



# MODERN USER INTERFACES FOR C++

Milian Wolff / KDAB

# OUTLINE

- Introduction
- Widgets
- QML / Quick
- Essentials & Addons
- Misconceptions

# 1. INTRODUCING QT

# THE C++ UI FRAMEWORK

- 20+ years of innovation
- thousands of developers

# CROSS PLATFORM

- Windows
- Mac OS & iOS
- Linux & Android
- QNX
- ...

# WHY I LIKE QT

- C++
- simple, consistent API
- excellent documentation
- open governance
- versatile
- practical

# WHAT I DISLIKE ABOUT QT

- code legacy
- performance traps
- lack of work with isocpp

# 2. QT WIDGETS

KMail

<https://projects.kde.org/kdepim>

mailstorage/qt-devel - KMail

File Edit View Go Folder Message Tools Settings Help

New Print Check Mail Reply Forward Previous Next Move to Trash Spam Ham Create To-do

Inbox (kdb) inbox (Local Folders) Inbox (mailstorage) foss bugs (mailstorage) kdevelop (mailstorage) (3)

Search

Name	Unrea	Total
kde-pim		626
kde-promo		2308
kdevelop	3	13725
kdevelop-users		1427
kgraphviewer		143
kommerz		186
kwrite		250
libunwind		329
massif-visualizer		95
milianw.de		
ml-adminn		3066
perf		1050
posteingang_bis_		9196
qt-creator		2434
qt-devel		10992
quanta		731
review		1124

Message Search...<Alt+Q> More...

Today

- [Development] HEADS-UP: Qt 5.6 feature freeze and branching coming Heikkinen Jani development@qt-project.org Today 12:55:12
- [Development] QDateTime is missing shared null optimization Milian Wolff Qt Development Friday 17:20:02
- [Development] Found wince info in Gerrit ,which makes me panic Gunnar Roth development@qt-project.org Friday 18:40:02
- [Development] Qt 5.5 for Android - More Regressions Since 5.3 Robert Iakobashvili development@qt-project.org 06.07.2015 08:49
- [Development] 5.5 CI having problems on tst\_qmimedatabase-xml Thiago Macieira development@qt-project.org 13.07.2015 23:48

Saturday

- [Development] Backwards compatibility break in Qt 5.5 Niklai Marchenko development@qt-project.org 27.07.2015 00:25

Friday

Thursday

Wednesday

Tuesday

July

June

May

April

March

February

January

December 2014

November 2014

October 2014

September 2014

August 2014

July 2014

June 2014

[Development] QDateTime is missing shared null optimization

From: Milian Wolff <milian.wolff@kdab.com> (KDAB (Deutschland) GmbH&Co KG, a KDA Group company)  
To: Qt Development <development@qt-project.org>  
Date: Friday 17:20:02

Spam Status: Spamassassin

**Signed by milian.wolff@kdab.com.**  
Hello all,  
I just realized that QDateTime in Qt 5 is still not implementing the "shared null optimization" - I'm not sure it's the right term.  
What I mean is that you'll incur a heap allocation when doing QDateTime::invalidDate;  
See:  

```
QDateTime::QDateTime()
: d(new QDateTimePrivate)
{}
```

Most other classes in Qt are cheap to generate in an empty/invalid state, not so for QDateTime. Is there a reason for that, or is it just an oversight? If so, I'd like to amend that. Would it be preferred to introduce a shared null like QTypedArrayData::sharedNull, or should I rather use a nullptr to indicate the invalid state, like in QImage? Or am I missing something and these tricks are not applicable to QDateTime?  
Grepping for QSharedDataPointer, I find many more classes that do not leverage a shared null state. Should we try to implement some helper functions/macros there which can be leveraged by every user of this class to implement a shared null state?  
Bye  
--  
Milian Wolff | [milian.wolff@kdab.com](mailto:milian.wolff@kdab.com) | Software Engineer

Spotify Premium – Linux Preview

Bestand Bewerken Beeld Afspeelen Help

Zoeken

Bladeren  
Ontdekken  
Radio  
Volgen  
Toplijsten  
Berichten  
Afspeelwachtrij  
Apparaten  
App Finder  
3voor12  
Billboard Top Charts  
Blue Note  
Classify  
The Complete Collection  
Digster  
Domino  
The Guardian  
Jazzify  
Last.fm  
The Legacy Of  
Moodagent

Feeling Good  
Nina Simone

Jazz en blues

ALLE SUBGENRES ▾

AFSPEELLIJSTEN

FROM A SMOKEY BAR  
From a Smokey Bar  
Enjoy this jazz and blues and start imagine yourself in a smokey bar. Time traveling to...

