Andrejs Sorstkins

BCI and Virtual Home – Final Report

Research Project BSC Computer Science

University of Essex

Table of Content

1. Introduction

We live in a rather interesting time, when technologies that seemed like science fiction yesterday are gradually entering our lives today. Or at least they are taking the first timid steps towards it. One example is the technology of neurocomputer interfaces. On the one hand, this is just another way of interaction between man and machine, and on the other, something more revolutionary...

The goal of the project is to explore the current scientific developments in this area of neuroscience and implement them in a system that will be able to help paralyzed people to control a -smart home using the power of thought.

Main goals of the project:

1. Create a virtual environment for product testing
2. Explore all possible technologies and methods available at the moment and choose the best one in terms of end user experience
3. Understand as much as possible the nature of imagination and try to create a BCI system that uses it
4. Make the system simple to use for the end user

Big part of this project consists of research work due to experimental nature of a project because of lack of full image about human brain working mechanism. I am not expecting that final result will have accuracy more than 70% due to complexity and weaknesses of human brain signal to capture by different brain signal recording systems.

Main Objectives of a Project:

1. Unity - create room
2. Create TV- on/off functionality
3. Create Lamp - on/off functionality
4. Create character and character movement part
5. Use one or several BCI technology for movement
6. Use one or several BCI technology for objects control
7. Muscle signal extraction and calculation
8. C# language learning (it is needed for Unity Engine)
9. Medical and BCI data collection and use
10. Extrapolation of a system

There are 2 main parts of a project development. It is Unity engine part in which we create area for our simulation and product testing. Virtual Area is needed because it gives us freedom to transit main code to real life environment for future development of a system.

This research project will consist of 3 main parts:

- Part 1: Research in Neuroscience, BCI technology and Unity documentation - my aim to spend autumn on all "education" and research part because I have zero level experience in BCI and Unity at the beginning of the project
- Part 2: Implement knowledge from Part 1 to create MVP - my aim to spend wintertime period on Part 2
- Part 3: Exchange system by adding more features and user interaction interface

My main aim for choosing this type of project despite its complexity and my lack of knowledge in neuroscience is interest in human brain, psychology and consciousness. I think after finalizing this project, I will come closer in understanding human and living creatures’ nature

2. Background & Literature review

Modern ways to control a computer are a mouse, keyboard, touch screen. Gradually, control through gestures, voice comes into life. The computer already knows how to follow our pupils, the direction of our gaze. The next stage of evolution of human-machine interaction is the direct reading of signals from the nervous system, that is, the brain-computer interface.

A neurocomputer interface (NCI) is a system designed to exchange information directly between the brain and a computer. The concept was described by J. C. R. Licklider back in 1960 in an article with the sonorous title "Man-Computer Symbiosis".

To put it very simply, the human nervous system during operation generates, transmits and processes electrochemical signals in different parts of the body. And the electrical component of these signals can be tried to "read" and interpret.

For these purposes, you can use different methods, which have their own advantages and disadvantages. For example, you can take signals using magnetic resonance imaging (MRI), but the devices are too bulky

You can constantly introduce special marker substances, but by doing so you can harm the body. Finally, small sensors can be applied or implanted to specific parts of the body to record electrochemical signals. There was developed a lot of method bet all of them has its own pros and cons

2.1 Neuroscience Background - Structure of Neuron

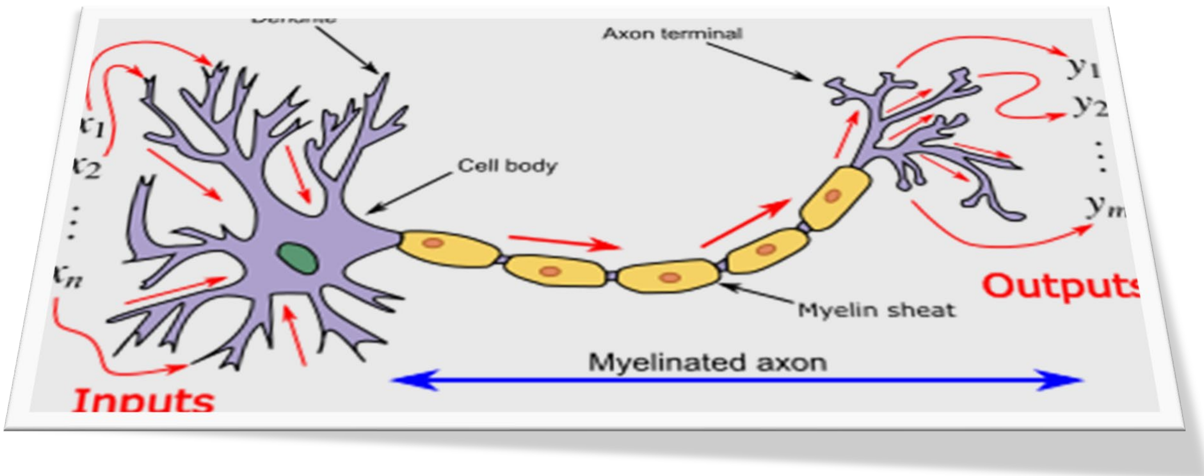


Figure 1 Neuron Scheme

A bio neuron consists of a body with a diameter of 3 to 100 microns, containing a nucleus and processes. There are two types of shoots. The axon is usually a long process adapted to conduct excitation from the body of the neuron. Dendrites are usually short and very branched processes that serve as the main site for the formation of excitatory and inhibitory synapses that affect the neuron (different neurons have a different ratio of the length of the axon and dendrites).

A neuron may have several dendrites and usually only one axon. One neuron can have connections with 20 thousand other neurons. The human cerebral cortex contains 10 billion neurons.

Bio neuron is an important element of the cells of the nervous system and the building material of the brain. Neurons come in several forms, depending on their purpose and location, but in general they are identical in structure.

Each neuron is an information processing device that receives signals from other neurons through a special input structure consisting of dendrites. If the total input signal exceeds the threshold level, then the cell transmits the signal further to the axon, and then to the signal output structure, from which it is transmitted to other neurons. Signals are transmitted using electronic waves. (During a person's life, the number of neurons does not increase, but the number of connections between them increases as a result of learning).

2.2 Neuroscience Background - Structure of a Brain

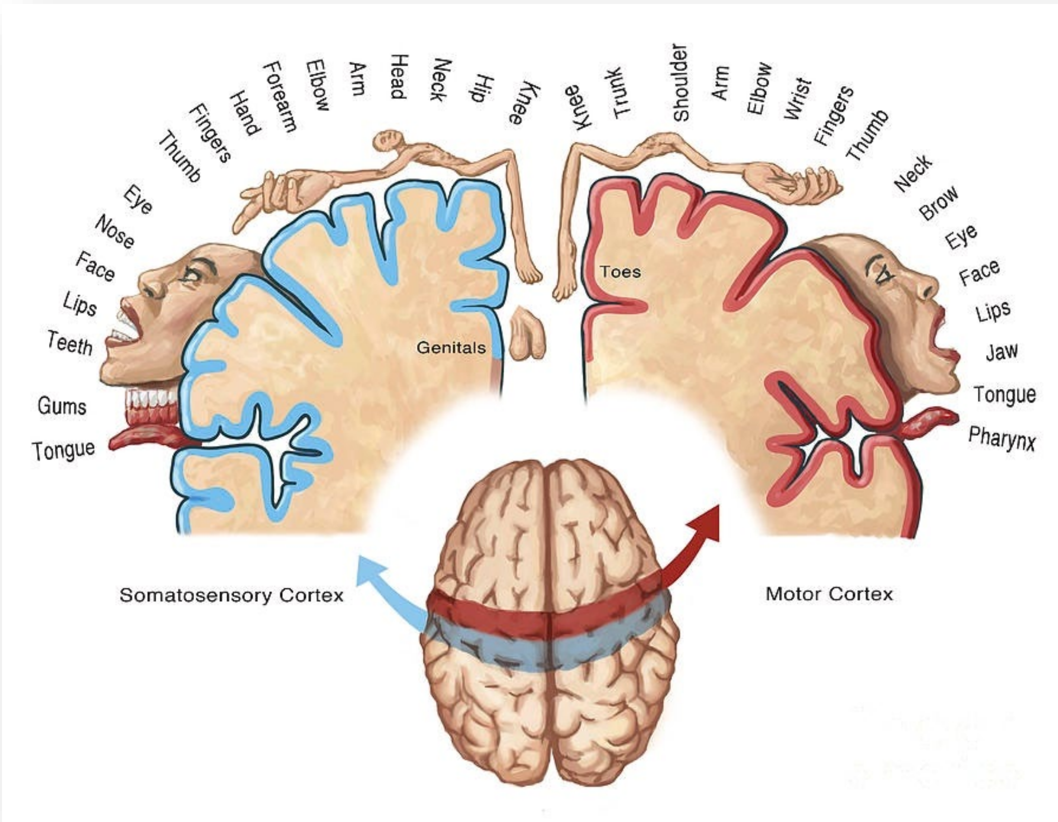


Figure 2 Diagram of brain functions relative to body parts

I am going to discuss several components of a brain which can be used for signal extraction.

The Central Nervous System consists of the brain and spinal cord.

- The brain is the main part of the CNS and is located in the skull.
- The spinal cord is a long whitish cord located in the spine and connecting the brain with the rest of the body. It acts as a kind of information highway between the brain and the body, passing information from the brain to the body.

Basal Ganglia: subcortical neural structures responsible for motor functions. They receive information from the cortex and brain stem, process it and re-project it into the cortex, medulla oblongata and brain stem, providing coordination of movements. Made up of several departments:

- The caudate nucleus is a C-shaped nucleus involved in the control of conscious movements, as well as in the processes of learning and memory.
- The amygdala plays a key role in controlling emotions, especially fear. The amygdala helps store and categorize memories triggered by emotions.

Temporal lobe: separated from the frontal and parietal lobes by the Sylvian sulcus and the borders of the occipital lobe. Involved in hearing and speech, as well as memory and emotion control.

Frontal lobe: the largest lobe of the cerebral cortex. It is located in front of the skull behind the forehead. Extends from anterior to Roland's furrow. This is the center of command and control of the brain, the conductor of the orchestra. It is closely related to executive functions, i.e. responsible for planning, reasoning, problem solving, judgment, impulse control, and the regulation of emotions such as empathy and generosity, behavior.

As u can see from this brief description, processing of human activity is happening mainly in a different region (it is good for data extraction), bet all regions is interconnected to each other which make extraction harder, also structure of brain of each person is similar on its basis, bet have big differences if we dive into details and aim to get precise result for a different user.

2.3 Main Components of BCI

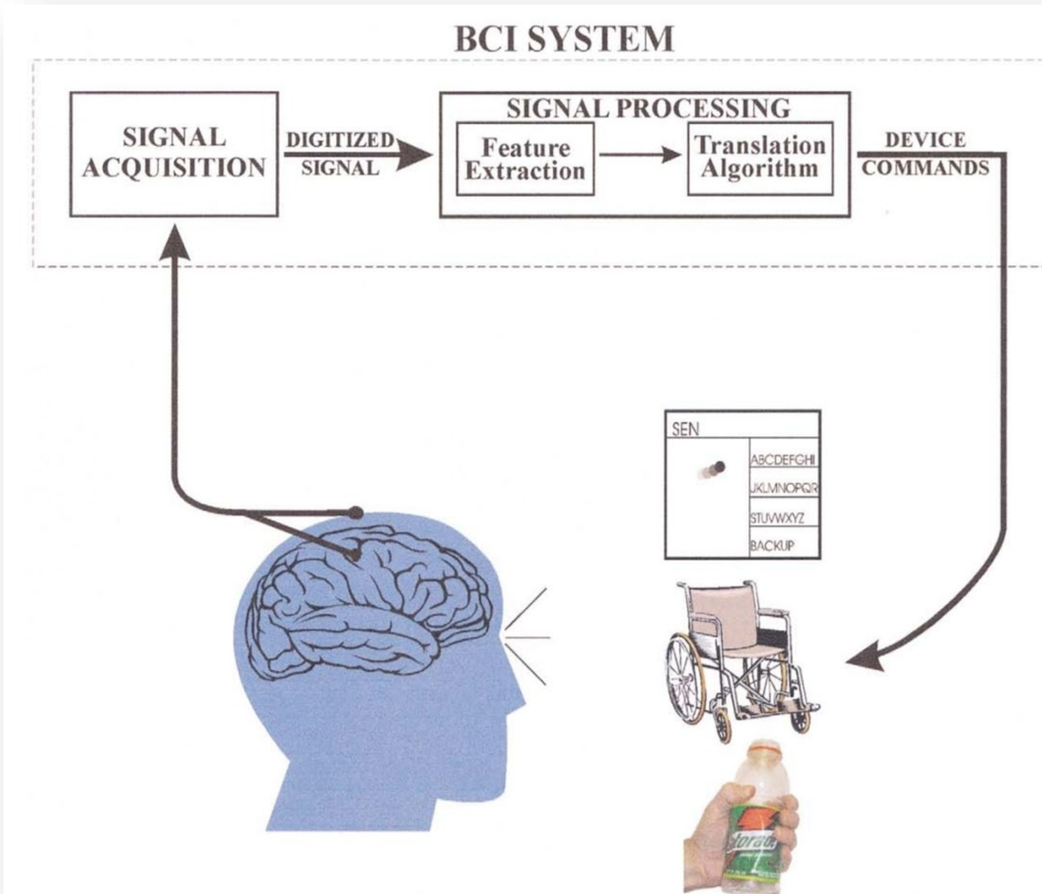


Figure 3 Basic design and operation of any BCI system. Signals from the brain are acquired by electrodes on the scalp or in the head and processed to extract

