Departamento de Ciencias de la Computación
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile
CC-3501 Modelacion y Computacion Grafica para Ingenieros

TAREA N°3

OpenSCAD vs OpenGL

Integrante: José Miguel Pacheco
Profesor: Nancy Hitschfeld
Auxiliares: Mauricio Araneda
Pablo Pizarro
Pablo Polanco
Ayudantes: Iván Torres
Maria Josá Trujillo
Fecha de entrega: 21 de agosto de 2018
Santiago, Chile

Índice de Contenidos

1. Descripción del problema **1**

2. Modelación en OpenSCAD **1**

3. Utilidad CSG **2**

4. Modelacion en OpenGL **2**

5. Ventajas y desventajas **3**

6. Resultados **3**

7. Conclusiones **6**

Lista de Figuras

1. Parte frontal de Bronzor modelado en OpenSCAD 3
2. Parte trasera de Bronzor modelado en OpenSCAD 4
3. Parte frontal de Magneton modelado en OpenSCAD 4
4. Parte trasera de Magneton modelado en OpenSCAD 5
5. Parte frontal de Bronzor modelado en OpenGL 5
6. Parte trasera de Bronzor modelado en OpenGL 6

1. Descripción del problema

El problema consiste en la realización de objetos en tres dimensiones con el software OpenSCAD y con OpenGl, para luego comparar las ventajas y desventajas de la utilización de cada software. Para llevar a cabo esta comparación se tenían que modelar 3 pokemon, dos en OpenSCAD de los cuales uno tenia que ser 'simple' (Bronzor) y el otro mas 'avanzado' (Magneton) y el tercero se tenia que modelar en OpenGL de los cuales se eligió nuevamente Bronzor.

2. Modelación en OpenSCAD

La modelación en OpenSCAD se lleva a cabo gracias al modelo de geometría constructiva de solidos, sabiendo esto, la modelacion de 'Bronzor' se llevo a cabo de la siguiente forma: Primero se crearon dos módulos para facilitar la creación de ciertos objetos, el primero fue el modulo hoja() que ocupando la función *hull()* con un circulo y un polígono (que se parece a un triangulo) permite obtener un objeto similar a una hoja, que al unirla con un rectángulo (tallo) se obtiene una objeto que simula un hoja con su tallo. El segundo modulo que se utilizo se llamó *anillo(r1,r2)* el cual ocupando difference() entre dos cilindros de altura 1 se obtiene un anillo de radio interno r1 y radio externo r2 con una altura de 1.

Lo siguiente fue crear el çuerpo"de Bronzor, para lo cual se crean distintos cilindros, el primero de altura 4 y de radio 25 con un color azulado, luego para la parte interna se creo otro cilindro de altura 4.6 (Un poco mas alto para superponer los colores) de radio 18 de color un poco mas oscuro que el anterior y con un anillo interior de radio interno 10.8 y radio externo 11 de color negro. Lo siguiente fue crear las esferas que rodean el cuerpo central del Bronzor, para lo cual con un *for* se crearon 6 esferas en el borde del cilindro grande a 60° de distancia entre ellas con un radio de 5 cada una. Después se crearon los círculos internos que en realidad son 4 cilindros de radio 2 ubicados a 90° de distancia entre ellos que fueron creados con un ciclo *for*, luego se creo una esfera de radio 4.6 que es la esfera central se prosiguió con los ojos para lo cual se crearon cilindros de color blanco y negro pero con distinta escala que simula ser una elipse y luego se trasladaron a la posición donde corresponden, y por ultimo se modelaron las hojas de la parte trasera del pokemon para lo cual se crearon las hojas con el modulo *hoja* mencionado anteriormente y se posicionaron en la posición que correspondía, una vez ubicadas se creo un prisma rectangular uniendo todas las hojas creadas formando el diseño.

Para modelar a Magneton, el segundo pokemon, se comenzó creando el imán, para lo cual se creo el modulo *iman()* en el cual se aplico la función *hull* entre un circulo y un rectángulo para formar una 'L' que estuviera redondeada en la esquina,para luego hacer una reflexión y quede una 'U'en la que a las puntas se le une un cubo azul y rojo para representar el imán. Luego se creo el modulo *semi(r)* que usando la función *difference()* entre una esfera y un cubo entrega una semi esfera de radio 'r'. Con este modulo se logro crear otro modulo llamado *tornillo()* que con la union de un cilindro y una semiesfera entrega un objeto

similar a un tornillo para poder colocarlo en el Magneton. Siguiendo con la modelacion de Magneton se creo el modulo *ojo()* que con la diferencia entre una esfera y una semiesfera y luego con una unión con otra semiesfera pero ahora de color blanco se logra modela el 'ojo' de Magneton, y se realiza el mismo proceso para la pupila pero en este caso uniendo una semiesfera de color negro. Una vez teniendo todos los módulos mencionados anteriormente se crea otro modulo llamado *cabeza()* que une todos los anteriores para lograr crear una parte de Magneton, esto graacias a diferentes rotaciones, traslaciones y escalamientos de los módulos *iman()* y *tornillo()* y por ultimo con el modulo *cabeza()* se termino modelando Magneton completamente, ya que se crearon 3 las cuales se trasladaron y rotaron para obtener el objeto deseado.

