

Les tutoriels Nehe



par Jeff Molofee (NeHe) ([Autres Articles](#))

Date de publication : 23/03/2007

Dernière mise à jour : 23/03/2007

Les tutoriels Nehe sont la base et la référence de la programmation OpenGL. Nous proposons ici une traduction française de ces tutoriels afin de faciliter leur lecture. Sur cette page, nous proposons les premiers tutoriels de cette série. Les 10 premiers articles sont traduits et les derniers attendent des traducteurs...

I - Les tutoriels

I-0 - Contributions

I-1 - Introduction

I-2 - Premier triangle et premier quadrilatère

I-3 - Coloration

I-4 - Rotations

I-5 - Première pyramide et premier cube

I-6 - Les textures

I-7 - Les filtres et l'éclairage

I-8 - Transparences

I-9 - Déplacer des bitmaps dans un monde 3D

I-10 - Chargement et déplacement dans un monde 3D

II - Contributions

II.1 - Remerciements

I - Les tutoriels

I-0 - Contributions

Remarque : la traduction des tutoriels Nehe n'est pas encore terminée donc, si vous avez l'âme de traducteur ou si vous voyez des erreurs dans ces pages, n'hésitez pas à m'envoyer un petit **message privé** ou un mail à **nehe arobase redaction-developpez.com**.

I-1 - Introduction



L'ouverture d'une fenêtre

Dans ce tutoriel, je vais vous apprendre à mettre en place et utiliser OpenGL dans un environnement Windows. Le programme que vous allez créer dans ce tutoriel va ouvrir une fenêtre vide OpenGL, passer de plein écran à un mode fenêtré et attendre que l'utilisateur appuie sur la touche Echap ou sur la fermeture de la fenêtre. Si cela ne semble pas être grand chose, ce programme sera le point de départ de tous les autres tutoriels.

C'est important de savoir comment OpenGL fonctionne, quelles sont les étapes dans la création d'une fenêtre OpenGL et comment écrire un code facile à la compréhension. Vous pouvez télécharger le code source à la fin de ce tutoriel mais je vous recommande de le lire au moins une fois avant de commencer la programmation OpenGL.

Voici le lien vers ce tutoriel : [Introduction](#).

I-2 - Premier triangle et premier quadrilatère



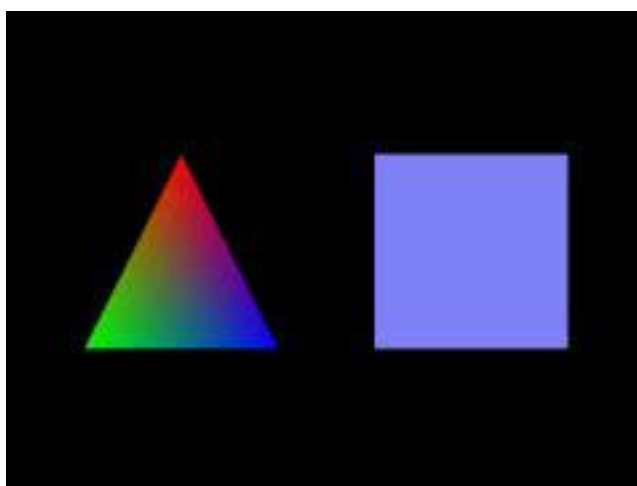
Premier triangle et premier quadrilatère

Utilisant le code du premier tutoriel, nous allons ajouter du code pour créer un triangle et un carré dans la fenêtre. Ce n'est pas encore un moteur 3D mais toute création OpenGL passe par ces deux primitives. La compréhension du code de rendu de ces deux formes géométriques en trois dimensions est donc primordiale. Suivez donc ce tutoriel avec intérêt !

Nous verrons comment fonctionnent les trois axes X, Y et Z, mais aussi comment faire des translations en 3D. A la fin, vous devrez être capable de placer un objet où vous le voulez.

Voici le lien vers ce tutoriel : [Premier triangle et premier quadrilatère](#).

