

## FORMATION "INTRODUCTION À QT (WIDGETS ET QML)"

### PROGRAMME

## Formateurs

La société KDAB possède une expérience de plus de 20 années dans le développement d'applications utilisant la bibliothèque Qt, ainsi que dans les formations à ce type de développement.

La filiale française KDAB (France) propose des services de développement, de conseil, d'assistance et de formation, autour de Qt.

La déclaration d'activité de formation de KDAB (France) a été enregistrée sous le **numéro 93 84 03221 84** auprès du préfet de région de Provence-Alpes-Côte d'Azur.

Le Directeur Général de KDAB France, David Faure, ingénieur en informatique de l'INSA de Lyon, est développeur KDE (bureau de logiciel libre basé sur Qt) depuis 1998, et a participé au sein de KDAB à de nombreux projets de développements basés sur Qt, depuis 2003. Il a aussi effectué de nombreuses formations Qt dans toute l'Europe. Il travaille depuis les bureaux de KDAB (France) dans la banlieue d'Avignon.

Nicolas Arnaud-Cormos, ingénieur d'études et développement de l'Université de Rennes, a rejoint KDAB en 2007, il est formateur Qt, ainsi que développeur et chef de projet, en particulier pour un projet de logiciel de visualisation scientifique 2D/3D. Il est basé sur Limoges.

Paul Lemire, ingénieur en informatique de l'Epitech de Paris, fait partie de l'équipe de développement de Qt 3D et participe à des projets OpenGL 2 (dont écriture de shaders). Il est formateur QML et est basé sur Brest.

Franck Arrecot, titulaire d'un Master Informatique et Développement Logiciel de l'Université de Toulouse utilise le framework Qt depuis 2011. Il a rejoint KDAB en 2014 en tant que développeur et formateur spécialisé sur les technologies QtWidgets et QtQuick. Il intervient régulièrement dans des conférences pour y parler techniques de développement ou faire découvrir le travail de la communauté KDE. Il est basé sur Bayonne.

Mike Krus, titulaire d'un doctorat en réalité virtuelle de l'Université Paris Sud, est un développeur C++ depuis de nombreuses années, et utilise Qt sur plusieurs plateformes, y compris mobiles, depuis plus de 12 ans. Il a rejoint KDAB en 2015 où il est également formateur Qt et OpenGL. Il participe au développement de Qt3D et est responsable du port de Qt pour tvOS. Il est basé en Ecosse.

## Durée

En cas de formation sur site, la formation "Introduction à Qt (Widgets et QML)" dure de 3 jours (21 heures) à 5 jours (35 heures), au choix du client.

En cas de formation inter-entreprises, la formation dure 5 jours (35 heures).

## Objectif

Cette formation enseigne les techniques de développement d'applications graphiques, réseau et/ou multitâches, en utilisant les technologies Qt Widgets et Qt Quick dans les langages C++ et QML.

A l'issue de la formation, les participants seront capables d'appliquer leur nouvelle compréhension des principes de Qt et leur nouveau savoir-faire pour le développement d'applications basées sur Qt.

## Audience

Cette formation est destinée aux programmeurs d'applications et systèmes ayant à connaître la programmation Qt.

## Prérequis

Une expérience professionnelle de la programmation objet est indispensable, et une expérience en C++ est fortement recommandée. Cependant si le client en exprime le besoin, la première demi-journée de formation pourra être consacrée à des rappels de C++, afin de s'assurer que les notions nécessaires à la suite de la formation sont bien connues par les participants.

