

Tutoriel : Installation d'OpenSceneGraph

par Vincent Bourdier ([Ange3d's world](#)) ([Blog](#))

Date de publication : 19 juin 2008


Dernière mise à jour : 19 juin 2008

Ce tutoriel vous présente OSG et son installation sous Windows.
En détail sont expliquées deux démarches pour l'installation avec Visual Studio 2005.

I - Introduction.....	3
II - Installation des sources pré-compilées sous Visual Studio.....	3
II-A - Téléchargement.....	4
II-B - Tests.....	4
II-C - Compilation d'un exemple.....	4
III - Compilation des sources.....	5
III-A - Téléchargement.....	5
III-B - Configuration.....	6
III-C - Compilation.....	6
III-D - Résultats.....	7
IV - Conclusion.....	7

I - Introduction

OpenSceneGraph (que nous appellerons plus communément OSG) est une immense boîte à outils multi-plateformes pour la création d'environnements 3D temps-réel. C'est un projet Open Source, toujours en développement. Les applications possibles sont presque illimitées... D'une manière plus pratique, OSG est une sorte de couche supplémentaire qu'on utilise pour ne pas avoir à manipuler les commandes OpenGL. Car bien sûr c'est de l'OpenGL par en dessous...

Le principe de base repose sur les **Scene Graph**. Pour ceux qui auraient déjà vu un peu de  **Java3d**, le principe est très proche, à savoir un arbre/graph qui est ordonné et bien structuré, et qui permet de gérer un environnement graphique 3d évolué. Pour exemple le viewer de base d'OSG permet déjà les interactions clavier-souris sur l'environnement 3D.

OSG gère en entrée des tas de format de fichiers 3d, en passant par le X3D/VRML (dont le format .osg est un dérivé compatible) mais aussi le flt, ... et bien sûr le .3ds

Que dire de plus pour introduire cela ?

Peut être quelques infos sur l'utilisation et l'installation...

OSG se décompose en une masse de fichiers source bien sûr, mais vous pouvez télécharger directement les fichiers dans une jolie installation Windows, contenant les sources précompilées sous Visual Studio.

Ensuite, libre à vous de commencer à coder... pas de limites, OSG gère les multi-vues, caméra multiples et mobiles, shaders (GLSL), textures multiples, couches d'opacité, bump mapping, transformations, déplacement, rotation, picking, effet spéciaux, particules, ombres, ... la liste sera trop longue pour arriver au bout en un temps raisonnable. D'une manière générale, comme OSG "surcharge" OpenGL, tout ce qu'OpenGL permet, OSG l'offre sur un plateau, et même plus.

L'utilisation sous windows est simple, après l'installation il suffit de régler quelques variables d'environnement pour être tranquille, et d'ajouter les libs dans les paramètres du linker. Nous en reparlerons pour l'installation.

OSG est en anglais bien sûr, et comme tout projet assez neuf et en constante évolution, le moyen par défaut d'aide et support sont : le wiki, les tutoriels (rares et pas forcément à jour mais très bien fait), et la mailing liste (et non pas de forum officiel...).

Et puis après tout, ça fait du bien de rédiger en english de temps en temps...

Sauf erreur de ma part, OSG a été porté sous Java, mais ça c'est une autre histoire que je ne peux pas vous raconter...

Bien sûr, libre à vous de découvrir tout cela sur le site officiel :

 openscenegraph.org

et aussi une véritable introduction à OSG :

[introduction a OSG](#)

Nous allons à présent attaquer la partie plus pratique, à savoir l'installation d'OSG sur votre système, pour pouvoir coder ensuite vos applications selon vos envies et besoins.

Je vais vous exposer ici deux méthodes pour le faire :

- [l'installation des lib nécessaires](#)
- [la compilation des sources d'OSG sur votre système](#)

Personnellement, pour coder votre projet OSG, je vous conseille la compilation des sources. Vous aurez la certitude que les libs sont adaptées à votre système, et vous aurez les sources des exemples sous la main, ce qui peut s'avérer tout à fait nécessaire parfois.

A vous de choisir la méthode de votre choix.

II - Installation des sources pré-compilées sous Visual Studio


Ce tutoriel détaille l'installation d'OSG sous Windows dans un environnement **Visual Studio (express 2005)**. Si votre configuration est différente, libre à vous de trouver dans ce tutoriel les informations que vous pourrez.

Cette méthode consiste à télécharger les sources, les installer (copie des libs) et à paramétrer un tout petit peu le PC pour que tout ça fonctionne comme sur des roulettes...

II-A - Téléchargement

Téléchargement des sources : Vous pouvez les trouver sur la page **Downloads** dans "Binaries" sous le nom "win32 Binaries Installer". (environ 20 Mo)

Ceci fait, lancez le fichier téléchargé et installez donc OSG... là où bon vous semble.

 **Remarque importante** : lors de l'installation on vous propose de définir des variables d'environnement... je vous conseille d'accepter, ça fera des détails en moins à régler.

