

ENTORNOS VIRTUALES

Modelos procedurales

Lidia Sánchez Mérida

Introducción

Los modelos procedurales son modelos generados automáticamente a partir de algoritmos en lugar de realizarlos de forma manual. Sus principales ventajas residen en la personalización de la generación de los modelos a través de la parametrización de los algoritmos, en la rapidez y capacidad de crear un mayor número de modelos, e incluso en la posibilidad de incluir cierta aleatoriedad para modelar un juego más interesante y con un comportamiento menos predecible.

Sin embargo, entre sus aplicaciones también se encuentra el sector de la construcción puesto que esta técnica también es ampliamente utilizada para añadir modificaciones a un modelo de arquitectura ya construido. De ese modo se pueden diseñar diversas variaciones de un mismo modelo para, por ejemplo, construir un vecindario de casas, cada una con unas características particulares.

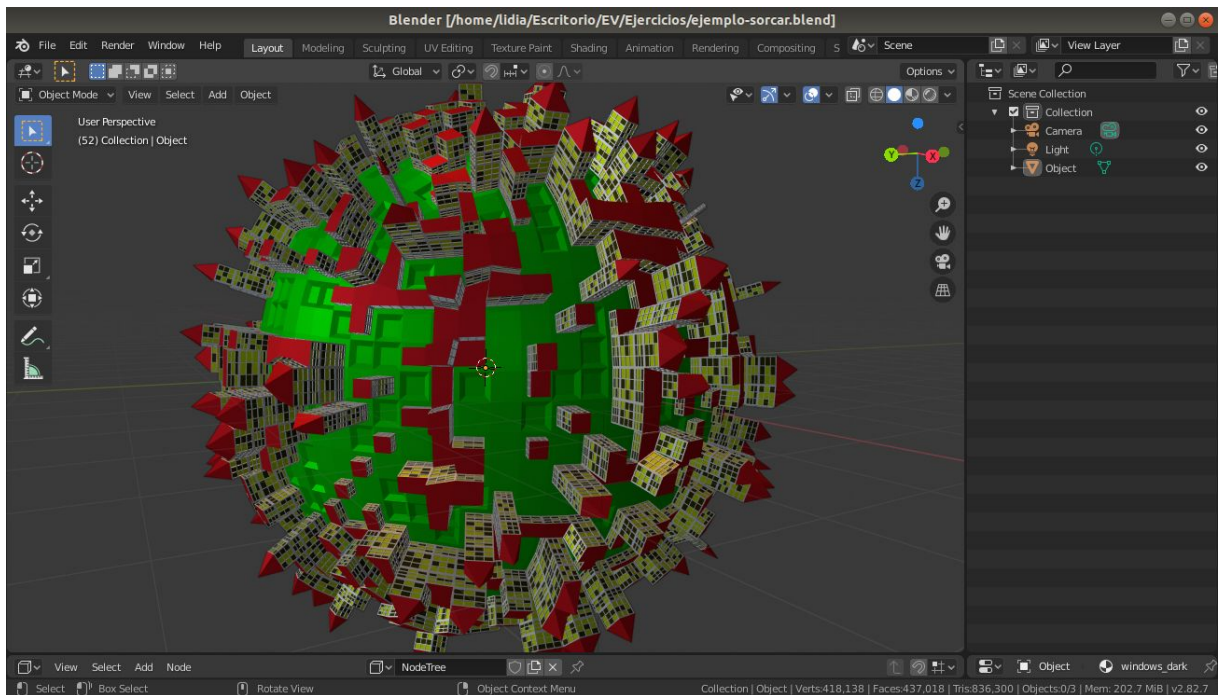
Herramienta *Sorcar*

Si bien existen varias herramientas para realizar modelos procedurales, [Sorcar](#) destaca por realizar modelos procedurales mediante un **sistema basados en nodos**. Para ello esta herramienta se postula como un **plugin para Blender** puesto que utiliza su API de Python para la creación de entornos virtuales en los que poder desarrollar los modelos procedurales. Esta herramienta se ha inspirado en una segunda denominada [Houdini](#), la cual también se basa en un sistema de flujos de trabajo pero a un nivel más profesional. Ambas disponen de un funcionamiento basado en proporcionar un editor en el que poder crear y visualizar en tiempo real flujos de trabajo, aplicando operaciones propias de Blender en el caso de *Sorcar*.

