

Informática Gráfica II: Práctica 0 (NO entregable)

Clase de laboratorio (18/09/2024)

Descargad la plantilla para el proyecto Ogre3D que utilizamos en la clase anterior. Recordad que está disponible en el Campus Virtual. También podéis utilizar el proyecto de la clase anterior.

Localizad los tres archivos de configuración **ogre.cfg**, **resources.cfg** y **plugins.cfg**, así como el fichero de log **ogre.log**. En la clase anterior ya modificamos el fichero **resources.cfg** para mostrar el dragón.

Examina la carpeta **media** del proyecto, en concreto las subcarpetas **materials**, **models** y **packs**. Examina las tres subcarpetas de **materials**, a saber, **programs**, **scripts** y **textures**. Averigua qué contienen cada una de ellas. Localiza la carpeta comprimida **Sinbad.zip**. Comprueba también que la carpeta **IG2App** está vacía.

Para compilar, no olvidéis utilizar el modo *Release* y *x86*.

Eliminando el archivo **ogre.cfg**, haz que se muestre la caja de diálogo para configurar los parámetros de renderización del proyecto. Esto mismo se consigue también con la siguiente sentencia:

```
mRoot->showConfigDialog(OgreBites::getNativeConfigDialog());
```

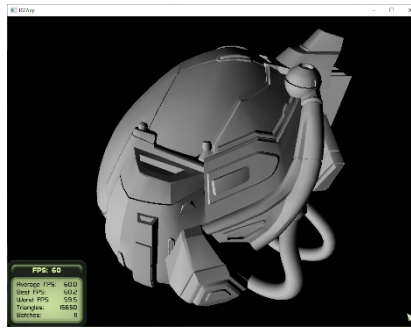
Comprueba entonces que la caja de diálogo aparece con cada ejecución del proyecto. Para saber dónde añadir la instrucción, observa que usa funcionalidad de **OgreBites** invocada por la raíz **mRoot**. Una vez que hayas hecho este apartado, pon comentario a la instrucción para que no vuelva a aparecer la caja de diálogo.

Identifica el lugar del proyecto donde se muestra **Sinbad**. Observa que **Sinbad** forma parte de la escena por lo que ha de aparecer en el setup de la escena – subprograma **setupScene** - que es donde ésta se puebla de elementos. Localiza, examinando **resources.cfg**, el lugar donde se encuentra la malla de **Sinbad** y sus materiales.

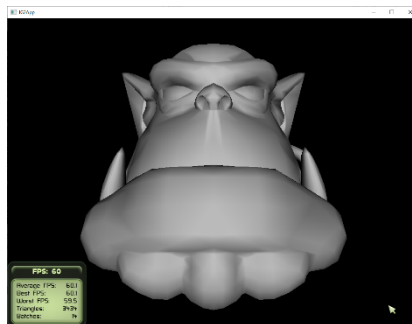
Haz que se muestre la espada de **Sinbad** en lugar de mostrar el propio **Sinbad**. Muéstrala como en la captura de la izquierda. La malla de la espada se llama **Sword.mesh**.



Haz que se muestre **DamagedHelmet.mesh** que es una especie de casco, que aparece en **Damagedhelmet.zip**, y que se muestra como en la captura adjunta. Recuerda que lo primero que tienes que hacer es averiguar si el comprimido aparece en el proyecto.



Haz que se muestre la cabeza de **ogre**, que es una malla llamada **ogrehead.mesh**. Examina otras mallas (**facial**, **fish**, **knot**, ...) que contiene el proyecto en **media/models**.

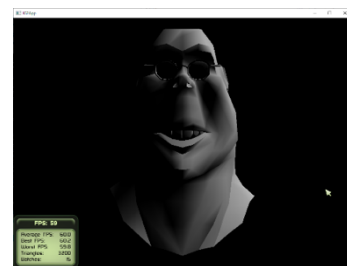
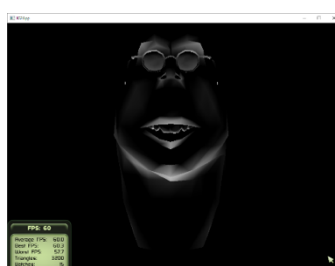


Como antes, el esqueleto ya contenía esta malla en **/models**, luego basta añadir la línea reseñada. Pero en lugar de modificar **resources.cfg** de esta manera, se puede optar por ponerlo en **IG2App** de manera que, en **resources.cfg**, en lugar de la línea anterior, se escribe la línea:

FileSystem=../Media/IG2App

La política es ir añadiendo a ese directorio todas las “cosas” que se van usando en nuestro proyecto, porque es más rápida la compilación y más segura.

Localiza la luz direccional de la escena que contiene el proyecto. Modifica sucesivamente la dirección de la luz para que **facial.mesh** se vea como en cada una de las capturas que se muestran abajo.

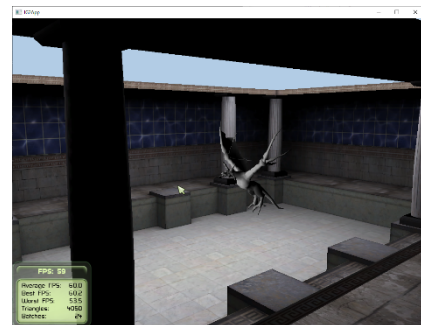


Algunos de los modelos anteriores no tienen texturas y, por ello, aparecen en blanco. Para que aparezcan con las texturas, quitad los comentarios correspondientes en el fichero **resources.cfg**, los cuales hacen referencia a los scripts que contienen los materiales. Además, podéis descargar de la última versión del SDK de Ogre modelos nuevos como **ogrehead.mesh**. Copiadlo en la carpeta **Media/IG2App** de vuestro proyecto.

Modificad la dirección de la luz para que la espada se vea como en la captura del apartado anterior.

Localiza dónde se define el color de fondo de la escena y cámbialo por **(0.7, 0.8, 0.9)**.

Crea cualquiera de las escenas adjuntas de abajo. Las mallas forman parte del pack **fresnelDemo.zip**.



Para finalizar vamos a crear el muñeco de nieve que vimos en clase de teoría. La idea es crear varias partes de forma estructurada, utilizando objetos de tipo **SceneNode**, para que podamos aplicar transformaciones a la cabeza, al cuerpo, o al muñeco entero. Tened en cuenta que nos hará falta una única malla (sphere.mesh) pero será necesario crear una entidad para cada parte.

Las siguientes capturas muestran un ejemplo del muñeco, la rotación de su cabeza, la rotación de su cuerpo y la rotación del muñeco entero, respectivamente.

