Creating the Italian Metro app with Qt Quick Components

This article explains how the "Italian Metro" Symbian app was built by using Qt Quick components

Introduction

Italian Metro is a little app that allows users to easily browse all the Italian metro lines, providing information, maps and route directions.

Application Screens

All the application screens, except for the Splash Screen, are implemented as Page elements, and the whole screen/history management is handled via the PageStack component.

All those screens have a black toolbar, anchored to the screen bottom, implemented via the ToolBar component, with two common buttons:

- a **back** button on the left side: used to navigate back in the page history, and to close the application
- a **view menu** button on the right side: used to open a menu with some common, cross-screen functionalities

The remaining toolbar space can be used by individual screens to place buttons/actions tied to the screen functionality.

Splash Screen

A splash screen is used to give immediate feedback of the application launch, while all the resources needed by the application itself are getting loaded.

The splash screen, showing the classical symbol and colors of the Italian metro sign, is implemented via standard QML elements, without any of the **Qt Quick components** in order to have the quickest possible initial load.

Loading of the app resources is implemented via the Loader element, that allows dynamical loading of QML resources.

City Selection Screen

(C) Copyright Nokia 2013. All rights reserved.

The city selection screen, that is the home screen of the app, allows users to select the desired city among the available ones. The view is implemented with the QML ListView, with custom Item elements used as delegates.

The custom delegate component display four different kind of information for each city:

- the **emblem of the city** - this is displayed by using two different Image elements: one for the emblem image itself, and another for the rounded background, giving a Nokia Belle familiar look-and-feel
- the name of the city
- the number of available lines
- the total extension, in Kilometers, of the city metro lines

A sub-item indicator, similar to the one used by the standard ListItem component, is used to give visual indication to the user of the available sub-items, in this case represented by the city metro lines.

A ToolButton placed on the toolbar allows the user to go to the "Nearest Stations Screen", illustrated below, where he can immediately check which are the nearest metro stations.

Metro Line Selection Screen

Once the user selects a city, he is presented with a screen containing two kind of informations:

- a brief excerpt of the available information about the selected city
- the list of available metro lines

(C) Copyright Nokia 2013. All rights reserved.

A Button component, placed just under the text excerpt, allows users to read the complete information that is available for the selected city.

The list of metro lines is realized with a standard QML ListView element with a custom Item delegate, composed of an Image element, displaying the metro line logo, and three Text elements used to display the metro line, boundary stations and the total number of stations.

A sub-item indicator is also used here to indicate the availability of sub-items (the metro stations).

Metro Detail Screen

By selecting a metro line, the user is presented with a screen with various informations related to the selected line:

- a brief text excerpt containing information about the line
- a map button, redirecting the user to the Metro Line Map Screen
- the list of stations of the selected line

(C) Copyright Nokia 2013. All rights reserved.

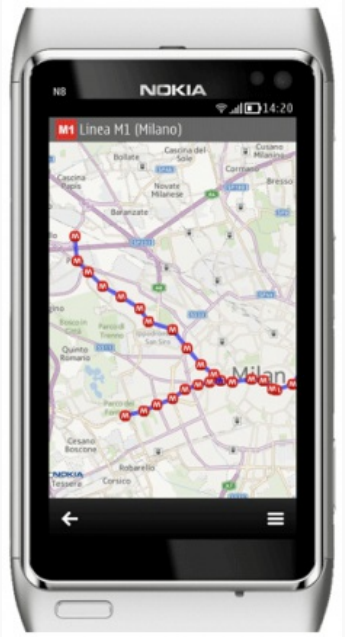
The stations list is composed of a QML ListView with custom item and section header delegates.

The item delegate custom component is composed of a Text element, used to display the station name, and a Row used to display images of the available services for that station. The Row element is dynamically populated with a Repeater that takes the services array as model and uses an Image element as delegate.

