

Qt OpenGL - Annexes

Sommaire

- Introduction à OpenGL et Qt 5.4
- Support d'OpenGL dans Qt
- Qt OpenGL - Générer un terrain
- Qt OpenGL - Envoyer des données au processeur graphique
- Qt OpenGL - Utilisation du pipeline programmable
- Qt OpenGL - Ajouter des lumières et des textures
- Qt OpenGL - Réaliser un rendu offscreen
- Qt OpenGL - Overpainting : dessiner en 2D avec QPainter sur une scène 3D
- Qt OpenGL - Gestion des extensions avec `QGLContext::getProcAddress()`
- Qt OpenGL - Annexes

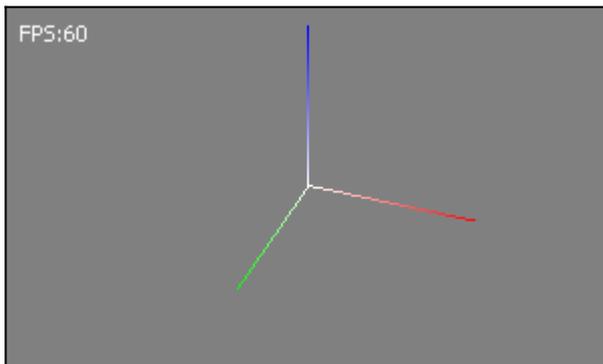
Voici les codes sources des exemples abordés dans ce tutoriel. Pour les tester, il suffit d'ouvrir le fichier .pro avec Qt Creator et de lancer l'application. Il est intéressant aussi d'étudier ces codes sources pour bien comprendre le fonctionnement et l'implémentation d'OpenGL dans Qt.

Remarque : les sources n'ont pas été testé avec les dernières versions de Qt 4. De plus, les versions Qt 5 sont une simple mise à jour du code écrit pour Qt 4 avec MingW 4.4 (ajout du module widgets, suppression des modules dans les includes). Il ne respecte donc pas les bonnes pratiques du C++11 et devra être réécrit pour utiliser les fonctionnalités OpenGL de QtGui et non de QtOpenGL. Pour terminer, le programme d'exemple OpenGL-offscreen ne fonctionne plus sur Qt 5 (problème avec le FBO ou le pBuffer ?), je regarderai le problème dès que possible.

L'ensemble des sources est disponible sur [GitHub](#).

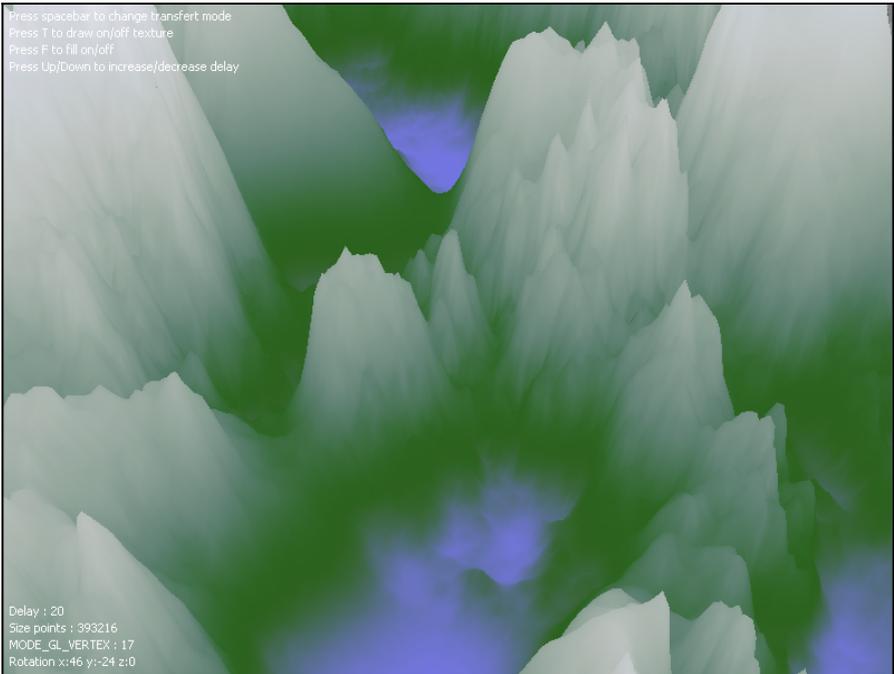
Application minimale QtOpenGL

Cette application permet de présenter le code minimal nécessaire pour initialiser un contexte OpenGL avec Qt. Elle affiche un simple repère orthonormé et le nombre d'images par seconde (FPS). Le bouton droit de la souris permet de tourner autour du repère et la molette permet de s'approcher et de s'éloigner.



