

Instituto Tecnológico de Buenos Aires

2Q - 2022

Computación Gráfica - TP1
Alumno: Pedro López - 60711

Introducción

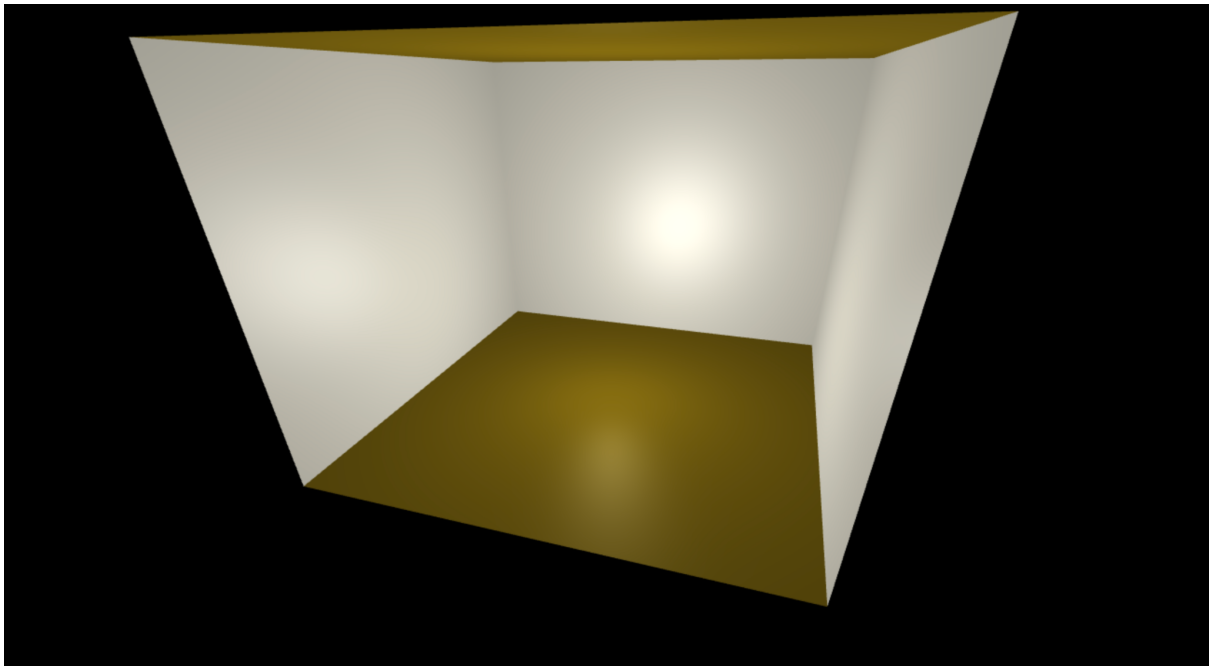
En este informe se busca detallar el desarrollo y funcionamiento del Trabajo Práctico 1 de la materia de Computación Gráfica. El mismo consistía en una escena en la que estaban presentes una impresora 3D, un carrito elevador y una estantería de dos niveles con ocho espacios cada uno. La impresora 3D debería imprimir una figura determinada, el carrito elevador debía ser capaz de recogerla y llevarla hasta un espacio libre en la estantería.

Para lograr esto, se utilizó la librería [Threejs](#) para el manejo del modelado y la renderización de la escena en un navegador.

