

Bonjour à tous

Les tutoriels sur internet sont une source d'information et d'auto-formation très intéressante sur internet. Cependant, il faut reconnaître qu'il est difficile de trouver des tutoriels de qualité, on doit souvent se contenter de tutoriels de qualité plus ou moins bonne, plus ou moins bien rédigés, plus ou moins exacts.

L'un des points que j'appréciais sur Developpez.com est le processus de validation en plusieurs étapes des articles. La vie d'un article commençait sur le forum privé de l'équipe de rédaction C++, l'auteur devait lancer une discussion pour avoir une relecture technique. Après validation et corrections, l'article passait ensuite en gabarisation pour mettre dans un format XML spécifique, utilisé pour générer les différentes versions (en ligne, hors ligne, PDF, ebook, etc). Pour terminer, l'article passait dans les mains des correcteurs, qui valident l'orthographe, la grammaire et parfois les tournures des phrases. Au final, 5 à 10 personnes contribuaient à chaque article, ce qui garantissait sa qualité.

Depuis que je rédige mes articles sur mon propre blog, je n'ai plus cette relecture technique et orthographique avant publication. Il me faut attendre les commentaires de lecteurs pour corriger d'éventuelles erreurs ou préciser certains points.

En revanche, il y a un défaut que je retrouve sur tous les sites : l'absence de mise à jour des tutoriels. Sur Developpez.com, quand je suis devenu responsable de la rubrique C++, il n'y avait plus de responsable C++ depuis plusieurs années. La majorité des tutoriels et la FAQ datent de plusieurs années (c'est encore le cas) et j'ai dû, à plusieurs reprises, déconseiller les cours de C++ de base aux débutants. La situation n'est pas mieux ailleurs, le tutoriel C++ débutant sur OpenClassRooms n'est pas à jour pour le C++ (C++11) et Qt (Qt 5).

Et je dois reconnaître que le problème se pose aussi pour mes propres tutoriels. Le tutoriel sur Qt Android, le tutoriel le plus consulté sur mon blog, n'est pas à jour pour Qt 5.2. Mon tutoriel sur le C++11 n'a pas été finalisé, ni mis à jour pour le C++14. C'est la raison pour laquelle je crée ce wiki : permettre à tous de corriger ou mettre à jour mes tutoriels. Et pour rappel, mes tutoriels sont tous (même mes anciens articles sur

Developpez.com) sous licence Creative Common BY-NC-SA. Je demande simplement de citer la source.

Même si je choisis les articles que je rédige en fonction de mes propres centres d'intérêt, je suis ouvert aux suggestions. Si vous le souhaitez, vous pouvez proposer des thèmes sur la page [Propositions de sujets d'article](#). Si vous souhaitez également que j'ajoute des fonctionnalités sur ce wiki (doluwiki), vous pouvez également le faire sur cette page.

Derniers articles publiés

- [C++1y - File System](#)
- [Les signaux et slots dans Qt 5](#)
- [Pourquoi le RAII est-il fondamental en C++ ? ?](#)

Articles C++

Documentation de référence : cppreference.com

- [Pourquoi le RAII est-il fondamental en C++ ? ?](#)
- [Ça ne sert à rien de se prendre la tête avec les nouvelles normes du C++ !](#)
- [Pourquoi le C++ est-il un langage plus adapté pour les débutants que le C ? ?](#)

Le C++ 1y

- [Les derniers jours du C++11 ? ?](#)
- [Dates de sortie des prochaines normes du C++](#)
- [Généricité, concepts et C++14](#)

1. C++1y - Les tableaux
2. C++1y - File System
3. 1. typage faible
4. C++1y - Parallelism & Concurrency
5. C++1y - Programming Langage C++
6. C++1y - Library Fundamentals

Le C++11

- Les nouvelles fonctionnalités du C++11
- Le design des bibliothèques en C++11

Les rvalue reference et la move semantic

1. Problématiques rencontrées dans le C++03
2. Quelques définitions
3. Les nouveaux concepts du C++11
4. Les rvalue references
5. Analyse de quelques implémentations
6. Ce que cela change pour la STL

Généralités

- Les accesseurs et les détails d'implémentation
- Commencer facilement avec Boost Graph
- Pile, Tas, portée et durée de vie

Articles Qt

Documentation de référence : qt-project.org

- [Les signaux et slots dans Qt 5](#)
- [Déployer une application Qt](#)
- [Les modules de Qt 5](#)
- [Introduction au module QTest](#)
- [QtScript - utilisation des prototypes](#)

Exemple de code Qt : création d'un colorpicker

1. [Enoncé de l'exercice](#)
2. [Version Qt](#)
3. [Version QML](#)
4. [Benchmark et optimisations](#)

Qt Android

- [Développez en natif pour Android avec Qt](#)
- [Qt Quick Controls sur Android](#)
- [Qt sur OUYA](#)

Utiliser une manette de jeux avec Qt

1. [Implémentation Windows \(première partie\)](#)
2. [Implémentation Windows \(deuxième partie\)](#)
3. [Implémentation Windows \(troisième partie\)](#)
4. [Utilisation du gamepad en QML](#)

Qt Quick et QML

- [Qt Quick Controls dans Qt 5.1](#)

Articles OpenGL

Techniques divers

- [Illumination globale et partitionnement de l'espace](#)
- [Débuguer avec OpenGL 4](#)
- [Introduction aux geometry shaders](#)
- [La technique d'instanciation](#)
- [Les queries objects](#)

Tutoriel : OpenGL et Qt

1. [Introduction](#)
 2. [Générer un terrain](#)
 3. [Envoyer des données au processeur graphique](#)
 4. [Utilisation du pipeline programmable](#)
 5. [Ajouter des lumières et des textures](#)
 6. [Réaliser un rendu offscreen](#)
 7. [Overpainting](#)
 8. [Gestion des extensions](#)
 9. [Annexes et codes sources des exemples](#)
- [Support d'OpenGL dans Qt](#)

Tutoriels vidéo

- [Introduction à Qt](#)
- [Installation et premier pas avec Qt 5.2 sur Windows](#)
- [Savoir utiliser la documentation de Qt 5.2](#)

Le C++ User Group France

- [Rencontre C++ Francophone - Introduction](#)
- [Du Polymorphisme dynamique au polymorphisme statique : abstraction sans perte de performances](#)
- [Ajouter une autre dimension à la qualité des logiciels](#)

Les éditions D-BookeR

- [Installation de Qt 5.0 sur Ubuntu via le dépôt Synaptic](#)
- [Installation de Qt 5.0 sur Ubuntu](#)
- [Installation de Qt 5.0 sur Windows avec Visual Studio](#)
- [Installation Qt 5.0 sur Windows avec MinGW](#)

Sélection de livres

- [Sélection de livres](#)

Mes autres sites

- [Blog](#)
- YouTube
- Google+
- GitHub
- Les éditions D-BookeR
- StackOverflow