Developing Software for Small Devices

Bradley T Hughes – Trolltech AS

About Me

- American - from Amarillo, Texas
- Software Engineer for the Qt library
- Came to Norway and Trolltech in May 2000

About Trolltech

- Founded in 1994 by Haavard Nord (CEO) and Eirik Chambe-Eng (President)
- Two product lines: Qt and Qtopia
- Dual licensing strategy
- Headquarters in Oslo, offices in Brisbane, Australia, Palo Alto, California and Bejing, China
- Over 130 employees from 19 countries

Challenges

- Limited resources
  - Screen
  - Processor and memory
  - Input capabilities
- Use model
  - Different tasks
  - Different mindset
  - Different environment

Typical Usage

- Used in short bursts
- For a specific purpose
- Used in uncontrolled environments
- Industrial use
  - Safety critical, non-distracting, vital information
- Consumer use
  - Fashion accessory, expressing personality

Adapting to Usage Requirements

- User interface must be fast and easy to use
  - Omit non-essential features
  - Highlight most important information
  - Streamline common operations
  - Minimize text input
  - Make it predictable (who has a phone manual?)
- Theming
  - Personalization, customization

Device Limitations

- Small screen
- Little memory (RAM/ROM)
- Slow processor
- No floating point unit
- No keyboard
- Inaccurate “mouse”

Common Display Types

- Small physical size
- Low resolution
  - PDA 320x240
  - Phone 176x220
- High-end devices have higher resolutions, but same physical size

Adapting to Limited Screen Size

- Use Qt's layout classes
- Simplify the user interface
  - Omit features
- Less text
- Use widgets that take less space
- Split into sub-dialogs; wizards
- Scrollbars and tabbed widgets

Input Limitations

- Text input is hard and/or slow
- Touch panels
  - Unstable/jittery
  - No mouse-over events
    - No enter/leave notification
    - No tooltip
    - No auto-raise/highlight
    - Cannot use cursor shape to convey information

Internationalization and Localization

- Decide early if i18n is needed
- Design user interface accordingly
  - Use Qt's tr() mechanism
- Translation is part of the development process
  - Changes to the UI are often necessary
  - Be careful not to postpone translation until just before product launch
- Icons - “a picture is worth a thousand words”

Internationalization and Localization (2)

- Be prepared for l10n issues
  - English: Info
  - Norsk: Opplysninger
- Apart from Asian languages, English is one of the shortest languages
- And the list goes on…
  - Symbols, currency, time, culture, reading direction etc.

Memory (RAM)

- Conserve memory
- Avoid memory leaks
  - Long time between reboots/restarts
- Measuring memory use is difficult
  - On Linux, top/ps include shared resources
  - Inaccurate (buffers, cached)
- Use a memory profiler

Memory (ROM)

- Tailor the library size
- Optimize application for size
  - Be careful with inline functions and templates
  - Experiment with compiler options for fine tuning
- Select fonts
  - True-type takes time to render, needs more code
  - Pre-rendered takes space

Processor Limitations

- Optimize for speed
  - Loop unrolling, cache coherency, etc.
  - Cache data
  - Lazy initialization
  - Take advantage of idle time
- Don't trust your instincts – use a profiler
- Perceived speed vs. actual speed
  - Give feedback, preliminary results

Power Management

- Device can sleep at any time
  - Application may experience sudden time jumps
  - May need to ask device to wake up at a later time
- Help conserve power
  - Keep application idle (never busy wait)
  - Avoid unnecessary, rapid timer events (animation)

ARM Platform Notes

- ARM processors have no FPU
  - Kernel traps are extremely slow
  - Best solution: soft-floats
    - Software emulation built into toolchain
    - Not available in older toolchains
  - Alternative: use custom fixed/floating point class
- Enable framebuffer write-combining in kernel
  - Patch developed by Intel for XScale

Development Tools

- Cross-compiling toolchain
  - Compiler, linker, loader, utilities
  - Libraries and header files
  - May be available from hardware vendor
  - May need to build your own
- Emulators
  - Commercially and freely available
  - May not always emulate limited resources

More Development Tools

- Valgrind
  - Memory checker, memory profiler, cpu profiler
- Virtual framebuffer (QVfb)
  - Simulated framebuffer on X11
  - Can emulate a touch screen, different depths, resolutions
  - Not an emulator, applications run natively

Development Hardware

- Reference boards
  - Useful for development phase
  - Easy to get software running
  - Not well-suited for usability testing
  - Form-factor boards
- “Off-the-shelf “ hardware
  - USB memory sticks, CompactFlash, BlueTooth
  - Ethernet for PC and/or device

Running on the Device

- Cross-compile, run from USB key, CompactFlash, etc.
- Cross-compile, run from NFS or SMB exported file system
- Flash new ROM via serial console (when developing an entire suite of applications)
- Use Qt's VNC graphics driver (run on the device, test from workstation)

Debugging Your Software

- Things will break
- Many emulators are also debuggers
- Debuggers for various devices exist
  - You actually debug on the device (a plus)
  - Tedious (constantly moving between workstation and device)
- Recommendation: remote debugging
  - Debug on device, control from workstation

What to Take Home

Writing software for small devices is very different when considering the development process, user interface design, hardware, end-user, and more.