3D simulation of cognitive workload to determine attention for any given user through a brain computer interface device

New Mexico

Supercomputing Challenge

Final Report

April, 2015

Team 40

Las Cruces YWiC

Team Members:

Ahmed Muhyi

Hoon Jeong Lee

Table of Contents

1.0 Abstract

This project was to find the maximum attention span and workload for any user. We wanted to find a method for users to determine their average concentration levels by using a simple test involving math, the user's current brain wave output, and time. The goal was to produce reasonably accurate data that the user may use and manipulate in any way. This was possible via a Python script and a Neurosky "Mindwave" headset. The program observes the users focus and amount of time taken for each problem. Focus is measured through what brainwave the user is actively emitting.We are focusing on Beta waves because Beta waves are linked with concentration. We found that a user can find his or her circadian rhythm (awake cycles, similar to sleep cycles) by using the program; however, that is just one of the numerous applications of this program.

2.0 Introduction

The human brain has many capabilities and abilities. The brain contains approximately 100 billion neurons. They communicate through synapses. One of the abilities of the brain is to adapt and grow by reorganizing neural pathways and even creating new ones, which is called neuroplasticity. Research shows both genetics and environmental factors have an effect on the development and physical structure of the brain. Different areas of the brain are responsible for different cognitive functions. These frontal and parietal areas are known to be involved in the control of attention and visuospatial working memory. Attention is "the human ability to concentrate on a certain objects and allocate processing resources accordingly".

A non-invasive NeuroSky's Brain Computer Interface was used and embedded with the applications to measure a user's attention.

2.1 NeuroSky's Brain Computer Interface (BCI)

NeuroSky's BCI is a non-invasive, dry, bio sensor used to read electrical activity in the brain to determine states of attention. It is able to measure and record raw electroencephalogram (EEG) brainwaves (Alpha, Beta, Theta, Gamma and Delta) by using three dry electrodes located on the left ear and one on the forehead. Using NeuroSky's original algorithm, attention is calculated based on EEG brainwave. Every second the headset computes the attention measures based on the user's brain activity. The output is a number from 0-100. The Neurosky BCI has been evaluated by Neurosky's company and several researchers to measure usability and accuracy. The Neurosky uses EEG brain waves that show in five different waves based on concentration, activity, restfulness, drowsiness, and sleep waves.

2.2 EEG Waves

Electroencephalogaphy (EEG) originated during 1924 by Hans Berger. There are two ways EEG is recorded, invasive and non-invasive to record the EEG waves. The method that was used, was non-invasive due to invasive being expensive and involving implants and surgery in the brain. Electrodes are placed around the brain for complete immersion but the headset from Neurosky uses only one electrode sensor that is placed on the forehead and one on the ear. Current generates in the brain when neurons communicate which is called action potential. The electrodes record when the neurons "discharge" by the opening and closing of the Sodium and Potassium ions. Diagram 1 shows a comparison between the five different EEG waves. Because the headset was a commercial product, the algorithm to calculate attention was not given to us so attention has to be given by the two EEG waves, gamma and beta.

