AlphaTrainer - an Android based neurofeedback system using low-cost consumer Brain-Computer Interfaces

Martin Poulsen and Pelle Krøgholt

IT University of Copenhagen

Supervisor: John Paulin Hansen

M.Sc. Thesis

December 2013

Abstract

Neurofeedback training denotes training brain activity towards some desired state by means of a real time feedback. This is a somewhat established clinical practice but consumer neurofeedback systems remain rare. This thesis explores the feasibility of a neurofeedback system comprising of a low cost consumer BCI and a mobile device through the design, implementation and evaluation of AlphaTrainer. It is concluded that the MindWave Mobile BCI is feasible for building an alpha feedback system based on its ability to measure alpha wave activity. A user evaluation of AlphaTrainer verifies the feasibility of performing neurofeedback training in an everyday context. However, the need for a longitudinal study of AlphaTrainer's training effect remains.

Acknowledgements

First and foremost, we would like to thank our supervisor John Paulin Hansen for outstanding supervision throughout the entire process.

Secondly, we would like to thank the research team around PitLab here at ITU and the Department of Applied Mathematics and Computer Science at DTU for guidances and support.

Thirdly, we would also like to thank the people at Neurosky, Op Innovations and the EEG-lab mailing-list for help and guidances.

Finally, we would like to thank all the people who participated in the experiments throughout the thesis process and in the final user evaluation.

To the brain...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Neurofeedback, stress and brain-computer interfaces

Neurofeedback training (or neurofeedback therapy) is based on giving real time feedback on brain activity. This feedback enables the brain to navigate towards some desired brain state. Certain brain states have been shown to be positively correlated with cognitive performance, de-stressing etc. Such brain states form the goal for the neurofeedback.

Stress - a big and growing problem to both society and individuals - is one of the conditions neurofeedback therapy is targeting within a clinical context. For example, it is used within the US army as treatment for veteran soldiers suffering from post-traumatic stress disorder (PTSD) [69]. However, neurofeedback systems targeting consumers remain rare and expensive due to the required specialized hardware - in order to sense brain states, a neurofeedback system includes a Brain-Computer Interface (BCI). Though most BCIs are expensive and aimed at professionals, within recent years a number of BCIs targeting consumers with prices starting from $ 50,- have emerged. Interaxon's *Muse* [1] BCI exemplifies the new generation of discrete consumer BCIs (Figure 1.1).

The combination of the emerging consumer interfaces and the prospects of neurofeedback motivates our hypothesis.

[1] http://www.interaxon.ca

Figure 1.1: Muse consumer BCI (Image courtesy of Interaxon).

1.2 Hypothesis

We hypothesize that it is feasible to build a neurofeedback system comprising of a consumer BCI and a mobile device which will enable neurofeedback training in an everyday setting.

To test our hypothesis, we have set the following goals:

G1

Evaluate relevant consumer BCI's feasibility for neurofeedback training.

G2

Design, implement and evaluate AlphaTrainer - a system enabling neurofeedback training in an everyday setting.

If such a system shows feasible, it would enable wide adoption of neurofeedback training in eliminating the obstacle of expensive hardware. This could move the neurofeedback practice out of a clinical setting and into the homes and workplaces of people motivated to reduce their stress.

1.3 Method

The vision and goal of AlphaTrainer - to enable neurofeedback training in an everyday context - is rooted in the ideas of the early pioneers of ubiquitous computing (ubicomp) who envisioned invisible computing, prototypes and the move away from desktop computers into devices that "*weave themselves into the fabric of everyday life*" [65] [66] [67].

In short, our method is to design, implement, deploy and evaluate AlphaTrainer. This approach is inspired by later adapters of ubicomp stressing the importance of deploying working systems for

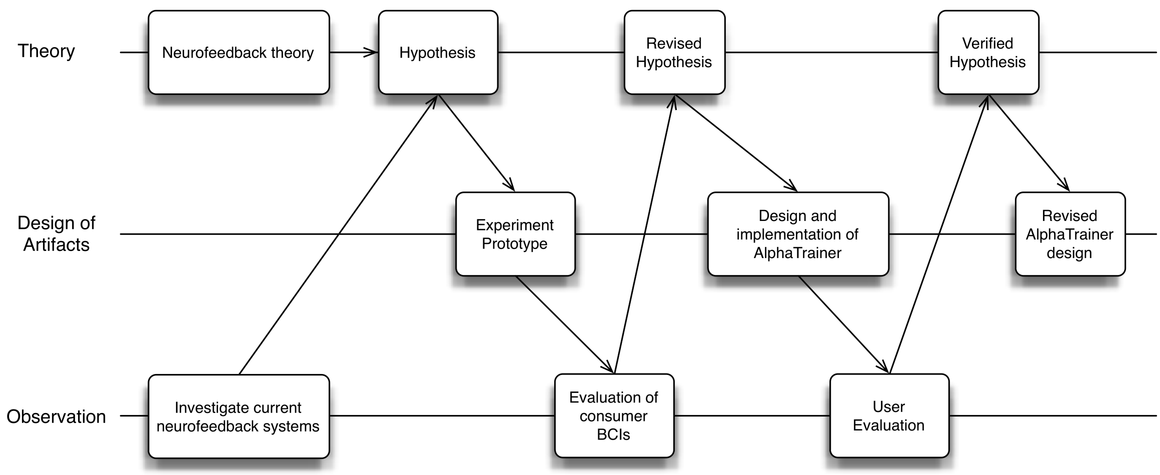


Figure 1.2: Mapping of the thesis activities and process - applying the triangulation framework proposed by Mackay and Fayard [41].

real usage. For example, Bardram and Friday argue that "*... the most valuable lessons to take from looking at successful ubicomp systems is the need to mature the system through actual use*" [8]. By deploying AlphaTrainer for actual use in an everyday context, we are able to learn about the system and the usage "*in situ*" which can not be investigated in a lab or by means of lo-fi prototypes.

The process of building AlphaTrainer involves several activities. Which can be mapped and understood through a framework proposed by Mackay and Fayard [41]. The framework explains how HCI research can benefit from triangulating across science and design disciplines while continuously producing artifacts. Figure 1.2 outlines the major activities of this thesis. The arrows between activities show when output of one activity has been fed into another thus mapping how activities have benefited from each other across disciplines.

1.4 Thesis overview

We outline the structure of this thesis below based on the thesis activities mapped in Figure 1.2.

In Chapter 2 we establish a broad overview of BCIs and neurofeedback training. The chapter describes related consumer products and related works within the area of consumer BCIs and neurofeedback. This is the foundation of our hypothesis and it frames *Neurofeedback theory* and *Investigate current neurofeedback systems.*

Chapter 3 describes the design of the *Experimental prototype* and the *Evaluation of consumer BCIs* within the designed experiment. This evaluation addresses our first goal - **G1** (Section 1.2). The gained knowledge of the BCIs capabilities leads us to a *Revised Hypothesis* which enables us to develop the system design in the next chapter.

The *Design and implementation of AlphaTrainer* is split in two chapters. Chapter 4 outlines the design activities, decisions and process that lead us to our final design, while Chapter 5 covers the implementation of the AlphaTrainer system.

In Chapter 6 we evaluate the system through a *User Evaluation of AlphaTrainer*. The chapter addresses directly our second goal - **G2** (Section 1.2) and leads us to a *Verified Hypothesis* and a *Revised AlphaTrainer design*.

Finally we make out the conclusion and outline future works - Chapter 7.

1.5 Limitations

This thesis does not attempt to make any clinical claims about AlphaTrainer's efficacy regarding stress treatment. Nor does it address the security aspects of data management which would be required for a system for clinical use.

Rather, it explores whether a neurofeedback system can actually be build using a mobile device and a low-cost consumer BCI. Furthermore, it explores through real world deployment of a prototype whether it makes sense for users to perform neurofeedback training in an everyday context.

Chapter 2

Background

In this chapter we start out briefly explaining what EEG is and how a Brain-Computer Interface (BCI) works including typical approaches to data processing and classical applications. We then move on to describe relevant consumer BCIs. Finally, we outline neurofeedback training and how it relates to stress reduction. We conclude the chapter with an overview of neurofeedback systems which are either aimed at consumers or using consumer BCIs.

2.1 BCI / EEG

The brain consists of billions of neurons. Communication between neurons is manifested in electrical signals. Electroencephalography (EEG) measures this electrical activity along the scalp. This is a well established technique from the 1920s. EEG has a low spatial but a high temporal resolution which makes it ideal for recording changes in brain activities in response to events. A BCI takes brain activity - for example in the form of an EEG signal (which is the case for all BCIs mentioned in this thesis) - as input [61]. For example, Figure 2.1 shows an EEG signal with visible alpha waves.

Figure 2.1: EEG signal with alpha waves marked with black boxes.

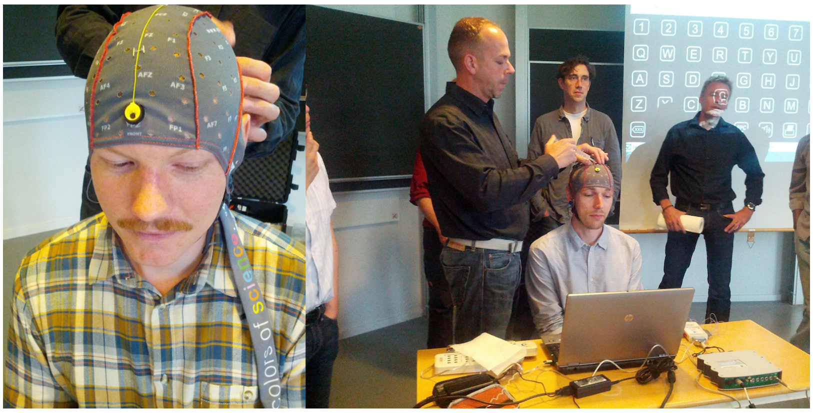


Figure 2.2: Martin and Pelle wearing a gamma-2-cap mounted by Gunther Krausz from g.tec

Brain activity is not the only source of electrical signals along the scalp. Muscle activation also relies on electrical signals, the measuring of which is called electromyography (EMG). Eye movement furthermore discharges electricity due to the eyes dipole properties - measuring this signal is called electrooculography (EOG). In the context of measuring EEG, EMG and EOG are typical noise artifacts [43].

EEG is measured either with stand-alone electrodes or electrodes attached to a cap. Examples of high fidelity EEG BCIs are the *gamma-2-cap* from *g.tec*[1] and *Easy Cap* from EASYCAP [2]. Some caps can even have up to 256 electrodes mounted [61]. Caps need mounting preparations with gel to improve the conductivity between scalp and electrodes. We have tried out the *gamma-2-cap* during a BCI seminar in Aalborg 2013 (Figure 2.2). A hi-fi EEG based BCI cost around 10-15.000 Euro and includes, for example, a cap with electrodes, cables and an amplifier.

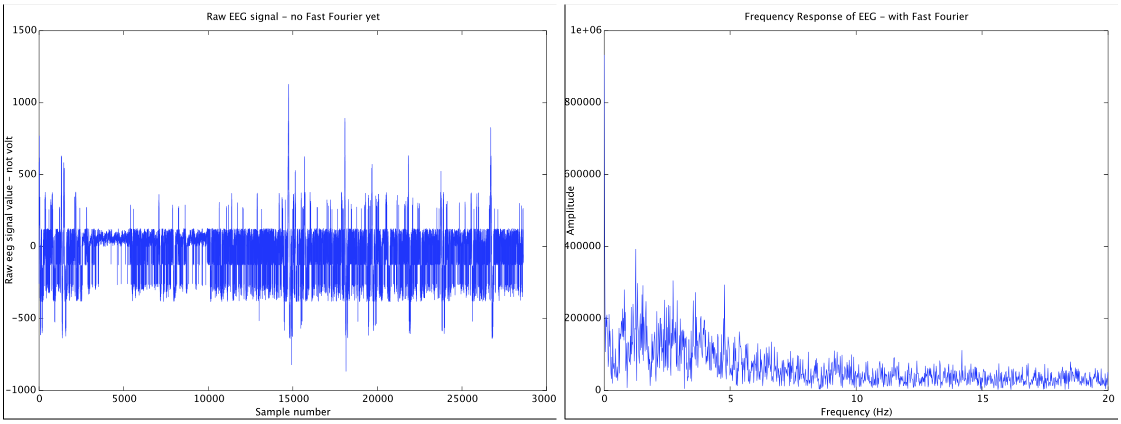
The EEG data is usually processed and analyzed off line with tools like EEG-lab [20] or FieldTrip [52] which are open-source plug-ins to MATLAB [3].

2.1.1 Classic BCI applications and techniques

There are different approaches to processing a raw EEG signal. Some of these and their typical applications are briefly covered below.

Evoked Potentials correlates visual/auditory stimuli with EEG responses. When an event of significance is perceived, the brain fires certain action responses. One widely used response is the P300

[1] http://www.gtec.at
[2] http://www.easycap.de/easycap
[3] http://www.mathworks.se/products/matlab



(a) Single channel raw EEG with 512 samples per second.

(b) Single channel EEG after Fast Fourier transform (FFT) has been applied.

Figure 2.3: Raw EEG and Fast Fourier transformed EEG.

which manifests itself as a peak in the EEG signal 300 ms after a stimulus - for example a flash of an image. In the *intendiX* [4] P300 speller application from *g.tec* different letters are flashed for user while the P300 action response is used to determine which letter the user wants to select.

Motor imagery is another classic approach to BCI in which an imagined movement of a body part causes motor cortex activity which is detected by the BCI. In this way imagined movement can be used for example to control wheel chairs and other vehicles [46]. This technique has been used in gaming as well, e.g. for trigger activation [18]. Motor imagery requires spatialization (localization) of brain activity especially within motor cortex. This presents a challenge for EEG based BCIs since they have a low spatial resolution.

Classification of EEG signals are widely used within BCI applications. For example, it has been used for unique identification of a person for authentication purpose [57]. Various research groups uses classifications to predict epilepsy attacks [58][47]. Others do pattern matching on walking motions for assisting in rehabilitation after strokes [36]. This approach has also been used to classify: (*i*) emotions like joy and anger; and (*ii*) human expressions like a happy facial expression or a mental mood [21] [11] [30].

Frequency analysis is another technique for processing EEG data. Neurons are organized in networks and communication among them is always ongoing in oscillatory patterns. Frequency analysis estimates the power of each frequency component. One common approach to frequency analysis is to apply Fast Fourier transform (FFT) - a simple example is plotted in Figure 2.3. When applying

[4] http://www.gtec.at/Products/Complete-Solutions/intendiX-Specs-Features

Brainwave Type	Frequency range	Mental states and conditions
Delta	0.5Hz to 3.5Hz	Deep sleep
Theta	3.5Hz to 8Hz	Falling asleep
Alpha	8Hz to 12Hz	Relaxed awake state (dominant with eyes closed)
Beta	12Hz to 30Hz	Mental activity, attention, concentration
Midrange Beta	16Hz to 20Hz	Thinking, aware of self & surroundings
High Beta	21Hz to 30Hz	Alertness, agitation
Gamma	30Hz to 100Hz	Reflects the mechanism of consciousness

Table 2.1: Generalized frequency bands

FFT we go from a time domain into a frequency domain as can be seen on the x-axis values of the plots before and after FFT. Frequency analysis is often used in conjunction with other methods of analysis - for example to extract features for classification [61]. It is also used stand-alone either for neurofeedback or in research aiming to correlate certain frequency patterns with some condition or cognitive task. This is very typical within EEG research exemplified by a study showing that "*... high resting theta power in healthy older adults is associated with better cognitive function*" [24].

Frequency analysis is interesting due to the correlation between frequencies and mental states [27] [54]. A rough overview is lined up in Table 2.1.

The hi-fi BCIs are getting mobile. This trend is exemplified by a mobile version of the *Easy Cap* and helmets with built in EEG sensors for soldiers [45]. Another branch of BCIs that have come far in getting mobile are the consumer BCIs as described in the next section.

2.2 Consumer BCIs

Within recent years consumer BCIs have emerged and moved BCIs outside the laboratories. An early consumer BCI was the Neural Impulse Actuator (NIA) [5] released in 2008 featuring a three forehead sensor configuration and connectivity through a desktop box with cables. NIA was intended primarily for gaming and cost around 100 USD (it is not in production any longer). Consumer headsets today typically offer additional sensors (accelerometer, gyroscope, etc) and wireless connectivity. An overview of current state consumer BCIs is presented in Table 2.2.

These headsets are interesting to our project. Relevant features and how they have been used in research is lined out in the next sections.

[5] http://ocz.com/consumer/company/newsroom/news/ocz-announces-availability-of-vista-64-bit-drivers-for-the-nia-neural-impulse-actuator-gaming-peripheral

Feature	Emotiv EPOC (Research)	MindWave Mobile	TrueSense Kit (OPI)	Muse	Emotiv INSIGHT
Raw EEG	No / Yes	Yes	Yes	Yes	Yes
Electrodes	14	1	2	4	5
Locations	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4	–	–	AF3, AF4, TP9, TP10	AF3, AF4, T7, T8, Pz
Sensors type	Wet	Dry	Dry/gel	Dry	Dry
Sampling rates	128Hz	512Hz	512Hz	100-600Hz	128Hz
Protocol	USB Dongle	Bluetooth	ZigBee	Bluetooth	Bluetooth
Power	12 hours	6-8 hours (AAA bat.)	18 hours	10 hours	4 hours
Off Line recording	N/A	N/A	Memory chip	N/A	microSD card
Extra Sensors	Gyroscope	None	Accelerometer, temperature	Accelerometer	Accelerometer, magnetometer
SDK	OSX, Windows	Android, IOS, Windows, OSX	Linux, Windows, OSX	Android, IOS, Linux, Windows, OSX	Android, IOS, Linux, Windows, OSX
Price	300/750 USD	100 EURO	40 EURO	269 EURO	300 USD
Released	2009	2011	2013	Ann. 2014	Ann. 2014

Table 2.2: Listing of consumer BCIs compared by selected features.

(a) Emotiv EPOC. The image show USB Dongle, electrodes and the EPOC headset

(b) NeuroSky MindWave Mobile headset.

(c) TrueSense Kit. The image show USB controller (w. ZigBee receiver), memory module and TrueSense sensor band.

Figure 2.4: Images of 3 consumer BCIs.

2.2.1 Emotiv EPOC

EPOC has out of the box a desktop SDK and a set of tools aimed at gaming. The closed source SDK provides detection of emotions, expressions, cognitive states and more [6] (Figure 2.4a).

There is also a research edition of the EPOC headset which can record raw EEG. It comes with the *TestBench* desktop application for recording and viewing the raw EEG data. TestBench can process EEG into various frequency bands and the raw EEG data can be exported in an EDF-format (multichannel biological and physical signals) [7] (Appendix *Emotiv EPOC TestBench*).

Before using the EPOC headset, the user has to moist each of its 14 electrode in a saline solution and then attach each electrode to the headset. This preparation took us about 10-15 minutes when some routine was achieved.

The EPOC has no SDK for mobile devices, but has been hacked for mobile usage in conjunction with the USB dongle - we return to this in Section 2.4.

[6] http://emotiv.com
[7] http://www.edfplus.info

EPOC seems to be the consumer BCI that appears most frequently in research papers. An overview of its application within research is given below.

The *FlyingBuddy2* uses motor imagery to make it possible for a disabled person to steer a Drone with the future perspective of steering a wheel chair [72]. A group in Spain uses the out of the box SDK classified facial expression (based on EMG) such as open and close clinch combined with EOG data to steer a tractor [26]. Again using SDK classifications, an emotion based chat application has been build featuring avatars that changes their expression from angry to happy based on the emotional state of a person [70]. In a recent M.Sc. thesis the EPOC was used for a brain wave biometrics authentication system [53]. The *NeuroPhone* project used a P300 approach to make phone calls on a smart phone - however the EEG processing was performed on a laptop [16]. Finally, EPOC has also been used in a human-robot interaction study where they used EEG to classify human satisfaction of the interaction with a robot [23].

EPOC has also been used by media researchers at the Danish Broadcasting Corporation (DR) as a supplemental tool to qualitative interviews and questionnaires. Throughout the video screening of a TV Drama production, the screening participants' brain states were measured in terms of EPOC SDK values such as excitement, frustration and attention. In an interview with Harddisken (a DR radio program about technology), Jacob Lyng Wieland - in charge of the experimental usage of BCI during video screenings - reported that they had skipped using the EPOC headset because it was too cumbersome to use [8].

2.2.2 NeuroSky MindWave Mobile

NeuroSky MindWave Mobile (MindWave) [9] offers desktop and mobile SDKs (IOS and Android). The closed source SDK features frequencies processing and analysis outputting values for the level of "attention" and "meditation" [50] (Figure 2.4b). The SDK also provides information about eye blinks and a number of frequency bands which we previously have lined out in Figure 2.1. The MindWave SDK outputs the following frequency bands: delta (0.5 - 2.75Hz), theta (3.5 - 6.75Hz), low-alpha (7.5 - 9.25Hz), high-alpha (10 - 11.75Hz), low-beta (13 - 16.75Hz), high-beta (18 - 29.75Hz), low-gamma (31 - 39.75Hz) and mid-gamma (41 - 49.75Hz) [51].

The headset is easy to use and the SDK includes a simple Bluetooth API that seamlessly supports device connectivity. Due to its connectivity options and SDK signal processing, the MindWave

[8] http://www.dr.dk/radio/player/ondemand/legacybyrid/1588020#!
[9] http://www.neurosky.com/products/mindwavemobile.aspx

Mobile requires little effort to embed in a prototype. This has been done, for example, in a recent paper by Marchesi. He uses MindWave in the BRAVO project to detect attention among school children in an e-learning setting. If a child's attention level is under some threshold, it its reported to the other children who are encouraged to offer their help [44]. In another study, a research group uses MindWave to measure attention during an online game. They specifically look at the attention levels provided by the SDK versus self reported attention levels among a group of participants. They conclude that the self reports and the SDK values are correlated [56]. Another paper uses the attention and meditation SDK values to examine the stress levels among participants while performing various tasks. It concluded that the MindSet [10] was able to measure an increase in stress induced by the tasks performed (Stroop test, Tower of Hanoi) [19]. In [71] the brain state - defined in terms of EEG frequency composition - of a test subject driving a car is measured by a predecessor to the MindWave. Raw EEG data is recorded to a mobile phone via bluetooth and its frequency composition is analyzed offline. Interestingly, the results show a change in the brain wave frequency pattern when the driver performed, for example, a phone call. Finally, in a M.Sc. thesis, classification of the raw EEG data from the MindSet is used to control a snake-like game aimed at children [37].

MindWave comes out of the box with a mobile application named *Brainwave Visualizer* which let its user inspect the current levels of the 8 frequency bands supported by the SDK. The app also provides simple neurofeedback by letting its user control the flying height of a ball by the SDK "meditation" value or the intensity of a flame by the SDK "attention" value. The same approach to neurofeedback is used in the third party app *Transcend* by Personal Neuro. During meditation the user can get a flower to grow by increasing the "meditation" SDK value [11].

2.2.3 TrueSense Kit

TrueSense Kit is the newest, cheapest, most portable headset. It comes with OPI Console, an open source desktop application for recording and viewing raw EEG data and analyzing sleep, meditation etc. from recorded EEG. It also enables exporting data as EDF-files for further processing and analysis in other applications. The OPI Console also offers sleep analysis and yoga performance analysis [12] (Figure 2.4c, Appendix *TrueSense Kit console*). The TrueSense Kit sensor(s) can be

[10] The experiment used a MindSet headset (the generation before MindWave but with the same chipset and electrode) [19].
[11] Android app: https://play.google.com/store/apps/details?id=com.personalneuro.transcend
[12] http://op-innovations.com

placed on various parts of the body for measuring blood flow, heart rate, body temperature and body movements.

TrueSense Kit records either directly to an internal memory module or transmit data over ZigBee radio to the OPI Console through a USB receiver. The sensors can be combined in a multi sensor configuration attached various places on the head or body.

TrueSense Kit provides no immediate mobile device connectivity but the OPI Console application can likely be ported to Android since it is build with the QT framework [13]. Another approach would be to build a native C++ Android module from the TrueSense Kit C++ SDK. Since few Android devices currently support ZigBee out of the box, this would require an external receiver board.

TrueSense Kit is not yet covered in any papers despite its support for flexible experimental setups. TrueSense Kit was warmly received by the quantified self community at the yearly QS conference in Amsterdam 2013 [14].

