffff00});
var material2 = new THREE.MeshBasicMaterial({color: 0x0042FF});
var geoCilindro = new THREE.CylinderGeometry(2, 2, 3, 32);
var cilindroExt = new THREE.Mesh(geoCilindro, material);
var geoCilindro = new THREE.CylinderGeometry(1.8, 1.8, 4, 32);
var cilindroInt = new THREE.Mesh(geoCilindro, material2);
var CilindroIntCSG = THREE.CSG.fromMesh(cilindroInt);
var CilindroExtCSG = THREE.CSG.fromMesh(cilindroExt);
var result = CilindroExtCSG.subtract(CilindroIntCSG);
cilindroExt = THREE.CSG.toMesh(result);
cilindroExt.material = material;
//scene.add(cilindroExt);
var geoBaseExt = new THREE.CylinderGeometry(2, 1.5, 1.5, 32);
var BaseExt = new THREE.Mesh(geoBaseExt, material2);
BaseExt.applyMatrix(new THREE.Matrix4().makeTranslation(0, 2,0));
var geoBaseInt = new THREE.CylinderGeometry(1.8, 1.3, 1.5, 32);
var BaseInt = new THREE.Mesh(geoBaseInt, material);
BaseInt.applyMatrix(new THREE.Matrix4().makeTranslation(0, 1.8,0));
/* ... */
scene.add(BaseExt);
scene.add(BaseInt);
}




Construccion.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de
Main.html
Construccion.js
csg.js
js
controls
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
Construccion.js
Escenario.js
Main.html
1 function objeto(){
2 //Material objeto
3 var material = new THREE.MeshBasicMaterial({color: 0xffff00});
4 var material2 = new THREE.MeshBasicMaterial({color: 0x0042FF});
5
6 var geoCilindro = new THREE.CylinderGeometry(2, 2, 3, 32);
7 var cilindroExt = new THREE.Mesh(geoCilindro, material);
8
9 var geoCilindro = new THREE.CylinderGeometry(1.8, 1.8, 4, 32);
10 var cilindroInt = new THREE.Mesh(geoCilindro, material2);
11
12 var CilindroIntCSG = THREE.CSG.fromMesh(cilindroInt);
13 var CilindroExtCSG = THREE.CSG.fromMesh(cilindroExt);
14
15 var result = CilindroExtCSG.subtract(CilindroIntCSG);
16
17 cilindroExt = THREE.CSG.toMesh(result);
18 cilindroExt.material = material;
19
20 //scene.add(cilindroExt);
21
22 var geoBaseExt = new THREE.CylinderGeometry(2, 1.5, 1.5, 35);
23 var BaseExt = new THREE.Mesh(geoBaseExt, material2);
24 BaseExt.applyMatrix(new THREE.Matrix4().makeTranslation(0, 2,0));
25
26 var geoBaseInt = new THREE.CylinderGeometry(1.8, 1.3, 2, 32);
27 var BaseInt = new THREE.Mesh(geoBaseInt, material);
28 BaseInt.applyMatrix(new THREE.Matrix4().makeTranslation(0, 1.8,0));
29
30 var BaseExtCSG = THREE.CSG.fromMesh(BaseExt);
31 var BaseIntCSG = THREE.CSG.fromMesh(BaseInt);
32
33 var SubBase = BaseExtCSG.subtract(BaseIntCSG);
34
35 BaseExt = THREE.CSG.toMesh(SubBase);
36 BaseExt.material = material2;
37 //BaseExt.applyMatrix(new THREE.Matrix4().makeTranslation(0,-2,0));
38 scene.add(BaseExt);
39 //scene.add(BaseInt);
40
41
42
43
44 }
Línea 26, Columna 57 — 44 líneas
INS
UTF-8
JavaScript
Tamaño del tabulador: 4