3. Utilidad CSG

La principal utilidad de la geometría constructiva de solidos es la posibilidad de trabajar con operadores booleanos, donde la operación unión corresponde a la fusión de dos objetos y la operación de intersección o de diferencia corresponde a la operación de vaciado, donde un objeto es el que se modela (pierde material) y el otro define la forma en que se hace el agujero. En particular para el software OpenSCAD la posibilidad de ocupar CSG es una gran ventaja, ya que permite que el trabajo sea muy intuitivo además de fácil de usar.

4. Modelacion en OpenGL

La modelacion en OpenGL es prácticamente la misma que en OpenSCAD ya que el proceso que se siguio es el mismo salvo que en OpenGL se utilizan sus propias funciones y funciones auxiliares que están en CC3501Utils que son cilindro, esfera y elipse (esta ultima añadida por mi para un mejor aspecto de los ojos). Para modelar el pokemon, en este caso Bronzor, primero se creo un cilindro con las diferentes tonalidades de azul para el 'cuerpo', posteriormente con un for se crearon los círculos interiores que en realidad son cilindros y con otro for se crearon las esferas externas, para la esfera interna simplemente se creo una esfera del mismo color que el cilindro exterior y para los ojos se ocupo la función elipse, creando una elipse blanca y dentro otra elipse mas pequeña de color negro. Luego para visualizarlo se ocupo el método utilizado en el auxiliar de 3D, con el cual se utilizan las flechas arriba y abajo para acercar y alejar respectivamente y las teclas 'w' y 's' para rotar la cámara hacia arriba y abajo respectivamente.

5. Ventajas y desventajas

Las ventajas de trabajar con OpenSCAD son muchas, ya que es mas sencillo e intuitivo al momento de modelar objetos en 3D y mucho menos engorroso que OpenGL, ya que en OpenGL se tiene que definir todo lo que se dibuja, las figuras, además que al momento de visualizar lo modelado, en OpenGL es mas difícil poder observar alrededor del objeto, en cambio en OpenSCAD la interaccion con el objeto es mucho mas amigable. Ademas la modelacion en OpenSCAD queda mas bonita que en OpenGL (es decir los graficos).

6. Resultados

A continuacion se muestran los resultados de modelar distintos objetos en OpenSCAD y OpenGL, los primeros dos en OpenSCAD y el tercero en OpenGL

