The Qt Company

| Whitepaper

Driving the User Interface – Trends in Automotive GUIs

Driving the User Interface –
Trends in Automotive GUIs

Sami Makkonen, Senior Manager, The Qt Company

Motor show concept cars have traditionally attracted attention through outrageous body styling, but today's advanced demonstrators place at least as much emphasis on high-tech gadgetry inside. Recent appearances by leading car brands have impressed audiences with large touch screens that centralize access to everything from cellphone contacts, multimedia content and social media to navigation, parking assistance and graphical vehicle diagnostics.

Many of the latest mid-range cars have now been launched with a graphical center-console for navigation, communication and diagnostics, while high-end brands are beginning to offer connected-car applications delivering Internet access and value-added services to their customers. Demands for such innovations are coming from several directions, as car buyers expect improved user experiences, legislators mandate systems aimed at improving road safety, and car makers seek to connect more closely with customers through electronic value-added services.

User Interface Design

One common aspect of the systems now emerging is that the touch screen head unit represents the nexus of an ever-growing diversity of input signals such as television and DVD, live video and graphics from advanced driver assistance systems, status information from various vehicle sensors, Bluetooth communications, GPS and mapping, and Internet content such as traffic updates, news feeds and social media notifications.

User interface design is critical if drivers are to gain the maximum benefit from interacting with the system without suffering distractions or information overload. Considerations for designers extend

beyond the layout and menu structure to encompass multiple ways of interacting with the system; touch, gesture and voice control will all be necessary, in addition to control using buttons on the console and steering wheel.

While the layout, graphics and control inputs must necessarily be intuitive and ergonomic, the performance of the underlying system is also extremely important. For faultless, distraction-free, timely performance on the road, the system needs to respond quickly, accurately and smoothly without delays or glitches.

Better Embedded Hardware

Designers of advanced in-vehicle infotainment systems can take advantage of high-performance embedded processors conceived for mobile applications. As such, they provide extremely high processing performance and power efficiency, support for connectivity, multimedia capabilities, and support for high-resolution displays. Also useful for designers creating advanced automotive user-interface experiences, graphics processing capabilities are extremely high, typically featuring a dedicated graphics core and support for industry standards such as OpenGL®.

There is also a great freedom to choose from a variety of suitable OS, such as embedded Linux, and real-time operating systems (RTOS) including QNX Neutrino, Green Hills INTEGRITY, and Wind River VxWorks for deterministic performance.

Software Development

With higher embedded processing performance available, and powerful automotive user-interface concepts already entering the market and setting examples for new designs to improve upon, there is a need for better software development platforms that will help designers make best use of such resources to create even more imaginative and attractive products.

Developers need a software platform that is easy and intuitive to use, allowing them to work within a graphical environment to build and evaluate complex designs quickly. Fundamentally, this requires tools vendors to deliver application development capabilities – such as layout and design tools – that are optimized for use in embedded projects.

Moreover, a flexible development framework is essential to keep up with the rapid pace of development in the automotive sector, and to allow customization to meet a variety of different requirements. In addition, optimizing the software development platform so as to minimize hardware dependencies can provide broad cross-platform portability. This can give developers the freedom to select the optimum processor for their application and migrate across platforms by simply recompiling for a different operating system as required.

Flexible Cross-Platform Framework

One application framework that is being adopted increasingly is Qt, which provides the flexibility and advanced tools that designers and programmers need to create high-quality user interfaces quickly and easily, taking full advantage of the available hardware. Qt includes an intuitive user interface technology called Qt Quick and a rich C++ class library with intuitive API and integrated development tools for GUI development and internationalization. The Qt framework also helps reduce development costs and shorten time to market as applications can be demonstrated on variety of different hardware platforms and operating systems before any actual system hardware is produced.

Qt Quick is well suited for iterative development of dynamic and animated UIs with high performance for multiple platforms. Qt Quick user interface creation is based on a CSS and JavaScript-like language called QML i.e. no C++ skills are needed for creating user interfaces and part of the application logic, knowledge of JavaScript helps. UI designers and developers can quickly iterate on the pixel-perfect UI and fine tune UI transitions and animations and see the changes in action immediately. Qt Quick also eliminates the need for separate prototyping technologies as it can be used to go from concept to design and directly to the end product, which increases productivity and shortens the time from

initial concept to end product drastically.

With Qt Quick users also have direct access to native APIs, which provides an easy way to integrate to underlying hardware and platform frameworks. Qt Quick uses hardware acceleration through OpenGL to deliver maximum performance. This can also be used for creating extra effects either by using Qt OpenGL APIs or incorporating OpenGL Shader Effects.

In addition, a complete multimedia framework is available, providing a rich feature set that enables developers to easily take advantage of a platform's multimedia capabilities and hardware. This ranges from the playback and recording of audio and video content to the use of devices such as cameras and radios.

