

## Programme de formation Librairie Qt

### • Objectifs

Formation de prise en main de la librairie Qt, une librairie multiplateforme principalement utilisée sur une base C++ (possibilité de l'utiliser aussi avec du javascript ou python). Cette formation s'adresse principalement aux développeurs concernés par la programmation d'IHM, bien que ce ne soit pas le seul avantage de la librairie. Ce cours pourra donc aussi servir de bases pour partir dans une toute autre utilisation de Qt (multithreading, réseau, xml, sql, son, vidéo, ...)

### • Pré requis

Connaissances de C++

### • Durée

5 jours

### • Public

Développeurs, Informaticiens

### • Plan de formation

#### Présentation de la librairie Qt et de son SDK

QtCreator  
Documentation

#### Idée derrière la librairie

« Code less, create more »  
Multiplateforme, rapide et facile à coder

#### Fondement d'une application

« Hello world ! » dans QtCreator  
Créer une application  
Découverte des widgets

#### Principe de base : Le QObject

Le principe de parent/enfants  
Communication par signaux  
Notification par event

#### Les widgets

QWidget  
widget les plus utilisés

#### Les layouts, arrangement de l'espace entre widgets

Principe des layout  
Les différents layouts  
Commander l'apparence de vos applications

#### Autour de l'application

Autour de la fenêtre principale

Stocker et retrouver les settings de l'application  
Les ressources de l'application  
Internationalisation : Traduction des textes  
Style : C ou stylesheets appelé qss (ie css)  
TP tout au long des différents points

#### Evaluation suivie d'une correction collective

Questions-réponses  
Retour sur des points à compléter selon les besoins

#### Popup Qt

Fenêtres de popup fournies par la librairie  
Mettre au point vos propres fenêtres de popup

#### QWidget personnalisé

Réaliser le dessin du widget  
Commander leurs comportements

#### Container

Les différents containers et leurs usages  
Les fonctions et macros associées  
Gestion de fichiers  
Popups utiles  
Interface de fichier  
Interface système  
Lire et écrire des fichiers

#### Variants

Avantages des variants, leur utilisation à

travers la librairie  
Classes supportées  
Comment rendre variant vos objets

## **Débogage et Bonnes habitudes de codage**

Truc et astuce pour faciliter le débogage  
Fonctions pour une application plus robuste

## **QDesigner**

Utilisation de Designer pour générer des écrans rapidement  
Écrire une librairie pour Designer, inclure vos widgets dans les UI

## **Thread**

Pourquoi utiliser des threads  
Les Thread et classes associées  
Développer une application Concurrente  
Comportement des signaux à travers les threads

## **Model/View**

Utilisation des modèles  
Isoler les adhérences entre interface, rendre vos applications plus faciles à maintenir  
Les vues génériques  
Déclarer vos propres modèles pour répondre aux structures les plus évolués  
Maîtriser l'affichage des données

## **QGraphicsView**

Utiliser l'interface de rendu 2D (cartographie, manipulation 2D, affichages graphiques)  
Utiliser les classes disponibles  
Créer ses propres items graphiques

## **QAnimation**

Donnez du mouvement à vos transitions, des allures d'application mobile

## **Pointeurs intelligent**

Utilisation de pointeur intelligent  
Les classe de pointeur Qt, dans quel contexte les utiliser

## **Optimisation et gestion de la mémoire**

Comprendre les optimisations  
Les données partagées