

## Introducción

Hoy en día existen numerosas técnicas para aplicar texturas sobre objetos 3D genéricos, pero los mecanismos para su creación son, en general, o bien complejos y poco intuitivos para el artista, o bien poco eficientes en aspectos como obtener un texturado global sin costuras. Recientemente, la invención de los policubos ha abierto un nuevo espectro de posibilidades a la hora de realizar estas tareas, e incluso otras como animación y subdivisión, de crucial importancia para industrias como el cine o los videojuegos. Desafortunadamente, no existen herramientas automáticas y editables que permitan generar el modelo de policubos base.

Un policubo es una agregación de cubos idénticos de forma que cada cubo tiene como mínimo en común una cara con otro cubo. Con la agrupación de estos cubos se pueden generar diferentes figuras espaciales.

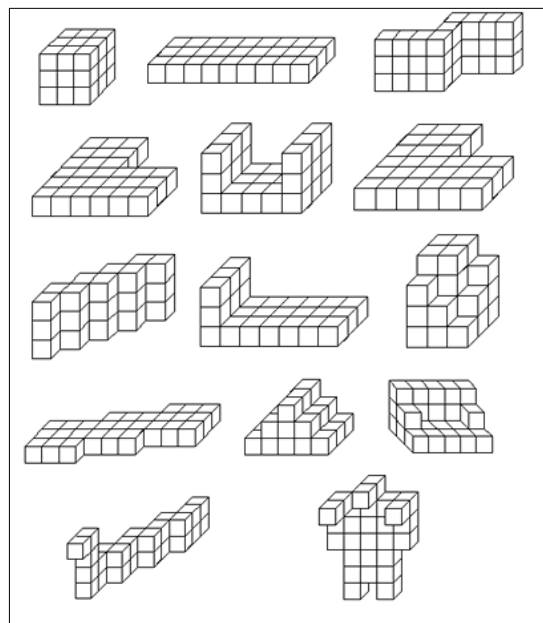


Figura 1 Ejemplos de policubos.

Como puede observarse en la figura 1, gracias a los policubos podremos generar una representación esquemática y regular de los objetos 3D, para conseguir así una fácil manipulación y procesado por parte del usuario.

## Requisitos del sistema

Los requisitos del sistema se dividen en dos grupos los funcionales y los no funcionales:

### Requisitos funcionales

Los requisitos funcionales de la herramienta serán los siguientes:

- Poder construir una estructura de policubo manualmente a partir de una interacción con el teclado y el ratón.
- Poder guardar y cargar la estructura de policubo.
- Poder borrar cubos de la estructura de policubo.
- Poder elegir la orientación de la estructura de policubo.
- Poder elegir el tamaño de los cubos.

### Requisitos no funcionales

Para la herramienta que se implementara se prestara atención a los siguientes requisitos no funcionales:

- Rendimiento, se optimizara la herramienta para que no utilice más memoria de la necesaria en la estructura de datos. Con ello se obtendrá un mejor rendimiento.
- Accesibilidad, se hará una herramienta que se pueda utilizar en cualquier ordenador actual y no sea necesario tener una estación de trabajo para poder modelar la estructura de policubo.
- Usabilidad, se diseñara una interfaz para el usuario que permita el manejo de la herramienta con más facilidad.

## Clases y métodos

Se han implementado las siguientes clases: PFCListener y PFCApplication, MatrizCubos. También se han utilizado clases ya existentes de las que se destaran las más importantes: Ogre::Camera, Ogre::SceneManager, Ogre::Entity, Ogre::SceneNode. En la siguiente Figura 2, se puede ver el diagrama de clases de la herramienta realizada.

