Farmácias de Serviço - app showcase

Farmácias de Serviço is an app for finding local pharmacies. This article explains how we ported 'Farmácias de Serviço' from Harmattan to Symbian using Qt Quick Components.

Introduction

After being developing Symbian Web Runtime apps for Symbian devices for some years we decided to take a look at Qt Quick and evaluate the benefits of using it. We decided to develop a new app from scratch and to target Harmattan because Symbian Web Runtime wasn't available in that platform (in Maemo 5 one could install it from extras-devel). The app was written entirely in QML, it used the Qt Components for MeeGo, and was designed to be simple to use.

It was migrated to Symbian in order to participate in the Symbian Qt Quick Components Competition 2012Q1.

QtComponents

The app is heavly based on Qt Components, both in Symbian and MeeGo versions, because they allowed us to focus in processing and presenting data to the user instead of spending most the time with navigation logic. Using `PageStack`, `Page`, `TabGroup` and `Toolbar` it's possible to create the basic structure for an application in a couple of hours and later to take the time to add contents to the pages.

Problem Areas

Migrating the app from MeeGo to Symbian wasn't as easy as we were expecting. Some Qt Components are different in both platforms and some don't have the same behavior. This was quite obvious when trying to get the most out of them.

- The app used some events, like `onVisibleChanged` in pages, that weren't fired in Symbian using the same code.
- The toolbar has different properties and doesn't behave the same way. In MeeGo the space between icons is calculated taking in account only the visible icons, in Symbian the place is preserved even if they are not visible.
- Toolbar look nicer in MeeGo because all the icons are separated evenly.

We had the choice to preserve the icon position when it was not needed by setting `.opacity` to 0 and allow the space to be distributed between the other icons setting `.visible` to `false`. In Symbian this can't be done. We would have to duplicate code and components, placing a toolbar inside every page, instead of having a global one.

The good stuff

Attaching events to properties (`onXXXXXChanged`) and connecting signals to functions (`tlbRefresh.clicked.connect(refreshTooltbarButtonClicked)`) makes it very easy to separate the app structure from the code the gathers and processes data and places it on screen.

Pages can use 'App Resources' defined in the basic structure without having to anticipate what kind of functionallity will have when defining it. This way pages can be developed as individual unit.

Maps - the challenge

Getting the Map component to work how we wanted was quite challeging. It provides an easy way to incorporate maps in apps but it still lacks most of the funcionalities that are available in other platforms. 'Pinch to Zoom' is not include, so we needed to place a PinchArea above it and code it. This meant that we weren't able to detected if the used clicked on a Pin or not. The Map shows Pins (`MapImage` components) in the same order that they were added. This can result in having a Pin that is located norther on top of another one this is located souther. Pins have different colors depending on how far they are from the user location (yellow - near, dark red - far). This is usefull when zooming out because Pins are just a couple of pixels away from each other.

In some devices the text written in maps (street names, ...) is to small to be confortably read. Zooming in doesn't help because the area is zoomed but the text remains the same size. We had to add an option to scale the map to 200% in order to overcome this problem.

Getting and displaying data

This was probably the easiest part of the app.

There's still a problem with `XmlListModel` that refused to parse feeds that don't have the `{{{1}}}` header. When working with third party feeds we don't have the choice of correct it so we had to get the XML using Javascript, add the header to the result and inject it into the ListModel (.xml = result).

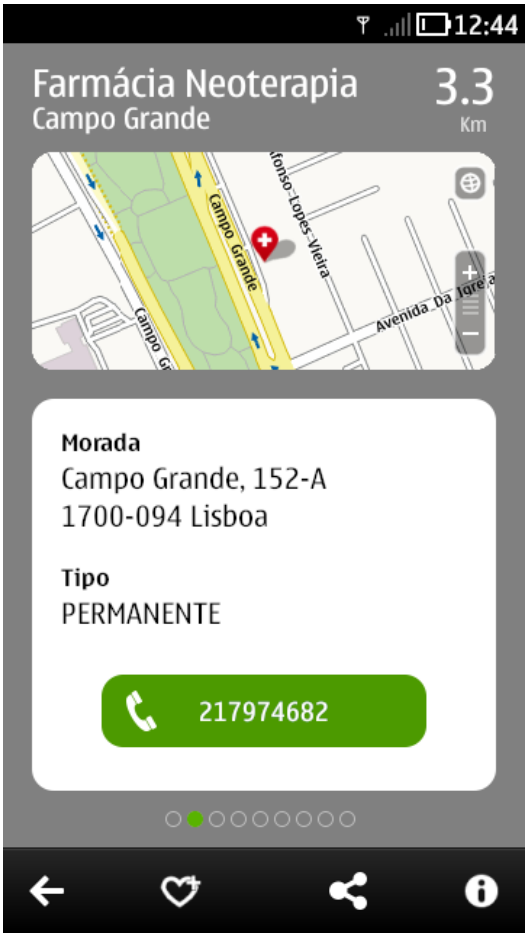
Application Icon

(C) Copyright Nokia 2013. All rights reserved.

Screenshots

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.



Download

'Farmácias de Serviço' for Symbian isn't available, yet, in Nokia Store. Until then you can get the SIS package at http://www.applicacoes.com/store/handout/symbian/FarmaciasSymbian.sis

More info

http://www.handout.pt/produto.asp?id=5-farmacias

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.