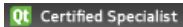


Porting Qt to a New Smartphone for Fun and Fame

Jarosław Staniek • Tomasz Olszak • @QtForTizen

 Qt Certified Specialist

 Qt Ambassador

 Qt Certified Developer

License: CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0>

 SmartDevCon²

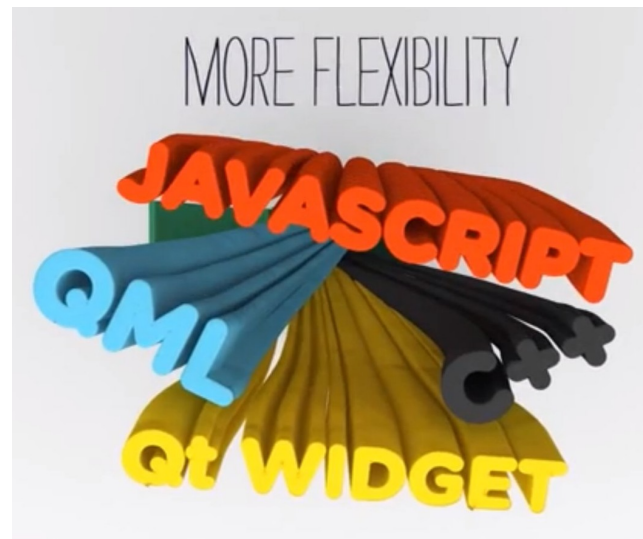
12-14th of September 2013 - Katowice, Poland

Agenda

- About Qt
- What makes a successful Qt port?
- What to port?
- How to port?
- QML & Qt Quick
- Tizen use case
- How easy?

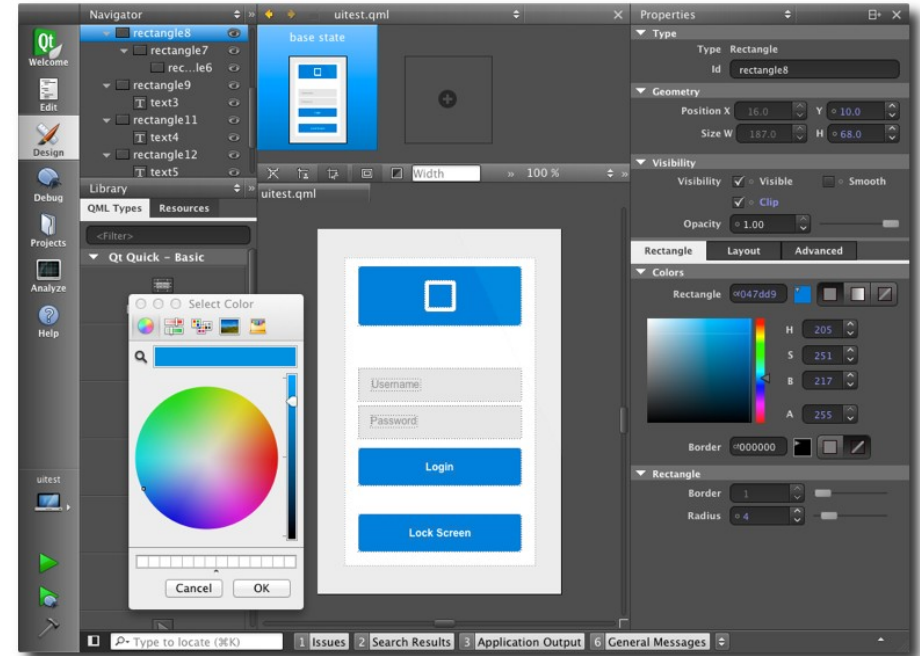
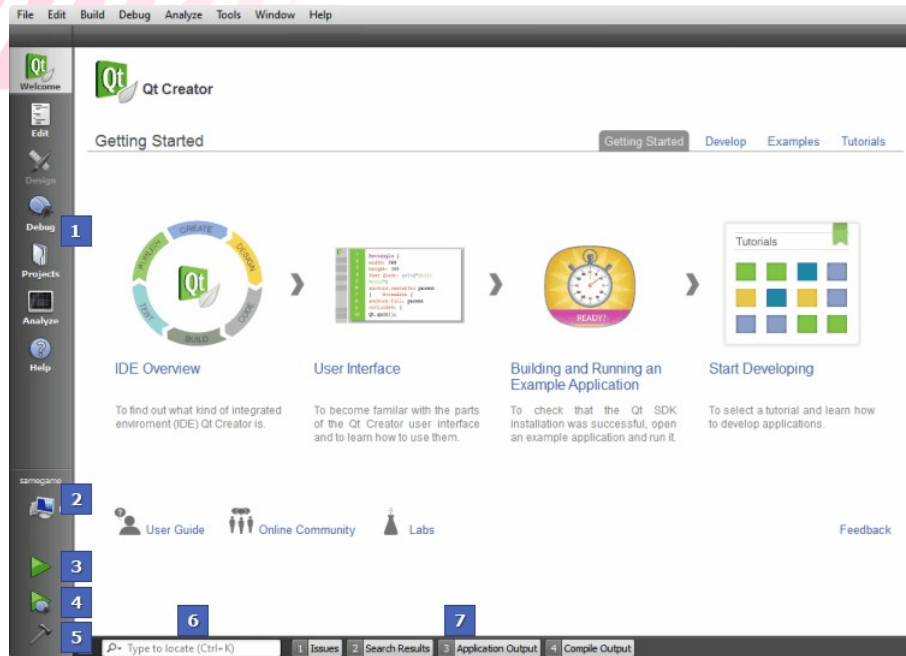
What's Qt?