Construccion.js (Geometria_Constructiva_de_Solidos) - Brackets

Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet

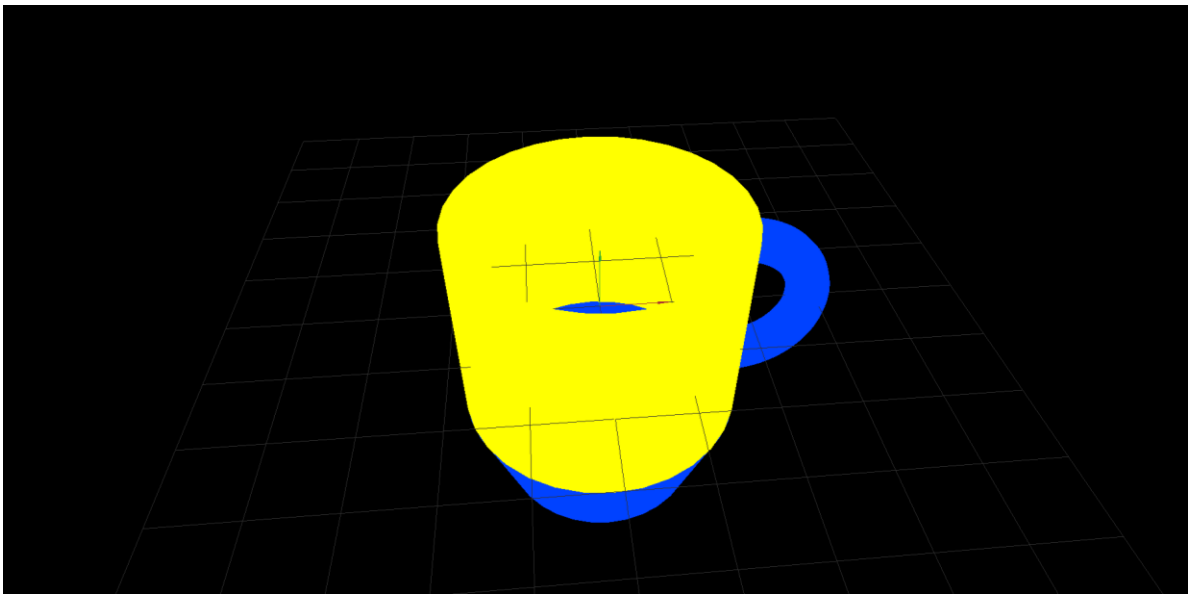
Geometria_Constructiva_de

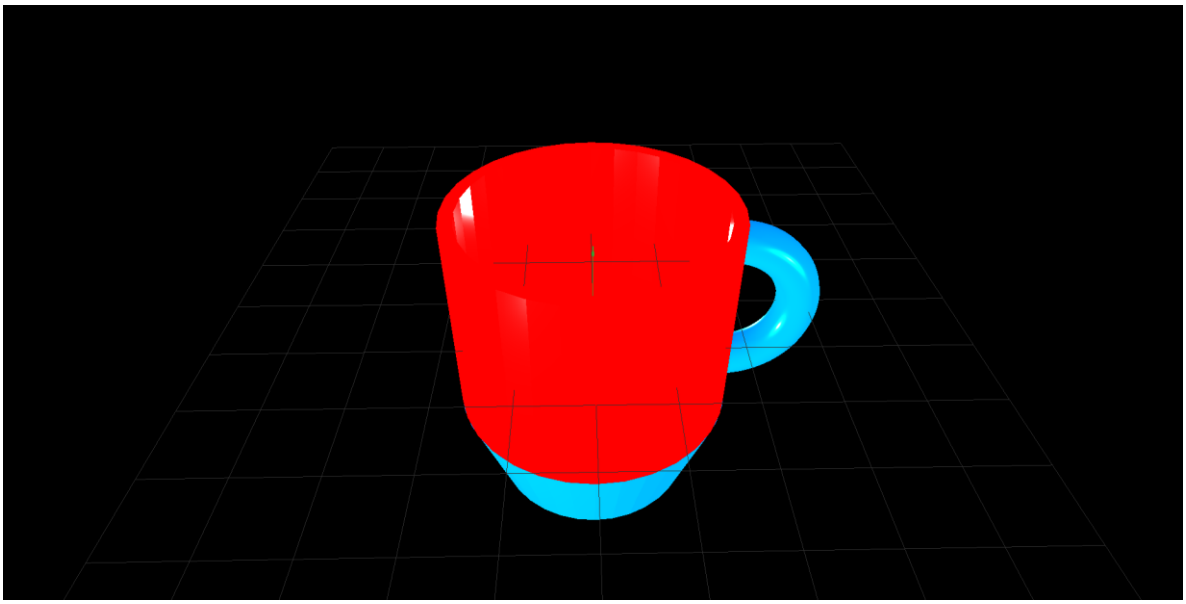
- js
  - controls
  - csg.js
  - dat.gui.min.js
  - three.js
  - THREE.CSG.js
- Construccion.js
- Escenario.js
- Main.html