2.2.4 Future headsets

New consumer BCIs are about to arrive, for example *Muse* [15] (as briefly mentioned in the Introduction Section 1.1) and *Emotiv INSIGHT* [16]. These new headsets have some characteristics in common which seem to be representative for the new generation of consumer BCIs:

- they support Bluetooth
- they use dry electrodes
- they are discrete and comfortable to wear

An interesting fact is that both of these headbands are crowdfunded. Muse raised 287,472 USD in 2012 from an unknown amount of supporters at Indiegogo [17]. EPOC Insight had pledged 1,643,117 USD by the end of September 2013 from nearly 5000 people on Kickstarter [18]. This trends an interest in low cost consumer BCIs and exemplifies pretotyping by presenting and selling the product before it has actually been build [32].

[13] http://qt-project.org
[14] http://quantifiedself.com/conference/Amsterdam-2013
[15] http://www.interaxon.ca
[16] http://www.kickstarter.com/projects/tanttle/emotiv-insight-optimize-your-brain-fitness-and-per
[17] http://www.indiegogo.com/projects/muse-the-brain-sensing-headband-that-lets-you-control-things-with-your-mind
[18] http://www.kickstarter.com

Most importantly, these new BCIs strengthen the possibility for neurofeedback among consumers in their daily settings. In the next section we focus on the neurofeedback concept.

2.3 Neurofeedback

When given real time feedback on its oscillations, the brain can learn to control and change them. This is interesting since the brains oscillations are significantly correlated with brain functions and behavior as well as with psychiatric diseases [73]. Neurofeedback training exploits this mechanism by providing feedback based on oscillation frequencies correlated with some desirable function or behavior.

The neurofeedback mechanism was discovered and developed in the 1960s, but the first controlled studies providing clinical evidence supporting neurofeedback training effects were published in the 1980s [5]. Since then, the efficacy of neurofeedback therapy has been documented in several studies [40] [6] [69] and neurofeedback [19] is listed among the treatments with highest evidence support for certain conditions according to The American Academy of Pediatrics (AAP) [1]. Neurofeedback is routinely used in treatment of a number of conditions including Attention Deficit Hyperactivity Disorder (ADHD), anxiety, epilepsy, and addictive disorders [5].

Besides its clinical usage, a number of studies show that neurofeedback training can increase cognitive performance. For example, it has been shown to increase semantic working memory, focused attention, perceptual sensitivity and reaction time [5]. Neurofeedback training has also been shown effective by real-life behavioral measures - e.g. by increasing musical performance in a stressful context among conservatory students in a study designed to ensure ecological validity [22].

2.3.1 Neurofeedback in practice

In a typical clinical context, neurofeedback sessions of 45-60 minutes duration are performed twice per week. The number of treatments depend on the individual response to the training and the condition being treated - e.g. 40-80 sessions are suggested in ADHD treatment [5].

Neurofeedback therapy is relatively expensive. For example, 40 training sessions cost > DKR 30.000,- at Ann-Helen Pettersen's clinic [20].

[19] a subset of biofeedback which is the term mentioned in the The American Academy of Pediatrics (AAC) recommendations.

[20] http://www.hjernetraening.dk/prisliste.html

2.3.2 Stress and alpha feedback training

Alpha feedback training is the subset of neurofeedback training for which the goal state of the feedback is defined in terms of the amount of alpha waves - thereby seeking to increase the alpha activity.

Alpha activity is associated with a relaxed consciousness [5]. Together with theta, alpha is the EEG frequency band in which effects of meditation are most significant [4] [15]. Alpha 'blocking' (i.e., reduction) is associated with alertness [5]. Thus, by increasing alpha levels, alpha feedback training has been shown - amongst other positive effects such as increased cognitive performance - to reduce stress and anxiety [73] [5] [69].

With a classification approach, EEG has been used to classify subjects from either a chronically stressed or a control group with a success rate higher than 90% [33]. This testifies to the manifestation of stress in EEG data.

The dominant frequency within the alpha band - the *alpha peak* - and the amplitude of the alpha band varies between individuals [9] [5] [73]. An alpha feedback training system can account for this by calibrating according to the individual alpha peak and the baseline amount of alpha. This is, for example, the approach taken in the alpha feedback system presented in [59]. The importance of giving feedback on individually determined frequency bands is investigated in [9] and concludes that "*Neurofeedback training applied in individual EEG frequency ranges was much more efficient than neurofeedback training of standard EEG frequency ranges*".

2.4 Related works

Having explained current state of consumer EEG BCIs, the neurofeedback mechanism and how alpha feedback training can help to reduce stress, we now present existing systems and research within the consumer neurofeedback domain. There is only a limited number of such systems and research for reasons already mentioned above:

1. Neurofeedback therapy is expanding but not widely adopted yet.

2. Consumer BCIs have only emerged within recent years. They are still maturing and not widely adopted yet.

Figure 2.5: Brainball BCI system in the new version named Mindball. A monitor displays the brain activity of the participants (Image courtesy of Interactive Productline IP AB).

This section present and discuss the commercially available systems Brainball and BioZen and the research project SmartphoneBrainScanner2.

2.4.1 Brainball

According to the researchers behind, Brainball "*... dwells in the realm between art and research, entertainment and science, method and object*" [29]. They present a game with a tangible user interface in which two opponents sit on a chair separated by a table. A steel ball lies between them. The players wear specialized BCIs mounted on their foreheads and somewhat similar to the NIA system (see Section 2.2). The EEG signal is analyzed into its frequency components and the ball will roll away from the most relaxed player (drawing on the correlation between alpha activity and relaxation). While playing, the players are able to see a screen visualizing their EEG activity. The creators of Brainball interestingly reports that playing the game leads to increased relaxation by measure of both Galvanic Skin Response (GSR) and self reports [29]. Brainball experienced a lot of attention including honorary mention at Ars Electronica 2000 and 100+ appearances on TV. However, it remains a niche product within gaming and entertainment due to the dependency on specialized hardware (BCI, ball, screen, etc). The Brainball BCI system is commercially available through a Swedish company under the name *Mindball* [21] (Figure 2.5).

[21] http://www.mindball.se

2.4.2 BioZen

BioZen [22] is a consumer biofeedback system developed by the National Center for Telehealth and Technology (T2) under the US Department of Defense. The fact alone that this organization is behind a biofeedback system witnesses to the increasing adoptation of the biofeedback (here under neurofeedback) method. The system consists of an Android application [23] in conjunction with one or more consumer bio-sensors. Several sensors are supported including heart rate, skin temperature, GSR and EEG sensors. For EEG measurements, the Neurosky MindWave (and some older Neurosky BCIs) are supported. The BioZen app uses processed data delivered from the Neurosky SDK (Delta, Theta, Low Alpha, High Alpha, Low Beta, High Beta, Low Gamma, Mid Gamma, (e)Attention, (e)Meditation) and these values can form the basis for neurofeedback training. Relying on the Neurosky SDK for frequency analysis, BioZen is bound to the limited set of frequency spectra mentioned above.

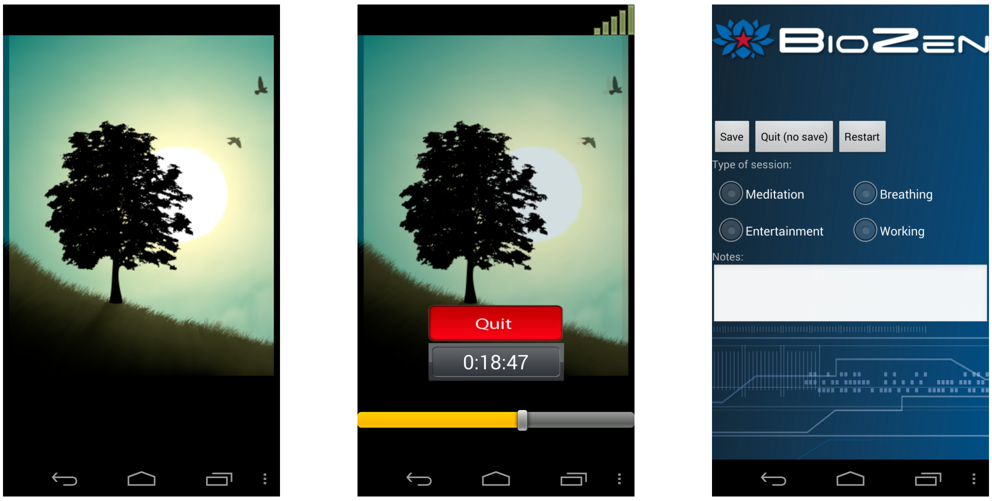
On the BioZen web page, T2 claims that "*BioZen is the first portable, low-cost method for clinicians and patients to use biofeedback in and out of the clinic*" and that "*[BioZen] takes many of the large medical sensors in a clinic and puts them in the hands of anyone with a smart phone*" [23]. In other words, it is promoted for clinical usage - a claim the authors of this thesis are very cautious about making on behalf of AlphaTrainer (see Section 1.5).

The feedback consists of an image of a hill in which the background brightness and the visibility of a foreground tree are the feedback variables. The background is brighter when the chosen parameter (e.g. some EEG power band) is higher while the foreground tree is more visible when the chosen parameter is more stable [60] (see Figure 2.6).

2.4.3 Smartphone Brain Scanner 2

The Smartphone Brain Scanner is developed at the Technical University of Denmark (DTU). The project aims at moving EEG research out of the laboratory by means of low cost wireless BCIs and smart phone based real-time neuroimaging software which "*may transform neuroscience experimental paradigms*" [59]. The important notion here is that while leaving the laboratory and using consumer interfaces, the focus is still on research.

[22] http://t2health.org/apps/biozen
[23] Android App: https://play.google.com/store/apps/details?id=com.t2



(a) BioZen feedback consisting of a sun varying from dark (low alpha) to light (high alpha). A new age soundtrack loops in the background (can be disabled in the configuration).

(b) The input gain of the feedback parameter (i.e. the alpha band) can be adjusted through a slider thus providing a manual calibration.

(c) Save feedback session with a tag (meditation, breathing, entertainment or working) and a note.

Figure 2.6: Screenshots of the neurofeedback from the BioZen Android app using the MindWave BCI (Images courtesy of BioZen).

They are in principle headset agnostic and support the *Easy Cap* and EPOC (described in Section 2.2.1) BCIs. The EPOC is both used in its standard configuration and in a modified configuration in which it is merged with hi-fi gel based electrodes. On the software side, they use their Smartphone Brain Scanner (SBS2) open source software framework including state of the art EEG signal processing such as source reconstruction, noise filtering and frequency analysis [24]. It is build on the QT C++ framework [14] which allows compilation to the major desktop and mobile operating systems. However, it is not trivial to embed a QT module inside another native applications e.g. on the Android platform. Wireless connection to the EPOC BCI goes via an USB dongle and requires an Android phone to be rooted to function, the platform does not provide an easy way of interfacing with BCIs via bluetooth. Furthermore it requires the research edition of the EPOC.

To validate the design of the Smartphone Brain Scanner, the research team behind has build 3 brain imaging applications including an alpha training application. Again, the focus is on research - specifically, the interface parameters of neurofeedback training are investigated. The efficacy of two different feedbacks were compared in a controlled study by measuring increase in alpha amplitudes with each interface during a week of intensive training. One feedback show a square changing

[24] : https://github.com/SmartphoneBrainScanner

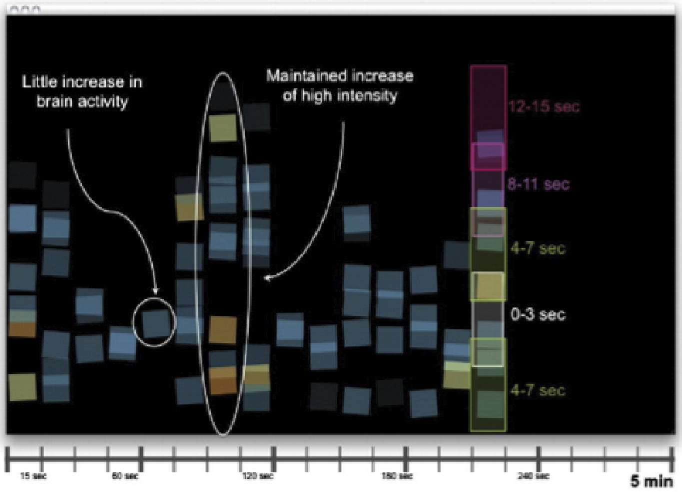


Figure 2.7: Smartphone Brain Scanner neurofeedback interface. High alpha amplitudes manifests itself in the creation of small boxes and their color. By keeping the created boxes visible during the 5 minute training period, the interface reveals performance history [59]. Image courtesy of the Smartphone Brain Scanner project.

colors between blue over gray to red for which red represents high alpha. In another interface, high alpha amplitudes manifests itself in the creation of small boxes and the color of the boxes created. By keeping the created boxes visible during the 5 minute training period, the interface reveals performance history thus "*allowing the user to easily compare methods for increasing the amplitudes*" [59] (Figure 2.7). The training effect measured by comparing baselines revealed only a statistic significant increase in alpha for the box changing colors while the alpha levels during training were significantly higher using the square creation interface.

The conclusions especially relevant to this thesis are that:

1. Alpha feedback training is feasible in a mobile setup.
2. Alpha levels during training (effect of feedback) is not necessarily correlated with a general increase in alpha levels (effect of training).

2.5 Sum up of background and related systems

Table 2.3 lists neurofeedback systems which are either commercially available or use a consumer BCI. These systems have been chosen because they outline the current state of systems in the domain of AlphaTrainer. The parameters highlighted in the table express parameters desirable or necessary for a consumer neurofeedback system. No existing system includes all parameters. For example, no

Parameters	Brainwave Visualizer & Transcend	Brianball	BioZen	Alpha feedback app - Smartphone Brain Scanner
Convenient (dry sensor + bluetooth connectivity).	yes	no	yes	no
Headset agnostic.	no	no	yes	(yes)
Individual feedback. spectra	no	n/a	no	yes
Efficacy documented.	no	no	no	yes
Available to customers.	yes	yes	yes	no
Low cost.	yes	no	yes	n/a

Table 2.3: Related systems

consumer available system includes the ability to give feedback on individually adapted frequency bands which is important for the effectiveness of the feedback training (see Section 2.3.2). This "gap" among current neurofeedback systems is our motivation for designing and building AlphaTrainer as discussed in the following chapters.

Chapter 3

Evaluation of 3 consumer BCIs

This chapter addresses **G1** (Section 1.2) by evaluating 3 of the consumer BCIs presented in the Background Chapter 2: EPOC (Section 2.2.1), MindWave (Section 2.2.2) and TrueSense Kit (Section 2.2.3). The BCIs are evaluated based on their feasibility for alpha feedback training and finally in the conclusion of this chapter we will select a BCI to use in the further development of AlphaTrainer. This conclusion will be based on several aspects such as comfort and connectivity. However, the main effort of this chapter is put into answering the least accessible - and arguably the most important - factor: "*are the BCIs able to measure alpha waves*"?

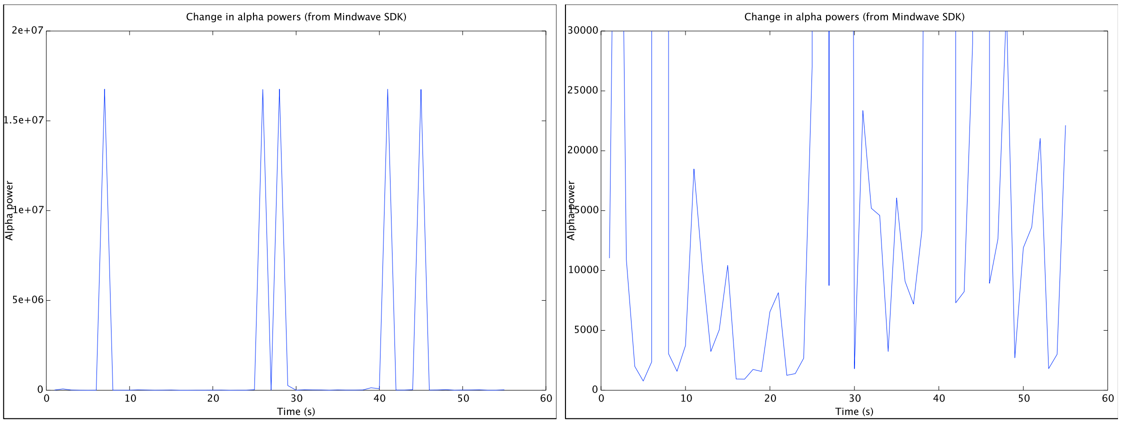
In designing an experiment to answer this, we did a pre-study which is explained in the next section. After that we move on to explain the setup, execution and data analysis of the actual experiment.

Method

To answer whether a BCI can measure alpha, we exploit the property of alpha activity, that it is generally higher when eyes are closed compared to when eyes are open [9]. By comparing alpha levels recorded under a closed-eyes condition to alpha levels recorded under an open-eyes condition, we get a measure of how able a BCI is to detect alpha activity.

3.1 Pre-study

The first BCI we got our hands on was the MindWave. Eager to find out whether it could measure alpha levels, we implemented a simple proof-of-concept prototype (explained in more detail in Chap-



(a) No y-axis zoom.

(b) With y-axis zoom (0 - 30,000).

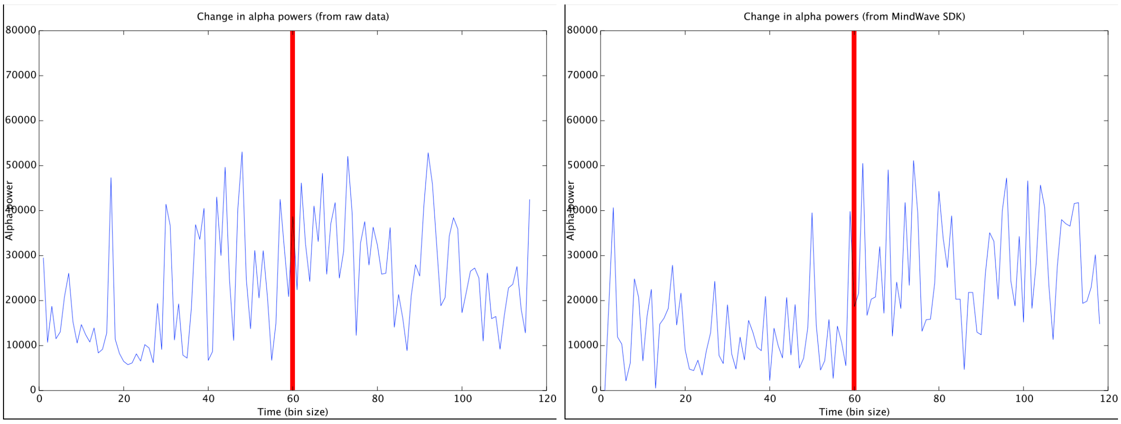
Figure 3.1: Two plots of the same MindWave SDK alpha powers with/without zoom.

ter 4) able to store SDK alpha values (high and low alpha bands) as well as the raw EEG signal. To get an initial idea whether the headset could measure alpha, we used this prototype to record and compare SDK alpha values under closed and open-eyes condition. Concretely, we performed some recordings in which a test subject would have open eyes the first half and closed eyes the last half of the recording. Figure 3.1 shows two plots of SDK values recorded in this way - the plot on the right has a y-axis limit while the plot on the left shows the actual values.

The plot is representative for many similar recordings we performed and show some giant outliers in the SDK values. Similar outliers were present in the other power bands as well. To find out whether the outliers originated from the raw data or the SDK data processing, we applied offline frequency analysis to the raw data in Octave [1] - an open-source equivalent to MATLAB. We were not able to reproduce the outliers which means they were not present in the raw data. We initiated a correspondence with NeuroSky technicians (see Appendix *MindWave forum posts*) but they were unable to pinpoint the origin of the outliers in spite of receiving both SDK values and the originating raw data. They were also unable to disclose how the SDK values are calculated due to the closed source nature of their SDK.

In the process of investigating the origin of the outliers, we also suspected EMG and EOG noise. This led us to compare recordings in which we induced such noise to recordings where we minimized it as much as possible by relaxing facial muscles and avoiding eye movement. While this did not prevent the outliers, it showed a significant impact of the noise in the form of a general increase in

[1] http://www.gnu.org/software/octave



(a) Alpha powers (from raw data).

(b) Alpha powers (from MindWave SDK, outliers removed).

Figure 3.2: Alpha levels recorded with open eyes (first half) and closed eyes (second half).

band power across all frequency bands. This led us to the insight that we would need to minimize facial muscles and eye movement in the actual experiment as much as possible.

From this point, we cleaned the MindWave SDK data by replacing the outliers with the mean value of the non-outliers. We carried on the recordings to compare the alpha levels when eyes are open to when they are closed. A representative plot of the resulting data can be seen in Figure 3.2. In this case we recorded for 4 minutes and summed the alpha levels in bins of 2 seconds. As exemplified in the plot, we measured a clear tendency of an increase in alpha when the test subject closed the eyes at the midpoint of the recording.

To avoid dependencies on black box SDK calculations and to get comparable results across different BCIs, we chose to rely only on our own signal processing from this point. Both the TrueSense Kit and the EPOC (research edition) BCIs support raw EEG out of the box through their bundled software OPI Console and TestBench (Appendix *TrueSense Kit console* and Appendix *Emotiv EPOC TestBench*).

Next step was to select which of the 16 electrode(s) to use from the EPOC headset. We chose to use one of the electrodes placed at location O1 or O2 (back of the head) - the electrodes used in the Smartphone Brain Scanner alpha feedback training application [59]. This placement has the advantage of little EOG and EMG noise (which we tested by comparing recordings from different electrodes while introducing eye movement and facial muscle activation). We did not experience significant difference between the two channels and we decided to use only the O1 electrode in order to make sure results could be directly compared to the results of the other single channel BCIs.

Furthermore we had to decide the placement of the TrueSense Kit BCI. It features flexible mounting by its small size and elastic band mount, so placement is not obvious. To obtain the advantages of recording from the back of the head, we tried mounting the sensor on top of the neck hair at O1. This gave us very noisy data both in the dry and the gel-pad electrode configuration probably due to low scalp conductivity. The same was the case when trying to place the sensor on the temple. We ended up placing the sensor at the side of the forehead as recommended by TrueSense Kit for measuring alpha [2]. Finally, we decided to record the TrueSense Kit over radio (ZigBee) rather than to the memory module attachable to the sensor. The latter would presumably give better recording quality. However, in a neurofeedback context offline recording is not relevant since the data must be fed into the feedback mechanism instantly.

This concludes the pre-study which incrementally has led to all the building blocks necessary for designing an experiment which will answer whether the BCIs of interest can measure alpha.

3.2 Experiment design

This section formalizes the design of our experiment which will answer to which degree the BCIs can measure alpha from guidelines by MacKenzie [42]. Revisiting our method, it is assumed that alpha levels are higher when eyes are closed than when eyes are open. From this assumption, we reach the hypothesis tested in the experiment:

H1

> The EEG recorded while eyes are closed contains higher alpha levels than EEG recorded while eyes are open.

The independent variables (factors) of the experiment are the *eye states*: (*i*) open; and (*ii*) closed. The dependent variable is simply the alpha level. The task of the participants is to relax as much as possible (keep blink, face muscles and eye-movement at a minimum). We did a pilot experiment on our selves and on a domain expert within experimental psychology to catch potential problems which led to a slight refining of the execution procedure. We later carried out the experiment on 6 participants (4 males and 2 females).

To ensure that we would not only measure a precedence effect or some other learning effect - i.e. gaining familiarity with wearing a brain interface and thus being increasingly comfortable and

[2] http://op-innovations.com/en/wearsensor

Participant	Headset 1	Headset 2	Headset 3
1	EPOC (o/c)	MindWave (c/o)	TrueSense Kit (o/c)
2	MindWave (c/o)	TrueSense Kit (o/c)	EPOC (c/o)
3	TrueSense Kit (o/c)	EPOC (c/o)	MindWave (o/c)
4	TrueSense Kit (c/o)	MindWave (o/c)	EPOC (c/o)
5	EPOC (o/c)	TrueSense Kit (c/o)	MindWave (o/c)
6	MindWave (c/o)	EPOC (o/c)	TrueSense Kit (c/o)

Table 3.1: Experiment order for headsets and participants - o = open and c = closed

(a) Using EPOC.

(b) Using TrueSense Kit.

(c) Using MindWave.

Figure 3.3: Experiment setup.

relaxed (which minimizes EMG and EOG noise) - we varied the BCI recording order as lined out in Table 3.2.1. We now describe in more detail the execution of the experiment.

3.2.1 Execution

The setting was a dark room with no sun light, a lap top computer with a still image to focus on (Hokusai, “The great wave”), a table and an office chair (Figure 3.3).

Before entering the recording room, the participant was instructed in a breathing exercise to help her relax. The exercise consisted of breathing slowly through the nose while continuously counting

	EPOC	MindWave	TrueSense Kit
Level of relaxation	7.17 ±1.60	8.17 ±1.17	8.33 ±0.82
BCI comfort	5.67 ±2.73	5.00 ±1.67	8.50 ±2.06

Table 3.2: Experiment micro questionary

1 down when exhaling. The count down starts from 4 and whenever 0 is reached it starts over from 4.

