

INFORME PARCIAL

Andrés Rodríguez Prada 1

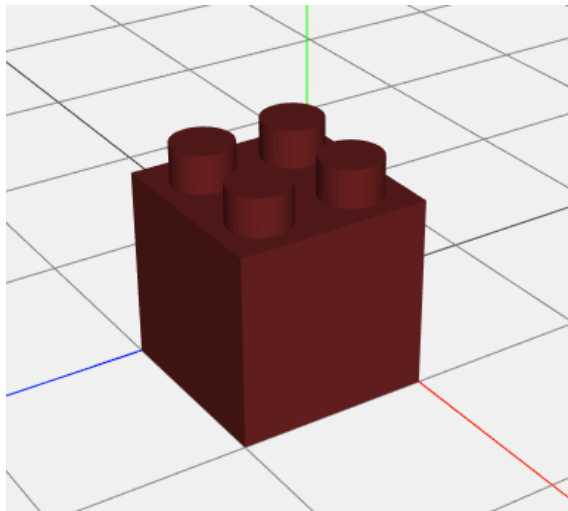
22 de abril de 2020

Computación Gráfica

Introducción

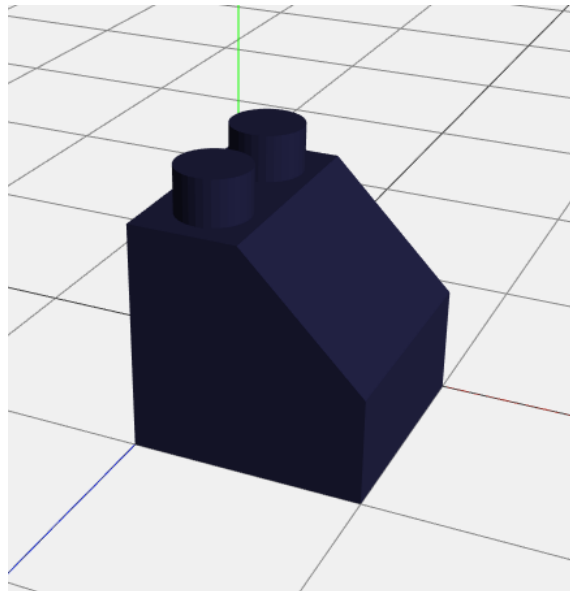
Para este ejercicio se utilizó el ejemplo de Voxel Painter de Threejs, para generar una aplicación que genere diferentes de fichas tipo Lego y se puedan ubicar en el espacio, el usuario puede seleccionar las fichas que desea presionando una tecla del 1 al 5, para agregar la ficha se pulsa click, el programa muestra una sombra que le ayuda al usuario a saber que ficha está escogiendo, puede eliminar una ficha si así lo desea utilizando shift + click. La cámara se puede acercar utilizando la rueda del ratón (scroll) y puede mover la cámara utilizando las flechas.

Ficha 1:

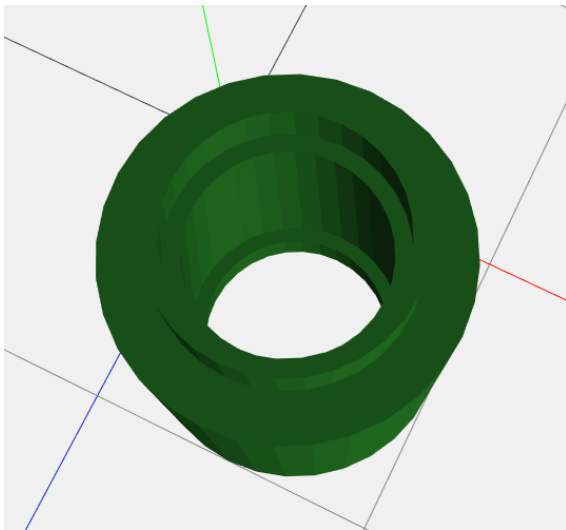
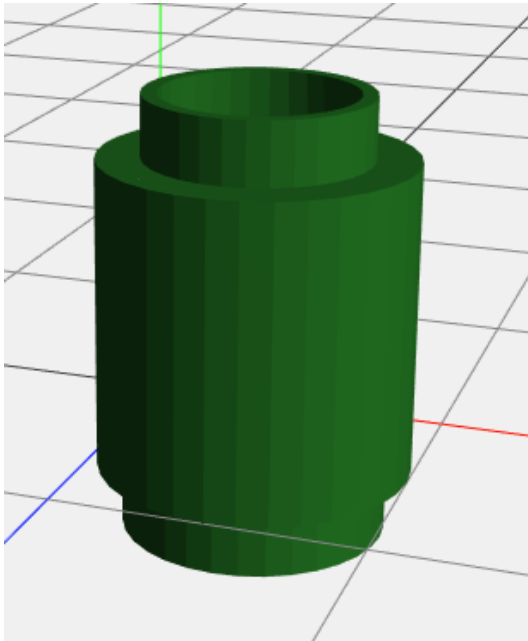