LATE NIGHT JAZZ  
Late Night Jazz  
The perfect audio backdrop to an evening spent chilling out with friends or relaxing in...

WOMEN OF JAZZ  
Women of Jazz  
Trace the history of jazz from timeless classics to contemporary favorites...

JAZZ CLASSICS  
Jazz Classics  
The top list of jazz classics. Features over 8 hours of unforgettable cuts...

Lowlands Festival [offi...  
5.537 VOLGERS  
VOLGEN

Paradiso heeft geluisterd naar  
Backstreet Lovers  
The Crookes

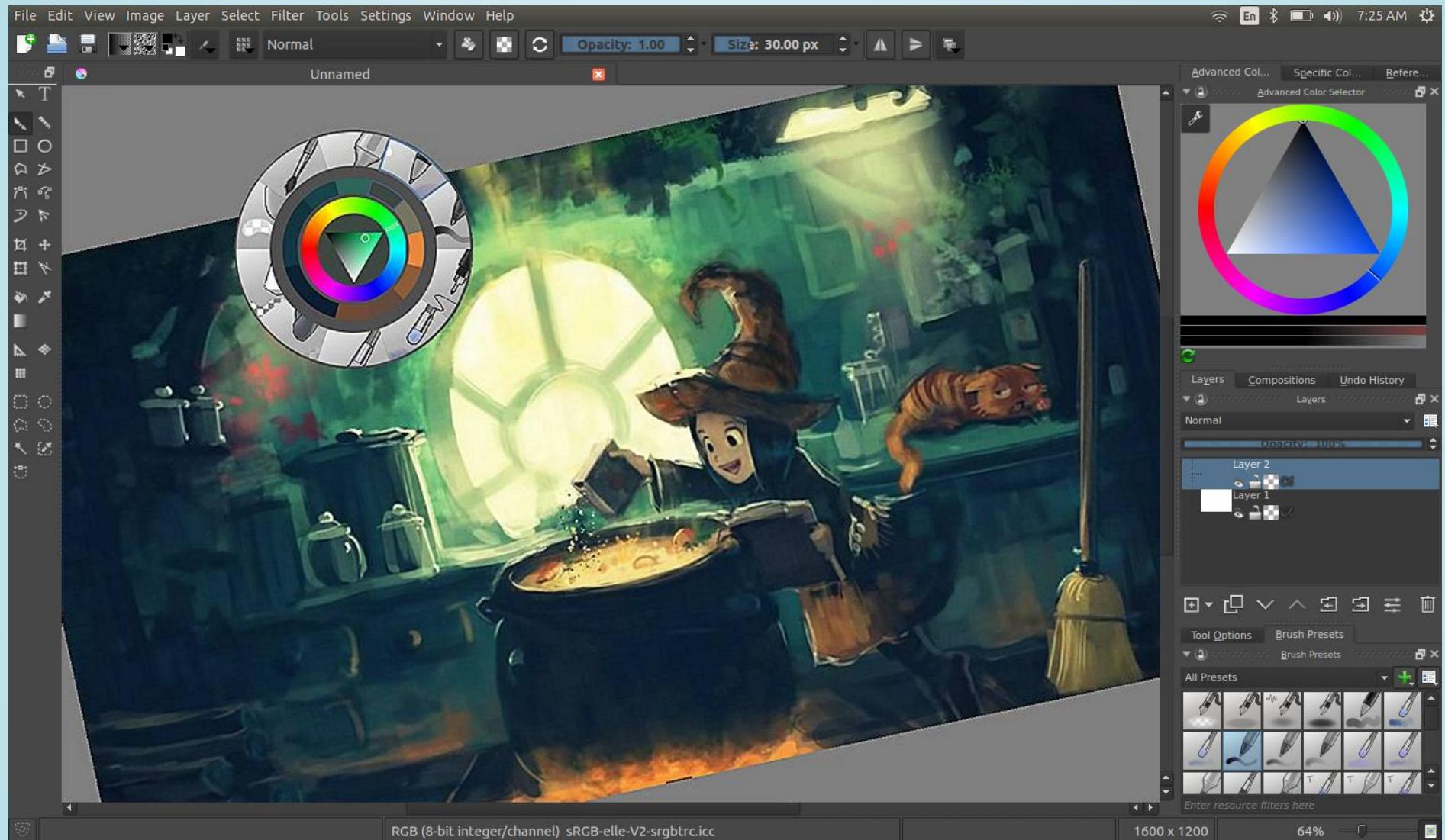
Jay Johnston heeft geluisterd naar  
C.I.D. in Uruk  
Amon Düül II