The procedure for each participant - BCI combination was this:

- The participant entered the recording room, sat down and got a headset mounted.
- The participant was instructed to sit still, keep eye movement and blinking to a minimum, relax face muscles and to do the simple breathing exercise instructed earlier.
- The participant was instructed to focus at a specific point in the image when having open eyes (to hold visual stimuli constant across all participants and headsets).
- 3 times in a row, we recorded for 5 minutes. After 2.5 minutes the participant is instructed verbally to close or open her eyes.

This procedure was repeated until all 3 BCIs had been recorded. After each session (3 recordings with the same headset) we asked the participant: (*i*) "*how relaxed did you feel during the recordings from 1 to 10, where 10 is very relaxed?*"; and (*ii*) "*how comfortable was it to wear the headset from 1 to 10, where 10 is very comfortable?*". The results of this questionnaire are listed in Table 3.2.2. They indicate that TrueSense Kit is the most comfortable BCI to wear and that the EPOC and MindWave BCIs are both "medium" comfortable. The results also indicates that the subjects were quite evenly relaxed across the different BCIs.

3.2.2 Data processing, cleaning and analysis

From the 5 minutes recorded, we extract 2 minutes of open eyes and 2 minutes of closed eyes - thus discarding a 15 second margin on either side of each extraction. These two extractions are subject for the frequency comparison.

For frequency estimation, we use the *P.D. Welch algorithm* which cuts up the data in small time windows and for each frequency in each window the density of that frequency is calculated/estimated using FFT [68]. These frequency densities are then squared and summed up within a frequency band resulting in a total estimation of the power of that frequency band in the input signal (Appendix *Experiment - signal processing*).

We compare alpha levels between the open/closed eyes conditions in two ways: (*i*) Direct comparison by dividing alpha band power under closed-eyes condition with alpha band power under open-eyes condition. This is how the alpha band power is calculated in the Smartphone Brain Scanner alpha feedback training experiment [59]. We call this the "*absolute alpha*" measure. (*ii*) We also compare the relative alpha levels of each condition. The relative alpha level is found by dividing the alpha band power with a broad "total" band power (in our case 5 Hz - 25 Hz) to get a measure of how much the total band power is made up of alpha. Dividing the relative alpha level under closed-eyes condition with the relative alpha level under open-eyes condition gives us the "*relative alpha*" measure. This approach is used in [24] to estimate theta band power.

$$\alpha_{\text{absolute}} = \frac{\sum \alpha_{\text{closed eyes}}}{\sum \alpha_{\text{open eyes}}} \tag{3.1}$$

$$\alpha_{\text{relative}} = \frac{\dfrac{\sum \alpha_{\text{closed eyes}}}{\sum \text{ EEG power closed eyes}}}{\dfrac{\sum \alpha_{\text{open eyes}}}{\sum \text{EEG power open eyes}}} \tag{3.2}$$

The absolute alpha measure compares the alpha levels under each condition (open/closed eyes) isolated from the total EEG power level (Equation 3.1). The relative alpha measure will account for the total EEG power by comparing alpha relative to the total EEG power under each condition (Equation 3.2). We expect the latter to be the most reliable measure of alpha in that it will be more resistant to EMG/EOG noise (under the assumption that such noise will raise the overall EEG power level somewhat evenly which matches our observations from recordings under induced EMG and EOG).

We did some simple filtering by discarding bins where the alpha or total band power deviates from mean by more than 4 * std.dev. In these cases, the power would mainly origin from EMG/EOG. For each headset, this cleaning excluded 2 of the 18 recordings. The mean values $\pm$ standard deviation are listed in Table 3.3. The mean value of the absolute and relative alpha measures inform

Measure	EPOC	MindWave	TrueSense Kit
Absolute alpha	4.50 ±3.12	2.37 ±1.54	0.87 ±0.40
Relative alpha	2.12 ±0.63	1.55 ±0.60	1.06 ±0.21

Table 3.3: Mean values of the absolute and relative alpha measures

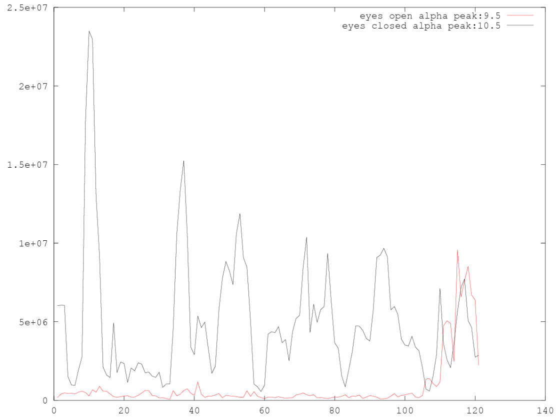
about the degree to which a higher alpha level has been detected under closed-eyes condition. The standard deviation informs about the consistency of the recording measure and BCI combination (lower std.dev. means more consistent). The resulting data suggests:

- The EPOC and MindWave BCIs can detect alpha (mean is > 1).
- The EPOC headset detect the most alpha.
- The relative alpha measure is the most reliable (i.e. lower std.dev.) as expected.
- The TrueSense Kit sensor only shows a vague difference in alpha between the two conditions which is unlikely to be sufficient for alpha feedback training. Furthermore Figure 3.4f exemplifies the general tendency that the recordings of TrueSense Kit are noisy and lack the expected alpha peak under closed-eyes condition. The narrow peak around 12,5 Hz is likely to be some power grid noise being picked up (which we experienced a lot with the TrueSense Kit). Figure 3.5f also show noisy data from the TrueSense Kit. In this case we see a lot of low frequency noise and its harmonics - the series of peaks appearing approximately every 3 Hz. This noise pattern is roughly the same under open and closed eyes conditions which indicates that the main source of the recording is not EEG (i.e. the alpha peak is missing).

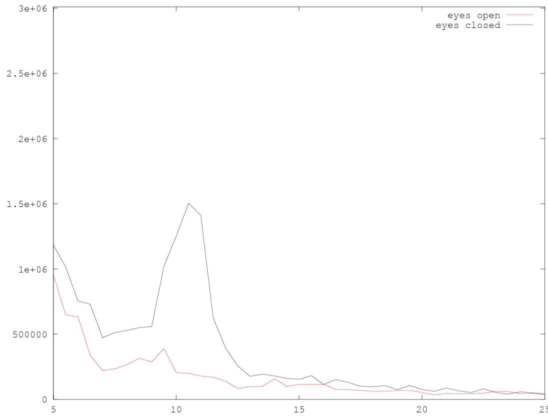
Figure 3.4 and Figure 3.5 show recordings of each BCI for two different participants. They exemplify how the alpha peak varies between individuals. For example, Participant 3 has an alpha peak under closed-eyes condition around 10.5 Hz while participant 5 has one around 11.5 Hz.

3.3 Choice of BCI

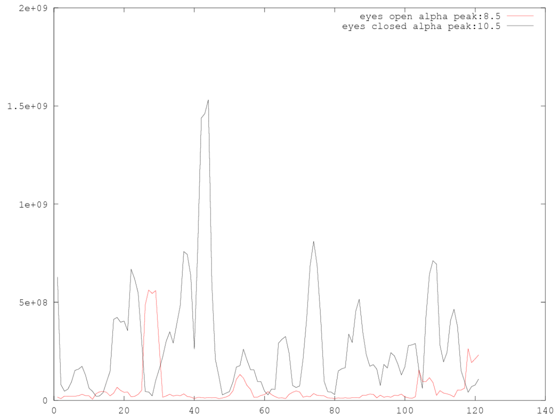
We have lined up the parameters of the evaluation in Table 3.4. We choose the MindWave BCI in the further development of AlphaTrainer because: (*i*) it can detect alpha; (*ii*) it does not require assistance or preparation time to use; (*iii*) it supports Bluetooth communication; finally (*ii-ii*) it has out of the box SDK support for mobile devices including access to raw EEG data.



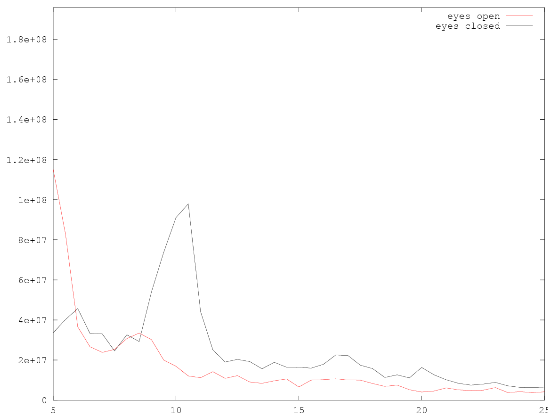
(a) EPOC alpha progress. Alpha peaks: 9,5 Hz (open eyes) and 10,5 Hz (closed eyes).



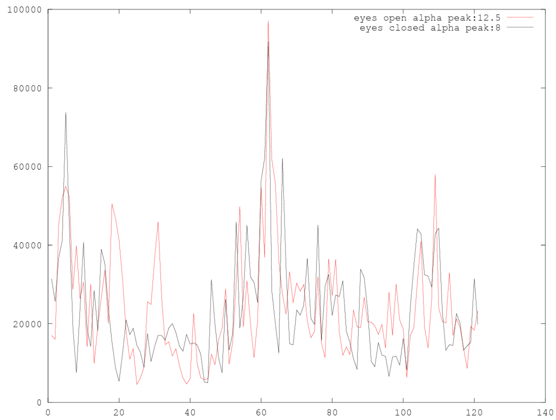
(b) EPOC power of each frequency.



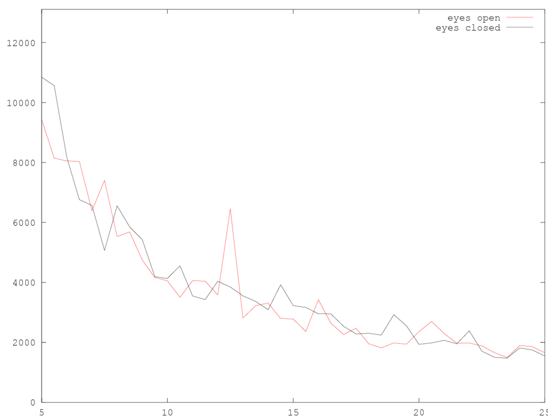
(c) Mindwave alpha progress. Alpha peaks: 8,5 Hz (open eyes) and 10,5 Hz (closed eyes).



(d) Mindwave power of each frequency.

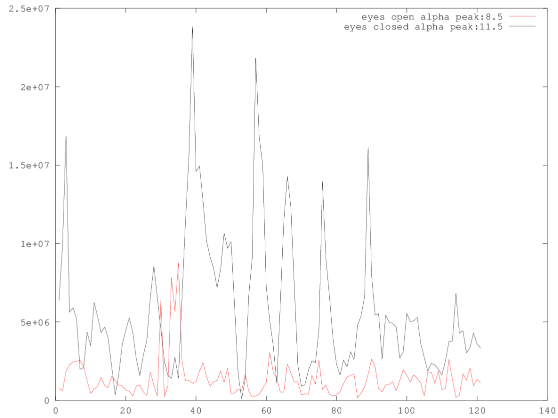


(e) True Sense Kit alpha progress. Alpha peaks: 12,5 Hz (open eyes) and 8 Hz (closed eyes).

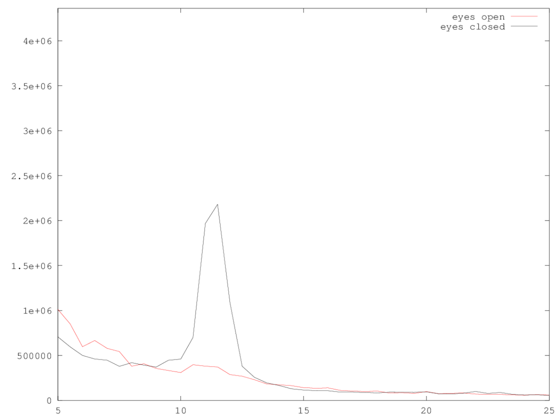


(f) True Sense Kit power of each frequency. Pay attention to the narrow (noise) peak around 12,5 with open eyes.

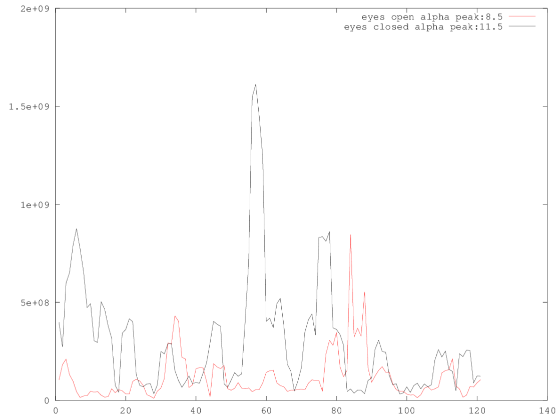
Figure 3.4: Plots from recordings of participant 3 from the experiment. On the left, alpha progress with eyes open (red line) and eyes closed (black line) are plotted. On the right, the power of each frequency is plotted - notice the alpha peak around 10 Hz in the EPOC and MindWave recordings under closed-eyes condition.



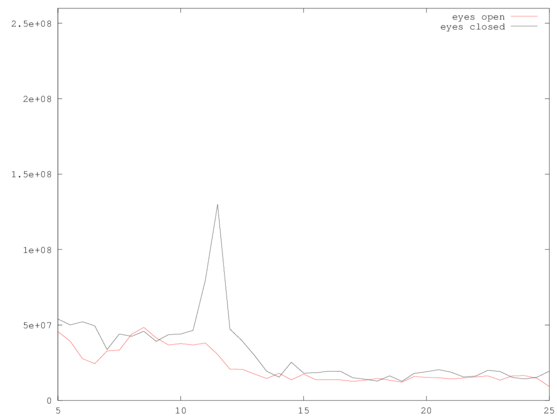
(a) EPOC alpha progress. Alpha peaks: 8,5 Hz (open eyes) and 11,5 Hz (closed eyes).



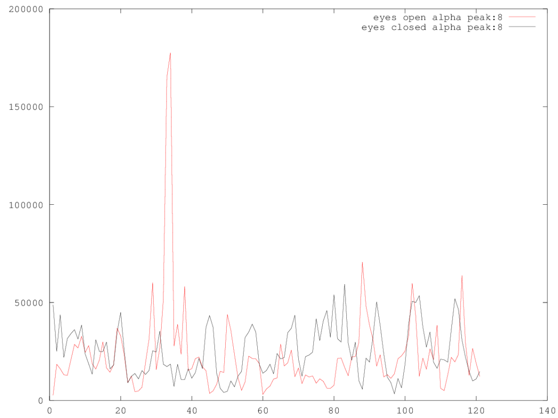
(b) EPOC power of each frequency.



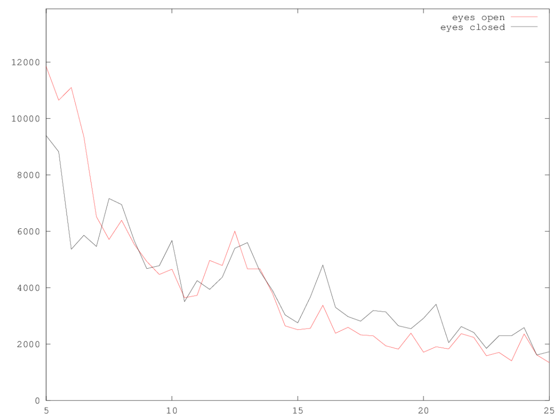
(c) Mindwave alpha progress. Alpha peaks: 8,5 Hz (open eyes) and 11,5 Hz (closed eyes).



(d) Mindwave power of each frequency.



(e) True Sense Kit alpha progress. Alpha peaks: 8 Hz (open eyes) and 8 Hz (closed eyes).



(f) True Sense Kit power of each frequency.

Figure 3.5: Plots from recordings of participant 5. On the left, alpha progress with eyes open (red line) and eyes closed (black line) are plotted. On the right, the power of each frequency is plotted - notice the alpha peak around 11.5 Hz in the EPOC and MindWave recordings under closed-eyes condition.

	EPOC	MindWave	TrueSense Kit
Outputs raw EEG	(yes)	yes	yes
Channels	14	1	1 and up
Ability to measure alpha (1-10)	10	7	0
Preparation time (minutes)	$>$ 10 min	$<$ 1 min	1-5 min
Dry sensors	No	Yes	Yes
Comfort to wear (1 - 10)	5.67 $\pm$2.73	5.00 $\pm$1.67	8.50 $\pm$2.06
Bluetooth support	no	yes	no
Mobile SDKs	(yes)	yes	(yes)

Table 3.4: BCI evaluation

The rationale behind these arguments is based on our overall method which includes building a system deployable for actual use. In a research context (or any context requiring spatial resolution), the EPOC is an excellent choice. And in a quantified-self context in which offline recording and processing is possible, the TrueSense Kit is the obvious choice due to its size and comfort.

Another advantage of the MindWave is that it seems to best represent the future of consumer BCIs. As mentioned in Section 2.2.4, the next generation of consumer BCIs feature Bluetooth connectivity and a dry sensor configuration. By using MindWave in the further development of AlphaTrainer, we are best prepared for adapting to future BCIs.

Chapter 4

Design

So far, this thesis has established some important notions that we use as input to the design of AlphaTrainer: (*i*) from the Background Chapter 2 we learned that alpha feedback training can benefit to reduce stress; (*ii*) from the BCI Evaluation Chapter 3 we learned that the MindWave Mobile BCI is feasible for building an alpha feedback training system; and (*iii*) in the Introduction Chapter 1 we presented our overall method which includes the need for a robust system deployable for real world use.

These three notions span the design space for building AlphaTrainer. This chapter first describes the design model and method used and then moves on to explain some of the important design activities and choices made during the design process and finally present the design of AlphaTrainer.

4.1 Design model and method

This section presents the model used throughout the design of AlphaTrainer and the set of methods used during the design process. We start out by placing the design problem at hand within a Human-Computer Interaction (HCI) context which motivates the choice of design model.

The ISO 9241-210:2010 standard for "Human-centred design for interactive systems" states the following about HCI design:

"*The complexity of human-computer interaction means that it is impossible to specify completely and accurately every detail of every aspect of the interaction at the beginning of development. Many of*

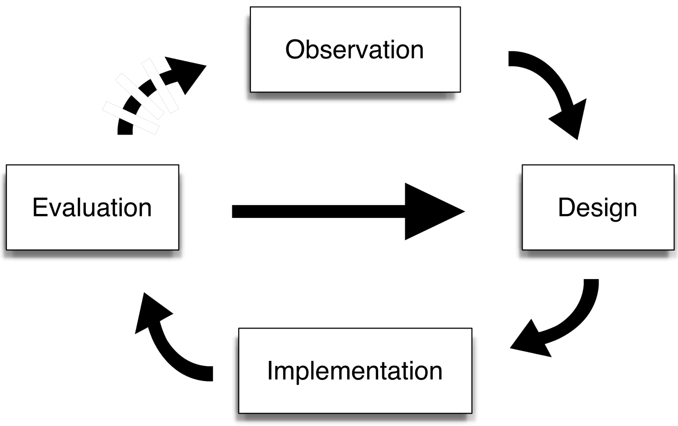


Figure 4.1: The iterative model.

the needs and expectations of users and other stakeholders only emerge [...] as the designers refine their understanding of users and their tasks, and as users are better able to express their needs in response to potential solutions“ [31] Section 4.5 p. 6.

Since we are designing a system which aims to enable a currently non-existing practice - namely to perform alpha feedback training on a mobile device in an everyday context - we are certainly struck by the complexity in regard to specifying user needs up-front.

When designing for ubicomp, additional aspects has to be taken in consideration which further increase complexity: (*i*) different devices; (*ii*) mobile users; and (*iii*) changing environment and context [8]. We recognize that we are not only designing interfaces but also designing interactions between people and the system through some artifacts embedded in an environment [10]. In a neurofeedback training system we have: (*i*) a user; (*ii*) using artifacts in form of the mobile device and the headset; and (*iii*) in an environment with a lot of parameters such as noise, changing lights, other people etc. This requires us to think of the interaction in context - *in-situ* - at home, at work or somewhere in between (e.g. when commuting) [12] [3].

To deal with the complexity and difficulty of specifying user needs and system requirements up front, we take a user centered approach to our design process. This approach enables specifications to emerge during the design process through experiments and prototypes from which we will learn and design new experiments and prototypes [12] [34]. We have drawn the model in Figure 4.1.

To envision the needs and goals of the users of our system we have been using personas and scenarios [35][13]. A persona simply model a certain user in order to delimit the target group for whom we are designing AlphaTrainer. We have created four personas named Morten, Niels, Olivia and Peter

Figure 4.2: Peter performing alpha training at work. Frame 1: Peter has a busy day with a lot of deadlines waiting around the corner. Frame 2: Before lunch, he finds a quiet spot. Frame 3: He performs 15 minutes of alpha training. Frame 4: He finds himself relaxed and gets back to work.

(Appendix *Design - personas*). We are using scenarios to frame the context and situation when and where the system is used. As a subset of scenarios we have worked with some simple storyboards to capture the setting, sequence and satisfaction of the actual alpha training (Appendix *Design - storyboards*).

One of the storyboards cover a work scenario and the person could be *Peter* (Figure 4.2). *Peter* has had a busy day with a lot of meetings and deadlines waiting around the corner. It is 11.30 and he realizes that he can squeeze in alpha feedback training just before lunch to get his mind clear. Peter finds a silent spot in the office space which mostly consists of big open spaces but luckily there has been arranged some quiet spots around. He chooses one of the feedbacks with sound because it is convenient to use when training with earphones in an office environment. First he does the calibration, then the reference recording as the app asks for and finally he performs three 5 minutes training sessions in a row. He improves his alpha during training and finds himself in a relaxed mental state after the training. Back to work.

4.2 Design process

Since we have deployed an iterative approach to the design process, we have decided to present important design activities and decisions chronologically. This clarifies how we continuously have fed the output of one design activity as input to another.

4.2.1 Initial experimental prototype

We started out by making a very simple prototype in the form of an Android app as proof of concept on getting data from the NeuroSky MindWave Mobile BCI and to tryout of their SDK signal processing. In a bottom up approach to investigate what we could control by means of alpha levels, we tried assigning different audio parameters (volume, pitch, placement in 3D perspective) to the SDK power band values for low and high alpha. The power band values are listed in Section 2.2.2.

We tried out the prototype informally on our selves and on fellow students and learned that the SDK alpha values contained giant outliers as already mentioned in the BCI Evaluation Chapter 3. Besides, we were unsure whether to use the low alpha or high alpha frequency band from the SDK since the literature states - as explained in Section 2.3.2 - that the alpha band varies between individuals and that neurofeedback is significantly more effective when the feedback is given on individually adapted frequency bands.

In researching how neurofeedback is practiced, we contacted Ann-Helen Pettersen - one of the few Danish psychiatrists specializing in neurofeedback therapy [1]. She also stressed the importance of accounting for individually determined frequency bands. When practicing neurofeedback therapy, she starts out by recording a map of EEG frequency intensities. This "brain map", as she calls it, serves as input to the choice of an appropriate feedback frequency band.

Based on the notions from the literature on alpha feedback training backed up by a clinician's neurofeedback practice, we decided that our alpha training system would need the ability to give feedback on an alpha band adapted to the individual alpha peak frequency. This led us to experiment with doing the frequency analysis our selves. This also enabled us to test whether we could reproduce the outliers experienced from the SDK alpha values which would inform about whether they originated from the raw data or from the SDK processing.

We did the signal processing offline (see Section 3.1) and interestingly we were not able to reproduce the outliers we got from the SDK. Another interesting find during the data analysis was that the alpha wave intensity comes in chunks of a few seconds length as can be seen, for example, in Figure 3.4c. The chunks of high alpha activity can even be observed directly from a raw EEG signal such as the one shown in Figure 2.1.

4.2.2 Early functioning Android prototype

As the next step we implemented a working Android alpha feedback training prototype. We focused on the needs for custom signal processing revealed in our previous prototype experiments and from

[1] http://www.hjernetraening.dk

the literature. Our approach to signal processing is covered in the Implementation Chapter 5. We also created a set of 5 different feedback views inspired by other neurofeedback systems.

