

Edición de Escenas 3D sobre OpenSceneGraph

Jesús Zarzoso, María Ten, Jordi Torres, Rafael Gaitán, Javier Lluch

Abstract

En este artículo, presentamos un sistema de edición y manipulación de objetos 3D multiplataforma y de código libre, programado sobre OpenSceneGraph. El sistema está compuesto por una librería, que permite la manipulación de los objetos mediante operaciones de traslación, rotación y escalado, y una aplicación de edición de escenas 3D basada en la librería, y desarrollada como una extensión de la plataforma Eclipse.

1. Introducción

La Consellería de Infraestructura y Transportes (CIT) de la Generalitat Valenciana está realizando actualmente una migración de sus sistemas informáticos a software libre. Como parte de este proceso, se decidió crear un Sistema de Información Geográfica (SIG) propio, llamado **gvSIG** [gvs03]. Un SIG es un sistema utilizado para capturar, almacenar, analizar y gestionar datos espacialmente referenciados.

Uno de los proyectos que forman gvSIG consiste en la creación de una visualización en 3D del SIG. El conjunto de librerías base de este proyecto está siendo desarrollado en el Instituto de Automática e Informática Industrial (AI2) de la Universidad Politécnica de Valencia.

Una de las aplicaciones de un SIG es la inclusión personalizada de símbolos en las entidades incorporadas a los mapas. En un SIG 3D, los modelos clásicos de simbología 2D son insuficientes. Con la inclusión de una simbología 3D, se necesita un sistema que permita poder transformar esos objetos dentro del mapa 3D, para adecuarlos correctamente a la vista requerida.

El trabajo que se presenta en este artículo, intenta dar una solución al problema de la edición de símbolos 3D en un SIG. Al ser un trabajo en desarrollo, se está llevando a cabo en varias fases. Inicialmente se ha realizado una librería base, que permite la manipulación de objetos 3D, posteriormente se ha creado una aplicación de edición que ha servido para validar la funcionalidad. La última fase es la integración con gvSIG, que todavía no ha comenzado.

El artículo se estructura de la siguiente forma. Primero se hace un repaso de las tecnologías en las que se ha basado el desarrollo. A continuación, se presentan los progresos obtenidos para la edición de objetos 3D. Posteriormente se

presenta la aplicación de edición y finalmente, se hace un repaso a los trabajos que de éste se puedan derivar, incluyendo la integración con gvSIG, así como las conclusiones a las que se han llegado después de la realización de este proyecto.

2. Antecedentes

La edición de objetos 3D es una tarea que puede realizarse de diferentes maneras. En [SSS01] y [SSS] se puede leer sobre distintas técnicas de manipulación 3D basadas en una interfaz de usuario 2D.

En trabajos previos se presentó GeoViewer3D [GTLS06] donde se comenzaba el desarrollo y validación de un conjunto de librerías base para SIG 3D. Actualmente la librería *OSG Virtual Planets* (osgVP) ha evolucionado su desarrollo permitiendo extensibilidad y nuevos módulos, combinando desarrollo en Java y en C++ haciendo uso de la tecnología JNI y la librería gráfica OpenSceneGraph(OSG) [OB04].

El proyecto OSG dispone de una librería de manipulación de objetos llamada **osgManipulator**. Ésta se ha desarrollado tomando como base el artículo [PSSS02]. A continuación, se presentan los conceptos básicos sobre los que se fundamenta su diseño.

Se puede definir a un **manipulador** como un conjunto de *dragger*, orden y selección.

