

# Qt: introduction

François Delobel

Master TechMed et Setsys, Université Clermont-Auvergne

- 1 Contexte
  - Apports techniques
  - Historique et Business Model
  - Portabilité
- 2 Outils
- 3 Panorama des technologies
  - QtQuick, QML, Qt, C++, Javascript
  - Interactions entre les technologies
- 4 Motivations

# Présentation

## François Delobel

- ❑ IUT - département info / Limos
- ❑ Bureau C30 (2ème étage)
- ❑ francois.delobel@uca.fr
- ❑ J'aime les questions!

## Supports de cours

- ❑ `https://codefirst.iut.uca.fr/git/francois.delobel/qt\_master.git`



# Programmation Graphique en Qt pour Master TechMed / Setsys

## Nouveau cours cette année!

- ❑ Motivation : vous donner des bases pour être capable de développer un client graphique lourd pour PC ou matériel médical.
- ❑ Besoin de bases en C++: ne pas hésiter à poser des questions pour réviser/étendre.
- ❑ Vous essayez les plâtres mais ce sera plus adaptable à vos besoin / votre niveau.

## Organisation du cours

- ❑ Mélange Cours / TP (plus de cours au début).
- ❑ Questions bienvenue \*\*pendant\*\* le cours. Pas la peine d'attendre de décrocher.
- ❑ Au début, des TP guidés, après un mini projet (en solo).

# Qt et ses principaux modules

- Ensemble de bibliothèques permettant (QtEssentials):
  - ▷ Graphisme (Widgets, Graphics ou QML), gestion de l'OpenGL
  - ▷ Réseau (TCP, UDP, HTTP, HTTP/2, SSL, bluetooth...)
  - ▷ Interfaçage avec bases de données (SQL)
  - ▷ Threads (portables)
  - ▷ Multimédia (Son, vidéo, photo...)
- Et dans les nombreux Qt add-ons...
  - ▷ Bluetooth, NFC, GPS, Sensors
  - ▷ D-Bus (Bus logiciel), ActiveX/Com, KNX, MQTT
  - ▷ SerialPort, WebSockets (Full duplex, communications Browser/server tunnelées)
  - ▷ XML (XPath, XQuery, XSLT, SCXML...)
  - ▷ Charts, Data Visualization (3D)

# Qt, ses ajouts à C++

## Qt ajoute des fonctionnalités au langage C++

- ❑ Ajoute un modèle événementiel (signaux et slots).
- ❑ Enrichissement de toute classe `QObject` à la compilation
  - ▶ Ajoute beaucoup d'introspection
- ❑ Qt est *plus* qu'un framework.

## Qt et QObject

- ❑ Gestion automatique de la mémoire par compteurs de références
- ❑ Identification des instances
- ❑ Métaclasse

# Qt: Historique et Business Model

- Naissance en 1991, chez TrollTech (Norvège)
  - ▷ Système de double licence (GPL/Commerciale)
  - ▷ Base de KDE
  - ▷ Multi-Os (Unixes, Windows, puis Mac)
  - ▷ Développement de la partie embarquée
- TrollTech Racheté en 2008 par Nokia
  - ▷ Développement de MeeGo pour remplacer Symbian
  - ▷ Téléphone N9 succès
- Détournement de Nokia par Microsoft
  - ▷ Officieux en 2011 (Windows Phone exclusif), officiel en 2013, Effectif en 2014
  - ▷ Qt cédé à *Digia* en 2012, devient *Qt Group* en 2016
- Qt Group (300 employés)
  - ▷ Toujours *Dual licencing*, et extension du nombre de plateformes
  - ▷ Développement partie embarquée, et info scientifique/industrielle