The views were variations over a bar changing height and a box changing color. We read about the bar feedback in [37] describing its clinical usage in ADHD treatment. The goal for the trainee is to raise the bar which in our case represents the magnitude of alpha waves. The basic bar feedback is shown in Figure 4.3e. The feedback consisting of a box changing colors was used in the Smartphone Brain Scanner alpha feedback training application as mentioned in the Background Chapter 2. In our case, the box gradually change from red over yellow to green. The trainees goal is to make and keep the box green which represents high alpha magnitude. The basic bar feedback is shown in Figure 4.3c. Inspired by the Smartphone Brain Scanner alpha feedback training app (Section 2.4.3) - which included another feedback in which performance history was visible - we implemented a version of each interface showing recent performance history of a sliding time frame of 30 seconds.

In the case of the box feedback, history was visible in the frame color - see Figure 4.3d. In the case of the bar, we implemented performance history in the form of a horizontal line showing performance history of a 30 seconds sliding time frame. Additionally, we visualized performance history of the entire training session in the form of background color changes - see Figure 4.3f. From our initial experience with the feedback bar, we thought the rather violent movements from low to high due to the intrinsic variance in alpha magnitude might introduce EOG noise. Therefor we made another variation of the bar interface in which the bar only grows. High alpha magnitudes makes it grow fast while low alpha magnitudes slows the growing down or stops it completely (Figure 4.3g).

In sum, the prototype featured: (*i*) alpha feedback training through 5 different feedback views; (*ii*) individual alpha peak detection; (*iii*) alpha feedback training based on flexible alpha band definition; and (*ii-ii*) a view showing performance history in a plot (Figure 4.3h).

Using the app relied on the following sequence of actions:

1. The user should turn on and mount the MindWave BCI and launch the Android application prototype.

2. The user should calibrate according to the individual alpha peak which is done by pushing the "Calibrate" button. The alpha peak frequency is used to delimit the alpha band for which to give feedback on.

3. The user should record a baseline while under influence of random feedback which is done by pushing the "Baseline" button. The baseline is used in mapping the current alpha level to a feedback state. Furthermore, the baseline recordings are used to track alpha activity gains over time, the *training effect* of using AlphaTrainer.

4. The app is now ready for training which is started by pushing the "Feedback" button.

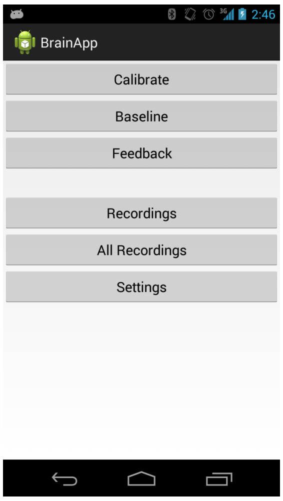
The prototype app uses standard Android UI components (see Figure 4.3a) and the interaction form is manual as it is clear from the training procedure listed above. In designing and building the prototype we thought in terms of *availability* regarding app functionality and the information dimensions present in the feedbacks (for example feedbacks showing current alpha state + performance history).

We tested the prototype informally on ourselves, fellow students, friends and a domain expert within BCI and HCI. From our testing three notions became clear.

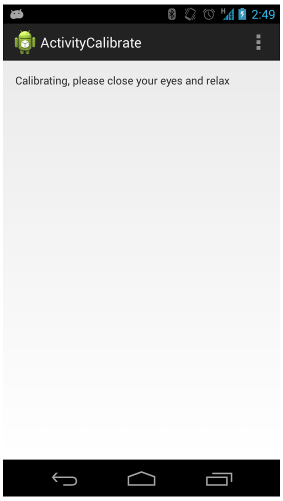
First, we had to revisit our manual interaction and focus on *accessibility* regarding app functionality. It is not enough that the individual alpha peak can be determined, baseline can be recorded, etc. when relying on the user to calibrate (twice a day) and record baseline (once a day) - the user should not even have to know about these concepts. We shifted our focus from *availability* to *accessibility* in delegating the responsibility of taking appropriate actions to the app by means of a proactive interaction described below in the design of AlphaTrainer (Section 4.3).

Second, we gained several important insights about the 5 feedbacks. Interestingly, the bar height was perceived inversely proportional to how it was designed. By design, a high bar represents high alpha from an alpha performance metaphor "more alpha is good". However, some subjects experienced the interface expressing state of mind from a "low mind activity is good" where the obvious goal state of the feedback would be a low bar. We adopted the latter metaphor when designing a new set of feedbacks for the next generation of interfaces. However, the bar was generally experienced to be unpleasant for other reasons - namely due to its radical shifts - why we excluded it from this point. During the testing of the prototype, the idea of supporting other modalities than sight was generated. This expanded the design space of the feedbacks and led to the development of audio and tactile feedbacks. We ended out with a set of 5 feedbacks: 2 visual (1 with history), 2 auditive (1 with history) and 1 tactile. They are described below in the design of AlphaTrainer (Section 4.3).

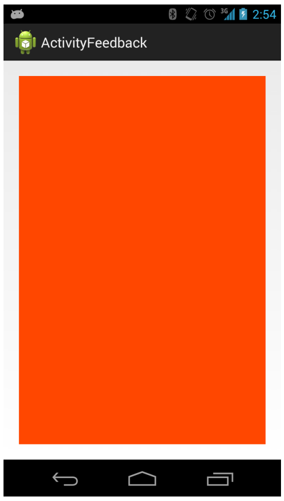
Third, we learned that the immediate impression of the app through the UI design was an important part of the general experience of using the app. When non-developers tried the prototype, they responded negatively to the standard Android layout (see Figure 4.3a) which looks a lot more rough



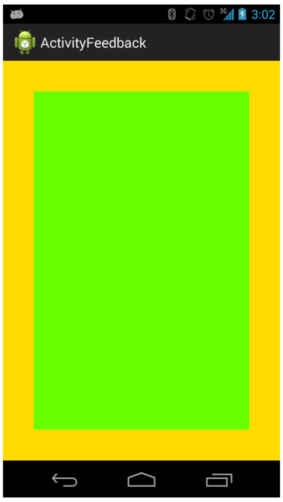
(a) Early prototype app screen start.



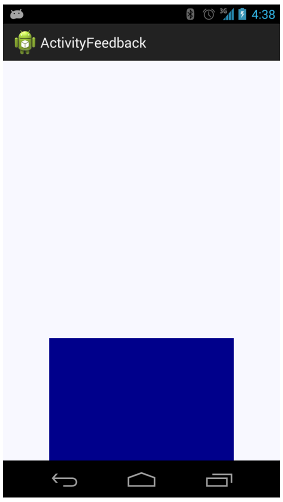
(b) Calibration screen.



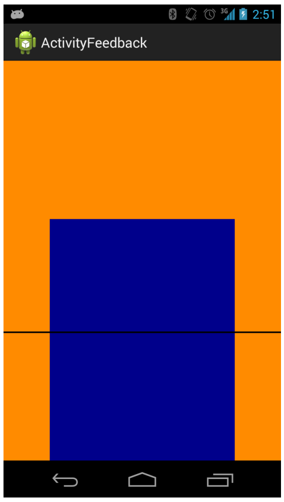
(c) Feedback box.



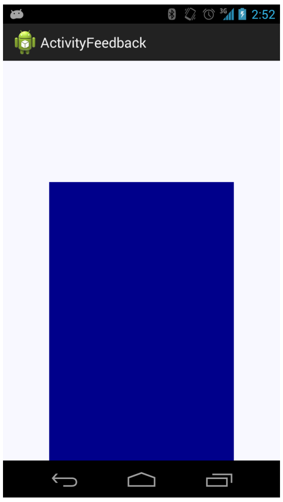
(d) Feedback box with history. Green is current feedback and border is history.



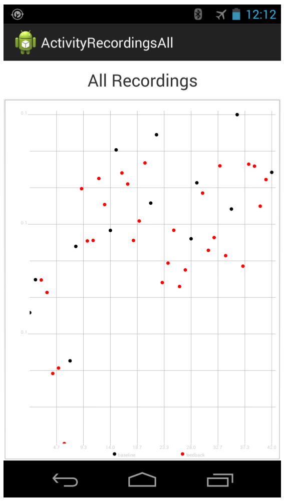
(e) Feedback bar goes up and down based on alpha level.



(f) Feedback bar with history.



(g) Feedback bar growing gradually based on the summed alpha.



(h) History screen: Baselines (black dots) and feedbacks (red dots).

Figure 4.3: Screenshots from the early prototype Android application using default Android UI components and manual interaction.

than they are used to from other apps. We noted that the general user is not able to abstract away from the graphical layout. Since we wanted to deploy the app for real world usage, we wanted to remove this obstacle of users distancing from the app due to its layout. Assisted by a digital designer, we updated the graphical design. The result can be seen below in the final design of AlphaTrainer (Figure 4.4).

Finally the last iteration of our prototype consisted of a test and analysis by an interaction design expert and a pilot evaluation performed by our selves. This had a set of concrete outcomes:

- The app should yield training performance immediately after a training has been performed.
- When headset connection fails, try to reconnect and/or show the user as precise as possibly where the problem is.
- History of training performance should be kept in the simple app design with a *high ink-to-information ratio* [62].

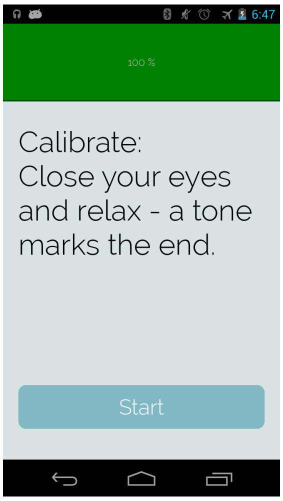
This ends our chronological journey through the design process. We now move on to describe the AlphaTrainer system in its current state - the version used in the user evaluation (Chapter 6).

4.3 AlphaTrainer

This chapter concludes with a description of the resulting AlphaTrainer prototype design. When the app is launched, the user is met by the home screen (Figure 4.4a). The home screen contains a logo and the 3 buttons: “Training”, “History” and “Settings”. The logo is the Greek letter alpha, which we imagine the user of AlphaTrainer (our personas) might recognize. The colors form a toned down blue palette and in conjunction with the logo we aim at conveying that AlphaTrainer is a serious tool.

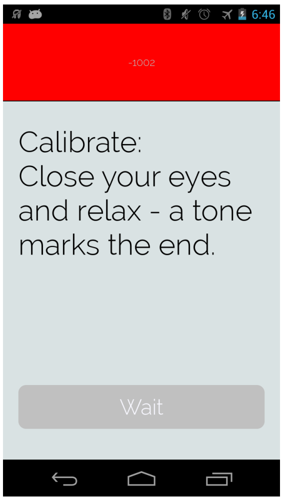
When the user clicks “Training”, the app takes on a proactive interaction. It decides whether it needs to calibrate or to record a baseline before training can be performed. The appropriate action is decided from these rules: (*i*) if calibration has not been performed within 8 hours, calibrate now; (*ii*) if a baseline has not been recorded today, record one now; finally (*iii*) if the first two criteria are met, the user is set to perform alpha feedback training. The headset connection status is shown in top of the screen both as a number (0 - 100%) and by a color varying between red (0%) over yellow (between 0% and 100%) to green (100%) (Figure 4.4b). While connection is not yet established,

(a) Start screen.

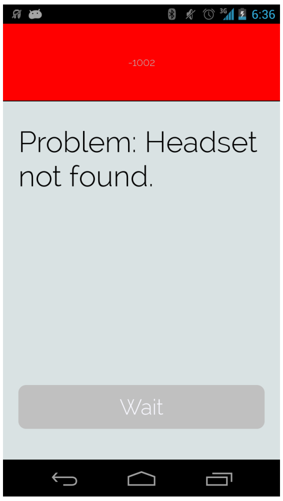


(b) Calibrate step 1.

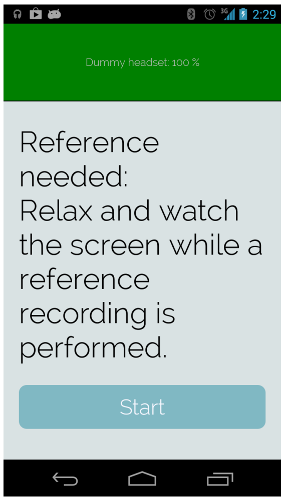
(c) Calibrate step 2.



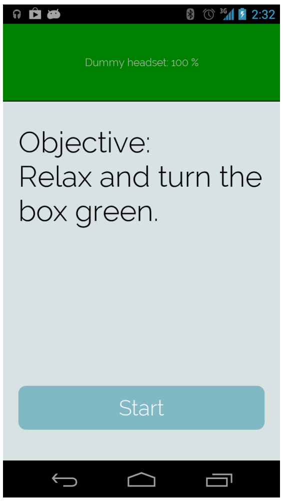
(d) Not connected (headset).



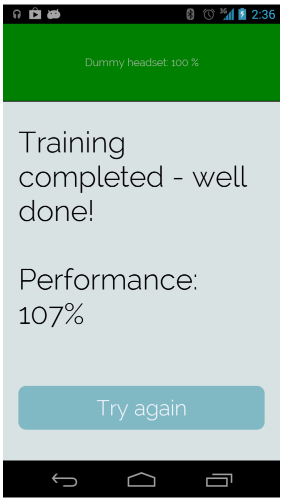
(e) No headset found.



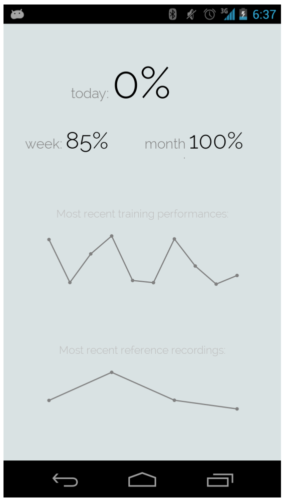
(f) Start ref. recording.



(g) Training feedback instructions.



(h) Training completed.



(i) Training history (per day, week and month). Plots the last 10 feedback recordings and the last 5 reference recordings.

Figure 4.4: Screen shoots from the app flow of the final AlphaTrainer Android prototype.

the button for starting the appropriate action is grayed out. Should connection to the headset fail, the app will try its best to tell the user what the problem is - for example in case the headset is turned off (Figure 4.4d). The app continuously tries to establish connection while being inside the "Training" area.

When ready to perform alpha feedback training, the app looks up which feedback has been selected in the settings. The trainees objective is presented, e.g. "Relax and turn the box green" in case of the box changing colors feedback mentioned earlier. We ended up with 5 different feedbacks:

1. Box changing colors. This feedback contains no performance history (Figure 4.5a).

2. Circles that move according to alpha level from the logic that high alpha amplitude (quiet state of mind) makes circle movement less. The circles are affected by gravity which conveys the recent performance history - when alpha level increases, it takes some time for the circles through the gravity to reach a state in which they are still (Figure 4.5b).

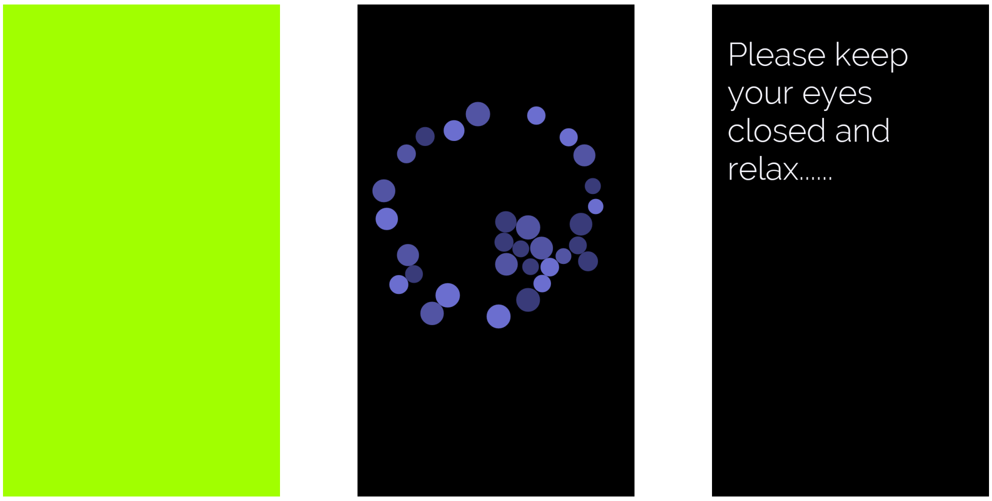
3. Vibration feedback in which high alpha amplitude makes the phone vibrate less. This feedback contains no performance history (Figure 4.5c).

4. Tone generator in which the pitch of the tone represents the current alpha level from the logic that high alpha amplitude (quiet state of mind) lowers the pitch. This feedback contains no performance history (Figure 4.5c).

5. Bell sounds appearing in which the number of bells represent the alpha level from the logic that high alpha amplitude (quiet state of mind) makes a quiet sound scape by having fewer bell sounds appear. When alpha level increase, it takes some time for the bell sounds to disappear. Thereby the duration of the bell sounds convey performance history (Figure 4.5c).

After a training has been performed, the user is immediately presented with a number representing performance in percent. This number is calculated by comparing the currently ended training session to the most recent training before that (Figure 4.4h). The user can then choose to train again by pushing the "Try again" button. This concludes the "Training" area of the app.

When the user selects "History" from the home screen, she is presented with a simple history dashboard (Figure 4.4i). The training performance of current day, current week and current month is presented in the top and conveys the set of trainings compared to all training sessions performed. The dashboard also has two simple graphs showing the recent training sessions and the recent baseline recordings.



(a) Feedback 1 with changing colors objective is to turn the color green (from red).

(b) Feedback 2 with moving circles - objective less movements.

(c) Vibration, tone generator and bell sounds - they all share the same screen instruction. Objectives make less vibration, lower pitch of tone and less bells.

Figure 4.5: Screen shoots from the 5 feedbacks of the final AlphaTrainer Android prototype.

When the user selects "Settings" from the home screen, she is presented with a conventional Android settings screen which enables her to choose among the 5 feedbacks described above.

We have now presented the AlphaTrainer and are now ready to cover the actual implementations in detail in Chapter 5.

Chapter 5

Implementation

AlphaTrainer is a system composed of several components. This chapter presents each of their implementation. Figure 5.1 gives an overview of the different parts of AlphaTrainer and their relationship.

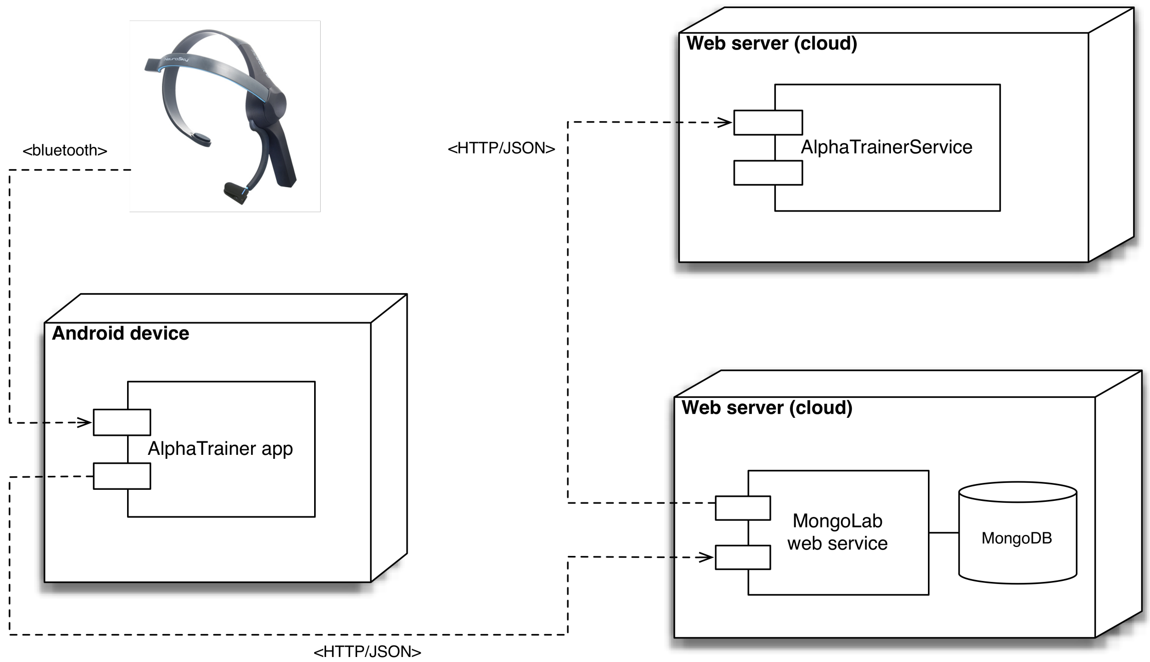


Figure 5.1: Overview of the AlphaTrainer system components

5.1 Signal processing library

Starting low level, we have implemented a small *C++* library for processing raw EEG data (Appendix *AlphaTrainer signal processing library*). It is based on the open-source OpenCV 2 library [1].

OpenCV is primarily used for computer vision and contains methods for image analysis and processing. OpenCV 2 has been rewritten from *C* to *C++* and is optimized to handle numerical data like vectors or matrices of float values [7]. This is exactly what we need when processing EEG data which at this level is represented as arrays of float values.

The implementation of this library is based on the signal processing and EEG data knowledge gained from the BCI Evaluation Chapter 3. The strategies, algorithms and data cleaning are roughly mirrored 1:1 in this library the main difference being that this library is intended for real-time usage.

Implementing the signal processing in C++ was chosen due to its performance advantages over Java in regard to numerical manipulation [55]. This may seem overkill for the current MindWave BCI configuration in which 512 data points are processed every second, but it secures scalability for future headsets presumably increasing sample rate and depth. For the same reason, we refrain from making the assumption that a headset carries only one channel by letting the library support multichannel EEG data [2].

The library can be called from the Java Virtual Machine (JVM) through the Java Native Interface (JNI) [3] [39].

Usage of the library is defined in the `opencvbrainprocessor.h` interface:

```
#include "opencv2/core/core.hpp"
#include "signalprocessingutil.h"

#ifndef OPENCVBRAINPROCESSOR_H_
#define OPENCVBRAINPROCESSOR_H_

float getBrainProcessed(float eeg[], int channels, int samples, int Fs,
                        float lowCutFq, float highCutFq, float alphaPeak);
float getAlphaPeak(float eeg[], int channels, int samples, int Fs);
```

[1] http://opencv.org
[2] Multichannel EEG would require small adjustments in regard to the communicating with the library.
[3] http://docs.oracle.com/javase/7/docs/technotes/guides/jni.

```
void getMinMax(float alphaLevels[], float* result, int alphaLevelsLength, int factor);
#endif /* OPENCVBRAINPROCESSOR_H_ */
```

This interface is implemented in `opencvbrainprocessor.cpp`. It contains three methods which in combination make up our signal processing. We now briefly explain the substance of each method.

The `getRelativeAlphaLevel()` method computes the relative alpha level taking raw EEG data and the alpha peak frequency as input. The procedure consists of: (*i*) applying FFT to the EEG data; (*ii*) then for each frequency bin of the FFT output, the value is squared (to go from amplitude to power); finally (*iii*) return the relative alpha level which is calculated by dividing the power of the alpha band (defined as 1 Hz to either side of the alpha peak frequency) with the power of the total band (defined as the band between the lowCutFq and highCutFq parameters). During the BCI evaluation we had good experiences with a low cut frequency of 5 Hz and a high cut frequency of 25 Hz (3.2.2). These values are currently used by AlphaTrainer.

The `getAlphaPeak()` method computes which frequency has the highest occurring amplitude within the alpha band (7 Hz - 13 Hz) taking raw EEG (recorded during calibration) as input. The procedure is this: (*i*) apply FFT to the EEG data; (*ii*) trim the resulting frequency domain down to the alpha band and; (*iii*) find the maximum amplitude and return the corresponding frequency.

Finally, the method `getMinMax()` computes how alpha levels should be mapped to feedback states. The resulting min and max values determine, for example, the thresholds for the Box feedbacks red and green states. It takes an array of relative alpha levels (from a baseline recording) as input. The procedure is to: (*i*) calculate mean and standard deviation of the input alpha levels; (*ii*) set min/max values to mean +/- the standard deviation multiplied with a factor (a method parameter); finally (*iii*) if min < 0 it is replaced by the lowest appearing value in the input data.

By design, this signal processing library is kept stateless. For example, it receives the alpha peak frequency and sample rate for every alpha level processing call. Keeping the processing module stateless simplifies the integration with the app by allowing a low coupling. This will be elaborated in the following section in which the library's user - the Android app - is explained.

5.2 Android app

Android is an open source operating system (OS) for mobile devices initiated by Google. Our rationales for choosing this platform includes that: (*i*) fits well for prototyping; (*ii*) has gained wide adoption; and (*iii*) has unrestricted deployment to end users through the Google Play store.

Android is a mature OS and is currently released in version 4.4. AlphaTrainer requires Android 3.0 (API level 9) or later [4].

