Porting the Premier League app from WRT to Qt Quick

This article shows the main guidelines used to port a Symbian application from WRT to Qt, by using Qt Quick and Qt Quick Components

22 Apr 2012

Introduction

Porting applications from other technologies to Qt is a task that can be achieved by following some definite style and functionality guidelines. By using the Qt Quick Components, the porting process can be speeded up, thanks to the reusable components that offer ready-to-use functionality for several use cases and patterns.

This article highlights some of the steps and considerations done while porting the Premier League app, available on the Nokia Store (http://store.ovi.com/content/66117), from Web Runtime to Qt Quick.

UI porting

The porting process was performed with two main focuses:

- follow the Symbian Design Guidelines
- maintain the look-and-feel of the original WRT version of the app

To put into practice that, the UI porting process involved two different kinds of activity:

- replacing some HTML components with Qt Quick components
- building some QML components from scratch, when suitable Qt Quick components were not available

Porting the main UI structure

Three main UI elements characterize the overall user interface of the WRT app:

- the header, containing the app name and a reload button
- the softkeys bar
- the tab bar

(C) Copyright Nokia 2013. All rights reserved.

The following sub-sections show how each of these elements are ported to Qt Quick by following the design guidelines.

The tab bar

The app uses three different tabs to allow users to browse the different available data:

- Results
- Teams standings
- Scorers standings

The tab bar, in the WRT version, is positioned in the lower area of the screen, just above the softkeys.

Porting this component to Qt Quick is quite straightforward:

- the TabBar component provides all the needed functionality, so it is used
- by following the Design Guidelines for the Tab Bar component, the tab bar is repositioned in the upper area of the screen
- while in the WRT version the tabs only show a label, the Qt Quick version will use a combination of an icon and a label for each tab, as stated in the mentioned guidelines

The header

The design guidelines give clear indications about the usage of headers in Nokia Belle apps:

- in an app without tabs, a header can be used, and it should scroll together with the content
- in app with tabs, as the Premier League app is, header must not be used - the tab themselves must give enough indication about the content

For this reason, the WRT header is totally removed in the Qt Quick version. The reload button, originally placed in the header bar, will be repositioned in the toolbar, as shown in the following section.

The softkeys

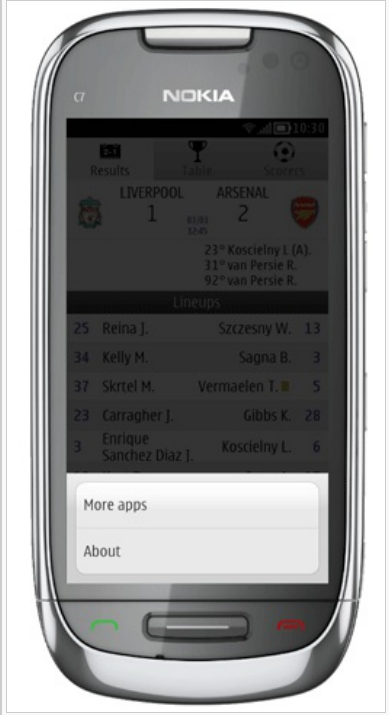
Classical softkeys in the Nokia Belle user interface style were replaced by a common toolbar, placed at the bottom of the screen, and containing buttons for the various available actions.

Converting the softkeys of the WRT version to the Qt Quick toolbar involves these steps:

- the Toolbar component is used
- the **back button** is placed on the left side of the toolbar, as stated in the guidelines for the Toolbar component
- the rest of the toolbar is used as follows: the important actions are displayed as Tool buttons on the toolbar, while the less important ones are placed in a menu by using the **View menu** button, placed on the right side of the toolbar

In the Premier League app, the important action that the user could perform in every section of the app is the **"refresh content"**, that reloads the data he is currently viewing. So, the reload button originally placed in the header bar is now placed in the toolbar by using the ToolButton component. The other actions, that are less frequently used, are placed in a Menu that is opened via the "View Menu" button placed on the toolbar.

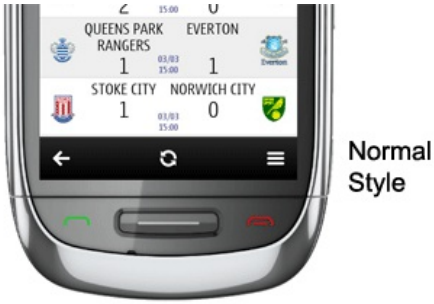
(C) Copyright Nokia 2013. All rights reserved.



The Menu

The Inverted Style

Many Qt Quick components provide a platformInverted property that can be used to change their graphic style, including the used color scheme. Given the color scheme of the app, that mainly uses light background colors, using the inverted style results in a better overall UI effect, as shown in the picture below.



Normal Style

Inverted Style

More info about the inverted style is available here.

The overall ported UI

The overall UI of the Qt Quick version, with the above considerations, is shown below:

(C) Copyright Nokia 2013. All rights reserved.