I-3 - Coloration

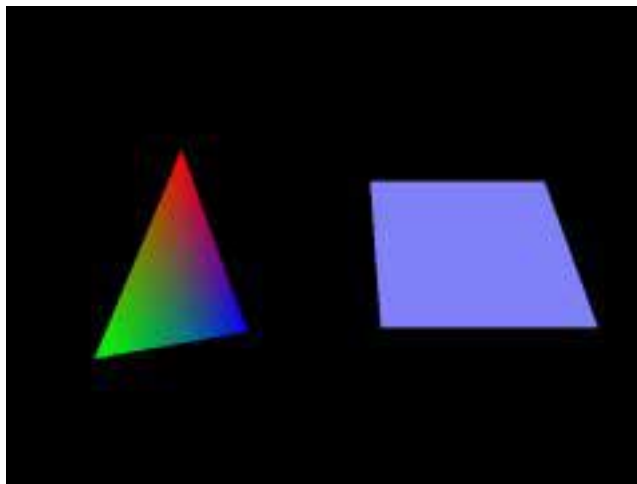


Coloration

En continuant le tutoriel précédent, nous allons voir comment ajouter de la couleur aux polygones. Nous verrons deux sortes de colorations. La coloration unie et le mélange de couleur. Si on regarde l'image, vous pouvez voir que le triangle mélange les couleurs par rapport au carré qui possède une couleur unie. Ajouter de la couleur permet vraiment de rendre les polygones plus intéressants.

Voici le lien vers ce tutoriel : [Coloration](#).

I-4 - Rotations

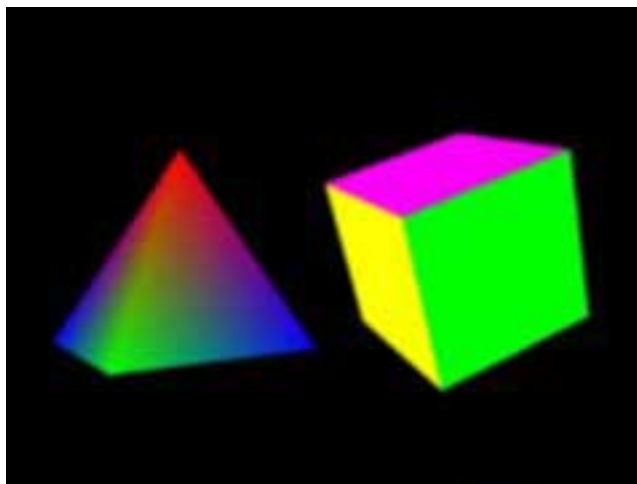


Rotations

Ce quatrième tutoriel va présenter les rotations en 3D. Nous allons faire pivoter un triangle selon l'axe des Y et un quadrilatère selon l'axe des X. C'est facile de dessiner des polygones basiques à l'écran mais les faire bouger rend tout le programme plus intéressant.

Voici le lien vers ce tutoriel : [Rotations](#).

I-5 - Première pyramide et premier cube



Première pyramide et premier cube

Maintenant que nous savons initialiser une fenêtre, dessiner des triangles et des carrés, les faire tourner sur eux-mêmes, il est temps de les transformer en objets 3D.

Nous allons donc transformer le triangle en pyramide et le carré en cube. Ce dernier aura une couleur unie pour chaque face et la pyramide utilisera un mélange de couleurs.

Voici le lien vers ce tutoriel : [Première pyramide et premier cube](#).

I-6 - Les textures



Les textures

Dans ce tutoriel, nous présentons comment utiliser des textures pour rendre les polygones nettement plus intéressants. Nous allons reprendre le code du premier tutoriel puisqu'il est plus facile de partir d'un code qui ne fait qu'ouvrir une fenêtre vide qu'un code qui fait tourner différents objets 3D.

Voici le lien vers ce tutoriel : [Les textures](#).