Basic design and operation of any BCI system. Signals from the brain are acquired by electrodes on the scalp or in the head and processed to extract specific signal features (e.g., amplitudes of evoked potentials or sensorimotor cortex rhythms, firing rates of cortical neurons) that reflect the user's intent

Step 1: Signal acquisition

BCIs can be categorized by whether they use **non-invasive** (e.g., EEG) or **invasive** (e.g., implanted chip) methodology. The main difference between invasive and non-invasive systems from system point of

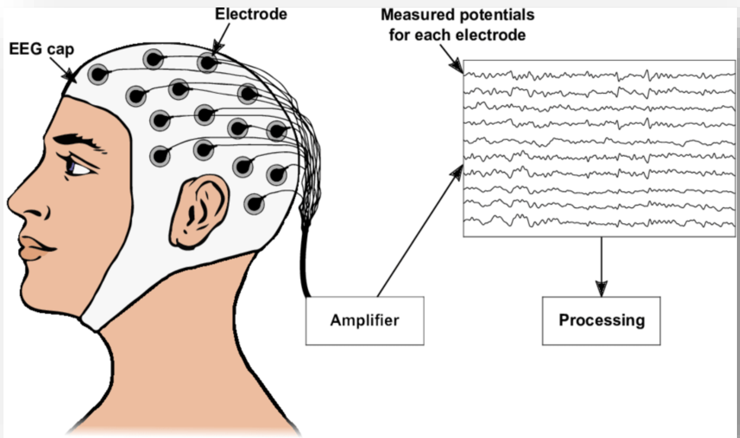


Figure 5 Electroencephalography (EEG) Diagram, non-invasive method

view is signal accuracy measurements. Invasive method will be much more accurate bet require surgical intervention which can be unacceptable for most of users from moral and ethical point of view. It can be good described with concert hall analogy. Non-invasive method - we are trying to record man and girl

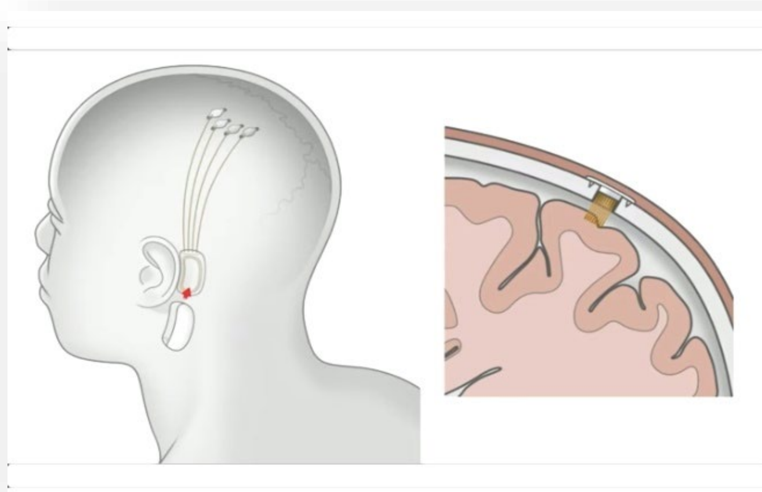


Figure 4 Neuralink chip, invasive method

conversation in the crowd of people in hall from scene. Invasive method - we came with microphone to man and girl and record their conversation.

They can also be categorized by whether they use evoked or spontaneous inputs. The main point of this distinction is that we don't know from what time point to start record data inside the brain because brain is not stopping and "sleeping". It constantly produces electrochemical signals. Evoked inputs - use some stimulus from outside which can be visible on recording of EEG, as example flesh light to an eye. Spontaneous inputs can describe by well-known internal patterns in a brain e.g., EEG rhythms over sensorimotor cortex, or some time slot.

Signal acquisition part is very important because by using different ways of measuring we are creating different BCI systems.

Step 2: Signal processing: feature extraction

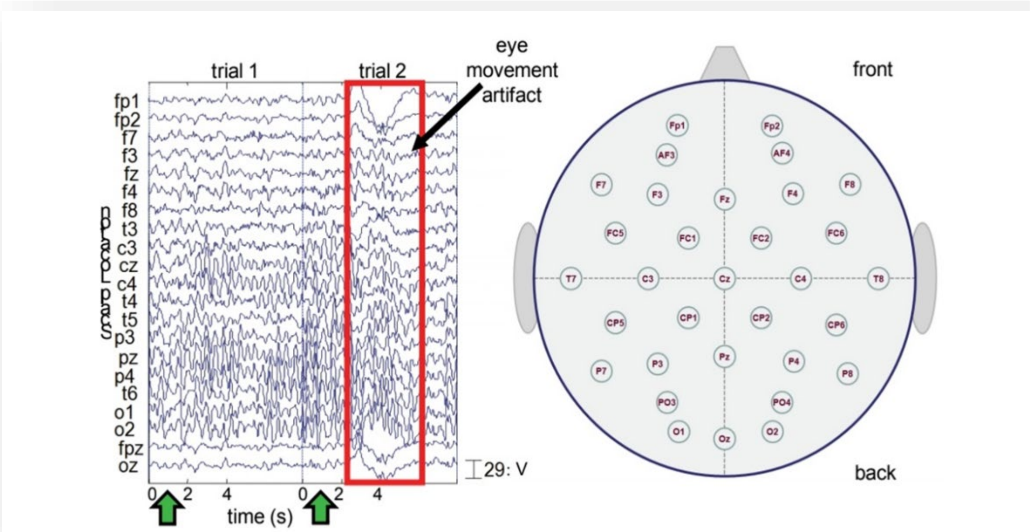


Figure 6 EEG Output Demonstration

The digitized signals are then subjected to one or more of a variety of feature extraction procedures, such as spatial filtering, voltage amplitude measurements, spectral analyses, or single-neuron separation. This analysis extracts the signal features that (hopefully) encode the user's messages or commands. BCIs can use signal features that are in the time domain (e.g., evoked potential amplitudes or neuronal firing rates) [1]

In general, the signal features used in present-day BCIs reflect identifiable brain events like the firing of a specific cortical neuron or the synchronized and rhythmic synaptic activation in sensorimotor cortex that produces a mu rhythm

Step 3: The translation algorithm

The next stage, the translation algorithm, translates these signal features into device commands orders that carry out the user's intent. This algorithm might use linear methods (e.g., classical statistical analyses or nonlinear methods (e.g. neural networks). Whatever its nature, each algorithm changes independent variables (i.e., signal features) into dependent variables (i.e., device control commands).

Effective algorithms adapt to each user on 3 levels.

> *First level*, when a new user first accesses the BCI the algorithm adapts to that user's signal features, e.g. If the signal feature is mu rhythm amplitude, the algorithm adjusts to the user's range of mu-rhythm amplitudes

> *Second level*, EEG and other electrophysiological signals typically display short- and long-term variations linked to time of day, hormonal levels, immediate environment, recent events, fatigue, illness, and other factors.

Third level, the third level of adaptation accommodates and engages the adaptive capacities of the brain, therefore we must take in account that we can teach not only system to understand brain bet also we can teach brain to use system and give as needed patterns of electrochemical signals.

Proper design of this third level of adaptation is likely to prove crucial for BCI development. Because this level involves the interaction of two adaptive controllers, the user's brain and the BCI system

Step 4: The output device

In a simple word, we must output result of an algorithm. Usually, result of an output is displayed on a computer screen bet it is also can be any mechanical device, e.g., wheelchair

Step 5: The operating protocol

Each BCI has a protocol that guides its operation. This protocol defines how the system is turned on and off, whether communication is continuous or discontinuous, whether message transmission is triggered by the system (e.g., by the stimulus that evokes a P300) or by the user, the sequence and speed of interactions between user and system, and what feedback is provided to the user.

2.4 Signal acquisition technology

There are 2 main types of signal acquisition technologies. It is **non-invasive** or **invasive**.

Invasive technology is much more accurate and precise bet requires surgical intervention and increase complexity of finding people who are willing to take part in such experiments from moral, legal and ethical point of view. Also, post-surgery damage risks (including possible irreversible loss of neural tissue) are the main issues. Process of finding of such group of people and compliance of all legal norms can take by itself years to do, therefore, decision was taken to avoid using invasive technologies at this stage of the project.

The decision was made to use non-invasive BCI. There are numerous types of non-invasive technology with their pros and cons.

Figure 7 Functional magnetic resonance imaging (fMRI)

1. Functional magnetic resonance imaging (fMRI) - This is a powerful technology used for functional brain mapping based on hemodynamics (i.e., blood flow and oxygenation changes).

Figure 8 Positron emission tomography (PET)

2. Positron emission tomography (PET) - In principle this technology can be used for brain mapping, usually through radioactive oxygen or glucose given to the user.

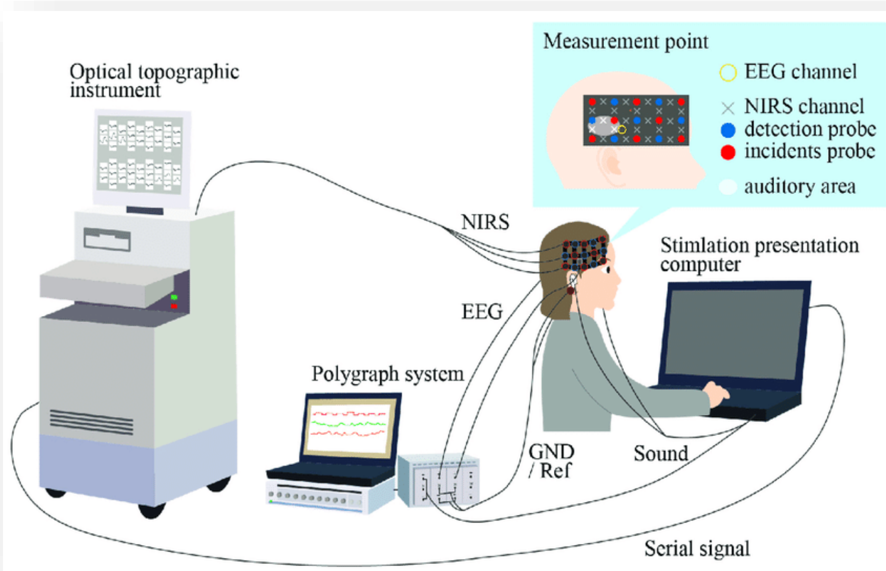


Figure 9 Near infrared spectroscopy (NIRS):

3. Near infrared spectroscopy (NIRS): This method too is based on hemodynamics. In this case blood oxygenation changes are linked to the amount of reflected near-infrared light applied on the brain thorough transmitters on the scalp, the receiver being placed nearby on the scalp as well. The approach is similar to that used in existing sensors using mid-range infrared, but near infra-red has much deeper penetration in tissue (up to a several centimeters), lending itself to brain monitoring.