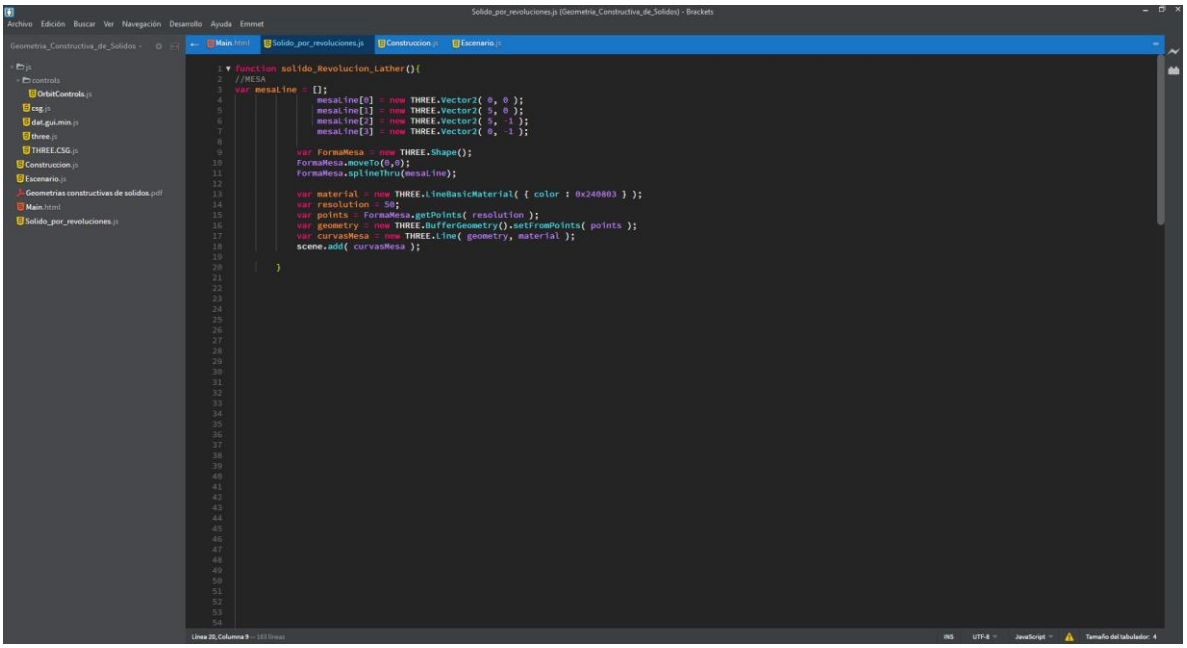
Main.html | Construccion.js | csg.js

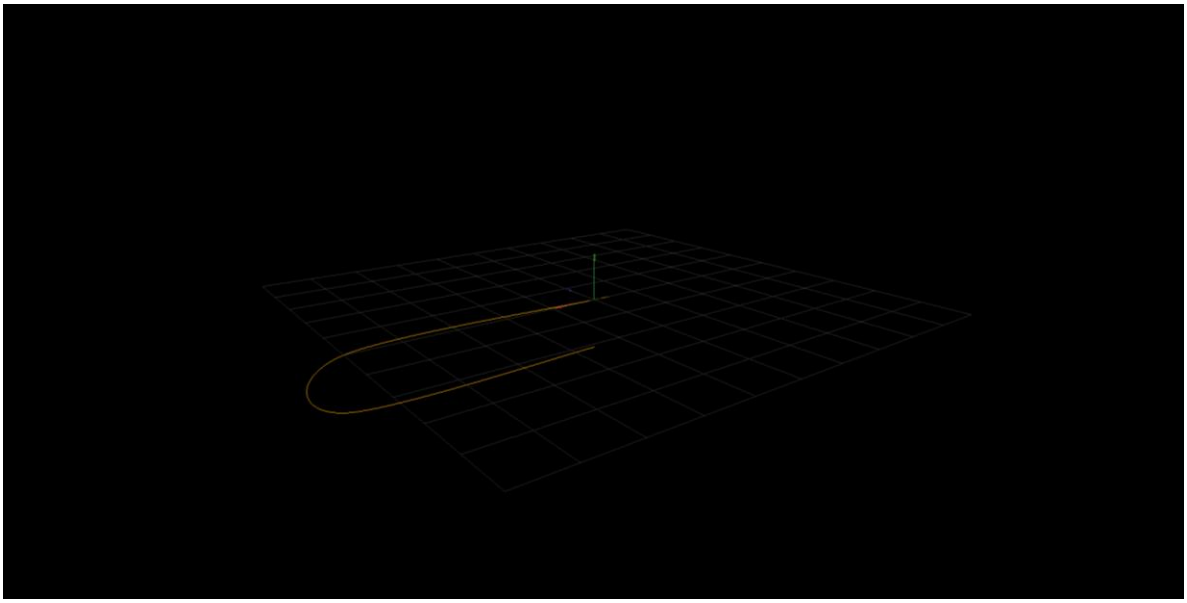
```
function objeto(){
    //Material objeto
var material = new THREE.MeshBasicMaterial( {color: 0xffff00} );
var material2 = new THREE.MeshBasicMaterial( {color: 0x0842FF} );

var geoCilindro = new THREE.CylinderGeometry( 2, 2, 3, 32 );
var cilindroExt = new THREE.Mesh( geoCilindro, material );

var geoCilindro = new THREE.CylinderGeometry( 1.8, 1.8, 4, 32 );
var cilindroInt = new THREE.Mesh( geoCilindro, material2 );

var CilindroIntCSG = THREE.CSG.fromMesh( cilindroInt );
var CilindroExtCSG = THREE.CSG.fromMesh( cilindroExt );

var result = CilindroExtCSG.subtract( CilindroIntCSG );

cilindroExt = THREE.CSG.toMesh( result );
cilindroExt.material = material;

scene.add( cilindroExt );

var geoBaseExt = new THREE.CylinderGeometry( 2, 1.5, 1.5, 35 );
var BaseExt = new THREE.Mesh( geoBaseExt, material2 );
BaseExt.applyMatrix(new THREE.Matrix4().makeTranslation(0, 2.25,0));

var geoBaseInt = new THREE.CylinderGeometry( 1.8, 1.3, 2, 32 );
var BaseInt = new THREE.Mesh( geoBaseInt, material );
BaseInt.applyMatrix(new THREE.Matrix4().makeTranslation(0, -1.8,0));

var BaseExtCSG = THREE.CSG.fromMesh( BaseExt );
var BaseIntCSG = THREE.CSG.fromMesh( BaseInt );

var SubBase = BaseExtCSG.subtract(BaseIntCSG);

BaseExt = THREE.CSG.toMesh( SubBase);
BaseExt.material = material2;
scene.add(BaseExt);


var geoToroide = new THREE.TorusGeometry(  1, 0.3, 16, 60,Math.PI);
var Mango = new THREE.Mesh( geoToroide, material2 );



Mango.applyMatrix(new THREE.Matrix4().makeTranslation(1.9,0,0));
    var RotacionMango = new THREE.Euler(0,0, 1.55, 'XYZ');
Mango.setRotationFromEuler(RotacionMango);

scene.add( Mango );

}
```

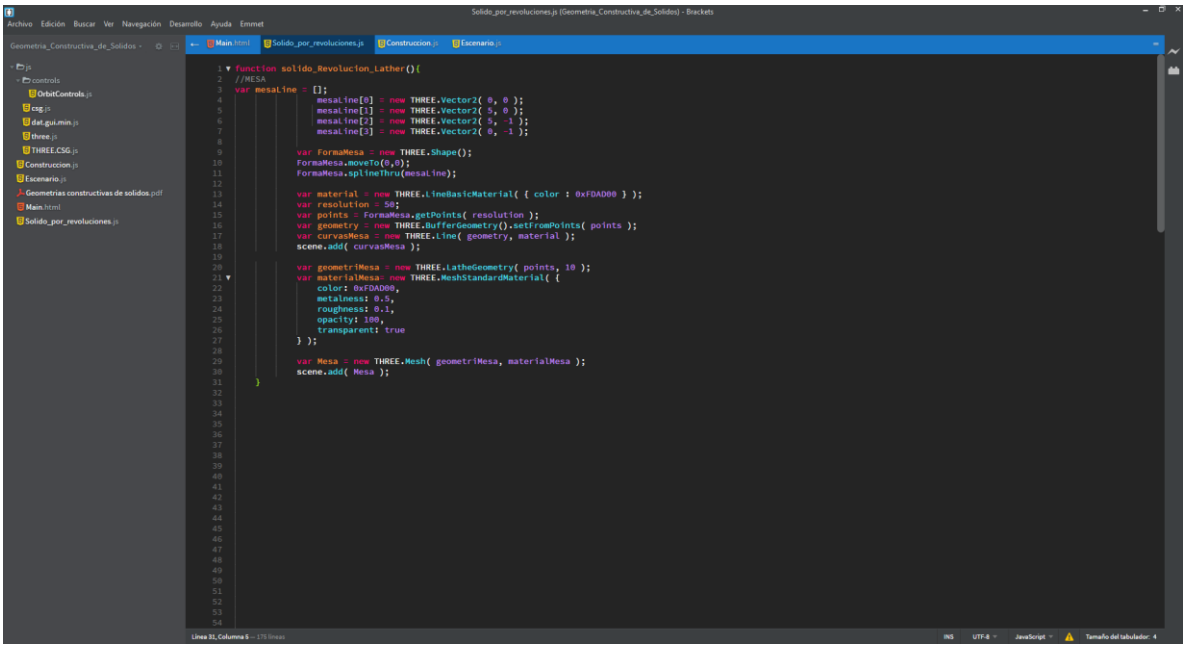
Linea 47, Columna 43 — 51 líneas | INS | UTF-8 | JavaScript | Tamaño del tabulador: 4