An introduction to the Android programming model is out of scope - the interested reader is referred to the comprehensive official documentation [5]. Instead, this section will focus on the overall architecture and the most important design patterns used in the AlphaTrainer Android app implementation.

5.2.1 Overview

An overview of the architecture is shown in Figure 5.2. The diagram is by no means exhaustive and the aim is to highlight the relationships between BCIs (headsets), data processing, feedback views, application state and persistent storage. This diagram will function as the reference throughout the presentation of the Android app in the succeeding sections.

A main architectural goal of the implementation is a modular design which enables replacement of the different components i.e. the headset, data processor or the feedback view. How modularity is achieved will emerge during the discussion of the individual components and their relations to other components.

5.2.2 State management

We take a singleton approach [14] to global app state management. The singleton is implemented in the `App` class which extends the Android framework `Application`. In short, the Android framework guarantees that exactly one instance of the class is loaded at all times throughout the lifetime of the application, which makes it safe to write a `getInstance()` method [6]. Our application singleton is the center point for: (*i*) reading user preferences which takes place in the `SessionManager`; (*ii*)

[4] Main parts of the AlphaTrainer Android App work even from version 2.3 (API level 9) and up except of the Circle feedback covered within Section 5.2.6 which relies on browsers that implement Scalable Vector Graphics (SVG).
[5] http://developer.android.com/
[6] http://developer.android.com/reference/android/app/Application.html

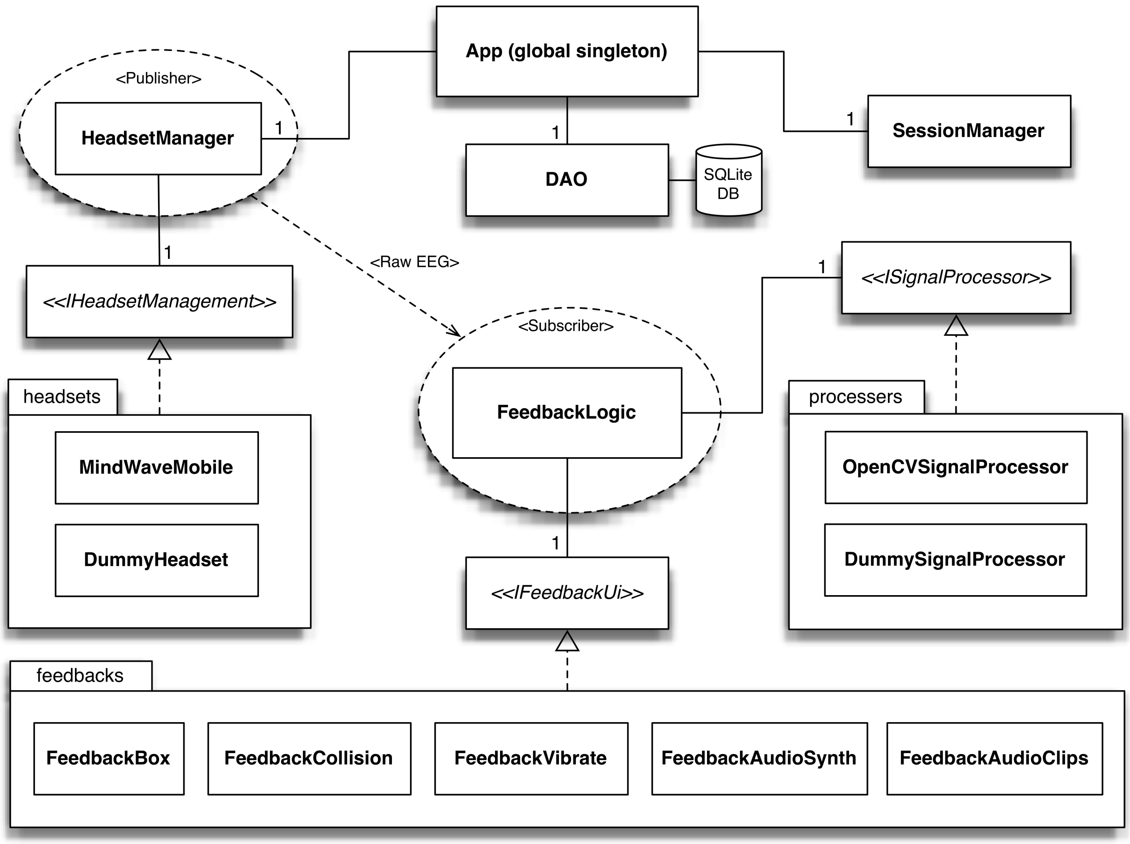


Figure 5.2: AlphaTrainer Android app architecture overview.

reading and writing from persistent storage which takes place in the data access object `DAO`; and (*iii*) managing headset connection which is takes place in `HeadsetManager` as well as in the singleton itself.

5.2.3 Headset

To support future consumer BCIs such as the already announced Muse and Emotiv INSIGHT (see Section 2.2.4), we have put effort into being as headset agnostic as possible. Implementing the signal processing ourselves is one way of doing so since this allows us not to rely on the signal processing abilities of a specific BCIs SDK.

Another way to obtain headset agnosticism is by communicating with the specific BCIs through a thin interface thus abstracting from the specific BCIs API by wrapping it in an implementation of the <<`IHeadsetManagement`>> interface:

```java
package dk.itu.alphatrainer.interfaces;
public interface IHeadset {
    public int getNrOfChannels();
    public int getSampleRate();
    public int getIcon();
}
public interface IHeadsetManagement extends IHeadset {
    public void startDataStream();
    public void stopDataStream();
}
```

Only the HeadsetManager uses the <<IHeadsetManagement>> interface. All other communication with the headset goes through the methods defined by <<IHeadset>> through which meta-data about the headset can be accessed.

AlphaTrainer currently contains two <<IHeadsetManagement>> implementations: DummyHeadset and MindWaveMobile - both in the dk.itu.alphatrainer.headset package. The DummyHeadset is used for development and testing and it outputs a random signal. The MindWaveMobile implements and uses the *ThinkGear* API [7] bundled with the MindWave Mobile headset (Section 2.2.2). Concrete communication with the MindWave headset goes through the TGDevice class. An example of the <<IHeadsetManagement>> interface implementation is shown below:

```java
package dk.itu.alphatrainer.headset;
...
import com.neurosky.thinkgear.TGDevice;
public class MindWaveMobile implements IHeadsetManagement {
    private BluetoothAdapter bluetoothAdapter;
    private TGDevice tgDevice;
    ...
    public MindWaveMobile(IHeadsetListener listener) {
        this.listener = listener;
    }
    ...
    @Override
    public void startDataStream() {
```

[7] http://developer.neurosky.com/docs/doku.php?id=thinkgear_connector_tgc

```
        ...
        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        ...
        if (bluetoothAdapter != null) {
            tgDevice = new TGDevice(bluetoothAdapter, handler);
        }
        ...
        tgDevice.connect(true);
        ...
    }
    @Override
    public void stopDataStream() {
        tgDevice.stop();
        tgDevice.close();
    }
    ...
```

Distribution of data from the headset is implemented in a publish subscribe pattern. In order to receive data from the headset, a class must implement the <<IHeadsetDataListener>> interface:

```
package dk.itu.alphatrainer.interfaces;
    public interface IHeadsetDataListener {
    public void onDataPacket(int channels, float[][] data);
}
```

Whenever a headset receives data, it calls its <<IHeadsetListener>> with the data formatted in the form of an array of float arrays each representing an EEG channel. This will always be the `HeadsetManager` which is the only class implementing the <<IHeadsetListener>> interface which generalizes the <<IHeadsetDataListener>> and <<IHeadsetConnectionStatusListener>> interfaces. The HeadsetManager then distributes the data to each of its registered <<IHeadsetDataListener>> thus carrying out the publisher act. Below we see an example showing how the private class `CalibrationDataHandler` within `ActivityCalibrate` first registers itself for receiving EEG data (l. 7) and later calls the headset for meta-data (l. 8-9).

```
1  package dk.itu.alphatrainer.calibration;
2  ...
```

```java
private class CalibrationDataHandler implements IHeadsetDataListener {
    ...
    public CalibrationDataHandler(ActivityCalibrate parent) {
        this.parent = parent;
        App.getInstance().getHeadsetManager().subscribeData(this);
        Fs = App.getInstance().getHeadsetManager().getHeadset().getSampleRate();
        int numberOfChannels = App.getInstance().getHeadsetManager().getHeadset().getNrOfChannels();
        ...
```

The distribution of connection status updates is similarly implemented in a publish subscribe pattern using the <<`IHeadsetConnectionStatusListener`>> interface.

Since the headset can be changed by the user through a settings entry and we generally want the app to depend as little as possible on the concrete headset implementations, we have implemented the factory pattern. Specifically, the headsets are instantiated by the `HeadsetFactory` in the package `dk.itu.alphatrainer.factories` which is responsible for instantiating a concrete headset appropriate for the current headset settings entry.

5.2.4 Signal processing

The implementation of the signal processing takes the same approach as was taken with the headsets to achieve modularity. Again, the communication goes through a thin interface which allows for easy replacement of the signal processing module.

```java
package dk.itu.alphatrainer.interfaces;
import dk.itu.alphatrainer.model.AlphaMinMax;
public interface ISignalProcessor {
    public void getBandPower(float[][] data, int Fs, float alphaPeak);
    public float getAlphaPeak(float[][] data, int Fs);
    public AlphaMinMax getMinMax(float[] alphaPowers);
}
public interface ISignalProcessingListener {
    public void onSignalProcessed(float bandPower);
}
```

<<ISignalProcessor>> is currently implemented by DummySignalProcessor and OpenCVSignalProcessor found in the dk.itu.alphatrainer.signalprocessing package. The DummySignalProcessor is for development and testing, and it simply outputs dummy values. OpenCVSignalProcessor encapsulates the OpenCV based library described in Section 5.1. In the example below, important lines of OpenCVSignalProcessor are shown in which it receives its listener in the constructor and declares its usage of the C++ library (Section 5.1) through the native keyword:

```
package dk.itu.alphatrainer.signalprocessing;
...
public class OpenCVSignalProcessor implements ISignalProcessor {
    ...
    private ISignalProcessingListener listener;
    public OpenCVSignalProcessor(ISignalProcessingListener listener) {
        this.listener = listener;
    }
    ...
    public native float getRelativeAlphaLevel(float[] eeg, int samples, int Fs, float alphaPeak);
    ...
```

5.2.5 Data flow

This section describes the data flow and call hierarchy involved from the point where the BCI receives raw EEG data up till the point where the feedback view is called with a processed and normalized alpha level.

As can be seen in the app overview diagram Figure 5.2, the FeedbackLogic class connects the raw EEG data, the signal processing and the feedback UI. Figure 5.3 shows a flow diagram of the calls involved from raw EEG is received to the feedback UI is called. Notice how all calls between the classes are interface method calls and would be exactly the same for other combinations of headset, processor and feedback UI.

The diagram also shows our usage of the publish subscribe pattern in that the HeadsetManager distributes the received EEG data to all listeners registered for receiving data.

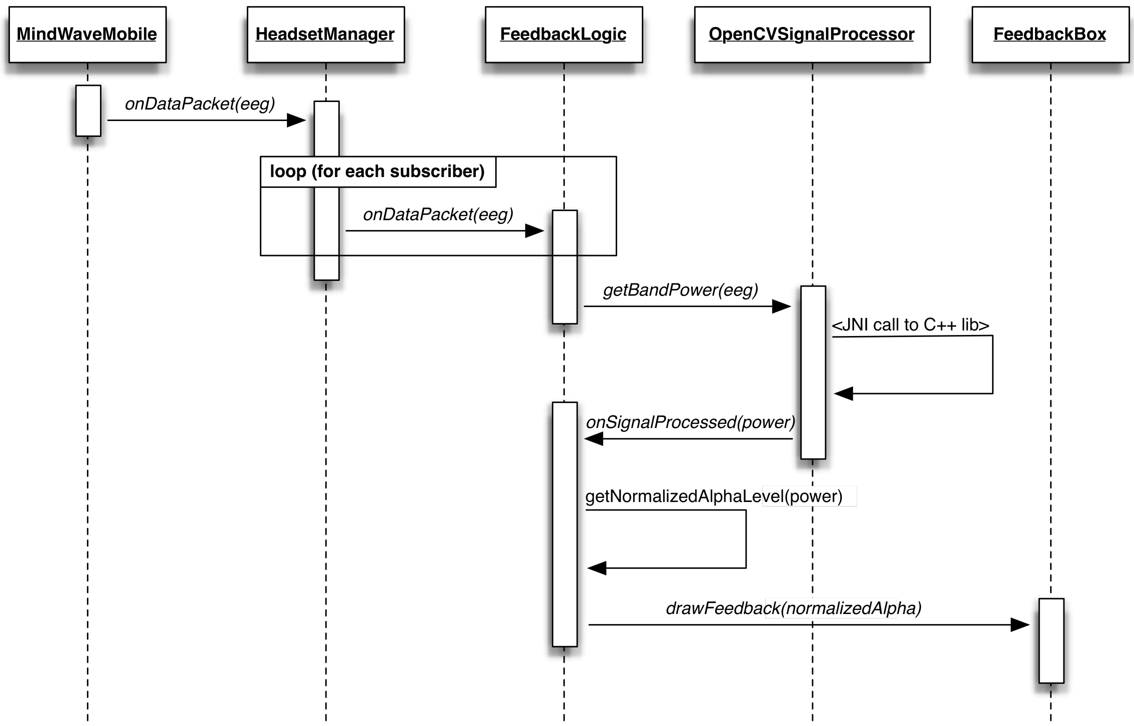


Figure 5.3: Flow of data from from headset receiving raw EEG to the feedback is drawn.

5.2.6 Feedback user interfaces

A feedback UI is generalized in a simple interface:

```
package dk.itu.alphatrainer.interfaces;
public interface IFeedbackUi {
    public void drawFeedback(int i);
    public void stop();
}
```

The `drawFeedback()` method takes a normalized alpha level from 0 to 100 and transforms this number into a visible, auditive or tactile feedback to the user. We have currently implemented 5 different feedback UIs found in the `dk.itu.alphatrainer.ui.feedback` package - reference the Design Chapter 4 for a visual and conceptual explanation of each feedback Figure 4.5. The feedback package consists of:

1. *Box* is implemented in the `FeedbackBox` class. It uses build-in Android animation features to make the transitions from red to green depending on the normalized alpha level.

2. *Circles* is implemented in the `FeedbackCollision` class. It uses an Android WebView and the circles are based on the open-source visualization Javascript library *d3.js* [8]. The WebView loads the Javascript and HTML/CSS from the Android specific *assets* folder within the AlphaTrainer Android app. On receiving a normalized alpha level, `FeedbackCollision` dispatches it to the WebView which handles the feedback actuation in a Javascript method.

3. *Vibration* is implemented in the `FeedbackVibrate` class and utilizes the Android system service architecture through which the vibration of the phone can be accessed.

4. *Tone* is implemented in the `FeedbackAudioSynth` class. It uses the build in Android audio library in conjunction with pure Java to crate a small synthesizer.

5. *Bells* is implemented in the `FeedbackAudioClips` class. It uses another build in Android audio library able to load and play sound clips from files.

Similar to the headsets, the feedbacks can be changed by the user through a setting. For the same reason, we also use the factory pattern to instantiate a concrete feedback UI class. This responsibility is delegated to the `FeedbackUiFactory` which read the user setting and instantiate the appropriate feedback.

5.2.7 Models and persistent storage

We have benefited of the built in SQLite [9] database within the Android framework. All training data such as alpha peak frequencies and the relative alpha levels are timestamped and persistently stored in the SQLite database from where they can be queried with SQL [10]. The raw EEG data is currently stored in plain text files on the mobile device's SD card.

All concrete database communication takes place in the `DAO` class from the `dk.itu.alphatrainer.datastorage` package. This encapsulation represents another typical design-pattern with Java and Android [64].

We have modeled a training and a processed alpha level respectively as `Recording` and `AlphaLevel` classes within the `dk.itu.alphatrainer.model` package. Below a snippet from the `AlphaLevel` class:

```
package dk.itu.alphatrainer.model;
...
import com.google.gson.annotations.Expose;
```

[8] *d3.js* is written in Javascript and it heavily relies on SVG - http://d3js.org
[9] http://www.sqlite.org/
[10] http://en.wikipedia.org/wiki/SQL

```java
import com.google.gson.annotations.SerializedName;
...
public class AlphaLevel {
    @Expose
    @SerializedName("alpha_level")
    private float alphaLevel;
    @Expose
    @SerializedName("normalized_alpha_level")
    private int normalizedAlphaLevel;
    @Expose
    @SerializedName("time_stamp")
    private long timeStamp;
    public AlphaLevel(float alphaLevel, int normalizedAlphaLevel, long timeStamp) {
        this.alphaLevel = alphaLevel;
        this.normalizedAlphaLevel = normalizedAlphaLevel;
        this.timeStamp = timeStamp;
    }
    public float getAlphaLevel() {
        return alphaLevel;
    }
    public int getNormalizedAlphaLevel() {
        return normalizedAlphaLevel;
    }
    public long getTimeStamp() {
        return timeStamp;
    }
}
```

Some fields of the `AlphaLevel` class are annotated with `@Expose` and `@SerializedName`. This is part of the Gson library [11] which we use to serialize the models to JavaScript objects (JSON) [12]. With the annotations we can control which fields of the class should be serialized and how the naming should be. We post the JSON serialized training data to a cloud based web service over HTTP, which is elaborated below in Section 5.3). The JSON training object has this structure:

[11] https://code.google.com/p/google-gson
[12] http://json.org

```
{
    "user_id" : "9cc6e095-4efa-4cc2-85ff-c792b34e40f5" ,
    "type" : "Baseline" ,
    "alpha_levels" : [
        {
            "time_stamp" : 1382695986 ,
            "normalized_alpha_level" : 6 ,
            "alpha_level" : 0.016940186
        },
        ...
    ] ,
    "headset_type" : "mindwave_mobile" ,
    "feedback_ui_type" : "feedback_audio_clips" ,
    "average_alpha_level" : 0.07930513 ,
    "time_stamp_start" : 1382695986 ,
    "time_stamp_end" : 1382696286 ,
    "length" : 300 ,
    "max_alpha_level" : 0.15 ,
    "min_alpha_level" : 0.008 ,
    "alpha_peak_fq" : 9.362069
}
```


The `user_id` is in an id anonymously generated client side (in the Android app) using the `UUID` class from the standard Java library. Generating and storing the id client side ensures that the web service cannot know who the user behind the id is.

The actual post to the cloud based storage is handled in a non-blocking asynchronous task - the `PostRecordingToServiceTask` class in the `dk.itu.alphatrainer.cloud` package - which is called when a feedback or baseline recording is finished. If the recording is not successfully posted to the service (which is identified by the absence of a returned row-number from the service), this will be attempted again later.

5.3 Cloud based storage

The AlphaTrainer system currently relies on a cloud based NoSQL database to store training data from its users (Appendix *Mongolab cloud storage*). We currently use the open-source MongoDB [13] database hosted in the cloud as a service by MongoLab [14] - but MongoDB can in principle be installed anywhere. MongoDB is a document based database opposed to a traditional relational database like SQLite which we use within the AlphaTrainer Android app (Section 5.2.7). MongoDB stores data, for example a training session including an array of alpha levels, as documents grouped into collections - the documents are very similar to JSON objects [49].

This database storage choice for the current AlphaTrainer system is founded in the *software as a service* (SaaS) concept - "*... SaaS delivers software and data as a service over the internet ...*" [25]. Advantages of the cloud based MongoDB at MongoLab include that we can access it: (*i*) through command line and do queries directly in the database; (*ii*) through a specific language dependent driver for example for Java; finally (*iii*) MongoLab exposes an RESTful web service API out of the box offering Create, Read, Update and Delete (CRUD) operations [17]. This cloud service component of the AlphaTrainer system enables the AlphaTrainer Android app to upload training data which can be consumed by the AlphaTrainerService Client explained below in Section 5.3.1 and any future clients with the need for accessing the data - for example a client created for usage by a clinical professional.

At last we show our usage of one of the features MongoDB offers, namely the ability to map reduce large data set - the map reduce concept is taken from functional programming [2]. The following example shows how all the average alpha level per user interface can be queried from all recordings in the training dataset through map reduction:

```
var mapFeedbackUIAvgAlpha = function() {
  emit(this.feedback_ui_type, this.average_alpha_level);
};
var reduceAvgAlpha = function(feedbackUIType, averageAlphaLevel) {
  return Array.avg(averageAlphaLevel);
};
function showFeedbackUITypeAvgAlphaLevelAvg() {
  db.trainings.mapReduce(
```

[13] http://www.mongodb.org
[14] https://mongolab.com

```
    mapFeedbackUIAvgAlpha,
    reduceAvgAlpha,
    { out: "feedback_ui_alpha_average" });
  return db.feedback_ui_alpha_average.find();
}
showFeedbackUITypeAvgAlphaLevelAvg();
```

First the function maps all the feedback user interfaces and average alpha levels from all the trainings and then reduce it by averaging the items created in the map step. This procedure produces the following JSON response:


```
{ "_id" : "feedback_box", "value" : 0.09779977771875001 } // Box
{ "_id" : "feedback_collision", "value" : 0.09649314506956522 } // Circles
{ "_id" : "feedback_vibrate", "value" : 0.14137012283529413 } // Vibrate
{ "_id" : "feedback_audio_clips", "value" : 0.0934931038888889 } // Bells
{ "_id" : "feedback_audio_synth", "value" : 0.11333089719230771 } // Tone
```


This in an example of a query which would be really interesting to run when AlphaTrainer (hopefully) has been used by many people over a long period of time. We return to this perspective in the Future works Section 7.1.

5.3.1 AlphaTrainserService Client

We have created a simple *webapp* client - AlphaTrainserService which can consume the cloud based storage at MongoLab (Appendix *AlphaTrainerService client*). At the moment this webapp simply allows viewing the recorded data grouped by (anonymous) users and their training sessions, see Figure 5.4.

The AlphaTrainserService webapp is implemented with the Play Framework [15] which is written in Scala [16] - a functional programming language built on top of Java [28]. The webapp uses the the Bootstrap front-end framework [17] for the user interface combined with BackBone.js [18] - a MVC based JavaScript framework. Bootstrap is highly flexible and met our simple UI requirements out

[15] http://www.playframework.com
[16] http://www.scala-lang.org
[17] http://getbootstrap.com
[18] http://backbonejs.org

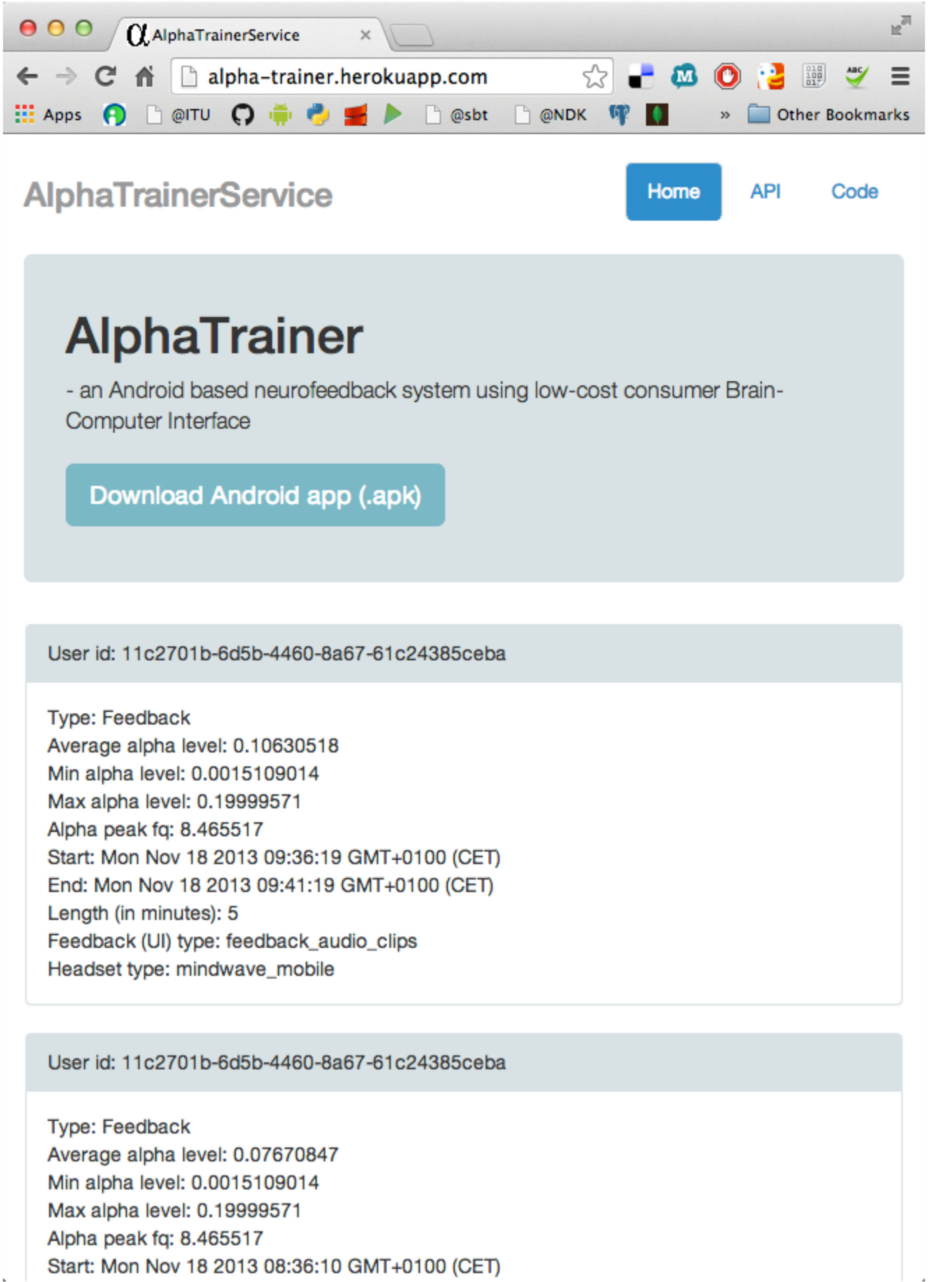


Figure 5.4: Screenshot of the AlphaTrainerService webapp

of the box. BackBone.js is used to continuously pull in new training recordings without reloading the webpage [48].