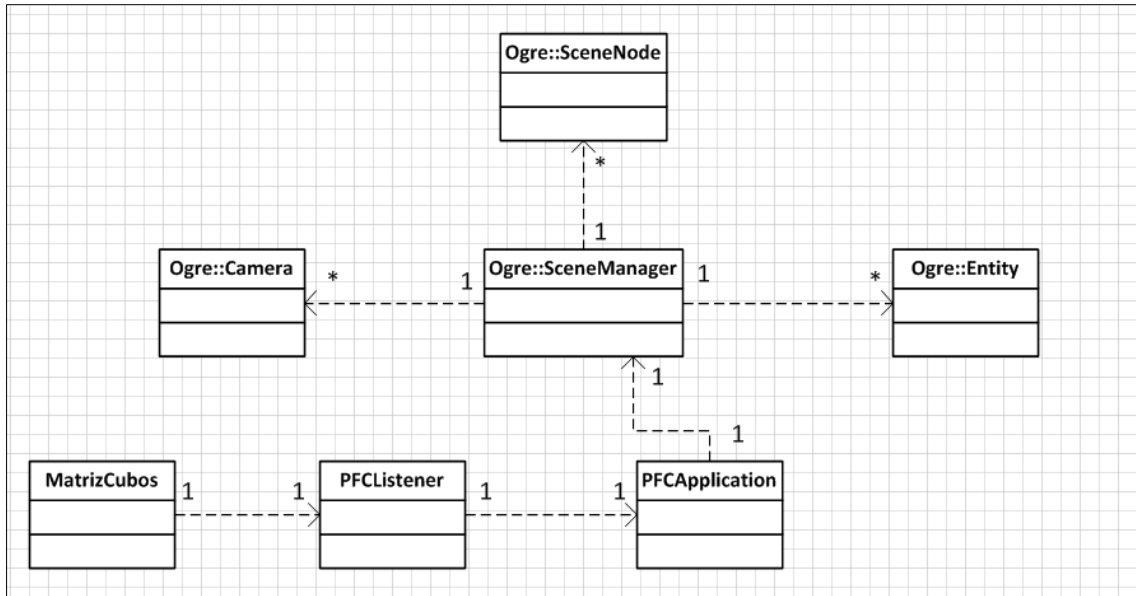


Figura 2 Diagrama de Clases.

## Resultados

Se han realizado todos los apartados planificados inicialmente. A continuación se pueden ver algunas capturas del trabajo realizado.



Figura 3 Captura de la herramienta.

En la Figura 3, se puede ver el aspecto general de la herramienta, con la mesh del ninja cargado.

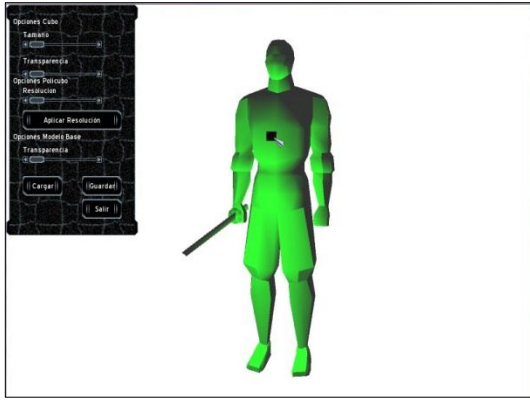


Figura 4 Captura de la inserción del primer cubo.

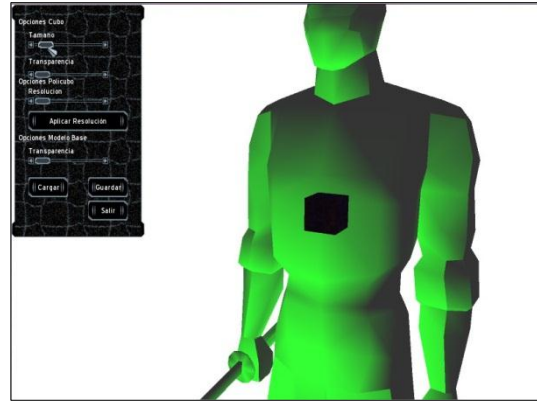


Figura 5 Captura de un cambio de tamaño al cubo inicial.

En la Figura 4, se puede ver la inserción del primer cubo. En la Figura 5, se puede ver como se cambia el tamaño del primer cubo.