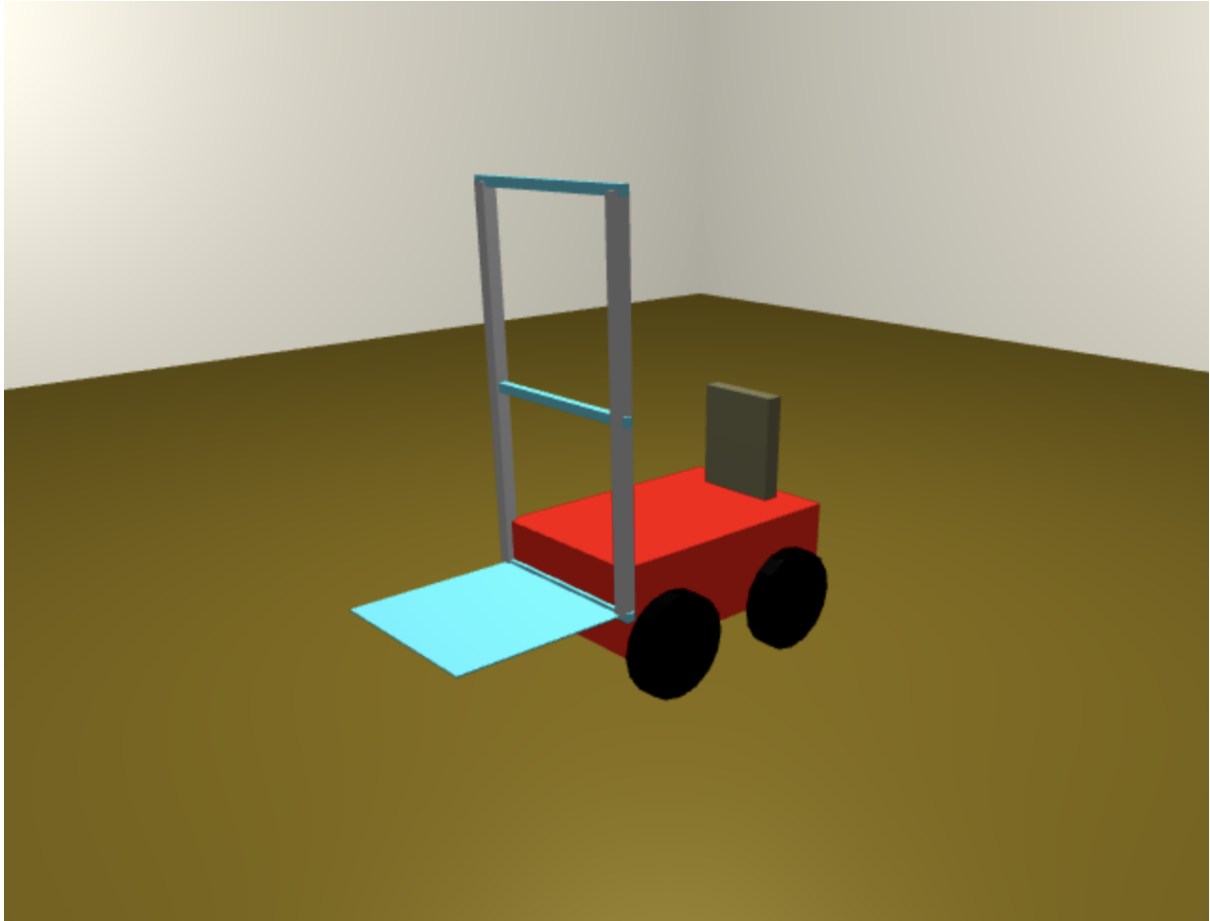
Desarrollo de la Aplicación

Modelado

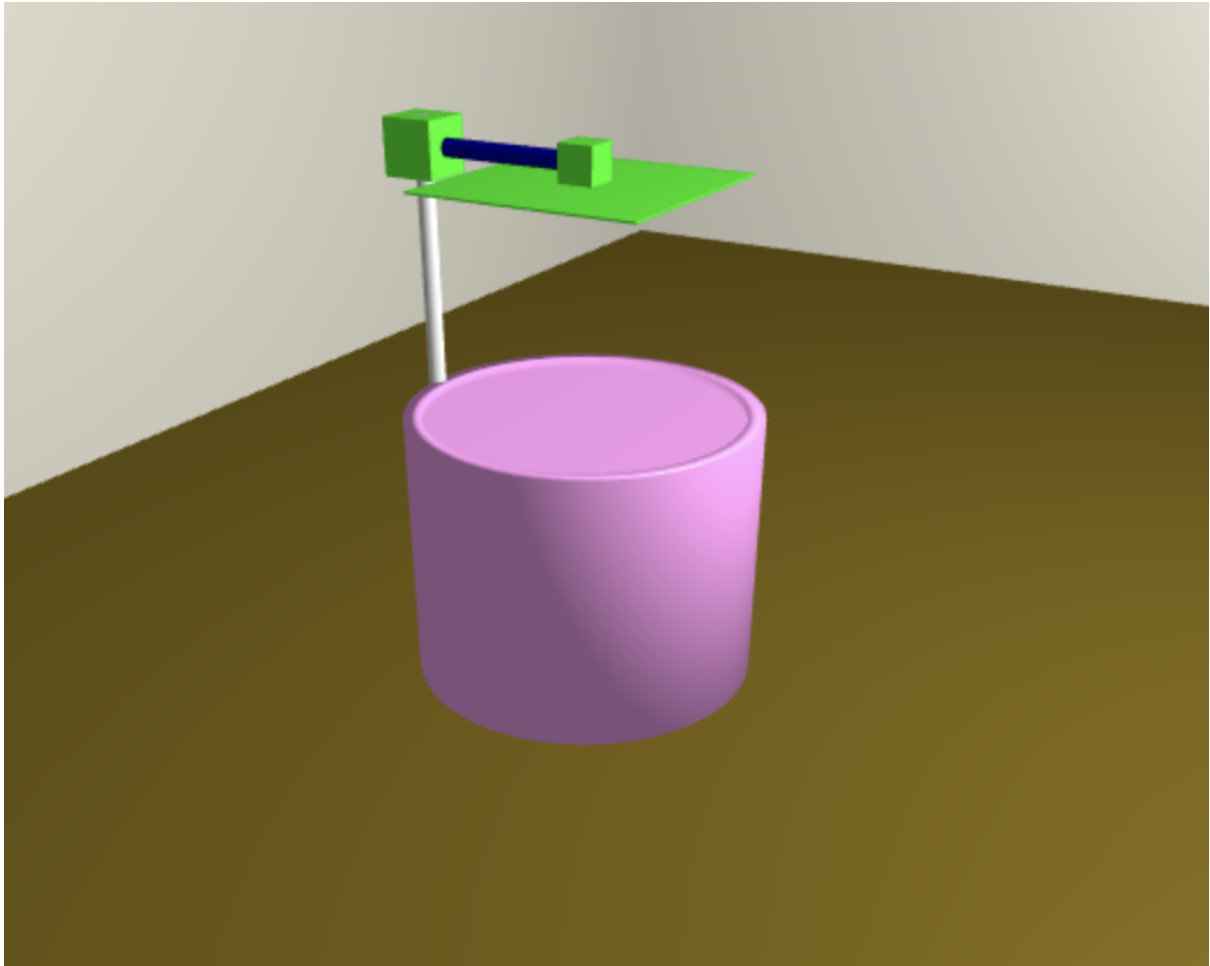
El primer paso para el desarrollo de la aplicación fue el modelado de todos los objetos que iban a ser necesarios para la escena. Se comenzó con el modelado de un cuarto compuesto por 4 paredes, un piso y un techo. Cada componente tenía una geometría plana y un material del tipo Phong.



