

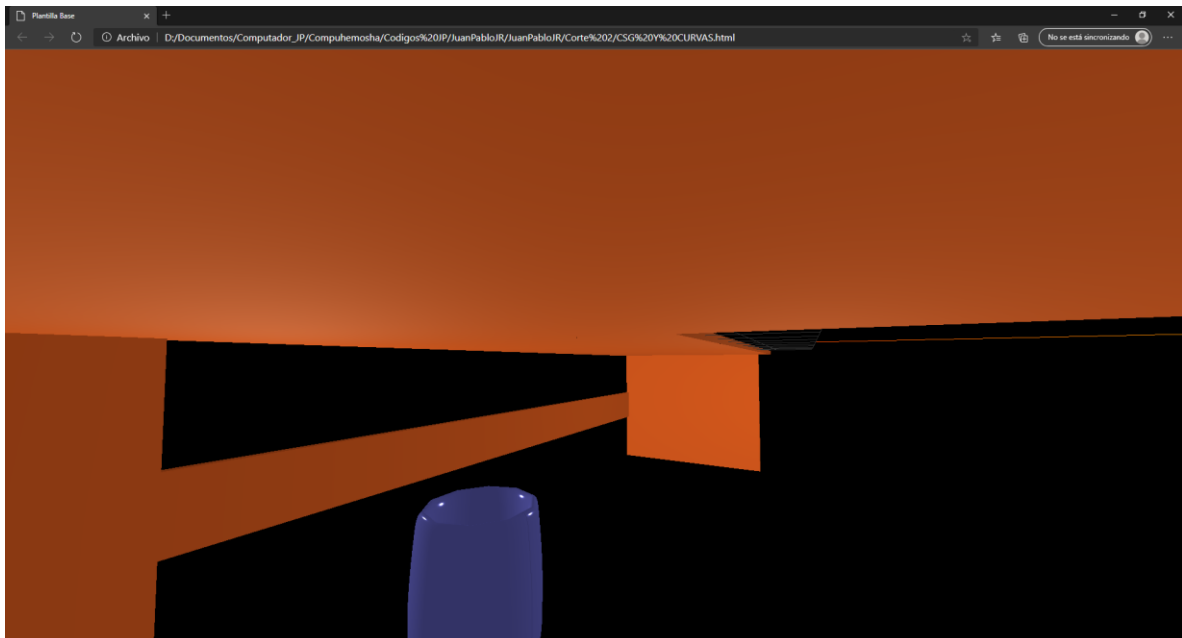
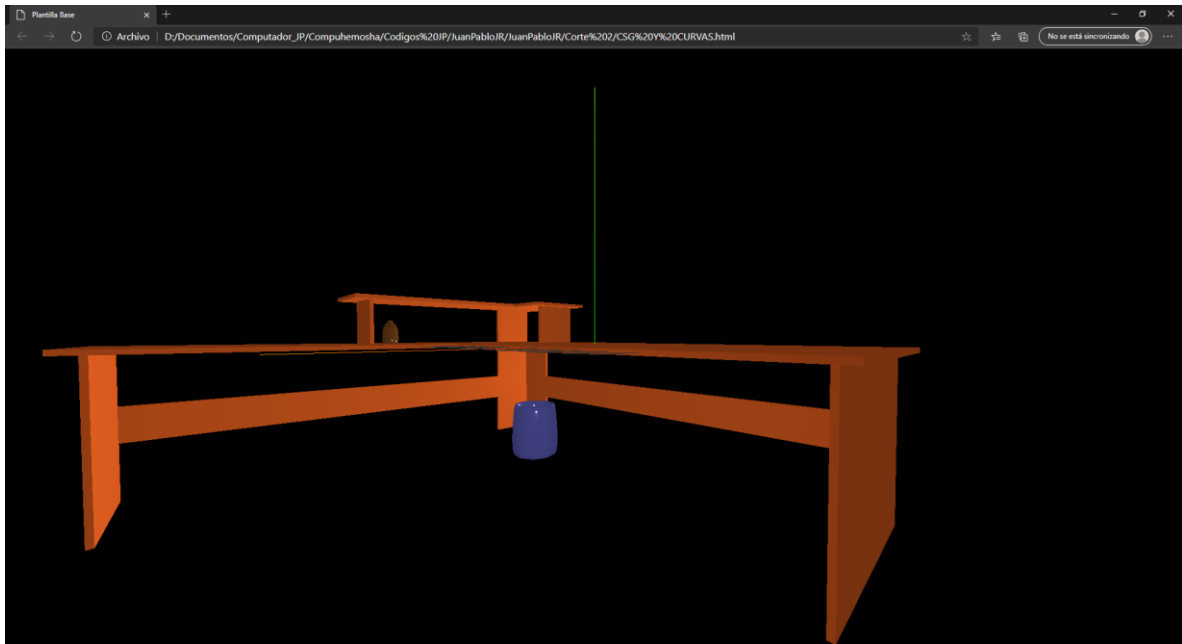
Nombre: Juan Pablo Jiménez Rodríguez