I-7 - Les filtres et l'éclairage

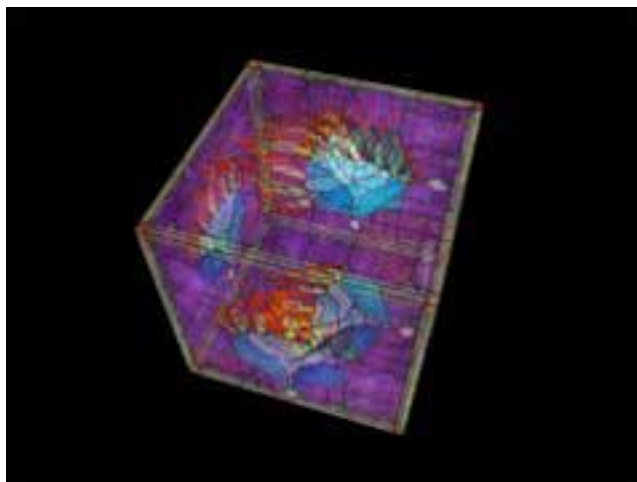


Les filtres et l'éclairage

Ce tutoriel est assez important et donc assez long ! Si vous ne vous sentez pas à l'aise avec les six premiers, relisez les avant de continuer ! Nous allons voir comment utiliser des filtres sur les textures, comment utiliser le clavier et mettre en place des lumières !

Voici le lien vers ce tutoriel : [Les filtres et l'éclairage](#).

I-8 - Transparences



La transparence

Ce tutoriel continue le précédent en ajoutant une autre partie importante d'un code OpenGL : la transparence. Cela nous permet de dessiner des vitres, de l'eau et bien d'autres éléments.

Voici le lien vers ce tutoriel : [Transparence](#).

I-9 - Déplacer des bitmaps dans un monde 3D

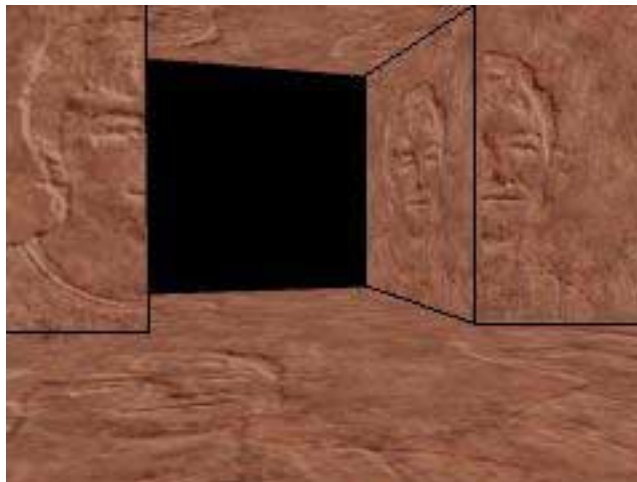


Déplacer des bitmaps dans un monde 3D

Une fois que nous savons afficher un cube transparent pivotant, il serait intéressant de déplacer des images BMP dans un univers 3D en enlevant des parties des images qui seront considérées comme transparentes. Ce tutoriel fait cela en reprenant toutes les idées des tutoriels précédents. Soyez sûr de tout comprendre avant de continuer car le niveau de celui-ci est un peu plus élevé.

Voici le lien vers ce tutoriel : [Déplacer des bitmaps dans un monde 3D](#).

I-10 - Chargement et déplacement dans un monde 3D



Chargement et déplacement dans un monde 3D

Voici un tutoriel qui présente un code intéressant : charger un fichier de données et représenter un petit monde en trois dimensions. De plus, il permet de se déplacer dans ce monde avec une idée intéressante pour faire croire qu'on se balade vraiment à l'intérieur, je vous laisse découvrir.

Voici le lien vers ce tutoriel : [Chargement et déplacement dans un monde 3D](#).

II - Contributions

La traduction des autres tutoriels est en cours mais votre aide serait appréciée ! Si vous voulez aider, n'hésitez pas à envoyer un **message privé** ou un mail vers **nehe arrobasede redaction-developpez point com**.

II.1 - Remerciements

Merci à toute l'équipe de Developpez.com qui a participé et participe encore à cette traduction !