Un *dragger* es un elemento gráfico que responde a las acciones que se realizan con el ratón sobre él. Dependiendo de la acción realizada y del tipo de *dragger* se ejecutarán unas órdenes u otras. Existen muchos tipos de *dragger*s. En la figura 1 se ven algunos de los disponibles. Hay de escalado, de giro, de traslación, e incluso algunos que permiten distintas operaciones al mismo tiempo. Las **órdenes** se generan

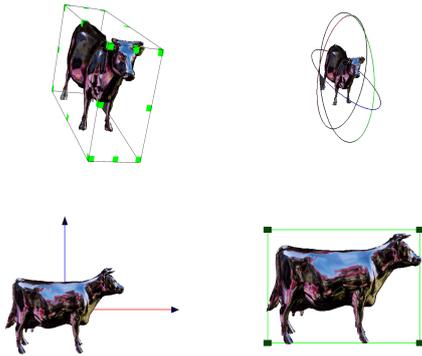


Figure 1: Distintos tipos de draggers

por las interacciones del ratón con los *draggers*, y son traslaciones, rotaciones y escalados. La **selección** es el objeto o conjunto de objetos sobre los que actúan las órdenes.

Existe en la actualidad una aplicación de edición 3D construida sobre OSG llamada OSGEEdit [osg02]. Sin embargo, presenta bastantes carencias en cuanto a la interfaz de usuario y su usabilidad. En primer lugar, la selección de los objetos que se desean manipular no se puede realizar interactivamente con el ratón, y para añadir nuevos elementos a la escena deben crearse nuevos nodos desde la vista del grafo, por lo que una persona sin experiencia de trabajo con grafos de escena encontrará dificultades al usar este programa. Otra carencia destacable es la ausencia de distintos tipos de manipuladores. Existe únicamente un manipulador, el cual permite hacer las operaciones de traslación, escalado y rotación.

Para el desarrollo de aplicaciones de escritorio Java, existe la plataforma de desarrollo Eclipse [ecl01] que ofrece la posibilidad de crear aplicaciones que hagan uso de su interfaz de usuario y de sus propios componentes nativos al sistema operativo. Éstas pueden construirse como extensiones, que se cargan junto con el entorno Eclipse, o pueden ser creadas como programas independientes, llamadas RCP (Rich Client Platform). Esta última opción es de especial interés puesto que utilizando este método, se pueden obtener productos finales con una interfaz muy flexible y de apariencia profesional sin dedicar mucho tiempo a su creación.

En la actualidad, una gran cantidad de software, tanto libre como comercial, se ha desarrollado como un RCP de Eclipse. Aplicaciones conocidas como Azureus [azu03] o Worldwindrcp (RCP que implementa Nasa Worldwind) [wor05] son ejemplos de RCP. Se pueden ver más ejemplos de ello en [rcp08].

3. Análisis y Desarrollo

El objetivo del proyecto es la creación de una librería que permita editar objetos 3D. Las operaciones de edición que

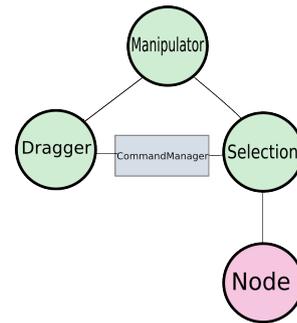


Figure 2: Estructura de un nodo Manipulator

debe permitir son traslaciones, rotaciones y escalados. Todas las operaciones se harán mediante acciones del ratón, simplificando al máximo la usabilidad de las aplicaciones que la utilicen.

La implementación de la librería osgVP-manipulator se ha realizado sobre osgManipulator, aprovechando todas las características que ofrece. Puesto que el uso de esta última se hace complicado y poco cómodo para el programador, se ha tratado de lograr una funcionalidad más sencilla. Para lograr esto, se ha adoptado el patrón de diseño **Facade** [EJ94], introduciendo un nuevo tipo de nodo; **Manipulator**. También se han añadido nuevas funcionalidades mediante la creación de un nodo llamado **EditionManager**.

Como se ha visto, la librería osgManipulator implementa el concepto de manipulador, sin embargo, esta implementación está realizada de una forma poco intuitiva. No existe un nodo manipulador que se pueda añadir al grafo de escena, en su lugar, existen varios nodos que, conectados de forma adecuada, forman el manipulador. Entre las clases que ofrece podemos destacar las siguientes: *Dragger*, *Selection* y *CommandManager*.