Qt also provides Qt WebEngine, an integrated Chromium-based web browser engine, which allows developers to enrich their products by incorporating real-time web content and services into their applications smoothly and easily.

3D GUI in Action

Figure 1 illustrates a gesture-driven 3D automotive user-interface developed using the Qt framework.

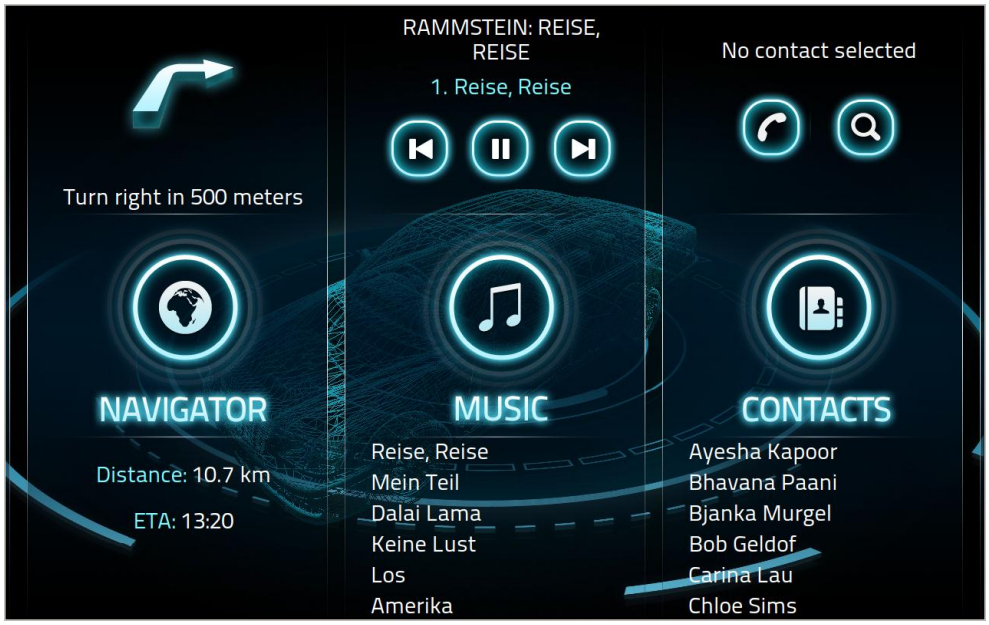

Figure 1. 3D automotive user interface concept built using Qt Commercial.

The 3D HMI concept prioritizes easy and intuitive user interaction. Allowing the use of swipe gestures eliminates the need for precise pointing. By automatically focusing on the information most relevant to the context, the system helps users to remain fully aware without becoming overwhelmed by the increasing amount of information from sources within the vehicle such as intelligent reminders, navigation, entertainment, and vehicle diagnostics.

The HMI concept is based on dedicated information-display panels, with easy navigation between panels as well as within any panel by simple up/down and left/right gestures. Each panel captures the most relevant and commonly used information, to give the driver an overview of common topics from the main level. For detailed information on any item, the user can easily drill down into the panel. In addition, the system is designed to be context aware and capable of pointing the user to the needed information.

3D technologies leveraging OpenGL through the Qt OpenGL API give easy-to-understand visual guidance for the maintenance and operation of the vehicle. For example, common items such as tire-pressure level, diagnostics information and safety notifications can be visualized via the 3D model of the vehicle. This helps drivers to assess the status of the vehicle, and any alerts from sensors, quickly and intuitively with a single glance. Users can control the interface via multiple input mechanisms including track pad and speech recognition, in addition to direct touch screen control.

The concept is also designed to be easily adapted and scaled to meet a variety of customer and application demands. It provides flexibility to integrate other sensors and sources, such as sonar, intelligent cruise control and rear-view camera. Moreover, by leveraging the cross-platform capabilities of the Qt framework, the code for this application can be ported to a large number of other hardware platforms and operating systems.

About Qt

Qt (“cute”) is a leading cross-platform application and UI development framework for leading desktop, embedded and mobile operating systems. Qt uses standard C++ and is widely used for developing software applications with a GUI and also for developing non-GUI applications with features such as file handling, database access, XML parsing, thread management and network support.

Qt’s powerful full-framework capabilities allows for the creation of highly-performing native applications as well as for hybrid development where the developer can choose which tools provide the best user experience.

About The Qt Company

The Qt Company, a wholly owned subsidiary of Digia Plc., is responsible for all Qt activities including product development, commercial and open source licensing together with the Qt Project under the open governance model.

Together with our licensing, support and services capabilities, we operate with the mission to work closely with developers to ensure that their Qt projects are deployed on time, within budget and with a competitive advantage.

http://www.qt.io