Ejemplo: modelado de un planeta

Para añadirlo como *plugin* es necesario disponer de una **versión de Blender 2.8 o superior**, por lo que he tenido que descargar esta función para probar esta herramienta. Para incorporarla basta con bajarse el repositorio completo de la herramienta y añadir el fichero *zip* como *plugin* en Blender. Investigando acerca de esta herramienta he encontrado un foro específico de Blender especializado en Sorcar en el que se proporcionan una multitud de ejemplos. Entre ellos he seleccionado el que me ha parecido más completo e interesante para recrearlo yo misma. El objetivo es elaborar un grafo para modelar un

planeta compuesto por edificios alrededor del mismo. La siguiente captura ilustra mejor el resultado de la escena.



A continuación se resume la creación de los nodos adecuados así como su interconexión para llegar al resultado final mostrado anteriormente.

1. Una vez disponemos del editor de nodos de *Sorcar*, creamos un nuevo grafo cuyo primer nodo va a **crear un cubo** en la escena. Sobre él se aplicarán una serie de **transformaciones**, cada uno en un nodo diferente, para subdividir la pieza en más caras y así proporcionarle cierta forma esférica.
2. A continuación creamos nuevos nodos para, en primer lugar seleccionar aleatoriamente un conjunto de caras con la operación *Select Random* para posteriormente separarlas del objeto con la opción *Separate*. El objetivo es poder modelar los edificios con dichas caras de forma separada al modelado del terreno del planeta.
3. Sobre las caras seleccionadas anteriormente, se escogen, a su vez, subgrupos de forma aleatoria para añadir más caras sobre ellas de manera que se modelen **las plantas de los edificios**. Para ello haremos uso de la operación anterior *Select Random* y la operación *Inset* para insertar más caras dentro. Cada una de estas operaciones a aplicar se representará como un nodo individual conectado al esquema actual.
4. Al mismo tiempo en el que se modela la estructura de los edificios se añaden nodos intercalados para aplicar su **color** correspondiente. Para ello se han definido cinco materiales: uno para el terreno de color verde, otro para los tejados de color rojo, las paredes en un tono amarillo pastel y dos materiales más para representar las ventanas con luces en su interior (amarillas) y las ventanas sin luz (color oscuro).

5. Una vez hemos terminado la estructura de los edificios, vamos a modelar los **tejados** de algunos de ellos. Para ello añadimos un nodo más al esquema para aplicar la operación *Poke* que convierte las caras seleccionadas en triángulos.
6. A continuación modelamos las **ventanas de los edificios** realizando el mismo procedimiento que en pasos anteriores, es decir, subdividimos los edificios creando un mayor número de caras para posteriormente insertar nuevas con las que componer los marcos de las ventanas. De nuevo las operaciones a utilizar son *Subdivide* e *Inset* y cada una de ellas es un nodo diferente y conectado al grafo actual.
7. A continuación añadimos el material para simular las **ventanas con luz** en el interior añadiendo un nodo más para aplicarlo. Para darle un mayor realismo, ahora seleccionamos algunas de ellas aleatoriamente con un nodo que aplique la operación *Select Random* y le aplicamos el material más oscuro para simular **ventanas sin luz** en el interior de sus habitaciones.
8. Finalmente cuando ya disponemos del terreno y los edificios modelados, **unimos ambos objetos** añadiendo un nuevo nodo para aplicar la operación *Merge*. Para evitar que existan vértices dobles añadimos dos nodos más para seleccionarlos todos mediante la operación *Select All* y los eliminamos con la operación *Remove Doubles*.

Conclusiones

Este tipo de modelado me parece muy útil puesto que es una forma rápida de poder recrear un modelo personalizando cada una de sus versiones con unos detalles particulares. En particular, en mi opinión **Sorcar facilita este proceso** puesto que en lugar de código se proporciona un editor con el que elaborar un grafo cuyos nodos son las operaciones a realizar para diseñar la escena. Además, otra de sus ventajas es que dichas **operaciones son las propias de Blender**, por lo que si estás familiarizado con esta herramienta la incorporación de *Sorcar* no supone ningún problema sino más bien bastantes beneficios.

Sin embargo, existen dos aspectos negativos a destacar. El primero de ellos es que para escenas medianamente complejas como la que he recreado el disponer de cada operación en un nodo individual hace que los grafos sean gigantescos. Este inconveniente ya ha sido detectado según el control de versiones en *GitHub* puesto que como tarea futura van a implementar la **agrupación de nodos** de modo que se pueda visualizar el grafo de forma más sencilla.

El último inconveniente que he encontrado es que **cuanto más complejo es el grafo, mayor lentitud** a la hora de procesar los cambios. Tanto es así que en varias ocasiones, al aplicar las operaciones de los últimos nodos he experimentado el cierre de Blender de forma inesperada. He estado investigando acerca de este suceso pero no he encontrado explicación alguna, por lo que mi teoría es que este tipo de procesos requieren de una gran cantidad de recursos y la gestión y limitación de estos es crucial para evitar este tipo de comportamientos anómalos.