



Лаборатория системного программирования и
информационных технологий СПбГУ

Библиотека Qt

Библиотека OpenCV

Александр Корявко





План

- ▶ Библиотека Qt
 - ▶ общий обзор и история развития
 - ▶ обзор библиотек
 - ▶ инструменты разработки
 - ▶ система сборки cmake
 - ▶ механизм сигналов и слотов
 - ▶ надстройки на Qt
- ▶ Библиотека OpenCV
 - ▶ возможности и особенности
 - ▶ архитектура
 - ▶ пример – распознавание лиц

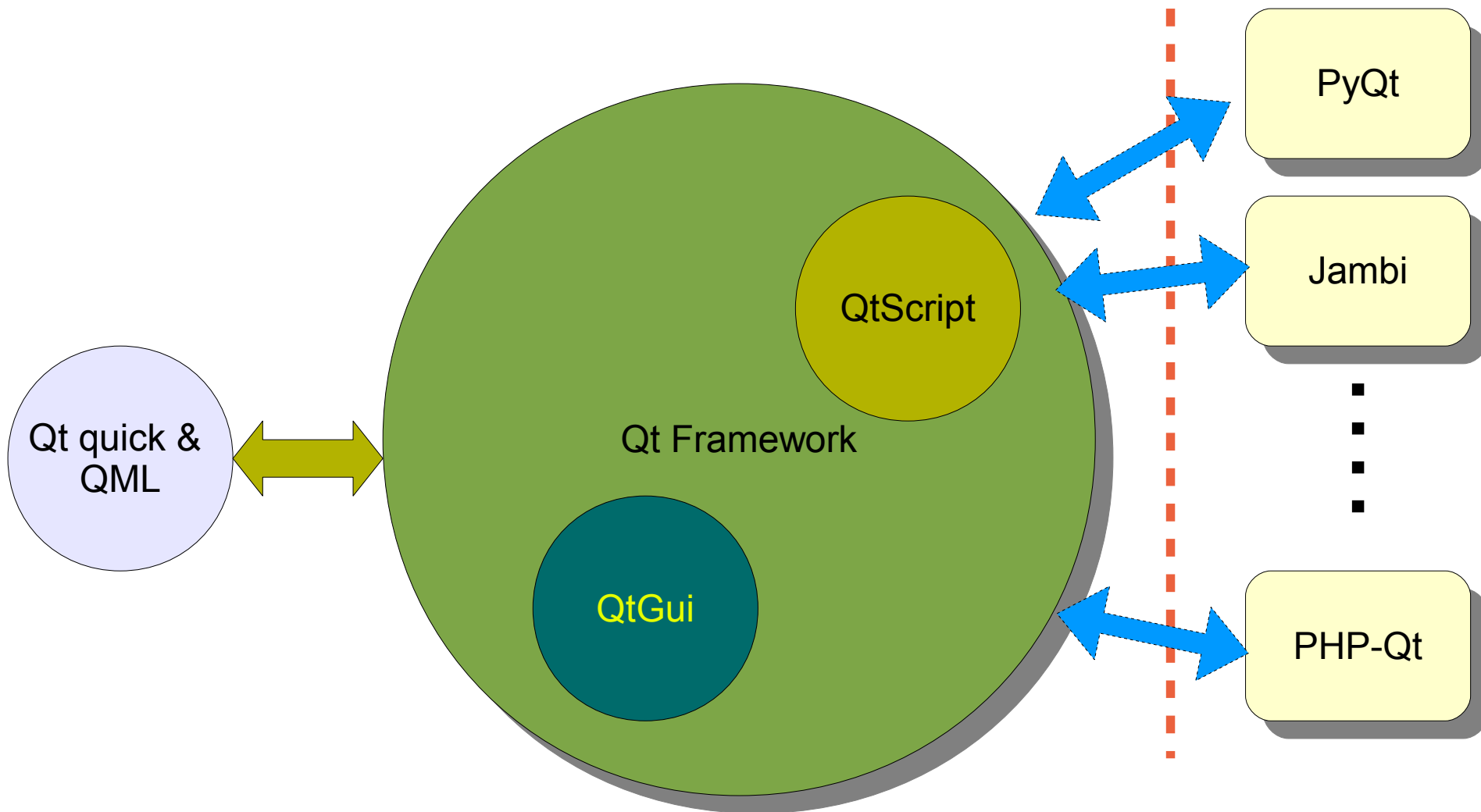


Технология Qt





Что такое Qt?





