

# CC3501 - Modelación y Computación Gráfica para Ingenieros

## Tarea 2

Prof. Ivan Sipiran

10 de setiembre del 2021

Fecha de entrega: Viernes 01 de octubre del 2021 a las 23:59

### 1 Introducción

El objetivo de esta tarea es modelar un objeto 3D complejo usando primitivas simples (cubos, esferas , conos y cilindros) y modelación jerárquica (grafo de escena).

### 2 Trabajo a realizar

Para realizar esta tarea, hay que considerar lo siguiente:

1. **Importante:** actualice su repositorio local del curso antes de hacer esta tarea. Esto permitirá tener acceso a los programas de prueba y recursos para la tarea.
2. Se hace entrega de un conjunto de objetos 3D (en formato .obj), de los cuales debe escoger sólo uno para modelar. El enlace para descargar los modelos 3D está disponible en la descripción de la presente tarea en U-Cursos. Hay cuatro clases de objetos dentro de la colección: carros, aviones, guitarras y bicicletas. En cada clase hay a lo más 20 objetos 3D.
3. En el repositorio del curso hay un programa en Python llamado *ex\_show\_object.py* (**es necesario actualizar el repositorio para ver este programa**). Este programa les va a permitir visualizar alguno de los modelos descargados e inspeccionar su geometría para que puedan reproducirla con su modelación. Para ejecutar el visualizador es necesario que instalen el paquete **open3d** via el comando **pip**. El comando para ejecutar el programa es el siguiente:

```
>> python ex_show_object.py --model=<ruta a modelo 3D>
```

donde la ruta del modelo 3D debe incluir la ruta completa y el nombre del archivo en donde se encuentra el objeto.

4. Una vez que escogió el modelo 3D que le guste, debe modelarlo usando las primitivas básicas implementadas en el archivo *basic\_shapes.py* (**es necesario actualizar el repositorio para ver este archivo**). En este archivo hay implementaciones para las formas cubo, cilindro, esfera y cono. Estas formas básicas deben ser usadas junto a un grafo de escena que permita modelar jerárquicamente el modelo 3D seleccionado.

**Importante:** Su escena debe usar de forma óptima las primitivas que utilice. Por ejemplo, si su escena contiene cinco cubos amarillos, pero con distintas transformaciones, debería existir un único nodo hoja con un cubo amarillo referenciado desde distintos caminos del grafo.

Las primitivas básicas tienen las siguientes características:

- **Cubo:** centrado en (0,0,0) y con extensión 1.0 hacia cada lado. La función que crea este objeto se llama *createColorCubeTarea2*.
- **Esfera:** centrada en (0,0,0) y con radio 1.0. La función que crea este objeto se llama *createColorSphereTarea2*.
- **Cono:** centrado en (0,0,0), de altura 2.0 (1.0 hacia arriba de su centro y 1.0 hacia abajo de su centro) y de base circular con radio 1.0. La función que crea este objeto se llama *createColorConeTarea2*.
- **Cilindro:** centrado en (0,0,0), de altura 2.0 (1.0 hacia arriba de su centro y 1.0 hacia abajo de su centro) y de base circular con radio 1.0. La función que crea este objeto se llama *createColorCylinderTarea2*.

Todas las funciones para crear primitivas reciben como parámetro el color del objeto.

**Nota:** Estas funciones han sido creadas para que funcionen correctamente con el programa de shader del archivo de la tarea. Se recomienda hacer uso de estas funciones para que su modelo final pueda observarse correctamente en su escena. Si usted decide cambiar estas funciones o usar otras definidas por usted, es posible que requiera de trabajo extra para hacer que sus funciones propias funcionen bien.

El programa *tarea2\_v0.py* muestra un ejemplo de cómo crear los modelos y usarlos en un grafo de escena inicial (ver figura 1). En este programa, puede usar las teclas del 1 al 7 para ver su escena desde varios puntos de vista.

5. El modelo jerárquico debe contener como mínimo 15 primitivas y debe hacer uso de transformaciones para ubicar las primitivas, de tal forma que logre reproducir lo mejor posible el objeto seleccionado. La nota estará en función de cuanto se asemeja su modelo jerárquico final al modelo 3D seleccionado.
6. Documente su tarea usando comentarios en el mismo programa. Los comentarios deben permitir comprender el trabajo realizado y las decisiones tomadas. Un programa sin comentarios será penalizado con 0.5 puntos sobre el puntaje final

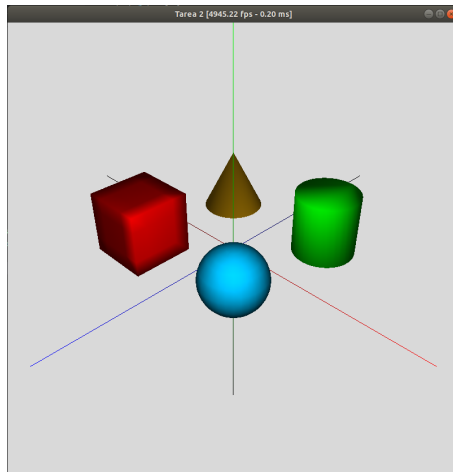


Figure 1: Escena inicial de la Tarea 2

logrado en la tarea. Un programa parcialmente documentado será penalizado con algún puntaje entre 0.0 y 0.5, según corresponda.

7. Indique claramente cuál es el modelo 3D seleccionado (la clase y el código del modelo), así se facilita la corrección para los ayudantes.

Si su programa utiliza algún parámetro que necesita usarse para poder ejecutarse, es necesario que especifique claramente cómo ejecutar el programa.

### 3 Entregable

Debe subir a U-Cursos su programa en Python (extensión .py) debidamente documentado y en el plazo de entrega.

### 4 Consideraciones

Debe tener en cuenta lo siguiente:

- El plazo de entrega es inamovible.
- El trabajo es individual.
- No está permitido el plagio del trabajo de sus compañerxs.

### 5 Bono a la creatividad

Se recibirá un bono de hasta 0.2 puntos en el promedio final de tareas a quien implemente una animación en donde intervenga el modelo jerárquico obtenido. También se puede

ganar esta bonificación si usa más de 30 primitivas en su modelación. Por ejemplo, algunas ideas de animaciones podrían ser:

- Hacer volar una flota de aviones que hacen piruetas (recuerde que si puede crear un avión con modelación jerárquica, puede instanciar muchos de ellos para una animación).
- Hacer girar las ruedas de una bicicleta, que se cae al final de su trayectoria.
- Crear una escena de tráfico con carros

Es necesario documentar el trabajo realizado para acceder a este bono y debe estar claramente señalado en los comentarios dónde se realizan los cambios que son parte del mismo.