Contents

1 Abstract

2 Introduction

2.1 The Idea

The Digital Life Tracking App is designed to enable users to track their daily tasks and the time spent on them. To make it more convenient to trigger an activity, the start and end of an activity may be triggered in different ways. Possible activation mechanisms could be:

- Two claps
- moving the smartphone in a certain way (gestures)
- activation based on GPS position
- pressing a button

Users should be able to define their own activities and select an activation function from a list and assign it to one of their activities. The user should then be able to display his history in form of charts. A separate device (particle photon board) could also be used to trigger a specific activity.

2.1.1 The Deep Dive

For the prototype in the context of the assigment we decided to focus mainly on the detection of double clapping as an activity trigger.

3 First Prototype (Board)

Explain the state of what we achieved with the board and what problems arised.

3.1 Setup

- show board setup
- some code we used

3.2 Analysis

- Diagrams that we had in the presentation. Explain problems.

Sample rate to low, memory problems, not so easy to debug (no way of stepping through code, no UI for giving fast feedback only LEDs)

3.3 Conclusion

move to android.

4 Second Prototype (Android)

4.1 Research for clap detection in android

Java is known to have many open source libraries. For the processing of audio signals, the Tarsos library was a good choice. Among other things, this offers a ready-made percussion detection which enables to detect sudden peaks in a frequency. However, Tarsos also offers more basic functionalities, such as transforming an audio signal into the frequency domain using Fast Fourier Transformation.

In order to implement and test the necessary functionality for the state machine and the UI, the percussions detection from Tarsos was initially used to detect a loud noise.

4.2 Implementation for data analysis

Some code was soley written for getting data out of the app and was removed for later for the final app.

4.2.1 TODO Google Drive Docu

When working on the first prototype with the Particle Photon Board, the roundtrip time required to get the data from the board to a PC, where we can analyze it, was particularly noticeable. For this reason, an automatic upload of the CSV data to Google Drive has been implemented. This made it possible to quickly transfer all recorded data to a central location and analyze it from there on a PC.

4.2.2 CSV Button

- why we removed it / android API doesn't allow to have to AudioDispatcher run at the same time (but for CSV we want to record 3 seconds and not only less than a second, for detection we don't want to wait 3 seconds for getting a FFT)

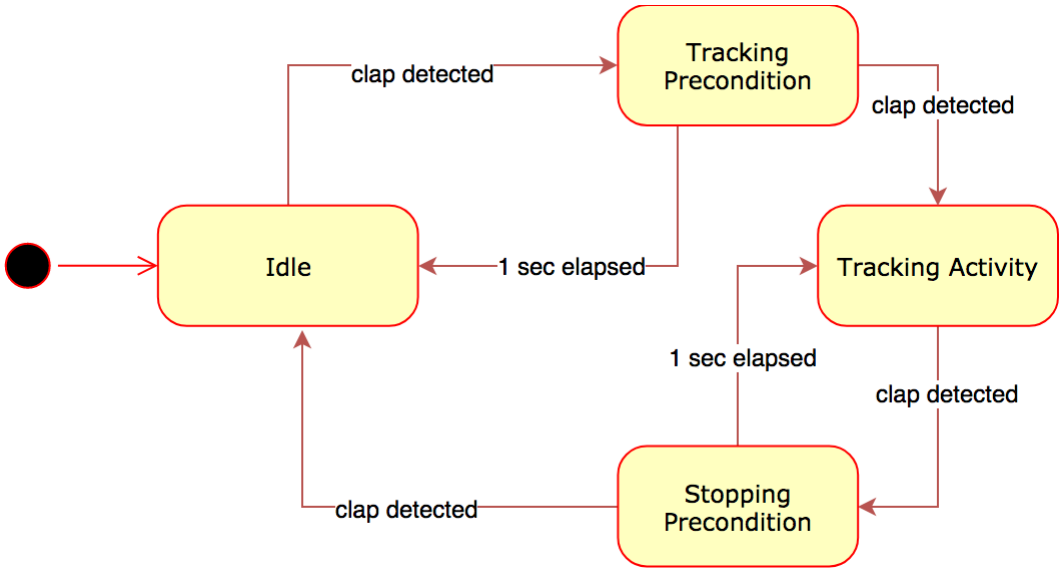
4.3 Implementation

4.3.1 Calp Detector

- Tarsos library (for FFT)

4.3.2 State-machine

A state machine was implemented to represent the different states of the app. The following diagram shows the implemented state machine.



There are a total of four states in which the app can be in.

- Idle: The initial state when starting the app and the state after an activity tracking has been completed.
- StartPrecondition: If the app is in the idle state and a clap is detected, the state machine switches to this state. When switching to this state, a timer is started which defines the time window in which the second clap must occur in order to switch the state to TrackingActivity. If the timer expires before another clap is detected, the state machine switches back to the idle state.
- TrackingActivity: After a second clap is detected while the timer of the start precondition has not yet elapsed, the state machine changes to this state and starts capturing the time by saving a time stamp.
- StoppingPrecondition: If the state machine is in the TrackingActivity state and a clap occurs, then the state machine switches to this state, which behaves in the same way as the StartingPrecondition, except that on a successful second clap, it changes to the idle state and the tracking of the current activity is ended.

4.3.3 Architecture (aus mainAc raus, UML)

4.4 Evalutation

- How reliable can our implementation detect clap.
- Benchmark
  - How many time false positives where detected (20x husten, 20x schnipsen, 20 klatschen)

- Show statistics by trying it out (maybe in different environments (loud, silent rooms, outdoors)

5 Conclusion

5.1 Current State

Refer to to evalution part above. State how difficult this was and the time needed to try out more advanced solutions (AI) was not enough.

5.2 Project Outlook

Maybe add more debug functionallity inside the App, be able to not only tweak parameters inside the code, but also with UI Controls inside the app.

Whistling detection instead of clapping.

6 Referernces

http://www.klangfuzzis.de/showthread.php?679817-Was-hat-in-etwa-wie-viel-hz

7 Aufgaben für uns noch:

7.0.1 TODO Tarsos Code rausnkopieren

7.0.2 TODO State-machine implement

7.0.3 TODO Fork vom android und unsere repo reinkopieren