Мотивация использования фреймворка

- ▶ Обёртка с простым интерфейсом для порой очень сложных API
- ▶ Единый интерфейс для разных платформ
 - ▶ Linux / Unix
 - ▶ Mac OS X
 - ▶ Windows
 - ▶ Symbian
 - ▶ Windows CE / Mobile
 - ▶ ...
- ▶ Отказ от API ОС в пользу API Qt обеспечивает переносимость кода.
- ▶ Приложение использует мощность native-интерфейсов целевой платформы.



История развития

- ▶ Разработка начата в 1991 Haavard Nord и Eirik Chambe-Eng, основавшими в последствии компанию *Quasar Technologies*, затем переименованную в *Trolltech*
- ▶ Первый релиз – 1995 год: библиотека графических компонент для X11/UNIX и Windows под двойной лицензией для UNIX-версии и проприетарной для Windows
- ▶ 2005 – в Qt 4 действие GPL распространено на версии для всех платформ
- ▶ 17 июня 2008 – *Trolltech* приобретена компанией *Nokia*
- ▶ 2009 – для версии 4.5 добавлена лицензия LGPL



Ключевые библиотеки Qt

- ▶ QtCore – базовые примитивы, не имеющие отношения к GUI
- ▶ QtGui – примитивы GUI
- ▶ Phonon и QtMultimedia – библиотеки для работы с мультимедиа
- ▶ QtNetwork – работа с сетью
- ▶ QtOpenGL – поддержка OpenGL-графики
- ▶ QtXml – работа с XML
- ▶ QSql – работа с SQL-базами данных
- ▶ QtScript – позволяет использовать скриптовый язык, аналогичный JavaScript в Qt-приложениях
- ▶ QtWebKit – позволяет работать с веб-движком WebKit



QtMobility

- ▶ Дополнительная библиотека (не часть дистрибутива)
- ▶ Быстро развивается, текущая версия Qt Mobility 1.2
- ▶ Предоставляет интерфейс для функциональностей, специфичных для мобильных устройств:
 - ▶ сервисы (GSM-связь, Bluetooth)
 - ▶ записная книжка
 - ▶ мгновенные сообщения
 - ▶ органайзер
 - ▶ устройства позиционирования
 - ▶ сенсоры (акселерометр, датчик освещённости)



Ключевые инструменты разработки на Qt

- ▶ QtCreator – IDE для разработки на C++ и QML (язык построения интерфейсов на Qt)
- ▶ QtDesigner – наглядное создание графических интерфейсов
- ▶ QtLinguist – локализация интерфейса
- ▶ QtAssistant – система справки
- ▶ Qt Simulator – эмулятор мобильных устройств
- ▶ qmake – система сборки
- ▶ moc – метаобъектный компилятор
- ▶ uic – компилятор графических интерфейсов
- ▶ rcc – компилятор ресурсов





Система сборки qmake

- ▶ Сборка Qt-приложений осуществляется make
- ▶ qmake генерирует Makefile на основе файла проекта .pro (язык файлов проекта проще языка Makefile)
- ▶ Возможна автоматическая генерация файла проекта
 - ▶ qmake -project
- ▶ Процесс сборки контролируется установкой переменных qmake (выбор компилятора, директории сборки и т.д.)
- ▶ В проектных файлах можно использовать условные блоки и некоторые простые функции



qmake – пример

```
CONFIG += qt debug
QT += core gui network
HEADERS += hello.h
SOURCES += hello.cpp main.cpp
LIBS += -lcurses

win32 {
    SOURCES += helloworld.cpp
}

unix {
    SOURCES += helloworld.cpp
}

!exists( main.cpp ) {
    error( "No main.cpp file found" )
}
```





