

# Outils de conception d'interfaces graphiques – Mise en œuvre d'IHM avec Qt 5

## Présentation de la librairie Qt

- Historique de Qt et de ses version
- L'architecture de la solution Qt : QtCore, QtGui, QtWidgets, QtSql, ...
- Qt, une API multiplateformes

## Le modèle de programmation objet de Qt

- Présentation des types/fonctions/macros de base (QtGlobal)
- Gestion des chaînes de caractères en Qt
- La classe mère QObject
- Topographie des principales classes Qt
- La gestion de la mémoire sous Qt
- La réflexion avec Qt : QMetaObject et QMetaClassInfo
- Les classes de collections propres à Qt et les liens avec l'architecture de la STL
- Gestion des propriétés
- Travaux graphiques : utilisation de quelques classes QT non graphiques (collection, ...)

## Architecture d'un projet Qt

- Le fichier de build et l'outil qmake
- Les fichiers de d'interface graphique (.ui)
- Les fichiers de code Qt
- Les fichiers propre à l'internationalisation
- Travaux pratiques : génération d'un projet de type « Hello World »

## Introduction au développement d'applications graphiques

- Mise en oeuvre d'une interface graphique et utilisation de la classe QMainWindow
- Utilisation de quelques widgets simples
- Utilisation des stratégies de positionnement de Widget
- Mise en oeuvre de barres de menu, de barres d'outils et de barres de statuts
- Utilisation du mécanisme de docking (QDockWidget)
- Développement de boîtes de dialogue personnalisées (QDialog)
- Travaux pratiques : une interface graphique est progressivement enrichie en y intégrant les différents concepts présentés

## La gestion des événements en Qt

- Le modèle de slots/signaux
- La gestion du Focus
- Travaux pratiques : ajout de gestionnaires d'événement sur notre interface graphiques

## Prise en main de l'outil Qt Designer

- Le contenu d'un fichier .ui
- Génération d'une interface via Qt Designer en mode autonome
- Intégration des outils Qt dans Eclipse/CDT
- Travaux pratiques : manipulation du designer pour produire une interface graphique

## Utilisation du modèle MVC dans une application Qt

- Présentation du modèle MVC (Model/View/Controller)
- Les MVCs intégrés à Qt
- Mise en oeuvre du MVC sur un composant propre au projet
- Travaux pratiques : mise en oeuvre d'un MVC dans le cadre d'un jeu d'échec (vues texte et 2D)
- Utilisation d'un modèle liés à une base de données relationnelle
- Travaux pratiques : réalisation d'un affichage d'une table de base de données

## Aspects complémentaires

- Mise en oeuvre de thèmes graphiques et les liens avec CSS
- Gestion du drag'n drop
- Utilisations de conteneurs
- Développement de widgets personnalisés
- Gestion de l'internationalisation, de la localisation, d'unicode et de l'outils Qt Linguist
- Gestion de la sérialisation en Qt (classe QDataStream)

## Durée

5 jours / 35 heures

## Prix HT / stagiaire

2500€

## Objectifs pédagogiques

- Cette formation vous permet d'apprendre les techniques de développement d'applications graphiques modernes en utilisant la librairie Qt 5 et les outils associés.

## Niveau requis

- Avoir suivi le « module 1C : Développer avec le langage C++ » ou avoir les connaissances équivalentes

## Public concerné

- Cette formation s'adresse aux développeurs C++ confirmés.

## Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

## Moyens pédagogiques et techniques

Salles de formation (accessibles et adaptables aux besoins des personnes en situation de handicap) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD

Documents supports de formation projetés  
Apports théoriques, étude de cas concrets et exercices

Mise à disposition en ligne de documents supports à la suite de la formation

Dispositif de suivi de l'exécution de l'évaluation des résultats de la formation

Feuilles d'émargement (signature électronique privilégiée)

Evaluations formatives et des acquis sous forme de questions orales et/ou écrites (QCM) et/ou mises en situation

Questionnaires de satisfaction (enquête électronique privilégiée)

## Programmation multi-tâches en Qt

- Pourquoi faire de la programmation multi-tâches ?
- La notion de Thread en Qt (classe QThread)
- Synchroniser les accès concurrents
- Problématiques synchronisations/performances
- Utilisation de timers

## Introduction au développement d'applications graphiques

- Mise en oeuvre d'une interface graphique et utilisation de la classe QMainWindow
- Utilisation de quelques widgets simples
- Utilisation des stratégies de positionnement de Widget
- Mise en oeuvre de barres de menu, de barres d'outils et de barres de statuts
- Utilisation du mécanisme de docking (QDockWidget)
- Développement de boîtes de dialogue personnalisées (QDialog)
- Travaux pratiques : une interface graphique est progressivement enrichie en y intégrant les différents concepts présentés

## La gestion des événements en Qt

- Le modèle de slots/signaux
- La gestion du Focus
- Travaux pratiques : ajout de gestionnaires d'événement sur notre interface graphiques

## Prise en main de l'outil Qt Designer

- Le contenu d'un fichier .ui
- Génération d'une interface via Qt Designer en mode autonome
- Intégration des outils Qt dans Eclipse/CDT
- Travaux pratiques : manipulation du designer pour produire une interface graphique

## Utilisation du modèle MVC dans une application Qt

- Présentation du modèle MVC (Model/View/Controller)
- Les MVCs intégrés à Qt
- Mise en oeuvre du MVC sur un composant propre au projet
- Travaux pratiques : mise en oeuvre d'un MVC dans le cadre d'un jeu d'échec (vues texte et 2D)
- Utilisation d'un modèle liés à une base de données relationnelle
- Travaux pratiques : réalisation d'un affichage d'une table de base de données

## Aspects complémentaires

- Mise en oeuvre de thèmes graphiques et les liens avec CSS
- Gestion du drag'n drop
- Utilisations de conteneurs
- Développement de widgets personnalisés
- Gestion de l'internationalisation, de la localisation, d'unicode et de l'outils Qt Linguist
- Gestion de la sérialisation en Qt (classe QDataStream)

## Programmation multi-tâches en Qt

- Pourquoi faire de la programmation multi-tâches ?
- La notion de Thread en Qt (classe QThread)
- Synchroniser les accès concurrents
- Problématiques synchronisations/performances
- Utilisation de timers