II-B - Tests

Une fois installé, vous avez deux raccourcis sur le bureau : osgLogo et osgShell.

Si vous lancez osgLogo, le viewer d'OSG va lancer la démo de base.

Navigation :

- 1 La touche espace réinitialise la vue
- 2 le clic gauche permet de pivoter la caméra autour de la scène (restez appuyez et bougez la souris)
- 3 le clic droit permet de zoomer
- 4 le clic central permet de déplacer la vue horizontalement et verticalement.
- 5 la touche ESCAPE permet de quitter la scène.
- 6 ...

Le raccourci osgShell vous permet d'accéder à une console pour pouvoir lancer manuellement des exemples ou les programmes d'OSG.

En effet si vous allez voir dans votre répertoire d'installation d'OSG/bin vous allez voir une ribambelle de programmes... ce sont tous des exemples compilés d'OSG qui vous permettent d'avoir une idée des possibilités avec OSG.

Bien entendu les .dll sont nécessaires, et c'est pourquoi si vous voulez lancer un exe d'OSG ailleurs que dans votre répertoire d'installation, vous allez juste devoir ajouter le répertoire /bin de votre installation dans la variable d'environnement PATH. (En effet windows cherche les dll d'abord dans le répertoire courant, et ensuite dans le PATH)

II-C - Compilation d'un exemple

Nous allons à présent tester l'installation en compilant un exemple.

Créez un nouveau projet Visual C++ (vide de préférence)

Configurons le projet :

Dans les propriétés projet,

C++ :

dans Général : les répertoire d'inclure ajoutez : "C:\Program Files\OpenSceneGraph\include" (ou l'équivalent sur votre PC)

dans Génération de code : vérifiez que la Bibliothèque Runtime est bien en DLL de Débogage multithread

Linker :

Général : Répertoire de bibliothèques supplémentaire, ajoutez : "C:\Program Files\OpenSceneGraph\lib" (ou l'équivalent sur votre PC)

Entrée : ajoutez les .lib que votre application OSG va demander... pour le moment nous allons mettre osg.lib, osgDB.lib et osgViewer.lib

Validez le tout.

Nous sommes prêts à compiler, mais toujours pas de code à compiler... alors nous allons juste en coller un pour tester.

Dans le main(), collez le code suivant :

```
osgViewer::Viewer viewer;  
viewer.setSceneData(osgDB::readNodeFile("cessna.osg"));
```

```
viewer.run();
```

N'oubliez pas les inclusions :

```
#include <osgViewer/Viewer>  
#include <osgDB/ReadFile>
```

Compilez en mode Release... et c'est parti ^^

Si vous avez des erreurs de dll manquantes c'est que votre PATH n'est pas correct...

Si vous avez d'autres problèmes, n'hésitez pas à m'en faire part sur le **forum**.


III - Compilation des sources

La compilation des sources d'OSG est conseillée. Cela vous permet d'avoir les sources, de les recompiler, de recompiler et modifier les exemples, etc... Aussi, vous avez la certitude qu'OSG sera compatible avec votre système. Nous allons procéder selon les étapes suivantes :

- 1 Téléchargement des sources et des données nécessaires
- 2 Configuration
- 3 Compilation
- 4 Résultats

III-A - Téléchargement

Dans un 1er temps nous allons télécharger les données nécessaires : les sources, les data, et les 3rdParty

 *En date du 8 juin 2008 :*

OSG 2.4.0 : dernière version stable

OSG 2.5.1 : dernière développeur release

Vous devez choisir une des deux versions proposées... ou plus récente selon la date à laquelle vous suivrez ce tutoriel. Attention, Cmake n'est pas le même à chaque fois, utilisez les bons liens !

OSG 2.4 :

Les sources : [Source](http://www.openscenegraph.org/files/OpenSceneGraph-2.4.0/) <http://www.openscenegraph.org/files/OpenSceneGraph-2.4.0/>
OpenSceneGraph-2.4.0.zip

Data : <http://www.openscenegraph.org/files/OpenSceneGraph-2.4.0/OpenSceneGraph-Data-2.4.0.zip>

3rdParty : disponibles uniquement via SVN sur <https://osgtoy.svn.sourceforge.net/svnroot/osgtoy/3rdParty/branches/>

Ensuite pour compiler tout ça nous avons besoin de :

CMake 2.4 : <http://www.cmake.org/files/v2.4/cmake-2.4.8-win32-x86.exe>

Visual Studio SP1 installé (pour la version 2005 de VS).

En effet, les 3rdParty sont compilées avec le SP1, donc pour éviter les conflits, installez le SP1 de Visual Studio.