The AlphaTrainserService Client is currently deployed to Heroku [19] - a SaaS hosting environment built on top of the Amazon AWS infrastructure [20].

JSON and RESTful web service are first citizen concepts within the Play Framework. This means AlphaTrainserService could also expose a web service if, for example, the AlphaTrainer system should encounter requirements which can not be meet by the MongoLab web service. Additionally, the Play Framework can be used to build type-safe and a highly scalable web services as well.

With this cloud based client and data storage in hand we have a solid, flexible and highly scalable storage as an important infra structure underneath the AlphaTrainer Android application.

5.4 Summary

In this chapter, we have explained the central concepts of the implementation of AlphaTrainer. Most importantly, several design patterns have been applied to ensure a modular architecture of the Android app. This enables easy support for future headsets and new feedback views as well as replacement of the signal processing. The modularity ensures that the contribution of implementing the AlphaTrainer system still has value even if it should become apparent, for example, that the current state of BCIs are not good enough for performing neurofeedback.

Through the explanation of the Android app, only few Android specific components were mentioned. The same is true for the app overview diagram in Figure 5.2 which has no dependencies on Android. This carries the point that we have separated the logic as much as possible from the Android specific components. This makes it easy to port the application to, for example, iOS even though it will have to be rewritten in Objective C for the iOS environment. The concepts are the same and the native signal processing library can even be compiled to and called from iOS as well. The cloud based storage is of course agnostic in regard to the OS of its clients as long as they speak HTTP.

The scalable cloud based approach to data storage and the simple webapp consuming this data makes the proof of concept that the AlphaTrainer system could easily be integrated in, for example, a clinical context - in which case a client for the clinician would have to be designed and build for this

[19] https://www.heroku.com
[20] http://aws.amazon.com

purpose. This would raise some solvable security issues deemed out of scope for this AlphaTrainer system prototype.

Chapter 6

User evaluation

The evaluating of AlphaTrainer seeks to validate its design and investigate whether AlphaTrainer actually enables alpha feedback training in an everyday context. We derived a set of research questions targeting A the AlphaTrainer design and B its use in context:

A: How is training perceived and affected by specific feedbacks?

- are specific feedbacks perceived to be more pleasant/precise/etc than others?
- does specific feedbacks enable/disable training in certain situations?

B: How does alpha feedback training on a mobile device fit into an everyday context?

- is it hard to make time/room for training?
- are there (practical) obstacles for alpha feedback training in an everyday context?
- is alpha feedback training perceived to be rewarding or hard work?

Limitations and scope

The biggest limitation of this evaluation lies in the number of evaluation participants. Due to the hardware dependency on a BCI, we can only involve a limited number of participants. We have 2 such interfaces at our disposal which means we can run our study with 2 participants at a time. We have chosen a duration of 1 work week per participant and we will run the evaluation for 2 weeks which gives us a total of 4 participants only.

In regard to initially verifying the *design* of AlphaTrainer, we can gain useful data from even a few number of participants. For example, it is argued by Virzi that 4 participants is enough to uncover

80% of a systems usability problems [63]. We compensate for the low number of participants by collecting both quantitative, qualitative and logged data.

Regarding the *usage in context*, 4 participants are too few to represent a target group. Accordingly, we will treat the evaluation findings only as an initial hint as to whether it makes sense for people to perform alpha training in an everyday setting on a mobile device.

We will not try to make any statistical arguments based on data collected from our 4 evaluation participants. However, we still collect and discuss numerical data but only to look for hints and suggestions.

The evaluation participants are acquaintances of us which carries the possibility that they are biased and want to "perform good". We will convey to our best ability that anything negative they might experience is valuable information to us. Furthermore, we will keep this potential bias in mind while observing and interviewing them.

Since we want to learn about the effects of different feedbacks, we choose a factorial experiment [42] in which the different *feedbacks* are independent variables (i.e. factors). This is somehow a trade-off since a longitudinal study [42] in which the *amount of practice* is the independent variable would have been exiting as well. We choose the former since a week is a very short period anyway for a longitudinal study trying to establish a training effect. As mentioned in the Background Chapter 2, neurofeedback training typically involves 40-80 sessions of 45-60 minutes duration. Thus, the investigation of training effect is deemed out of scope in this evaluation. We will pick up the challenge of setting up an experiment to document the training effect of AlphaTrainer in the Future Works Section 7.1.

6.1 Method

Before the experiment start, we met with each test subject to install AlphaTrainer on his phone (participant 3 + 4) or hand out a phone with AlphaTrainer installed (participant 1 + 2). The test subject is also handed out a MindWave BCI and instructed how to use the system, connect to the headset etc. in order to abstract away any learning curve of the system.

In order to learn about the different feedback types and modalities, the participants are instructed to change feedback every day according to a schema which is also handed out before experiment

Participant	Day 1	Day 2	Day 3	Day 4	Day 5
1	Box	Vibration	Bells	Circles	Tone
2	Vibration	Circles	Box	Tone	Bells
3	Circles	Tone	Vibration	Bells	Box
4	Tone	Bells	Circles	Box	Vibration

Table 6.1: The distribution of feedbacks across participants during their 5 days of training.

start (Appendix *User evaluation*). To avoid order effects, the distribution of feedbacks across the participants is balanced as shown in Table 6.1.

The experiment lasts for 5 working days in which the subjects are committed to perform at least two training sessions per day - preferably one training within a working context and one in a home context. Before initiating the evaluation, we ran a pilot for a week on ourselves in order to catch any experiment flaws and to validate the Android application robustness.

During the 5 days of the experiment, we collect data in the following ways: (*i*) survey; (*ii*) participant observation and a semi-structured interview; and (*iii*) data logging.

6.1.1 Survey

The participants are asked to answer a small questionnaire after each AlphaTrainer usage (Appendix *User evaluation*). A 5-point Likert scale is used, in which participants mark their agreement with the statements below from 1 (strongly agree) to 5 (strongly disagree) [38].

The first statement target the perceived relaxation benefit feedback type/modality:

1. I feel more relaxed after training than before

The next statements target the specific feedback type/modality:

2. The feedback was pleasant

3. The feedback was precise

The next 3 statements target the training in context:

4. I was disturbed by my surroundings

5. My surroundings were disturbed by me

6. It was easy to find time for training

The final statement targets the comfort of the MindWave headset:

7. It was comfortable to wear the headset

6.1.2 Participant observation and a semi-structured interview

Since alpha training in an everyday setting is not a well understood activity, we can not be sure which interesting issues could arise let alone capture them through a survey. Therefor we observe the participants while they perform their training which enables us openly to address the issues that might arise. Observing is done via *Skype* which is both more flexible (we can sit "stand by" until the participant is ready to train) and less intrusive for the training practice than if we were sitting next to the participant in their home or at their work.

While observing the training, we pay attention to: (*i*) interaction with app/headset, how is it used?; (*ii*) interaction with surroundings, does something enable/disable training?; and (*iii*) anything that has influence on the training.

We write notes while observing and anything interesting can be followed up in a short semi-structured post training interview structured by this brief interview guide: (*i*) follow up on observations; (*ii*) how was the training experience?; (*iii*) did the setting make a difference?; (*ii*-*ii*) did the feedback make a difference?; and (*v*) when and how have you been training since last time?

6.1.3 Data logging

While performing alpha feedback training, the processed training data and meta data (e.g. alpha levels, feedback type, etc.) is: (*i*) posted to a web service (Section 5.3); and (*ii*) stored locally on the phone as processed alpha levels along with the raw EEG data (Section 5.2.7). The collection of this data enables us to look for hints as to whether the system works as expected. E.g. it is expected that: (*i*) baseline alpha levels are generally higher when participants have closed eyes; and (*ii*) alpha levels are higher when the user actually controls the feedback compared to the baseline recording of the same feedback (feedback effect).

Furthermore, the raw EEG enables us to follow up on unexpected results in pursuit of what might be the cause (e.g. noise).

6.2 Participants

The participants are chosen among our acquaintances matching our target group - manifested in our personas - as much as possible (Section 4.1). They have been guaranteed anonymity in this thesis and have verbally agreed that we can use the data collected during the evaluation and show their photo. They count 3 males and 1 female with an average age of 33 years.

6.3 Result

After two weeks of evaluation we got 4 sets of data each including observation notes, semi-structured interview notes, questionnaires and the logged training data. The survey results are first presented and discussed. Then we move on to present the most important notions from the observations and interviews after which we present some insights from the logged data.

In total, 63 training sessions and 19 baselines were performed and 39 questionnaires were answered.

6.3.1 Survey results

To get an overview of the results, we have calculated mean and standard deviation of the questionnaire answers. One participant missed a training during the evaluation so the results are based on 39 filled out questionnaires. The results are shown in Table 6.2

The first statement **I feel more relaxed after training than before** relates to the immediately experienced effect of performing alpha training. It shows us that the participants were generally feeling slightly more relaxed after training. This does not vary significantly over the different feedbacks which could suggest that the main source of the increased relaxation is to be found in something general for the alpha training practice. That alpha training is perceived to be immediately rewarding is backed up by several observations during the evaluation period. For example, participant 1 stated that she experienced to become very relaxed using AlphaTrainer when training at work (she is a teacher and trained in her classroom). She also used it as a way to stress down when she got home after work.

The next 2 statements **The feedback was pleasant** and **The feedback was precise** relates to how the specific feedbacks were experienced. The Box and the Bells feedbacks are generally perceived to be a little less pleasant then the others. Looking at the std.dev. these two feedbacks are also the ones causing greatest division between the participants. Again, this is backed up during observations where participant 3 describes the Bells feedback as the best metaphor for training relaxation while participant 4 states that he got a shock whenever a bell sound was triggered. Regarding precision, the results are quite similar - the Box and the Bells feedbacks are perceived slightly less precise. Overall, the two questions do not reveal big differences between the feedbacks, rather that the participants had different feedback preferences.

The next 3 statements - **I was disturbed by my surroundings.**, **My surroundings were disturbed by me** and **It was easy to find time for training** - relates to the usage of AlphaTrainer in context. It appears that the participants were at times slightly disturbed by their surroundings and that the degree varies between trainings and participants. It also appears that the participants were hardly disturbing their surroundings. In general across all training sessions the participants found it quite easy to find time for training though this also varied some between the trainings. This means that if they only had to train once a day - which a participant mentioned during observations would be more natural than 2 times a day - it would become easy to find the time.

The answers to the final statement **It was comfortable to wear the headset** informs us that the participants were generally perceiving the MindWave as neutrally comfortable (which matches the findings in the BCI Evaluation Chapter 3) and thereby not a significant obstacle for the participants' usage of AlphaTrainer.

6.3.2 Observation and interview

In the process of analyzing the notes taken during the observations and the interviews, we condensed them into short sentences which were coded and categorized. From this overview, some general themes appeared which we present in this section.

First of all, none of the 4 evaluation participants experienced any technical problems using the system. At one point, we observed a training situation in which the AlphaTrainer app was not able to connect to the MindWave headset. The participant (1) immediately found out that her hair was tangled up in the power switch which had turned the headset off. She turned it on after which the app successfully connected to the headset and she continued her training. This was the closest to a technical problem we observed or heard about. This testifies to the robustness of our design and validates the technical feasibility of AlphaTrainer.

Statement	Box	Vibration	Circles	Tone	Bells
I feel more relaxed after training than before.	2.1 ±0.4	2.1 ±0.4	2.5 ±0.5	2.0 ±0.8	1.8 ±0.4
The feedback was pleasant.	2.6 ±1.0	2.1 ±0.6	2.0 ±0.9	1.9 ±0.7	2.7 ±1.4
The feedback was precise.	2.7 ±0.8	2.2 ±0.7	2.1 ±0.6	2.1 ±0.9	2.5 ±1.0
I was disturbed by my surroundings.	3.7 ±1.0	2.6 ±1.2	2.4 ±1.4	2.4 ±1.0	3.5 ±1.2
My surroundings were disturbed by me.	4.4 ±0.5	4.2 ±1.0	3.9 ±1.6	3.6 ±1.1	4.3 ±0.5
It was easy to find time for training.	3.1 ±1.1	2.6 ±1.2	3.0 ±1.1	3.0 ±1.4	2.2 ±1.0
It was comfortable to wear the headset.	3.0 ±1.0	2.1 ±1.0	2.9 ±0.6	2.7 ±1.0	2.7 ±0.5

Table 6.2: Post training survey results showing the degree to which participants agree with the statements from 1 (strongly agrees) to 5 (strongly disagree). Results are based on 4 participants answering a total of 39 questionnaires.

One of the themes emerging from the observation and interview notes relates to the interaction metaphors encountered in the Design Chapter 4. It became apparent that the participants perceive alpha training through different interaction metaphors. Especially one participant (2) who can be described as competitive in general noted that the feedbacks seemed opposite. He described that it would be more natural for him to experience greater feedback effect when his alpha level was highest, thus perceiving the feedback through the performance centered "more alpha means more feedback" metaphor. Conversely, another participant described how he felt that the Circles and Bells feedback were the most aligned with his mental model of the feedback describing his state of mind, thus perceiving the feedback through the "quiet mind (more alpha) means less feedback" metaphor.

Another theme from the observations is how the feedback in itself can have a negative effect on the relaxation and thereby on alpha performance. All participants at some point expressed that they would momentarily be interrupted from a relaxed state by the feedback. For example, one participant (4) explained that he got a shock every time a bell sound popped up as he is generally very sensitive to auditive stimuli. In most cases, however, the negative feedback effect occurred in situations where it was conveying poor alpha performance causing the participants to focus their attention on increasing alpha. As we know from the Background Chapter 2, this attention and state of alertness can reduce alpha activity. We suspect this to be somewhat related to the chosen metaphor behind the feedback in which the feedback is highest when alpha performance is lowest thus giving a "negative feedback". Interestingly, this negative effect was higher for certain feedbacks and it varied between the participants which feedbacks were giving this effect the most. For example, participant 2 finds the Tone stressing while participant 1 oppositely felt that the Tone was the least stressing and that the Box was most stressing feedback. The degree to which a feedback was - in their own words - "stressing" was often a reason given by the participants when stating that one feedback is better than another.

A somehow related theme from the participant observation and interview notes is that knowledge of the nature of alpha waves is an important component in perceiving the feedback. Participant 2 explained that he experienced to get stressed when the Tone feedback increased pitch (meaning lower alpha) and that it did so in oscillations of a few seconds. We explained that alpha levels naturally follow the oscillating pattern that he had described which made him perceive the feedback very differently - he went from disliking the Tone feedback into liking it the most. For him, at least, the knowledge that high alpha activity naturally come in chunks of a few seconds made him much less stressed when he was given feedback on low alpha level by the Tone feedback. A similar experience was documented when interviewing participant 1 and the Box feedback.

Last, we present some general notions on alpha training *in situ*. Most notably, the usage of AlphaTrainer was generally very successful. The participants all found it rewarding to perform alpha training in their daily lives. This is an interesting and promising result - that the actual training practice is perceived as beneficial - since the training effect (increased alpha over time) presumably have changed very little during the week of evaluating. The participants related the alpha training practice to yoga (participant 2) and meditation (participant 3 and 4) and that AlphaTrainer fit among such de-stressing practices makes a strong argument for its feasibility.

6.3.3 Analysis of logged data

As mentioned earlier, the data logged during the evaluation is not suitable for statistical analysis due to the low sampling size, the varying feedback views and the uncontrolled environment. However, we can still look for some indications that the system performs as expected.

The first thing we look for is whether we can see higher alpha levels in the baselines recorded with closed eyes (Vibration, Tone and Bells feedbacks) than the baselines recorded with open eyes (Box and Circles feedbacks). This comparison is the closest we can get to the method we used when comparing BCIs ability to measure alpha in Chapter 3, and we expect higher alpha levels when the participants have closed eyes. That said, there are a number of factors not controlled in this comparison (e.g. the recording setup) and we can not account for the feedbacks impact on the participants ability to relax. In sum, we must be careful to not draw unsupported conclusions from the result of this comparison. To get a little more data, we decided to include the data collected during the pilot evaluation since the procedure for training did not change between the pilot and the user evaluation. The results are shown in Table 6.3.

The results show the tendency expected, that the baselines recorded with eyes closed are generally higher than the baselines recorded with eyes open. Participant 2, however, differs for some reason consistently from this tendency. In his baselines, all recordings with open eyes are higher than the recordings with closed eyes. This could be due to a number of reasons including noise, a negative effect of the feedback stimulus or that he did not actually close his eyes eyes.

Another way of looking at the data is to order each participants baselines according to alpha level (Appendix *Ordered baseline recordings*). Excluding the baselines from participant 2, it only appears in 3 cases of the 24 remaining baselines that a baseline recorded with closed eyes was lower than a baseline recorded with open eyes. If AlphaTrainer was not able to detect alpha levels, we would have expected the order of the baselines to be random. The order of baselines and the general trend that baselines recorded with closed eyes are higher than baselines recorded with open eyes suggest that alpha levels are actually measured by AlphaTrainer.

The second thing we will look for in the data is the feedback effect - the difference between the baseline and the training recordings under the same feedback. We expect alpha levels to be highest during training when the participant's alpha level control the feedback. We have calculated each relationship between a training and a baseline recording (by dividing them) - which expresses the feedback effect - and for each feedback, we have calculated the mean feedback effect per participant.

User	Average baselines (open eyes)	Average baselines (closed eyes)	Closed eyes / open eyes
Participant 1	0.0885 (2)	0.0941 (3)	1.0632
Participant 2	0.1202 (2)	0.1021 (2)	0.8488
Participant 3	0.1120 (2)	0.1143 (3)	1.0208
Participant 4	0.1041 (2)	0.1867 (3)	1.7933
Pilot 1	0.0885 (2)	0.1098 (3)	1.2406
Pilot 2	0.0720 (2)	0.1041 (2)	1.4469

Table 6.3: Average alpha of baselines per user grouped by open and closed eyes feedbacks. They are divided to get their relationship. We expect baselines under closed-eyes condition to be higher than baselines with open-eyes condition.

Again, we have chosen to include the data from the evaluation pilot. The results are shown in Table 6.3.4. The number in parenthesis to the right of the feedback effect shows how many training sessions it is based on. The table also shows the average feedback effect per feedback weighted according to the number of training sessions.

Looking at the feedback effect for participant 3 and 4, they are for some reason quite consistently negative. The weighted average feedback for participant 3 is 0.90 and for participant 4 it is 0.86. Again, it is hard to comment on the reasons for this, but some reasons could be their mental strategy while training, noise (for example, person 4 mostly trained in uncomfortable backstage areas) or that they are so-called "non-responders" unable to alter their brain frequencies which according to [59] has been repeatedly reported and which excluded 26% (6 of 23) of the participants in their study. For this reason, we have calculated the weighted average feedback effect excluding participant 3 and 4.

The results show a slight general tendency of alpha levels being higher when the participants are in control of the feedback. Especially when excluding participant 3 and 4, we see a quite consistent positive feedback effect with only 3 out of 18 exceptions from the pattern that alpha levels are higher during feedback than while recording the baseline. Looking at participant 1, the feedback effect seems especially consistent across the 21 recordings she performed during the evaluation (5 baselines and 16 training sessions). The consistent pattern of alpha level being higher under feedback control (opposite but still consistent in case of participant 3 and 4) together with a clear feedback effect of participant 1 suggest that AlphaTrainer's feedback mechanism does have an effect. If there

	Box	Circles	Vibration	Tone	Bells
Participant 1	1.24 (2)	1.11 (3)	1.20 (4)	1.23 (3)	1.11 (4)
Participant 2	1.24 (2)	0.91 (2)	1.08 (3)	1.02 (3)	N/A
Participant 3	0.81 (3)	0.83 (6)	1.02 (4)	0.85 (5)	1.11 (2)
Participant 4	0.80 (4)	1.01 (3)	0.93 (3)	0.86 (3)	0.64 (2)
Pilot 1	1.25 (2)	0.97 (1)	0.80 (1)	1.01 (3)	1.10 (3)
Pilot 2	1.23 (6)	1.23 (6)	N/A	1.02 (5)	1.10 (3)
Weighted Average all	1.08	1.02	1.05	0.98	1.04
Weighted Average all excl. 3 + 4	1.24	1.12	1.11	1.06	1.03

Table 6.4: Average of all alpha per feedback per user divided with the baseline for that specific feedback. The number of times the feedback was used per user is in (<num>). Some baselines are unfortunately not available and mark with N/A. The Pilot data is included.

was no feedback effect, we would expect to see a more random distribution of alpha levels - especially across the participants.

We will refrain from drawing conclusions on the difference between the feedbacks due to no clear indications in the data and the small sample size. This would be very interesting to explore in a future experiment though.

6.3.4 Sum up of results

First of all, the biggest result of the evaluation - directly addressing our thesis hypothesis - is that *it is feasible to build a system for performing alpha feedback training using a mobile device and a consumer BCI*. We base this result on these notions from the evaluation of AlphaTrainer: (*i*) the participants generally responded positively to using AlphaTrainer. The training practice was understood by the participants as a form of de-stressing practice and perceived to be immediately rewarding; (*ii*) all participants could see themselves using AlphaTrainer in the future - several of them could see them selves buying a BCI in order to use AlphaTrainer (though some would wait for the BCIs to get more discrete, comfortable and cheaper); (*iii*) the participants found it relatively easy to find the time to use AlphaTrainer; (*ii-ii*) during the evaluation, AlphaTrainer was successfully used in several situations including at home, at work and while traveling (e.g. in backstage areas); (*v*) while deployed for real world use, AlphaTrainer worked as expected without any technical problems. It was observed to handle volatility by reestablish BCI connection after an evaluation participant got the BCI's power switch tangled with her hair which turned the BCI off; additionally (*vi*) the data collected during the evaluation suggest that AlphaTrainer is able to measure the amount of alpha and that the feedback mechanism seems to positively affect the participants alpha level (in case of some participants at least).

Besides verifying our hypothesis, the evaluation has given some valuable insights to many aspects of AlphaTrainer. We now try to operationalize these insights by coming up with design changes to address the problems revealed and the ideas generated during the evaluation. The set of design revisions point towards the future work of AlphaTrainer.

- It became apparent during the evaluation that the participants perceived the feedbacks from two different metaphors, "more alpha means more feedback" and "quiet mind (more alpha) means less feedback". To support both perceptions, we suggest a user setting for reversing the highest feedback reaction from low to high alpha.

- A participant forgot to train one day. If he had not forgot, he would easily have had the time to train. We support his request for a reminder function to which could initiate a reminder notification if the user has not trained for some fixed amount of time.

- Another participant experienced many notifications - especially from emails which he receives a lot of. As a counter measure, we suggest to include some sort of in-app shortcut to turn on a quiet phone profile. An approach to this could be to rely on scripts for third party event-based applications such as On.X [1] or Tasker [2] which would allow a "put phone in silent mode and enable Bluetooth" script to be executed when the app launches.

- At one point during the evaluation, a participant had started training with a feedback scheduled for the day before. After changing to the feedback of the day, the AlphaTrainer app did not ask for a baseline recording since the logic only states that a baseline recording has to be performed once a day. As a result, we did not get a baseline with the feedback of that day which we missed for data analysis. Furthermore the participant did not get optimal feedback since the feedback intensity (e.g. when the Box feedback is red or green) was based on a baseline performed under a different feedback. As a countermeasure, we suggest to add the rule to the proactive interaction logic that switching feedback should always be proceeded by a baseline recording before training is allowed.