- Popular app/UI framework & tools
- Use it with C++, QML, JavaScript or... ~20 other languages



What's Qt?

- Qt Creator forms the Qt IDE



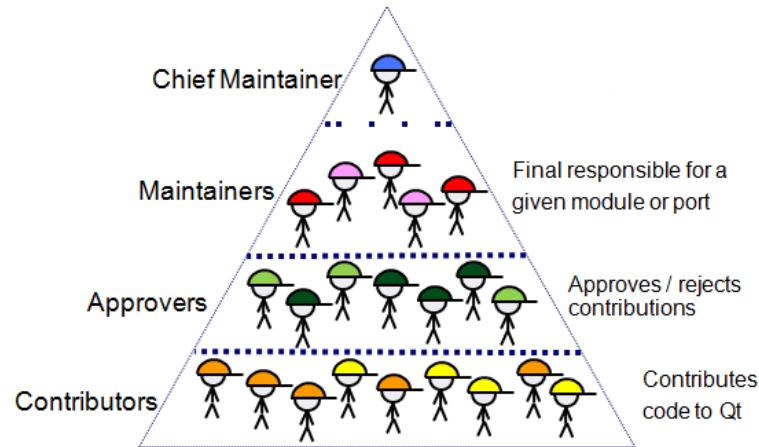
Platforms supported by Qt5

Platforms	External (community) ports
Windows	OpenSolaris
OS X	Haiku
X11	OS/2
Embedded Linux	webOS
Wayland	Amazon Kindle DX
QNX/ Blackberry 10	Mir (Ubuntu)
Android	AmigaOS
iOS	Tizen
VxWorks	Sailfish OS



Who makes Qt?