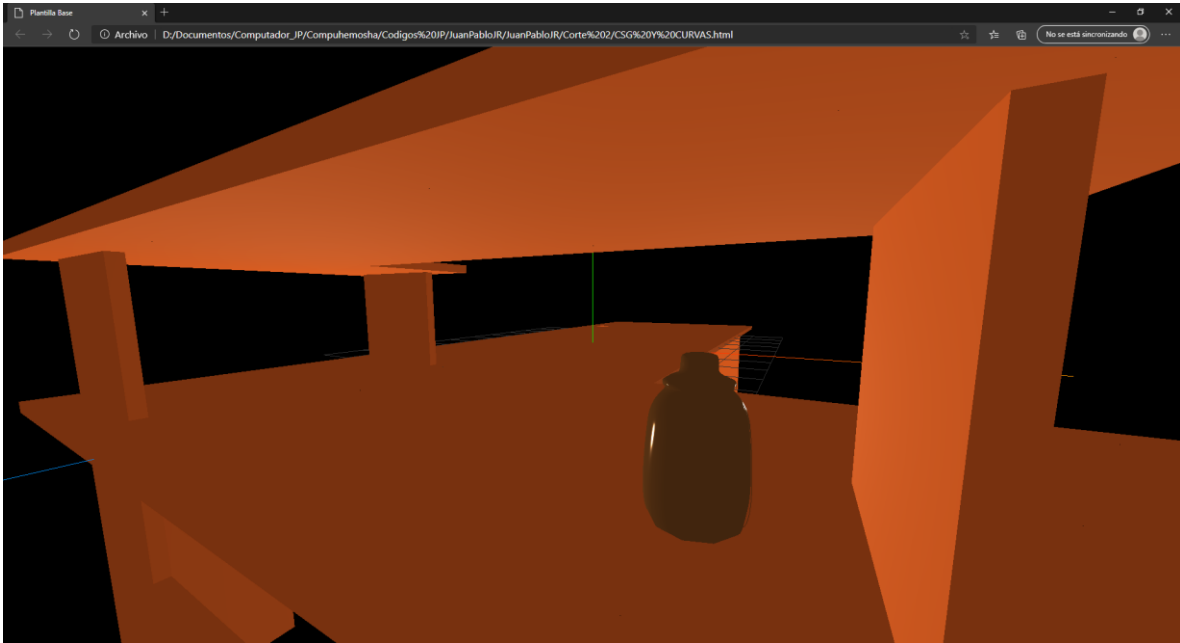
Código: 6000328

Trabajo en clase

Modelado de mi escritorio

Primer Punto:





Se realizó el escritorio con la ayuda de la librería CSG, donde se utilizó de 15 figuras diferentes.

13 figuras son cajas y una es un cilindro.

El escritorio cuenta con un segundo piso y tiene un agujero que permite pasar los cables que se necesiten.

```

63
64 //CREAR LAS GEOMETRÍAS
65 var boxGeometry = new THREE.BoxGeometry( 7,0.15 , 19 );
66 var boxGeometry2 = new THREE.BoxGeometry( 10,0.15 , 5 );
67 var boxGeometry3 = new THREE.BoxGeometry( 0.15,5 , 4 );
68 var boxGeometry4 = new THREE.BoxGeometry( 6.1,5 , 0.15 );
69 var boxGeometry5 = new THREE.BoxGeometry( 2,5 , 0.15 );
70 var boxGeometry6 = new THREE.BoxGeometry( 0.15,5 , 2 );
71 var boxGeometry7 = new THREE.BoxGeometry( 0.15,1.2 , 18 );
72 var boxGeometry8 = new THREE.BoxGeometry( 14,1.2 , 0.15 );
73 var boxGeometry9 = new THREE.BoxGeometry( 2,2 , 0.15 );
74 var boxGeometry10 = new THREE.BoxGeometry( 2,2 , 0.15 );
75 var boxGeometry11 = new THREE.BoxGeometry( 3,0.15 , 5 );
76 var boxGeometry12 = new THREE.BoxGeometry( 0.15,2 , 2 );
77 var boxGeometry13 = new THREE.BoxGeometry( 8,0.15 , 3 );
78
79
80 var cilindroGeometry = new THREE.CylinderGeometry( 0.3,0.3,1,20 );
81