- Lastly the history view caused some confusions at times. Since alpha levels are higher when eyes are closed and the feedbacks shifted every day, the recent performances and baselines

[1] on{X} Android app: https://play.google.com/store/apps/details?id=com.microsoft.onx.app
[2] Tasker Android app: https://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm

often varied very much from day to day. A way of handling this could be to only include baselines and training performances of the same feedback as the currently selected. Another way to handle this would be to subtract some sort of "closed-eyes offset" which might allow for comparisons across different feedback modalities. Whether any of these suggestions are viable solutions to the problem would of course have to be tested in a future design iteration.

Finally we conclude that the method of putting AlphaTrainer out in real world usage has been productive. Through real world usage, we have learned a lot about neurofeedback training in practice and validated the design of AlphaTrainer in a way not possible in a lab study.

Chapter 7

Conclusion

This thesis has examined the feasibility of building a system comprising of a low cost consumer BCI and a mobile device which enables alpha feedback training embedded in everyday life. We started our investigations by evaluating three consumer BCIs of interest - especially regarding their ability to measure alpha waves. Based on the results, we chose to use the MindWave Mobile BCI since it is able to measure alpha and is the best exponent for the next generation of consumer BCIs.

We continued our investigation by designing, implementing and evaluating AlphaTrainer, an Android based alpha feedback training system. The AlphaTrainer app features:

- Neurofeedback with multiple modalities (vision, audio and tactile).
- Analysis of raw EEG data which enables feedback on individually determined frequency bands.
- Performance history tracking.
- Proactive interaction which keeps track of when to calibrate and record baseline.
- A modular architecture which enables addition of headsets, data processing modules and feedbacks.

We performed a real world evaluation in which 4 evaluation participants used AlphaTrainer for a week at home, at work and while traveling. We learned that:

- The participants experienced using AlphaTrainer to be immediately rewarding and perceived the training practice as a de-stressing exercise.

- Participants were generally positive about using AlphaTrainer and could see themselves use it and even buy a BCI in order to perform alpha training.
- The data collected during the evaluation indicates that AlphaTrainer deployed in a real world scenario is able to measure alpha and that the feedback mechanism increases alpha.
- The AlphaTrainer system performed as expected during the evaluation without noticeable technical problems.

From the evaluation results we are able to verify our hypothesis and conclude that *it is possible to build a neurofeedback system comprising of a consumer BCI and a mobile device.*

An interesting next step would be to investigate the training effect of using AlphaTrainer over a longer period of time. Though this is deemed out of scope in this thesis, we have initiated the first steps to make such an investigation. We have done so by planning a release of AlphaTrainer during December 2013 to the *Google Play Store* [1].

7.1 Future work

In the short term, we will finish an iteration of the AlphaTrainer design taking the concrete outcomes of the user evaluation as input.

In the long term, it would be interesting to investigate the efficacy of AlphaTrainer. As mentioned above, the AlphaTrainer app support data collection which enables anonymous tracking of individual development in alpha levels over time. Still, this method would not conclusively answer whether any observed increase in alpha was caused by the feedback mechanism - for example, an increase in alpha could be caused by just taking the time to relax, by a method (such as meditation) performed while using AlphaTrainer or some unrelated change in the subjects situation. The neurofeedback component of AlphaTrainer could be isolate and investigated in a controlled longitudinal study including three groups: (*i*) one group would perform alpha feedback training; (*ii*) a “placebo” group would think they performed alpha feedback training but without actually controlling the feedback (which could for example be pre-programmed); finally (*iii*) a third group would perform a control activity i.e. performing a de-stressing breathing exercise or just lying comfortably on the floor while getting their EEG activity measured. These three groups would perform an equal amount of their activities respectively over a substantial period of time. Comparing the development in baseline

[1] https://play.google.com

alpha levels between the groups would inform about the efficacy of the neurofeedback component of AlphaTrainer.

If AlphaTrainer in conjunction with the MindWave BCI - the configuration investigated in this thesis - turns out not to be effective, it would be interesting to see how it performs with future headsets. Same thing goes for new feedbacks and processing modules which are replaceable due to the modular architecture of AlphaTrainer. We speculate that a consumer BCI headset, at some point at least, will meet the requirements for neurofeedback training. This point illustrates that this thesis' contribution of designing, implementing and evaluating AlphaTrainer still has value should the case be that the current MindWave configuration of AlphaTrainer turn out not to be efficacious.

If (or when) AlphaTrainer on the other hand turns out to be effective, it opens up to a lot of exciting future research. It would be extremely interesting to investigate how AlphaTrainer or the neurofeedback principles of the AlphaTrainer app might fit in a clinical context. For example, patients could be ordered neurofeedback training on their phone as part of their treatment. This would give the patients the convenience of performing neurofeedback training where and when it fits them best. Clinical professionals would be able to monitor the training progress. The system could include an event model in which a clinical professional would be notified if a certain patterns in the training data occurred. The frequency band on which the feedback is given (the alpha band in case of AlphaTrainer) could be changed in order to widen the neurofeedback application areas to for example ADHD in which case beta and theta frequency bands are often the basis for neurofeedback training.

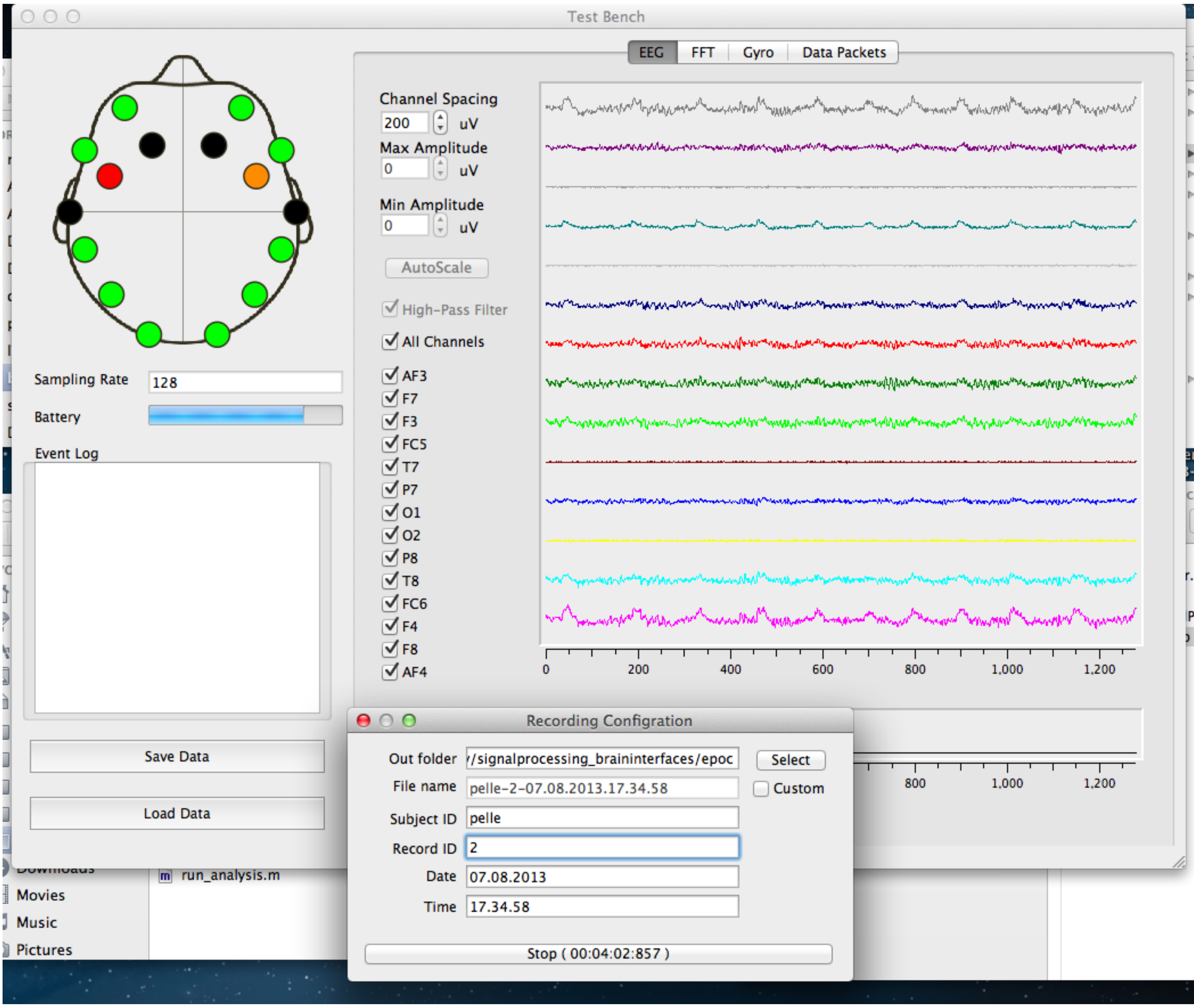
In a consumer context, it is an open and exiting question whether there is a base for adopting neurofeedback training as it is expressed in AlphaTrainer and how this potential consumer base develop over time when new generations of consumer BCIs appear. The evaluation hints that there could be a potential for AlphaTrainer adaptation - at least our 4 test participants found value in using AlphaTrainer. To further investigate this potential, our plan is to collect as much data as possible when releasing AlphaTrainer for free (or rather letting users pay with anonymous data) to the Google Play Store. This data, usage patterns and user feedback is valuable and could lead to the development of the next generation of AlphaTrainer, which might realize some commercial potential.

The vision for AlphaTrainer's future is to put the prospects of neurofeedback into the hands of everyone with a smart phone and a low-cost consumer BCI.

Appendix A

Emotiv EPOC TestBench

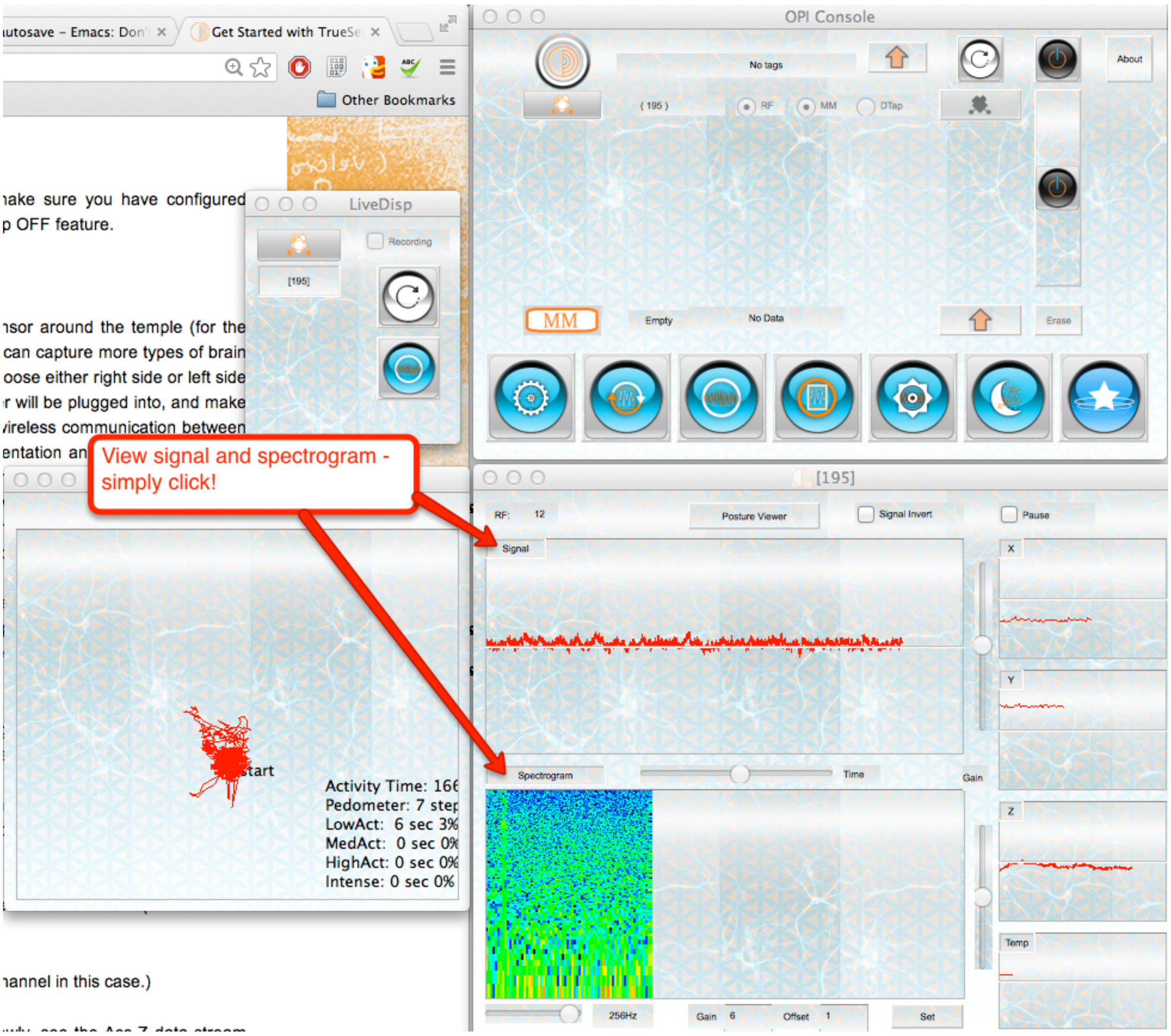
TestBench in action recording EEG. Dot marks the status of each electrode: (*i*) green if connected; (*ii*) yellow if nearly connected; (*iii*) read not connected - there is no skin contact at all; and (*ii*-*ii*) black marks a reference.


Test Bench
EEG
FFT
Gyro
Data Packets
Channel Spacing
200
uV
Max Amplitude
0
uV
Min Amplitude
0
uV
AutoScale
High-Pass Filter
All Channels
AF3
F7
F3
FC5
T7
P7
O1
O2
P8
T8
FC6
F4
F8
AF4
0
200
400
600
800
1,000
1,200
Sampling Rate
128
Battery
Event Log
Save Data
Load Data
Recording Configration
Out folder
/signalprocessing_braininterfaces/epoc
Select
File name
pelle-2-07.08.2013.17.34.58
Custom
Subject ID
pelle
Record ID
2
Date
07.08.2013
Time
17.34.58
Stop (00:04:02:857)
Movies
Music
Pictures
run_analysis.m

Appendix B

TrueSense Kit console

TrueSense Kit console in action recording EEG. Clock-wise:: (*i*) *LiveDisp* to enable a sensor for example number 195 - multiple sensors can be attached; (*ii*) console with access to live view and to the build in applications for yoga and sleep analysis; (*iii*) live viewer for sensor 195 printing signal as raw eeg and processed as a spectrogram; and (*ii-ii*) the posture viewer drawing movements.


View signal and spectrogram - simply click!
OPI Console
LiveDisp
Other Bookmarks
No tags
About
MM
Empty
No Data
Erase
Recording
[195]
Posture Viewer
Signal Invert
Pause
Signal
Spectrogram
Time
Gain
256Hz
Offset
Set
Temp
Activity Time: 166
Pedometer: 7 step
LowAct: 6 sec 3%
MedAct: 0 sec 0%
HighAct: 0 sec 0%
Intense: 0 sec 0%

Appendix C

Experiment - signal processing

C.1 Run analysis

C.1.1 Download code

Download code from (clone or zip): https://github.com/AlphaTrainer/alpha_bci_experiment/tree/final_handin

C.1.2 Prerequests

Octave 3.6.4 and up [1].

C.1.3 Load and run

```
$ cd alpha_bci_experiment
# load signal processing package of Octave
octave> libs_install_and_load
...
  class opts;
  ^~~~~
  struct
1 warning generated.
```

[1] http://www.gnu.org/software/octave/download.html

Just ignore the warning for now we have loaded the signal package successfully and are ready to run the analysis - do:

```
octave> run_analysis
number_of_files =  24
mindwave/1-20130814-pelle-2-open.mat
OPEN EYES FIRST
Discarded bins:0
Discarded bins:0
alpha_peak_start_sample =  19
alpha_peak_end_sample =  23
...
...
```

Coffee break while the whole data of the experiment gets processed and analyzed - at the end we get the result that is added to Table 3.3.

Appendix D

MindWave forum posts

Support | Public Discussions | Knowledge Base | Go to NeuroSky - Home Page → Log in or Create a profile

NeuroSky
Brain Wave Sensors for Every Body

Search discussions SEARCH

Home → SDK/Development Question →

Outliers in EEG powers on Android

Martin Poulsen
May 29, 2013 @ 07:58 AM

Hi,

I am experiencing giant outliers for all frequency bands on Android (TGDevice.MSG_EEG_POWER). Attached graphs show 1 minute of recorded lowAlpha (same data in both graphs with different "zoom" on y-axis).

Any ideas to what I might be doing wrong or how I solve this?

sdk_lalpha_power.pdf
23.9 KB

sdk_lalpha_power__y_limited_to_30000_.pdf

5 *people watching.*

~~New Issue~~ ✓

~~Conversation Started~~ ✓

The discussion is closed

No more actions from NeuroSky - Home Page or the discussion starter are required.

Re-open the discussion

Permissions

*This discussion is **private**. Only you and NeuroSky - Home Page support staff can see and reply to it.*

Comments Feed

NeuroSky
Brain Wave Sensors for Every Body

Search discussions SEARCH

Outliers in EEG powers on Android

Martin Poulsen
May 29, 2013 @ 07:58 AM

Hi,

I am experiencing giant outliers for all frequency bands on Android (TGDevice.MSG_EEG_POWER). Attached graphs show 1 minute of recorded lowAlpha (same data in both graphs with different "zoom" on y-axis).

Any ideas to what I might be doing wrong or how I solve this?

sdk_lalpha_power.pdf
23.9 KB
sdk_lalpha_power__y_limited_to_30000_.pdf

5 *people watching.*

~~New Issue~~ ✓
~~Conversation Started~~ ✓

The discussion is closed
No more actions from NeuroSky - Home Page or the discussion starter are required.

Re-open the discussion

Permissions
This discussion is ***private****. Only you and NeuroSky - Home Page support staff can see and reply to it.*

Comments Feed

Support | Public Discussions | Knowledge Base | Go to NeuroSky - Home Page → Log in or Create a profile

NeuroSky
Brain Wave Sensors for Every Body

Search discussions SEARCH

Home → SDK/Development Question →

Outliers in EEG powers on Android

Martin Poulsen
May 29, 2013 @ 07:58 AM

Hi,

I am experiencing giant outliers for all frequency bands on Android (TGDevice.MSG_EEG_POWER). Attached graphs show 1 minute of recorded lowAlpha (same data in both graphs with different "zoom" on y-axis).

Any ideas to what I might be doing wrong or how I solve this?

sdk_lalpha_power.pdf
23.9 KB
sdk_lalpha_power__y_limited_to_30000_.pdf

5 people watching.

New Issue ✓
Conversation Started ✓

The discussion is closed
No more actions from NeuroSky - Home Page or the discussion starter are required.

Re-open the discussion

Permissions
This discussion is private. Only you and NeuroSky - Home Page support staff can see and reply to it.

Comments Feed

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

interesting frequencies they are just ignored.

I hope that helps a little.
-David

Ashley Schultz **closed** this discussion on Jun 20, 2013 @ 06:27 PM.

Re-open and reply to this **closed** discussion or start a new discussion →

Appendix E

Design - personas

E.1 4 personas

E.1.1 Morten

Age: 52 years.

Marital status: Married, 3 children.

Profession: Project manager.

Phone: Sony Xperia Z (always the newest generation of Android phones, switches approximately every 9 months).

Has had stress related problems and has picked up mindfulness meditation 5 years ago.

Eats healthy and generally takes it serious to take care of his health

Morten like to listen to *Kraut-rock* and *New Age* music from his young days especially *Brian Eno*. When he has the time he read a good novel last one red was *The Savage Detectives* by *Roberto Bolaño*.

Morten recently got a grandson and a granddaughter and dreams about remaining physical and mental fit for many more years.

E.1.2 Niels

Age: 28 years.

Marital status: Single, had a boyfriend for some time ago but likes to be independent.

Profession: Business IT Consultant, newly graduated from CBS

Phone: Has an IPhone 5, looking for a replacement and has been wanting to switch since the HTC One. He will likely get the One soon but is locked into Apples ecosystem

Niels is a techie, has been over-clocking his computer since middle school and has a general interest in how technology can support all aspects of life.

Has used self-tracking to track sleep, exercise and eating.

Has career ambitions but has noticed the tough consultant environment and seriously consider how he can be able to deal with this stress over time.

He dreams about traveling especially Asia attracts him. Some years ago he backpacked through Thailand, Cambodia and Laos.

E.1.3 Olivia

Age: 42.

Marital status: divorced, 2 children.

Profession: HR manager at a medium sized private company.

Phone: Has a Motorola MotoX.

Struggles daily with navigation between work time and spare time.

Practices Yoga, gives her a lot of stress relief and energy.

She uses daily around 1 hour to commute between work and home.

On a silent evening she like to read her all time favorite is *To the Lighthouse* by *Virginia Woolf* perhaps combined with a music in the background which could be some classical music by *Arvo Pärt*.

Olivia want to live a healthy life and be happy - which she see as a skill and is willing to put in some effort in order to achieve.

E.1.4 Peter

Age: 35, in a relationship, no children (yet)

Profession: IT Professional, system architect Phone: Has a private IPhone and a work HTC Android phone

Works a lot, loves his work but has lately wanted to balance his work life with his private life in considering to start a family with children etc.

Is very interested in enhancing his performance by means of technique and technology - has for example been polyphase sleeping to get more wake hours a day.

If his girlfriend forces him to relax with a book on their holidays it could probably be some novels by *Haruki Murakami*. Together the like to listen to some pop music and when he listen lonely he like to pick up some electronic music perhaps by *Alva Noto* flavored by some piano music by *Ryuichi Sakamoto*.

Exercises with kettle weights after thorough cost benefit research of different exercise methods.

Uses Sleep Cycle to analyze his sleeping patterns.

Peter dreams about traveling more and perhaps one day to run New York Marathon.

Appendix F

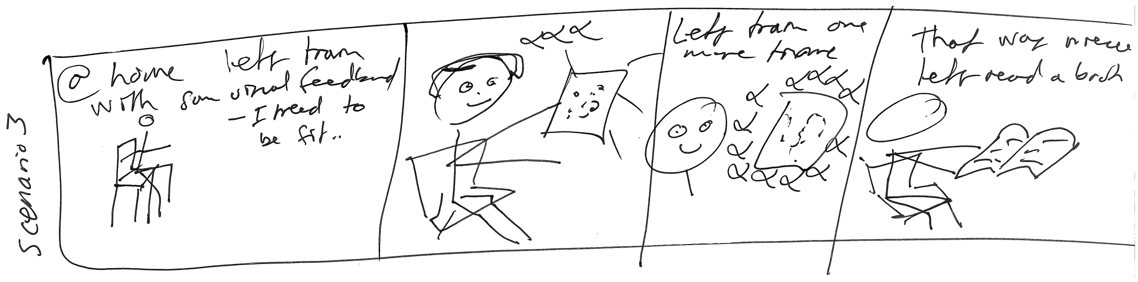
Design - storyboards

The storyboards frame 3 scenarios of AlphaTrainer system: (*i*) home; (*ii*) work; and (*iii*) in between (commutting).

Scenario 1: Peter performing alpha training at work. Frame 1: Peter has a busy day with a lot of deadlines waiting around the corner. Frame 2: Before lunch, he finds a quiet spot. Frame 3: He performs 15 minutes of alpha training. Frame 4: He finds himself relaxed and gets back to work.

Scenario 2: Peter performing alpha training on the train home. Frame 1: Peter waits for the train. Frame 2: He finds a seat. Frame 3: To best abstract from the surroundings, he chooses to train with

closed eyes. Frame 4: He performs 5 minutes of alpha training and raises his alpha level substantially which gives him a score of 120%.



Scenario 3: Peter performing alpha training at work. Frame 1: Peter sits at home and decides to train some alpha with his favorite visual feedback. Frame 2: He trains and gets more relaxed. Frame 3: After a 5 minute training session has finished, he decides to train again. Frame 4: He feels relaxed and moves on to reading a book.

Appendix G

Software

G.1 AlphaTrainer signal processing library

Download code from (clone or zip): https://github.com/AlphaTrainer/opencvbrain/tree/finalhandin

G.1.1 Prerequests

CMake [1] and OpenCV [2].