Paradiso heeft geluisterd naar  
[Won't Be] Coming Home  
Robert Cray

Spotify USA heeft New Releases on Spotify - US bijgewerkt  
Eat for Free  
Haley Bonar

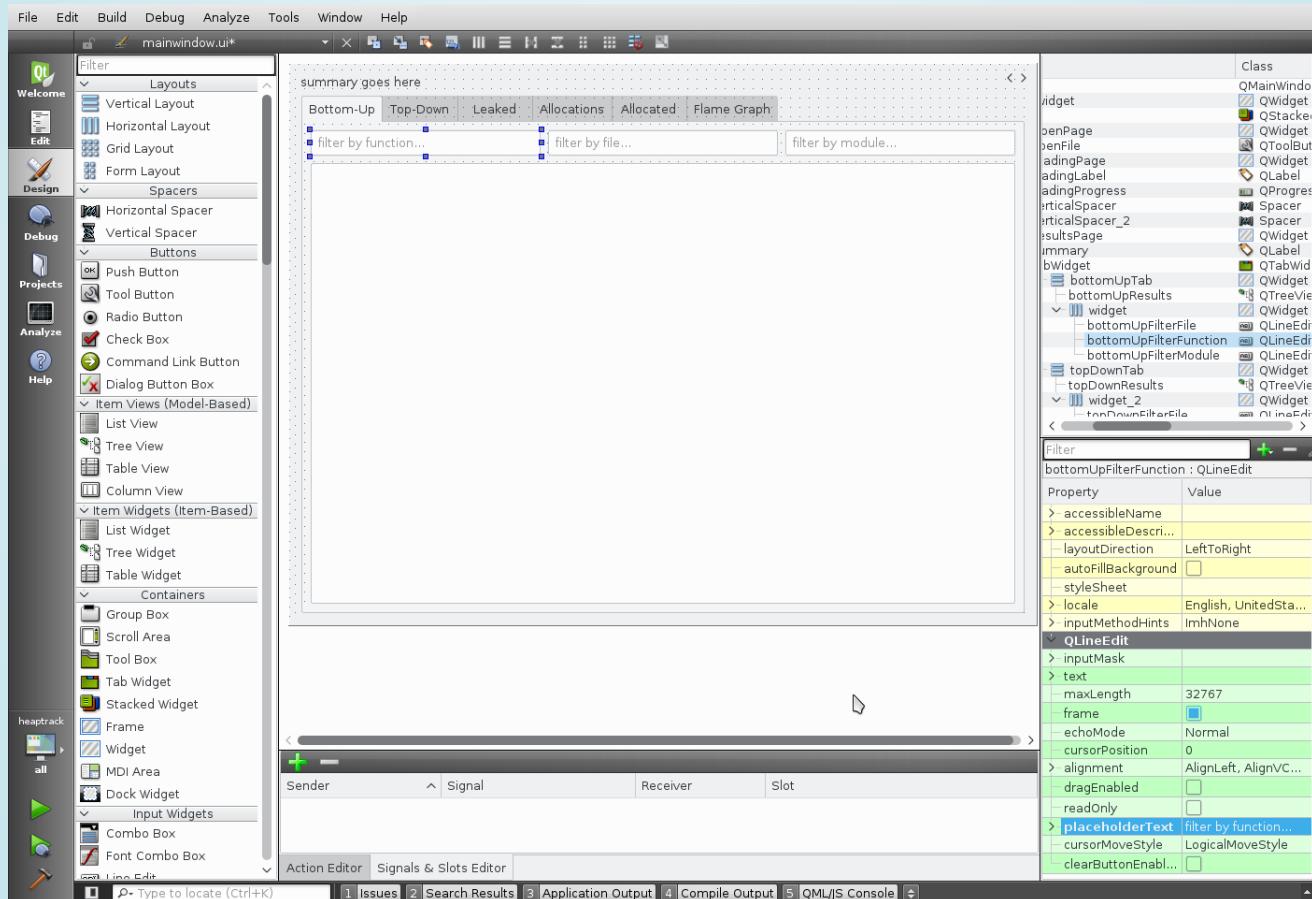
0:11 2:57

Spotify on Linux



Krita  
<https://projects.kde.org/calligra>

# USING QT WIDGETS



Qt Creator and Qt Designer  
<http://code.qt.io/cgit/qt-creator/qt-creator.git>

```
// generated by uic
#include <ui_mainwindow.h>
// part of KF5::ItemModels
#include <KRecursiveFilterProxyModel>

MainWindow::MainWindow(QWidget* parent)
    : QMainWindow(parent)
    , m_ui(new Ui::MainWindow)