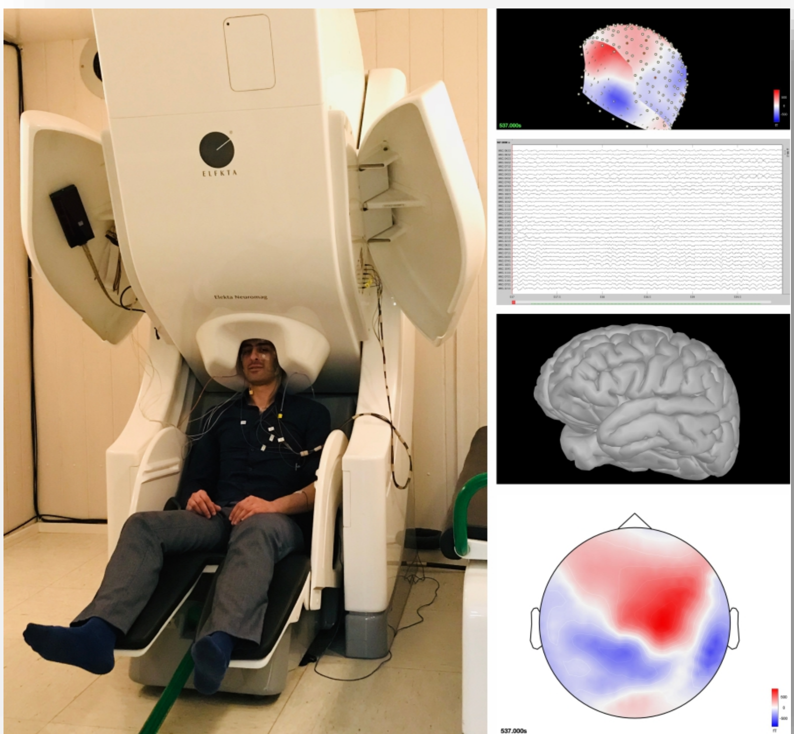


Figure 10 Magnetoencephalography (MEG)

4. Magnetoencephalography (MEG) - MEG records the magnetic fields orthogonal to the electric fields generated by ensemble neural activity, although there is evidence suggesting that the source of detectable magnetic fields in the brain is physiologically different from those generating EEG

5. Electroencephalography (EEG): This is by far the oldest of all the devices discussed here, having been available at least since the 1920's (Swartz and Goldensohn, 1998). In EEG, the electrodes are usually placed on the scalp and record the electrical activity taking place in the brain tissue underneath, which reach the electrode region by volume conductor processes. For neural activity to be detectable using EEG, it must be both fairly near the cortical surface and include many millions of cells in synchrony so that total sum of the activity is large enough to be detected from the scalp. Still, the largest EEG potentials seen under most conditions have amplitudes in the order of a few tens of microvolts.

2.5 Translation algorithm and operating protocol

The main challenge for extraction of features and overall processing of data is that EGG brain signal data set is looks like noise. Good analogy which describes complexity of task can be trial of record the chirping of a bird in 20 meters out of microphone in a crowd of people and passing cars around it. There are several methods which try to solve this task.

Signature - is characteristics of the brain signal which are uniquely caused by a mental process or state.

The signatures that have shown to be useful for BCI can be broadly categorized into evoked and induced responses.

Evoked responses are time- and phase-locked to an event. This means that averaging repeated signals will increase the signal-to-noise ratio.

Induced responses are not phase locked but the power, rather than the phase, is time locked to the stimulus. That is, the power in specific frequency bands has to be calculated before averaging across trials [2]. The measured response is usually referred to as an event-related potential (ERP) or event-related field (ERF) [3]

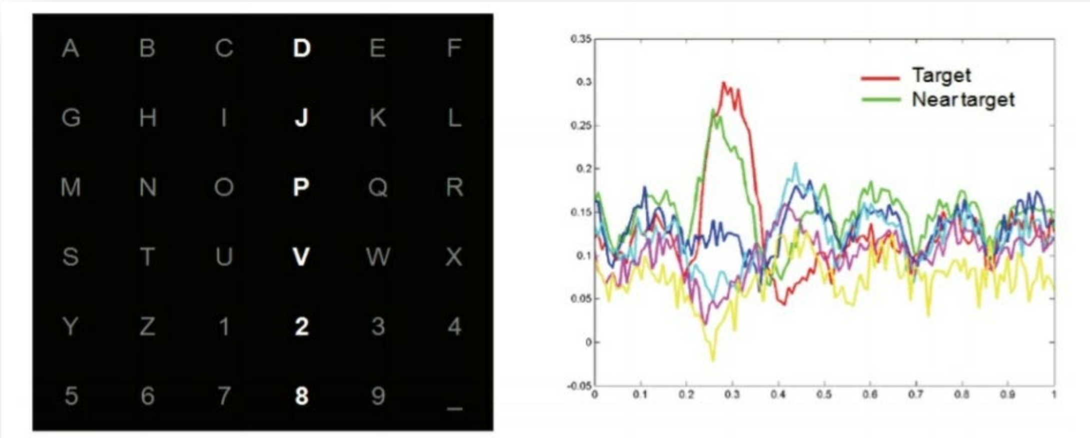


Figure 11 Menu of P300 mechanism

1. P300 - This approach falls under the event-related potential category

Typically, each row and column will flash for a short period (about 100ms) in a random sequence on a computer screen. When the row or column containing the desired choice flashes, the user adds 1 to a mental counter to signal that a target has flashed. For example, if the user wants to type the letter P using a BCI, she/he will count every time a row column containing it flashes.

On average, when a target row/column flashes, a strong signal is seen (especially in the centro parietal electrodes, Fig. 11) which will peak at about 300ms after the desired object flashed, hence the P300 name

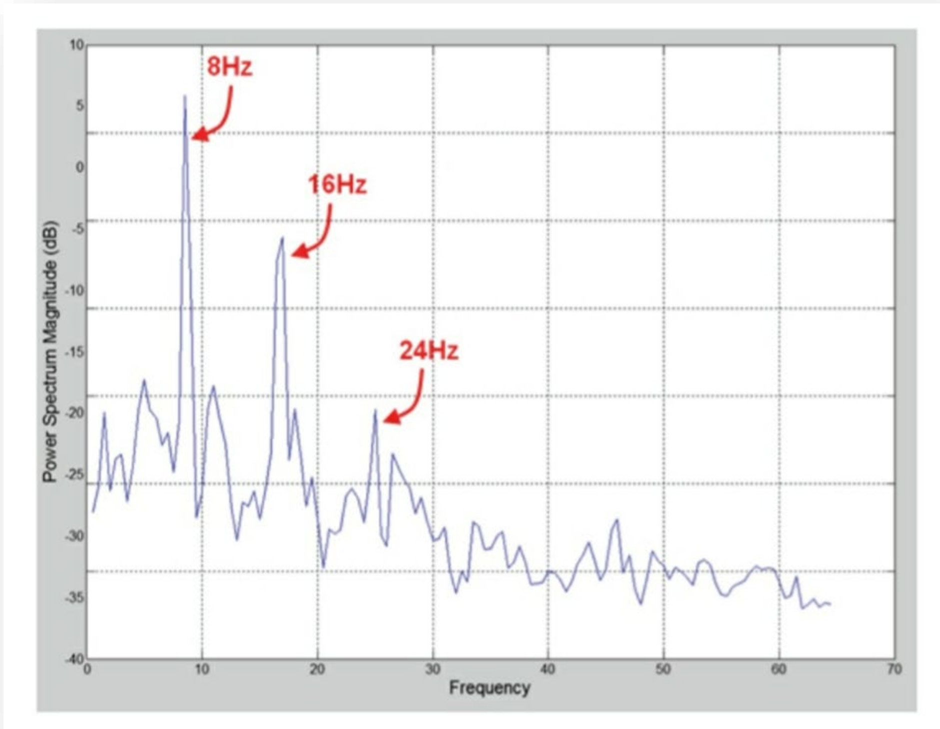


Figure 12 SSVEP visual cortex, red arrow show light fleshing in different frequency

2. Steady state visually evoked potential (SSVEP) - The P300 method above is like the SSVEP approach in that the user is presented with an array containing flashing objects from which the user chooses one. However, in the SSVEP method each object flashes at a different frequency, usually between 6Hz and about 35Hz (Gao et al., 2003). When the user fixates his/her gaze on a flashing object, that object's flashing frequency will be seen as a strong frequency-domain element in the EEG recorded

from areas above the visual cortex (occipital areas, Fig. 3). For example (Fig. 9), if the user is interested in number 7 on a number array, fixating his/her gaze on that object (which is this example is flashing at 8Hz) will produce the power spectrum shown on the right panel in Fig. 9, which is an average of five trials (i.e., target flashing cycles).

Notice that the user must have eye gaze control for this approach to work, but, as mentioned above, this ability is retained by the vast majority of potential BCI users, both disabled and able-bodied.

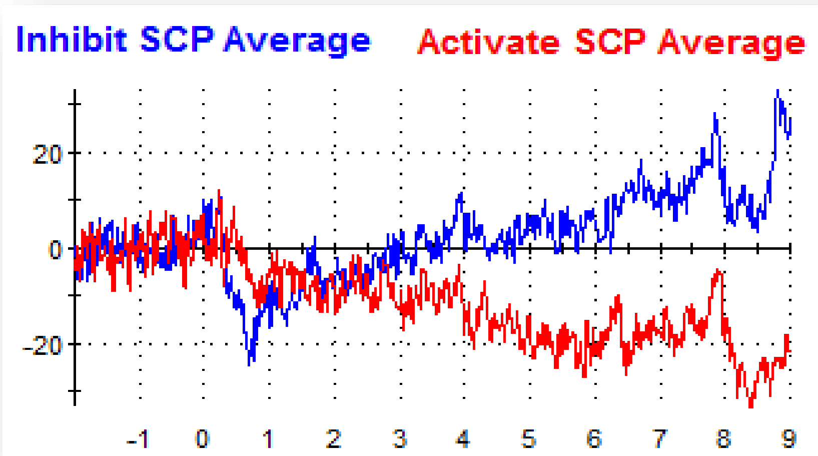


Figure 13 SCP, active and inhibit average

3. SCP - slow cortical potential - Negative SCPs develop without any voluntary or learned intention for regulation in situations where motor or cognitive preparation is involved. These preparatory cortical responses were called contingent negative variation. An anticipatory threshold regulation mechanism was proposed as the neurophysiological basis of SCP to consist of a neostriatum-neocortex-thalamic loop which may also be responsible for voluntary regulation of cortical negativity and positivity during BCI control.

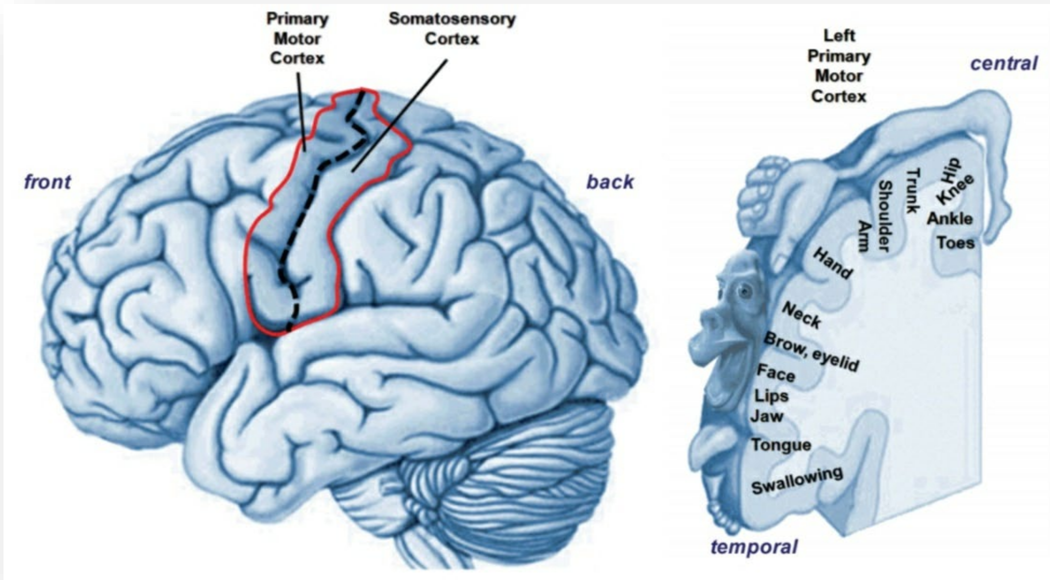


Figure 14 Motor Imaginary Cortex

4. Motor Imagery (MI) - is straight forward task and mechanism to use for a person. It involves imaging of movement for different part of a body.

Neuroscience mechanism for motor Imaginary is localized primarily in motor cortex

5. Sound Imaginary (SI)- potentially this method can replace all others due to it simplicity and naturality for humans. It involves telling commands with person inner voice.

In terms of functional neuroanatomy, the processes involved in covert speech have not been fully study, but it is known that it involves the auditory cortex (around Brodmann areas) and, for speech-related imagery, Wernicke’s area.

3. Design & Implementation

3.1 Design

This section will consist of description of different technologies which can be used in a project, their pros and cons in terms of the project and design decisions made during development process

3.1.1 Signal acquisition stage

1. Functional magnetic resonance imaging (fMRI)

Pros: It provides excellent spatial resolution.

Cons: The equipment is very large, heavy and expensive.

