

Animal: Mono.

Nuestro animal, busca ser una versión “caricaturesca” de un mono, con esto se puede apreciar que en un espacio animado, se integra muy bien al ambiente.

Se quiso resaltar su rasgos más característicos como lo son, sus grandes orejas, su cola, sus manos y pies, también se resaltaron los ojos ya que como se mencionó anteriormente se busca una representación más animada.



Brayan Sebastian Becerra Beltrán	u1202052
----------------------------------	----------

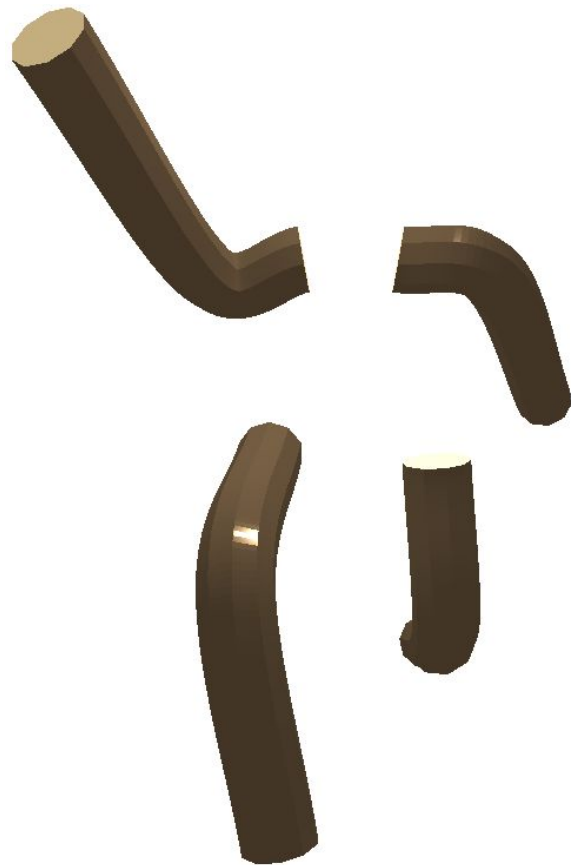
Cabeza

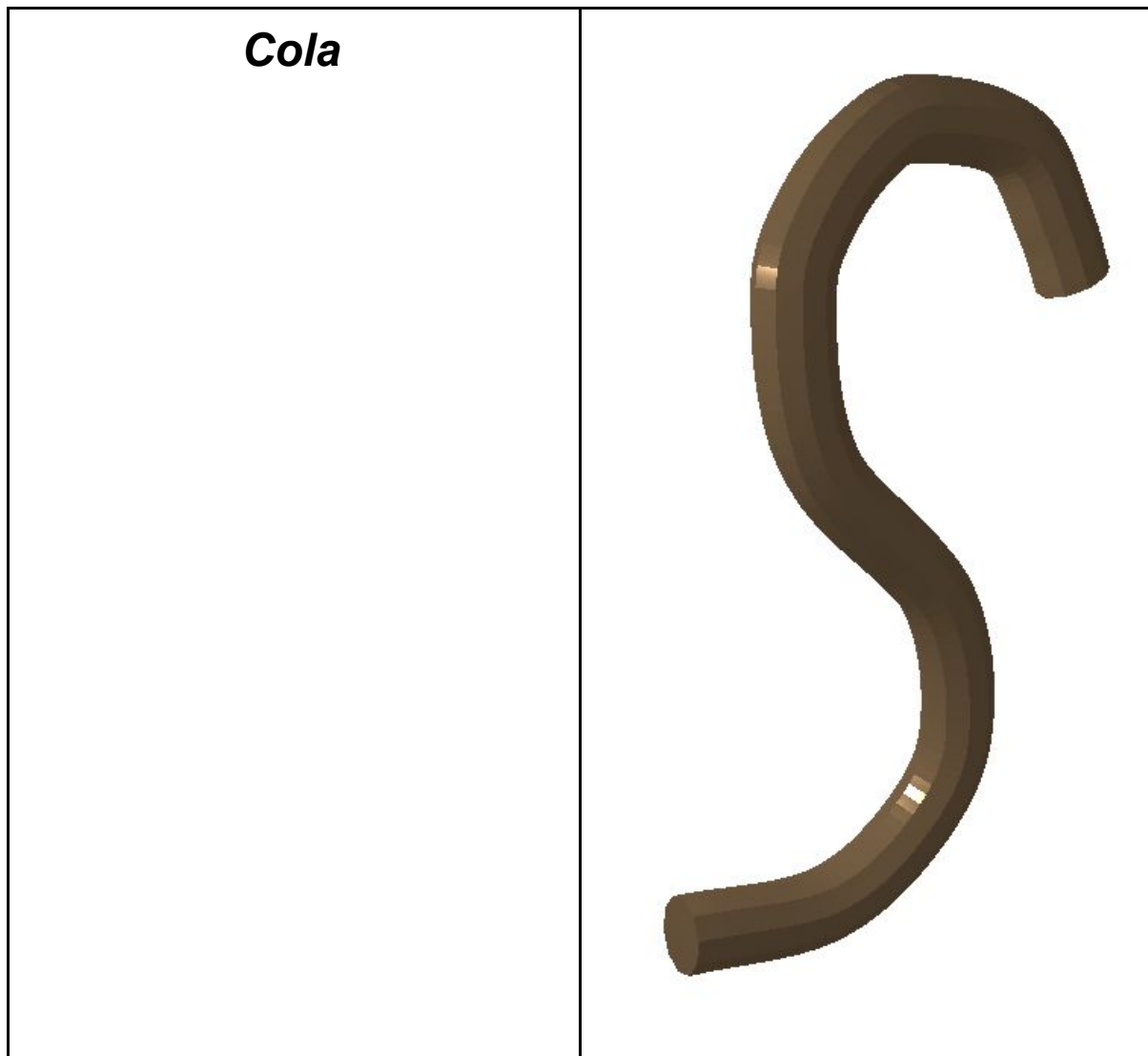
- Estructura
- Orejas
- Hocico
- Ojos



Extremidades

- Brazos
- Piernas





- Cabeza

- Estructura:

- *Strip y Fan*: Se realizó la estructura de la cabeza con ayuda de los métodos `Strip` y `fan` por medio de clases `class TRIANGLE_STRIP{}` y `class TRIANGLE_FAN{}`, modificando sus tamaños y posiciones en el espacio.

- Orejas:

- Se inició declarando las geometrías a utilizar que en este caso esferas, a partir de esto se importó la *librería* `CSG` para poder hacer operaciones booleanas como son (unión, intersección, Diferencia).

Se creó un función llamada `function orejas(ox, oz, oy){}` ya que esta función la vamos a utilizar 2 veces y ahorramos

código, donde los parámetros “ox”, “oz” y “oy”, estos nos indican las posición en el espacio de objetivo que vamos a crear.

- Hocico:

- *Strip y Fan*: Se realizó la estructura de la cabeza con ayuda de los métodos Strip y fan por medio de clases “`class TRIANGLE_STRIP{}`” y “`class TRIANGLE_FAN{}`”, modificando sus tamaños y posiciones en el espacio.

- Ojos:

- Primitivas: Se utilizaron las figuras geométricas de three js; en este caso se utilizó la esfera como recurso.

Se creó un función llamada “`function ojos(ox, oz, px, pz){}`” ya que esta función la vamos a utilizar 2 veces y ahorramos código, donde los parámetros “ox”, “oz”, “px” y “pz”, estos nos indican las posición en el espacio de objetivo que vamos a crear.