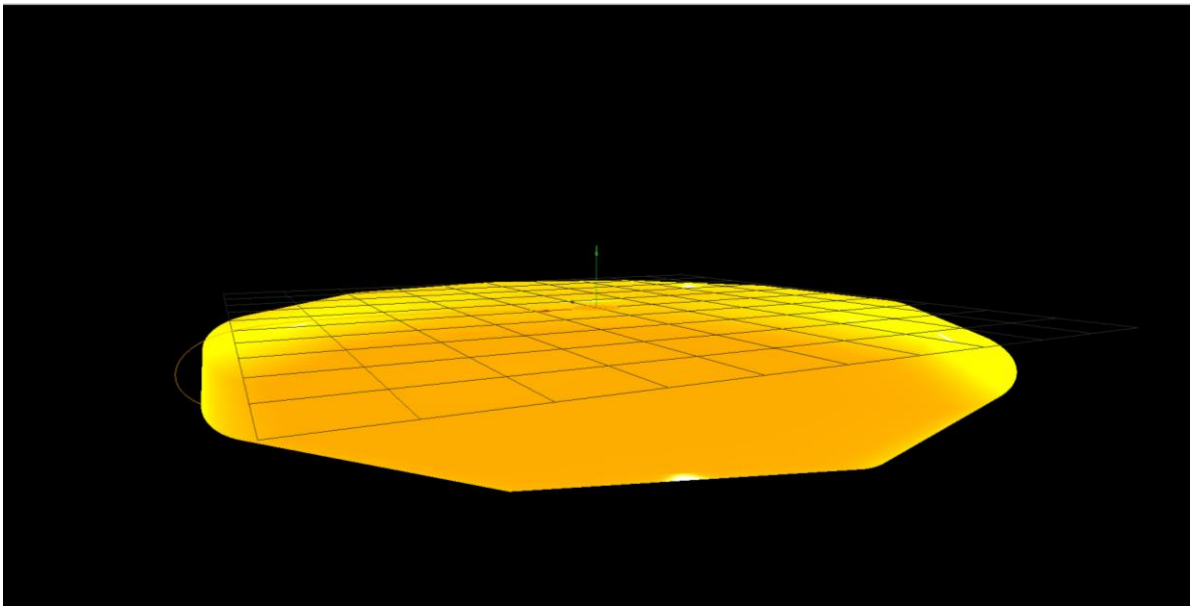


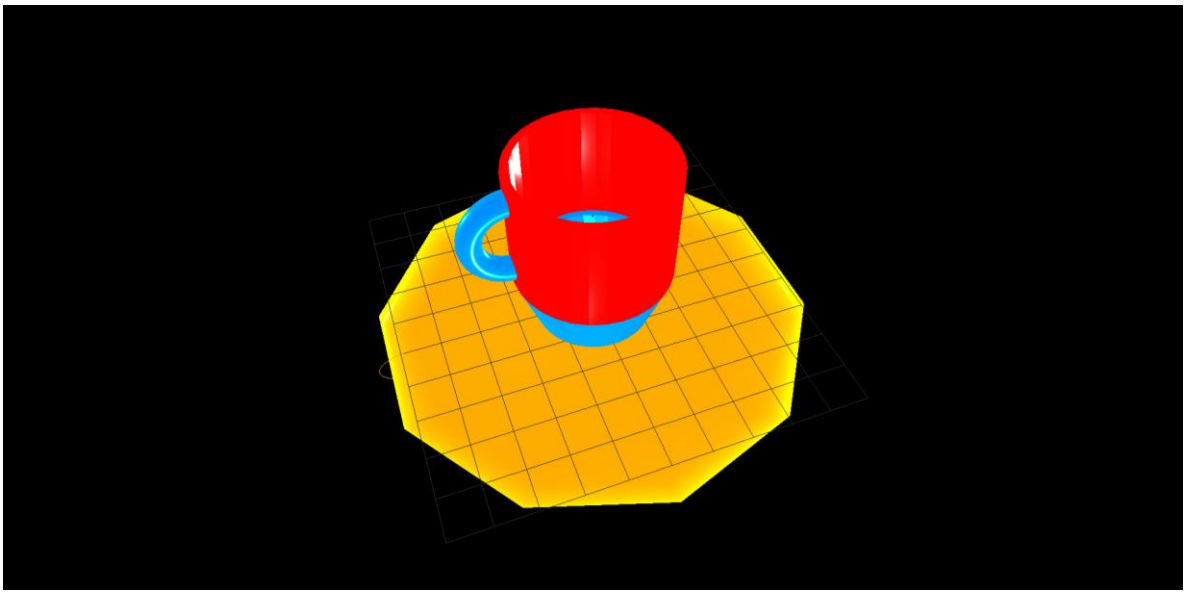


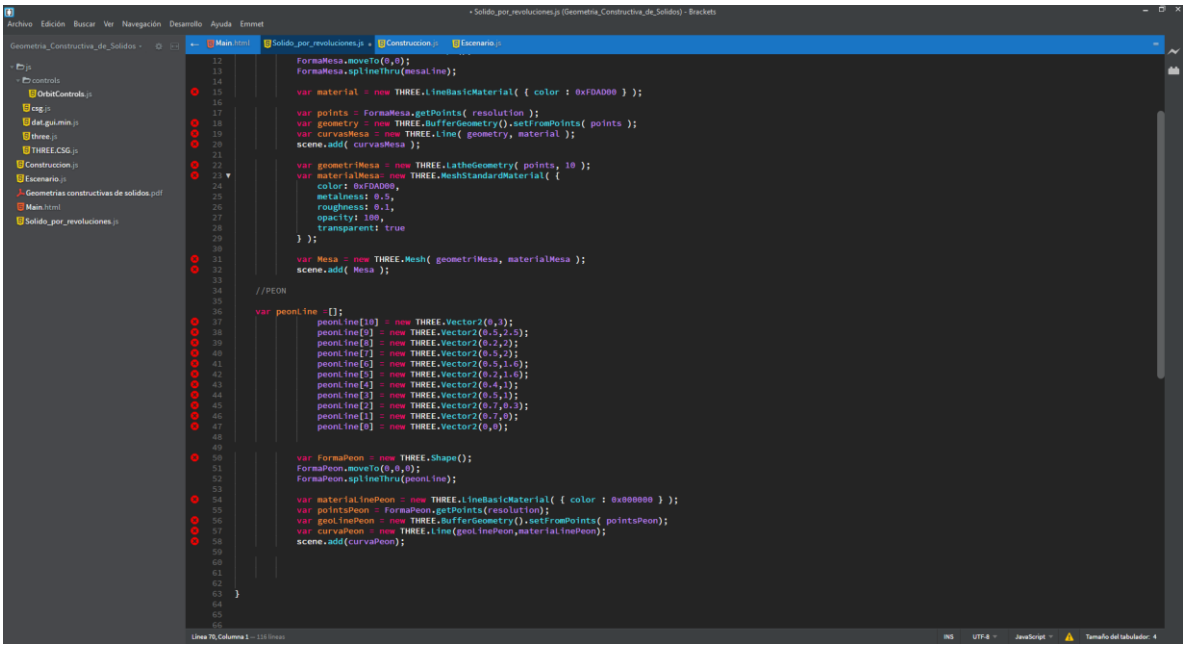

Solido_por_revoluciones.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de_Solidos
Main.html
Solido_por_revoluciones.js
Construccion.js
Escenario.js
function solido_Revolucion_Lather(){
//MESA
var mesaLine = [];
mesaLine[0] = new THREE.Vector2(0, 0);
mesaLine[1] = new THREE.Vector2(5, 0);
mesaLine[2] = new THREE.Vector2(5, -1);
mesaLine[3] = new THREE.Vector2(0, -1);
var FormaMesa = new THREE.Shape();
FormaMesa.moveTo(0,0);
FormaMesa.splineThru(mesaLine);
var material = new THREE.LineBasicMaterial({ color : 0x240803 });
var resolution = 50;
var points = FormaMesa.getPoints(resolution);
var geometry = new THREE.BufferGeometry().setFromPoints(points);
var curvasMesa = new THREE.Line(geometry, material);
scene.add(curvasMesa);
}



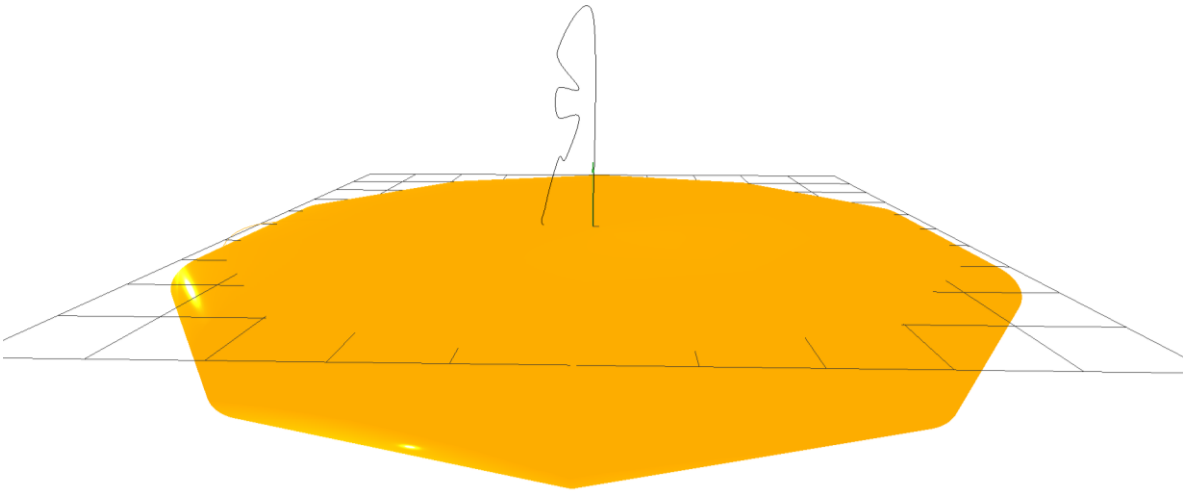

Solido_por_revoluciones.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de_Solidos
Main.html
Solido_por_revoluciones.js
Construccion.js
Escenario.js
function solido_Revolucion_Lather(){
//MESA
var mesaLine = [];
mesaLine[0] = new THREE.Vector2(0, 0);
mesaLine[1] = new THREE.Vector2(5, 0);
mesaLine[2] = new THREE.Vector2(5, -1);
mesaLine[3] = new THREE.Vector2(0, -1);
var FormaMesa = new THREE.Shape();
FormaMesa.moveTo(0,0);
FormaMesa.splineThru(mesaLine);
var material = new THREE.LineBasicMaterial({ color : 0xFDAD00 });
var resolution = 50;
var points = FormaMesa.getPoints(resolution);
var geometry = new THREE.BufferGeometry().setFromPoints(points);
var curvasMesa = new THREE.Line(geometry, material);
scene.add(curvasMesa);
var geometriMesa = new THREE.LatheGeometry(points, 10);
var materialMesa= new THREE.MeshStandardMaterial({
color: 0xFDAD00,
metalness: 0.5,
roughness: 0.1,
opacity: 100,
transparent: true
});
var Mesa = new THREE.Mesh(geometriMesa, materialMesa);
scene.add(Mesa);
}