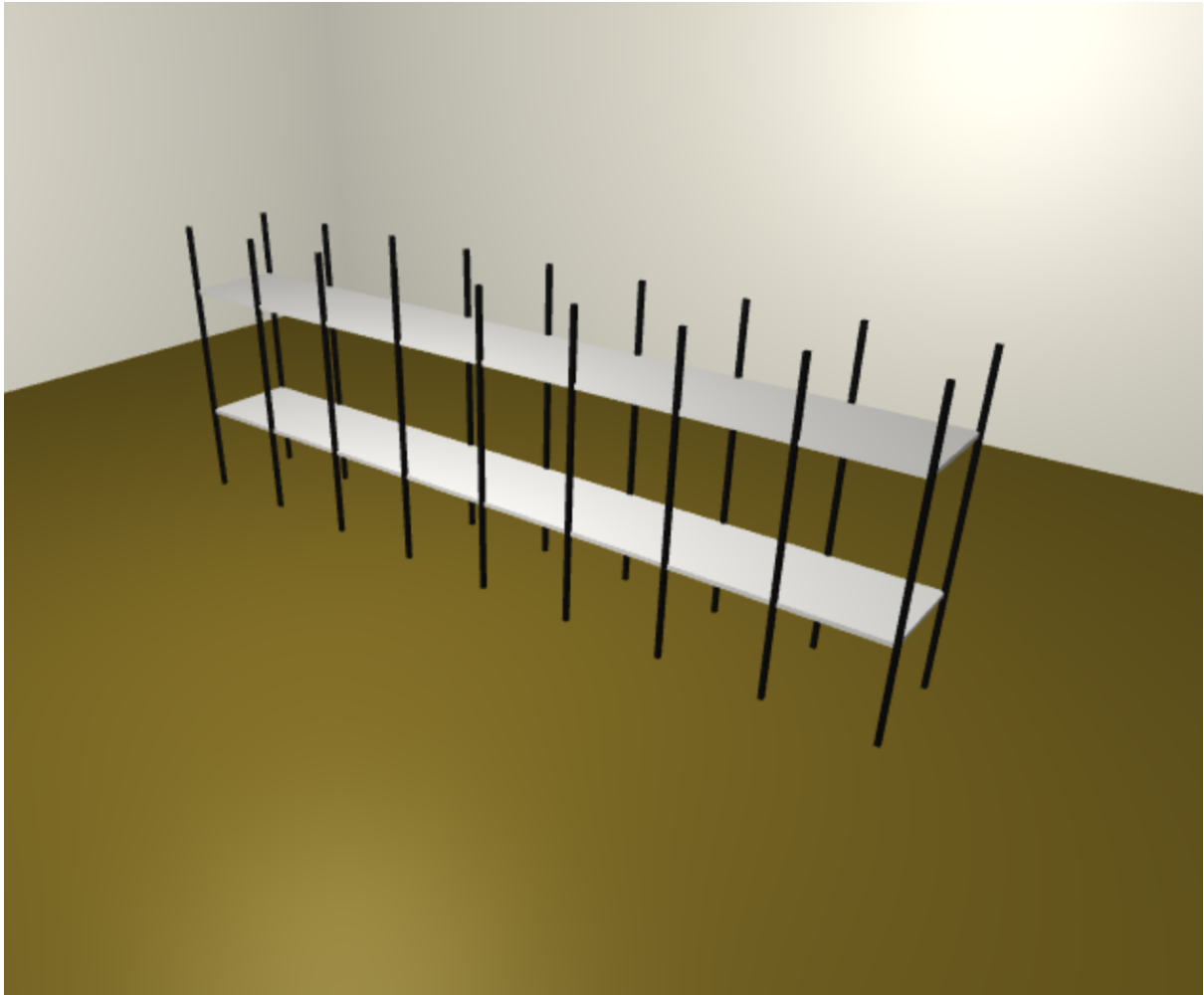
Luego se avanzó con el modelado del carrito elevador. Este está compuesto por una geometría del tipo box, la cual es el elemento padre del carro en la escena, a la cual se le agregaron cuatro geometrías cilíndricas rotadas sobre el eje z, para representar las ruedas. Además, se agregó una geometría box sobre el cuerpo del carrito para ilustrar un asiento. Luego se crearon geometrías box para el elevador del carro y por último una de tipo box con la dimensión en y muy cercana a 0 para la placa del elevador. Todos los materiales son de tipo Phong, con los colores como pueden apreciarse en la imagen.



Para continuar se modeló la impresora en 3D. La cual presenta una base la cual se construyó a través de una geometría de Revolución. Esta base se asemeja a una forma cilíndrica. Luego se creó un poste al cual se le agregó un brazo compuesto por varias geometrías y una plana al final para representar el plato del cual nacía la impresión. Al igual que con los demás modelos, los materiales utilizados son del tipo Phong.



Por último se modeló la estantería, la cual consiste de dos estantes creados a partir de geometrías del tipo box, y dieciocho patas creadas también a partir de geometrías del tipo box. Además en el centro de cada estante, se creó un objeto 3D vacío que serviría como contenedor de una posible impresión que fuese a ser depositada en ese lugar.



Luces

Se decidieron utilizar dos luces para la escena. Una de ambiente y una puntual.

Ambas luces son blancas. La luz de ambiente posee una intensidad de 0.5 en una escala de 0 a 1, y la luz puntual posee una intensidad de 0.75. La posición de la luz puntual es de 5 unidades sobre el origen en el eje y, simulando una lámpara en el techo de la habitación.

Controles

Para empezar, se desarrollaron los controles de movimiento del carrito elevador agregando un event listener del teclado. Utilizando las flechas del teclado o la popular

disposición W,A,S,D puede moverse el carrito, luego utilizando las teclas Q y E puede hacerse descender y ascender, respectivamente, la plataforma del elevador.

Luego, para recoger una impresión cercana a la plataforma del carrito, se utiliza la tecla G, al igual que si el carrito posee una impresión y desea dejarla en una estantería libre cercana.

Para la selección de cámaras, tenemos las teclas del 1 al 6 inclusive para seleccionar entre las distintas posibles cámaras.

Por último, con la tecla K se realiza una impresión si no hay otra presente en la base de la impresora. Con las teclas O y P puede hacerse zoom si la cámara seleccionada lo permite.

Cámaras

La escena posee 6 cámaras distintas, las cuales están contenidas en un objeto llamado "CameraController".

