



**UNIVERSIDAD MILITAR NUEVA GRANADA**  
**FACULTAD DE INGENIERÍA**  
**DEPARTAMENTO DE TECNOLOGÍAS DEL CONOCIMIENTO**  
**DOCENTE: GABRIEL EDUARDO AVILA BUITRAGO**

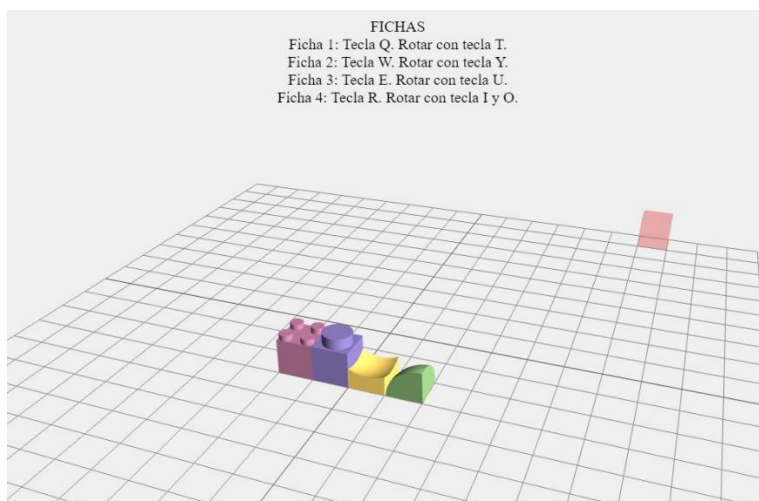
Daniela Rivera Chavarro<sup>1</sup>, 6000365

## FICHAS DE CONSTRUCCIÓN

---

### 1.RESUMEN.

A partir de los temas vistos en clase, se logró realizar el proyecto. Allí se muestra un escenario vacío, en donde el usuario puede interactuar, mediante el teclado, el escenario. Además, se pueden ingresar 4 diferentes fichas de construcción, las cuales se pueden rotar mediante el teclado, por lo que, el usuario tiene la libertad de manipular las fichas a su gusto.



En la imagen se muestra la escena y las cuatro fichas ya colocadas.

Estas se colocan mediante el mouse, pero se cambian mediante el teclado: Con las teclas Q, W, E, T, y se rotan con Y, U, I, O, y P respectivamente.

Las figuras mostradas se hicieron en un archivo a parte y una vez terminadas se agregaron al código original.

Para la construcción de las fichas se trabajó con CSG. Todas las fichas tienen una implantación CSG, ya sea de unión, intersección o substracción.

---

<sup>1</sup> Ingeniería Multimedia

A continuación, se mostrará el código de estas:

```
264
265 //CONVERTIR A CSG
266
267 var boxCSG = THREE.CSG.fromMesh( cube );
268 var cylinderCSG = THREE.CSG.fromMesh( cylinder );
269 var cylinderCSG1 = THREE.CSG.fromMesh( cylinder_clone );
270 var cylinderCSG2 = THREE.CSG.fromMesh( cylinder_clone2 );
271 var cylinderCSG3 = THREE.CSG.fromMesh( cylinder_clone3 );
272
273
274
275 var boxCSG1 = THREE.CSG.fromMesh( cube_clone );
276 var cylinderCSG4 = THREE.CSG.fromMesh( cylinder2 );
277
278 var sphereCSG = THREE.CSG.fromMesh( sphere );
279 var sphereCSG2 = THREE.CSG.fromMesh( sphere2 );
280
281 //APLICAR LAS OPERACIONES
282
283 var result1 = boxCSG.union( cylinderCSG );
284 .union( cylinderCSG1 );
285 .union( cylinderCSG2 );
286 .union( cylinderCSG3 );
287
288 var result2 = boxCSG1.union( cylinderCSG4 );
289
290 var result3 = boxCSG.subtract( sphereCSG );
291
292 var result4 = sphereCSG2.intersect( boxCSG );
293
294
295
296 //CONVERTIR A MALLA DE THREE JS NUEVAMENTE
297
298 pieza = THREE.CSG.toMesh( result1 );
299 pieza2 = THREE.CSG.toMesh( result2 );
300 pieza3 = THREE.CSG.toMesh( result3 );
301 pieza4 = THREE.CSG.toMesh( result4 );
302
```

Figuras hechas en CSG

Aquí se hacen las operaciones del CSG, luego se pasan a “resultados”.

Luego, al convertir en THREE JS nuevamente, se manda el resultado final (“pieza”)

Se crearon varios voxeles para recibir la pieza, más específicamente un voxel por pieza, como se muestra a continuación:

```

} else {
  if (Q){
    voxel = new THREE.Mesh( pieza.geometry, cubeMaterial );
    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 (W){
    voxel2 = new THREE.Mesh( pieza2.geometry, material1 );
    voxel2.position.copy( intersect.point ).add( intersect.face.normal );
    voxel2.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel2 );

    objects.push( voxel2 );
  }

  if (E){
    voxel3 = new THREE.Mesh( pieza3.geometry, material2 );
    voxel3.position.copy( intersect.point ).add( intersect.face.normal );
    voxel3.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel3 );

    objects.push( voxel3 );
  }

  if (R){
    voxel4 = new THREE.Mesh( pieza4.geometry, material3 );
    voxel4.position.copy( intersect.point ).add( intersect.face.normal );
    voxel4.position.divideScalar( 50 ).floor().multiplyScalar( 50 ).addScalar( 25 );
    scene.add( voxel4 );

    objects.push( voxel4 );
  }
}

```

Al voxel se le manda la geometría de la pieza y el material que se creó.