## Compétences professionnelles visées

- Compiler et exécuter des applications Qt avec qmake et QtCreator
- Utiliser les signaux et les slots pour faire communiquer des objets entre eux
- Développer une interface graphique à l'aide des widgets Qt, en C++
- Développer une interface graphique à l'aide des widgets Qt, en utilisant Qt Designer
- Développer un widget avec un rendu personnalisé
- Maîtriser les bases du langage QML
- Développer une interface graphique à l'aide des éléments QtQuick, en QML
- Utiliser les propriétés liées et les gestionnaires de signaux pour gérer l'interaction avec l'utilisateur
- Ecrire un modèle pour fournir des données à une vue de type liste, table ou arborescente
- Personnaliser le rendu et l'édition des items dans une telle vue, au moyen du delegate
- Utiliser les classes fournies par Qt, en particulier QtCore, en suivant les bonnes pratiques établies par la communauté de développeurs Qt au fil des années
- Créer des applications hybrides C++/QML

## Thèmes abordés

### 1. Rappels de C++ (si demandé par une majorité de participants)

- Qu'est-ce que C++
- Une pile en C
- Une pile en C++
- Constructions de bases

- Héritage
- Pointeurs et références
- Surcharge d'opérateurs
- Templates
- Exceptions
- Différences entre Java et C++
- Références de lecture pour approfondir

## 2. Les bases de Qt

- L'historique de Qt
- Développer une application Hello World (Widgets et QML)
- Hello World avec Qt Creator
- Conseils pratiques pour développeurs

## 3. Les objects Qt

- Fonctionnalités communes aux objects Qt
- Communication entre objets à base de signaux et slots
- Variations sur les signaux/slots
- Gérer des événements avec Qt

## 4. Classes QtCore

- Gestion des chaînes de caractères
- Classes conteneur
- Gestion de fichiers
- QVariant et système de propriétés

## 5. Widgets

- Widgets courants
- Gestion des layouts
- Conseils pour les widgets personnalisés

## 6. Dessin et styles de widgets

- Dessiner un widget
- Gestion des couleurs
- Opérations de dessin

## 7. Boîtes de dialogue et Qt Designer

- Principes généraux
- Boîtes de dialogue fournies par Qt

## 8. Qt Designer

## 9. Introduction à Qt Quick

- Première rencontre de Qt Quick
- Les concepts

## 10. Composer des interfaces utilisateur

- Eléments imbriqués

- *Éléments graphiques disponibles*
- *Élément texte*
- *Layout « ancrage »*

#### 11. *Interaction avec l'utilisateur*

- *Interaction à la souris*
- *Interaction tactile*
- *Interaction au clavier*

#### 12. *Composants*

#### 13. *Animations*

#### 14. *Présenter des données*

- *Ordonner des éléments graphiques*
- *Modèles de données simples*
- *Vues*
- *Personnaliser les vues*
- *La vue « chemin » (Path View)*

#### 15. *Intégrer QML avec C++*

- *Environnement déclaratif*
- *Exporter des objets C++ en QML*
- *Exporter des classes en QML*
- *Exporter des classes non graphiques*
- *Exporter des classes graphiques*
- *Utiliser des types non-standard*

*Les formations de plus de 3 jours comportent aussi un nombre variable de sujets supplémentaires, que le client (ou les participants, à la majorité, pour les formations inter-entreprises) devra choisir parmi les sujets suivants :*

#### 16. *Sujets supplémentaires QML*

- *Machines à états*
- *Modèle de données complexes*
- *Dessin sur Canvas et effets visuels*
- *Élément "Loader"*
- *Modules*
- *Plugins*
- *Traductions (internationalisation) avec QtQuick*
- *WebKit*
- *Glisser/déplacer*

#### 17. *Interaction utilisateur*

- *Support des gestes (gestures)*
- *Support des écrans tactiles (touch)*
- *Animations*

#### 18. Multitâches

- *Concepts*
- *Primitives de synchronisations*
- *Fondations*
- *QtConcurrent*

#### 19. XML, JSON et SQL

- *Utiliser XML dans Qt*
- *QtXmlPatterns*
- *XML Schema*
- *JSON*
- *Bases de données SQL*

#### 20. Infrastructure d'applications

- *Système d'évènements*
- *Model/View*
- *Traduction (internationalisation)*
- *Aide en ligne*
- *Plug-ins*
- *Machine à état*
- *Resources*
- *Sauvegarde de paramètres (settings)*
- *Multimédia (phonon)*