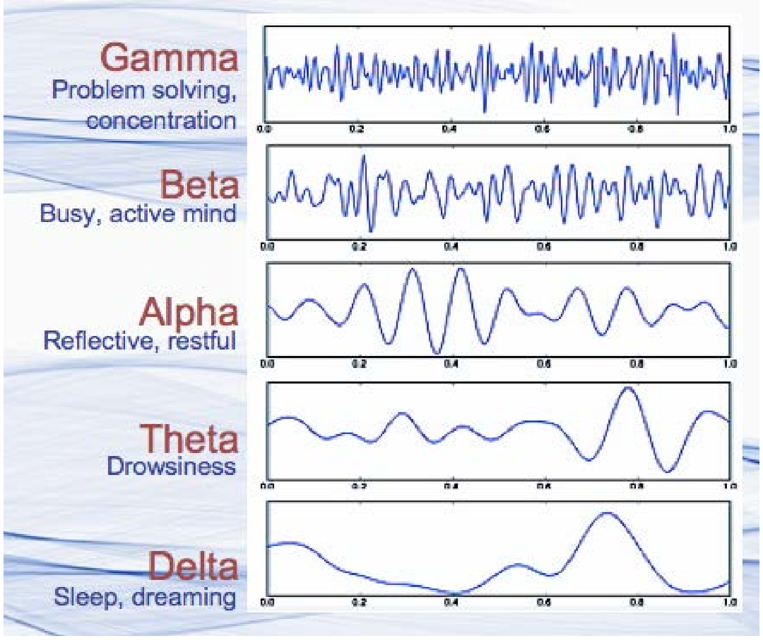


Diagram 1: A comparison between the five different EEG waves.

Gamma waves have a frequency of 31 Hz and up. The waves are short and dense and associated with concentration and problem solving. Gamma waves are focused with Beta waves, as they are both associated with attention. **Beta** waves have a frequency of 12 to 30 Hz. These waves are small and fast and associated on how busy or active the mind is. **Alpha** waves have a frequency of 7.5 to 12 Hz. These waves are slower and associated with relaxation and restfulness. **Theta**

waves have a frequency of 3.5 to 7.5 Hz. These waves are wider and amplitude is not as high as alpha. These waves are associated with inefficiency, daydreaming and drowsiness. **Delta** waves have a frequency of 0.5 to 3.5 Hz and are the slowest EEG waves. These waves are associated with sleeping.

3.0 Problem

The main goals of this project is to develop a systems to help measure the individual cognitive workload using a single channel brain computer interface. The other goal of this project is to create a

model and 2D and 3D simulation of the brain wave activity using the open source Openvibe platform.

4.0 Mathematical model

The mathematical formula for finding how the attention span was derived proved to have its challenges. The chip located in the headset automatically transfers the unfiltered, raw data from the sensors and filters this data. The filtration process uses an algorithm called the Cooley-Tukey Fast Fourier Transform (FFT) algorithm by J.W. Cooley and John Tukey in 1965 (equation 1).

The FFT algorithm is based on a *Divide and Conquer* algorithm. This means the workload of the program is divided to make the calculations more efficient. The algorithm is based on the Discrete Fourier Transform (DFT) (equation 2). By applying the algorithm, the EEG signal makes it possible to separate the different EEG waves.

DFT uses a computational complexity of O[N^2] for calculations while the more efficient FFT uses a computational complexity of O[NlogN]. This means the FFT algorithm is more efficient when calculating DFT of x(n). (O means order).

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\left(\frac{2\pi}{N}\right)nk} (k = 0,1,\ldots,N-1)$$

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} (k = 0,1,\ldots,N-1)$$

Equation 1: Formula for the FFT algorithm using O[NlogN]

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2k\pi \frac{n}{N}} \qquad k = 0, \ldots, N-1$$

Equation 2: Formula for the DFT algorithm using O[N^2] (From larsen)

5.0 Computational Model

The entire program works with a Python script and computer communicating with a brain computer interface headset from Neurosky. A test is given to the user and in real time, the headphone sends data pack to the program with the current user’s brainwave level. At the end of the test, the program takes in the data and divides it by the amount of time and questions taken to give you the average level of hertz that specific user was operating under. The user can then use this information to determine the level of concentration they were at. There are two tests. One test is in addition, with increasing difficulty, and it suitable to children under the age of 13. The second test is in multiplication, with increasing difficulty, and it is suitable to adults and anyone who finds the addition section too easy. Both tests should be completed under 5 minutes.

5.1 Libraries and Dependencies

Pybluez require blue-libs 2.10 our greater, libbluetooth-dev, and other developer packages in GNU/Linux to operate. You will also need to install whatever missing dependencies such as binaries and other static libraries. Numpy is the second library used and it requires you to build the package by source. Taking the general steps of “./configure”, “make”, and “sudo make install”.