- **Community** through **openly governed**, inclusive, meritocratic development model



- **Open Source** (LGPL, GPL) for any use
- Optional broad **commercial support/licensing**

Recent history of Qt

2005	4.0 release
2008	Nokia acquires Qt
2009	Qt 4.5 goes LGPL
2011	Qt Project starts (open governance)
2012	Digia acquires Qt business
Dec 2012	5.0 release
July 2013	5.1 release

(1st release in 1995)

Community-supported devices

- **blackberry-playbook-armv7le-qcc**
- **blackberry-playbook-x86-qcc**
- **linux-archos-gen8-g++**
- **linux-arm-amlogic-8726M-g++**
- **linux-arm-trident-pnx8473-g++**
- **linux-beagleboard-g++**
- **linux-imx53qsb-g++**
- **linux-imx6-g++**
- **linux-maemo-n9-g++**
- **linux-mipsel-broadcom-97425-g++**
- **linux-rasp-pi-g++**
- **linux-sh4-stmicro-ST7108-g++**
- **linux-sh4-stmicro-ST7540-g++**
- **linux-snowball-g++**
- **linux-tegra2-g++**
- **linux-g++-tizen-mobile**
- **linux-g++-tizen-ivi**



ONE FRAMEWORK TO RULE THEM ALL

What makes a successful Qt port?

You want to:

- build/run/debug
console apps & GUI apps
- use Qt Creator for that
- target device and simulator

What makes a successful Qt port?

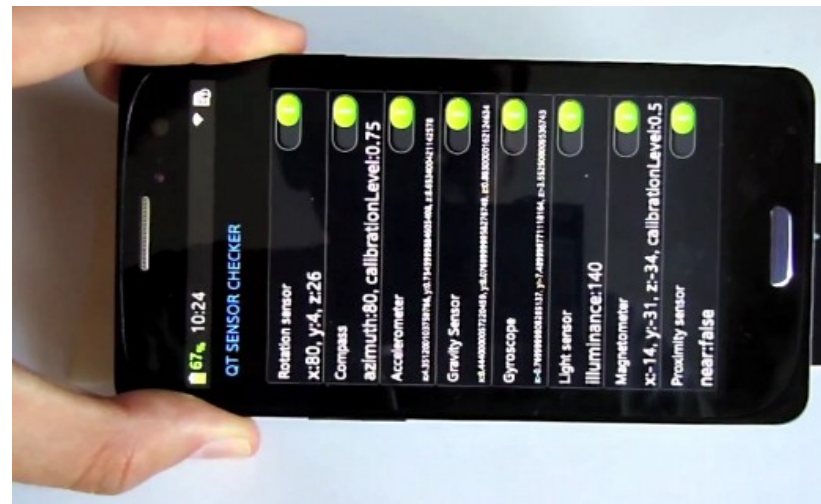
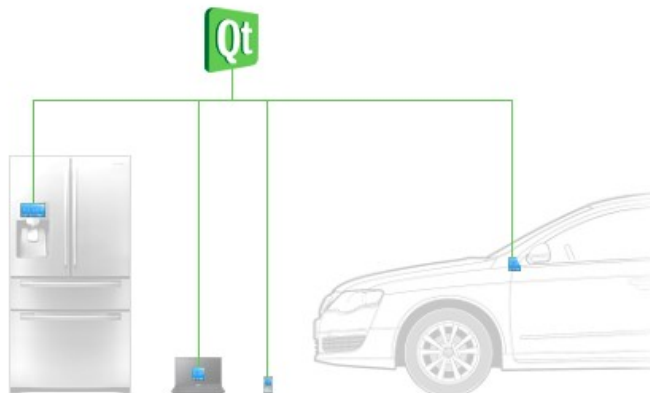
Fine-tuned **native** look & feel for apps



What makes a successful Qt port?

Handle & simulate:

- mobile phone sensors
- middleware like message broker (car IVI...)
- device controls (set-top-box, TV, refrigerator)



What to port?

Qt frameworks

Qt tools

+ Expose platform specifics for Qt apps

Device-specific porting

Depending on:

- CPU architecture
- Compiler
- Operating system
- Device form factors
 - (resolution, screen size, phone, tablet, large displays, tv, set top box, ivi)



EN™

TIZ

with

How to port?

Now

t

Q

How to port Qt frameworks?

