

Implementación de Booleanos, Extrude y Curvas en Revolución

Arturo Pinzón Granados
6000335unimilitar.edu.co
Profesor: Gabriel Ávila Buitrago

I. INTRODUCCIÓN

Se utilizara la operación entre booleanos para agregar, sustraer y obtener la diferencia de sólidos, de igual manera a partir de curvas crearemos un sólido y le realizaremos un extrude o la revolucionaremos para crear un sólido.

A. Marco teórico

Tipos de operaciones que se pueden realizar gracias a los booleanos:

- Unión, recorta las áreas compartidas de las superficies seleccionadas y crea un sólido individual a partir de las áreas no compartidas.
- Diferencia, recorta las áreas compartidas de las superficies seleccionadas con otro conjunto de superficies.
- Intersección, recorta las áreas no compartidas de las superficies seleccionadas.
- Partición, parte las áreas compartidas de las superficies seleccionadas y crea individuales a partir de las partes compartidas.

Funciones de booleanos:

- El primer paso es Convertir a CSG con `THREE.CSG.fromMesh`.
- Luego realizar las operaciones.
 - Union
 - Subtract
 - Intersect
- Para finalizar hay que regresar los objetos de nuevo a sus geometrías de Three js con `THREE.CSG.toMesh`

Extrude:

- Los objetos con extrude pueden pasar de una dimensión a una geometría 3D.

Curvas en Revolución:

- Las curvas en revolución se producen cuando giramos una línea alrededor de su eje, se genera

una superficie de revolución. La línea que genera la superficie se llama generatriz. Las circunferencias perpendiculares al eje se llaman paralelos de la superficie. Los planos que contienen al eje cortan a la superficie determinando los meridianos.

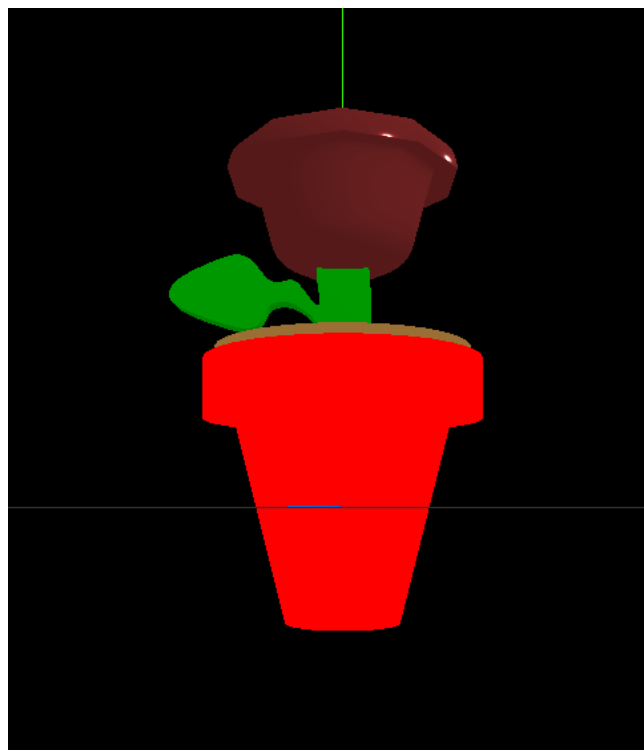


Imagen1.