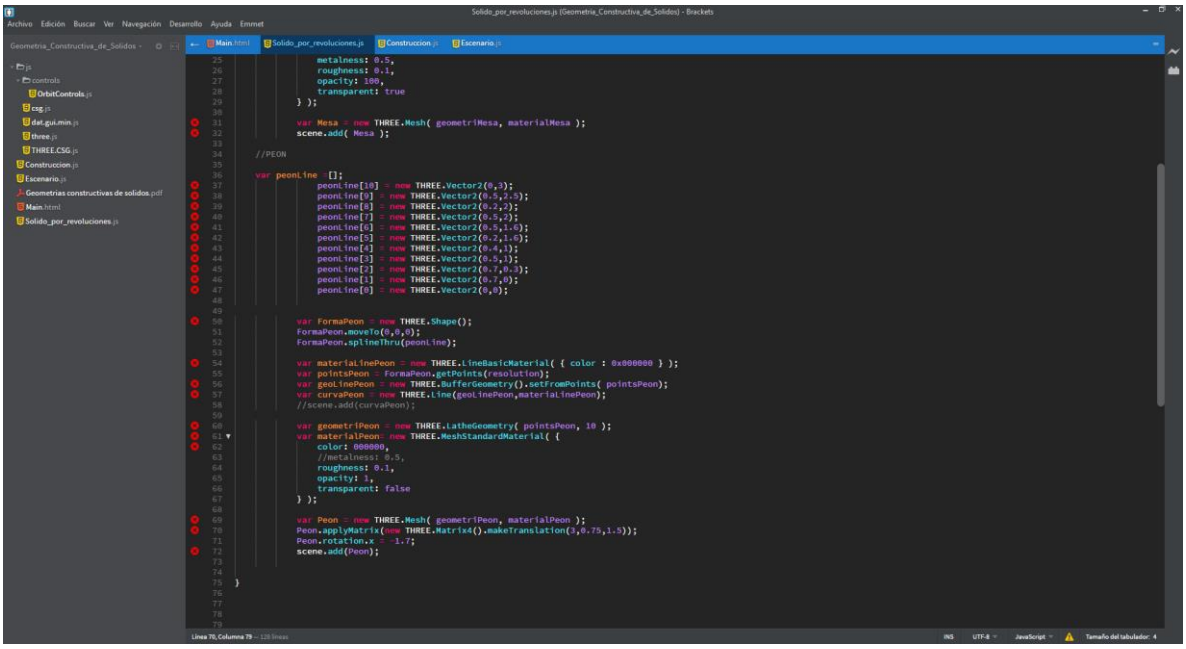


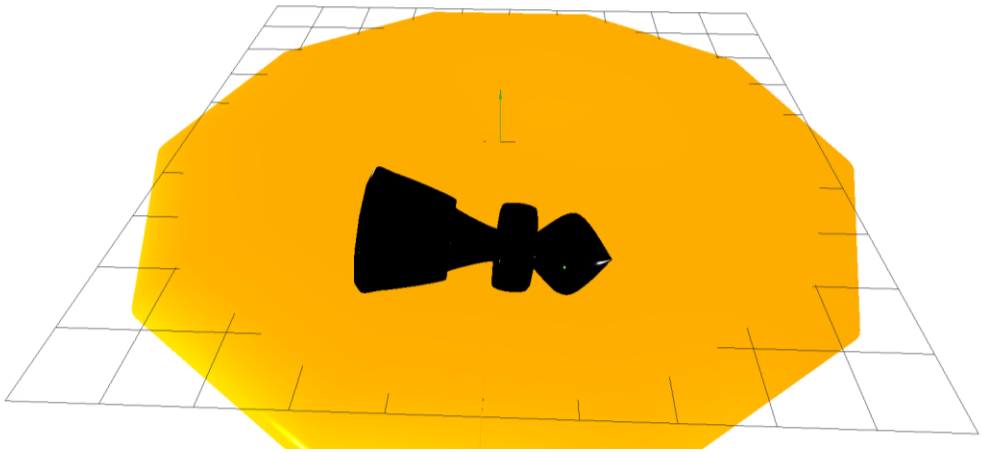


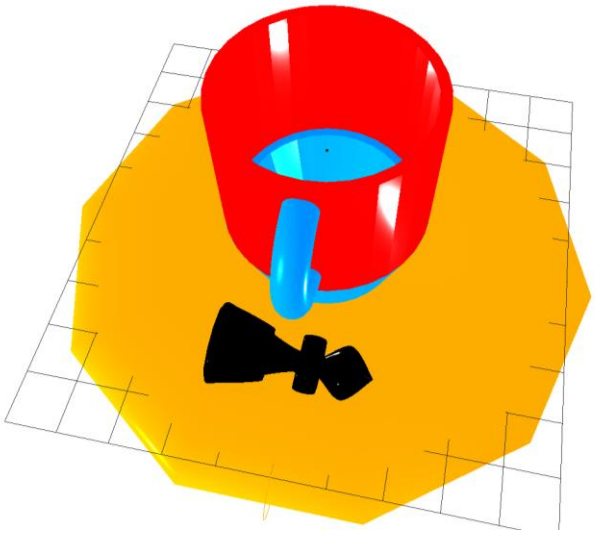


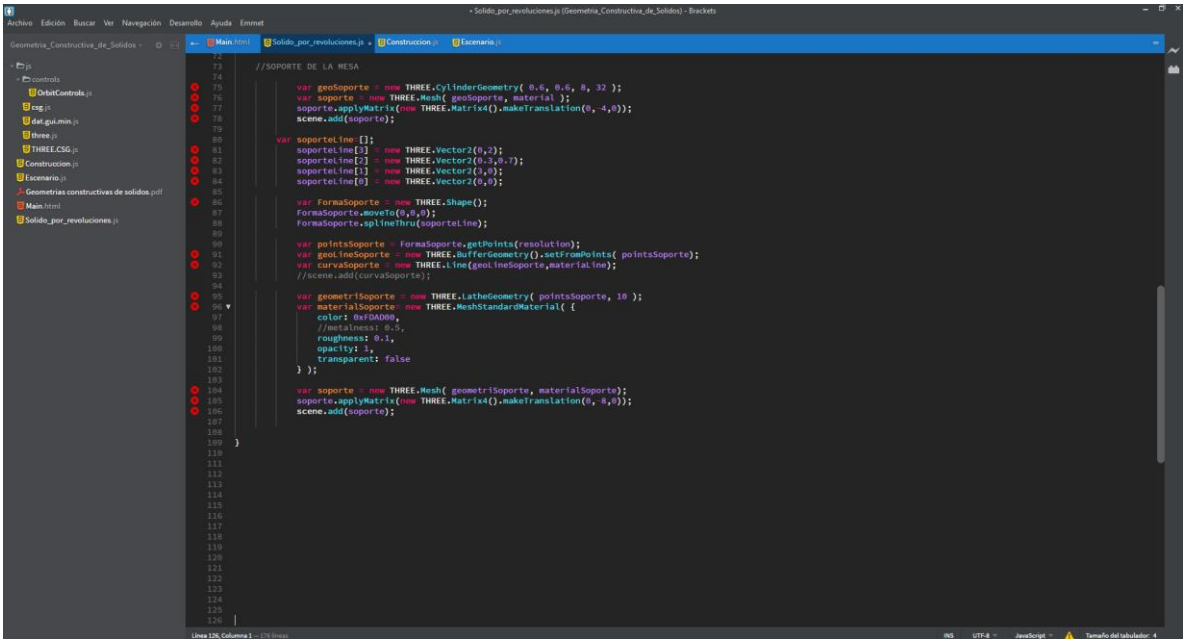
```
            FormaMesa.moveTo(0,0);
            FormaMesa.splineThru(mesaLine);

            var material = new THREE.LineBasicMaterial( { color : 0xFDAD00 } );

            var points = FormaMesa.getPoints( resolution );
            var geometry = new THREE.BufferGeometry().setFromPoints( points );
            var curvasMesa = new THREE.Line( geometry, material );
            scene.add( curvasMesa );

            var geometriMesa = new THREE.LatheGeometry( points, 10 );
            var materialMesa= new THREE.MeshStandardMaterial( {
                color: 0xFDAD00,
                metalness: 0.5,
                roughness: 0.1,
                opacity: 100,
                transparent: true
            } );

            var Mesa = new THREE.Mesh( geometriMesa, materialMesa );
            scene.add( Mesa );

    //PEON

    var peonLine =[];
                peonLine[10] = new THREE.Vector2(0,3);
                peonLine[9] = new THREE.Vector2(0.5,2.5);
                peonLine[8] = new THREE.Vector2(0.2,2);
                peonLine[7] = new THREE.Vector2(0.9,2);
                peonLine[6] = new THREE.Vector2(0.5,1.6);
                peonLine[5] = new THREE.Vector2(0.2,1.6);
                peonLine[4] = new THREE.Vector2(0.4,1);
                peonLine[3] = new THREE.Vector2(0.5,1);
                peonLine[2] = new THREE.Vector2(0.7,0.3);
                peonLine[1] = new THREE.Vector2(0.7,0);
                peonLine[0] = new THREE.Vector2(0,0);


            var FormaPeon = new THREE.Shape();
            FormaPeon.moveTo(0,0,0);
            FormaPeon.splineThru(peonLine);

            var materialinePeon = new THREE.LineBasicMaterial( { color : 0x000000 } );
            var pointsPeon = FormaPeon.getPoints(resolution);
            var geoLinePeon = new THREE.BufferGeometry().setFromPoints( pointsPeon);
            var curvaPeon = new THREE.Line(geoLinePeon,materialinePeon);
            scene.add(curvaPeon);


}
```