- Define **build specifications** for Qt
- Use **QPA** (Qt Platform Abstraction) for platform features
- Use **QStyle** APIs for native QWidgets look
- Or use **Qt Quick Controls** APIs for native Qt Quick look

How to “port” Qt tools/integration?

- Define platform’s **kit** for Qt Creator
- Implement plugin(s) for Qt Creator:
 - communication with device(s)
 - app deployment/packaging
 - simulator

How to “port”?

QPA: Qt Platform Abstraction

- In Qt 5 replaces QWS (Qt Windowing System from Qt 4.x for Embedded Linux)
- Defines device/OS abstraction layer for platform specifics & window system
- Set of C++ interface classes

EN™

TIZ

with

QML & Qt Quick

Now

t

Q

Qt Qml

- Implements the QML language & engine
(*Qt Meta/Modelling Language*)
- Offers extensibility APIs

Qt Quick 2

- Built on top of **Qt Qml**, adds:
 - types for GUI app development
 - visual canvas, states, transitions, effects, animations, user input, model/view



Qt Quick 2

- Requires  &  2
- Recommended for mobile/embedded instead of QtWidgets/QStyle



- **OpenGL** for **Embedded Systems**
- a subset of the OpenGL 3D graphics APIs known on desktops

EN™



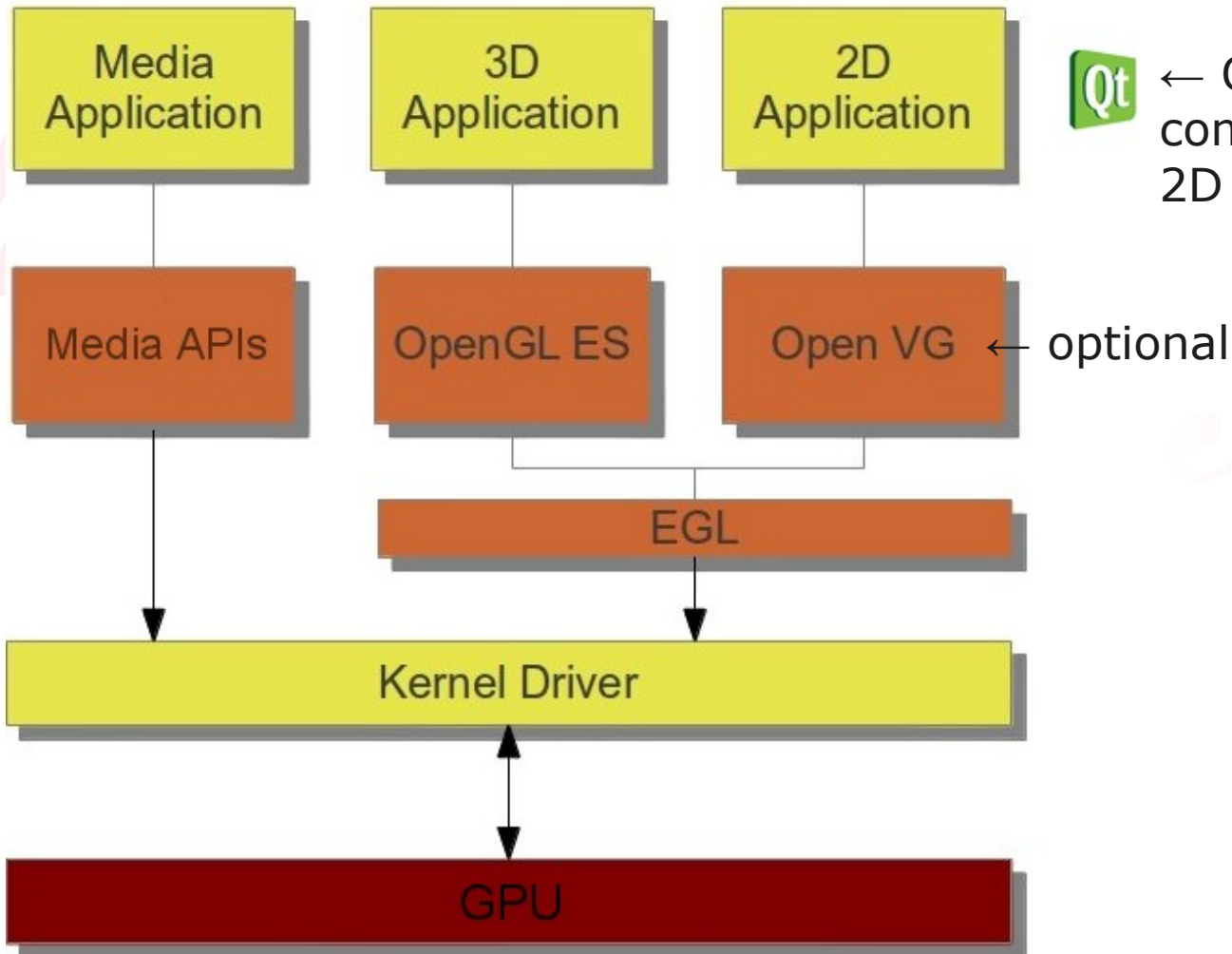
TIZ

- **Embedded-System Graphics Library**
- Accelerated, 2D/3D rendering
- Also used by Android, Wayland, Mesa 3D, Mir, SDL...

t

Q

How it plays together?