2. Positron emission tomography (PET)

Pros: spatial resolution is better than with EEG.

Cons: First and foremost, then use of radioactive substances precludes use of this technology in BCIs, although in extreme cases it can be useful for validating other methods as it is a well-established technology, having been available for several decades.

3. Near infrared spectroscopy (NIRS):

• Pros: This is currently the cheapest hemodynamics-based technology available, although the equipment is still more expensive than for EEG. Compared to other hemodynamics devices, NIRS equipment is also fairly portable, and wireless systems have been recently developed as well (Muehlemann et al., 2008).

• Cons: As in other hemodynamics-based systems, time resolution is poor. On the other hand, different from PET and fMRI, spatial resolution is poor due to significant scattering of the near-infrared light in tissue. NIRS systems are also very sensitive to transmitter and sensor motion and environmental NIR sources.

4. Magnetoencephalography (MEG):

• Pros: It has much better time resolution than hemodynamics-based systems. Electrical and magnetic field changes reflect the underlying neural activity within a few milliseconds. Also, in contrast with fMRI, PET and NIRS, MEG only monitors brain signals and does not deposit any matter or energy on the brain. It is thus a truly non-invasive technology in that it does not disturb the object of study.

• Cons: MEG is still very large, comparable in size with fMRI equipment. It is also very costly.

5. Electroencephalography (EEG):

Pros: EEG is the least expensive technology for brain monitoring. EEG systems are also very portable and provide excellent time resolution. Due to their passive nature (from the brain's point of view), they are very safe as well.

Cons: There are two main limitations in EEG systems. One (poor spatial resolution) is inherent while the other (poor usability) can still be tackled. Poor spatial resolution is inherent due to the volume conductor effects through which signals from nearby (even up to a few cm apart) areas are irreversibly mixed together. While various approaches exist to minimize this problem, sub-centimeter EEG features have little meaning. Poor usability stems from the need to use electrode gel to reduce the impedance between the electrodes and the scalp, but there are commercial systems currently available that employ user-friendly wet electro

3.1.2 Discussion of design choice for signal acquisition

The main **criteria** for our project:

1. Technology must be compact (We are creating product for smart home) and
2. Technology must be safe to use
3. Technology must have good time resolution for request, response mechanism between client and control system. des or dry (usually capacitive) electrodes, at the expense of signal quality.

We reject use of PET due to second and third criteria. fMRI is very big and expensive to use for a project. NIRS has low time resolution, and it is also very sensitive to transmitter and sensor motion which will cause degradation of user experience.

The decision was made to use EEG based system due to fact that it is satisfy all 3 criteria's and has cheap market cost.

Magnetoencephalography is the best option for our project due to fact that it is have amazing time, spatial resolution and it didn't disturb client. Main current disadvantage is size and cost. Potentially, this technology can replace EEG based system for such project like this if advances in technology will decrease size and cost of it

3.1.3 Translation algorithm and operating protocol

1. P300 -

Pros - this system is very well established on the market and a lot of frameworks is already done to use it

Cons - it takes time to learn to use it and can be non-intuitive to use for some users, requires second screen with options to use

2. Steady state visually evoked potential (SSVEP) -

Pros - quite intuitive to use, have several amounts of already exist frameworks.

Cons - requires second screen to use, not suitable for blind people

3. SCP - slow cortical potential:

Pros - requires much less computation power to use, much easier mechanism from algorithmic point of view to extract

Cons - requires long term training to use it, had only 2 potential outputs (positive, negative)

4. Motor Imagery (MI):

Pros - allows the user to multitask, e.g., he/she does not need to focus on the BCI computer and can thus interact with the environment more freely than with methods, area of data extraction is very well localized.

Cons - complexity of computation is much higher, usually motor Imaginary success rate didn't exceed 70-75%

5. Sound Imaginary (SI):

Pros - very natural for all human beings

Cons - extremely complex to extract and achieve high accuracy

3.1.4 Discussion of design choice for operating protocol

Program will require to process at least 7 commands:

1. Go Forward
2. Turn Right
3. Turn Left
4. Turn On Lamp
5. Turn off Lamp
6. Turn off TV
7. Turn ON TV

Therefore, system must be able to differentiate between 7 inputs. Main goal for a project is to simplify end user experience and give maximum freedom for system use

SCP creates only 2 outputs (positive/true or negative/false) and also requires big amount of effort from user to learn how to use it. It is possible to create menu which will be able to process all commands (e.g., positive can represent enter button for a command, bet negative represent movement to next menu position) bet it will be hard to call user friendly.

P300 and SSVEP is very similar technology from a user point of view, both of them will requires second display for input and both of them require focusing on commands on a screen to enter it (P300 – enter by mental counting when commands flashes, SSVEP – enter by focusing gaze on desirable command when it flashes). These 2 methods can be used with quite good success rate to put any number of commands to a system bet input rate will be slow due to fact that user must wait till command will flashes.

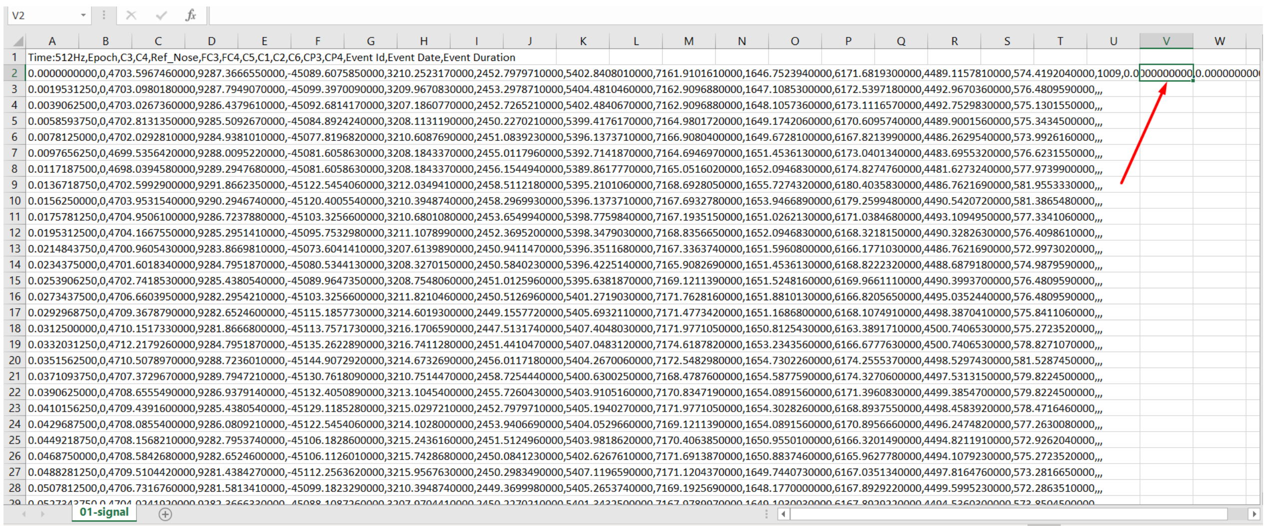
Motor Imagery (MI) and Sound Imaginary (SI)- theoretically can be used to process any number of commands and not require hard trainings from user, can be used with blind people Allows multitasking. Main limitation of this technology is accuracy rate due to fact that requires extraction of data from different brain areas and sound imaginary mechanism is not fully research. These 2 mechanisms are very user friendly if good accuracy is achieved.

The decision was made to use Motor Imaginary and Sound Imaginary due to simplicity for final user and multitasking opportunity which can be crucial for paralyzed user

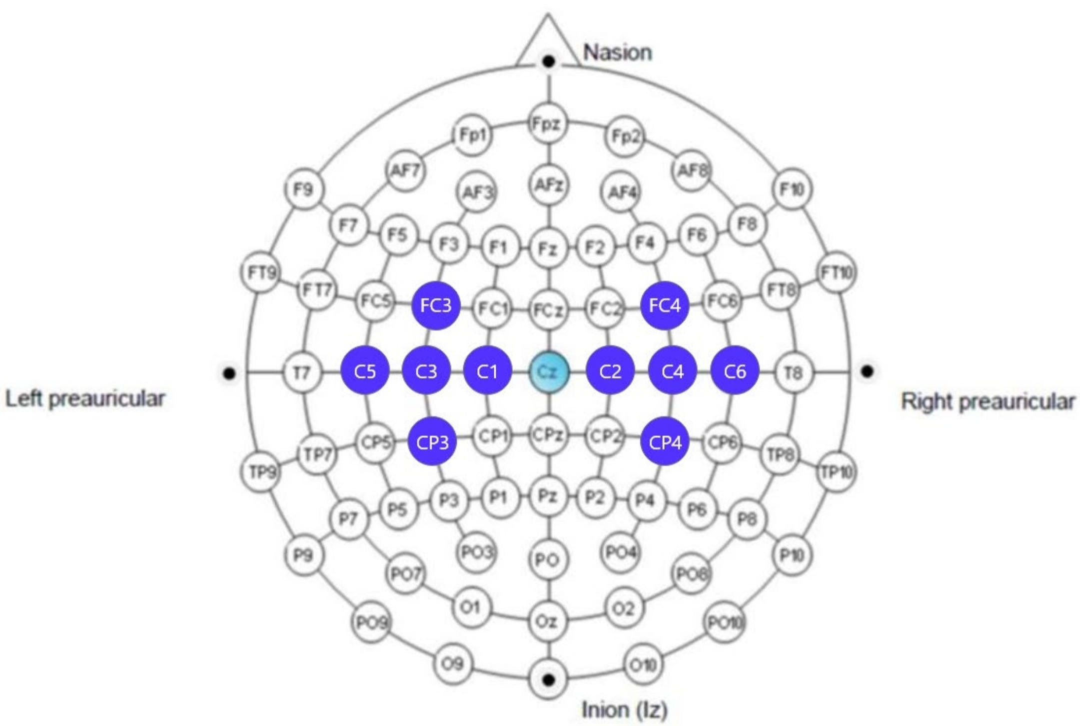
3.2 Data Description

3.2.1 Data description – Data Set 1

First data set was provided by Essex University BCI research lab.