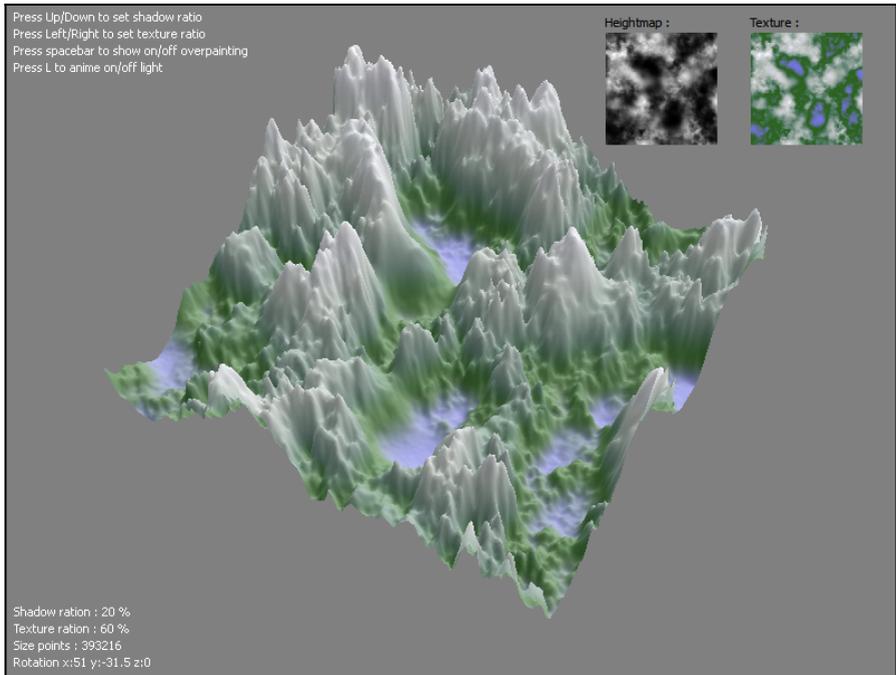
Comparaison entre les différents modes de transfert de données au GPU

Cette application permet de présenter les différentes méthodes de transfert des données entre le CPU et le GPU (directe, vertex array, vertex buffer, indices). Elle affiche une heightmap avec texture. Le bouton droit de la souris permet de tourner autour du repère et la molette permet de s'approcher et de s'éloigner. La touche "T" permet d'afficher ou non la texture. La touche "F" permet d'afficher la heightmap sous forme de fil de fer ou non. La barre d'espace permet de changer de mode de transfert. Les flèches haut et bas permettent de modifier le délai d'affichage entre deux images. Pour déterminer le taux de FPS maximal, il suffit de mettre le délai à 0.