    , m_treeModel(new MyTreeModel(this))
{
    m_ui->setupUi(this);

    auto proxy = new KRecursiveFilterProxyModel(this);
    proxy->setSourceModel(m_treeModel);
    m_ui->treeView->setModel(proxy);

    connect(m_ui->treeFilter, &QLineEdit::textChanged,
            proxy, &KRecursiveFilterProxyModel::setFilterRegExp);
}
```

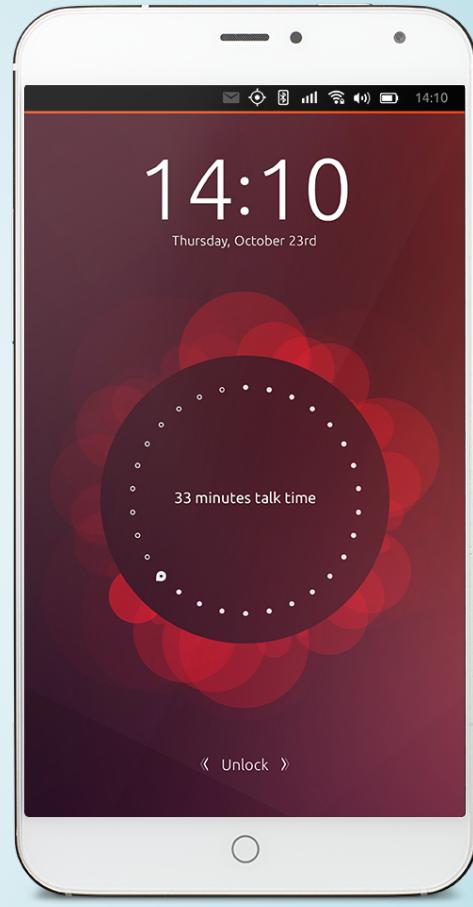
```
debuggee: kcrash  
total runtime: 2.995s  
bytes allocated in total (ignoring deallocations): 78.82 MiB (26.3 MiB/  
calls to allocation functions: 418513 (139737/s)  
peak heap memory consumption: 18.4 MiB  
total memory leaked: 1.2 MiB
```

