

# Réalisation d'un moteur 3D en C++, introduction



par [Laurent Gomila](#)

Date de publication : 16/01/2005

Dernière mise à jour : 15/09/2005

Cette petite introduction a pour but de vous présenter tout ce dont vous aurez besoin pour aborder cette série de tutoriels.

- 1 - Pourquoi cette série d'articles ?
- 2 - A qui s'adresse-t-elle ?
- 3 - Quelques mots sur le moteur
- 4 - Ce qui accompagne les articles
- 5 - Conclusion
- 6 - Téléchargements

## 1 - Pourquoi cette série d'articles ?

Les articles et tutoriels concernant la programmation graphique sont assez répandus, sur des sites spécialisés comme [Flipcode](#) ou [Gamedev](#) par exemple. On peut y trouver tout ce qu'il faut pour l'élaboration d'un moteur 3D : conseils, techniques en tout genre, API graphiques, ... Cependant, malgré cette mine d'informations pour nous développeurs en quête de savoir, il reste à mon avis de gros manques :

- Les articles sont souvent focalisés sur une technique précise, rares sont ceux qui traitent de la réalisation d'un projet de A à Z.
- Le C++ est la plupart du temps utilisé, mais beaucoup laissent de côté l'aspect objet ainsi que la puissance de ce langage.
- Il existe peu (voire pas) d'équivalent français à toute cette documentation.

C'est dans cette optique que je démarre cette série d'articles : proposer un tutoriel couvrant complètement la réalisation d'un moteur, en détaillant les choix de conception, les erreurs rencontrées, les techniques utilisées tant au niveau du C++ que des APIs graphiques, et en livrant bien sûr le code source qui va avec.

## 2 - A qui s'adresse-t-elle ?

L'un des buts de cette série est d'initier à des techniques utiles, pratiques ou peu connues, que ce soit au niveau de la 3D, du C++ ou plus généralement de la conception d'applications de ce type. Ces tutoriels s'adressent donc non seulement à celles et ceux que la programmation 3D intéresse, mais également à un public plus large de programmeurs C++.

Cette série s'adresse aux personnes ayant déjà acquis les bases élémentaires du C++ et de la programmation 3D. Ceux qui n'ont jamais entendu parler de templates pourront trouver tout le nécessaire dans les nombreux cours et documents présents sur internet, tout comme ceux qui n'ont jamais abordé une quelconque API graphique. Si vous savez écrire des applications 3D et voulez passer à la vitesse supérieure, si vous cherchez des techniques et concepts pour améliorer votre code, si vous butez sur des problèmes de conception, si vous êtes simplement curieux et voulez enrichir votre panel de techniques de coding, si vous ne savez pas comment aborder la réalisation d'un tel projet, ou bien sûr si vous souhaitez écrire votre propre moteur 3D... alors ces tutoriels devraient vous intéresser. Ce sont en tout cas les buts fixés, et l'état d'esprit dans lequel est rédigée cette série.

### 3 - Quelques mots sur le moteur

Cette série d'articles se base sur un projet concret, l'élaboration de mon moteur 3D (baptisé le Yes::Engine). Le moteur en lui-même est prévu pour être multi-API (implémentation DirectX et OpenGL). Il tournera sur les configurations récentes tout comme sur les plus modestes, en exploitant au maximum les possibilités du hardware. Son codage en C++ est fortement orienté objet, utilise des techniques évoluées du langage, de nombreux design patterns, et est totalement standard (sauf bien sûr la partie fenêtrage et événements, qui utilise pour le moment l'API Win32). Enfin sa distribution sera open-source, son code sous license GPL sera inclus au fur et à mesure avec les articles correspondants.

## 4 - Ce qui accompagne les articles

Deux fichiers seront systématiquement joints aux différentes parties de cette série : la version PDF de celles-ci ainsi que le code source associé.

Le code source évoluera au fil des tutoriels : vous y trouverez notamment l'implémentation complète de ce qui est décrit dans l'article, puis à partir du tutoriel n°3 une version fonctionnelle et à jour du moteur, qui s'enrichira au fil de l'évolution de la série, ainsi qu'une petite démo illustrant les concepts vus. Le tout est développé sous Visual Studio.NET 2003, mais compile sans erreur ni avertissement sous Dev-C++ 4.9.9.1. Les versions Dev-C++ et VS.NET des fichiers projets et des binaires sont également inclus au code source accompagnant chaque article. Les fichiers projets pour Code::Blocks sont également de la partie à partir du 8ème tutoriel. De cette manière vous devriez vous retrouver avec une version fonctionnelle du moteur ainsi qu'une demo complète, le tout compilant sous votre environnement de programmation préféré.

Le moteur ne compile pour le moment pas avec Borland C++ Builder 6, celui-ci ayant quelques difficultés à gérer les templates trop imbriqués.

La documentation de référence du moteur, générée à l'aide de [DOxygen](#), est également disponible à partir du tutoriel n°3.

## 5 - Conclusion

Bonne lecture !

## 6 - Téléchargements

Une version PDF de cette introduction est disponible : [Télécharger \(37 Ko\)](#)