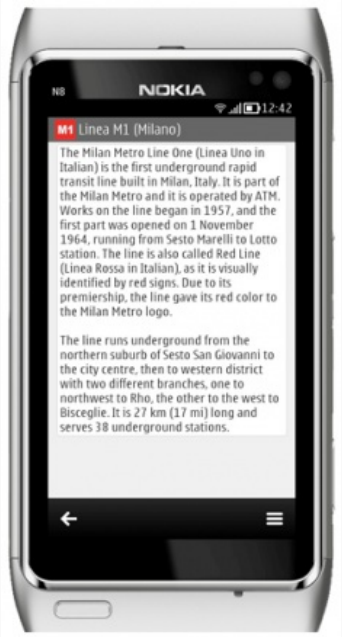
Sub-Screens

Just below the text excerpt, two Buttons are placed:

- an Info button, allowing the user to read the complete information available for the selected line
- a Map button, redirecting the user to the Metro Line Map Screen, displaying the complete map of the line



The line map sub-screen



The line info sub-screen

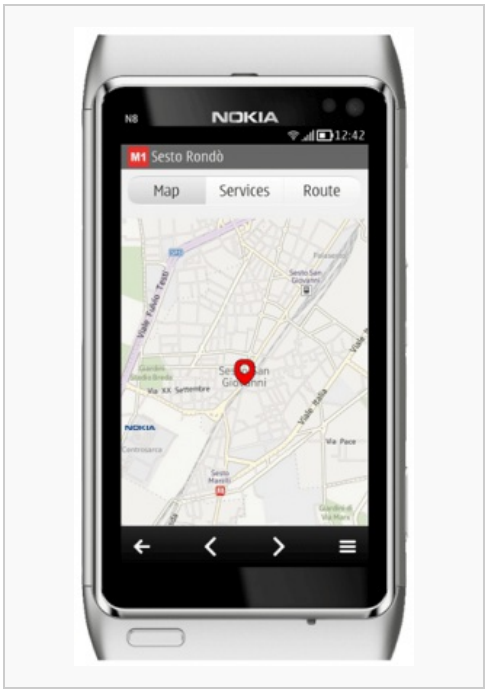
The Map screen uses the QML PinchArea element to add the pinch-to-zoom functionality to the Map element, so improving the overall user experience.

Metro Station Screen

(C) Copyright Nokia 2013. All rights reserved.

The Metro Station Screen shows three different kinds of information:

- a map, displaying the station location
- the list of available services
- the route from the user's location to the station



Map section

Services section

Route section

The ButtonRow component is used, together with three Buttons, to let the user switch among the available sections.

The map section is composed of a Map element, from Qt Mobility, with a dynamically placed MapImage.

The route section uses a ListView that gets populated with the dynamically calculated route path elements. The route evaluation functionality is implemented through the QGeoRoutingManager.

The BusyIndicator component is used to give visual feedback while the map and route are getting loaded.

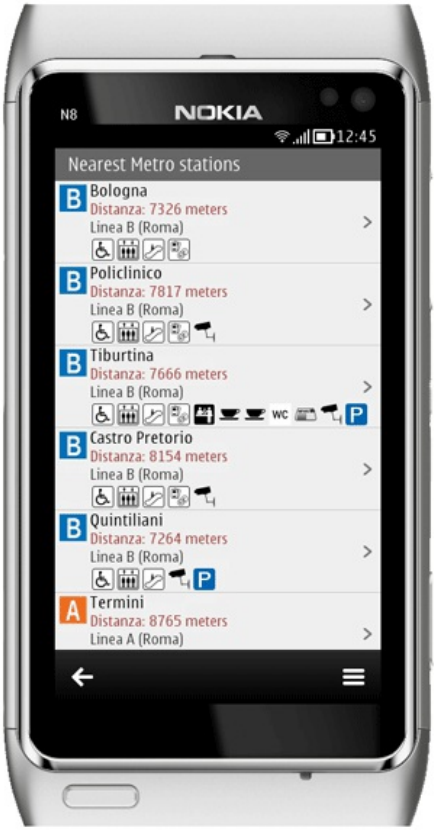
Two ToolButtons placed on the toolbar allows the user to quickly navigate to the next or previous station in the selected metro line.