	A
1	Time:512Hz,Epoch,C3,C4,Ref_Nose,FC3,FC4,C5,C1,C2,C6,CP3,CP4,Event Id,Event Date,Event Duration
2	0.0000000000,0,4703.5967460000,9287.3666550000,-45089.6075850000,3210.2523170000,2452.7979710000,5402.8408010000,7161.9101610000,1646.7523940000,6171.6819300000,4489.1157810000,574.4192040000,1009,0.0000000000,0.000000000
3	0.0019531250,0,4703.0980180000,9287.7949070000,-45099.3970090000,3209.9670830000,2453.2978710000,5404.4810460000,7162.9096880000,1647.1085300000,6172.5397180000,4492.9670360000,576.4809590000,,,
4	0.0039062500,0,4703.0267360000,9286.4379610000,-45092.6814170000,3207.1860770000,2452.7265210000,5402.4840670000,7162.9096880000,1648.1057360000,6173.1116570000,4492.7529830000,575.1301550000,,,
5	0.0058593750,0,4702.8131350000,9285.5092670000,-45084.8924240000,3208.1131190000,2450.2270210000,5399.4176170000,7164.9801720000,1649.1742060000,6170.6095740000,4489.9001560000,575.3434500000,,,
6	0.0078125000,0,4702.0292810000,9284.9381010000,-45077.8196820000,3210.6087690000,2451.0839230000,5396.1373710000,7166.9080400000,1649.6728100000,6167.8213990000,4486.2629540000,573.9926160000,,,
7	0.0097656250,0,4699.5356420000,9288.0095220000,-45081.6058630000,3208.1843370000,2455.0117960000,5392.7141870000,7164.6946970000,1651.4536130000,6173.0401340000,4483.6955320000,576.6231550000,,,
8	0.0117187500,0,4698.0394580000,9289.2947680000,-45081.6058630000,3208.1843370000,2456.1544940000,5389.8617770000,7165.0516020000,1652.0946830000,6174.8274760000,4481.6273240000,577.9739900000,,,
9	0.0136718750,0,4702.5992900000,9291.8662350000,-45122.5454060000,3212.0349410000,2458.5112180000,5395.2101060000,7168.6928050000,1655.7274320000,6180.4035830000,4486.7621690000,581.9553330000,,,
10	0.0156250000,0,4703.9531540000,9290.2946740000,-45120.4005540000,3210.3948740000,2458.2969930000,5396.1373710000,7167.6932780000,1653.9466890000,6179.2599480000,4490.5420720000,581.3865480000,,,
11	0.0175781250,0,4704.9506100000,9286.7237880000,-45103.3256600000,3210.6801080000,2453.6549940000,5398.7759840000,7167.1935150000,1651.0262130000,6171.0384680000,4493.1094950000,577.3341060000,,,
12	0.0195312500,0,4704.1667550000,9285.2951410000,-45095.7532980000,3211.1078990000,2452.3695200000,5398.3479030000,7168.8356650000,1652.0946830000,6168.3218150000,4490.3282630000,576.4098610000,,,
13	0.0214843750,0,4700.9605430000,9283.8669810000,-45073.6041410000,3207.6139890000,2450.941147000,5396.3511680000,7167.3363740000,1651.5960800000,6166.1771030000,4486.7621690000,572.9973020000,,,
14	0.0234375000,0,4701.6018340000,9284.7951870000,-45080.5344130000,3208.3270150000,2450.5840230000,5396.4225140000,7165.9082690000,1651.4536130000,6168.8222320000,4488.6879180000,574.9879590000,,,
15	0.0253906250,0,4702.7418530000,9285.4380540000,-45089.9647350000,3208.7548060000,2451.0125960000,5395.6381870000,7169.1211390000,1651.5248160000,6169.9661110000,4490.3993700000,576.4809590000,,,
16	0.0273437500,0,4706.6603950000,9282.2954210000,-45103.3256600000,3211.8210460000,2450.5126960000,5401.2719030000,7171.7628160000,1651.8810130000,6166.8205650000,4495.0352440000,576.4809590000,,,
17	0.0292968750,0,4709.3678790000,9282.6524600000,-45115.1857730000,3214.6019300000,2449.1557720000,5405.6932110000,7171.4773420000,1651.1686800000,6168.1074910000,4498.3870410000,575.8411060000,,,
18	0.0312500000,0,4710.1517330000,9281.8666800000,-45113.7571730000,3216.1706590000,2447.5131740000,5407.4048030000,7171.9771050000,1650.8125430000,6163.3891710000,4500.7406530000,575.2723520000,,,
19	0.0332031250,0,4712.2179260000,9284.7951870000,-45135.2622890000,3216.7411280000,2451.4410470000,5407.0483120000,7174.6187820000,1653.2343560000,6166.6777630000,4500.7406530000,578.8271070000,,,
20	0.0351562500,0,4710.5078970000,9288.7236010000,-45144.9072920000,3214.6732690000,2456.0117180000,5404.2670060000,7172.5482980000,1654.7302260000,6174.2555370000,4498.5297430000,581.5287450000,,,
21	0.0371093750,0,4707.3729670000,9289.7947210000,-45130.7618090000,3210.7514470000,2458.7254440000,5400.6300250000,7168.4787600000,1654.5877590000,6174.3270600000,4497.5313150000,579.8224500000,,,
22	0.0390625000,0,4708.6555490000,9286.9379140000,-45132.4050890000,3213.1045400000,2455.7260430000,5403.9105160000,7170.8347190000,1654.0891560000,6171.3960830000,4499.3854700000,579.8224500000,,,
23	0.0410156250,0,4709.4391600000,9285.4380540000,-45129.1185280000,3215.0297210000,2452.7979710000,5405.1940270000,7171.9771050000,1654.3028260000,6168.8937550000,4498.4583920000,578.4716460000,,,
24	0.0429687500,0,4708.0855400000,9286.0809210000,-45122.5454060000,3214.1028000000,2453.9406690000,5404.0529660000,7169.1211390000,1654.0891560000,6170.8956660000,4496.2474820000,577.2630080000,,,
25	0.0449218750,0,4708.1568210000,9282.7953740000,-45106.1828600000,3215.2436160000,2451.5124960000,5403.9818620000,7170.4063850000,1650.9550100000,6166.3201490000,4494.8211910000,572.9262040000,,,
26	0.0468750000,0,4708.5842680000,9282.6524600000,-45106.1126010000,3215.7428680000,2450.0841230000,5402.6267610000,7171.6913870000,1650.8837460000,6165.9627780000,4494.1079230000,575.2723520000,,,
27	0.0488281250,0,4709.5104420000,9281.4384270000,-45112.2563620000,3215.9567630000,2450.2983490000,5407.1196590000,7171.1204370000,1649.7440730000,6167.0351340000,4497.8164760000,573.2816650000,,,
28	0.0507812500,0,4706.7316760000,9281.5813410000,-45099.1823290000,3210.3948740000,2449.3699980000,5405.2653740000,7169.1925690000,1648.1770000000,6167.8929220000,4499.5995230000,572.2863510000,,,



It is CSV file which consist of:

- Time (512Hz)
- Epoch
- Reference Channel

Was recorded 8 brain areas, it is:

- C5
- C3
- C1
- CP3
- FC3
- C2
- C4
- C6
- FC4
- CP4

This data set was made to record Motor Imaginary part of brain for further processing. Participants was asked to imagine their right- and left-hand movement. Time slots of trial was marked in a data set.

3.2.2 Data description – Data Set 2

Second data set was provided by Berlin Brain Computer Interface.

It consists of 4-class EEG data. it is:

- left hand – label 1
- right hand – label 2
- foot – label 3
- tongue – label 4

The task was to perform imagery left hand, right hand, foot or tongue movements according to a cue. The order of cues was random. The experiment consists of several runs (at least 6) with 40 trials each; after trial begin, the first 2s were quite, at t=2s an acoustic stimulus indicated the beginning of the trial, and a cross "+" is displayed; then from t=3s an arrow to the left, right, up or down was displayed for 1 s; at the same time the subject was asked to imagine a left hand, right hand, tongue or foot movement, respectively, until the cross disappeared at t=7s. Each of the 4 cues was displayed 10 times within each run in a randomized order

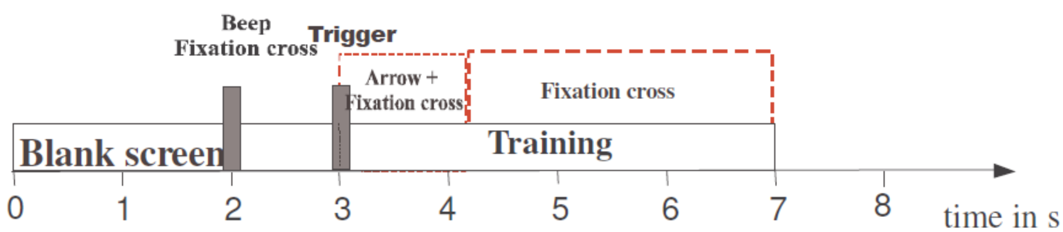


Figure 15 Data collection timeline

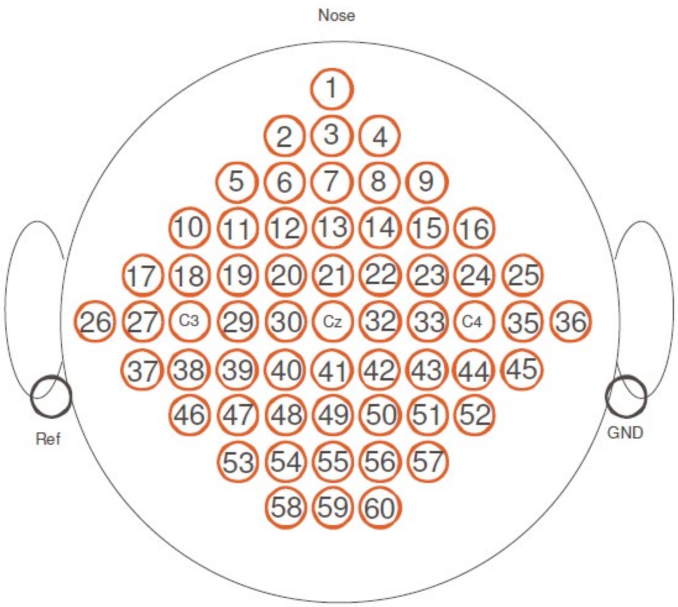


Figure 16 Position of EEG electrodes

1x1 struct with 41 fields

Field	Value
SPR	1
NRec	986780
SampleRate	250
FLAG	1x1 struct
EVENT	1x1 struct
Label	60x1 cell
LeadIdCode	1x60 double
PhysDimCode	1x60 double
PhysDim	60x1 cell
Filter	1x1 struct
PhysMax	1x60 double
PhysMin	1x60 double
DigMax	1x60 double
DigMin	1x60 double
Transducer	60x1 cell
Cal	1x60 double
Off	1x60 double
GDFTYP	1x60 double
TOffset	1x60 double
ELEC	1x1 struct
Impedance	1x60 double
fZ	1x60 double
AS	1x1 struct
Dur	0.0040
REC	1x1 struct
Manufacturer	1x1 struct
InChanSelect	1x60 double
Calib	61x60 double
FILE	1x1 struct
Classlabel	360x1 double

Figure 17 GDF file parameters

3.3 Implementation – creation of simulation area

The Unity3D engine makes it possible to develop games and simulation environments. It uses a component-oriented approach, in which the developer creates objects (for example, the main character) and adds various components to them (for example, the visual display of the character and ways to control it).

The second advantage of the engine is the presence of a huge library of assets and plugins, with which you can significantly speed up the game development process. They can be imported and exported, whole blanks can be added to the game - levels, enemies, AI behaviour patterns, and so on.

First step of creation was to develop virtual room from 3D components which unity provides.

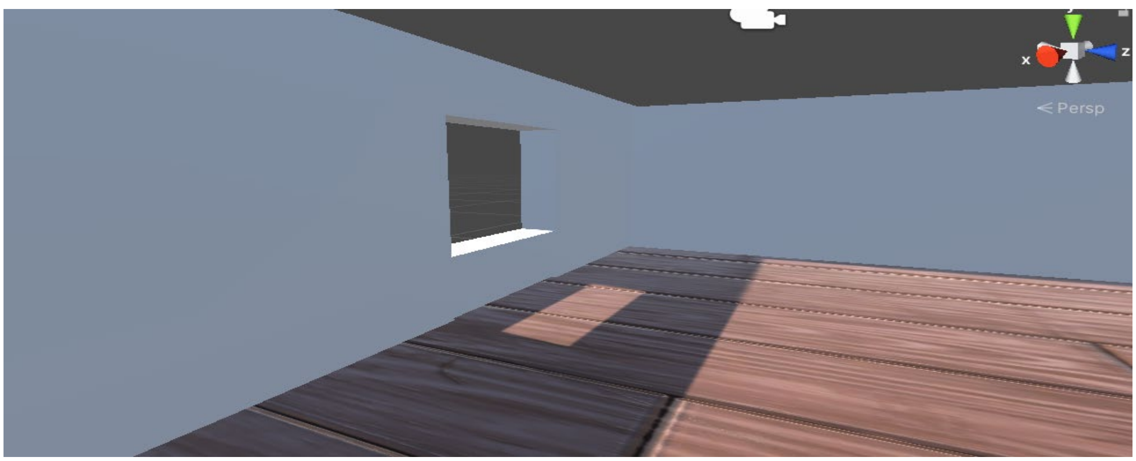


Figure 18 Room in Unity, starting state

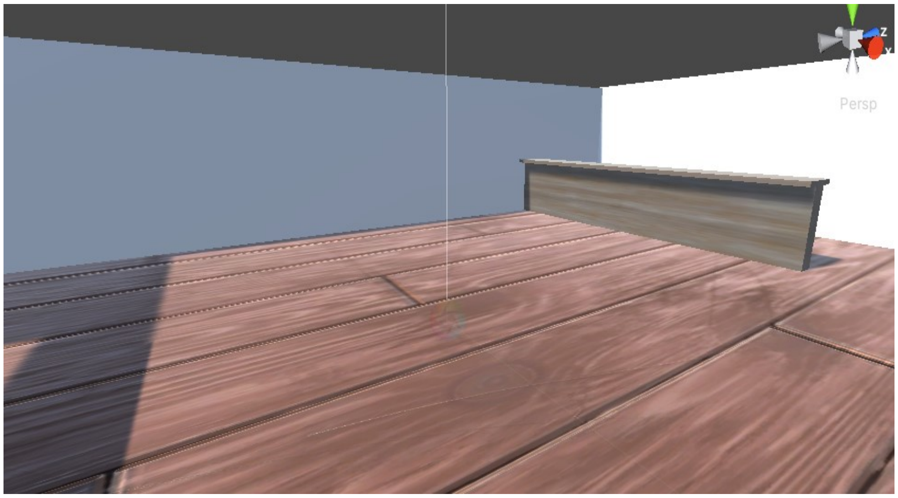


Figure 19 Room in Unity, starting state

Second stage was to create decoration for a room. The basic idea is that this room represent studio flat of 1 person. There were 2 options to solve this problem:

- Option 1: Create assets personally in 3D modeling programs (e.g., Autodesk 3ds Max, Blender)
- Option 2: Find created assets in Unity Shop

Option 2 was chosen due to time limitation of a project. All assets in a project were taken from Unity Assets Store with Free license to use.