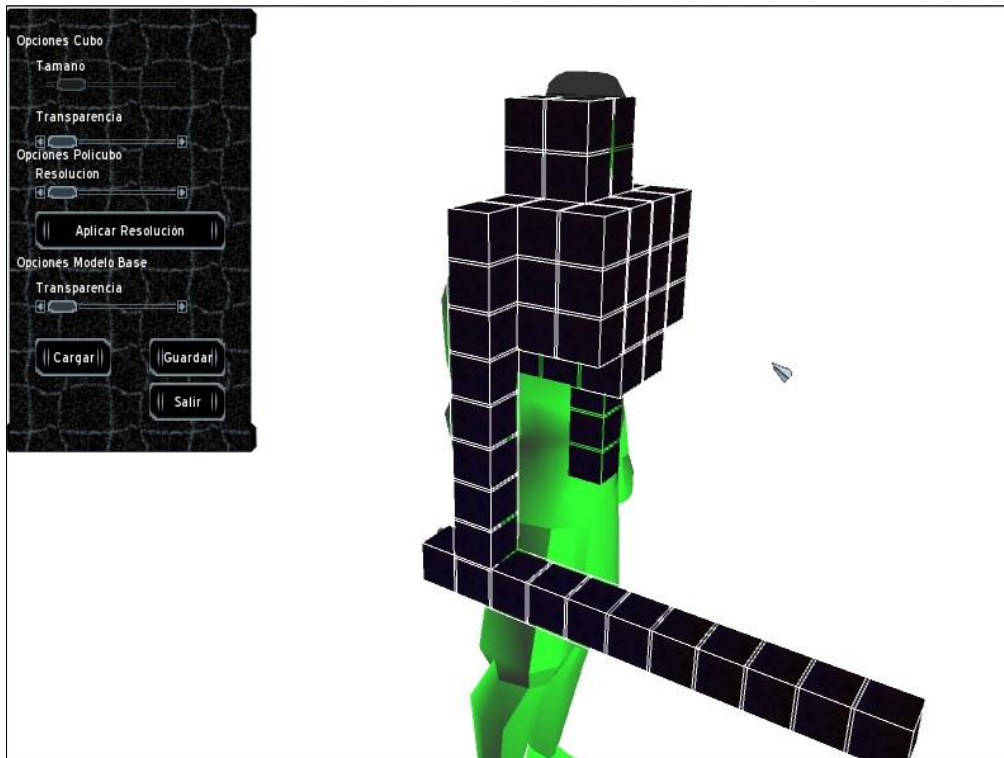


Figura 6 Captura de un ejemplo de policubo con modelo base de un ninja

En la Figura 6 se puede ver como va tomando forma el policubo gracias a la herramienta.

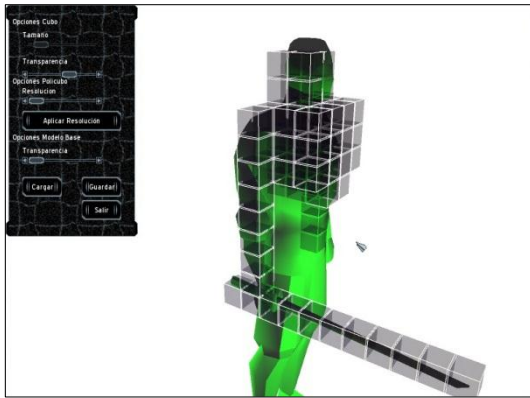


Figura 7 Captura de un cambio de transparencia al policubo

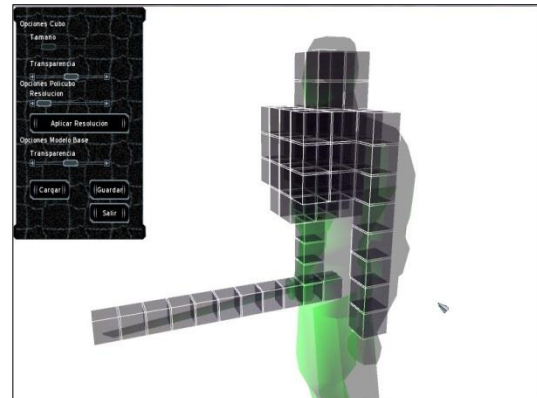


Figura 8 Captura de un cambio de transparencia al modelo base

En la Figura 7, se puede ver como se aplica transparencia al policubo. En la Figura 8, se puede observar como se le aplica transparencia al modelo base.

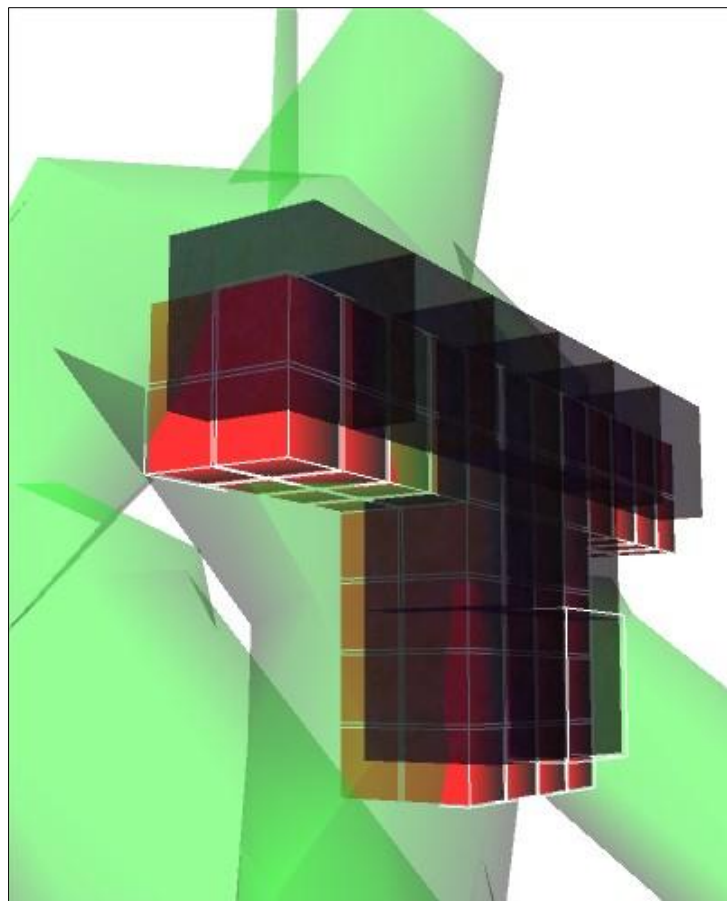


Figura 9 Captura de un cambio de resolución del policubo.

En la figura anterior se puede observar en color negro el policubo creado por el usuario y en rojo el policubo resultante creado por la herramienta al aplicar la resolución.