El nodo *Dragger* implementa el concepto de *dragger* que se ha visto con anterioridad. osgManipulator dispone de varios tipos, aunque se pueden desarrollar nuevos si se desea. El nodo *Selection* es un nodo al que se añade el subgrafo a editar. Interiormente tiene una matriz de transformación, que será la que recibirá las órdenes enviadas por el *dragger*, y hará que sus nodos hijos queden transformados. La clase *CommandManager* sirve para crear la conexión que permite mandar las órdenes desde el *dragger* a la selección.

Esta forma de crear los manipuladores implica que por cada uno de los que se quiera incluir haya que crear un nodo *Dragger*, uno *Selection*, y conectarlos mediante un *CommandManager*.

El nuevo nodo Manipulator permite, de una forma muy sencilla, añadir un nodo al grafo que incluya un manipulador sobre un objeto. En la figura 2 se observa cómo está formado el subgrafo creado por este nodo.

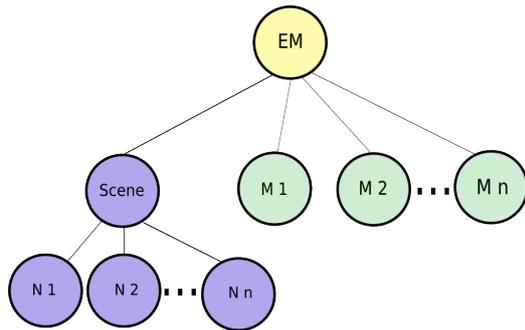


Figure 3: Estructura de EditionManager: EM es el nodo EditionManager, N son nodos y M son manipuladores

Cuando se crea un nodo Manipulator, es posible hacer una serie de acciones sobre él, como incluir un objeto, borrar un objeto, cambiar el tipo de *dragger* o quitar el *dragger* a la selección.

El nodo **EditionManager** actúa como un gestor de edición, y es el encargado de llevar el control de todos los manipuladores presentes en la escena, así como de realizar operaciones con ellos. En la figura 3 se puede observar su estructura interna. Consiste en un nodo, el cual contiene como primer hijo, un subgrafo con la escena a editar. El resto de hijos serán nodos **Manipulator** que se irán creando por el propio EditionManager cuando reciba órdenes de añadir nodos a manipuladores. Una aplicación que desee editar una escena, deberá crear un nodo EditionManager y añadir la escena a editar. Después de esto, las acciones más importantes que se pueden realizar son: añadir un objeto a un manipulador, extraer un nodo de un manipulador, agrupar objetos y desagrupar objetos.

- Añadir un objeto a un manipulador: Consiste en descolgar el nodo de la escena y crear un nuevo manipulador con él dentro.
- Extraer un nodo de un manipulador: No se descolgará directamente el nodo del manipulador, si no que se creará un nodo tipo transformación con la matriz de transformación igual a la del nodo seleccion. Esto es así para preservar las transformaciones, de otro modo el objeto quedaría como estaba antes.
- Agrupar objetos: Varios objetos contenidos en distintos manipuladores se agrupan en uno solo. Así, las transformaciones se aplicarán a todos los objetos al mismo tiempo. Se extraerán los nodos de todos los manipuladores y se añadirán a uno nuevo, quedando todos bajo la misma matriz de transformación, pero conservando todas las transformaciones anteriores.
- Desagrupar objetos: Una de las opciones más interesantes ofrecidas por el gestor de edición que se ha desarrollado es la posibilidad de desagrupar los objetos presentes en un manipulador. Esta operación sólo estará disponible