Each asset consists of several components, it is:

- Texture of material - this is a regular image that is superimposed on the surface of the finished model to give it color, properties and relief.
- Material - is a file that contains information about the illumination of an object by this material (texture)
- 3D model of an object – it is model of simulated object which can consist of several objects.

By combining different assets room was decorated with kitchen, TV, sofa and etc.

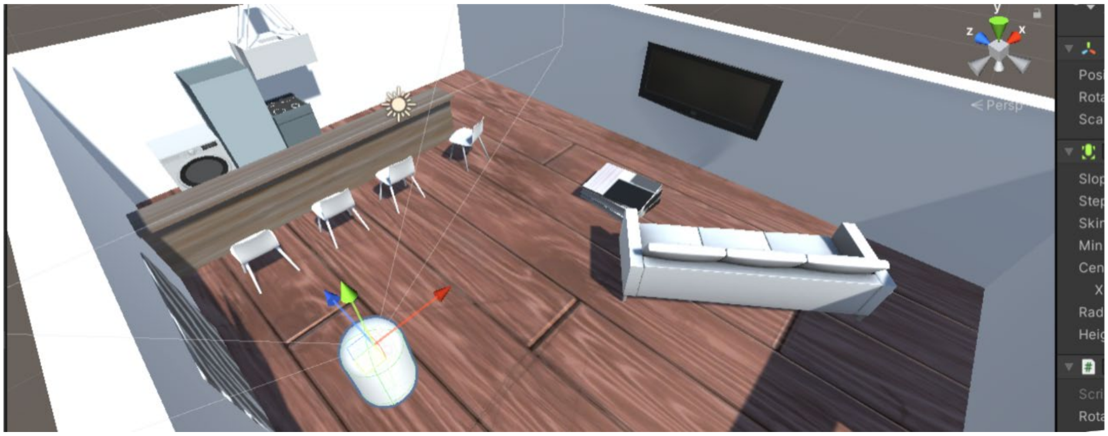


Figure 20 Room after decoration

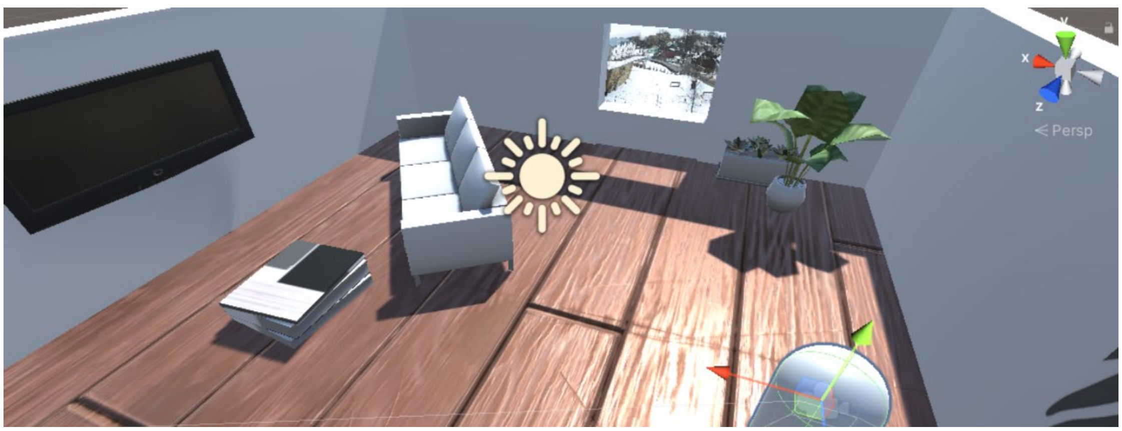


Figure 21 Room after decoration

Third stage was to create first person character which will be able to move and interact with environment. I was used cylinder with camera attached to it due to fact that character will see everything from first person perspective (therefore client/player will never see himself). All code was written on C# language.

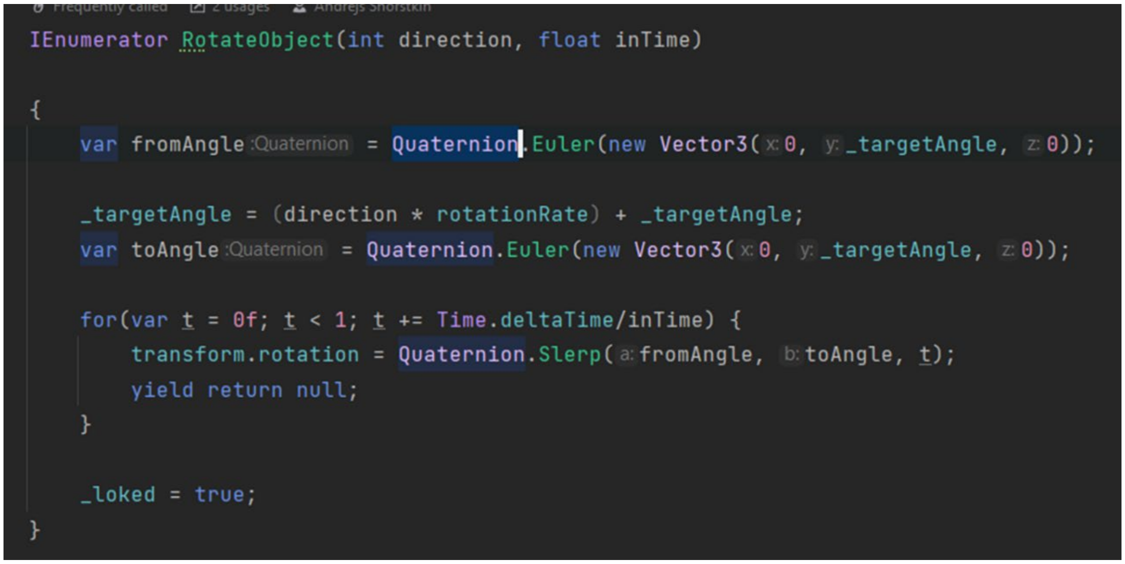
First iteration of movement system was simple transition of an object towards X, Y, Z axes.

```csharp
public float rotationRate = 360;
public float moveSpeed = 0.01f;
AudioSource audioSource;

private void Start()
{
    audioSource = GetComponent<AudioSource>();
}

// Update is called once per frame
void Update()
{
    float moveAxis = Input.GetAxis("Vertical");
    float rotateAxis = Input.GetAxis("Horizontal");

    ApplyInput(moveAxis, rotateAxis);
}


private void ApplyInput(float moveInput, float rotateInput)
{
    transform.Translate(Vector3.forward * moveInput * moveSpeed);
    transform.Rotate(0, rotateInput * rotationRate * Time.deltaTime, 0);
}
```

Figure 22 Code of first iteration of a movement system

Next stage of development was to adapt and simplify movement system for use in perspective of BCI system. There were 2 main components:

1. Simplify forward movement – make character to teleport on n number of meters forward after input command.
2. Make smooth rotation – make animation of rotation by using quaternions



```
IEnumerator RotateObject(int direction, float inTime)

{
    var fromAngle = Quaternion.Euler(new Vector3(0, _targetAngle, 0));

    _targetAngle = (direction * rotationRate) + _targetAngle;
    var toAngle = Quaternion.Euler(new Vector3(0, _targetAngle, 0));

    for(var t = 0f; t < 1; t += Time.deltaTime/inTime) {
        transform.rotation = Quaternion.Slerp(fromAngle, toAngle, t);
        yield return null;
    }

    _loked = true;
}
```

Figure 23 Code of rotation animation

Fourth stage was to implement lighting system for a room.

There are 3 lighting areas in a room, it is 2 wall lamps and sun lighting outside the room. Special material was created for a light which is coming from lamps. Sun lighting was made by using default settings of Directional Light component

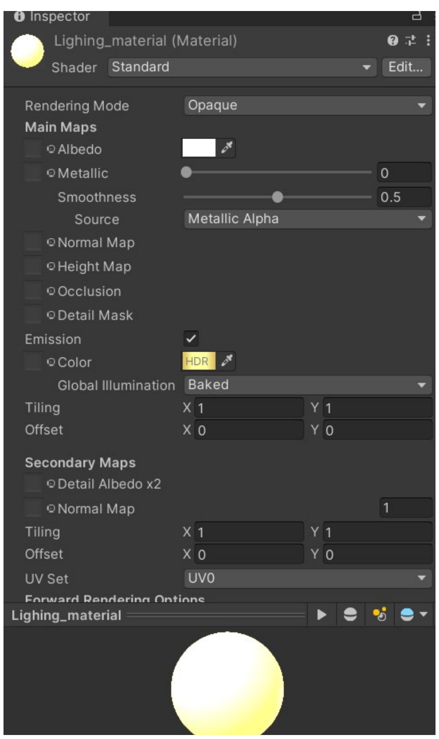


Figure 26 Material for Light

Figure 24 Lamp 1

Figure 25 Sun Light

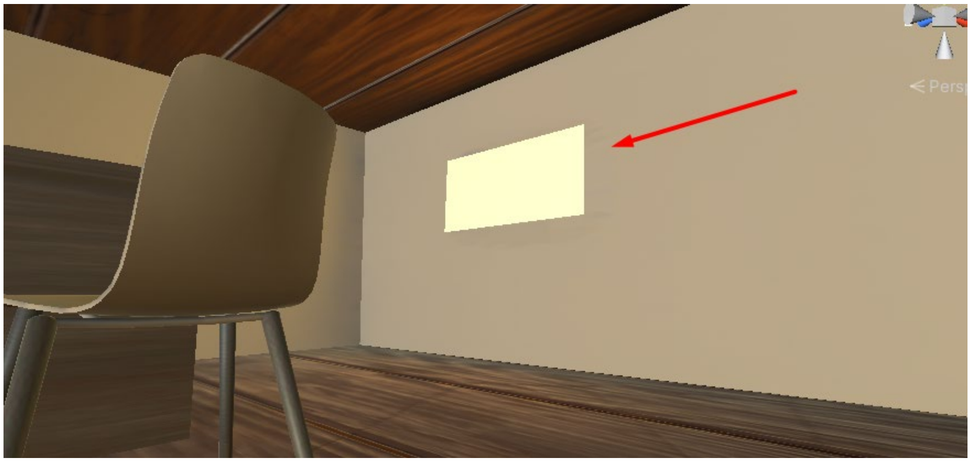


Figure 27 Lamp 3

There are 3 main algorithms for calculating lighting system in Unity:

- Light Mode: Realtime
- Light Mode: Mixed
- Light Mode: Baked

I was using baked mode in calculating it. Baked mode calculate light by using process called "Lightmapping". Lightmapping is the process of pre-calculating the brightness of surfaces in the Scene. It stores the information it calculates in a chart or lightmap for later use. Lightmaps allow to add global illumination, shadows, and ambient lighting at a relatively low computational cost. [4]. Lightmapping is very good for our simulation because simulation consist of very big number of static objects (Only character is moving in a scene)

Fifth Stage was to create main menu and option menu.

Before making main menu, was done splitting game area on several levels.

Levels:

- Level 0: Main Menu
- Level 1: Manual Control

- Level 2: Control by using 2 class EEG (right and left arm)
- Level 3: Control by using 4 class EEG (left, right hand, foot and tongue)

First Iteration of Main Menu consisted of 4 buttons:

- Start Simulation – lead to Level 2: Control by using 2 class EEG (right and left arm)
- Manual Control – lead to Level 1: Manual Control
- Option Menu – lead to option menu (holds 1 option parameter which configure LSL connection between OpenVibe and Unity)
- Exit – terminate program

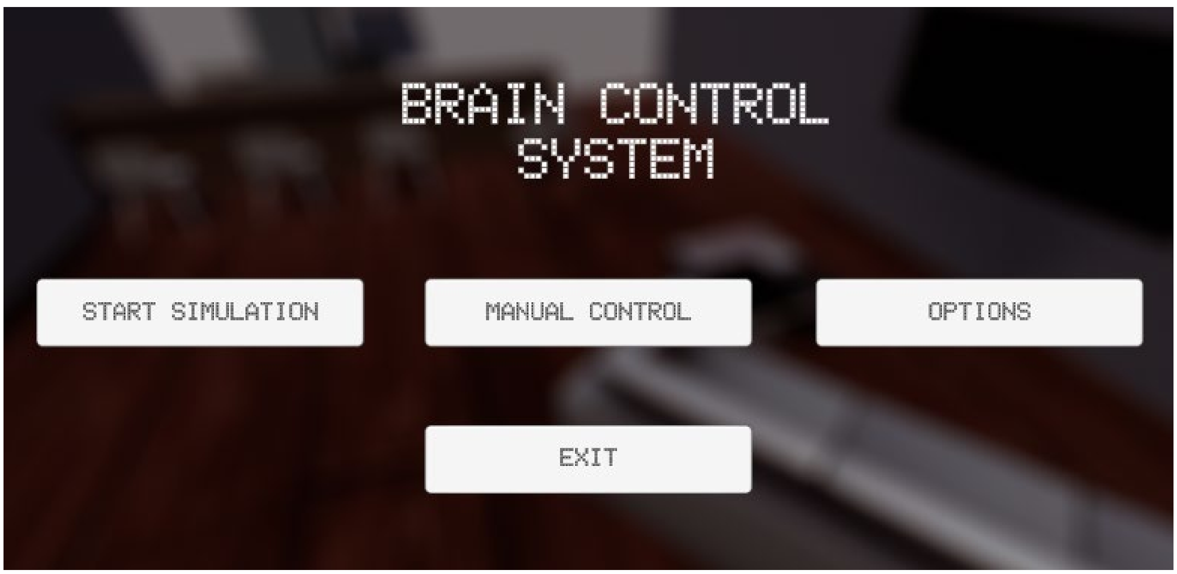


Figure 24 Program Main Menu

Second iteration of main menu consist of 3 different levels and 3 buttons which lead to these levels.

