Qt on the Mac

James Turner

<james@kdab.com>
Who me?

- Mac developer for sixteen years
- Classic, Carbon, Cocoa, Qt, iOS
- Qt developer for ten years
Introduction

- Motivation
  - Problems with Qt UIs on Mac
- Bottom-up improvements
- Top-down Mac design
- Deployment and eco-system
Why are you here?
Motivation

- Qt GUIs are cross-platform
- Functionality is portable
- Qt widgets follow native appearance
  - including size hints
- We’re finished, let’s ship on Mac
“it doesn’t feel right”
Expectations

- Mac users are pixel-picky
  - Spacing
  - Icons
  - Fonts
- Follow the Apple applications
  - Keynote, Mail, Aperture
- Subconscious perception of differences
  - ‘uncanny valley’ of UI similarity
In-depth consideration of your UX

- Less is *always* more

- Engineers design terrible UIs

- Your CEO, marketing and sales people use Macs

- Irritating media & reviewer types use Macs
The lazy option

- Tolerate a functional UX
  - Possible in certain vertical markets
  - Mac users are an ungrateful lot
  - Reviewers will not be kind
Go wild

- Totally custom UI on all platforms
  - CSS stylesheet or custom QStyle
  - QtQuick UI
• Motivation
  • Problems with Qt UIs on Mac
• Bottom-up improvements
• Top-down Mac design
• Deployment and eco-system
Bottom-up Improvements

- Use a unified toolbar
- Reduce widget and window framing
- Parent dialogs
- Tune layouts and spacing / borders
- Use custom widgets, especially buttons
Toolbars

• Mac toolbar is unified, flat, and immovable
• Standard editing UI
You’ve been framed

- Mac UI minimizes framing & borders
  - Contrast to Windows, especially Win 2k
- Few 3D highlight effects
- Nesting `QGroupBox` or `QFrame` looks terrible
- Convert to a plain `QWidget` on Mac
● Window specific modal dialogs should appear as sheets

● Qt handles this if you set the modality and parent widget correctly on QDialog
Layouts and spacing

- Use a custom stylesheet on Mac
  - Set custom margin, border, spacing
  - By class or id (objectName)
- Use an alternate .ui file on Mac
- Avoid stretching widgets
  - Especially QPushButton
Mac widgets come in three sizes

- Default is ‘regular’ (large)
- Smaller is often better
- Don’t mix in a window / dialog / palette

Qt::WA_MacNormalSize
Qt::WA_MacSmallSize
Qt::WA_MacMiniSize
Custom widgets

- Avoid widgets which aren’t native to Mac
  - QDial, QToolButton
- Use QTabBar documentMode
- QPushButton setFlat()
- Make a segmented button control
- Wrap a native control in QMacCocoaViewContainer

My ... | Se... | T... | Tab IV: A N... | The Undiscovere...
• Many additional button styles
  • Compact
• Suitable in many places
• `QPushButton` is not
• `Qocoa` provides some easy wrappers
• Plain `QPushButton` on other platforms
Item views

- Default list, table appearance
- Mac is emphasizing composite list items over tables
- Needs a custom delegate
- Standard add, remove and edit UI
• Motivation
  • Problems with Qt UIs on Mac
• Bottom-up improvements
• Top-down Mac design
• Deployment and eco-system
Top-down design

- Design a Mac UI from scratch
- Identify
  - Areas which can share UI with other platforms
  - UI elements missing in Qt
  - Concepts that different entirely on Mac
- Structure the app
  - Share logic and common UI
  - Keep platform UI distinct
Main window

- Main window is critical
  - Usually the most radically different
- `QMainWindow` is not a big help
  - avoid dock widgets, movable toolbars, status bar, MDI
- Support an intelligent full-screen mode
Use Cocoa

- Create your main window using Cocoa
- Populate
  - Cocoa widgets
  - Qt widgets wrapped in QMacWidget
- Bridge Interface Builder (IB) actions to Qt slots
Motivation

- Problems with Qt UIs on Mac
- Bottom-up improvements
- Top-down Mac design
- Deployment and eco-system
Ecosystem

- Available (and update) through the App Store
- Support iCloud storage
- Make full-screen mode really good
App Store deployment

- Code-signing
- Only access files in your sandbox
  - Use iCloud for document storage
- Don’t download code or scripts
- Update through the App Store
Code-signing

• Pay Apple $100 / year
• Prove your identity
• Sign your final bundle with the certificate
• Radically different file dialogs
• Works via **NSDocument** and **NSDocumentController** APIs in Cocoa
  • not exposed to Qt
• Might affect deep structures of your app
• Think about UX early, not late
• Ensure the person working on your Mac UI loves the Mac
• Think different
Questions?