Las cámaras 1, 2 y 3 poseen un comportamiento orbital, cada una centrada en un objetivo distinto. Dichos objetivos son: Centro de la habitación, Impresora 3D, Estantería. Estas cámaras poseen un funcionamiento similar, en la cual se calcula su posición en cada eje utilizando coordenadas esféricas centradas en el objetivo de la cámara. El movimiento del mouse en las coordenadas x, y de la pantalla modifican el valor de los ángulos utilizados en dicho sistema de coordenadas. El nivel de zoom que controlan las teclas O y P modifican la distancia entre la cámara y el objetivo.

La cámara 4 se sitúa por delante del carrito mirando en la misma dirección en la que apunta el carrito. Para conseguir esto se agregó la cámara al objeto del carrito, de esta manera su posición y rotación dependen de la del carrito y no hace falta hacer un trabajo extra de cálculo.

La cámara 5 se sitúa detrás del carrito. Al igual que la cámara 4, su posición y rotación dependen del carrito, con un ligero offset para no apuntar directamente al carrito sino un poco más adelante. Esto le da una vista un poco más cómoda a la hora de utilizarla.

Por último, la cámara 6 también tiene posición y rotación dependiente del carrito, pero visto desde costado.

Modelado de Impresiones

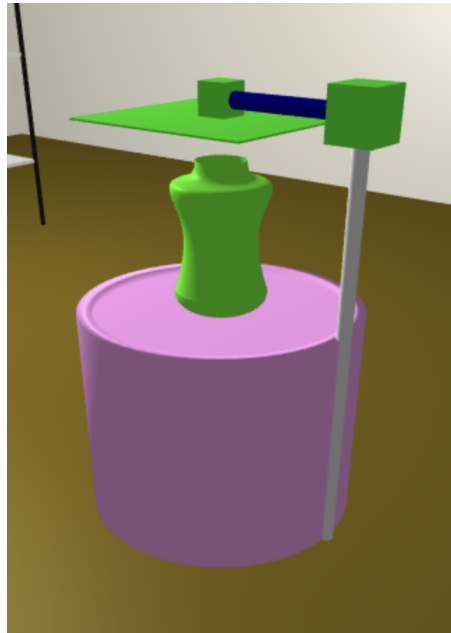
Tenemos dos tipos de impresiones 3D. Las de extrusión y las de Revolución. En ambos métodos es necesario trabajar con curvas Bézier y las clases "Shape" y "Path" de Threejs. Para facilitar el modelado de dichas curvas, se utilizó una [herramienta de visualización](#) en la cuál puede irse construyendo la curva a partir de las coordenadas dadas en forma de texto. Una vez construída la curva, se utilizó un pequeño script en Python para traducir dichas coordenadas al código necesario en javascript para crear los objetos "Shape" y "Path" necesarios. Este script se encuentra en los archivos.

Se creó un objeto llamado "Curves" del tipo key-value el cuál identifica las curvas por nombre y cada una posee en su value una función que retorna el objeto "Shape" o "Path" correspondiente.

En ambos tipos de impresión se recalcularon las normales de las geometrías utilizando la función built-in en Threejs, para tener un comportamiento esperado con la luz.

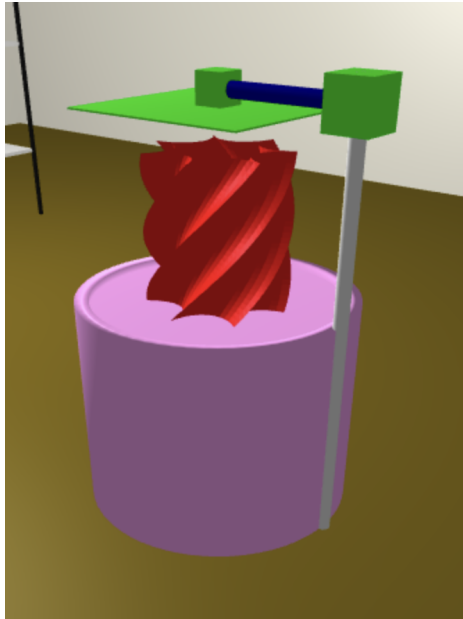
Impresiones de Revolución

Para las impresiones de revolución se utilizó la geometría de tipo “Lathe” de Threejs. A partir de la curva correspondiente, se realiza una rotación de vuelta completa. Luego se crea un material del tipo Phong con el color correspondiente.