← Qt apps can combine Media, 2D and 3D features

← optional

Porting Qt to a new platform: Tizen use case

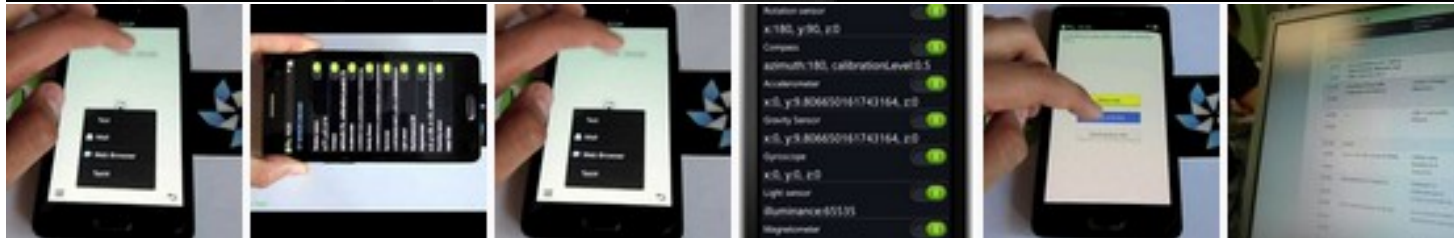
Based on:



Qt for Tizen
@QtForTizen

A community-driven port of Qt 5 to #Tizen smartphones, cars (IVI), home appliances and PC/netbook/notebooks.

qt-project.org/wiki/Tizen



Porting Qt to a new platform: Tizen use case

Step 1/6: Make it build & run console apps

- Use platform's toolchain
- Build Qt Base module with minimal platform plugin
(no OpenGL, EGL dependencies)
- Fix issues with platform header and library paths, etc.

Porting Qt to a new platform: Tizen use case

Step 2/6: Make it build & run GUI apps

Look at the platform windowing system:

- Is it X11-based? Adapt existing Qt xcb platform plugin
(Tizen mobile case)
- Is it Wayland-based? Adapt existing Qt Wayland plugin.
(Tizen IVI case)
- Without WS? Adapt the Qt EGLFS (EGL fullscreen) plugin
- Other: write your own plugin

Porting Qt to a new platform: Tizen use case

Step 3/6: Integrate with platform middleware:

This can be exposed to Qt APIs:

- Sensors: accelerometer, proximity sensor, light sensor, compass sensor, magnetic sensor...
- Support screen-orientation-change events
- Input method (maalit, iBus, SCIM...)
- Application life cycle and hardware buttons
- Language change events, location/GPS, NFC, color palette, low battery warnings, system notifications...