Signals and Slots

- ▶ Способ обмена сообщениями между объектами
- ▶ Каждый класс может объявлять сигналы, которые он может отправлять и слоты, которые можно ассоциировать с конкретными сигналами.
- ▶ Сигналы и слоты могут иметь аргументы
- ▶ Реализован как надстройка над синтаксисом C++ (исходный файл обрабатывается метакомпилятором moc, который генерирует вспомогательные файлы)
- ▶ Несовершенство метакомпилятора накладывает ограничения на допустимые конструкции C++



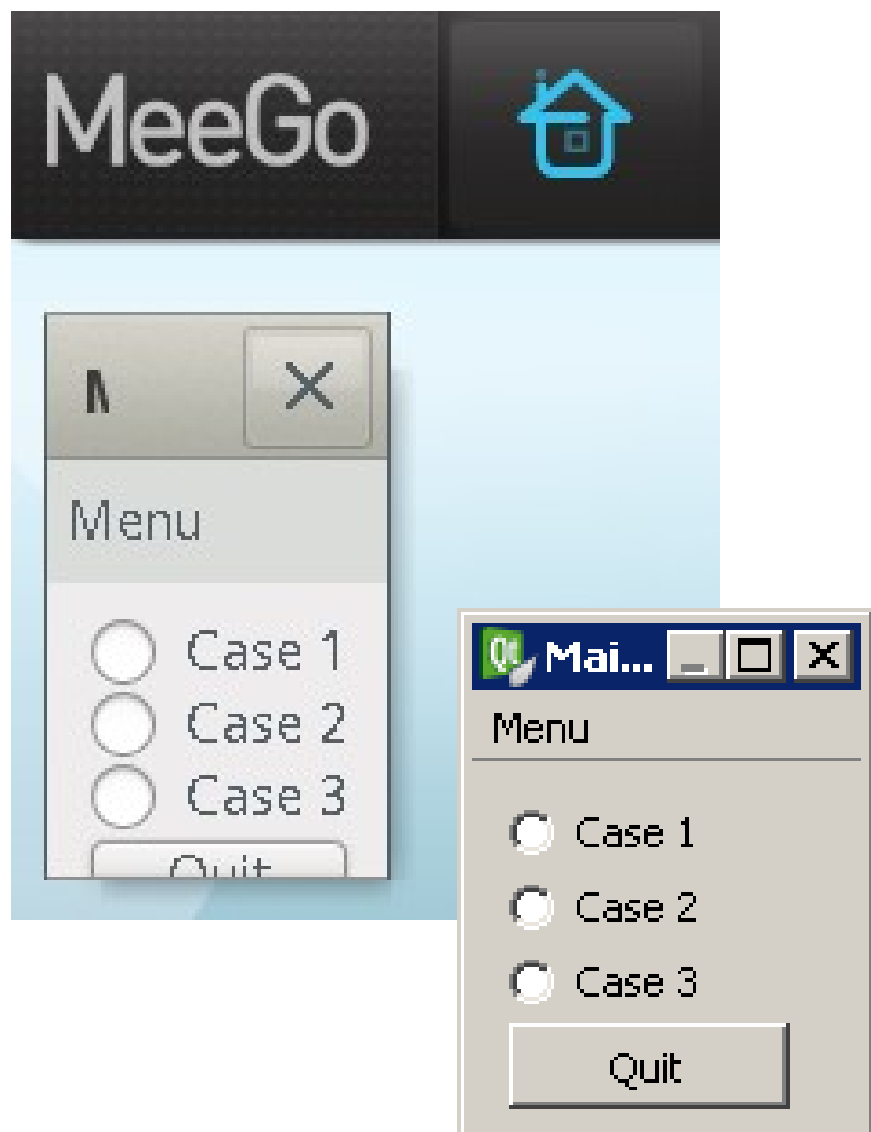
Signals and Slots – пример

```
#include <QObject>
class Counter : public QObject
{
    Q_OBJECT
public:
    Counter() { m_value = 0; }
    int value() const {
        return m_value;
    }
public slots:
    void setValue(int value);
signals:
    void valueChanged(int newValue);
private:
    int m_value;
};
```

```
void Counter::setValue(int value)
{
    if (value != m_value) {
        m_value = value;
        emit valueChanged(value);
    }
}
...
Counter a, b;
QObject::connect(&a,
    SIGNAL(valueChanged(int)),
    &b, SLOT(setValue(int)));

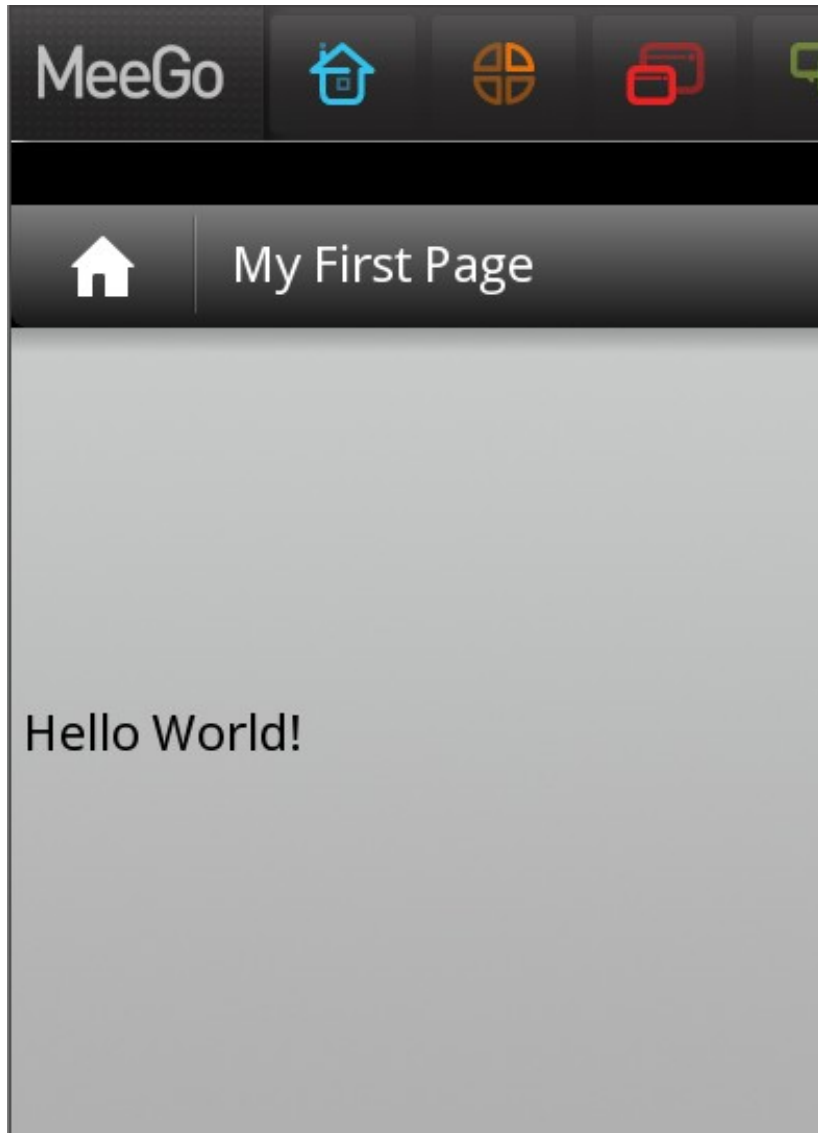
a.setValue(12); // a.value() == 12,
               // b.value() == 12
b.setValue(48); // a.value() == 12,
               // b.value() == 48
```