5.2 Bluetooth

The Pybluez library works in the program by sending data to the headphone in a digital medium, and receives then converts analog signals to readable data that the computer can use.

5.3 Python

Python was chosen for the project due to its readability, available libraries, and overall good performance. It is a useful language because it is easy enough to learn and use to create small scripts like the one we used in our project. Python is also generally known as a “scientific” programming languages because most people who do use Python in the STEM field use it to quickly carry out their calculations and models.

5.4 OpenVIBE

OpenViBE was used due to its easy compatibility with other languages and easy to use interface. The purpose of OpenViBE is to get data from the acquisition device through the Acquisition Server and then send it to one or more clients. This client is usually, for now, the OpenViBE designer. The Acquisition Server and the clients (Designers) can be either on the same machine or different machines on the same network, or any combination of these. Diagram 2 explains the possibilities. Image 1 and 2 shows the OpenViBE code and what the visual model of the 3D brain looks like.

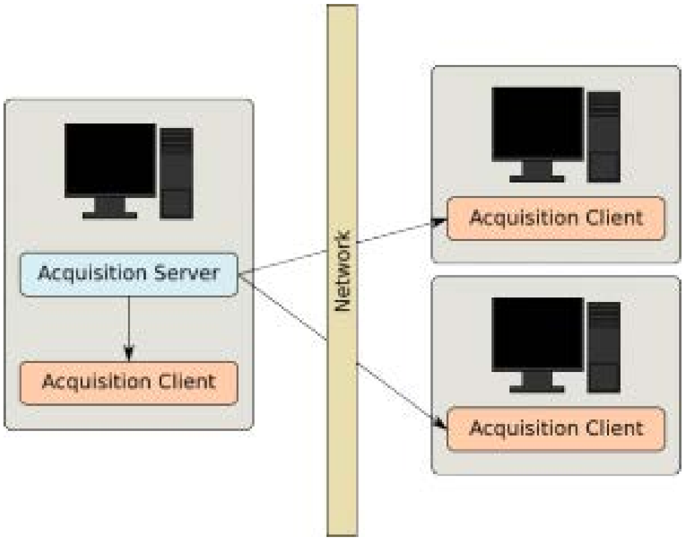


Diagram 2: A diagram of the possibilities of the Acquisition Server connecting onto itself or other clients.

