Innovating in Automotive Infotainment with QtQuick

November 2012

Ekelundsgatan 4, SE-411 18 Göteborg, Sweden
www.pelagicore.com
Bio

• Johan Thelin
  johan.thelin@pelagicore.com
  I like questions! Just ask!

• Written articles, blogs, book, etc
• Co-founder of QtCentre
• Worked with Qt 10+ years
• Worked with automotive software for 5+ years
At a glance…

• Founded in 2009
• We provide products and services for Open Source Infotainment
• Customers are OEMs and their suppliers in the automotive industry
• Our core competencies are:
  – GENIVI/Linux based Automotive Infotainment
  – User Experience development using Qt
  – Automotive Silicon support
• Owned by Employees and Fouriertransform
• Offices in Sweden and Germany
Eco system

- Members since 2009, Member of the System Architecture Team, key contributor

- Driving eco-system of streaming, networked media

- Members of the steering committee

- Silver member

- Active contributors

- Members of the R-Car alliance focusing on Linux based applications
Differentiators

- Reliability
- Safety
- Performance
- Comfort
- Beauty
- Infotainment
IVI history

• Radio
IVI history

• Radio
  - RDS
  - TMC
  - FM / AM
  - Sirius / XM
  - DAB
  - Music meta-data
  - Time-shift
  - Automatic tuning
  - etc
Complexity
Integration
Infotainment market size

- Approx. 41.2B$ 2016
- Growth approx. 5.5% per year
- Mostly hardware but considerable software spend

Source: IHS iSupply Research, February 2012
Platform market share

- QNX market lead, but declining
- Windows stable, mostly in Asia
- GENIVI/Linux up to 24% in 2018

Source: IHS iSupply Research, April 2012
User expectations
IVI challenges
IVI challenges

• A car is not a phone
• Driver workload?
• Information relevance?
• Information complexity?
Interaction surfaces

- The IVI UX is an integrated part of a larger system
  - Screens
  - HUD
  - LEDs
  - Knobs
  - Sound
  - Voice control and speech feedback
Interaction methods

• A number of tricks to present information
  – Transitions to guide the eye
  – Gestures instead of hard-to-hit buttons
  – 3D for data visualization
    • Qt3D
    • Kanzi – integrated into QtQuick
Multiple applications

- Media player
- Navigation
- Vehicle status messages
- Bluetooth calls and messages
- E-mail
- Traffic alerts
Safety first

- Do not startle or interrupt the driver
  - Sudden sound
  - Moving pictures
  - Require visual focus away from the road
  - Asking complex questions at the wrong time
  - etc

- Applications needs to be managed and prioritized
Shared resources

- Everybody uses a limited set of resources
  - Main speakers
  - Head-unit screen
  - USB ports
  - Optical media drivers
  - etc

- Resources need to be managed and applications prioritized
Volatile resources

- Resources come and go over time
  - Bluetooth devices
  - Network coverage
  - Radio reception
  - USB sticks come and go
  - etc

- Resources needs to be managed
Pelagicore Resource Framework

• A framework for managing resources
  - Monitoring
  - Access control and priorization
  - Aggregation
  - etc
Engineering challenges

- A car is not a phone
- You pay €500 for a phone and use it for a year
- You pay €30k+ for a car and use it for 3+ years
Availability

• Start-up time
  - Availability on networks (CAN)
  - Early video streams (rear-facing cameras, parking assist, etc)
  - Early audio
  - etc

• Stability
  - Safety
  - Consumer expectations
Hardware platforms

• Silicon
  - Temperature ranges
  - Vibration requirements
  - Multiple screens
  - Connectors
  - Heat management
  - Power management
  - Component life-time

• Buses
  - CAN
  - LIN
  - EAVB
  - MOST
  - Flexray
  - etc
Product development

- Development time over multiple years
  - Choice of technology
  - Ensuring that Open Source components survive
Product life-time

• Expect 10+ years
• Always benchmarked to consumer electronics
  – Set-top boxes
  – Phones
  – Gaming consoles
  – Cloud services
Standardization

• GENIVI
  - Selecting components
  - Developing components
  - Adapting components
  - Build a reference base distro
  - Defining a Linux environment

• Pelagicore activities
  - Baseline Integration Team
  - System Architecture Team
  - Community Manager
  - Provides a compliant distro based on Ubuntu
Building the system

Client Application
QtQuick run-time
Resource Framework
Component
Base distribution
Building the system - example

- Media Player
- QtQuick run-time
- Phone Manager
- QtQuick run-time
- DLNA Server
- Resource Framework
- Spotify daemon
- Media Indexer
- BlueZ
- Base distribution
Resource Framework

- Monitoring
- Access control
- Aggregation
- etc
Creating the UX – Media Player

ListView {
    model: MediaSearchModel { ... }
}

AudioZone {
    name: "main"
    onAvailableChanged: {
        if (!available) {
            _oldState = player.state;
            player.pause();
        } else {
            player.state = _oldState;
        }
    }
}

Aggregated Media Search Results
Available Media Sources
Audio Zone Resource
etc
QtQuick and non-visuals

• Easy to handle non-visuals

• Simply derive from QObject instead of QQuickItem / QAbstractItemModel

• Fits well into the paradigm of managed resources
What QtQuick adds

- Designer friendly!
- Allows for quick feedback cycles – agile development
- Clearly separates UX from backends
- Easy to build a trivial simulator backend
Tested for real

- Demonstrators and evaluation platforms for both OEMs and Tier 1 companies
- Ongoing R&D projects based on the complete platform with OEMs
- Deployed to multiple hardware platforms
Our experiences

- QtQuick provides a clean interface between design and backend engineering
- A single API can be used for a multitude of UXes – only one backend to maintain
- QtQuick makes it easy to be agile, testing ideas
- QtQuick makes it easy to build next generation IVI systems
Thank you!

Visit our booth!

Ekelundsgatan 4, SE-411 18 Göteborg, Sweden
www.pelagicore.com