Las condiciones que se ven acá son las que permiten que se cambie la figura, a través de la tecla señalada.

Todo lo mostrado permite que las figuras se agreguen a la escena y, además, que el usuario interactúe para este mismo fin.

Por otro lado, en la función render se agregaron las respectivas rotaciones para las cuatro figuras.

```
432 function animate(){
433   render();
434   controls.update();
435   requestAnimationFrame( animate );
436
437 }
438
439
440
441
442 function render() {
443   renderer.render( scene, camera );
444
445   if (T){
446     velX = 0.1;
447     voxel1.rotateX(velX);
448   }
449
450   if (Y){
451     velX = 0.1;
452     voxel2.rotateX(velX);
453   }
454
455   if (U){
456     velX = 0.1;
457     voxel3.rotateX(velX);
458   }
459
460   if (I){
461     velX = 0.1;
462     voxel4.rotateY(velX);
463   }
464
465   if (O){
466     velX = 0.1;
467     voxel4.rotateX(velX);
468   }
469
470 }
```

La condición señala que, si se oprime la tecla, la figura guardada en el voxel debe rotar.

Las teclas para la rotación son: T, Y, U, I y O, respectivamente. Para la última ficha, se agregó dos formas de rotación, puesto que ésta es mejor manipularla, tanto en x, como en y. Por ello, como se puede ver, las otras fichas solo tienen rotación en x.

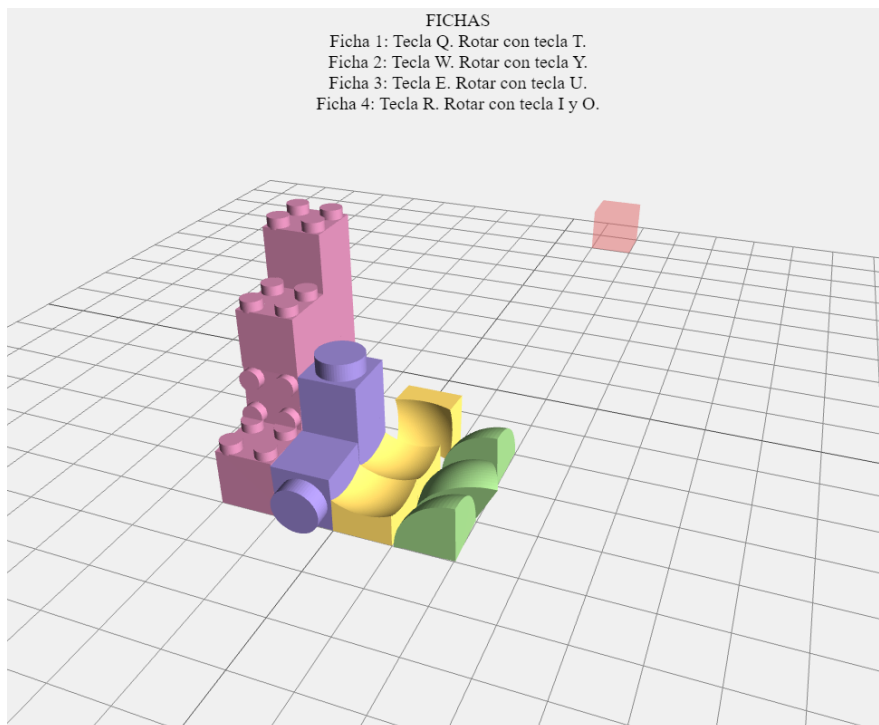
```
92
93
94
95 //Cambiar de fichas
96 case 81: Q = true;
97 break;
98
99
100 case 87: W = true;
101 break;
102
103 case 69: E = true;
104 break;
105
106 case 82: R = true;
107 break;
108
109 //Rotar fichas
110
111 case 84: T = true;
112 break;
113
114 case 89: Y = true;
115 break;
116
117 case 85: U = true;
118 break;
119
120 case 73: I = true;
121 break;
122
123 case 79: O = true;
124 break;
125
126 }
127
128
129
130 var onKeyUp = function ( event ) {
131   switch ( event.keyCode ) {
132
133     //Cambiar de fichas
134     case 81: Q = false;
135     break;
136
137     case 87: W = false;
```

En la parte superior del código se puede encontrar lo elementos de teclado, están separados y organizados de tal manera en la que se puede entender mejor el código: Cambios de fichas, por un lado, y rotaciones por el otro.

El escenario se puede manipular mediante el mouse, para hacer zoom o moverlo directamente.

Además, las piezas se pueden colocar unas encima de otras y así construir más libremente.

Además, en la parte superior de la pantalla hay un “menú” explicativo, para que el usuario entienda mejor como puede manipular el programa.



### **3. CONCLUSIONES.**

A partir de lo aprendido en clase y mediante la práctica, se logró crear el proyecto deseado y se concluyó con que se utilizó adecuadamente los recursos y los temas planteados por el profesor.