Impresiones de Extrusión

Para las impresiones de extrusión se utiliza la geometría de tipo “Extrude” de Threejs. El problema es que esta geometría no tiene ningún método para realizar una torsión sobre el eje de la extrusión. Para resolver éste problema se utilizó una modificación de una [solución encontrada](#) luego de investigación en internet. Para lograr el efecto, se recorrieron los vértices de la figura, rotando el vértice en función del eje de extrusión dependiendo de la coordenada z del mismo, la altura total de la impresión y el ángulo total de torsión.



Animación de la impresión

Para simular que el objeto de la impresión estaba siendo creado y que no apareciera instantáneamente, se utilizó un Clipping Plane, que se sincronizó con la posición y orientación del plato de la impresora. De esta forma, da la sensación de que el objeto se va creando a medida que sube el brazo de la impresora.

Movimiento de la Impresión

Para trasladar la impresión desde la impresora hacia un estante, debe recogerse con el carrito elevador y moverse hasta posicionarse en uno de los estantes disponibles. Para lograr esto se definió una función para mover el objeto de impresión desde la base de la impresora a la plataforma del carrito, y otra función para mover el objeto de impresión desde la base del carrito a la estantería disponible más cercana. Dichas funciones chequean si la distancia entre la impresión y el objeto al que se la quiere trasladar es menor a cierto margen. Si es cierto, se remueve la impresión de los hijos del objeto que la contiene y se agrega al objeto al que se traslada. De esta forma no hace falta modificar la posición del objeto de impresión. Para calcular la distancia entre los objetos se tuvo que utilizar una funcionalidad de Threejs para obtener las coordenadas relativas al origen de la escena.

GUI

Para el desarrollo de la Graphic User Interface se utilizó la librería [dat.gui](https://github.com/donmccurdy/dat.gui). En ella se definieron distintos controles que se sincronizaban con los atributos de un objeto llamado "GUIController".

MANUAL DE USUARIO

Para ejecutar la aplicación, es necesario abrir el archivo ***index.html*** ubicado en la carpeta raíz.

Controles teclado:

- **Movimiento del carrito:** W,A,S,D o flechas
- **Elevar y descender la plataforma del carrito:** E, Q
- **Recoger/Dejar Impresión:** G
- **Realizar Impresión:** K

Controles GUI:

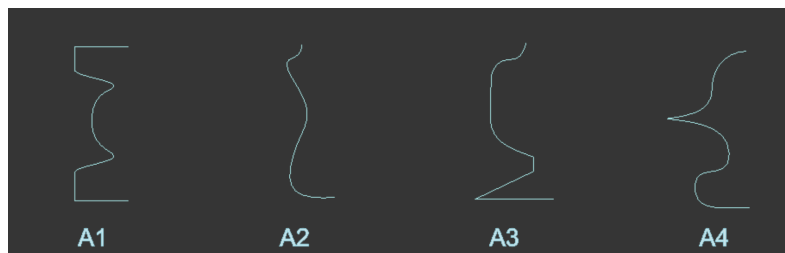
- **Print type:** Tipo de impresión a realizar. Puede ser “Extrusion” o “Revolution”
- **Revolution Shape:** La forma de la impresión por revolución.
- **Extrusion Shape:** La forma de la impresión por extrusión.
- **Twist Angle:** Ángulo de torsión total de la extrusión en radianes. De 0 a 2π
- **Print Height:** Altura de la impresión. De 0 a 1.5
- **Print Steps:** Cantidad de subdivisiones de la impresión. Mejora la calidad
- **Print Color:** Color de la impresión.

Elija el tipo y forma de impresión que se desea, entre revolución (se construyen girando una curva durante una vuelta entera) o extrusión (se construye expandiendo una curva en 2D de forma vertical, y luego se rota generando torsión). Luego, elija la altura y ángulo de torsión si es del tipo extrusión. Seleccione un valor de las subdivisiones de la impresión (mientras más sean, más suave será esta). Por último seleccione el color que más desee.

Maneje con el carrito hasta la impresión, acerque lo más que pueda la plataforma y presione G para recogerla. Maneje hasta la estantería y acerque la plataforma con la impresión lo más posible a un lugar libre con la estantería. Presione G para depositar la impresión en el lugar seleccionado de la estantería.

Tipos de curvas:

- Perfiles de Revolución:



- Formas de Extrusión:

