Manual de Instalación:
Ogre 3D y proyectos para Ogre 3D

Informática Gráfica II
Grado en Desarrollo de Videojuegos

Facultad de Informática
Universidad Complutense de Madrid

2025

Alberto Núñez

Índice

Instalación de Ogre 3D (Modo *Release*)

1º Instalar Visual Studio 22. Puede descargarse gratuitamente (a día de hoy, 4/sep/2025) desde la Web https://visualstudio.microsoft.com/es/downloads/

En la instalación, seleccionar Desarrollo en C++ para ordenadores de escritorio (*Desktop development with C++*)

2º Instalar el SDK con el que realizaremos las prácticas iniciales: SDK de Microsoft "Windows 10 SDK, versión 1803 (10.0.17134.12)", el cual se puede descargar desde el siguiente enlace: https://developer.microsoft.com/es-es/windows/downloads/sdk-archive/

Si no aparece en esta lista, acceder a los SDKs anteriores: https://developer.microsoft.com/es-es/windows/downloads/sdk-archive/index-legacy

En una misma instalación de Visual Studio 2022 se pueden instalar varios SDKs y elegirlos fácilmente para usarlos en distintos proyectos. Por ello, es conveniente contar con esta versión, aunque se disponga de una más actualizada. El motivo es que, con esta versión, contamos con el motor precompilado y definidas tanto las rutas a las bibliotecas como los ficheros de configuración, lo que nos permite asegurarnos de que el resto de elementos están bien configurados en el ordenador, como *drivers* y *plugins*.

3º Instalar el último SDK compatible con nuestro Sistema Operativo. En este punto, dependerá de si tenemos Windows 10 o Windows 11 instalado. En la parte en la que nos pregunte los elementos que deseamos instalar, los seleccionaremos todos.

4º Descargar Ogre de: https://www.ogre3d.org/download/sdk/sdk-ogre

- La versión en código fuente (NO la pre-compilada)
- La versión actual (estable) es la 14.3.4
- Link (4/sep/2025): https://github.com/OGRECave/ogre/archive/v14.3.4.zip

5º- Descomprimimos y cambiamos la carpeta de nombre. Tened en cuenta que aquí estarán los ficheros con el código fuente y no se modificará durante el desarrollo de la asignatura, lo cual nos permitirá, por ejemplo, recompilar utilizando otra arquitectura. Por ejemplo, nombrad a esta carpeta `ogre-14.3.4-src`

6º Creamos la carpeta donde se compilará Ogre. Aquí tendremos las bibliotecas, *plugins*, ficheros de configuración, etc. Por ejemplo, ponedle el nombre `ogre-14.3.4-build`

Es este punto tendremos dos carpetas:

- `ogre-14.3.4-build`, vacía, donde se compilará Ogre 3D
- `ogre-14.3.4-src` con el código fuente de Ogre 3D

7º Descargar e instalar (si es el caso) **cMake** de https://cmake.org/download/. Instalaremos, por ejemplo, la versión *Windows x64 Installer* (`cmake-4.1.1-windows-x86_64.msi`).

8º Ejecutamos `cmake-gui` desde Windows, ya sea escribiendo el nombre en una consola o seleccionándolo desde el menú de inicio.

En el campo ***Where is the source code*** introduce la ruta al directorio del código fuente de Ogre 3D, y en el campo ***Where to build the binaries*** introduce la ruta al directorio de compilación que creaste anteriormente. Pulsa ***Configure***, aparecerá un cuadro de diálogo que te pedirá seleccionar un generador, donde seleccionaremos Visual Studio 2022.

En este paso es IMPORTANTE. En el campo "*optional platform*" podremos incluir la plataforma para la que se generarán los ficheros compilados, por ejemplo, X64 o x86. A partir de este paso, es necesario elegir siempre la misma arquitectura para realizar una compilación correcta.

En los check-box de abajo seleccionaremos "Use default native compilers" y presionaremos ***Finish*** para que cMake obtenga información del entorno y las dependencias necesarias para la compilación.

Cuando acabe el proceso, pulsaremos ***Generate***. Esto instalará los ficheros necesarios para la compilación en el directorio donde se compilará Ogre.

9º Localizamos el fichero `Ogre.sln` (en el directorio donde se ha instalado Ogre 3D) y lo abrimos (*double-click*) con Visual Studio 2022.

Una vez abierto, seleccionamos la plataforma para la que se compilará Ogre 3D. Por defecto suele aparecer X64. Además, en este caso, seleccionaremos el modo ***`Release`***, en el combo-box de la izquierda.

En el frame derecho, seleccionamos el proyecto `ALL_BUILD` (*right-click*) y elegimos `Seleccionar como proyecto de inicio`.

Ahora ya podremos compilar. Iremos al menú `compilar` y elegimos `compilar solución`.

Una vez compilado el proyecto, seleccionamos el proyecto `INSTALL` como proyecto de inicio e instalamos Ogre 3D accediendo a `compilar-> Generar INSTALL`.

Una vez instalado correctamente, se pueden repetir los pasos para realizar también la instalación en modo ***`Debug`***. Para esto, consultar la sección Compilación de Ogre 3D (Modo debug).

Crear un proyecto Ogre 3D nuevo (modo *Release*)

1º Abrimos Visual Studio 2022

2º Creamos un proyecto nuevo, de tipo **Aplicación de Consola C++**

3º Incluimos el código con el ejemplo básico. Ejemplo de Sinbad.

4º Añadimos los ficheros necesarios, o bien los copiamos si ya los tenemos. En el caso, por ejemplo, de `SinbadExample`, copiaremos los tres ficheros en la carpeta del proyecto. Luego, para que aparezcan en el explorador de soluciones, hacemos *right-click* en el proyecto, y luego seleccionamos `Agregar -> Elemento existente`. Seguidamente seleccionamos todos los ficheros que queramos incluir.

A partir de este punto tendremos en cuenta varias consideraciones:

- Las rutas relativas hacen referencia a la instalación de Ogre 3D: `ogre-14.3.4-build`
- Elegiremos la configuración `Release`. Posteriormente podremos configurar el modo `Debug` siguiendo los mismos pasos, pero seleccionando los directorios y ficheros correspondientes a este modo.

5º Añadimos las rutas a los ficheros de cabecera necesarios. Esto se hace accediendo a las `propiedades del proyecto -> C++ -> General -> Directorios de inclusión adicionales`. En este apartado incluimos:

- `sdk\include\OGRE`
- `sdk\include\OGRE\Bites`
- `sdk\include\OGRE\Overlay`
- `sdk\include\OGRE\RTShaderSystem`
- `Dependencies\include\SDL2`

6º De forma similar, añadimos los directorios con las bibliotecas necesarias para la compilación. Para ello, accedemos a `propiedades del proyecto -> Vinculador -> General -> Directorios de bibliotecas adicionales`. En este apartado incluimos:

- `lib\Release`

7º Añadimos las referencias a las bibliotecas necesarias para la compilación. Para ello, accedemos a `propiedades del proyecto -> Vinculador -> Entrada -> Dependencias adicionales`. En este apartado incluimos:

- `OgreMain.lib`
- `OgreBites.lib`
- `OgreOverlay.lib`
- `OgreRTShaderSystem.lib`

En este punto debemos tener el proyecto compilado en modo `Release`. Sin embargo, si ejecutamos, obtendremos varios errores. Esto se debe a que el compilador no encuentra las

bibliotecas (.dll) del motor Ogre 3D necesarias. Hay varias soluciones para resolver este paso. Una de ellas es copiar los ficheros necesarios en el directorio donde se genera el ejecutable del proyecto. Este directorio depende de la arquitectura. Por ejemplo, si usamos una arquitectura x64, el directorio se llamará `x64`. Además, dentro podremos encontrar dos directorios más: `Release` y `Debug`. Por ahora nos centramos en la configuración **`Release`**.

8º Copiamos los ficheros .dll del directorio `bin/release` en `x64/Release`.

9º De forma similar, copiamos los ficheros `samples.cfg`, `resources.cfg` y `plugins.cfg` del directorio `bin/release` en `x64/Release`.

10º Editamos el fichero `resources.cfg` para configurar las rutas, deben apuntar a los subdirectorios correspondientes en el directorio de instalación de Ogre 3D. Nota: No lo hagáis manualmente, utilizad la herramienta "Reemplazar" que traen casi todos los editores de texto.

11º Ejecutamos! Si aparecen errores, generalmente se deberán a que no se han copiado los ficheros en el directorio correspondiente, o que las rutas en el fichero `resources.cfg` no son las correctas.

En este punto debería aparecer una pantalla para elegir la configuración de visualización, donde elegiremos, entre otros parámetros, el modo de renderizado. Por ahora elegiremos Open GL.

Los pasos anteriores describen la configuración de un proyecto en modo **`Release`**. Sin embargo, en algunas ocasiones podemos necesitar el depurador (modo **`Debug`**) para corregir posibles errores. Los pasos a seguir son los mismos, salvo que se deberá seleccionar el modo **`Debug`** en cada uno de los pasos anteriores.