OSG 2.5.1 :

Les sources : [Source](http://www.openscenegraph.org/downloads/developer_releases/) http://www.openscenegraph.org/downloads/developer_releases/
OpenSceneGraph-2.5.1.zip

Data : <http://www.openscenegraph.org/files/OpenSceneGraph-2.4.0/OpenSceneGraph-Data-2.4.0.zip>


3rdParty : disponibles uniquement via SVN sur <https://osgtoy.svn.sourceforge.net/svnroot/osgtoy/3rdParty/branches/>

Ensuite pour compiler tout ça nous avons besoin de :

CMake 2.6 : <http://www.cmake.org/files/v2.6/cmake-2.6.0-win32-x86.exe>

Visual Studio SP1 installé (pour la version 2005 de VS).


En effet, les 3rdParty sont compilées avec le SP1, donc pour éviter les conflits, installez le SP1 de Visual Studio.

 *Explications : les Data sont les fichiers nécessaires aux exemples : textures, modeles, ... 3rdParty contient les lib externes : JPEG, GIF, PNG, freetype, ... Cmake permet de générer un fichier solution pour votre version de visual Studio (ou autre) qui vous permettra de compiler les sources ensuite.*

III-B - Configuration

Ensuite, nous allons organiser un peu la structure des fichiers pour ne pas s'y perdre.

Décompressez les sources, data et mettez les 3rdParty dans un meme répertoire.

 *les 3rdParty contiennent 2 sous répertoires... choisissez celui de votre configuration et renommez le en 3rdParty*


Le résultat devrait etre semblable à ceci :

Ensuite, nous installons **Cmake**.

Nous allons configurer Cmake avec nos sources OSG, et ce dernier va générer des fichiers projet pour compiler ensuite les sources.

Lancez Cmake.

Sélectionnez vos répertoires Source et destination

 *Le répertoire de destination est celui qui contiendra ensuite vos sources compilées.*

Cliquez ensuite sur "Configure" et sélectionnez votre environnement de développement.

Cmake va chercher automatiquement les répertoires et fichier dont il a besoin.

Dans un premier temps, commencez par vérifier/modifier le chemin de 3rdParty dans le champ "ACTUAL_3RDPARTY_DIR" en tete de liste.

Ensuite configurez à nouveau.

Vous pouvez ensuite selectionner la compilation des exemples, des plugins, etc... dans les quelques premiers champs.

Une fois terminé, vérifiez que les champs suivants sont bien renseignés automatiquement :

FREETYPE_LIBRARY_DEBUG

GIFLIB_LIBRARY_DEBUG

JPEG_LIBRARY_DEBUG

PNG_LIBRARY_DEBUG

TIFF_LIBRARY_DEBUG

Validez avec Configure encore une fois.

Cmake vous met en valeur le champ BUILT MFC EXAMPLE si vous avez selectionné la compilation des exemples.

Faites votre choix puis configurez à nouveau.

Si tout est correct, le bouton "ok" est a présent disponible.

Cliquez sur OK. Cmake va générer les fichiers projet et quitter lorsqu'il aura terminé.

III-C - Compilation

Dans votre dossier Binaries il y a a présent des fichier projet et tout ce qu'il faut pour la compilation.

Ouvrez le fichier All_BUILT.vcproj

Dans la liste des projets OSG, sélectionnez ALL_BUILT et lancez la génération. Choisissez le mode **Debug** ou **Release**, ou faites les deux tour à tour selon vos besoins.

Pendant la compilations, les fichiers .dll seront générés dans votre répertoire de sortie /bin

La compilation peut prendre un certain temps, selon votre machine, et ce que vous avez inclu ou non dans le Cmake.

Après la compilation, nous pouvons ajouter les chemins suivants dans le PATH de votre système. (Variable d'environnement)

En effet, sans cela, l'exécution ne pourra pas se produire, car les .dll nécessaires ne seront pas trouvés par le système. Ajoutez donc les chemins suivants : (adaptez selon votre installation)

```
... \OpenSceneGraph\2.4\Binaries\bin  
... \OpenSceneGraph\2.4\Binaries\bin\osgPlugins-2.4.0  
... \OpenSceneGraph\2.4\3rdParty\bin
```

III-D - Résultats

Nous pouvons à présent créer un nouveau projet que nous allons configurer ainsi :

C/C++ :

Include : D:\codage\OpenSceneGraph\2.4\OpenSceneGraph-src\include

Editeur de lien :

Général : Répertoire de dépendances supplémentaire : D:\codage\OpenSceneGraph\2.4\Binaries\lib

Entrée : osg.lib osgGA.lib osgDB.lib osgViewer.lib osgText.lib osgUtil.lib OpenThreads.lib

(note : pour certaines applications vous pourrez oter celles que vous n'utilisez pas, ou en rajouter bien sur)

Vous pouvez à présent créer votre projet OSG en toute liberté !!

IV - Conclusion

Et voilà, vous avez à présent tout ce qu'il vous faut pour créer vos applications OSG.

Bon courage.

Pour les questions merci d'utiliser le [forum](#).

Si vous en avez terminé avec l'installation, mais n'avez pas les bases d'OSG, je vous conseille [les tutos de cheveche4](#).

Merci à [loka](#) et [raptor70](#) pour leur relecture :-).