We are the 1 %

Porting KWin to Qt 5

Martin Gräßlin

2013-07-13 | Bilbao | Akademy

Be Free. KDE

About me

Martin Gräßlin
mgraesslin@kde.org
http://blog.martin-graesslin.com

KWin Maintainer
Focus on Porting to Qt 5 and Wayland

bluesystems

Disclaimer

I think all changes in the Qt API presented in this presentation are an improvement!

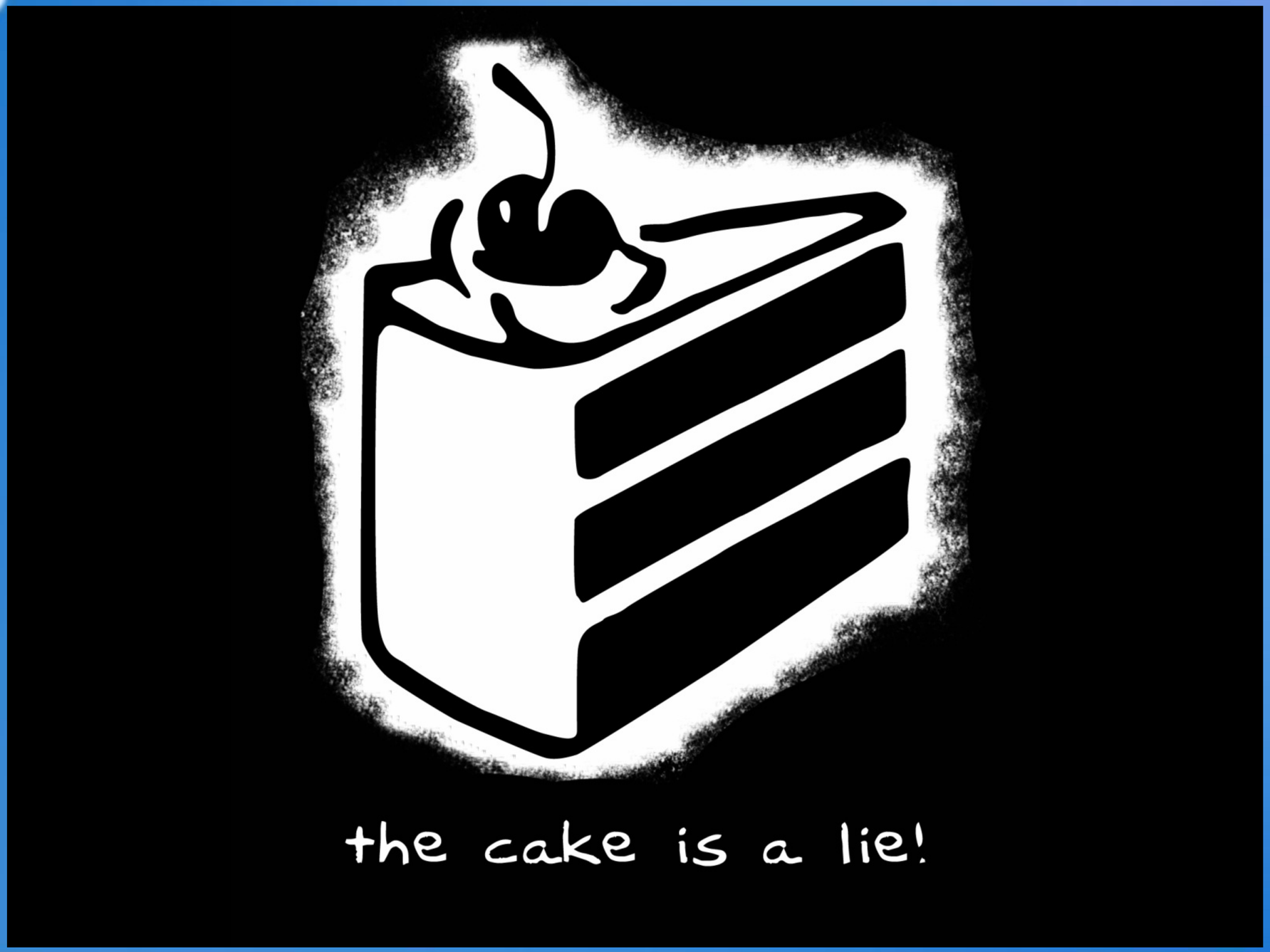
Using the abstraction over X11 in previous versions of KWin was the correct decision by the developers back then!

And there be cake

Compatible with Qt 4

99% Source compatible

(Qt 5 Roadmap, Keynote at Qt Developer Days 2012 by Lars Knoll, slide 19)



You filter XEvents?