Bottom-Up	Top-Down	Leaked	Allocations	Allocated	Flame Graph
sonnet		filter by file...			filter by module...
Allocations [-]	Peak [B]	^	Leaked [B]	Allocated [B]	Location
✓ 9	11248056	0	16872084	allocate_nodes in ?? (/usr/lib/libhspell.so.0)	
✓ 9	11248056	0	16872084	read_dict in ?? (/usr/lib/libhspell.so.0)	
✓ 9	11248056	0	16872084	hspell_init in ?? (/usr/lib/libhspell.so.0)	
✓ 3	5624028	0	5624028	0x7fd51fe2073 in ?? (/usr/lib/qt/plugins/kf5/sonnet/hspell.so)	
✓ 3	5624028	0	5624028	0x7fd51fe153d in ?? (/usr/lib/qt/plugins/kf5/sonnet/hspell.so)	
✓ 3	5624028	0	5624028	Sonnet::Loader::loadPlugin(QString const&) in /home/milian/projects/kf5/src/frameworks/sonnet/src/core/loader.cpp:289 (/ho...	
> 6	5624028	0	11248056	HSpellDict in /home/milian/projects/kf5/src/frameworks/sonnet/src/core/loader.cpp:265 (/home/milian/proj...	
> 117966	4912314	20387	14007140	QArrayData::allocate(unsigned long, unsigned long, unsigned long, QFlags<QArrayData::AllocationOption>) in ?? (/usr/lib/lib...	
✓ 49150	2343968	0	2345290	HashMgr::add_word(char const*, int, int, unsigned short*, int, char const*, bool) in ?? (/usr/lib/libhunspell-1.3.so.0)	
✓ 48743	2325963	0	2327233	HashMgr::load_tables(char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.so.0)	
✓ 48743	2325963	0	2327233	HashMgr::HashMgr(char const*, char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.so.0)	
✓ 48743	2325963	0	2327233	Hunspell::Hunspell(char const*, char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.so.0)	
✓ 48743	2325963	0	2327233	HunspellDict in /home/milian/projects/kf5/src/frameworks/sonnet/src/plugins/hunspell/hspelldict.cpp:36 (/home/milian/proj...	
✓ 48743	2325963	0	2327233	HunspellClient::createSpeller(QString const&) in /home/milian/projects/kf5/src/frameworks/sonnet/src/plugins/hunspell/huns...	
✓ 48743	2325963	0	2327233	Sonnet::Loader::createSpeller(QString const&, QString const&, const in /home/milian/projects/kf5/src/frameworks/sonnet/sr...	
> 407	18005	0	18057	HashMgr::add_hidden_capitalized_word(char*, int, int, unsigned short*, int, char*, int) in ?? (/usr/lib/libhunspell-1.3.so.0)	
> 1093	1261688	8360	1590168	QHashData::rehash(int) in ?? (/usr/lib/libQt5Core.so.5)	
> 28	525056	0	918848	__alloc_dtor in ?? (/usr/lib/libc.so.6)	
> 2	463600	0	463600	HashMgr::load_tables(char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.so.0)	
> 7383	409360	1600	426232	QMapDataBase::createNode(int, int, QMapNodeBase*, bool) in ?? (/usr/lib/libQt5Core.so.5)	
> 1	393216	0	393216	initialize_utm_tbl() in ?? (/usr/lib/libhunspell-1.3.so.0)	
> 9520	314328	24280	359552	QHashData::allocateNode(int) in ?? (/usr/lib/libQt5Core.so.5)	
> 2819	270880	320	423648	QListData::realloc(int) in ?? (/usr/lib/libQt5Core.so.5)	
> 162	208896	0	663552	realpath@GLIBC_2.3 in ?? (/usr/lib/libc.so.6)	
> 2158	169680	64	1206176	QString::reallocData(unsigned int, bool) in ?? (/usr/lib/libQt5Core.so.5)	
> 40145	152694	0	152694	0x7fd51990b6a in ?? (/usr/lib/libhunspell-1.3.so.0)	
> 9121	143456	160	295552	QListData::detach_grown(int*, int) in ?? (/usr/lib/libQt5Core.so.5)	
> 6	131072	0	196608	0x7fd6b417acd in ?? (/usr/lib/libbz2.so.1)	
> 998	124600	0	199600	QFileInfo::QFileInfo() in ?? (/usr/lib/libQt5Core.so.5)	
> 11292	121593	384	1122262	QByteArrayList::reallocData(unsigned int, QFlags<QByteArrayData::AllocationOption>) in ?? (/usr/lib/libQt5Core.so.5)	
> 1062	117400	0	212400	0x7fd6eefae7b in ?? (/usr/lib/libQt5Core.so.5)	
> 726	108416	0	162624	hspell_init in ?? (/usr/lib/libhspell.so.0)	
> 132	106488	0	205248	QJsonDocument::fromBinaryData(QByteArray const&, QJsonDocument::DataValidation) in ?? (/usr/lib/libQt5Core.so.5)	

# QT WIDGETS

- proven technology
- perfect for desktop applications
- not going to die anytime soon
- contemporary, if not modern