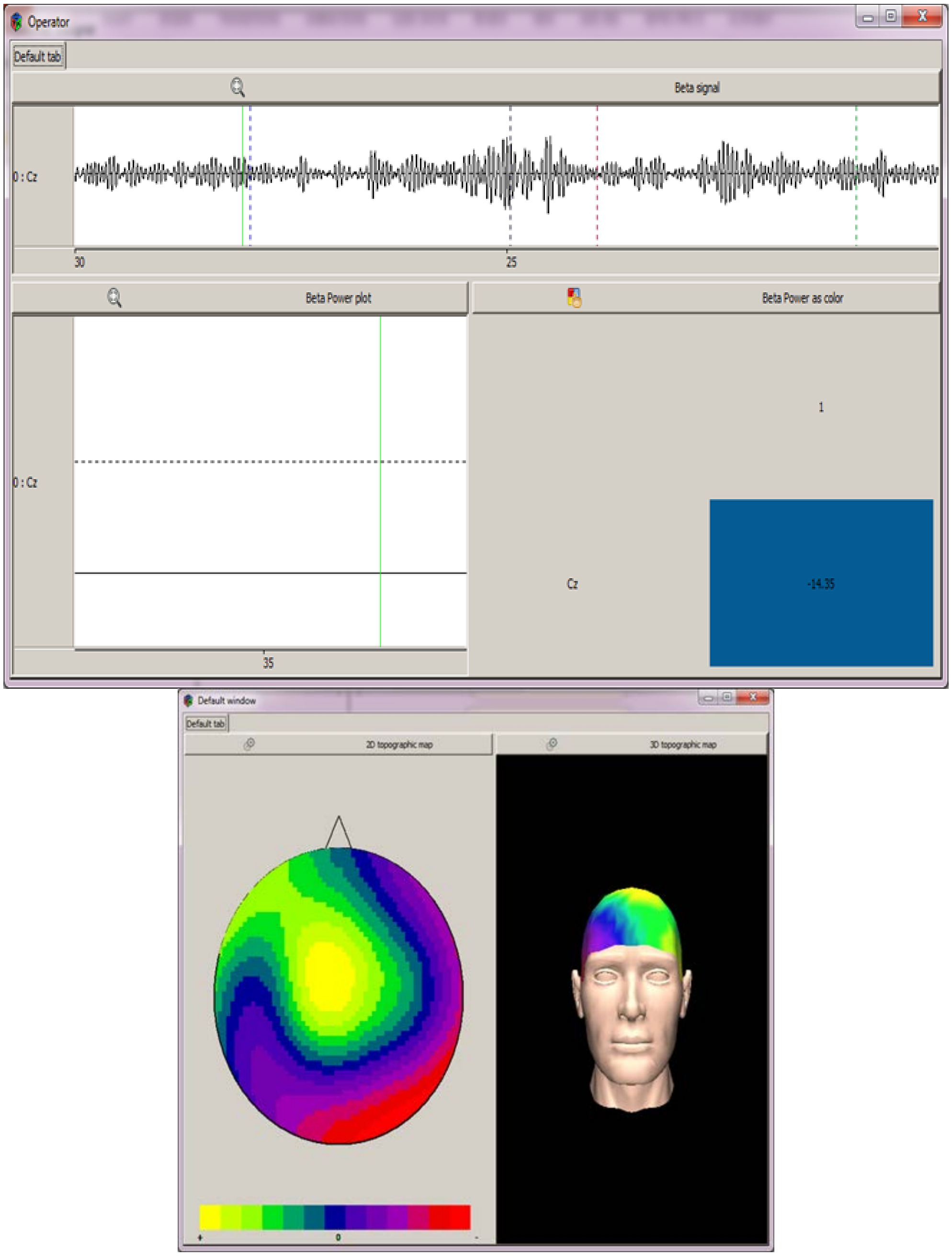
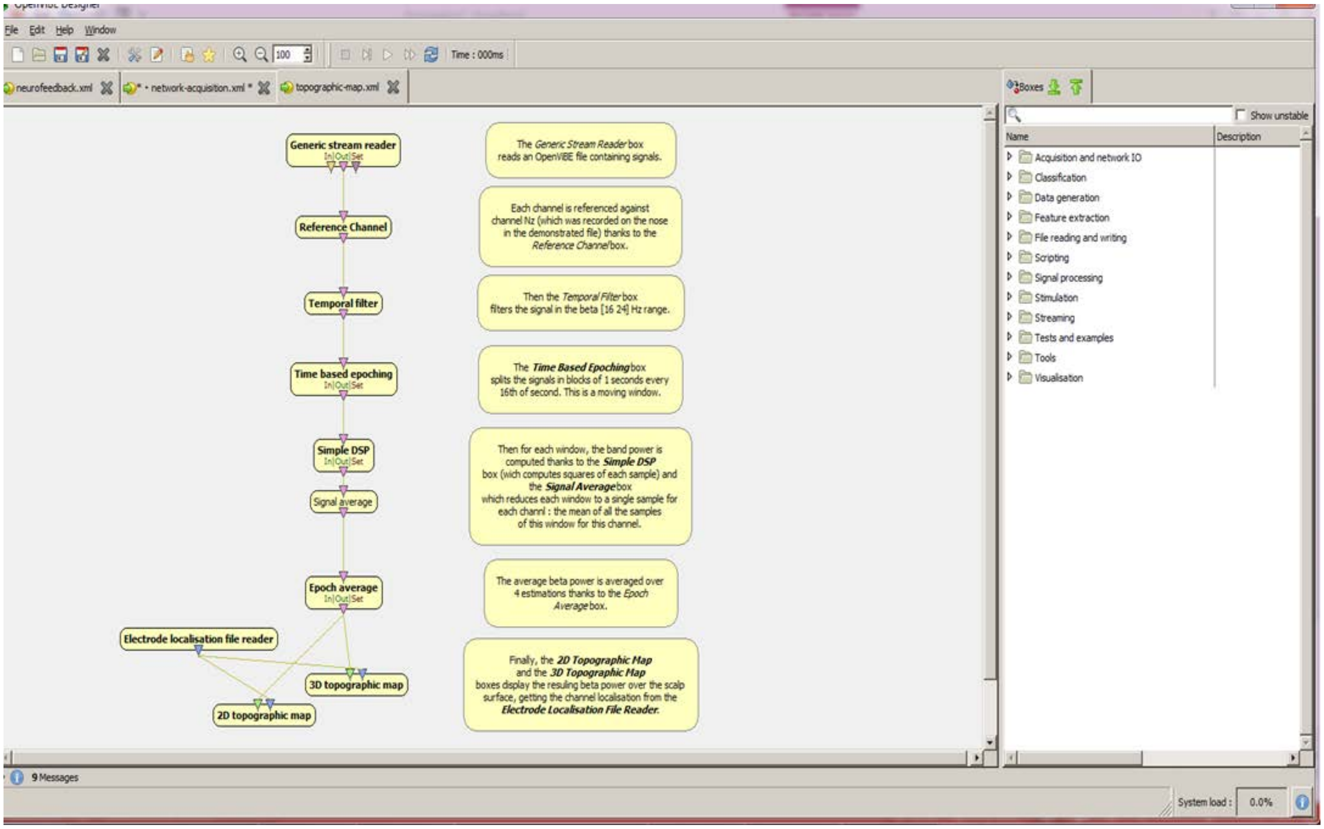


