

OpenGL dans Qt5

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](#)

Le support d'OpenGL dans Qt5 a été modifié pour mieux l'intégrer avec les nouveaux modules de Qt : QtQuick2 et Qt3D. Cet article présente les modifications apportées dans Qt5.

OpenGL dans Qt4

Dans Qt4, les fonctionnalités d'OpenGL sont implémentées dans un module spécifique, QtOpenGL. Ce module était utilisé par différentes classes pour bénéficier de l'accélération matérielle. Il existe plusieurs méthodes pour activer l'accélération matérielle :

- pour activer par défaut l'utilisation de OpenGL, utiliser la ligne de commande « -graphicssystem opengl » ou la fonction `QApplication::setGraphicsSystem(« opengl »)` ;
- pour `QGraphicsView` (`QtGraphics`) ou `QDeclarativeView` (`QtQuick`), utiliser la fonction `setViewport(new QGLWidget)` ;

- on peut également utiliser QPainter directement sur un QGLWidget ;
- avec le QML Viewer, utilisez la commande « -opengl ».

L'organisation des modules

Dans Qt4, on a donc un problème dans l'organisation des modules. Beaucoup de classes de QtGui peuvent dépendre des classes appartenant à QtOpenGL alors que ce module dépend normalement de QtGui. C'est pour cela que de nouvelles classes font leur apparition dans Qt5 et que les différents modules ont été réorganisés :

- les classes QOpenGL appartiennent au module QtGui et fournissent les fonctionnalités de base permettant l'accélération matérielle pour toute les classes de rendu de Qt ;
- la classe QWidget n'est plus le parent de toutes les classes de rendu. Cette classe et ses dérivées (QGraphicsView) sont transférées dans le module « widgets » ;
- QtQuick 2 utilise maintenant scenegraph à la place de QWidget et la classe QDeclarativeView devient QQuickView et OpenGL est utilisé par défaut ;
- QtOpenGL continue d'exister pour fournir la classe QGLWidget et ses dérivés. Le code OpenGL Qt4 est donc compatible avec Qt5 ;
- le module Qt3d fait son apparition pour fournir un moteur 3D plus avancé. Ce module est suffisamment important et fera donc l'objet d'un article spécifique.

Remarque : il faut faire attention de ne pas confondre le module QtOpenGL, contenant les classes commençant par QGL, et le module QtGui, contenant les classes commençant par QOpenGL (sans t).

Les classes de QtGui dans Qt5

Le module QtGui de Qt5 réutilise des classes existantes du module QtOpenGL de Qt4. Les noms sont explicites :

- QOpenGLBuffer ;
- QOpenGLContext ;
- QOpenGLFramebufferObject ;
- QOpenGLFramebufferObjectFormat ;
- QOpenGLShader ;
- QOpenGLShaderProgram.

Plusieurs nouvelles classes font leur apparition :

- QOpenGLContextGroup : groupe de contextes OpenGL pouvant partager des ressources. Cette classe est gérée automatiquement par les objets QOpenGLContext ;
- QOpenGLFunctions : fournit un accès indépendant de la plateforme aux fonctions d'OpenGL ;
- QOpenGLPaintDevice : permet de dessiner dans un contexte OpenGL avec QPainter ;
- QOpenGLStaticContext : manque de documentation ;
- QOpenGLContextData : manque de documentation ;
- QWindowsGLContext : manque de documentation.

Que faut-il modifier pour utiliser OpenGL dans Qt5

La procédure d'installation ci-dessous concerne les anciennes versions de Ubuntu

Tout d'abord, il faut Qt5. Le plus simple est d'utiliser les dépôts PPA sur Ubuntu : <https://launchpad.net/~forumnokia/+archive/fn-ppa>. Qt5 sera installé dans le répertoire /opt/qt5. Pour ajouter cette version de Qt dans QtCreator, il faut aller dans le menu « Outils » puis « Options... », allez dans « Compiler et exécuter... » puis l'onglet « Versions de Qt ». Cliquer sur « Ajouter » et aller chercher le fichier « /opt/qt5/bin/qmake ». Voilà, Qt5 est prêt à être utilisé.

Pour les versions récentes de Ubuntu (à partir de 13.04 je crois ?), il est possible d'utiliser les paquets libqt5* ou le paquet ubuntu-sdk

```
sudo apt-get update && sudo apt-get install libqt5*  
// ou  
sudo apt-get update && sudo apt-get install ubuntu-sdk
```

Je vous conseille d'installer également les dernières versions des compilateurs C++ : GCC 4.9 et Clang 3.4 (il y a également Clang 3.5 dans les version unstable je crois, à tester) :

```
sudo apt-get update && sudo apt-get install gcc-snapshot  
clang-3.4 clang-3.5
```

Modifier le code

Il est également nécessaire de modifier un peu (ou beaucoup, si vous utilisez des fonctionnalités dépréciée de Qt 4) votre code. La première chose à faire est d'ajouter le module "widgets" (remarque : l'ajout des modules "core" et "gui" n'est pas obligatoire puisqu'ils sont inclus par défaut, je les remets pour que cela soit plus clair) :

```
QT *= core gui opengl // *= n'ajoute que les nouvelles  
valeurs  
greaterThan(QT_MAJOR_VERSION, 4) {  
    QT += widgets  
}
```

Le seconde chose à faire est de supprimer la déclaration des modules dans les includes. Par exemple :

```
#include <QtGui/QWidget>  
// devient  
#include <QWidget>
```

Dans la majorité des cas, ces deux modifications seront suffisantes. Si ce n'est pas le cas, il faudra modifier en cas par cas, n'hésitez pas à demander sur les forums.

[OpenGL,, Qt,, Qt5](#)