G.1.2 Load and run

```
$ cd opencvbrain/opencvbrain/build
$ cmake ..
$ make
# run the main_brain.cpp of library
$./opencvbrain
-- The C compiler identification is Clang 4.2.0
...
[100%] Built target opencvbrain
--------------------------------
getBrainProcessed(...): 6.22933e-17
```

[1] http://www.cmake.org.
[2] http://opencv.org

```
expected: 6.22933e-17
--------------------------------
getAlphaPeak(): 10
expected: 10
--------------------------------
getMinMax(): 0 - 1.03917e+13
expected: 0 - 1.03917e+13
--------------------------------
```

G.2 AlphaTrainer Android App

Download code from (clone or zip):

https://github.com/AlphaTrainer/AlphaTrainerAndroid/tree/finalhandin

G.2.1 Quick

Simply install the AlphaTrainerApp-debug.apk

```
$ wget https://github.com/AlphaTrainer/AlphaTrainerAndroid/raw/finalhandin/AlphaTrainerApp-debug.apk
$ adb install <path-to-apk>/AlphaTrainerApp-debug.apk
```

G.2.2 Build from scratch

** Prerequests **

We assume all Android tools (ADT) like adb, NDK [3], etc are setup.

OpenCV Android SDK 2.4.6 [4] and have it set in the build path:

```
$ ls $OPENCV_ANDROID_SDK
etc java    native
```

** Use the shell scripts **

[3] http://developer.android.com/tools/sdk/ndk/index.html
[4] http://opencv.org/downloads.html

```
$ cd <some dir that have a check out of opencvbrain>
$ ls
AlphaTrainerAndroid opencvbrain
...
$ cd AlphaTrainerAndroid
# step 0: build native lib stand alone
$ build_0_opencvbrain.sh
...
# step 1: build native lib into app with ndk
$ build_1_ndk_brainapp.sh
# step 2 build and install android app:
$ build_2_android_brainapp.sh
```

G.3 AlphaTrainerService

Runs at MongoLab [5] and Heroku [6]: http://alpha-trainer.herokuapp.com/.

G.3.1 Mongolab cloud storage

Using the the version 2 beta we can for example query all feedback trainings and ignore the *alpha_levels*:

```
GET https://data-api.mongolab.com/v2/apis/dk5jpmcf2g1bg/collections/trainings
/documents?fields=%7B%22alpha_levels%22%3A0%7D&query=%7B%22type%22%3A%22Feedback%22%7D</code>
```

Last part not url encoded:

```
/documents?q={"type":"Feedback"}&fields={"alpha_levels":0}</code>
```

Reference the full API at http://alpha-trainer.herokuapp.com/.

G.3.2 AlphaTrainerService client

Download code from (clone or zip):

[5] https://mongolab.com
[6] https://www.heroku.com

https://github.com/AlphaTrainer/AlphaTrainerService/tree/finalhandin

Download the *Typesafe Activator* a tool on top of the *Play Framework* [7]. Ensure to have *Activator* in your build path:

```
$ which activator
<path to>/activator-1.0.0/activator
```

OK then start it:

```
$ cd <path to>/AlphaTrainerService
$ ./activator run
# listen to file changes use
$ ./activator ~run
```

And then open the client in a browser.

[7] http://www.playframework.com/download

Appendix H

User evaluation

H.1 Hand out

Schema handed out to the participants.

Participant

Gender: F | M Age: ---------------

Participant: 1 | 2 | 3 | 4 (filled in by the evaluators)

Day <number>

Choose feedback through the AlphaTrainer application: Box | Vibration | Circles | Tone | Bells (marked by the evaluators)

Training A

Where did you train? -------------------------------- How may times did you train in a row ? ---------------------------

Please select the number below that best represents how you feel about your recent alpha training for each statement.

	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree
I feel more relaxed after training than before (mark an "x" in this line)	1	2	3	4	5
The feedback was pleasant (mark an "x" in this line)	1	2	3	4	5
The feedback was precise (mark an "x" in this line)	1	2	3	4	5
I was disturbed by my surroundings (mark an "x" in this line)	1	2	3	4	5
My surroundings were disturbed by me (mark an "x" in this line)	1	2	3	4	5
It was easy to find time for training (mark an "x" in this line)	1	2	3	4	5
It was comfortable to wear the headset (mark an "x" in this line)	1	2	3	4	5

Training B

Repeated whats in *Training A*.

Helper

1. Turn on headset - a blue lamp should be blinking.

2. Bring on headset - ensure to attach clip to the ear-lob and that the sensor is attached to the skin.

3. Start the AlphaTrainer Android application.

4. Take of headset and turn off the power.

5. Remember to charge the Android phone nightly.

H.2 Ordered baseline recordings

The data is to order by each participants baselines according to alpha level.

Person 1

```
Baseline|feedback_box|0.0748699083924294

Feedback|feedback_box|0.0795906856656075

Feedback|feedback_box|0.105853267014027


Baseline|feedback_vibrate|0.0955281630158424

Feedback|feedback_vibrate|0.0991984605789185

Feedback|feedback_vibrate|0.132098972797394

Feedback|feedback_vibrate|0.108336806297302

Feedback|feedback_vibrate|0.119316846132278


Baseline|feedback_audio_clips|0.0966365113854408
```

Feedback|feedback_audio_clips|0.11055614054203

Feedback|feedback_audio_clips|0.111081063747406

Feedback|feedback_audio_clips|0.120809569954872

Feedback|feedback_audio_clips|0.0853465273976326

Baseline|feedback_collision|0.102063372731209

Feedback|feedback_collision|0.118450820446014

Feedback|feedback_collision|0.116510726511478

Feedback|feedback_collision|0.10384389013052

Baseline|feedback_audio_synth|0.0900116935372353

Feedback|feedback_audio_synth|0.099056139588356

Feedback|feedback_audio_synth|0.134804680943489

Feedback|feedback_audio_synth|0.0988185256719589

Person 2

Baseline|feedback_vibrate|0.0987723246216774

Feedback|feedback_vibrate|0.100397661328316

Feedback|feedback_vibrate|0.119221583008766

Feedback|feedback_vibrate|0.101614952087402

Baseline|feedback_collision|0.119308114051819

Feedback|feedback_collision|0.0917378216981888

Feedback|feedback_collision|0.125906825065613

Baseline|feedback_box|0.121146105229855

Feedback|feedback_box|0.155871793627739

Feedback|feedback_box|0.144879370927811

??? - BASELINE NEEDED

Feedback|feedback_audio_synth|0.101205192506313

Feedback|feedback_audio_synth|0.120678253471851

Baseline|feedback_audio_synth|0.105333186686039

Feedback|feedback_audio_synth|0.1014259532094

??? - BASELINE NEEDED

Feedback|feedback_audio_clips|0.105146333575249

Feedback|feedback_audio_clips|0.0893903225660324

Person 3

Baseline|feedback_collision|0.105134434998035

Feedback|feedback_collision|0.10310272872448

Feedback|feedback_collision|0.0958361625671387

Feedback|feedback_collision|0.0887904837727547

Feedback|feedback_collision|0.0813872665166855

Feedback|feedback_collision|0.0867800489068031

Feedback|feedback_collision|0.0706308037042618

Baseline|feedback_audio_synth|0.130459442734718

Feedback|feedback_audio_synth|0.129751697182655

Feedback|feedback_audio_synth|0.105608582496643

Feedback|feedback_audio_synth|0.0996931046247482

Feedback|feedback_audio_synth|0.11997053027153

Feedback|feedback_audio_synth|0.0965548679232597

Baseline|feedback_vibrate|0.130258351564407

Feedback|feedback_vibrate|0.11822721362114

Feedback|feedback_vibrate|0.140906989574432

Feedback|feedback_vibrate|0.147401303052902

Feedback|feedback_vibrate|0.123775370419025

Baseline|feedback_box|0.118772350251675

Feedback|feedback_box|0.0973722115159035

Feedback|feedback_box|0.0930206328630447

Feedback|feedback_box|0.0985453799366951

Baseline|feedback_audio_clips|0.0821172967553139

Feedback|feedback_audio_clips|0.0767084732651711

Feedback|feedback_audio_clips|0.106305181980133

P4 (Morten)

Baseline|feedback_audio_synth|0.184045940637589

Feedback|feedback_audio_synth|0.161301583051682

Feedback|feedback_audio_synth|0.157192721962929

Feedback|feedback_audio_synth|0.156435072422028

Baseline|feedback_audio_clips|0.207016110420227

Feedback|feedback_audio_clips|0.127788901329041

Feedback|feedback_audio_clips|0.136331424117088

Baseline|feedback_collision|0.0952290520071983

Feedback|feedback_collision|0.0937773808836937

Feedback|feedback_collision|0.111754521727562

Feedback|feedback_collision|0.0840806216001511

Baseline|feedback_box|0.112990573048592

Feedback|feedback_box|0.0992246344685555

Feedback|feedback_box|0.0916369184851646

Feedback|feedback_box|0.0899822413921356

Feedback|feedback_box|0.0804533064365387

Baseline|feedback_vibrate|0.169038251042366

Feedback|feedback_vibrate|0.168128445744514

Feedback|feedback_vibrate|0.138531193137169

Feedback|feedback_vibrate|0.164972335100174

Pilot 1

Baseline|feedback_box|0.0903623178601265

Feedback|feedback_box|0.126605808734894

Feedback|feedback_box|0.0992843806743622

Baseline|feedback_collision|0.0866829007863998

Feedback|feedback_collision|0.0840329304337502

Baseline|feedback_vibrate|0.118690125644207

Feedback|feedback_vibrate|0.0947581827640533

Baseline|feedback_audio_synth|0.111464120447636

Feedback|feedback_audio_synth|0.114507429301739

Feedback|feedback_audio_synth|0.117919556796551

Feedback|feedback_audio_synth|0.105000719428062

Baseline|feedback_audio_clips|0.0992960855364799

Feedback|feedback_audio_clips|0.107017315924168

Feedback|feedback_audio_clips|0.107197441160679

Feedback|feedback_audio_clips|0.114198744297028

Pilot 2

Baseline|feedback_audio_clips|0.0793051272630692

Feedback|feedback_audio_clips|0.0856634080410004

Feedback|feedback_audio_clips|0.0882096067070961

Feedback|feedback_audio_clips|0.0881557464599609

Baseline|feedback_audio_synth|0.128948673605919

Feedback|feedback_audio_synth|0.130064263939857

Feedback|feedback_audio_synth|0.128819465637207

Feedback|feedback_audio_synth|0.135159030556679

Feedback|feedback_audio_synth|0.128667920827866

Feedback|feedback_audio_synth|0.133776649832726

??? - BASELINE NEEDED

Feedback|feedback_vibrate|0.106828294694424

Feedback|feedback_vibrate|0.129459515213966

Feedback|feedback_vibrate|0.126018390059471

Feedback|feedback_vibrate|0.117967203259468

Feedback|feedback_vibrate|0.120155863463879

Feedback|feedback_vibrate|0.12348672747612

Baseline|feedback_collision|0.0796105414628983

Feedback|feedback_collision|0.101417362689972

Feedback|feedback_collision|0.107857331633568

Feedback|feedback_collision|0.105241321027279

Feedback|feedback_collision|0.0695364400744438

Feedback|feedback_collision|0.0933116674423218

Feedback|feedback_collision|0.108181580901146

Baseline|feedback_box|0.0643191933631897

Feedback|feedback_box|0.0712279677391052

Feedback|feedback_box|0.069434218108654

Feedback|feedback_box|0.105725206434727

Feedback|feedback_box|0.0832352340221405

Feedback|feedback_box|0.0682065561413765

Feedback|feedback_box|0.0753804966807365

Bibliography

[1] AAP Evidence-Based Child and Adolescent Psychosocial Interventions. Technical report, The American Academy of Pediatrics mental health, 2013.

[2] H. Abelson, G. J. Sussman, and J. Sussman. *Structure and interpretation of computer programs.* MIT Press ; McGraw-Hill, Cambridge, Mass.; New York, 1996.

[3] G. D. Abowd, E. D. Mynatt, and T. Rodden. The human experience [of ubiquitous computing]. *Pervasive Computing, IEEE*, 1(1):48–57, 2002.

[4] L. I. Aftanas and S. A. Golosheikin. Changes in Cortical Activity in Altered States of Consciousness: The Study of Meditation by High-Resolution EEG. *Human physiology*, 29(2):143–151, Mar. 2003.

[5] E. Angelakis, S. Stathopoulou, J. L. Frymiare, D. L. Green, J. F. Lubar, and J. Kounios. EEG Neurofeedback: A Brief Overview and an Example of Peak Alpha Frequency Training for Cognitive Enhancement in the Elderly. *The Clinical Neuropsychologist*, 21(1):110–129, Jan. 2007.

[6] M. Arns, d. S. Ridder, U. Strehl, M. Breteler, and A. Coenen. Efficacy of Neurofeedback Treatment in ADHD: the Effects on Inattention, Impulsivity and Hyperactivity: a Meta-Analysis. *Clinical EEG and Neuroscience*, 40(3):180–189, 2009.

[7] D. L. Baggio. *Mastering OpenCV with practical computer vision projects: step-by-step tutorials to solve common real-world computer vision problems for desktop or mobile, from augmented reality and number plate recognition to face recognition and 3D head tracking.* Packt Pub., Birmington, 2012.

[8] J. Bardram and A. Friday. Ubiquitous Computing Systems. In *Ubiquitous computing fundamentals*, pages 38–88. Chapman & Hall/CRC Press, Boca Ragon, 2010.

[9] O. Bazanova. Comments for Current Interpretation EEG Alpha Activity: A Review and Analysis. *Journal of Behavioral and Brain Science*, 02(02):239–248, 2012.

[10] M. Beaudouin-Lafon. Designing interaction, not interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, page 15–22, 2004.

[11] M. Y. Bekkedal, J. Rossi, and J. Panksepp. Human brain EEG indices of emotions: Delineating responses to affective vocalizations by measuring frontal theta event-related synchronization. *Neuroscience & Biobehavioral Reviews*, 35(9):1959–1970, Oct. 2011.

[12] D. Benyon. *Designing interactive systems: people, activities, contexts, technologies.* Addison-Wesley, Harlow, England ; New York, 2005.

[13] D. Benyon. *Designing interactive systems: a comprehensive guide to HCI and interaction design.* Addison Wesley, Harlow, England; New York, 2010.

[14] J. Bloch. *Effective Java.* The Java series. Addison-Wesley, Upper Saddle River, NJ, 2nd ed edition, 2008.

[15] T. Brandmeyer and A. Delorme. Meditation and Neurofeedback. *Frontiers in Psychology*, 4(688), 2013.

[16] A. Campbell, T. Choudhury, S. Hu, H. Lu, M. K. Mukerjee, M. Rabbi, and R. D. Raizada. NeuroPhone: brain-mobile phone interface using a wireless EEG headset. page 3. ACM Press, 2010.

[17] G. F. Coulouris, editor. *Distributed systems: concepts and design.* Addison-Wesley, Boston, 5th ed edition, 2012.

[18] D. Coyle, J. Garcia, A. R. Satti, and T. M. McGinnity. EEG-based continuous control of a game using a 3 channel motor imagery BCI: BCI game. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2011 IEEE Symposium on*, page 1–7, 2011.

[19] K. Crowley, A. Sliney, I. Pitt, and D. Murphy. Evaluating a Brain-Computer Interface to Categorise Human Emotional Response. *2010 10th IEEE International Conference on Advanced Learning Technologies*, pages 276–278, July 2010.

[20] A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, Mar. 2004.

[21] L. Duan, X. Wang, Z. Yang, H. Zhou, C. Wu, Q. Zhang, and J. Miao. An Emotional Face Evoked EEG Signal Recognition Method Based on Optimal EEG Feature and Electrodes Selection. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, B.-L. Lu, L. Zhang, and J. Kwok, editors, *Neural Information Processing*, volume 7062, pages 296–305. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[22] T. Egner and J. H. Gruzelier. Ecological validity of neurofeedback: modulation of slow wave EEG enhances musical performance. *Neuroreport*, 14(9):1221–1224, 2003.

[23] E. T. Esfahani and V. Sundararajan. USING BRAIN–COMPUTER INTERFACES TO DETECT HUMAN SATISFACTION IN HUMAN–ROBOT INTERACTION. *International Journal of Humanoid Robotics*, 08(01):87–101, Mar. 2011.

[24] S. Finnigan and I. H. Robertson. Resting EEG theta power correlates with cognitive performance in healthy older adults: Resting theta EEG correlates with cognitive aging. *Psychophysiology*, 48(8):1083–1087, Aug. 2011.

[25] A. Fox and D. Patterson. *Engineering software as a service: an agile approach using cloud computing.* Strawberry Canyon LLC, San Francisco, CA, 2013.

[26] J. Gomez-Gil, I. San-Jose-Gonzalez, L. F. Nicolas-Alonso, and S. Alonso-Garcia. Steering a Tractor by Means of an EMG-Based Human-Machine Interface. *Sensors*, 11(12):7110–7126, July 2011.

[27] D. C. Hammond. What Is Neurofeedback? *Journal of Neurotherapy*, 10(4):25–36, Mar. 2007.

[28] P. Hilton, E. Bakker, and F. Canedo. *Play for Scala: covers Play 2.* Manning Publications, 2013.

[29] S. I. Hjelm. Research+ design: the making of Brainball. *Interactions*, 10(1):26–34, 2003.

[30] S. ichi Ito, Y. Mitsukura, K. Sato, S. Fujisawa, and M. Fukumi. Association between ego scores and individual characteristics in EEG analysis: Basic study on individual brain activity. In C. A. Avizzano and E. Ruffaldi, editors, *RO-MAN*, pages 210–215. IEEE, 2010.

[31] Iso. *ISO 9241-210:2010 - Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems.* 1 edition, 2010.

[32] C. Jeremy. *Pretotyping@Work*. PretotypeLabs.com, 2012.

[33] R. Khosrowabadi, C. Quek, K. K. Ang, S. W. Tung, and M. Heijnen. A Brain-Computer Interface for classifying EEG correlates of chronic mental stress. In *IJCNN*, pages 757–762. IEEE, 2011.

[34] S. R. Klemmer, B. Hartmann, and L. Takayama. How bodies matter: five themes for interaction design. In *Proceedings of the 6th conference on Designing Interactive systems*, page 140–149, 2006.

[35] J. Kollmann, H. Sharp, and A. Blandford. The Importance of Identity and Vision to User Experience Designers on Agile Projects. pages 11–18. IEEE, Aug. 2009.

[36] N. Kuramoto, S. ichi Ito, K. Sato, and S. Fujisawa. Trigger pattern detection method for assisting in ambulation rehabilitation based on EEG analysis. In *RO-MAN*, pages 646–652. IEEE, 2012.

[37] E. A. Larsen and A. I. Wang. *Classification of EEG Signals in a Brain- Computer Interface System*. PhD thesis, Norwegian University of Science and Technology - Department of Computer and Information Science, 2011.

[38] J. Lazar. *Research methods in human-computer interaction*. Wiley, Chichester, West Sussex, U.K, 2010.

[39] F. Liu. *Android native development kit cookbook*. Packt Publishing, Birmingham, 2013.

[40] N. Lofthouse, L. E. Arnold, S. Hersch, E. Hurt, and R. DeBeus. A review of neurofeedback treatment for pediatric ADHD. *Journal of attention disorders*, 16(5):351–72, July 2012.

[41] W. E. Mackay and A.-L. Fayard. HCI, natural science and design: a framework for triangulation across disciplines. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*, page 223–234, 1997.

[42] I. S. MacKenzie. *Human-computer interaction: an empirical research perspective*. Elsevier, Amsterdam, 2013.

[43] K. Majumdar. Human scalp EEG processing: Various soft computing approaches. *Applied Soft Computing*, 11(8):4433–4447, Dec. 2011.

[44] M. Marchesi and B. Riccò. BRAVO: a brain virtual operator for education exploiting brain-computer interfaces. page 3091. ACM Press, 2013.

[45] R. Matthews, P. J. Turner, N. J. McDonald, K. Ermolaev, T. Manus, R. A. Shelby, and M. Steindorf. Real time workload classification from an ambulatory wireless EEG system using hybrid EEG electrodes. *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2008:5871–5875, 2008. PMID: 19164053.

[46] D. J. McFarland and J. R. Wolpaw. Brain-computer interfaces for communication and control. *Communications of the ACM*, 54(5):60, May 2011.

[47] Q. Meng, W. Zhou, Y. Chen, and J. Zhou. Feature analysis of epileptic EEG using nonlinear prediction method. *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2010:3998–4001, 2010. PMID: 21097278.

[48] V. Mirgorod. *Backbone. js Cookbook*. Packt Publishing Ltd, 2013.

[49] A. Nayak. *Instant MongoDB*. Packt Publishing, 2013.

[50] NeuroSky. Brain Wave Signal (EEG) of NeuroSky, Inc. Technical report, NeuroSky, Inc., 2009.

[51] NeuroSky. ThinkGear Development Guide for Android. Technical report, NeuroSky, Inc., 2012.

[52] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen. FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience*, 2011:1–9, 2011.

[53] C. K. Petersen and Klonovs. *Development of a Mobile EEG-Based Feature Extraction and Classification System for Biometric Authentication.* PhD thesis, Aalborg University Copenhagen, 2012.

[54] M. Rangaswamy, B. Porjesz, D. B. Chorlian, K. Wang, K. A. Jones, L. O. Bauer, J. Rohrbaugh, S. J. O'Connor, S. Kuperman, and T. Reich. Beta power in the EEG of alcoholics. *Biological psychiatry*, 52(8):831–842, 2002.

[55] S. Ratabouil. *Android NDK discover the native side of Android and inject the power of C/C++ in your applications: beginner's guide.* Packt Pub., Birmingham, U.K., 2012.

[56] G. Rebolledo-Mendez, I. Dunwell, E. A. Martínez-Mirón, M. D. Vargas-Cerdán, S. Freitas, F. Liarokapis, and A. R. García-Gaona. Assessing NeuroSky's Usability to Detect Attention Levels in an Assessment Exercise. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and J. A. Jacko, editors, *Human-Computer Interaction. New Trends*, volume 5610, pages 149–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[57] D. Schmalstieg, A. Bornik, G. Müller-Putz, and G. Pfurtscheller. Gaze-directed ubiquitous interaction using a Brain-Computer Interface. pages 1–5. ACM Press, 2010.

[58] C. Stam. Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. *Clinical Neurophysiology*, 116(10):2266–2301, Oct. 2005.

[59] A. Stopczynski, C. Stahlhut, M. K. Petersen, J. E. Larsen, C. F. Jensen, M. G. Ivanova, T. S. Andersen, and L. K. Hansen. Smartphones as pocketable labs: Visions for mobile brain imaging and neurofeedback. *International Journal of Psychophysiology*, Aug. 2013.

[60] T2. BioZenUsersMan_1_7_0.pdf. Technical report, Telehealth and Technology (T2) under the US Department of Defense, 2013.

[61] S. Tong and N. V. Thakor. Quantitative EEG analysis methods and clinical applications. In *Quantitative EEG analysis methods and clinical applications*, Artech House series engineering in medicine & biology, pages 1–107. Artech House, Boston, 2009.

[62] E. R. Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, 1990.

[63] R. A. Virzi. Refining the test phase of usability evaluation: how many subjects is enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 34(4):457–468, 1992.

[64] J. Wei. *Android database programming exploit the power of data-centric and data-driven Android applications with this practical tutorial.* Packt Pub. Ltd., Birmingham, 2012.

[65] M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):66–75, Jan. 1991.

[66] M. Weiser. The world is not a desktop. *interactions*, 1(1):7–8, Jan. 1994.

[67] M. Weiser. Some computer science issues in ubiquitous computing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):12, 1999.

[68] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967.

[69] N. E. White and L. Richards. Alpha–theta neurotherapy and the neurobehavioral treatment of addictions, mood disorders and trauma. In *Introduction to quantitative EEG and neurofeedback advanced theory and applications*. Academic Press/Elsevier, Amsterdam [etc.], 2009.

[70] F. P. Wright. *EMOCHAT EMOTIONAL INSTANT MESSAGING WITH THE EPOC HEADSET*. PhD thesis, University of Maryland, Baltimore County, 2010.

[71] Y. Yasui. A Brainwave Signal Measurement and Data Processing Technique for Daily Life Applications. *Journal of PHYSIOLOGICAL ANTHROPOLOGY*, 28(3):145–150, 2009.

[72] Y. Yu, D. He, W. Hua, S. Li, Y. Qi, Y. Wang, and G. Pan. FlyingBuddy2: A Brain-controlled Assistant for the Handicapped. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 669–670, New York, NY, USA, 2012. ACM.

[73] B. Zoefel, R. J. Huster, and C. S. Herrmann. Neurofeedback training of the upper alpha frequency band in EEG improves cognitive performance. *NeuroImage*, 54(2):1427–1431, Jan. 2011.