Porting Qt to a new platform: Tizen use case

Step 4/6: Native look & feel for Qt apps

- Find platform UI/UX guidelines, graphics, icons, color palettes
- Use that in styles compliant with the Qt Quick Controls APIs
- Result: native look & feel, code portability to other platforms
- Add platform-specific set of controls if standard Qt Quick Controls are not sufficient

Porting Qt to a new platform: Tizen use case

Step 5/6 Optimizations

- Add **preloading** process to boost startup of Qt apps.

(similar to MeeGo Harmattan's booster
and Android's zygote)

Porting Qt to a new platform: Tizen use case

Step 6/6: Integrate with Qt Creator

- Add QtCreator plugin for making deploying easier for app developers
- Soon to be released in Alpha 4

Porting Qt – How easy?

quickcontrols-tizen module

Language	Files	Lines of code
QML	39	1870
Javascript	3	1758
Bourne Shell	3	100
IDL(pro)	3	93
C++	2	580
C/C++ Header	2	27
Total	52	3906

qtsensors module

Language	Files	Lines of code
C++	10	685
C/C++ Header	9	173
IDL(pro)	1	26
Total	20	884

Thanks!

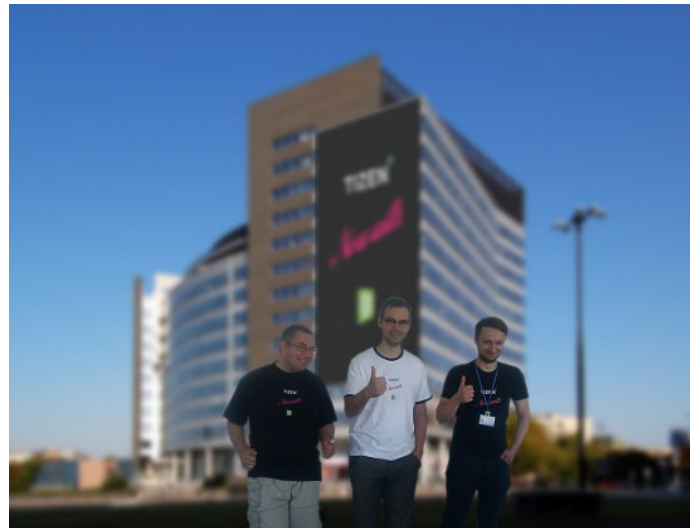
Questions welcome!

TIZEN^{OS}

Now with



Join us @ <http://qt-project.org/wiki/Tizen>



References

- <http://qt-project.org/wiki/Tizen>
- [http://en.wikipedia.org/wiki/Qt_\(framework\)](http://en.wikipedia.org/wiki/Qt_(framework))
- <http://my.safaribooksonline.com/0131872494/pref04?portal=oreilly>
- <http://qt-project.org/wiki/Qt-Platform-Abstraction>
- <http://qt-project.org/videos/watch/qpa-the-qt-platform-abstraction>
- <http://qforever.wordpress.com/2012/04/10/qt-platform-abstraction-starter-guide/>
- <http://qt-project.org/doc/qt-5.0/qtqml/qtqml-cppintegration-definetypes.html>
- <http://qt-project.org/doc/qt-5.0/qtquick/qtquick-index.html>
- [http://en.wikipedia.org/wiki/EGL_\(OpenGL\)](http://en.wikipedia.org/wiki/EGL_(OpenGL))
- http://en.wikipedia.org/wiki/OpenGL_ES
- <http://doc-snapshot.qt-project.org/qt5-stable/qtquickcontrols/qtquickcontrols-overview.html>
- <http://doc-snapshot.qt-project.org/qt5-stable/qtquickcontrolsstyles/qtquickcontrolsstyles-index.html>