- Extremidades

- Brazos y piernas:

- Solidos de revolucion y extrude: Con ayuda de la extrusión y los sólidos de revolución se crearon figuras cilíndricas en los planos “x”, “y”, y “z”, con esto se crearon objetos más orgánicos en el animal.

Los solidos de revolucion se lograron gracias a la matriz de puntos, donde se modificó el tamaño y la cantidad de vértices del cilindro.

```
var pts = [], numPts = 5;

for ( var i = 0; i < numPts * 2; i ++ ) {
    var l = i % 2 == 1 ? 1 : 1;
    var a = i / numPts * Math.PI;
    pts.push( new THREE.Vector2( Math.cos( a ) * l, Math.sin( a ) * l ) );
}
```

Para cada brazo y pierna se utilizó una diferente extrusión gracias a el cambio de orientación que se daba en el “`ClosedSpline = new THREE.CatmullRomCurve3`”.

```
var closedSpline = new THREE.CatmullRomCurve3( [  
    new THREE.Vector3( -1, -4, 1 ),  
    new THREE.Vector3( -3, -4.5, 3 ),  
    new THREE.Vector3( 0, 0, 10 ),  
] );  
  
closedSpline.curveType = 'catmullrom';  
closedSpline.closed = false;  
  
var extrudeSettings = {  
    steps: 100,  
    amount: 1,  
    bevelEnabled: false,  
    bevelThickness: 1,  
    extrudePath: closedSpline  
};
```


- Cola
 - Solidos de revolucion y extrude: Con ayuda de la extrusión se y los sólidos de revolución se crearon figuras cilíndricas a partir de coordenadas `new THREE.Vector3` en los planos “x”, “y”, y “z”, con esto se crearon objetos más orgánicos en el animal.

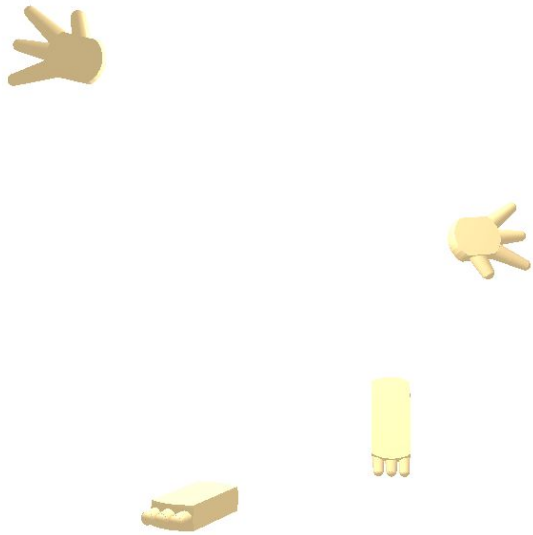

Los solidos de revolucion se lograron gracias a la matriz de puntos, donde se modificó el tamaño y la cantidad de vértices del cilindro.

```
for ( var i = 0; i < numPts * 4; i ++ ) {  
    var l = i % 2 == 1 ? .5 : .5;  
    var a = i / numPts * Math.PI;  
    pts_col.push( new THREE.Vector2( Math.cos( a ) * l, Math.sin( a ) * l ) );  
}
```