Image 1: Here is a model to simulate the real-time Neurosky Attention and EEG brain waves activity.



The *Generic Stream Reader* box
reads an OpenViBE file containing signals.

Each channel is referenced against
channel Nz (which was recorded on the nose
in the demonstrated file) thanks to the
Reference Channel box.

Then the *Temporal Filter* box
filters the signal in the beta [16 24] Hz range.

The ***Time Based Epoching*** box
splits the signals in blocks of 1 seconds every
16th of second. This is a moving window.

Then for each window, the band power is
computed thanks to the ***Simple DSP***
box (wich computes squares of each sample) and
the ***Signal Average*** box
which reduces each window to a single sample for
each channl : the mean of all the samples
of this window for this channel.

The average beta power is averaged over
4 estimations thanks to the *Epoch
Average* box.

Finally, the ***2D Topographic Map***
and the ***3D Topographic Map***
boxes display the resuling beta power over the scalp
surface, getting the channel localisation from the
Electrode Localisation File Reader.

Image 2: Here is what the visual model code looks like in OpenViBE.

6.0 Methods

Data was collected on how well the subject held his/her attention span in a simple mathematical game. How well the person focused on the problem and how long was recorded also. The game consisted of different “levels” of where equations were set up to test an individual's ability to focus based on their mathematical skill level.. Beta and Gamma waves were monitored due to their high frequency in the brain, and as discussed before, they focus on the activity of the mind (Beta) and the concentration (Gamma). The program’s need for different difficulties is based on Diagram 3. The user inputs their skill level. If the skill level of the user is low, the program will put the user through a series of questions based on their ability and find the “Flow” or the maximum attention span of the user. This is true for a higher skilled user.

