Support d'OpenGL dans Qt

Le support d'OpenGL dans Qt 5 a été modifié pour mieux l'intégrer avec les nouveaux modules de Qt : QtQuick2 et Qt3D. Cet article présente les modifications apportées dans Qt 5.

Activer OpenGL dans Qt 4

Dans Qt 4, 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, utilisez la ligne de commande "-graphicssystem opengl" ou la fonction `QApplication::setGraphicsSystem("opengl")`, dans la fonction `main` par exemple ;
- pour `QGraphicsView` (QtGraphics) ou `QDeclarativeView` (QtQuick), utilisez la fonction `setViewport(new QGLWidget)` ;

- on peut également utiliser `QPainter` directement sur un `QGLWidget`, qui est une classe dérivée de `QWidget` avec un contexte OpenGL ;
- avec le QML Viewer, utilisez la commande ”-opengl”.

L’organisation des modules dans Qt 5

Dans Qt 4, le support d'OpenGL était donc optionnel, dans un module dédié. Il fallait créer spécifiquement un contexte OpenGL et le passer en paramétré pour bénéficier de l’accélération matérielle.

Dans Qt 5, l'objectif a été de fournir un support minimal d'OpenGL dans QtGui, ce qui permet de l'utiliser dans tous les modules graphiques (widgets, Qt Quick) qui dépendent de QtGui.

- les classes `QOpenGLXxx` 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` par exemple) sont transférées dans le module “widgets” ;
- les vues QtQuick 2 ne sont plus basées sur `QWidget`, la classe `QDeclarativeView` devient `QQuickView` et OpenGL est utilisé par défaut ;
- le module QtOpenGL continue d’exister pour fournir la classe `QGLWidget` et ses dérivés (les classes qui commencent par `QGLXxx`). Ce module permet de conserver le code Qt 4 compatible avec Qt 5, mais ces classes sont dépréciées au profit des classes `QOpenGLXxx`.

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

Les classes QOpenGL dans Qt 5.4

Vous trouverez la liste de toutes les classes `QOpenGLXxx` dans la documentation de Qt.

- QOpenGLBuffer : tableau de données en mémoire (buffer), permet de transférer des données avec le GPU ;
- QOpenGLContext
- QOpenGLContextGroup : groupe de contextes OpenGL pouvant partager des ressources. Cette classe est gérée automatiquement par les objets QOpenGLContext ;
- QOpenGLDebugLogger
- QOpenGLDebugMessage
- QOpenGLFramebufferObject
- QOpenGLFramebufferObjectFormat
- QOpenGLFunctions : fournit un accès indépendant de la plateforme aux fonctions d'OpenGL ;
- QOpenGLFunctions_1_0
- QOpenGLFunctions_1_1
- QOpenGLFunctions_1_2
- QOpenGLFunctions_1_3
- QOpenGLFunctions_1_4
- QOpenGLFunctions_1_5
- QOpenGLFunctions_2_0
- QOpenGLFunctions_2_1
- QOpenGLFunctions_3_0
- QOpenGLFunctions_3_1
- QOpenGLFunctions_3_2_Compatibility
- QOpenGLFunctions_3_2_Core
- QOpenGLFunctions_3_3_Compatibility
- QOpenGLFunctions_3_3_Core
- QOpenGLFunctions_4_0_Compatibility
- QOpenGLFunctions_4_0_Core
- QOpenGLFunctions_4_1_Compatibility
- QOpenGLFunctions_4_1_Core

- QOpenGLFunctions_4_2_Compatibility
- QOpenGLFunctions_4_2_Core
- QOpenGLFunctions_4_3_Compatibility
- QOpenGLFunctions_4_3_Core
- QOpenGLFunctions_ES2
- QOpenGLPaintDevice : permet de dessiner dans un contexte OpenGL avec QPainter ;
- QOpenGLPixelTransferOptions
- QOpenGLShader
- QOpenGLShaderProgram
- QOpenGLTexture
- QOpenGLTimeMonitor
- QOpenGLTimerQuery
- QOpenGLVersionProfile
- QOpenGLVertexArrayObject
- QOpenGLWidget
- QOpenGLWindow

Notes de mise à jour
Ce tutoriel concerne que les classes de Qt Gui, non QtOpenGL.

OpenGL,, Qt,, Qt5