



2016 Summer Internship

Aptoide TV Hardware Detection



Intern: Francisco Aleixo

Mentor: Gonçalo Ribeiro

Weekly Report
12 - 15 July

15 July/2016

Daily Reports

Tuesday - July 12th

- Introduction to Aptoide and Caixa Mágica.
- Meeting mentor, AptoideTV team and introduction to internship project.

Wednesday - July 13th

Task: Workspace preparation (IDE, git, wiki, etc)

- Setup of working computer.
- Setup of workspace (IDE, VirtualBox, Genymotion, Git, etc.).

Task: Task estimation

- Initial task planning and calendar planning (using Asana and Instagantt).

Task: Implement bluetooth detection

- Lightweight research about Bluetooth Detection and initial coding of a test android project.

Others

- Starting to prepare for initial internship presentation.
- Initial setup of Caixa Magica's Internship wiki.

Thursday - July 14th

Task: Task estimation

- Finishing up task planning and calendar planning using previously mentioned tools.

Task: Implement bluetooth detection

- Setting up test tablet, test bluetooth tv remote and adb via tcpip (Wi-Fi).
- Testing first android test project, and successfully testing Bluetooth detection and logging of different connectivity changes (such as device found, disconnected, connected, etc.).

Others

- Finishing initial internship presentation based on mentor's feedback and ideas.
- Finishing up Caixa Magica's Internship initial wiki entry.

Friday - July 15th

Task: Give user feedback relative to bluetooth condition changes

- Understanding and starting to code ways of giving clear feedback about connectivity changes.

Task: Implement Bluetooth detection

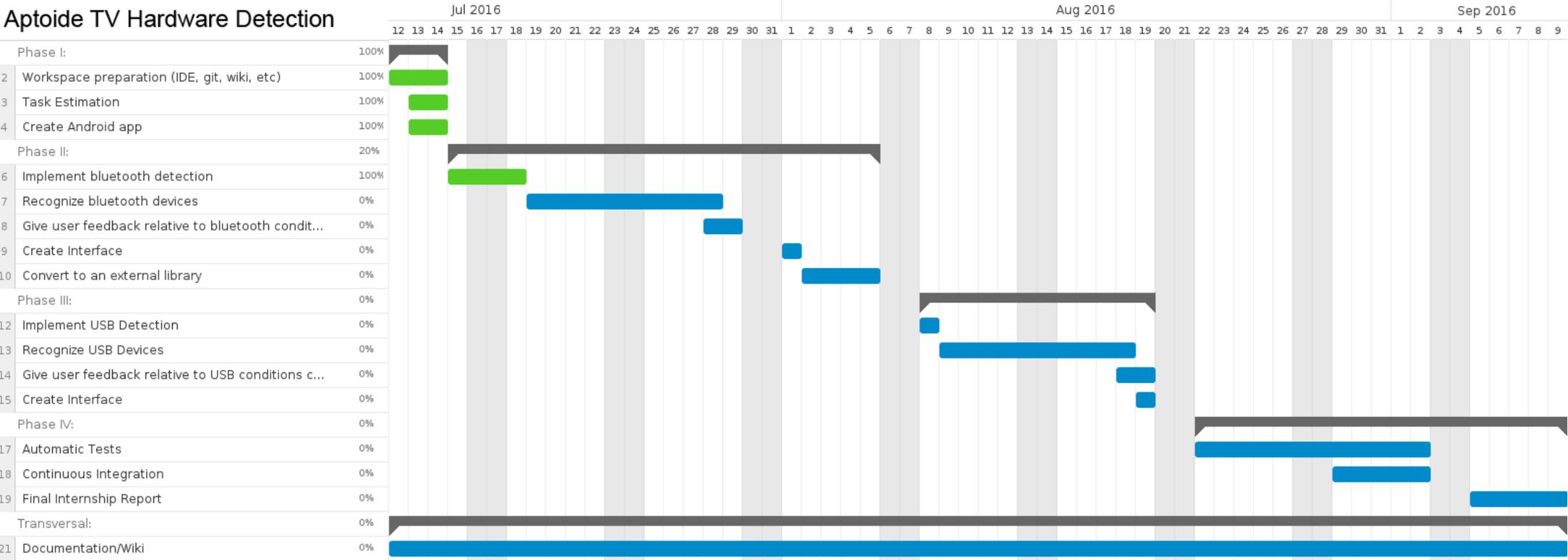
- More in-depth research and testing about Bluetooth detection, BluetoothDevice classes and possible ways to implement controller recognition (read notes below).

Others

- Initial Internship presentation and lunch/meeting.
- Elaboration of the first weekly report.

Task review

Based on planned tasks for this week, it was possible to complete them all in time and as planned. Bluetooth detection was also implemented but it still needs a few tweaks. The chart below shows not only the planned tasks for the project, but what has been accomplished so far in a simplistic but clear way.



Notes / Brainstorming

- Bluetooth detection and detecting connectivity changes seems to be fairly easy and a rudimentary implementation is already done.
- What kind of feedback is expected from the last point?

- In terms of detection and connectivity changes, the important flow seems to be:

- Found → `device.action.FOUND`
- Pairing/Unpairing → `device.action.BOND_STATE_CHANGED`
- Connect to input → `device.action.ACL_CONNECTED`
- Disconnect from input → `device.action.ACL_DISCONNECTED`

- It seems there's no clear way of recognizing a specific controller aside from a somewhat rudimentary and general approach by analyzing `BluetoothClass.Device` (see <https://developer.android.com/reference/android/bluetooth/BluetoothClass.Device.html>).

If this is the case, there might be the need to study and contact specific manufacturers in the future and possibly build a database for it.

- Further tests with Nexus Remote imply that many will have a `BluetoothClass.Device.Major` as `UNCATEGORIZED` (value 7936) which further supports the idea of having to create our own database.
- The first idea that comes to mind is to categorize them based on their MACs, since they generally have to be unique and traceable, so there might be a way to categorize them assuming the manufacturer uses some logic behind it? There might be better ways to further explore.