Compilación de Ogre 3D (Modo debug)

Si intentamos compilar Ogre 3D en modo **Debug** – tras la generación de los ficheros con cMake anterior-- en ocasiones las bibliotecas SDL y zlib no se generan en esto modo, lo que puede llevar a posibles conflictos. Por ello, es deseable repetir los pasos anteriores utilizando el modo **debug**, desde que creamos la configuración con cMake hasta que compilamos Ogre 3D. Así, tendremos dos carpetas independientes, cada una con el motor Ogre3D compilado en cada modo.

Puede probarse a generar la configuración **Debug** en el mismo directorio utilizado para la versión **Release**. En algunos casos, no se genera correctamente la biblioteca `SDL2d.dll`, necesaria para ejecutar en modo depuración. De esta forma, aunque los programas se compilan correctamente, no se ejecutan.

Por ello, una alternativa para solventar esto consiste en generar el proyecto de Ogre en modo **Debug** en otra carpeta. En este apartado resumiré los cambios necesarios para compilar Ogre3D en modo **Debug**, ya que casi todos son los mismos que los llevados a cabo en el modo **Release**.

1º Creamos una carpeta donde se instalará Ogre 3D en modo **Debug**, por ejemplo, **ogre-14.3.4-debug**. Así, tendremos Ogre 3D instalado en dos carpetas:

- `ogre-14.3.4-build` -> Modo release
- `ogre-14.3.4-debug` -> Modo debug

2º Ejecutamos cMake, pero esta vez desde la consola para que se genere de forma correcta la biblioteca SDL2. Para ello:

- Abrimos una consola, pulsando `cmd.exe` en el inicio.
- Localizamos el directorio del proyecto con el código fuente de Ogre 3D, *right-click*-> **Abrir en Terminal** y ejecutamos:
    - `cmake -S . -B C:\Users\NombreUsuario\Desktop\ogre-14.3.4-debug -G "Visual Studio 17 2022" -A x64 -DCMAKE_BUILD_TYPE=Debug`

Esto generará los ficheros necesarios para compilar Ogre 3D en modo **Debug**.

3º Abrimos el proyecto, seleccionamos tanto la arquitectura **x64** como el modo **Debug** en los combo-box de Visual Studio.

4º Compilamos **ALL_BUILD**, igual que hicimos en el modo **Release**.

Crear un proyecto Ogre 3D nuevo (modo *Debug*)

Para poder ejecutar nuestro proyecto en modo **Debug**, debemos realizar algunos cambios en el mismo para que, al cambiar de modo, la compilación tenga en cuenta las bibliotecas y rutas correctas.

Primero, en cada paso de este apartado, elegiremos la Configuración **Debug** en las propiedades del proyecto. Es el combo-box situado en la parte superior izquierda en el panel de propiedades.

Tened en cuenta que, en este apartado, las rutas relativas hacen referencia a la instalación de Ogre 3D en modo **Debug**: `ogre-14.3.4-debug`

1º Añadimos las rutas a los ficheros de cabecera necesarios. Esto se hace accediendo a las **propiedades del proyecto -> C++ -> General -> Directorios de inclusión adicionales**. Estas son las mismas que en modo Release. En este apartado incluimos:

- `sdk\include\OGRE`
- `sdk\include\OGRE\Bites`
- `sdk\include\OGRE\Overlay`
- `sdk\include\OGRE\RTShaderSystem`
- `Dependencies\include\SDL2`

2º Añadimos los directorios con las bibliotecas necesarias para la compilación. Para ello, accedemos a **propiedades del proyecto -> Vinculador -> General -> Directorios de bibliotecas adicionales**. En este apartado incluimos:

- `lib\Debug`

3º Añadimos las referencias a las bibliotecas necesarias para la compilación. Para ello, accedemos a **propiedades del proyecto -> Vinculador -> Entrada -> Dependencias adicionales**. En este apartado incluimos:

- `OgreMain_d.lib`
- `OgreBites_d.lib`
- `OgreOverlay_d.lib`
- `OgreRTShaderSystem_d.lib`

4º Copiamos los .dll del directorio `bin/debug` en `x64/Debug`.

5º De forma similar, copiamos los ficheros `samples.cfg`, `resources.cfg` y `plugins.cfg` del directorio `x64/Release` en `x64/Debug`. Estos ficheros serán los mismos para ambas configuraciones.