```

Imagen 3. Creación de geometrías

En esta parte del código creamos las geometrías necesarias para crear el sólido que queremos. Para esto, se utilizó únicamente geometrías primitivas.

```

82 //CREAR LOS MATERIALES
83 var material1 = new THREE.MeshStandardMaterial( { color: colorCafe, metalness: 0.5, roughness: 0.1 } );
84
85 var material2 = new THREE.MeshStandardMaterial( { color: colorG, metalness: 0.5, roughness: 0.1 } );

```

Imagen 4. Creación de materiales

Aquí cree el material que le aplicamos a todo el escritorio, un material estándar, donde se le aplicó un color naranjoso, además es metálico para darle como brillo y un poco de aspereza para tomar un poco el color al mate.

```

86
87 //CREAR LAS MALLAS
88 var cube = new THREE.Mesh( boxGeometry );
89 var cube2 = new THREE.Mesh (boxGeometry2);
90 var cube3 = new THREE.Mesh (boxGeometry3);
91 var cube4 = new THREE.Mesh (boxGeometry4);
92 var cube5 = new THREE.Mesh (boxGeometry5);
93 var cube6 = new THREE.Mesh (boxGeometry6);
94 var cube7 = new THREE.Mesh (boxGeometry7);
95 var cube8 = new THREE.Mesh (boxGeometry8);
96 var cube9 = new THREE.Mesh (boxGeometry9);
97 var cube10 = new THREE.Mesh (boxGeometry10);
98 var cube11 = new THREE.Mesh (boxGeometry11);
99 var cube12 = new THREE.Mesh (boxGeometry12);
100 var cube13 = new THREE.Mesh (boxGeometry13);
101 var cylindric = new THREE.Mesh (cilindroGeometry);
102

```

Imagen 5: Creamos las mallas, en otras palabras, metemos las figuras en un mesh y luego les damos sus posiciones

```

162 //CONVERTIR A CSG
163 var boxCSG = THREE.CSG.fromMesh( cube );
164 var boxCSG2 = THREE.CSG.fromMesh(cube2);
165 var boxCSG3 = THREE.CSG.fromMesh(cube3);
166 var boxCSG4 = THREE.CSG.fromMesh(cube4);
167 var boxCSG5 = THREE.CSG.fromMesh(cube5);
168 var boxCSG6 = THREE.CSG.fromMesh(cube6);
169 var boxCSG7 = THREE.CSG.fromMesh(cube7);
170 var boxCSG8 = THREE.CSG.fromMesh(cube8);
171 var boxCSG9 = THREE.CSG.fromMesh(cube9);
172 var boxCSG10 = THREE.CSG.fromMesh(cube10);
173 var boxCSG11 = THREE.CSG.fromMesh(cube11);
174 var boxCSG12 = THREE.CSG.fromMesh(cube12);
175 var boxCSG13 = THREE.CSG.fromMesh(cube13);
176 var CSGcylindric = THREE.CSG.fromMesh( cylindric );
177
178 //APLICAR LAS OPERACIONES
179 //var result = boxCSG.subtract( sphereCSG ); //ACTIVAR PARA VER: CAJA MENOS ESFERA
180 //var result = boxCSG.intersect( sphereCSG ); //ACTIVAR PARA VER: ESFERA MENOS CAJA
181 var result = boxCSG.union( boxCSG2 );
182     result= result.union( boxCSG3 );
183     result= result.union( boxCSG4 );
184     result= result.union( boxCSG5 );
185     result= result.union( boxCSG6 );
186     result= result.union( boxCSG7 );
187     result= result.union( boxCSG8 );
188     result= result.union( boxCSG9 );
189     result= result.union( boxCSG10 );
190     result= result.union( boxCSG11 );
191     result= result.union( boxCSG12 );
192     result= result.union( boxCSG13 );
193     result= result.subtract( CSGcylindric );
194
195 //ACTIVAR PARA VER: ESFERA MENOS CAJA
196 //var result = sphereCSG.subtract( boxCSG ); //ACTIVAR PARA VER: ESFERA MENOS CAJA
197
198 //CONVERTIR A THREE
199 cube = THREE.CSG.toMesh( result );
200 //cube = THREE.CSG.toMesh( result2 );
201 cube.material = material1;
202 //sphere.material = material2;
203

```

Imagen 6. Esos mesh anteriores los dejamos en CSG.fromMesh y luego de esos podemos crear las soluciones

En este caso usamos 13 sumas y una resta.