Para la cola se utilizaron 13 coordenadas para lograr la forma que en “s” que tiene.

```
var closedSpline5 = new THREE.CatmullRomCurve3( [  
  new THREE.Vector3( -2.5, -9.75, -2.5 ),  
  new THREE.Vector3( -4, -10, -4 ),  
  new THREE.Vector3( -5, -9.25, -5),  
  new THREE.Vector3( -5.5, -8, -5.5),  
  new THREE.Vector3( -5.25, -6, -5.25),  
  new THREE.Vector3( -4, -4, -4),  
  new THREE.Vector3( -3.75, -2, -3.75),  
  new THREE.Vector3( -4, -1, -4),  
  new THREE.Vector3( -4.5, -.25, -4.5),  
  new THREE.Vector3( -5, 0, -5),  
  new THREE.Vector3( -6, -.25, -6),  
  new THREE.Vector3( -6.5, -1.25, -6.5),  
  new THREE.Vector3( -6.75, -2, -6.75),  
] );
```

Omar Fernando Gonzalez Ramirez	u1202045
<p><i>Torso</i></p> <ul style="list-style-type: none">• Estructura• Panza• Ombligo	

<i>Manos y Pies</i> <ul style="list-style-type: none">• Manos• Pies	
<i>Nariz</i> <ul style="list-style-type: none">• Nariz	

- Torso:
 - Estructura:
 - Strip y Fan: Se realizó la estructura de la cabeza con ayuda de los métodos Strip y fan, modificando sus tamaños y posiciones en el espacio para ir dándole forma y estética a la estructura .
 - Barriga
 - Strip y Fan: Se realizó la estructura de la cabeza con ayuda de los métodos Strip y fan, modificando sus tamaños y posiciones en el espacio para ir dándole forma y estética a la estructura .
 - ombligo:
 - Primitivas: Se utilizaron las figuras geométricas de three js; en este caso se utilizó la esfera como recurso le dimos su tamaño y la ubicamos en (x, y, z).
- Extremidades
 - Manos:

- Se inició creando una función con 6 parámetros lo cual son las ubicaciones (X, Y, Z) y los ángulos en (X, Y, Z)

```
function manos(ox, oy, oz, ax, ay, az){
```

ya que esta función la vamos a utilizar 2 veces y ahorramos código, declaramos las geometrías a utilizar que en este caso fueron los cilindros, cubos y esferas y a partir de esto se importó la librería CSG para poder hacer operaciones booleanas como son (unión, intersección, Diferencia) y a partir de estar crear la forma un problema que tuve en esta función es que al crear esta pieza quedo muy grande con relación al animal y cambiar valores se iba a distorsionar todo el objeto, la solución que pude implementar era agregar una transformación de escalación para que se escalara y no altere ninguna parte del diseño .

```
t.set( sx, 0, 0, 0,  
      0, sy, 0, 0,  
      0, 0, sz, 0,  
      0, 0, 0, 1 );  
C2T2.applyMatrix(t);
```

- Pies:

- Se inició creando una función con 6 parámetros lo cual son las ubicaciones (X, Y, Z) y los ángulos en (X, Y, Z)

```
function pies(ox, oy, oz, ax, ay, az){
```

ya que esta función la vamos a utilizar 2 veces y ahorramos código, declaramos las geometrías a utilizar que en este caso fueron los cilindros, cubos y esferas y a partir de esto se importó la librería CSG para poder hacer operaciones booleanas como son (unión, intersección, Diferencia) y a partir de estar crear la forma un problema que tuve en esta función es que al crear esta pieza quedo muy grande con relación al animal y cambiar valores se iba a distorsionar todo el objeto, la solución que pude implementar era agregar una transformación de escalación para que se escalara y no altere ninguna parte del diseño .

```
t.set( sx, 0, 0, 0,  
      0, sy, 0, 0,  
      0, 0, sz, 0,  
      0, 0, 0, 1 );  
C2T2.applyMatrix(t);
```

- Nariz:

- Se inició declarando las geometrías a utilizar que en este caso fueron cubos y esferas y a partir de esto se importó la librería CSG para poder hacer operaciones booleanas como son (unión, intersección, Diferencia) y a partir de esto hacer la forma.

LIBRERÍAS:

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

- *PROCEDIMIENTO DE ELABORACIÓN PASO A PASO:*

FRONTAL

LATERAL

PERSPECTIVA