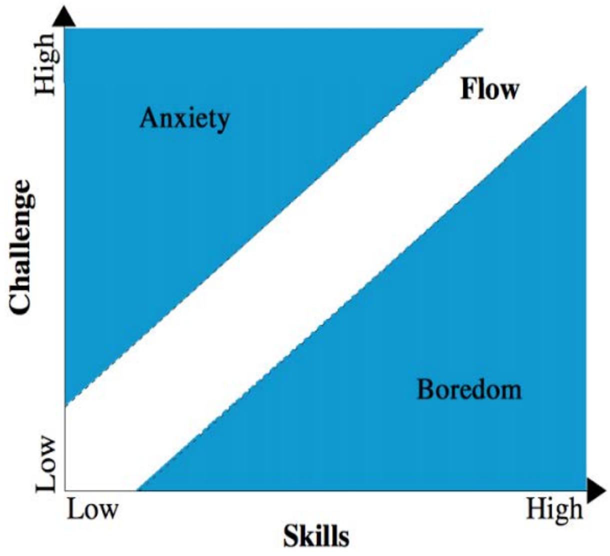
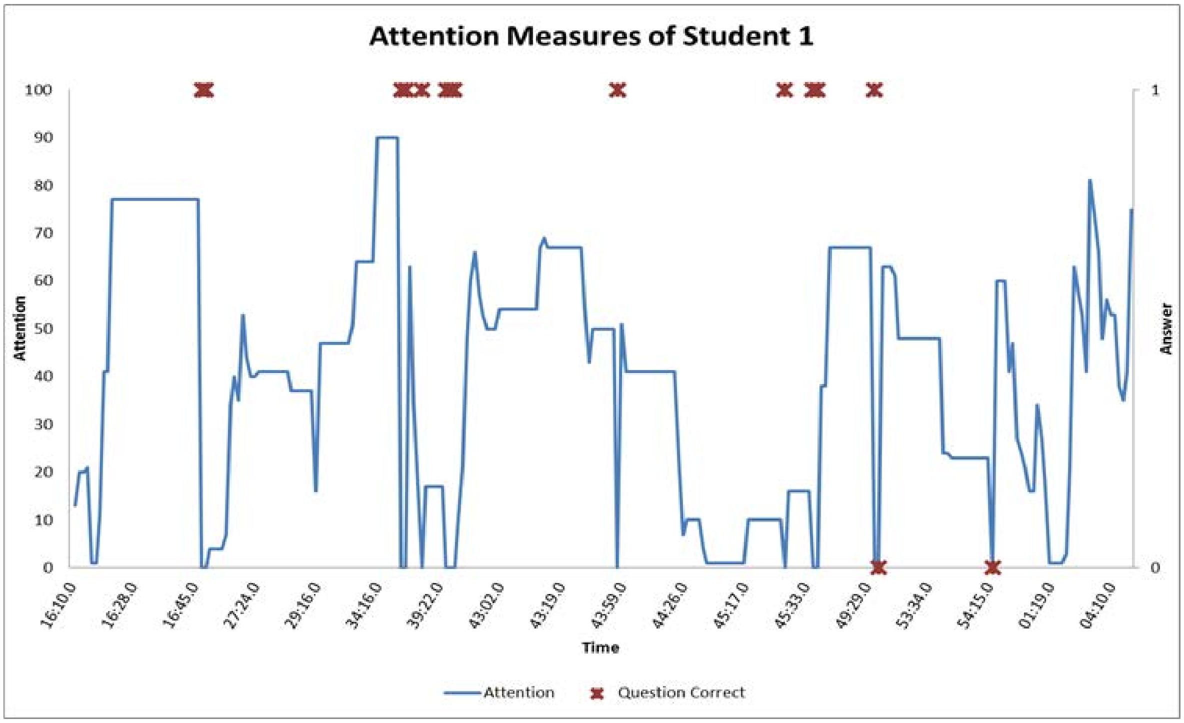


Diagram 3: A two-dimensional model of the flow (or attention span)

7.0 Results/Applications



Graph 1: A screenshot of the attention measures of student1 while working on average workload

Task Workload level (difficulties)	Average attention	Average Number of Correct Answers

Basic	40	10
Medium	60	7
High	35	5

Table 1: Subject 1 attention and performance measures over 5 minutes

Task Workload level (difficulties)	Average attention	Average Number of Correct Answers
Basic	60	10
Medium	80	10
High	55	8

Table 2 : Subject 1 attention and performance measures over 10 minutes

Table 1 and table 2 shows the subject1 attention and performance measures over time. The attention reading is helping measuring the subjects workload to support the best and the ideal test time per individual. Some have very short attention span and it is better to take a test that is divided into sections such as SAT and other do perform better when they have more time. Anxiety is another factor when it comes to taking tests. Adding more time may be an option for some students. This system could help assessing these types of factors.