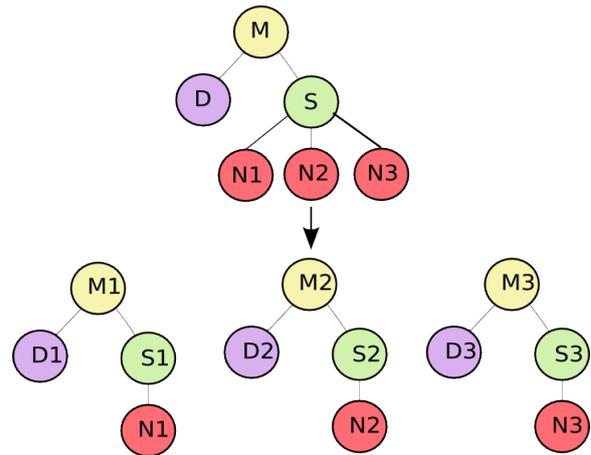


Figure 4: Proceso de desagrupar: M son manipuladores, D draggers, S selecciones y N nodos

cuando haya un sólo manipulador activo. El algoritmo recorrerá el subgrafo contenido en la selección hasta encontrar un nodo con más de un hijo, entonces, a partir de ahí, dividirá el manipulador en tantos como hijos tenga. La figura 4 muestra cómo un manipulador con tres nodos se separa en tres manipuladores distintos. Esta operación es muy útil cuando se tiene objetos con su geometría creada de forma jerárquica. De esta forma, se puede editar cada parte por separado y no todo el objeto a la vez.

4. Aplicación: Editor RCP

La aplicación Editor RCP hace uso de osgVP-manipulator y permite la carga y manipulación de objetos 3D para realizar la composición de escenas. Dispone de una interfaz de usuario sencilla, con distintas vistas y editores.

La aplicación permite importar modelos 3D desde diversos tipos de archivo (3ds, iva, osg, etc.) y editarlos directamente con el ratón. Se pueden crear tantas vistas de la escena como sea más cómodo al usuario, modificando su tamaño y posición en pantalla de forma muy sencilla.

En la figura 5 se puede observar la interfaz de usuario de la aplicación. En la imagen se observan cuatro zonas:

1. Barra de botones con todas las acciones que se pueden realizar. De izquierda a derecha: creación de nuevo proyecto, abrir proyecto desde archivo, guardar proyecto, añadir un objeto a la escena desde archivo, borrar la selección, desagrupar, agrupar, cambiar el tipo de *dragger*, mostrar/ocultar los ejes, activar/desactivar la iluminación, cambiar a vista ortográfica/perspectiva, guardar una captura de pantalla, activar/desactivar modo alámbrico, activar distintos manipuladores de cámara (planeta, trackball, vuelo, conducción).

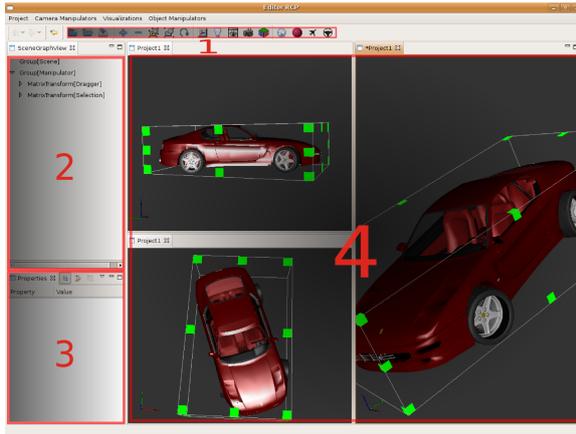


Figure 5: Captura de pantalla de la aplicación Editor RCP

2. Vista que incluye la representación jerárquica del grafo de escena. Esta vista permite realizar distintas acciones sobre la escena. Se pueden añadir manipuladores sobre los nodos haciendo doble click sobre ellos, también es posible cambiar las texturas y *shaders* de los nodos mediante un menú contextual que aparece al hacer click con el botón derecho del ratón sobre ellos.
3. Vista de propiedades del elemento seleccionado en la vista del grafo de escena. Las propiedades mostradas son diferentes según el tipo de nodo seleccionado. Es posible modificarlas mediante acciones sobre cada campo en concreto. Permite cambiar el nombre de nodo, la textura, etc.
4. Vista 3D de la escena. Se pueden crear tantas vistas como sean necesarias, y ajustar su tamaño y posición como se prefiera.