- Start Simulation class 2 – lead to Level 2: Control by using 2 class EEG (right and left arm)
- Start Simulation class 4 – lead to Level 3: Control by using 4 class EEG (right and left arm, tongue, foot)
- Manual Control – lead to Level 1: Manual Control

- Option Menu – lead to option menu (holds 1 option parameter which configure LSL connection between OpenVibe and Unity)
- Exit – terminate program

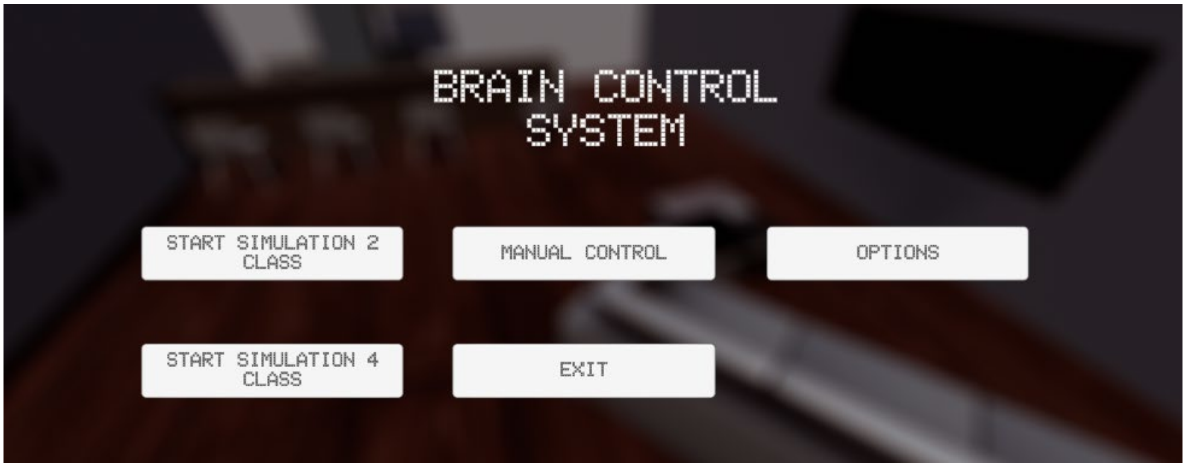


Figure 25 Second Iteration of Main Menu

Option Menu consist of information for setup LSL input stream with OpenVibe, it is:

- Input field for LSL identification name
- Input field for LSL data type

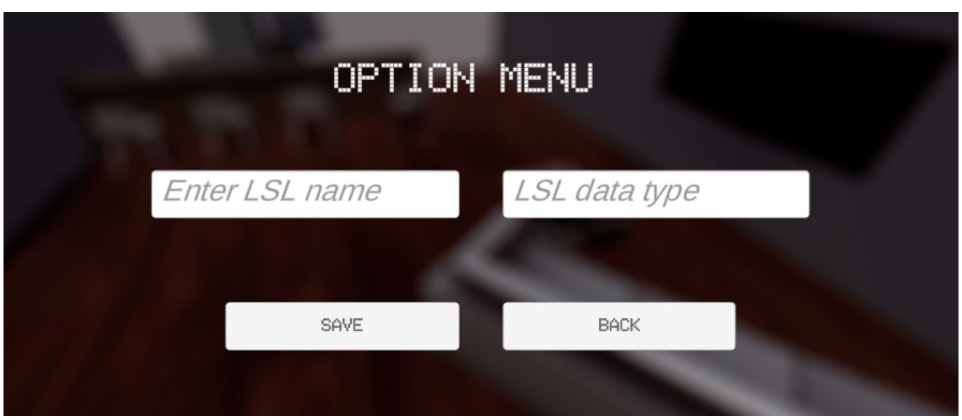


Figure 26 Option Menu

Sixth stage of implementation was to add TV on/off functionality. I created special material for "TV on" state and add it to TV plane

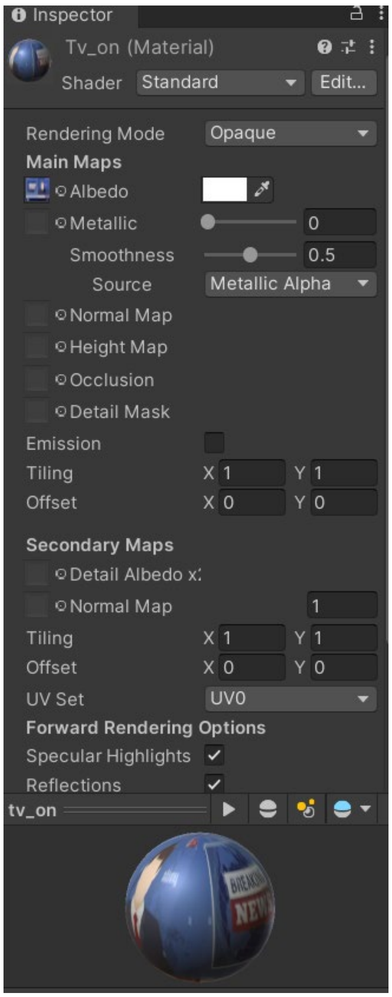


Figure 28 Material Setup

Figure 27 Photo and ready material

Essentially, the algorithm of TV and lamp is very simple, it just enables or disables 3 objects (TV, lamp 1, lamp3) simultaneously after Boolean command input to a system.

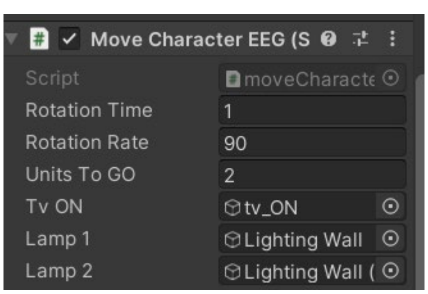


Figure 29 Unity Object Setup

```
private void electricalDevicesONOFF(Boolean pressQ)
{
    if (_loked == true && pressQ)
    {
        _loked = false;
        if (this.tvON.activeSelf)
        {
            this.tvON.SetActive(false);
            this.lamp1.SetActive(false);
            this.lamp2.SetActive(false);
            _loked = true;
        }
        else
        {
            this.tvON.SetActive(true);
            this.lamp1.SetActive(true);
            this.lamp2.SetActive(true);
            _loked = true;
        }

    }
}
```

Figure 32 Code Implementation of TV and lamp

Figure 31 TV On

Figure 30 TV Off

3.4 Implementation – creation of BCI algorithm (2 classes)

All work on BCI side was done by using OpenVibe program for data extraction, pre and post processing and algorithm training. OpenVibe give graphical interface with most popular tools, mathematical models and functions used in creation and visualization of data and algorithms used in BCI field.

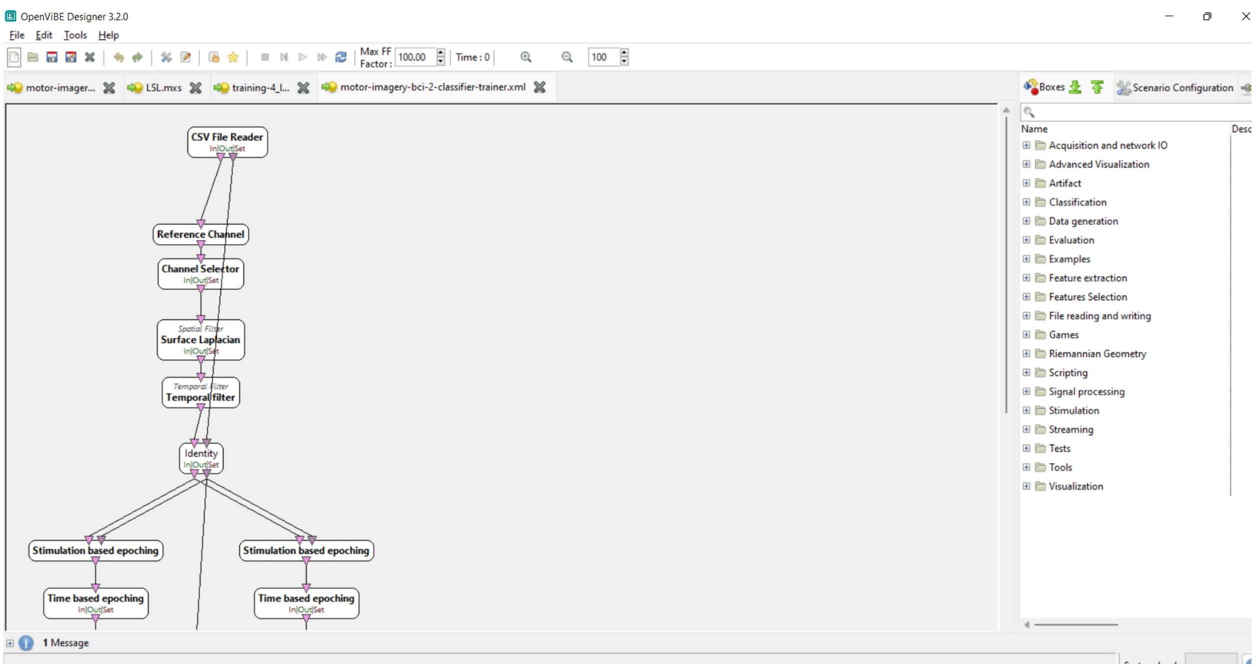
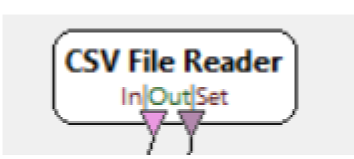


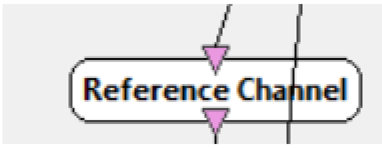
Figure 33 OpenVibe Interface

3.4.1 Algorithm training

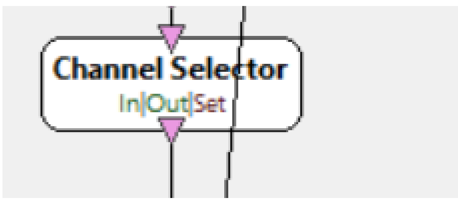
Step 1: Loading data from CSV to OpenVibe. Data is loaded from CSV file to OpenVibe part of ram.



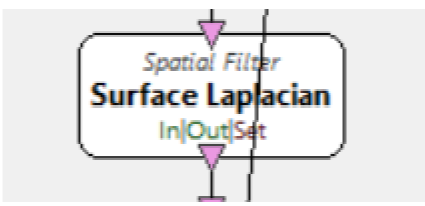
Step 2: Reference channel subtraction. Subtraction of signal from all channels by reference channel is happening to decrease amount of noise created by environment in which experiment is hold.



