Plasma Widgets

API and Theming approaches

Marco Martin

What is a plasma widget?

- KDE 4.0: everything from scratch
- 4.1/Qt4.4: QGraphicsWidget
- Preferably a QGraphicsProxyWidget
  - This means: every Qt widget you can imagine
- Exception of container widgets, like TabWidgets and Frames
  - We want to contain GraphicsWidgets and GraphicsLayouts here

Standard API: why wrap?

- Normally you would do this:

```
QPushButton *nativeButton = new QPushButton();
QgraphicsProxyWidget *proxy = new QgraphicsProxyWidget(this);
proxy->setWidget(nativeButton);
```

- Very ugly code
- And in scripting too, where we would want something better
- Very scarce possibilities of theming
- So every widget has its own subclass with a set of standard methods

What to wrap?

- Most used functions
- Makes the code simpler
- In particular for scripted applets
- Important functions
  - setText() text() when the underlying widget has it
  - setImage() getImage() aware of Plasma theme, so you can do setImage(“widgets/clock”)

Additional stuff

- setStyleSheet()/styleSheet(): applet creators can easily tweak the widget appearance with standard Qt stylesheets
- nativeWidget()
  - ...when a basic api is not enough :)
  - Not available in EcmaScript bindings to keep api clean
  - Available in richer bindings like ruby

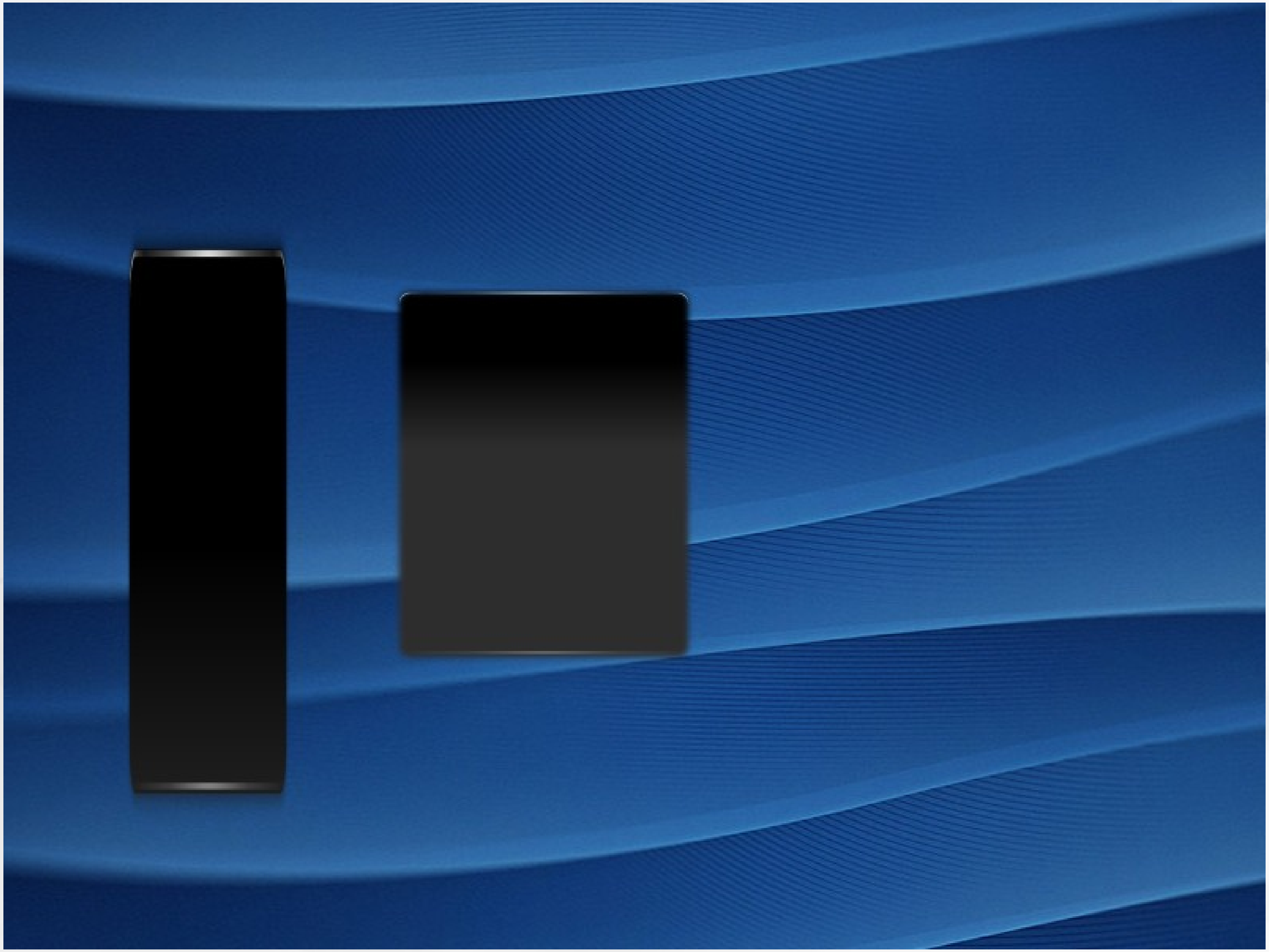
Theming

- 3 approaches
  - Overriding the paint function of the proxywidget
    - Better graphics results (no clipping and float arithmetics)
    - Used for instance in Plasma::PushButton
  - Subclass the QWidget and override paintEvent
    - When the widget is really complex and we need other stuff like a custom sizehint (and notification of its change)
    - Used in the tabbar
  - Using a Kstyle: open possibility for scrollbars, easier to apply when they are only available children of other widgets (i.e. scrollviews)

What are themes?