La implementación de Editor RCP se ha realizado como una extensión de Eclipse. Su arquitectura interna está compuesta por dos extensiones, la primera de ellas ofrece la funcionalidad de visualización 3D, mientras que la segunda añade todas las operaciones de edición y gestión de proyectos. La aplicación final se ha creado como un RCP que agrupa los plugins y puede ser ejecutado de manera independiente, sin necesidad de Eclipse.

5. Conclusiones y trabajos futuros

Se ha conseguido implementar una librería de manipulación 3D que permite editar objetos de forma sencilla mediante acciones con el ratón. Se ofrece una amplia variedad de manipuladores y operaciones que facilitarán las acciones de edición. El uso de la librería es muy accesible y permite una integración con otras aplicaciones desarrolladas sobre OSG. La librería está totalmente preparada para su integración en gvSIG.

Por último, se ha desarrollado una aplicación de edición

que permite importar modelos 3D y generar escenas que se podrán exportar en distintos tipos de archivo. Se ha conseguido una funcionalidad más sencilla e interactiva que otras soluciones similares basadas en OSG, a la vez que una interfaz de usuario con muchas más posibilidades.

Todavía no se ha realizado la integración de la librería en gvSIG, tarea que se realizará en los próximos meses. Se planea también integrar la librería en GeoViewer3D.

Se está realizando actualmente una ampliación que permite la edición de geometrías por vértice. Con ello se puede realizar la deformación de terrenos y objetos modificando la posición de sus vértices individualmente.

6. Agradecimientos

Este trabajo está siendo financiado por la Conselleria d'Infraestructures i Transport de la Generalitat Valenciana (Spain) y en parte por el proyecto TIN2005-08863-C03-01 del Ministerio de Educación y Ciencia.

References

- [azu03] Azureus, 2003. <http://azureus.sourceforge.net/>.
- [ecl01] Eclipse, 2001. <http://eclipse.org/>.
- [EJ94] ERICHGAMMA RICHARDHELM R., JOHNVLISIDES: *Design Patterns: Elements of Reusable Object-Oriented Software*. AddisonWesley Professional (November 10, 1994), 1994.
- [GTLS06] GAITÁN R., TEN M., LLUCH J., SEVILLA L. W.: Geoviewer3d: 3d geographical information viewing. In *Ibero-American Symposium on Computer Graphics (SIACG) (2006)*, Brunet P., Correia N., Baranowski G., (Eds.).
- [gvs03] gvsig, 2003. <http://www.gvsig.gva.es/>.
- [OB04] OSFIELD R., BURNS D.: *Openscenegraph*, www.openscenegraph.org, 2004.
- [osg02] Osgedit, 2002. <http://osgedit.sourceforge.net/>.
- [PSS02] P. S. STRAUSS P. I., SHRAG J.: The design and implementation of direct manipulation in 3d, 2002.
- [rcp08] Otros rcp, 2008. <http://www.eclipse.org/community/rcp.php>.
- [SSS] SALZMAN ., STACHNIAK S., STUERZLINGER W.: Unconstrained vs. constrained 3d scene manipulation.
- [SSS01] SMITH G., STÜRZLINGER W., SALZMAN T.: 3D scene manipulation with 2D devices and constraints. In *Proceedings of Graphics Interface 2001 (2001)*, Watson B., Buchanan J. W., (Eds.), pp. 135–142.
- [wor05] Worldwindrcp, 2005. <http://code.google.com/p/worldwindrcp/>.