Nearest Stations Screen

The Nearest Stations Screen allows the user to easily see which are the nearest stations. The section is composed of a ListView with a custom delegate to represent each of the found metro stations, built up with various UI elements:

(C) Copyright Nokia 2013. All rights reserved.

- an Image element to display the logo of the station line
- three different Text elements to display the station name, the line name, and the distance from the user, calculated in meters
- multiple Image elements to display the available services



Design tips

View labelling

Italian Metro is a multi-view app, so it uses labels to give indication to the user about its current location within the app. As stated in the Labelling Views section of the Symbian Design Guidelines, view headers should scroll away with the content, so that they don't take up space permanently.

The Italian Metro app implements this behavior, so that all the UI elements, except for the bottom toolbar, can be scrolled.

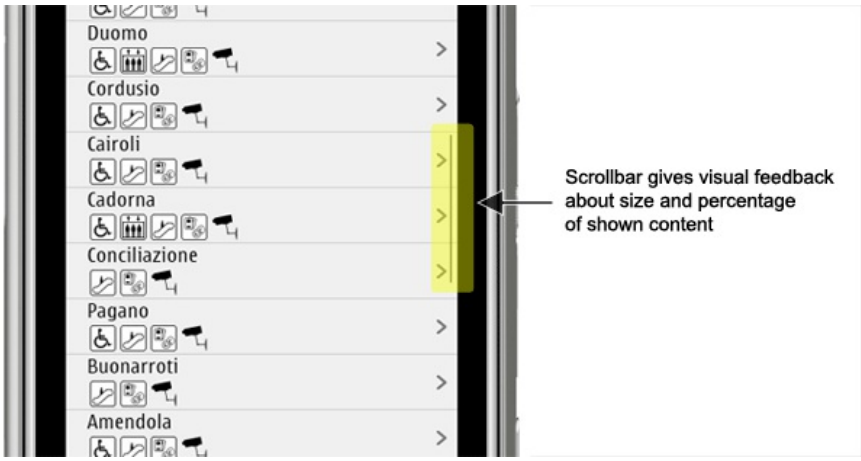
(C) Copyright Nokia 2013. All rights reserved.

Landscape support

The application gracefully supports both portrait and landscape orientation modes. When dealing with landscape mode, it is important to have the maximum vertical space available for content, so also having scrollable view headers, as explained before, also benefits this point. More information about handling rotation in a Symbian application is available in the Symbian Design Guidelines.

Scrollbars

When dealing with scrollable content, it is important to give users a visual feedback of the position of the position and percentage of of the shown content in relation to the available data.



The Italian Metro app uses the ScrollDecorator component throughout the various app screens. For more information about scrolling inside Symbian apps, you can check out the dedicated section on the Symbian Design Guidelines.

Pinch zooming

The app displays maps in various screens. A common feature that the user expects to see when playing with a map is the pinch-to-zoom feature, as described in the Symbian Design Guidelines. In QML, the pinch-to-zoom functionality can be easily implemented by using the PinchArea element, so adding value to the overall user experience provided by your application.

Implementation notes

Localization

The app is fully localized in Italian and English language, to allow both Italian citizens and foreign tourists to use the app to find their way in the Italian cities. Localization support was easily implemented thanks to **Qt Linguist**, shipped as part of the Qt SDK.

Access to network

The app is designed to use network connectivity only when strictly necessary as, for instance, when map tiles have to be retrieved from network.

In order to implement this behavior, loading of certain QML elements is delayed until they are strictly needed. Ax example are the various Map elements used in the application, that are dynamically loaded via the Loader element.

Download

A self-signed version of the Italian Metro app is available here: Metropolitane.sis (this SIS file is currently on QA on Nokia Publish).

(C) Copyright Nokia 2013. All rights reserved.

Summary

Thanks to **Qt Quick components**, the base skeleton of the app can be easily built without the need to write everything from scratch. This way, the development can focus on the app design and features, with the result of having properly behaving apps that are nicely integrated in the native device UI.

Note: This is an entry in the Symbian Qt Quick Components Competition 2012Q1

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.