Note: The native events that can be filtered this way depend on the QPA backend chosen at runtime. On X11, XEvents are replaced with `xcb_generic_event_t` due to the switch to XCB, which requires porting the application code to XCB as well.

(QtDoc 5.0: C++ API changes - https://qt-project.org/doc/qt-5.0/qtdoc/sourcebreaks.html)

QPA

Platform ports got removed.

Example: QPixmap::handle()

Returns the pixmap's handle to the device context.

Note that, since QPixmap make use of implicit data sharing, the detach() function must be called explicitly to ensure that only *this* pixmap's data is modified if the pixmap data is shared.

Warning: This function is X11 specific; using it is non-portable.

Warning: Since 4.8, pixmaps do not have an X11 handle unless created with fromX11Pixmap(), or if the native graphics system is explicitly enabled.

...and in Qt 5

QPlatformPixmap* handle() const;

- No longer listed in documentation
- Still referenced in the documentation
- Not mentioned in C++ API changes
- Not deprecated in Qt 4

X11 specific

We have been warned!

For normal Developers

For KWin Developers

How to find the Problems?

- Not documented
- Not deprecated
- Compiling against Qt 5 not an option to find the issues

Educated Guesses

- Search for handle()
- Search for fromX11() toX11()
- Search for XEvent

http://community.kde.org/KWin/QT5

Example for Abstraction

```
supportWindow = new QWidget(NULL,
    Qt::X11BypassWindowManagerHint);
XLowerWindow(display(), supportWindow->winId());
rootInfo = new RootInfo(this, display(), supportWindow->winId(),
"KWin", protocols, 5, info.screen());
```

- supportWindow is not used as a QWidget
- only used as an abstraction over XWindow
- assumes that a QWidget is an XWindow (hello Wayland!)

How that looks in 4.11

- Creates a low level xcb_window_t
- supportWindow no longer hold by KWin, but only in RootInfo

Gotchas:

- Roundtrip to X-Server needed before creating the RootInfo
- Size of supportWindow needs to be valid

QCursor - Usage in KWin

- QCursor::pos()
- QCursor::setPos(const QPoint &pos)
- QCursor::handle()

QCursor::pos()

- Performs X roundtrip each time
- Also in Qt 5
- KWin has optimized replacement
- In Wayland KWin controls cursor pos

QCursor::setPos(...)

- XWarpPointer
- warping not (yet) possible in Wayland

QCursor::handle()

- QCursor references an xcb_cursor_t
- Allows low level setting of cursor on an X Window (xcb_change_window_attributes)
- Who needs that?

QCursor solution

- KWin::Cursor replaces QCursor
- Uses our optimized ::pos() version
- Provides mapping from Qt::CursorShape to XCursor
- Additional features like mouse polling

The QPixmap story

- KWin composites X Pixmaps
- QPixmap wraps an X Pixmap
- QPainter is nicer API to render to XPixmaps than XRender

QPixmap and raster

- Qt 4.5 (?) introduced graphics system raster
- QPixmap does not reference an XPixmap
- handle() returns 0
- All assumptions break

Solution: hard enforce native

The QPixmap story

native (XRender) sucks

The QPixmap story

QPixmap::fill(Qt::transparent)
expensive with many drivers

The QPixmap story

Texture from Pixmap sucks

The QPixmap story

The Wonder Patch

- Use raster for OpenGL compositing
- Use QPixmap::fromX11Pixmap() if we need an XPixmap
- XRender forced into continue to use native

The QPixmap story

XRender compositing sucks with all modern parts of KWin

The QPixmap story

For Qt 5 two possible solutions:

- Drop XRender
- Improve XRender on Raster

- XRender compositor mostly rewritten in 4.11
- Supports raster in window decorations
- No more fromX11Pixmap
- But still used on native

The hidden gems

- QRegion references an XRegion, used for window shapes
- QSessionManager not implemented for X11 in Qt 5

The XEvent handler

- A window manager needs to process XEvents
- KWin implements QApplication::x11EventFilter(XEvent* e)
- Event filter is heart of KWin
- 1500 lines of huge switch statements
- Parts in kdelibs
  - NETWinInfo
  - NETRootInfo

Obstacles for Porting

- Switching to XCB not possible in Qt 4
- Not all features used from XLib are available in XCB
- Unit testing not possible

Status of Porting

Done
- QPixmap
- QCursor
- QRegion

TODO
- Session Management
- Event filter

It compiles

But doesn't run

Questions????

Be Free. KDE