# Qt : Multi-OS, multi-architecture

## Portabilité

- Généralistes:
  - ▶ Officielles: Linux(Wayland et X11), Windows, MacOS.
  - ▶ Externes: Unices (Sailfish, BSD, OpenSolaris) , Haiku, Amazon Kindle DX, Tizen, webOS...
- Téléphones/Tablettes (Officielles)
  - ▶ Android, iOS/tvOs/watchOS, *BlackBerry 10(QNX)*, *WindowsCE*, *Windows-RT/Windows Phone...*
  - ▶ SailfishOS: OS de Jolla (ex Nokia): Full Linux sous Wayland avec Qt.
- Temps réel (industrie)
  - ▶ QNX, VxWorks, Integrity (EAL6)



# Développement: Environnement logiciel (SDK)

- ❑ QtCreator: IDE. Appuyé sur clang code model pour refactor, coloration...
- ❑ QtDesigner: conception d'UI. Peut être intégré à QtCreator
- ❑ QtAssistant: browser de doc Qt
- ❑ QtLinguist: assistant de traduction
- ❑ QMake: générateur de `makefile` propre à Qt. Remplacement progressif par `cmake`.
- ❑ Bonne cohabitation avec clang(d|-tidy|-modernize), Valgrind (débug mémoire), gdb (debug).
- ❑ Documentation en ligne: <http://doc.qt.io/>
- ❑ Non utilisés pour le cours: 3D Studio (création contenu/interfaces 3D).

## Développement: Contexte à l'université

- Développement sur les PC dual boot de l'université.
- Ancienne version de Qt: Qt 5 mais compatible dans l'ensemble avec Qt6.
- Possibilité de développer sur vos PC. Mais toujours plus facile d'installer un SDK sous Linux :)

## Définitions – suite –

- ❑ *QtQuick*: couche graphique Qt récente (par oppositions aux QWidgets obsolète), qui utilise du QML.
- ❑ *QML*: Qt Markup Language. Langage de description d'UI pour QtQuick. Ressemble fortement à du JSON.
- ❑ *JSON*: Javascript Object Notation. Langage de description d'objets peu verbeux et complet. Inspiration du QML et langage de sérialisation.
- ❑ Javascript: langage interprété ou compilé à la volé utilisé en QtQuick pour programmer la logique de l'UIX.
- ❑ C++: langage utilisé pour coder la couche métier d'une application en QtQuick, ou toute une application en Qt avec widgets.

# Interface graphique et événements

## Description des interfaces

- Écrit en QML: langage de *description* qui met en place des *éléments* graphiques.
- Tout élément a des propriétés.
- Forme un graphe de scène affiché très rapidement.
- Doit pouvoir être écrite par un non informaticien.

## Logique de comportement des interfaces

- Codée en Javascript. Interactions QML/Javascript.
- Peut être dans des fichiers Javascript indépendants ou directement dans le QML si c'est simple
- Réservé aux opérations simples et peu coûteuses.
- Doit pouvoir être écrite par un non informaticien.

# Couche métier

- ❑ Fait les calculs importants.
- ❑ Codée en C++ pour économies CPU/énergie.
- ❑ Possède des *slots* (méthodes) qui peuvent être appelés par le lancement de *signaux* (événements).
- ❑ Les slots de C++ peuvent être appelés depuis le Javascript de l'interface.
- ❑ Le QML et le C++ utilisent des propriétés pour échanger simplement des valeurs.

# Objectifs du cours et évaluation

À la fin de ce cours, vous devriez:

- ❑ Savoir développer une application graphique en Qt.
- ❑ Connaître (en partie au moins) les langages suivants:
  - ▶ C++ → couche métier, développement de bibliothèque, concepts fondamentaux à d'autres langages
  - ▶ Qt, QML → Développement multi-plateformes
  - ▶ Javascript → logique de l'UI, peut servir en devpt web, langage de script de certaines applications

Évaluation :

- ❑ Sûr: exam final (dernier cours)
- ❑ Possible: noter mini projet (avancement sur une grosse semaine?)
- ❑ Possible: exam machine