# 3. QT QUICK







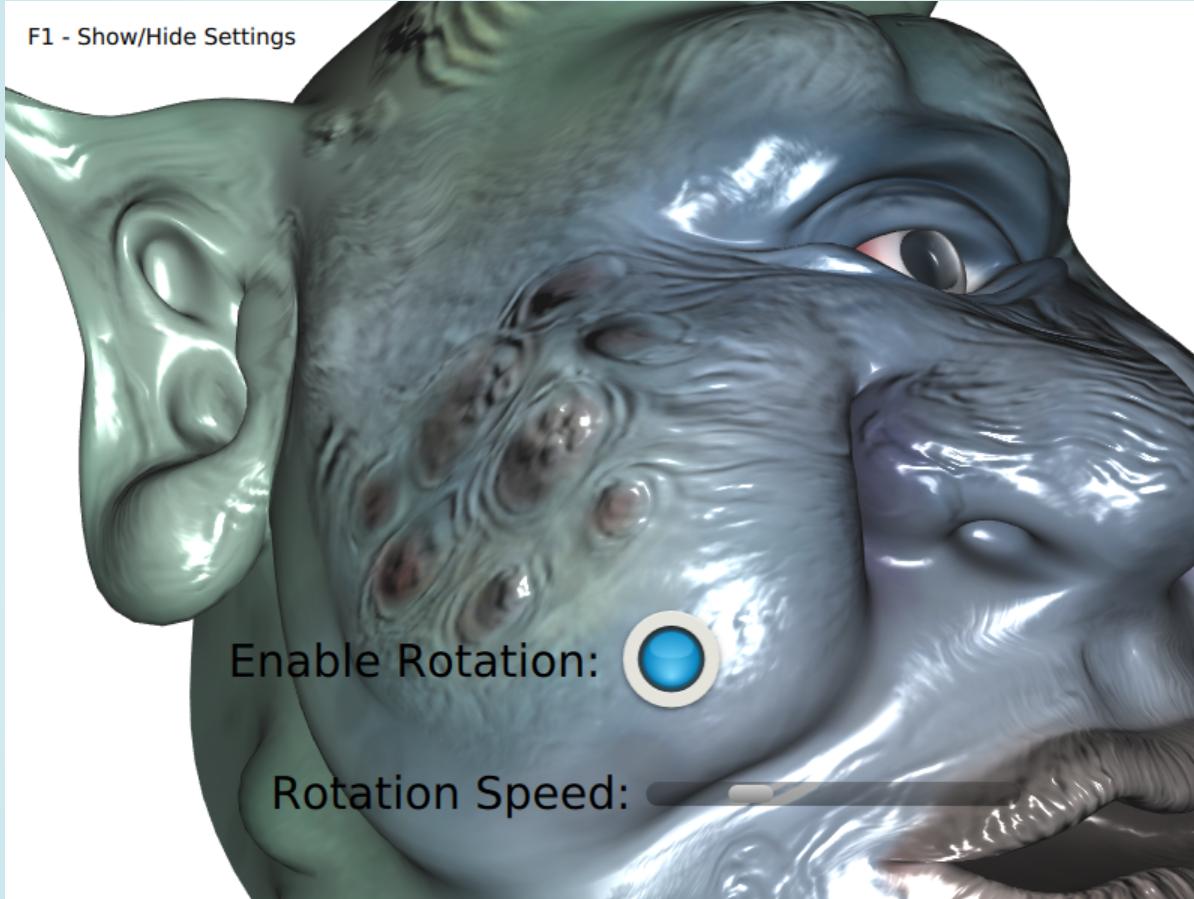


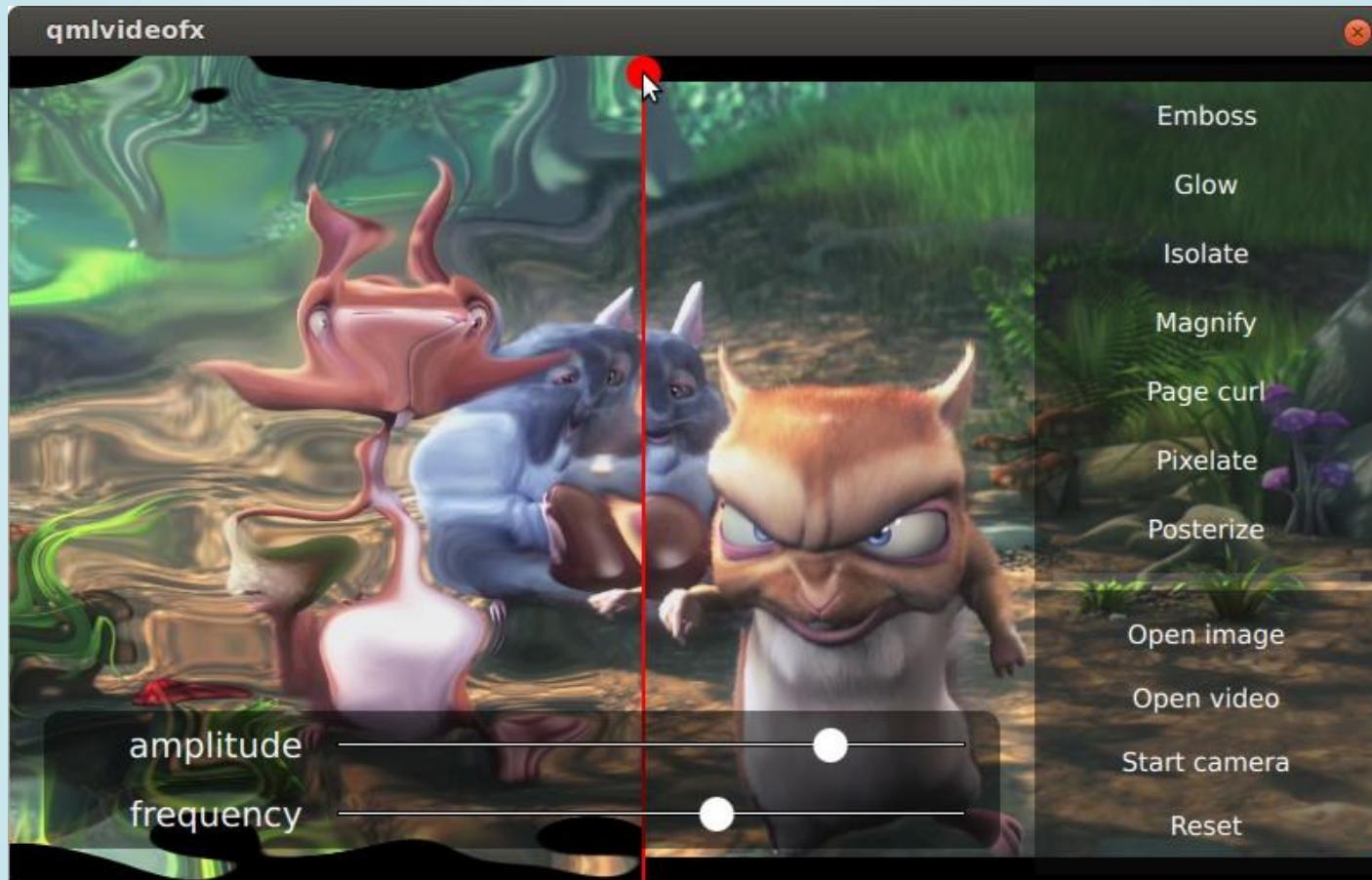
F1 - Show/Hide Settings

Enable Rotation:



Rotation Speed:





# QT QUICK

```
import QtQuick 2.2
import QtQuick.Controls 1.4

ApplicationWindow {
    visible: true
    width: 400

    Button {
        text: "Check Me"
        checkable: true

        x: checked ? parent.width - width : 0
        Behavior on x {
            NumberAnimation {
                duration: 1000
                easing.type: Easing.OutBounce
            }
        }
    }
}
```

- declarative QML + JavaScript
- hardware accelerated scene graph
- trivial to create fancy animated UIs
- perfect for embedded devices, phones
- ideal for OpenGL overlays

- only for representation
- data and logic in C++
- keep JavaScript to minimum

## 4. ESSENTIALS & ADDONS

# QT THE TOOLKIT

mix and match with Boost, STL & others

- signals & slots
- unicode strings
- any type wrapper
- filesystem access
- networking
- state machines
- localization
- accessibility
- date & time
- settings
- JSON
- XML
- SQL
- ...

# QT THE TOOLKIT CONT'D

- 2D painting
- OpenGL & 3D
- modern HTML
- printing
- testing
- serial ports
- sensors
- multimedia
- location
- bluetooth
- NFC
- ...

# KDE FRAMEWORKS 5

[api.kde.org/frameworks-api/frameworks5-apidocs](http://api.kde.org/frameworks-api/frameworks5-apidocs)

- compressed IO
- model/view addons
- advanced i18n
- task based threading
- hardware integration
- spell checking
- plotting
- more widgets
- configuration
- ...



# INQLUDE.ORG

- listing of additional Qt libraries
- easy installation of libraries via command line
- add what's missing!



# 5. MISCONCEPTIONS

- code generators
- containers
- C++11
- development process

# QT CODE GENERATORS

- moc
- uic
- rcc
- ...

- improve productivity
- simplify code maintenance
- overcome C++ limitations

## BUILD SYSTEM INTEGRATION

- QMake
- CMake
- Visual Studio

# MOC: REFLECTION

- limited C++ parser to extract annotations
- generates **QMetaObject** for reflection:
  - class name
  - inheritance chain
  - signals
  - slots
- basis for QML, DBUS, WebChannel, Remote Objects