This kind of software can be used to test a users “awake cycle” or circadian rhythm. This is important to find out when a person’s mind has the greatest attention from start to finish. This can help because the greater the attention then the greater ability to work. Another application is that a user can find what is their attention span during different work loads. If there attention is steady at a higher workload than others then this can be described as this specific user has a ability to focus at high loads.

8.0 Conclusions

In conclusion, users can find their average level of concentration (maximum attention span) using the program. Our program is dynamic, so that the user can take the test in multiple ways. They can use the test to determine which time of day it is best for them to study, by taking the test multiple times of days and comparing results. The user can also find their circadian rhythm or awake cycles using the test. There are numerous applications for this program and there are still more ways to add onto it.

9.0 Future work

In the future extra components can be added such as higher levels of math or the program adjusting to the user based on their skill level during the test. Age will also be factored into the program as different age groups have different attention spans and thus resulting in varying workloads.

10.0 Appendix

10.1 Acknowledgements

We would like to thank our parents, our teacher Rebecca Galves, Dr. Cooper, and our coach Sam McGuinn.

10.2 References

[1] - https://code.google.com/p/pybluez/wiki/Documentation

[2] - http://neurosky.com/

[3] - http://www.cs.tufts.edu/~remco/publications/2014/CHI2014-UAV.pdf

[4] - http://dl.acm.org/citation.cfm?id=2332700.2349988&coll=DL&dl=GUIDE&CFID=448985143&CFTOKEN=53565371

[5] - http://www.ncbi.nlm.nih.gov/pubmed/22506483

[6] - http://www.ncbi.nlm.nih.gov/pubmed/17366276

[7] - http://4.bp.blogspot.com/-zD2hZjOxkYo/TZHdfVJOzXI/AAAAAAAABxA/WVA6fcnLwro/s1600/EEG+Brain+Waves.png

[8] - http://www.ncbi.nlm.nih.gov/pubmed/19145492

[9] - http://support.neurosky.com/kb/science/how-to-convert-raw-values-to-voltage

