Qt – a Framework for the Multi-Screen World

Nils Christian Roscher-Nielsen
The Qt Company
Who am I?

• Nils Christian Roscher-Nielsen
  • 6 years with Qt
    • (Trolltech)
    • Nokia
    • Digia
    • The Qt Company
  • Support Engineer
  • Sales Engineer

• Norwegian University of Science and Technology
  • M.Sc. Engineering Cybernetics
Content of this presentation

• Industry challenges
  • Where are we, and where are we going
  • Automotive, Automation, Mobile

• Technology Challenges
  • Why is Qt a good idea?
  • Will it be so, also in the future?

• Future opportunities
Four Concepts for the Multiscreen World

• Power of C++
  • Implement anything. Now.

• QML
  • Native performance. Your Look and Feel.

• Hybrid Application development
  • Deliver the same content, everywhere.

• Interconnectivity
  • Beauty in the Front. Logic in the Back. Data in the Cloud.
Application development for desktop and mobile

Embedded device creation HMI’s and applications

Software development kits (SDK) for third parties
Industry Challenges
A view on the Industry History

• Siloed development efforts
  • Multiple teams working on different parts

• Long development cycles
  • 3-5 years from need is identified to the product is released.

• Highly optimized for niche markets
  • Various segments, HW differentiation

• Big changes ahead
Set Top Box market

- No new large innovations at IBC, NAB
- Technology is now mature, and products become reality

- Comcast RDK
- FROG by Wyplay
- Hybrid products from DCC Labs

- Bring your own device
Technical Enablers
Power of Qt/C++

- High level cross platform APIs
- Interface with other libraries
- Write and interface with device drivers, communication busses, etc.

- Modern
  - C++11, C++14, VS2014, clang, etc.

- Best in class C++ API’s
- Signals and Slots
- Full Introspection
Stable application platform

- Medical
- Automotive
- In-flight systems
QML

• New HW possibilities require new tools

• Declarative UI creation

• Rapid prototyping and device creation

• Collaborative Designer-Developer workflow

• Unlock unparalleled graphics performance
QML

• Native Look and Feel

• Custom UI design
Native and Web Hybrid Development

• Qt WebEngine to the rescue!

• Fully integrated graphics stack.

• Tap into the best of native performance, and the broad set of HTML5 ecosystem and apps available
Practical Example on multiple platforms

Qt graphics underneath the web page

Actual web contents with Web Engine

Qt graphics overlaying web content

OpenGL shaders applied to web contents (colorize + 3D page curl)

Native Qt UI elements to interact with web content
Interconnectivity

• Beauty in the Front. Logic in the back. Data in the Cloud!

• Enginio Data Storage
  • Access your data
    • Anywhere
    • Anytime
    • From any device

• Qt WebChannel
• Qt Managed WebSocket (MWS)
• Qt Managed Application Runtime (MAR)
Going Multi-Screen with your Device
Going Multi Screen with your Device

GUI device

Headless device

Headless device

© 2014 The Qt Company
Going Multi-Screen with Your Device
Going Multi Screen with your Device

Qt Cloud Services

GUI device

Web page

Qt Cloud REST API

Desktop PC

Mobile device 1

Embedded device N

Headless device

Qt Cloud API

Qt Cloud API

Qt Cloud API

Qt Cloud API
Re-Using Code with Responsive User Interface

Tablet UI, portrait

Tablet UI, landscape

Mobile UI

Desktop UI

Qt Application

© 2014 The Qt Company
Optimal Re-Use-Structure of a Qt Multi-Screen Application

Tablet UI

Mobile UI

Desktop UI

Embedded Device UI

Common UI (Qt Quick)

Application Logic (Qt/C++)

Qt

© 2014 The Qt Company
Extending your Ecosystem
Qt as Technology Platform
Qt Creator

- Stable plugin architecture
- Many interesting customer projects
- Develop hybrid applications and UI’s
- Emulator that enables desktop and device development on the same machine.
And where are we headed?

• Always forwards!

• New mobile OS’es come and go every year.
• The Desktop and Mobile is seemingly merging
• Industrial applications look like app store games

• Continue to drive the state of the art UI development

• Every screen wants to have the same content and UX as a mobile
• No one screen size or platform suffers
Better Life Cycle Management

• Historically, shipping a Qt application has been a complicated task

• **Goals**
  1. Simplify dependencies and packaging steps for Qt 5 applications (Framework tasks)
  2. Provide wizard / automation tools to further improve workflow (Tooling tasks)

• Distribution Channel Independent

• As one potential distribution method the Qt Installer Framework will be integrated into Qt Creator
Summary

- Comprehensive application framework
- Rich set of application building blocks
- Build advanced user interfaces, faster

- Support multiple platforms – desktop and embedded - from a single source
- Boost productivity through integrated tools
- Rely on professional service and support

- Keep control of your value chain and monetization
Thank you!

www.qt.io