

## **Jolla Code Camp - Winter 2014 project report - Group 1**

### **Group 1**

Tatu Virta, 0388149	Netflix, Single Movie Pages, Graphics
Otto Laitinen, 0403134	Finnkino
Teemu Huovinen, 0372090	UI, Finnkino Search Filtering
Tatu Huttunen, 0358409	The idea, Blu-Ray, JSON

### **Idea**

Firstly we started to look up some open data sources from Internet. As we didn't want to create "just another weather app" and every open data provided by city of Lappeenranta were pretty dull, Tatu H. got an idea to create something more selling: everyone loves movies!

So, we wanted to create an app which could help the user to find info about Finnkino's "Now Playing" movies with Jolla mobile device. Finnkino Oy is a Finnish film distributor and the biggest cinema chain in Finland. Finnkino provides all their events and schedules in XML, so it would be easy to use with QML. We don't have Finnkino theatre in Lappeenranta yet, but according to the news Finnkino will open a new theatre in Lappeenranta in 2016.

Because just a plain Finnkino-app would have been so limited, we also wanted to add a possibility to browse Netflix's New Releases and recent Blu-Ray releases. We didn't find any sources for Finnish Netflix, but as US-version is much better, it has a larger selection and many Finns also prefer it, we decided to use it.

### **Schedule**

We got the idea right after the first tech briefing and we started to work with QML right away. We used the weather example made by Pami as a base of our first tests. We didn't do much coding on Monday, but we managed to get Sailfish IDE and version control up and running.

On Tuesday we started the actual coding. We also tried to figure out more ideas and more possibilities, but as we knew that our skills and knowledge were limited, we wanted to keep the project simple. On Tuesday and Wednesday we managed to create base pages for our project. Tatu H. successfully created function which parses the JSON BluRay data from Rotten Tomatoes and Teemu and Otto got Search Filtering to work.

By Wednesday evening we had all data collecting and data display working as intended so we were able to focus on application user interface and user experience all day on Thursday. During Thursday Teemu managed to get hang of QML as UI building tool and created the final UI while others focused on shining the rough edges and perfecting the user experience. Around seven on Thursday evening we were finally satisfied with the project.

## **Motivation**

When we designed the application our first criteria was the openness of the used data. We started to find different sources of open data while keeping in mind that we want to build an application we would use ourselves and which provides some value to user. We started filtering open data sources based on what seemed interesting and valuable to us and what data was already used in some other app in Jolla platform. First we found that Finnkino provides api in XML. Movies seemed like an interesting possibility and we started to dig more movie based data sources. We found the Netflix and Rotten Tomatoes apis to provide more content to our application. We thought that we had a pretty cohesive and useful idea for our application with those data providers and started to work on application design around those data providers.

## **Features**

Kinnostive is divided into three different categories: Finnkino, Netflix and Blu-Ray releases. In Finnkino category user is firstly asked to pick a city. After user selects a city the app displays currently playing movies in that particular city's Finnkino theatres. When user selects a movie the app shows more info

about the selected movie (runtime, genre, poster, short synopsis) and schedule for that movie in selected city for next three days.

Netflix category lists all new movie releases in US Netflix service on the first page which is displayed after user selects Netflix from main menu. User can view more detailed data about a movie by selecting it from a list. After a movie is select application opens a new page which shows movies poster, genre, runtime, IMDb rating and a short synopsis. Blu-Ray category is identical to Netflix category in features, expect the first page of the Blu-Ray category which lists new US Blu-Ray releases.

## **Tech**

We used Sailfish SDK for developing. We didn't need to use C++ at all, we managed to get everything right with QML and with few JavaScript functions. Application was tested on virtual machine and also with Jolla mobile device. For graphics we used CorelDRAW X6. With graphics Tatu V. needed to use Windows, everything else were made on Ubuntu. For version control we decided to use Mercurial, because of the horror Git spread in .NET Code Camp in December 2013.

## **Feedback**

Most of the team was rather inexperienced with developing in general, but it was still rather easy for us to fire up the SDK, start the default Sailfish project and start developing. It was also really appreciated and helpful to have a representative of Jolla to give us a presentation and also a code example. We found it really handy to be able to write JavaScript functions inside QML files.

We found debugging to be somewhat annoying. We used `console.log()` to show us our variable values, and we had to spend some time figuring out how to debug our application with it. I guess there's a better way, but we didn't find any.

Positioning of the UI elements gave us quite a headache. First we tried to have a SilicaListView that had

a PullDownMenu, then limit the SilicaListView to a part of the screen and still have the PullDownMenu on the top of the screen. This didn't work at all, but after some time we figured out to use SilicaFlickable to "dock" the PullDownMenu to the top of the screen. We also tried to position SilicaListView inside a Column to arrange it below our search field. When we did this, the SilicaListView only showed its first item (this could've been our fault, but we found no solution).

The lack of documentation ended up slowing the development quite a bit, considering our lack of experience. I understand why there's not as much documentation and Stack Overflow questions about Sailfish OS development, and in time this'll get better. I'm sure we ended up doing things the hard way, but after all the code camp is a course, and as such a learning experience, so it wasn't that bad.

## **Links**

Wiki: <http://www.codecamp.fi/doku.php/jolla2014/group1/start>

Demo video: <http://t.co/2ev0gkJzAg>

Presentation: [http://www.codecamp.fi/lib/exe/fetch.php/jolla2014/group1/kinnostive\\_rebort.pdf](http://www.codecamp.fi/lib/exe/fetch.php/jolla2014/group1/kinnostive_rebort.pdf)

Poster: [http://www.codecamp.fi/lib/exe/fetch.php/jolla2014/group1/kinnostive\\_poster.png?cache=](http://www.codecamp.fi/lib/exe/fetch.php/jolla2014/group1/kinnostive_poster.png?cache=)

Source: <https://bitbucket.org/TatuHuttunen/jollacc>

RPM-file: <http://www.codecamp.fi/lib/exe/fetch.php/jolla2014/group1/jollacc-0.1-1.armv7hl.rpm>

OpenRepos: <https://openrepos.net/content/nathorr/kinnostive>