[10] - http://www.phon.ucl.ac.uk/courses/spsci/dsp/fftbasis.html

[11] - https://engineering.purdue.edu/~ipollak/ee438/FALL04/notes/Section1.4.pdf

[12] - http://en.wikipedia.org/wiki/Fast_Fourier_transform

[13] - http://pe.org.pl/articles/2014/2/44.pdf

[14] - http://web.eecs.umich.edu/~fessler/course/451/l/pdf/c6.pdf

[15] - http://daim.idi.ntnu.no/masteroppgaver/006/6288/masteroppgave.pdf

10.3 Code

```
import bluetooth

correctAnswers1 = 0
incorrectAnswers1 = 0
correctAnswers2 = 0
incorrectAnswers2 = 0

#find device
print("performing inquiry...")

nearby_devices = bluetooth.discover_devices()

print("found %d devices" % len(nearby_devices))

for addr, name in nearby_devices:
	print("  %s - %s" % (addr, name))

#while bluetooth.server_sock() = open:
#	open bluetooth.service_(server_cock "Mindwave")
#	server_sock.recv(1024)
#	print "recieved [s%]" "wave"

#timer
timeit.default_timer()

#math problems
def mathProblems():
	print 'The levels of difficulty are as follows: \n 1 - Addition \n 2 - Multiplication'
	difficulty = input('Please select a difficulty by inputing a number')

	if dificulty == 1:
	userAnswers = input('What is the value of 1 + 1')
	if userAnswers == 2:
	print 'correct'
	global correctAnswers1
	correctAnswers1 += 1
	print 'So far you have answered', correctAnswer1, 'questions correct'
	else:
	print 'incorrect'
	global incorrectAnswers1
	incorrectAnswers1 += 1
	print 'So far you have answered', correctAnswer1, 'questions correct'

	userAnswersA1 = input('What is the value of 34 + 56')
	if userAnswersA1 == 90:
	print 'correct'
```

```
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA2 = input('What is the value of 96 + 52')
if userAnswersA2 == 148:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA3 = input('What is the value of 916 + 522')
if userAnswersA3 == 1438:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA4= input('What is the value of 196 + 152')
if userAnswersA4 == 348:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA5 = input('What is the value of 9102 + 52')
if userAnswersA5 == 9154:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
```

```
userAnswersA6 = input('What is the value of 73 + 498')
if userAnswersA6 == 571:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA7 = input('What is the value of 731 + 1498')
if userAnswersA6 == 2229:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA8 = input('What is the value of 58911 + 14261')
if userAnswersA8 == 73172:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA9 = input('What is the value of 48932 + 146459')
if userAnswersA9 == 195391:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
incorrectAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'

userAnswersA10 = input('What is the value of 932123 + 16129')
if userAnswersA10 == 948252:
print 'correct'
correctAnswers1 += 1
print 'So far you have answered', correctAnswer1, 'questions correct'
else:
print 'incorrect'
```

```
        incorrectAnswers1 += 1
        print 'So far you have answered', correctAnswer1, 'questions correct'

#       averageHertz = (bluetooth.Mindwave(int 1) * (correctAnswers1 / timeElapsed))
#       return averageHertz


        if dificulty == 2:
        userAnswers = input('What is the value of 1 X 1')
        if userAnswers == 1:
        print 'correct'
        global correctAnswers2
        correctAnswers2 += 1
        else:
        print 'incorrect'
        global incorrectAnswers2
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'

        userAnswers1 = input('What is the value of 4 X 5')
        if userAnswers1 == 20:
        print 'correct'
        correctAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'

        userAnswers2 = input('What is the value of 14 X 6')
        if userAnswers2 == 84:
        print 'correct'
        correctAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'

        userAnswers3 = input('What is the value of 14 X 6')
        if userAnswers3 == 84:
        print 'correct'
        correctAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
```

```
print 'So far you have answered', correctAnswer2, 'questions correct'

userAnswers4 = input('What is the value of 31 X 6')
if userAnswers4 == 186:
print 'correct'
correctAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'
else:
print 'incorrect'
incorrectAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'

userAnswers5 = input('What is the value of 21 X 61')
if userAnswers5 == 1281:
print 'correct'
correctAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'
else:
print 'incorrect'
incorrectAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'

userAnswers6 = input('What is the value of 4 X 523')
if userAnswers6 == 2092:
print 'correct'
correctAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'
else:
print 'incorrect'
incorrectAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'

userAnswers7 = input('What is the value of 41 X 223')
if userAnswers7 == 9143:
print 'correct'
correctAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'
else:
print 'incorrect'
incorrectAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'

userAnswers8 = input('What is the value of 141 X 298')
if userAnswers8 == 42018:
print 'correct'
correctAnswers2 += 1
print 'So far you have answered', correctAnswer2, 'questions correct'
```

```
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'

        userAnswers9 = input('What is the value of 1238 X 8463')
        if userAnswers9 == 10477194:
        print 'correct'
        correctAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'


        userAnswers10 = input('What is the value of 11411 X 29620')
        if userAnswers10 == 337993820:
        print 'correct'
        correctAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'
        else:
        print 'incorrect'
        incorrectAnswers2 += 1
        print 'So far you have answered', correctAnswer2, 'questions correct'

#       averageHertz = (bluetooth.Mindwave(int 1) * (correctAnswers1 / timeElapsed))
#       return averageHertz

print averageHertz
```