Segundo Punto:

```

79
80 //CURVAS-OPJETOS
81
82 var basura2D = [];
83
84     basura2D[0] = new THREE.Vector2 (0, 0);
85     basura2D[1] = new THREE.Vector2 (1, 0 );
86     basura2D[2] = new THREE.Vector2 ( 1, 2);
87     basura2D[3] = new THREE.Vector2 (0.8, 2);
88     basura2D[4] = new THREE.Vector2 (0.8, 0.2);
89     basura2D[5] = new THREE.Vector2 (0, 0.2);
90
91 var shape = new THREE.Shape();
92 shape.moveTo(0,0.2);
93 shape.splineThru (basura2D);
94
95 //CURVA BOTELLA
96
97 var botella2D =[];
98
99     botella2D[0] = new THREE.Vector2 (0, 0);
100     botella2D[1] = new THREE.Vector2 (0.3, 0 );
101     botella2D[2] = new THREE.Vector2 ( 0.3, 0.6);
102     botella2D[3] = new THREE.Vector2 (0.2, 0.8);
103     botella2D[4] = new THREE.Vector2 (0.21, 0.81);
104     botella2D[5] = new THREE.Vector2 (0.21, 0.85);
105     botella2D[6] = new THREE.Vector2 (0.12, 0.9);
106     botella2D[7] = new THREE.Vector2 (0.1, 1);
107     botella2D[8] = new THREE.Vector2 (0, 1);
108
109 var shape2 = new THREE.Shape();
110 shape2.moveTo(0,0);
111 shape2.splineThru (botella2D);
112
113

```

Imagen 7: Se crearon dos geometrías con curvas, una basurita que cuenta con 6 puntos y una botella de un líquido para lentes de contacto, que cuenta con 9 puntos.

```

118
119 //CREAR LOS MATERIALES
120 var material1 = new THREE.MeshStandardMaterial( { color: colorCafe, metalness: 0.5, roughness: 1 } );
121 var material = new THREE.LineBasicMaterial ( { color : colorB});
122 var material= new THREE.MeshStandardMaterial( {
123     color: colorB,
124     metalness: 0.5,
125     roughness: 0.1,
126
127     ...
128 } );
129
130 var materialb = new THREE.LineBasicMaterial ( { color : 0x41250E});
131 var materialb= new THREE.MeshStandardMaterial( {
132     color: 0x41250E,
133     metalness: 0.5,
134     roughness: 0.1,
135
136     ...
137 } );
138
139 var resolution = 100;
140 var points = shape.getPoints (resolution);
141 var points2 = shape2.getPoints (resolution);
142 var geometry = new THREE.BufferGeometry().setFromPoints(points);
143 var geometryb = new THREE.BufferGeometry().setFromPoints(points2);
144 var curveBasura = new THREE.Line(geometry, material);
145 var curveBotella = new THREE.Line(geometryb, materialb);
146 var geometry = new THREE.LatheGeometry( points, 10 );
147 var geometryb = new THREE.LatheGeometry( points2, 10 );

```

Imagen 8: A cada curva se le aplicó un material con diferente color

```

169
170 //var sphere = new THREE.Mesh( sphereGeometry );
171
172 curveBasura.translateX(0);
173 curveBasura.translateY(-5);
174 curveBasura.translateZ(4);
175
176 latheBasura.translateX(0);
177 latheBasura.translateY(-5);
178 latheBasura.translateZ(4);
179
180 curveBotella.translateX(5);
181 curveBotella.translateY(0.1);
182 curveBotella.translateZ(8);
183
184 latheBotella.translateX(5);
185 latheBotella.translateY(0.1);
186 latheBotella.translateZ(8);

```

Imagen 9: Se le asignó una posición tanto a las curvas como al objeto

```

286 //AGREGAR A LA ESCENA LOS DIFERENTES ELEMENTOS
287 scene.add( cube );
288 scene.add( latheBasura );
289 scene.add( curveBasura );
290 scene.add( latheBotella );
291 scene.add( curveBotella );

```

Imagen 10: Para terminar solo lo aplicamos a la escena