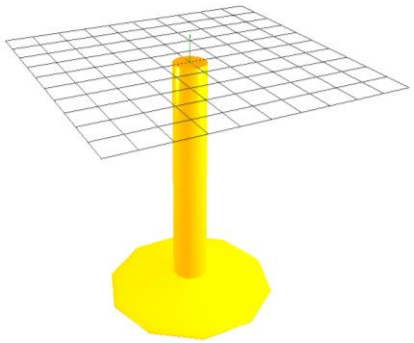


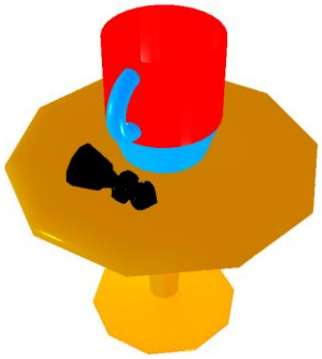

Solido_por_revoluciones.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de_Solidos
Main.html
Solido_por_revoluciones.js
Construccion.js
Escenario.js
Geometrias constructivas de solidos.pdf
OrbitControls.js
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
metalness: 0.5,
roughness: 0.1,
opacity: 100,
transparent: true
var Mesa = new THREE.Mesh(geometriMesa, materialMesa);
scene.add(Mesa);
//PEON
var peonLine = [];
peonLine[10] = new THREE.Vector2(0,3);
peonLine[9] = new THREE.Vector2(0.5,2.5);
peonLine[8] = new THREE.Vector2(0.2,2);
peonLine[7] = new THREE.Vector2(0.5,2);
peonLine[6] = new THREE.Vector2(0.5,1.6);
peonLine[5] = new THREE.Vector2(0.2,1.6);
peonLine[4] = new THREE.Vector2(0.4,1);
peonLine[3] = new THREE.Vector2(0.5,1);
peonLine[2] = new THREE.Vector2(0.7,0.3);
peonLine[1] = new THREE.Vector2(0.7,0);
peonLine[0] = new THREE.Vector2(0,0);
var FormaPeon = new THREE.Shape();
FormaPeon.moveTo(0,0,0);
FormaPeon.splineThru(peonLine);
var materialLinePeon = new THREE.LineBasicMaterial({ color : 0x000000 });
var pointsPeon = FormaPeon.getPoints(resolution);
var geoLinePeon = new THREE.BufferGeometry().setFromPoints(pointsPeon);
var curvaPeon = new THREE.Line(geoLinePeon,materialLinePeon);
//scene.add(curvaPeon);
var geometriPeon = new THREE.LatheGeometry(pointsPeon, 10);
var materialPeon= new THREE.MeshStandardMaterial({
color: 000000,
//metalness: 0.5,
roughness: 0.1,
opacity: 1,
transparent: false
});
var Peon = new THREE.Mesh(geometriPeon, materialPeon);
Peon.applyMatrix(new THREE.Matrix4().makeTranslation(3,0.75,1.5));
Peon.rotation.x = 1.7;
scene.add(Peon);
}
Línea 70, Columna 79 — 128 líneas
INS
UTF-8
JavaScript
Tamaño del tabulador: 4