Porting the app views

The app contains three main views, accessible via the main tabs:

- a Results view, containing the list of available matches for the selected matchday
- a Table view, showing the team standings
- a Scorers view, showing the top scorers standings

The other important view, accessible through the Results view, is the Match view, that allows users to see all the details of a single match, as scorers and lineups.

The Results view

The Results view shows a scrollable list of matches, and a selection item that allows the user to change the selected matchday.

While trying to maintain the original look-and-feel of the app, the color scheme is changed to better integrate in the default design of the Qt Quick components.

The match list is composed of a standard QML ListView element with custom Item elements used as delegates for the headers and items. Custom items are chosen against the ListItem and ListHeading components, provided by the Qt Quick components, to re-build the UI style of the original WRT app.

Qt Quick components are instead used to create the matchday selection menu, that was implemented with a standard HTML **SELECT** element in the WRT version: specifically, the SelectionDialog component provides all the necessary functionality to present the user with the list of available matchdays, allowing him to pick the one he desire. The SelectionDialog is opened through a custom item placed just above the match list, as in the original WRT version.

The Match view

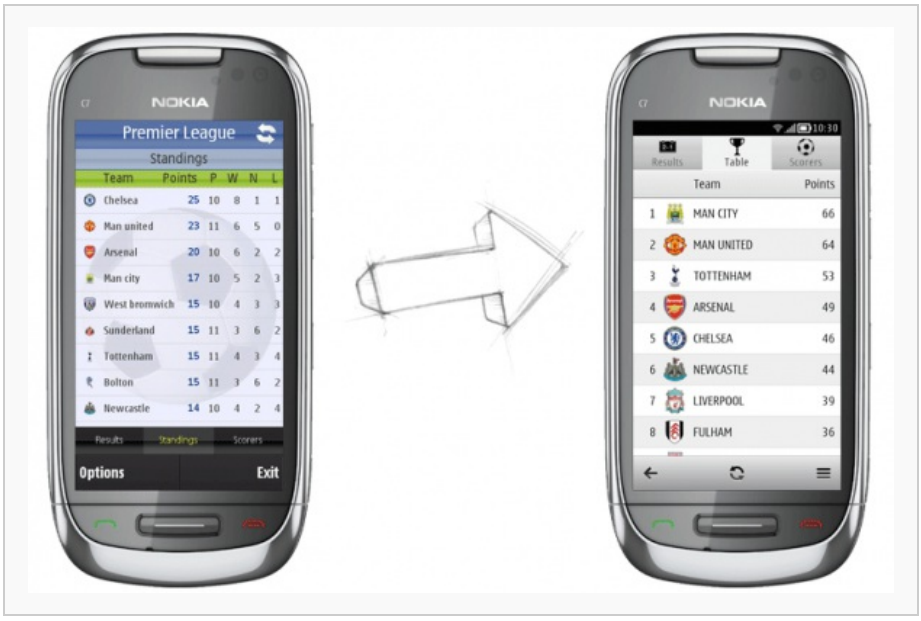
By pressing on a match in the Results view, the user is brought to the Match view, containing the main match data: team names and scores, scorers and lineups with substitutions/cards details. To maximize the readability, all the content scrolls so that space is occupied by fixed UI elements.

(C) Copyright Nokia 2013. All rights reserved.

Given the specific needs of the UI, the whole view is built by using custom QML elements.

The Standings and Scorers views

The Standings and Scorers view are structurally identical, presenting the user with a list of teams/players together with their points/scored goals. Here no Qt Quick components are used: custom QML elements are instead created to better maintain the original look-and-feel of the app.



The standings view

(C) Copyright Nokia 2013. All rights reserved.

The scorers view

The table headers, containing the labels indicating the standing content, scroll together with the content itself, so that the space used to display data is maximized.

Icon design

Following the design guidelines for the launcher icons, the app icon is redesigned using the Adobe Illustrator template. The resulting icon is shown below.

Porting functionality

The original app was built with Web Runtime, so functionality was written in the JavaScript language. Since JavaScript is also available in the Qt Quick environment, much of the app logic can be brought to Qt Quick with a little effort. Examples of functionality easily ported include:

- **networking**: the XMLHttpRequest object is available in both the WRT and Qt Quick environments, so it was used to perform network requests in both cases
- **data handling**: the same JavaScript data structures are used to hold and manage the data, so no modifications were needed on this side

View management

While the WRT version implemented a custom view management logic using JavaScript, basically showing and hiding **DIV** elements representing the different views, the Qt Quick version uses the PageStack component, that implements and provides a complete stack-based navigation model.

Final considerations

The porting process for the Premier League app was easy for various reasons, including the availability of Qt Quick components and the similarities among the WRT and Qt Quick environments.

Among the advantages of porting an application to Qt Quick, these can be mentioned:

- compliance to the Symbian Design Guidelines, so definitely a better integration of the app in the overall device UI style
- **better performances** that bring an overall better and smoother user experience to the user
- **more possibilities**: having access to all the device functionality, thanks to Qt, allows to build more powerful and useful apps

Download

The Qt version of The Football App, built following the guidelines described in this article, is available for download here: http://store.ovi.com/content/271187.

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.

(C) Copyright Nokia 2013. All rights reserved.