Développez vos talents
Certifiez vos compétences

Langage C++

Librairie C++ Qt5 et QML - Fonctionnalités avancées

3 jours (21h00) | ★★★★½ 4,6/5 | CPP-QT5PE | Code Certif Info : 94827 | Certification M2i Langages de Programmation (non incluse) | Évaluation qualitative de fin de stage | Formation délivrée en présentiel ou distanciel [1]

Formations Informatique › Langages et développement › Langage C++

À l'issue de ce stage vous serez capable de :

- Développer rapidement des applications avec le module Qt Quick
- Améliorer les performances des applications (code, graphique...)
- Gérer une programmation orientée flux de données
- Eviter les erreurs courantes et répondre mieux aux exigences des utilisateurs
- Créer des composants personnalisés.

Niveau requis

Avoir développé en C++. Avoir suivi la formation CPP-QT5IN "Librairie C++ Qt5 et QML - Initiation" ou connaître les bases de Qt Core, QML et Qt Quick.

Public concerné

Développeurs de logiciels.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

(1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel * (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes ** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel). Environnements de formation installés sur les postes de travail ou en ligne. Supports de cours et exercices.

* Nous consulter pour la faisabilité en distanciel. ** Ratio variable selon le cours suivi.

Programme

Rappels de Qt / QML

L'intégration entre C++ / Qt et QML / JavaScript et ses limites

La boucle d'évènement de Qt et son impact sur Qt Quick

Problèmes fréquemment rencontrés

- QQmlListProperty
- Threading
- Types non reconnus...

Sujets avancés sur Qt Quick

Création de composants visuels en C++

- Avec une base Qt Scene Graph : la solution idéale mais limitée
- Avec une base QPainter : la solution de compatibilité
- Le composant hybride C++ / QML : une solution largement répandue

Intégrer des images générées ou ne provenant pas d'un fichier local

- Les "images provider"
- Les composants spécialisés
  - Texture
  - Dessin procédural
  - FBO OpenGL...
- Les moyens pour se passer de C++ dans certains cas
  - SVG généré à la volée
  - Canvas...

Optimisation des performances

Les pratiques permettant d'améliorer naturellement les performances du code

- Le "proxying" des bindings pour éviter les goulets d'étranglement
- L'utilisation de C++ à la place de JavaScript pour les traitements lourds
- La résolution des noms en QML : l'épineux problème du "scoping"

Performances graphiques : éléments à prendre en compte

- Les objets graphiques superflus
- Les mauvaises utilisations dans les "delegates" d'une vue MVC
- Les effets de bords des mauvaises utilisations de composants basiques
  - Image
  - Text...

Techniques pour rendre le code plus compréhensible

La programmation orientée flux de données

- Différences avec l'orienté objet classique / naïf
- Application des principes en QML et C++
- Implications sur l'architecture globale

La séparation C++ / QML

- Bien placer la séparation entre les deux langages
- Eviter que le code QML soit pollué par de mauvaises utilisations de C++
- Simplifier le code côté C++ pour qu'il ne soit pas un frein à son utilisation

Eviter les erreurs courantes

- Rendre le code QML sémantiquement plus clair
- Fuir les cas où JavaScript peut produire des erreurs silencieuses
- Limiter au maximum la redondance dans le code, notamment entre C++ et QML

La création de composants personnalisés : l'importance du cloisonnement

- Empêcher les bugs potentiels par intrusion externe dans le composant
- Rendre un composant réellement réutilisable et portable pour un gain de temps
- La spécialisation en cascade : un bon moyen de coder proprement et simplement avec QML

Questions restantes : une fois que la technique pure est réglée

Ces problématiques qui apparaissent en fin de projet, lorsque le produit doit être diffusé

- La création de "vrais" modules de plug-ins QML
- Le déploiement sur les plateformes sans système de package (Windows...)
- La dynamique traduction de l'interface (Qt Linguist)

Certification (en option)

- Prévoir l'achat de la certification en supplément
- L'examen (en français) sera passé le dernier jour, à l'issue de la formation et s'effectuera en ligne
- Il s'agit d'un QCM dont la durée moyenne est d'1h30 et dont le score obtenu attestera d'un niveau de compétence

Modalités d'évaluation des acquis

L'évaluation des acquis se fait :

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation ou une certification (M2i ou éditeur)

Compétences visées

- Développer des applications ou logiciels conformément au cahier des charges
- Intégrer un produit ou un programme à partir des outils, méthodes ou langages
- Etre autonome dans l'exécution des tâches
- Préparer les plans de tests d'une application
- Conduire les tests unitaires
- Optimiser les tests de performance d'une application
- Assurer la veille technique et concurrentielle.