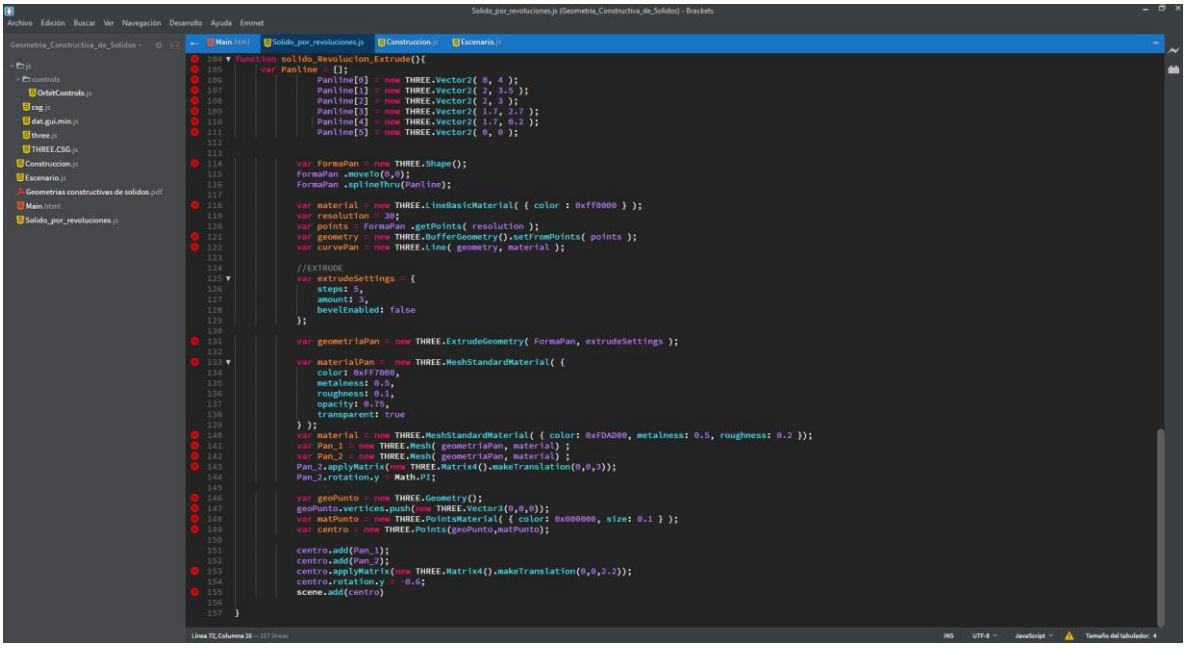





• Solido_por_revoluciones.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de_Solidos
js
controls
OrbitControls.js
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
Construccion.js
Escenario.js
Geometrias constructivas de solidos.pdf
Main.html
Solido_por_revoluciones.js
Main.html
Solido_por_revoluciones.js
Construccion.js
Escenario.js
//SOPORTE DE LA MESA
var geoSoporte = new THREE.CylinderGeometry(0.6, 0.6, 8, 32);
var soporte = new THREE.Mesh(geoSoporte, material);
soporte.applyMatrix(new THREE.Matrix4().makeTranslation(0, -4,0));
scene.add(soporte);
var soporteLine=[];
soporteLine[3] = new THREE.Vector2(0,2);
soporteLine[2] = new THREE.Vector2(0.3,0.7);
soporteLine[1] = new THREE.Vector2(3,0);
soporteLine[0] = new THREE.Vector2(0,0);
var FormaSoporte = new THREE.Shape();
FormaSoporte.moveTo(0,0,0);
FormaSoporte.splineThru(soporteLine);
var pointsSoporte = FormaSoporte.getPoints(resolution);
var geoLineSoporte = new THREE.BufferGeometry().setFromPoints(pointsSoporte);
var curvaSoporte = new THREE.Line(geoLineSoporte,materialLine);
//scene.add(curvaSoporte);
var geometriSoporte = new THREE.LatheGeometry(pointsSoporte, 10);
var materialSoporte= new THREE.MeshStandardMaterial({
color: 0xFDAD00,
//metalness: 0.5,
roughness: 0.1,
opacity: 1,
transparent: false
});
var soporte = new THREE.Mesh(geometriSoporte, materialSoporte);
soporte.applyMatrix(new THREE.Matrix4().makeTranslation(0, 0,0));
scene.add(soporte);
}
Línea 126, Columna 1
INS
UTF-8
JavaScript
Tamaño del tabulador: 4






Solido_por_revoluciones.js (Geometria_Constructiva_de_Solidos) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda Emmet
Geometria_Constructiva_de_Solidos
js
controls
OrbitControls.js
csg.js
dat.gui.min.js
three.js
THREE.CSG.js
Construccion.js
Escenario.js
Geometrias constructivas de solidos.pdf
Main.html
Solido_por_revoluciones.js
Main.html
Solido_por_revoluciones.js
Construccion.js
Escenario.js
function solido_Revolucion_Extrude(){
var Panline = [];
Panline[0] = new THREE.Vector2(0, 4);
Panline[1] = new THREE.Vector2(2, 3.5);
Panline[2] = new THREE.Vector2(2, 3);
Panline[3] = new THREE.Vector2(1.7, 2.7);
Panline[4] = new THREE.Vector2(1.7, 0.2);
Panline[5] = new THREE.Vector2(0, 0);
var FormaPan = new THREE.Shape();
FormaPan .moveTo(0,0);
FormaPan .splineThru(Panline);
var material = new THREE.LineBasicMaterial({ color : 0xff0000 });
var resolution = 30;
var points = FormaPan .getPoints(resolution);
var geometry = new THREE.BufferGeometry().setFromPoints(points);
var curvePan = new THREE.Line(geometry, material);
//EXTRUDE
var extrudeSettings = {
steps: 5,
amount: 3,
bevelEnabled: false
};
var geometriaPan = new THREE.ExtrudeGeometry(FormaPan, extrudeSettings);
var materialPan = new THREE.MeshStandardMaterial({
color: 0xFF7000,
metalness: 0.5,
roughness: 0.1,
opacity: 0.75,
transparent: true
});
var material = new THREE.MeshStandardMaterial({ color: 0xFDAD00, metalness: 0.5, roughness: 0.2 });
var Pan_1 = new THREE.Mesh(geometriaPan, material) ;
var Pan_2 = new THREE.Mesh(geometriaPan, material) ;
Pan_2.applyMatrix(new THREE.Matrix4().makeTranslation(0,0,3));
Pan_2.rotation.y = Math.PI;
var geoPunto = new THREE.Geometry();
geoPunto.vertices.push(new THREE.Vector3(0,0,0));
var matPunto = new THREE.PointsMaterial({ color: 0x000000, size: 0.1 });
var centro = new THREE.Points(geoPunto,matPunto);
centro.add(Pan_1);
centro.add(Pan_2);
centro.applyMatrix(new THREE.Matrix4().makeTranslation(0,0,2.2));
centro.rotation.y = -0.6;
scene.add(centro)
}
Línea 72, Columna 15 — 157 líneas
INS
UTF-8
JavaScript
Tamaño del tabulador: 4