La ficha uno es la unión de una caja con 4 cilindros ubicados en su cara superior, esto se puede realizar haciendo uso de la librería CSG.

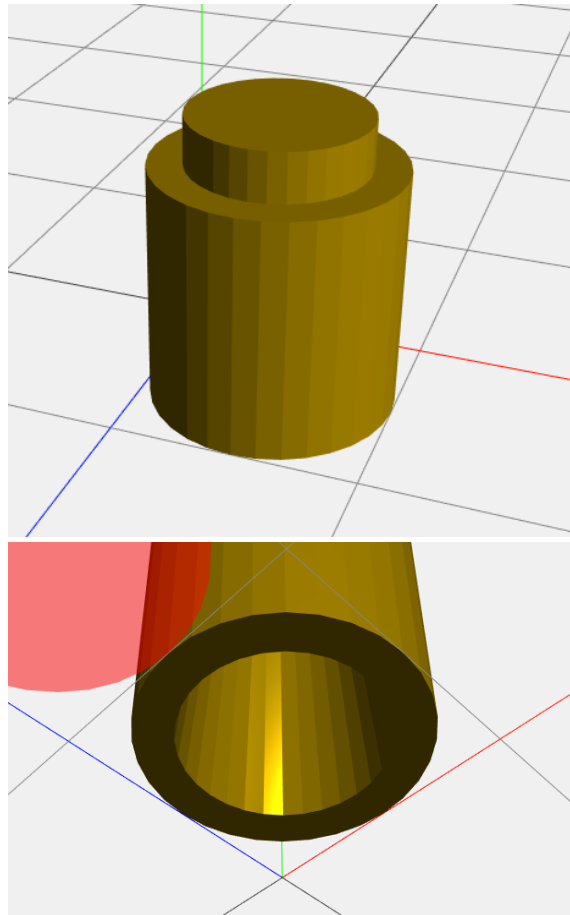
Ficha 2:



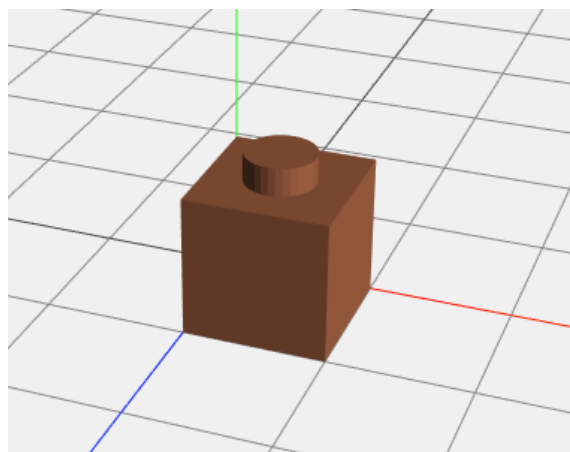
La ficha dos es la diferencia entre 2 cajas que poseen las mismas dimensiones solo que una de ellas esta rotada en el eje Z 45° y trasladada en el plano XY, al obtener la figura de la diferencia de estas dos cajas utilizo la unión a dos cilindros, para esto también se hizo uso de la librería CSG.

Ficha 3:

La Ficha 3 está diseñada a partir de 4 cilindros, esta ficha fue la de mayor complejidad en el uso de operaciones de CSG para obtener el resultado final, esta ficha es hueca. Para su desarrollo se tuvo que hacer 4 operaciones de CSG, las 3 primeras son la diferencia del cilindro 4 con los cilindros 1,2,3 (haciendo esto de a parejas es decir (4 con 1), (4 con 2)). De este resultado se obtuvo cilindros huecos y se procede a ubicarlos en el espacio para conformar la figura finalmente se aplica una unión a las 3 figuras obtenidas de las operaciones anteriores.

