Distributed Systems – Assignment 1

Florian Buenzli - Samuel Bryner
fbuenzli@student.ethz.ch - samuelbryner@student.ethz.ch
010-920-577 - 010-915-023

October 12, 2012

Abstract

Hallo Welt ;)
(Os simply 'Hello assistant'. Have fun playing with our software.)

1 Sensor

Not much is to be said about the first application. The implementation is quite straightforward and the problem lies more in learning the complex and extensive Android API. It was however a good introduction to programming with a Google Toy.

2 AntiTheft

This application locks the position of a smartphone. The user can put it on a table and start the service. As soon as the phone is moved even slightly, an alarm goes off. Multiple preferences exist to specify the exact behaviour such as a delayed alarm with the option to disarm it.

2.1 Application Usage

The app first has to collect a few seconds of data to learn how its stationary position 'feels'. (The length of this can be set in the options with 'Learing Time') After this, it is locked and watches for changes in orientation. The measured distance from the base orientation is displayed as a red graph which is normalized to the threshold (center line).

If 'Alert Immediately' is activated, the phone starts the alarm as soon as a single value is over the threshold whitout any possibility for the user to turn the alarm off. This makes for fun games as the phone now reacts to very small perturbations.

If 'Alert Immediately' is deactivated, the whole alarm system changes in two ways: First, there is a timeframe in which a certain amount of values have to be over the threshold (The length of the frame can be set via 'Significant Time Window' and the percentage of over-threshold-values can be set with 'Significant Percentage').

Second, when the alarm goes off it does so delayed. Only a dialog window pops up giving the user the chance to kill the alarm before it does any harm.

The alarm has two components: A short vibration and beep which always go off and a longer, louder alarm sound which can be deactivated ('Obnoxious Alarm') and tries to circumvent any silent mode or connected headphones.

2.2 Interesting Features

2.2.1 Sensor Analysis

The main and most fun feature is certainly the advanced sensor processing. It relies on a thing called the Mahalanobi Distance, an extended version of a gaussian estimator which takes directional differences of the variance into account and is scale and rotation invariant. This gives a high level of stability and makes it possible to detect even small movement such as a tap on the table or movement of an object on which the phone rests. You can therefore use it to secure not only your phone but also other possesions of yours. Or you can tape it to a door and alert if somebody opens (or closes) it.

2.2.2 Graph

A simple graph displays the raw values of the accelerometer (Acceleration in g in X-, Y- and Z-direction). It also shows the progress of the learning phase and displays the measured disturbance in comparison with the threshold.

2.2.3 A Bunch of Options

A bunch of options exist to fine tune the behaviour and allow different usage scenarios. Most of the settings are described in 'Application Usage'. Other preferences include a way to disable the alarm completely to better calibrate the threshold (Display the measured values with 'Show Sensor Value').

2.2.4 Loud Alarm

An optional alarm allows a very audible alert in case of movement. The program tries to disable any headphones or mutes and tries to use the inbuilt speakers of the phone. The only possiblity of deactivating it once it went off is to shutdown the application via taskmanager or letting the alarm sound finish.

2.3 Problems During Development

Certain systems gave more problems than others while implementing them.

2.3.1 Stable Service

The whole alarm is implemented as an Android Service which runs somewhat independently of the main Activity and should continue to do so when the user switches to another application. In order for that to work, a bunch of specific Android flags have to be set. Otherwise the Service gets restarted every time the phone is rotated. Also problematic and so far unsolved is that the sensors on some phones stop reporting when the screen is turned off.

2.3.2 Communication with Service

Working fine but in my opinion quite ugly is the communication between Service and Activity. There should be a nicer way than just giving each class a reference to the other. While this is very simple and posed not much problems there is room for inprovement in the software engineering side of this project.

2.3.3 Drawing

This was actually not a problem, as it is trivial to derive from a View and override onDraw. Other groups used SurfaceView for this which is much more powerful but also a lot more complex to set up and use. As the graph is not a highly complex game, the performance penalty of using a normal View can certainly be coped with.

2.3.4 Numerics

A lot of work went into the linear algebra behind the sensor data anlysis. What began as a simple, straighforward analyser quickly became a complete singular value decomposer based on QR-decomposition. Unfortunately, the chosen algorithms were not really numerically stable (in spite of some modifications to account for them) and gave problems with small covariances. This led to a lot of NaN's and Inf's and undefined behaviour, so we used an existing linear algebra library with optimized and more general algorithms. This would also make future extensions easier (eg. adding more sensors).

2.4 Possible Enhancements

2.4.1 More Sensors

The only sensor on our phone which gives high resoltion data for small changes is the accelerometer. However, including data from other sources (eg. a gyro) would of course be an interesting option.

It is also important to note that, while suitable for detection of movement, GPS or Wifi-triangulation does not nearly have the neccessary resolution for this kind of application. One could of course extend the application to include a second mode which locks the phone not in one extremly specific place but, say, in a room.