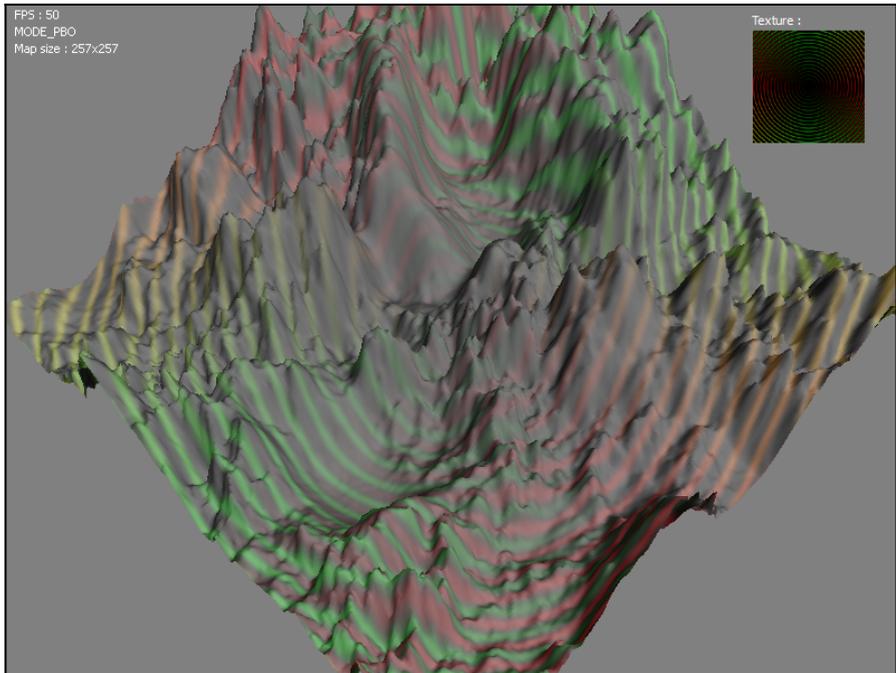
Utilisation des shaders

Cette application permet de présenter l'utilisation des shaders avec Qt. Elle affiche une heightmap avec texture et ombrage de Phong. La source de lumière est de type parallèle, c'est-à-dire que la source de lumière est à l'infini. L'angle d'incidence varie au cours du temps mais il est possible de stopper l'animation en appuyant sur la touche "L". Le bouton droit de la souris permet de tourner autour du repère et la molette permet de s'approcher et de s'éloigner. La barre d'espace permet d'afficher ou non le texte et les images 2D. Les flèches droite et gauche permettent de contrôler l'intensité de la texture. Les flèches haut et bas permettent de contrôler l'intensité des ombrages.



Rendu off-screen

Cette application permet de tester différents modes de rendu off-screen. Elle affiche une heightmap avec texture et ombrage de Phong. Le bouton droit de la souris permet de tourner autour du repère et la molette permet de s'approcher et de s'éloigner. La barre d'espace permet de changer de mode de rendu off-screen.



Références

Les tutoriels de Developpez

Les tutoriels proposés par la rubrique Qt de Developpez :

- Intégration d'OpenGL dans une interface Qt ;
- Qt Graphics et performance - Le moteur de rendu OpenGL ;
- Utiliser OpenCL avec Qt ;
- Intégrer Ogre à Qt ;
- Présentation de GLC_lib.

Les tutoriels proposés par la rubrique Jeux de Developpez :

- Afficher une heightmap avec OpenGL ;

- Génération de terrain par l'algorithme de Perlin ;
- Génération de terrain et triangulation de Delaunay ;
- Génération de textures de terrain ;
- Utilisation de l'algorithme DiamondSquare pour la génération de terrain ;
- Création d'une Skybox en OpenGL ;
- Une introduction à CUDA ;
- CUDA approfondi.

Site officiel OpenGL

Liste de toutes les documentations officielles d'OpenGL (toutes versions).
Voir en particulier :

- Documentation de référence pour OpenGL 2.1 ;
- Documentation de référence pour OpenGL 3.3 ;
- Tableau résumant les principales fonctions d'OpenGL 3.2 ;
- Documentation de référence pour OpenGL 4.1 ;
- Tableau résumant les principales fonctions d'OpenGL 4.1 ;
- Documentation de référence du GLSL.

Copyright

Certaines images utilisées dans ce tutoriel sont issues de Wikipédia. La licence et la liste des auteurs peuvent être consultées sur les pages suivantes :

- L'image de la heightmap. Les images de la texture et de la normal map ont été créées à partir de cette image ;
- L'image des composantes de l'ombrage de Phong.
- La représentation du pipeline graphique est issue du tutoriel OpenGL Shading Language Overview sur le site OpenGL.

Remerciements

Merci à LittleWhite, dourouc05, yan, bafman et AuraHxC pour leur relecture et leurs conseils. Merci à ClaudeLELOUP et à jacques_jean pour leur relecture orthographique très attentive.