Proyecto de Ogre 3D en los laboratorios de la FdI

En los laboratorios donde se imparte la clase de IG2, se ha instalado la última versión estable de Ogre 3D, tanto en modo *release*, como en modo *debug*. Las rutas a estas versiones son, respectivamente:

- `C:\software\programacion\ogre-14.3.4-x64-release`
- `C:\software\programacion\ogre-14.3.4-x64-debug`

El fichero `resources.cfg` contiene las rutas para localizar los recursos. En particular, es recomendable que todos los recursos utilizados en el proyecto – que no estén en la base de Ogre 3D – se copien en la carpeta `IG2Media`, tales como scripts, materiales y mallas adicionales, shaders, etc.

Para entregar la práctica y el examen se utilizará este proyecto. Es recomendable que la práctica la probéis en los laboratorios antes de realizar la entrega.

Los pasos para realizar la entrega son:

1.- Descomprimid en fichero `IG2Project.zip` en `c:\hlocal` (si lo hacéis en `c:\hlocal\IG2Project`, creará otra carpeta adicional)

2.- Copiad vuestro código fuente y los ficheros de los laberintos en `IG2Project\IG2Project`, es decir, en la misma carpeta que el fichero `IG2Project.vcxproj`. Eliminad el código de la plantilla: `IG2Project.cpp`, `IG2Project.h` y `Main.cpp`.

3.- Copiad el fichero de materiales en `IG2Project\IG2Project\IG2Media`. Únicamente si utilizáis ficheros adicionales (mallas, texturas, etc) que no estén en la base de Ogre, deberéis incluirlos también en este directorio. Por ejemplo, la malla `ogrehead.mesh` no hace falta copiarla.

4.- Copiad los `.dll` correspondientes en la carpeta `IG2Project\x64\Release` y/o `IG2Project\x64\Release`. Podéis encontrar estos ficheros en `c:\software\programacion\ogre-14.3.4-x64-debug\bin` y `c:\software\programacion\ogre-14.3.4-x64-release\bin`

5.- En este punto, ya podéis compilar y ejecutar la práctica para probarla. Si todo funciona correctamente, **cerrad Visual Studio**, ejecutad el script `clean.bat` (el cual elimina los binarios como .dll, .obj, .exe, etc..) y comprimid el directorio principal del proyecto en un fichero .zip.

Atentamente,
Alberto N.

Fichero de Configuración para el renderizado

Ogre 3D almacena la configuración seleccionada en el diálogo que aparece antes de la ejecución del proyecto, en un fichero llamado `ogre.cfg`. Además, tras cada ejecución, se almacena toda la información relativa a la carga de *plugins*, mallas y otros recursos en el fichero `ogre.log`.

Estos ficheros se almacenan en el Directorio `Documents` (o `Documentos`) ubicado en la cuenta de usuario. El directorio se determina por la siguiente línea del código fuente:

```
OgreBites::ApplicationContext ctx("NombreDirectorio");
```

Si queremos que el diálogo para configurar el renderizado vuelva a aparecer, deberemos borrar el fichero `ogre.cfg`.

Configuración de los recursos en la versión compilada de Ogre 3D

Los recursos están formados, entre otros ficheros, de scripts, texturas y shaders.

Estos ficheros pueden ubicarse en cualquier directorio, aunque es recomendable tenerlos dentro del directorio del proyecto. Así resulta más sencillo gestionar tanto código como recursos.

Por ejemplo, podemos crear un directorio llamado `IG2media`, dentro del proyecto.

Para que nuestro proyecto localice estos ficheros, debemos incluir una línea en el fichero `resources.cfg`.

```
FileSystem=/rutaAlProyectoOgre/IG2media
```

En esta versión, el fichero `resources.cfg` suele estar ubicado en el directorio que contiene el ejecutable del proyecto (con los `.dll` y el resto de los `.cfg`)

Configuración de los recursos en la versión pre-compilada (ver. 11) de Ogre 3D

Los recursos están formados, entre otros ficheros, de scripts, texturas y shaders.

En la versión pre-compilada, hay definida una carpeta que contiene los recursos que se utilizarán en los proyectos. La siguiente línea en el fichero `resources.cfg` lo indica:

```
FileSystem=/rutaAlDirectorioOgre/media/IG2App
```

Si queréis cambiar esta carpeta, basta con modificar la línea en el fichero `resources.cfg` para que haga referencia a la nueva ruta.

El fichero `resources.cfg` está localizado en la carpeta `bin`, relativa al directorio que contiene esta versión de Ogre 3D.