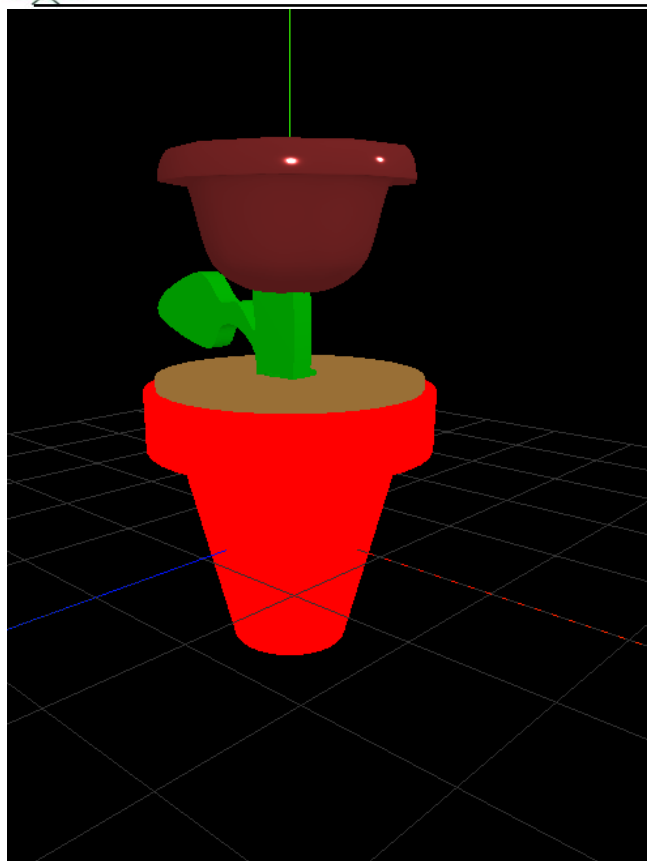


Imagen2.

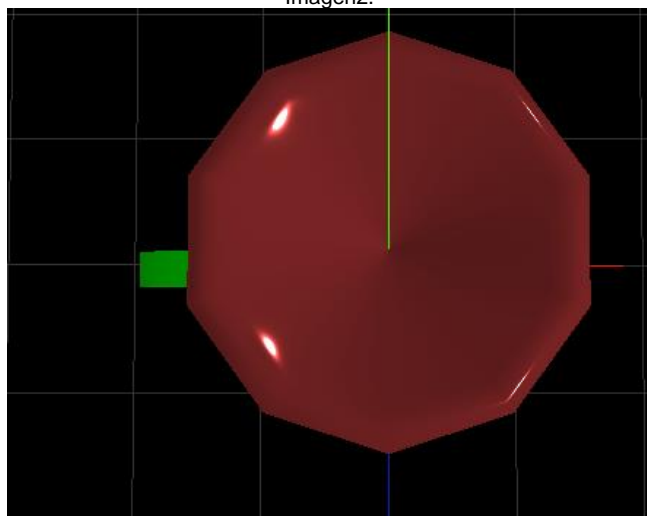


Imagen3.

II. COMPETENCIAS A DESARROLLAR

- Aprender a utilizar y aplicar las operaciones de Booleanos en THREE.js
- Aprender a utilizar y aplicar la extrusion en THREE.js
- Aprender a utilizar y aplicar la revolución de curvas en THREE.js

III. DESARROLLO DE LA PRÁCTICA

Para el desarrollo de la práctica, se utilizaron nuevas librerías la cual fueron:

```
<script src="js/csg.js"></script>
<script src="js/THREE.CSG.js"></script>
```

Además, se crearon objetos como la maceta parte a parte al igual que la tierra

```
var macetica1 = new THREE.Mesh( geometryK, mace );
var macetica2 = new THREE.Mesh( geometry1, mace );
var tierrita = new THREE.Mesh( geometry2, tierra );
```

Luego aplicar las operaciones de booleanos sin olvidar el cambio a CSG

```
var m1CSG = THREE.CSG.fromMesh( macetica1 );
var m2CSG = THREE.CSG.fromMesh( macetica2 );
var tCSG = THREE.CSG.fromMesh( tierrita );

//OPERACIONES
var result1 = m1CSG.union( m2CSG );
var result2 = tCSG.subtract( m2CSG );
var result3 = maticaCSG.union( tCSG );
var result4 = FlorCSG.union( maticaCSG );
```

y a su vez pasarlas a objetos de js y mostrarlas en la escena

```
k.material = mace;
J.material = tierra;
L.material = materialExt;
H.material = materialRev;
//

//AGREGAR A LA ESCENA LOS DIFERENTES ELEMENTOS

scene.add(k);
scene.add(J);
scene.add(L);
```

