

DURÉE : 3 JOURS (21 HEURES)

SANCTION DE LA FORMATION : ATTESTATION DE FIN DE STAGE ET DE PRÉSENCE, ÉVALUATION DES ACQUIS

NATURE DE LA FORMATION : ADAPTATION ET DÉVELOPPEMENT DE COMPÉTENCES

OBJECTIFS

- Maîtriser langage QML de programmation d'interface utilisateur tactile
- Maîtriser les concepts de la technologie Qt Quick 2
- Maîtriser Qt Quick Designer et l'environnement de mise au point QML Debugger
- Mettre en oeuvre des composants graphiques dynamiques complexes
- Développer des applications Qt Quick 2 tactiles en QML
- Concevoir une architecture logicielle avec des composants Qt Quick 2 et des objets C++ Qt 5 à base de plugins
- Développer des applications Hybrides QML/C++

PUBLIC ET PRE-REQUIS

La formation Qt Quick 2/QML est particulièrement destinée aux développeurs ayant de très bonnes notions de programmation Objet C++ sous Qt 5, confrontés aux problèmes de développement d'applications graphiques tactiles de type Tablette/Smartphone en QML avec l'environnement Quick 2 sous Qt 5.15 LTS.

1ère Journée

Introduction à QtQuick 2 Toolkit

- Éléments graphiques
- Élément textew
- Layout « ancrage »
- Bindings
- Handlers

Interaction avec l'utilisateur

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

Travaux pratiques

- Prise en main du Toolkit QML
- Premières applications QML tactiles

Les Composants

Les Animations

Travaux pratiques

- Mise en oeuvre d'applications graphiques QML animées simples

2ème Journée

Présentation des données

- Ordonner des éléments graphiques
- Modèles de données simples
- Les vues

Travaux pratiques

- Mise en oeuvre d'applications de présentation/saisie de données simples
- Mise en oeuvre d'applications de graphiques multi-interfaces/ multi-contrôles
- Mise en oeuvre du QtWebKit sous QtQuick 2 sur module phyCORE-i.MX 6

Rappel notions avancées C++ sous Qt 5 en vue d'être utilisées avec des composants Qt Quick

- Les Propriétés
- Metaclass et Introspection
- Mécanismes avancés des signaux/slot et QEvent et multithreading

3ème Journée

Gestion avancée des mécanismes des containers et des collections sous Qt

- QVariant
- Les containers : QVector, QList, ...
- Méthode de sérialisation d'objets

Travaux pratiques

- Implémentation de classes C++ métiers sous Qt sérialisables, dans un contexte de collections ou de signaux/slot pour une utilisation dans une architecture logiciel hybride C++/QML

Intégrer QML avec C++

- Exporter des objets C++ en QML
- Exporter des classes en QML
- Exporter des classes non graphiques
- Exporter des classes graphiques

Travaux pratiques

- Mise en oeuvre d'une application graphique QML de présentation de données sérialisées dans un fichier au travers une collection C++ Qt
- Usage et Analyse des traces du QML Profiler

Moyens pédagogiques

- Vidéo-projecteur. Tableau blanc ou paper-board.
- Un PC par binôme
- Une connexion internet

Supports :

- Kit de développement
- support de cours et travaux pratiques (version papier ou PDF)