moc is not required anymore for slots

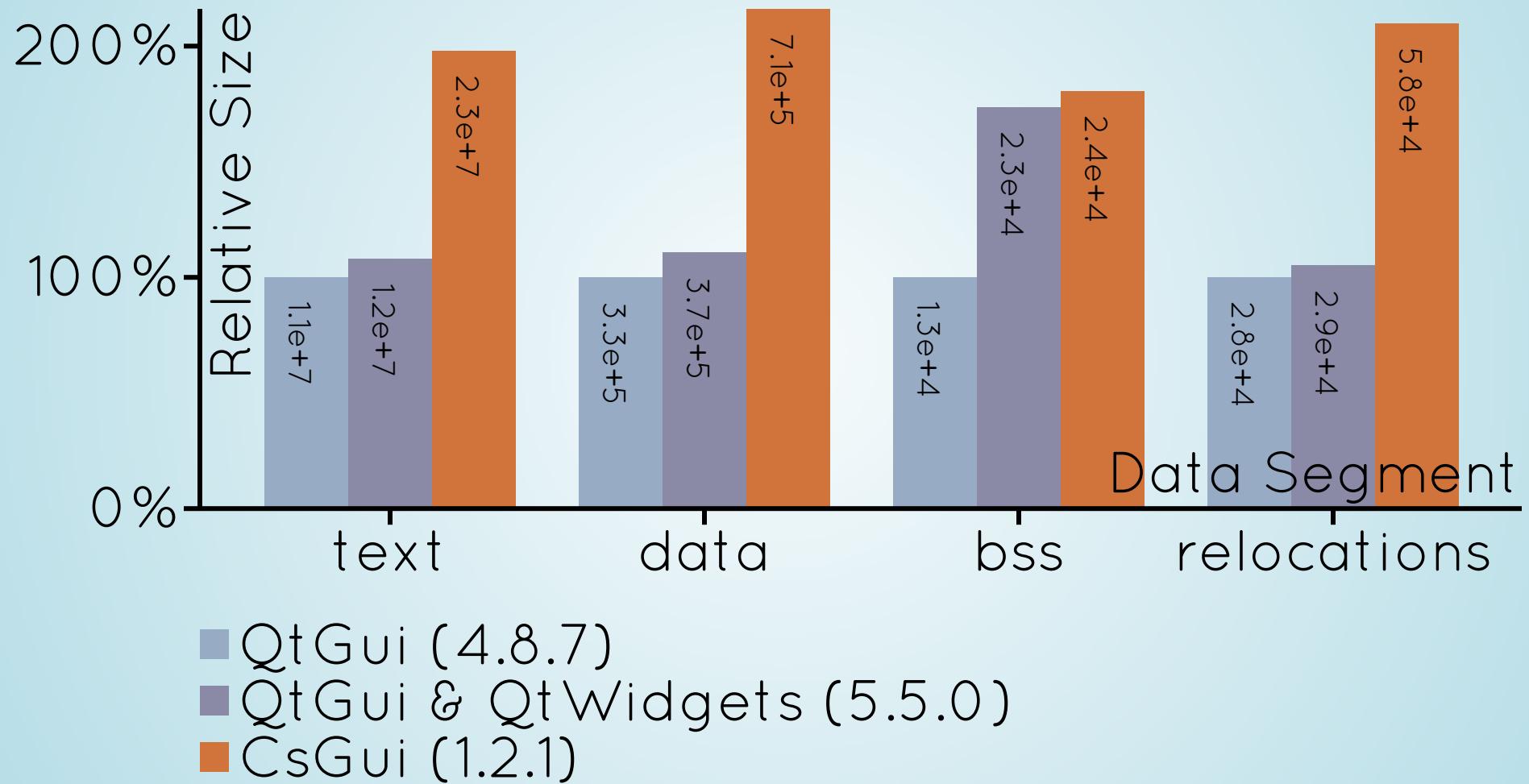
```
#include <QCoreApplication>
#include <QDebug>

int main(int argc, char** argv)
{
    QCoreApplication app{argc, argv};

    // execute lambda with 1s delay
    QTimer::singleShot(1000 /*msec*/, [&] () {
        qDebug() << "The time has come!";
        app.quit();
    });

    return app.exec();
}
```

## EFFICIENCY



# QT CONTAINERS

- QList
- QVector
- QMap
- QHash
- ...

- understand implicit sharing
- prefer to use **QVector** or **QHash**
  - STL / Boost containers are also fine!
- **QList** should die

# C++11 & QT

- internally:
  - C++11 optionally used up to Qt 5.6
  - Qt 5.7+ will depend on a C++11 compiler
- in your own applications:
  - use the latest and greatest C++XY

# DEVELOPMENT PROCESS

- open governance
- FOSS & commercial license
- KDE Free Qt Foundation
- CLA

# QUESTIONS?

[milian.wolff@kdab.com](mailto:milian.wolff@kdab.com)  
<http://www.kdab.com>

**What's new in C++11 and C++14?**  
C++ trainings run by KDAB