Qt в MeeGo ?



- ▶ Можно использовать QtCreator в полную силу
- ▶ Можно использовать QtDesigner
- ▶ Можно использовать эмулятор
- ▶ Работает отлично
- ▶ Внешний вид результата удручает...

Qt+MeeGo Touch в MeeGo!



- ▶ Сборка с MeeGo Touch добавлена в QtCreator
- ▶ Работает отладчик
- ▶ Нет встроенной справки
- ▶ Для MeeGo Touch не работает QtDesigner, но он и не нужен
- ▶ Внешний вид и элементы управления адаптированы для MeeGo



Библиотека OpenCV





OpenCV – назначение

- ▶ Набор алгоритмов компьютерного зрения:
 - ▶ Обнаружение объектов, в том числе лиц
 - ▶ Обнаружение плоских примитивов
 - ▶ Отслеживание движущихся объектов
 - ▶ Работа со стереокамерами
 - ▶ Стабилизация видео
 - ▶ ...
- ▶ Унификация интерфейсов подобных алгоритмов



OpenCV – особенности

- ▶ Открытый исходный код
- ▶ Язык C/C++, есть интерфейсы для Java, Python, Ruby,...
- ▶ Кроссплатформенность
 - ▶ Linux
 - ▶ Windows
 - ▶ Android
 - ▶ MacOS
 - ▶ iOS



OpenCV – архитектура версии 2.x

- ▶ **core** – базовые примитивы, включая объект Mat
- ▶ **imgproc** – базовая обработка изображений (примерно то, что умеет делать Photoshop)
- ▶ **objdetect** – поиск predetermined объектов, например лиц или машин
- ▶ **video** – анализ движения, отслеживание объекта и т.п.
- ▶ **calib3d** – поддержка стереокамер
- ▶ **highgui** – встроенная поддержка кодеков и GUI
- ▶ **gpu** – вычисления на графических ускорителях
- ▶ Интерфейсы для других языков программирования
- ▶ ...



Пример – распознавание лиц Инициализация

```
#include "opencv2/objdetect/objdetect.hpp"
```

```
#include "opencv2/highgui/highgui.hpp"
```

```
#include "opencv2/imgproc/imgproc.hpp"
```

```
Mat image = imread("face.jpg", 1);
```

```
CascadeClassifier cascade, nestedCascade;
```

```
nestedCascade.load(
```

```
    "../././data/haarcascades/haarcascade_eye.xml" );
```

```
cascade.load(
```

```
    "../././data/haarcascades/haarcascade_frontalface_alt.xml" );
```



Пример – распознавание лиц Обработка лиц

```
Mat gray( image.rows, image.cols );  
cvtColor( image, gray, CV_BGR2GRAY ); // Делаем черно-белым  
equalizeHist( gray, gray ); // Увеличиваем контрастность  
  
vector<Rect> faces;  
cascade.detectMultiScale( gray, faces,  
    1.1, 2,  
    CV_HAAR_SCALE_IMAGE,  
    Size(30, 30) );
```



Пример – распознавание лиц Отображение лиц

```
for( vector<Rect>::const_iterator r = faces.begin();  
    r != faces.end();  
    r++ )  
{  
    Point center;  
    center.x = cvRound((r->x + r->width*0.5));  
    center.y = cvRound((r->y + r->height*0.5));  
    int radius = cvRound((r->width + r->height)*0.25);  
    circle( image, center, radius, CV_RGB(0,0,255));  
}
```



Пример – распознавание лиц Обработка глаз

```
for( vector<Rect>::const_iterator r = faces.begin();  
    r != faces.end();  
    r++ )  
{  
    vector<Rect> nestedObjects;  
    Mat small = image(*r);  
    nestedCascade.detectMultiScale( small, nestedObjects,  
        1.1, 2,  
        CV_HAAR_SCALE_IMAGE,  
        Size(30, 30) );  
    // ...  
}
```



Пример – распознавание лиц Вывод результата

// или в файл

```
imwrite("output.jpg", image );
```

// или на экран

```
cvNamedWindow( "resultwindow" );
```

```
imshow( "resultwindow", image );
```




Пример – распознавание лиц

Фото 1 (340x300, 450 мс)





Пример – распознавание лиц

Фото 2 (475x356, 750 мс)





Для дополнительного чтения

1) Qt Documentation

(<http://doc.qt.nokia.com/4.7/index.html>)

2) OpenCV User Guide

(http://docs.opencv.org/doc/user_guide/user_guide.html)

3) Уроки OpenCV по-русски

(<http://robocraft.ru/page/opencv/>)





Вопросы?

avkoryavko@gmail.com



**Code less.
Create more.
Deploy everywhere.**