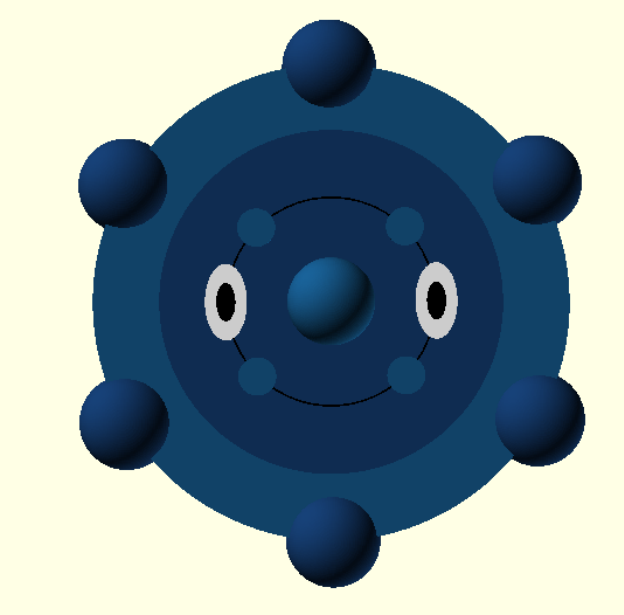

Figura 1: Parte frontal de Bronzor modelado en OpenSCAD

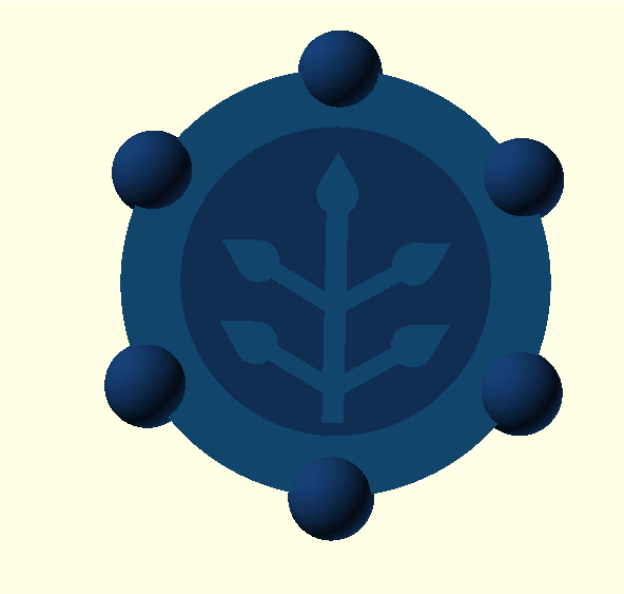

Figura 2: Parte trasera de Bronzor modelado en OpenSCAD

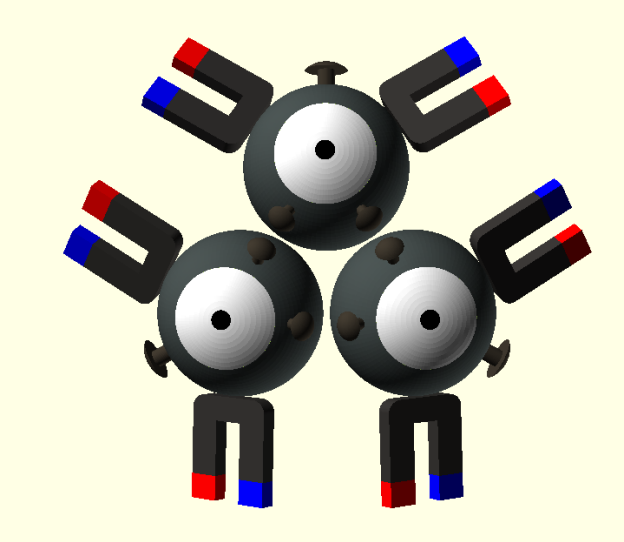

Figura 3: Parte frontal de Magneton modelado en OpenSCAD

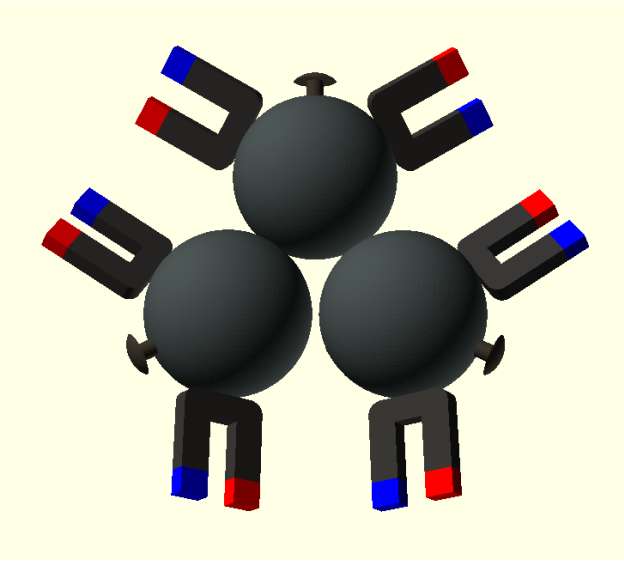

Figura 4: Parte trasera de Magneton modelado en OpenSCAD

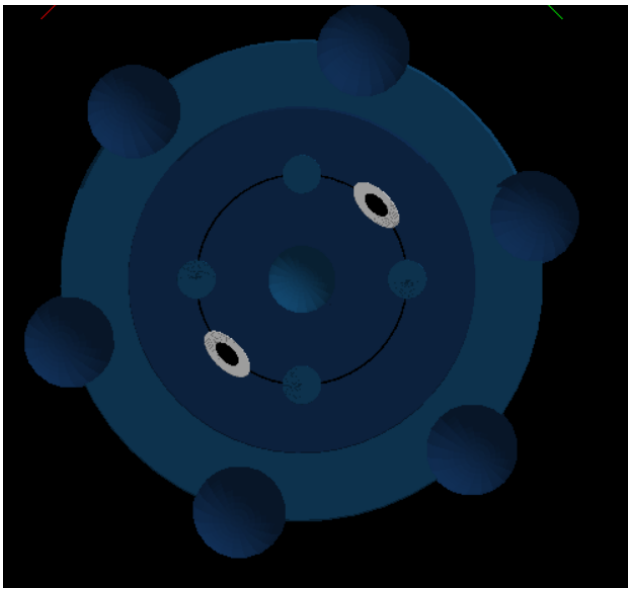

Figura 5: Parte frontal de Bronzor modelado en OpenGL

Figura 6: Parte trasera de Bronzor modelado en OpenGL

7. Conclusiones

Después de la modelacion de 3 objetos en 3D, dos en OpenSCAD y uno en OpenGL se puede concluir que el trabajo con OpenSCAD es mucho mas sencillo que con OpenGL y mucho mas intuitivo, ya que en OpenGL es mas engorroso a la hora de crear el objeto como a la hora de visualizar el objeto en la pantalla. Además OpenSCAD posee una mayor cantidad de funciones que hacen aun mas fácil el trabajo ya que es un Software dedicado a la modelación de objetos.