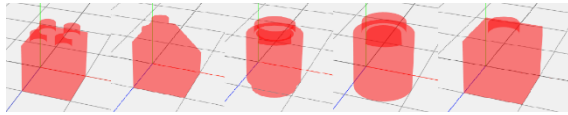
Ficha 4:

La ficha cuatro esta generada por 3 cilindros, a los 2 primeros se les hace una diferencia para generar un cilindro hueco y finalmente se hace una unión con el cilindro 3.

Figura 5:

La ficha 5 solo es la unión entre un cilindro y una caja utilizando las librerías de CSG.

Figuras Auxiliares:



Estas son las más figuras que se han creado previamente solo que se utiliza un material de color rojo y se le da la propiedad de que tenga cierto nivel de transparencia.

Interacción Botones:

Para el uso de las teclas del 1 al 5 se utilizó la función `onDocumentKeyDown`, además de 5 variables booleanas para saber cuándo el botón se había oprimido, los botones del 1 al 5 sirven tanto para los que están arriba de las letras, como para lo que se encuentran en el teclado numérico, esto se hizo con el fin de prever el caso de que la persona presione cualquiera de las 2 teclas “1” que se encuentra en los teclados de Pc de escritorio. Para obtener el número del Event Key utilicé la página <https://keycode.info/>.

Para la iniciación del programa la variable booleana `uno` inicia en `true`, esto para que el programa inicie con una figura seleccionada, en la función `onDocumentKeyDown` se cambian estos valores de acuerdo a la tecla pulsada, si se oprime 2 todas las variables cambian a `False` y la variable `dos` cambia a `True`, además se actualiza la figura auxiliar, removiendo de la escena la que se encontraba anteriormente y colocando la que corresponda a la tecla pulsada.

```
case 97:
  uno = true;
  scene.remove( rollOverMesh );
  Ficha_1.material = rollOverMaterial;
  rollOverMesh = Ficha_1;
  scene.add( rollOverMesh );
  dos = false;
  tres = false;
  cuatro = false;
  cinco = false;
```

Agregar Fichas a la escena:

Finalmente, quien se encarga de agregar los objetos a la escena es la función `onDocumentMouseDown`, se activa cuando se presiona un click izquierdo o derecho, entonces procede a verificar cual es la variable booleana que se encuentra en `True`, la que se encuentre en este estado hace referencia a la ficha que se desea dibujar, entonces crea una ficha con la geometría y el material asignado, luego a esta ficha le asigna una posición haciendo uso del Ray Caster finalmente agrega a la escena la ficha elegida y la agrega a un arreglo llamado `objects`.

```
if(unos){
  var voxel = new THREE.Mesh(
    Ficha_1.geometry, materialR);
  voxel.position.copy( intersect.point ).add(
    intersect.face.normal );
  voxel.position.divideScalar( 50
  ).floor().multiplyScalar( 50 ).addScalar( 25 );
  scene.add( voxel );
  objects.push( voxel );
}
```

```

if(uno){
    var voxel = new THREE.Mesh( Ficha_1.geometry, materialR);
    voxel.position.copy( intersect.point ).add( intersect.face.normal );
    voxel.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel );
    objects.push( voxel );
}
if(dos){
    var voxel = new THREE.Mesh( Ficha_2.geometry, materialB);
    voxel.position.copy( intersect.point ).add( intersect.face.normal );
    voxel.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel );
    objects.push( voxel );
}
if(tres){
    var voxel = new THREE.Mesh( Ficha_3.geometry, materialG);
    voxel.position.copy( intersect.point ).add( intersect.face.normal );
    voxel.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel );
    objects.push( voxel );
}
if(cuatro){
    var voxel = new THREE.Mesh( Ficha_4.geometry, materialY);
    voxel.position.copy( intersect.point ).add( intersect.face.normal );
    voxel.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel );
    objects.push( voxel );
}
if(cinco){
    var voxel = new THREE.Mesh( Ficha_5.geometry, materialS);
    voxel.position.copy( intersect.point ).add( intersect.face.normal );
    voxel.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel );
    objects.push( voxel );
}
}

```