#### 21. Développement et tests

- *Licenses Qt*
- *Déploiement*
- *Outils de développement sous Linux*
- *QMake*
- *Portabilité*
- *Aides au débogage*
- *Tests unitaires*

#### 22. Graphiques et styles

- *Graphiques 2D avec Graphics View*
- *Développement d'un style de widget*
- *Utilisation d'OpenGL dans Qt*
- *Transformations 2D*

#### 23. Communication inter-processus

- *Sockets réseau (QtNetwork)*

- QtDBus
- QProcess
- Mémoire partagée

#### 24. Modules Qt supplémentaires

- Fenêtre principale
- WebKit
- Validation de la saisie
- Texte enrichi
- Qt Script
- ActiveQt
- Glisser/déplacer

#### 25. Migration

- Techniques
- Portage Motif - Qt

#### 26. Qt pour Android

- Mise en place
- Extension à l'aide de JNI
- Conseils pratiques pour développeurs

*Il est bien sûr impossible de faire tous les sujets ci-dessus en 5 jours; les 2 derniers jours se limitent donc à une sélection d'un petit nombre de sujets parmi la liste ci-dessus.*

## Moyens pédagogiques

*Le support de formation comporte plus de 1300 pages en anglais, et est régulièrement mis à jour pour suivre les évolutions de Qt.*

### Formations en présentiel

*Le support de formation est projeté sur écran au moyen d'un vidéo-projecteur. Une copie imprimée du support de formation est remise à chaque participant.*

*Une clé USB est remise à chaque participant, avec les points de départ pour les travaux pratiques, leurs corrigés, et le code des exemples présentés pendant la formation. Chaque participant peut utiliser son propre ordinateur, ou prévoir au minimum un ordinateur pour deux participants, lors de formations sur site client.*

### Formations à distance

*Le support de formation est envoyé en PDF nominatif, par mail, à chaque participant. Ce même support est diffusé via un partage d'écran durant la formation.*

*Les points de départ pour les travaux pratiques, leurs corrigés, et le code des exemples présentés pendant la formation, sont eux aussi envoyés par mail, à chaque participant, avant le début de la formation.*

*L'outil gratuit Zoom est utilisé pour la visio-conférence pendant toute la formation. Les participants reçoivent dans leur convocation par mail le lien zoom qui permet de rejoindre la visio-conférence. En cas de soucis technique avec Zoom, le mail de convocation indique aussi comme utiliser un tchat (indépendant de Zoom) pour communiquer avec le formateur. Si cette solution ne fonctionnait pas non plus, les participants peuvent appeler le standard KDAB (04 90 84 08 53) afin d'être mis en relation avec un ingénieur en informatique expérimenté avec l'utilisation de Zoom pour obtenir de l'aide.*

*Lors des travaux pratiques, le formateur crée des salles virtuelles séparées avec 2 à 3 participants par salle, pour un travail en groupe sur les exercices. Le formateur fait régulièrement le tour des salles virtuelles pour aider les groupes. Le temps nécessaire à la réalisation de ces travaux est donc inclus dans la durée de la formation (pas de travail à effectuer le soir).*

## Résultats

*A l'issue de la formation, les participants seront capables d'appliquer leur nouvelle compréhension des principes de Qt et leur nouveau savoir-faire pour le développement d'applications basées sur Qt.*

## Évaluation

*Une évaluation écrite faite dans la dernière demi-journée permettra d'établir le niveau en développement Qt des participants à la formation.*

*Après l'examen individuel, les questions et les bonnes réponses sont présentées en séance pour débattre avec les participants. Le résultat individuel de chaque participant lui est indiqué à l'issue de la formation.*

## Formations à distance

*Le questionnaire est envoyé par mail aux participants au moment de l'évaluation (environ une heure avant la fin de la formation). Les participants renvoient par mail le questionnaire rempli.*

## Suivi après la formation

*Il est possible de compléter la formation par une formation avancée sur Qt Widgets, sur QML, sur le multitâche, ou sur Model/View (3 jours chacune). Il est aussi possible de commander une assistance au développement (conseils, débogage, ...) auprès de KDAB (France).*