

# Trolltech DevDays 2006 : Qt Graphics View

par Matthieu Brucher (<http://matthieu-brucher.developpez.com/>) (Blog)

Date de publication : 01/11/2006

Dernière mise à jour :

Qt4.2 propose un tout nouveau système pour remplacer QCanvas, disparu depuis Qt3.

- I - Qui est Gunnar Sletta ?
- II - Résumé de la présentation
  - II-A - Qu'est ce qu'un canevas ?
  - II-B - L'ancien QCanvas
  - II-C - Graphics View
    - II-C-1 - Les items
    - II-C-2 - La scène
    - II-C-3 - La vue
    - II-C-4 - L'accélération OpenGL
  - II-D - Porter des applis utilisant QCanvas vers le nouveau framework
- III - Conclusion

## I - Qui est Gunnar Sletta ?

Gunnar Sletta est développeur chez Trolltech et s'occupe plus particulièrement de QtJambi ainsi que du graphisme dans Qt au travers d'Arthur - le nom du système de dessin -, du framework Graphics View, ...

## II - Résumé de la présentation

### II-A - Qu'est ce qu'un canevas ?

Un canevas est un matériel sur lequel une peinture est effectuée. Pour Qt4.2, cela signifie que c'est un widget avec lequel un artiste peut travailler, que ce soit avec des SVG, des diagrammes, des jeux avec des sprites, ... Mais plus que cela, on peut dessiner toute forme, tout item, on possède un haut niveau d'abstraction - pas besoin de savoir qui on dessine à quel moment et comment -, le tout en mouvement avec des drag&drop.

Pour d'autres frameworks, un canevas n'est pas la même chose. Par exemple, pour Java et Tk, c'est un moyen d'obtenir un widget maison ; pour Gtk, c'est une API permettant de créer des graphiques structurés ; pour Qt3, c'est une amélioration de QwSpriteFiled permettant de dessiner des items 2D.

### II-B - L'ancien QCanvas

Dans Qt3, QCanvas consistait en une API basée sur les items. Il était possible de se déplacer et d'effectuer diverses transformations.

En revanche, impossible d'avoir une interaction entre les items, pas de regroupement, pas de contrôles avancés, pas rendu accéléré - Level Of Details, ... -, ...

### II-C - Graphics View

Un Graphics View consiste en trois parties, les items, une scène et une vue sur la scène.

#### II-C-1 - Les items

Les items possèdent une API fournie, mais simple. Les coordonnées peuvent être en nombre flottant, et un système de coordonnées locales est proposé, permettant d'être indépendant de l'orientation ou d'un facteur d'échelle.

Pour chaque item, il existe 2 niveaux d'utilisation. Le premier est le rectangle englobant. Il permet de simplifier les calculs de collision, ... Le deuxième est la forme dessinée elle-même.

Un item est dérivé de QGraphicsItem. Cela lui permet d'être transformé, tourné, déplacé, tordu, ... Tout cela sans affecter le système de coordonnées. De plus, un item peut contenir d'autres items, et les transformations peuvent être propagées, ce qui permet de gérer des squelettes. Enfin, on peut gérer un niveau de détails - LOD - dans la fonction de peinture, des événements, ...

#### II-C-2 - La scène

La scène est l'endroit où sont placés les items. Elle n'est pas responsable de la visualisation - si ce n'est pour un screenshot ou une impression -, ce n'est qu'un conteneur, avec un système de coordonnées globale.

Le suivi de la position des items est effectué à l'aide d'un arbre binaire. Il n'est pas optimisé pour les changements massifs de position, mais il est possible de le désactiver s'il ne fonctionne plus aussi bien que le système par défaut. Grâce à ce système, il est possible d'utiliser plusieurs dizaines de milliers d'items...

Outre la couche des items, il est possible d'avoir une couche sur le fond et au-dessus des items.

### II-C-3 - La vue

La vue est le widget sur lequel une partie de la scène sera affichée. On peut naturellement parcourir la scène, mais aussi zommer, effectuer des rotations.

### II-C-4 - L'accélération OpenGL

Pour l'utilisation d'OpenGL, il faut spécifier à la vue que le viewport est un QGLWidget. A partir de là, on peut utiliser QPainter ou des instructions OpenGL directement, dans le système local de l'item dessiné.

### II-D - Porter des applis utilisant QCanvas vers le nouveau framework

Les concepts entre QCanvas et Graphics View sont proches, de même que l'API. Le gros point est le changement du suivi de l'emplacement des items, il passe d'un système fixe à partir de morceaux fixes de la scène à un arbre binaire de recherche. Or par le passé, il fallait parfois utiliser le système de "chunks" pour dessiner rapidement. Heureusement, plusieurs exemples de portage sont donnés dans les exemples - canvas et asteroids -.

### III - Conclusion

Enfin le remplaçant de QCanvas, parti lors de la sortie de Qt4, revenu par la petite porte dans un QSolution, et voici une version avec son lot de nouveautés et d'améliorations. Pour plus de renseignements, **une page d'aide** sur le site de Trolltech est proposée.