Luego realizamos la curva para el tallo y hoja de la planta y aplicamos Extrude para darle profundidad a la curva



```
var curve2D = [];  
curve2D[0] = new THREE.Vector2( -1,0 );  
curve2D[1] = new THREE.Vector2( -1, 2 );  
curve2D[2] = new THREE.Vector2( -2,3 );  
curve2D[3] = new THREE.Vector2( -3,3 );  
curve2D[4] = new THREE.Vector2( -4,2 );  
curve2D[5] = new THREE.Vector2( -5,2 );  
curve2D[6] = new THREE.Vector2( -7,3 );  
curve2D[7] = new THREE.Vector2( -7,4 );  
curve2D[8] = new THREE.Vector2( -5,5 );  
curve2D[9] = new THREE.Vector2( -3,4 );  
curve2D[10] = new THREE.Vector2( -2,4 );  
curve2D[11] = new THREE.Vector2( -1,3 );  
curve2D[12] = new THREE.Vector2( -1,5 );  
curve2D[13] = new THREE.Vector2( 1,5 );  
curve2D[14] = new THREE.Vector2( 1,0 );  
curve2D[15] = new THREE.Vector2( 0,0 );  
  
var shape = new THREE.Shape();  
shape.moveTo(0,0);  
shape.splineThru(curve2D);  
  
var material = new THREE.LineBasicMaterial( { color : 0xffffff }  
);  
var resolution = 10;  
var points = shape.getPoints( resolution );  
var geometry = new THREE.BufferGeometry().setFromPoints( points );  
// Create the final object to add to the scene  
var curveObject1 = new THREE.Line( geometry, material );  
  
//EXTRUDE  
var extrudeSettings = {  
  steps: 2,  
  amount: 1,  
  bevelEnabled: false,  
};
```

```
scene.add(k);  
scene.add(J);  
scene.add(L);  
L.applyMatrix( new THREE.Matrix4().makeScale(.2,.2,.2) );  
L.translateY(1.1);  
scene.add(Flor);  
Flor.applyMatrix( new THREE.Matrix4().makeScale(.2,.2,.2) );  
Flor.translateY(2.2);  
  
//scene.add( curveObject1 );  
//scene.add( curveObject2 );
```

Se comento las curvas , ya que le quitan estética a la escena

<https://docs.mcneel.com/rhino/6/help/es-es/commands/booleanunion.htm#Boolean2Objects>

Como ultimo paso creamos la curva para la flor y le aplicamos el código de Revolución para agregarla a la escena

```
// re  
var curve3D = [];  
curve3D[0] = new THREE.Vector2( -1,-1 );  
curve3D[1] = new THREE.Vector2( -3,0 );  
curve3D[2] = new THREE.Vector2( -4,3 );  
curve3D[3] = new THREE.Vector2( -5,3 );  
curve3D[4] = new THREE.Vector2( -4,4 );  
curve3D[5] = new THREE.Vector2( 0, 0 );  
  
var shapeF = new THREE.Shape();  
shapeF.moveTo(0,0);  
shapeF.splineThru(curve3D);  
  
var resolution = 50;  
var pointsF = shapeF.getPoints( resolution );  
var geoFlor = new THREE.BufferGeometry().setFromPoints( pointsF );  
// Create the final object to add to the scene  
var curveObject2 = new THREE.Line( geoFlor, material );  
  
var geometryRev = new THREE.LatheGeometry( pointsF, 10 );  
var materialRev = new THREE.MeshStandardMaterial( {  
  color: 0xAA3333,  
  metalness: 0.5,  
  roughness: 0.1,  
  opacity: 0.75,  
  transparent: false,  
} );  
  
var Flor = new THREE.Mesh( geometryRev, materialRev );
```

Como últimos recursos utilizamos la translación y la matriz de escala para acomodar los objetos y que las operaciones booleanas concuerden mejor y así mostrar los objetos a la escena