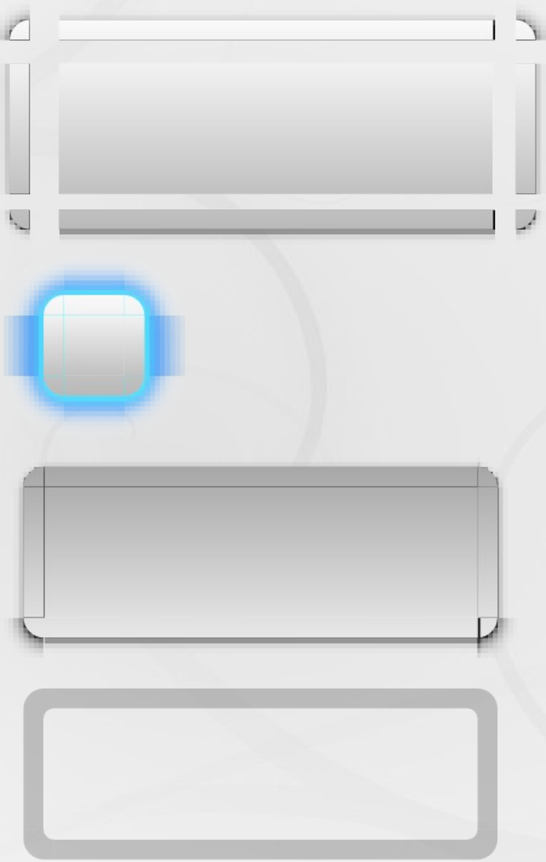
- QWidgets on canvas looks really out of place
- A color scheme provided by Plasma::Theme and many svgs, described in
- http://techbase.kde.org/Projects/Plasma/Theme
- Svg element naming format used by the PanelSvg class
- Useful to make more or less rectangular elements, so good for buttons, textboxes etc

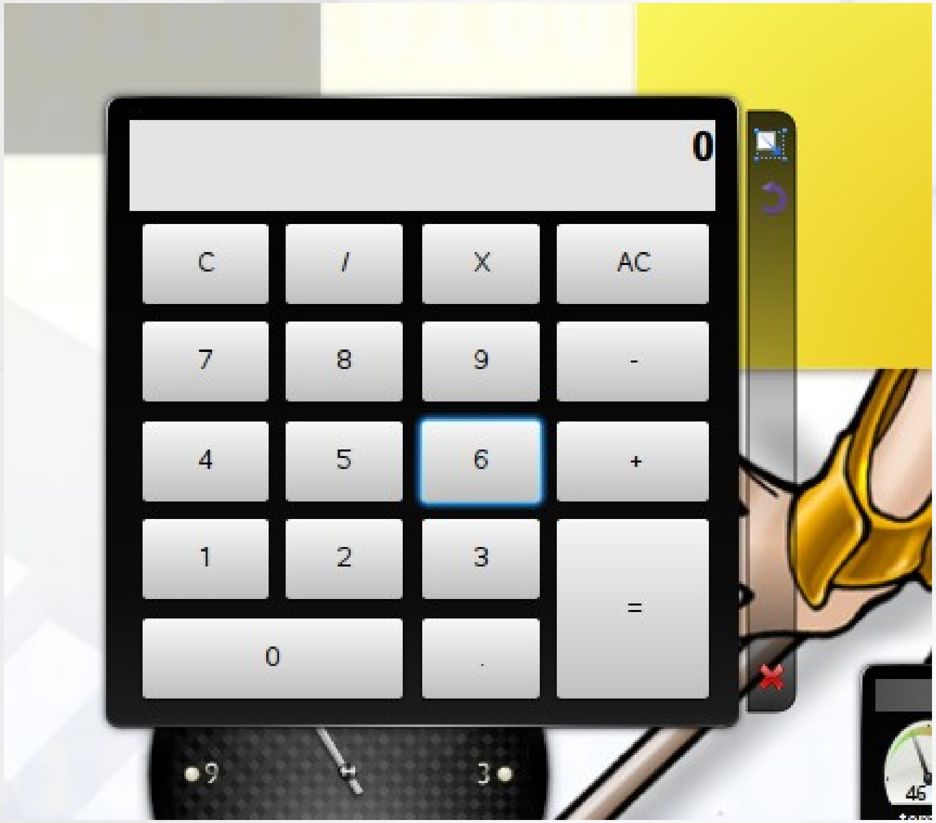
Why not a single svg?



Example

- 9 element for each prefix, here the “normal”, pressed, active and the focus rect




0
C
/
X
AC
7
8
9
-
4
5
6
+
1
2
3
=
0
.