



Kai Trott

13. Januar 2015

- Qt [kju:t]
- GTK+
- WxWidgets
- Swing
- Windows Forms
- ...

- GIMP-Toolkit
- erste Version 1998
- LGPL
- Sprache: C
- Plattformen:
  - Windows
  - Unix
  - Mac
  - Maemo

- erste Version 1992
- modifizierte LGPL
- Sprache: C++
- Plattformen:
  - Windows
  - Unix
  - Mac
  - Symbian OS
  - Palm OS
  - Windows CE

- erste Version 1997
- fester Bestandteil der Java-Runtime
- Plattformen:
  - jede mit Java-Runtime

- Mono
- MIT, GPL, LGPL
- Sprache: C#
- Plattformen:
  - Windows
  - Linux (Mono)
  - Mac (Mono)

1992 Entwicklung begann

1994 Gründung Trolltech (Oslo, Norwegen)

1998 "Qt Free Edition" für KDE e.V.

1999 Qt 2.0

2000 ■ Qt 2.2 unter GPL und "Qt Public License"  
■ zB. Qt/Windows noch kostenpflichtig!

2001 Qt 3.0

2003 Qt/Mac unter GPL

2005 ■ Qt 4.0  
■ Qt/Windows unter GPL  
■ KDE portiert auf Qt 4.0

- 2006 Trolltech geht als TROLL an die Börse
- 2008 Übernahme durch Nokia. Umbenannt in "Qt Software"
- 2009
  - Qt 4.5 mit GPL und LGPL
  - QtCreator
- 2010
  - Qt 4.7
  - QML und QtQuick
- 2011
  - Nokia entscheidet sich für Windows auf Mobilien Geräten
  - Lizenzverkäufe an Digia (Helsinki, Finnland)
- 2012
  - komplette Übernahme von Qt durch Digia für 4 Millionen Euro
  - Qt 5.0



- Windows
- Windows CE
- Mac OSX
- Unix (X11, Framebuffer, eglfs)
- VxWorks
- Symbian
- Maemo
- WebOS
- Android
- iOS
- BlackBerry
- UbuntuTouch
- SailFish OS
- AmigaOS
- eComstation
- Haiku

### Kernmodule:

- Qt Core
- Qt Gui
- Qt Network
- Qt WebKit
- Qt Multimedia
- Qt Widgets
- ...

### Addons:

- Qt Bluetooth
- Qt NFC
- Qt OpenGL
- Qt SerialPort
- Qt XML
- Qt Sensors
- ...

- LGPL als Primäre OpenSource Lizenz
- **Lizenzmodelle:**
  - Community (GPL, LGPL)
  - Indie Mobile
  - Professional
  - Enterprise

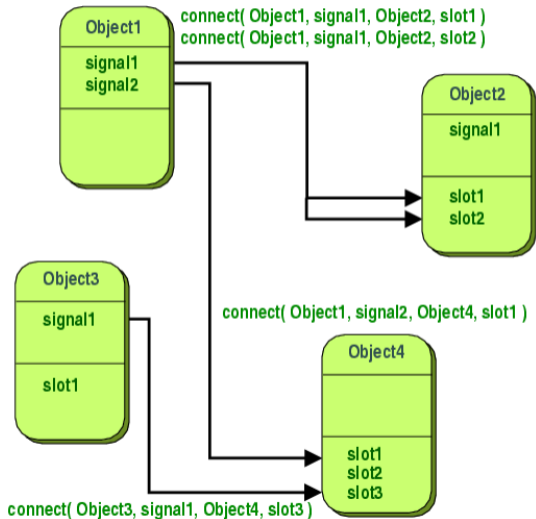
- Qt Creator
- Qt Designer

### Signale:

- werden bei Event "geschmissen"
- auch eigene Signale möglich
- können Zusatzinformationen beinhalten

### Slots:

- als normale Methoden verwendbar
- auch eigene Slots möglich
- können mit Signalen verbunden werden
- müssen nicht alle Zusatzinformationen verwenden



- wichtig?
- leicht umzusetzen?
  
- lupdate
- linguist
- lrelease

- eigene Programmiersprache für die Entwicklung von GUI's
- basiert auf JSON
- Javascript integrierbar



---

```
1 Rectangle
2 {
3   color: "green"
4   x: 5; y: 10
5   width: 50
6   height: 2 * width
7 }
```

---

---

```
1 MouseArea
2 {
3     onClicked:
4     {
5         console.log("Hello World");
6         label.text = "Guten Abend!";
7     }
8 }
```

---

Weitere Konzepte:

- dynamisch objecte erschaffen (per JavaScript)
- GUI erweitert sich selbst
- dynamisches Property-Binding
- integrierte Timer (Single-Shot und Recurring)