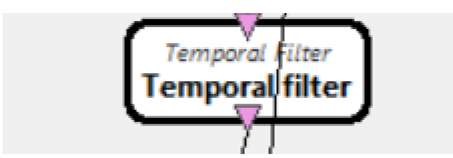
Step 3: Channel Selector, channels are selected according to it neuro physiological anatomy. Selected channels are C5, C3, C1, FC3, CP3, C2, C4, C6, FC4, CP4) due to fact that they correlate with motor cortex



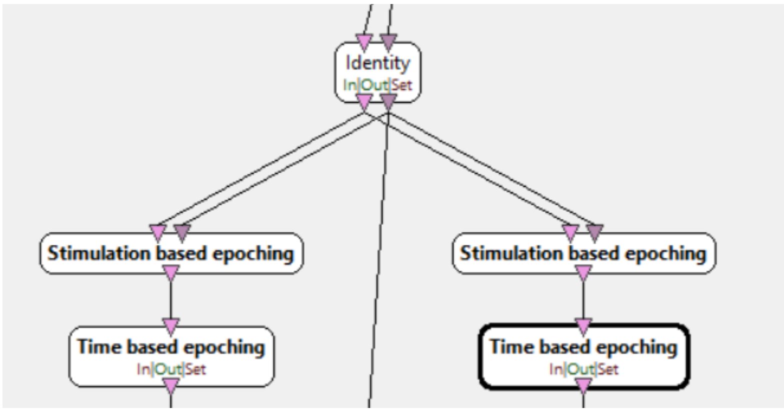
Step 4: Surface Laplacian, Surface Laplacian is applied to decrease signal noise ratio among taken channels (add more about it …..)



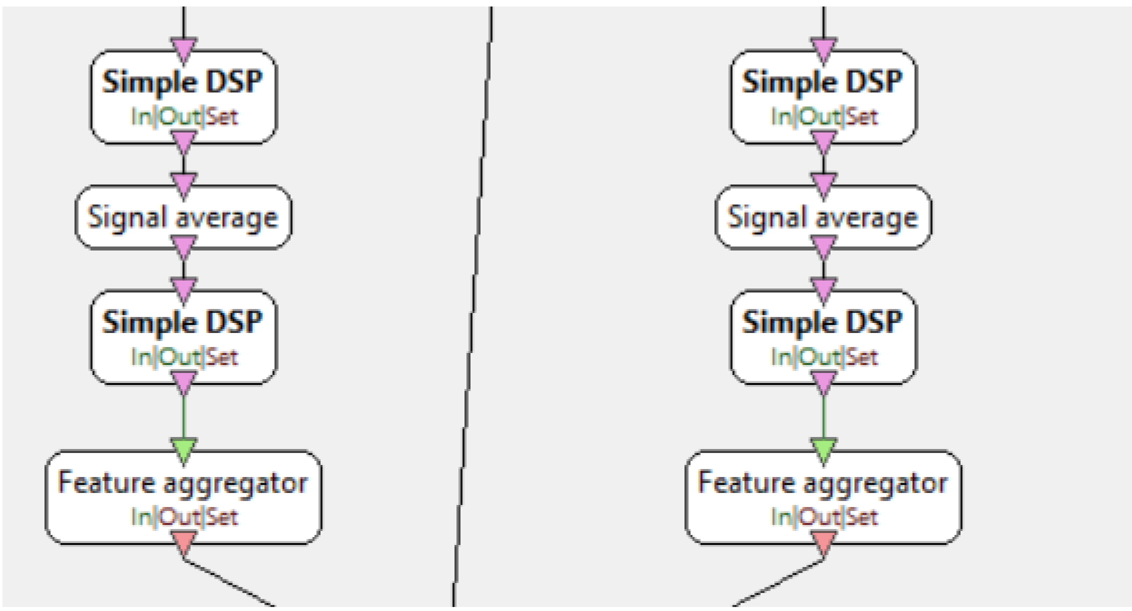
Step 5: Temporal Filter, temporal filtering aims to remove or attenuate frequencies within the raw signal, that are not of interest. Removal of frequencies is happening in 8 to 24 Hz amplitude because this frequency is highly correlate with signal taken from skin muscles and vines on a head.



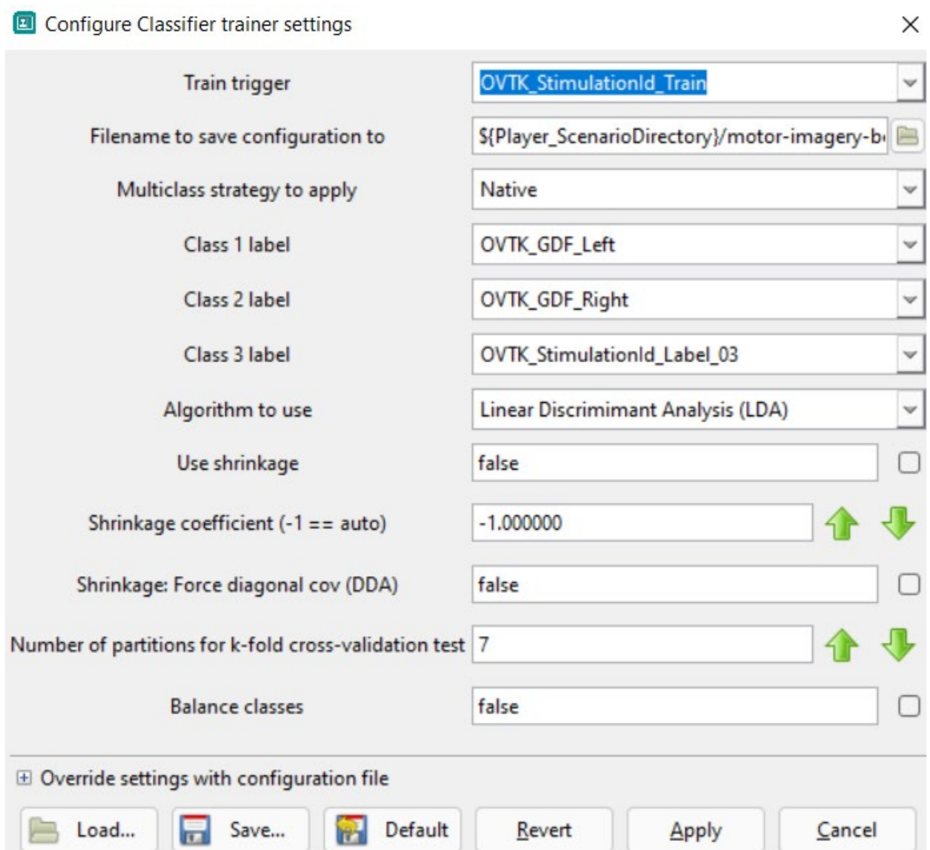
Step 6: Feature Extraction for Learning, data inside CSV file mark according to starting and finishing one of the events (right and left hand). OpenVibe extract this marked signal frame and split into equal time frames for learning algorithm.



Step 7: Mathematical signal compression applied to shrink signal, X^2 and log(1+x)



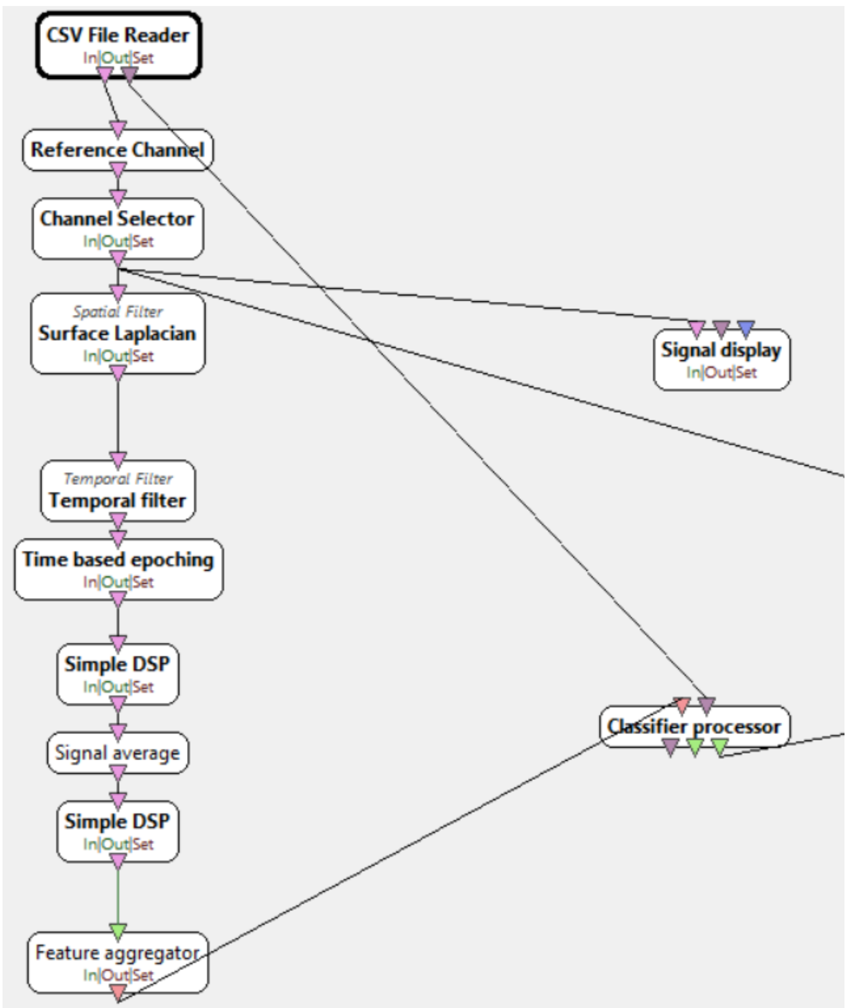
Step 8: Classifier learning to classify data from given signal frames.



It was chosen to use Linear Discriminant Analysis (LDA) algorithm for data classification. It is one of the simplest algorithms used in BCI area, bet one of the most efficient and popular due to its simplicity to process data. Classifier is producing CFG file which hold information for classification after training. This CFG file is used in final application.

3.4.2 Algorithm Use

Classification of signal if very similar to training due to fact that we are working with LIVE recording data from a brain. Steps 1 to 7 is the same as in training of algorithm (1. Loading data from equipment EEG, subtracting reference channel, Channel Selector, Laplacian Filter, Temporal Filter, time based epoching, mathematical shrinkage of a signal)



Classifier is producing probabilities of event occurred (Left or Right)

3.5 Implementation – creation of BCI algorithm (4 classes)

3.5.1 GDF and MATLAB

Evolution of a system from BCI point of view request to use completely different data format for data extraction GENERAL DATAFORMAT FOR BIOSIGNALS (GDF)

Biomedical signals are stored in many different data formats. Most formats have been developed for a specific purpose of a specialized community for ECG research, EEG analysis, sleep research, etc. So far none of the existing formats can be considered a general-purpose data format for biomedical signals. In order to solve this problem and to unify the various needs of the various biomedical signal processing fields, the so-called General Data Format for biomedical signals" (GDF) is developed. This GDF format is fully described and specified. Software for reading and writing GDF data is implemented in Octave/MATLAB and C/C++ and provided through BioSig – a free and open-source software library for biomedical signal processing. BioSig provides also converters from various data formats to GDF, and a viewing and scoring software. [6]

This data format is storing complete information about patients and a lot of other parameters. First step was to open it to read naming of electrodes attached to patient, sample per buffer rate and other information like reference channel.

MATLAB was used to open GDF to extract information. It is needed to install BioSig library to MATLAB to do it.

BioSig is an open-source software library for biomedical signal processing, featuring for example the analysis of bio signals such as the electroencephalogram (EEG), electrocorticogram (ECoG), electrocardiogram (ECG), electrooculogram (EOG), electromyogram (EMG) [7]

```
[s,hd]=sload('filename.gdf');

display(hd.Label) % to see the channel labels

% display(hd.EVENT.TYP) % to see the types of event markers in the
data

% display(hd.EVENT.POS) % to see the data point numbers where event
markers took place

% plot(s(:,1:3)) % plot the first three channels of EEG data
```

This code was used from BioSig library to open GDF file in MATLAB

Field	Value
SPR	1
NRec	986780
SampleRate	250
FLAG	1x1 struct
EVENT	1x1 struct
Label	60x1 cell
LeadIdCode	1x60 double
PhysDimCode	1x60 double
PhysDim	60x1 cell
Filter	1x1 struct
PhysMax	1x60 double
PhysMin	1x60 double
DigMax	1x60 double
DigMin	1x60 double
Transducer	60x1 cell
Cal	1x60 double
Off	1x60 double
GDFTYP	1x60 double
TOffset	1x60 double
ELEC	1x1 struct
Impedance	1x60 double
fZ	1x60 double
AS	1x1 struct
Dur	0.0040
REC	1x1 struct
Manufacturer	1x1 struct
InChanSelect	1x60 double
Calib	61x60 double
FILE	1x1 struct
Classlabel	360x1 double

Figure 34 GDG file in MATLAB. Highlight main parameter inside of it

3.5.2 Algorithm training

Main algorithm for the training was the same for new data set bet it needed to be adapted it for new data set and 4 classes

Step 1: Change Sample Rate

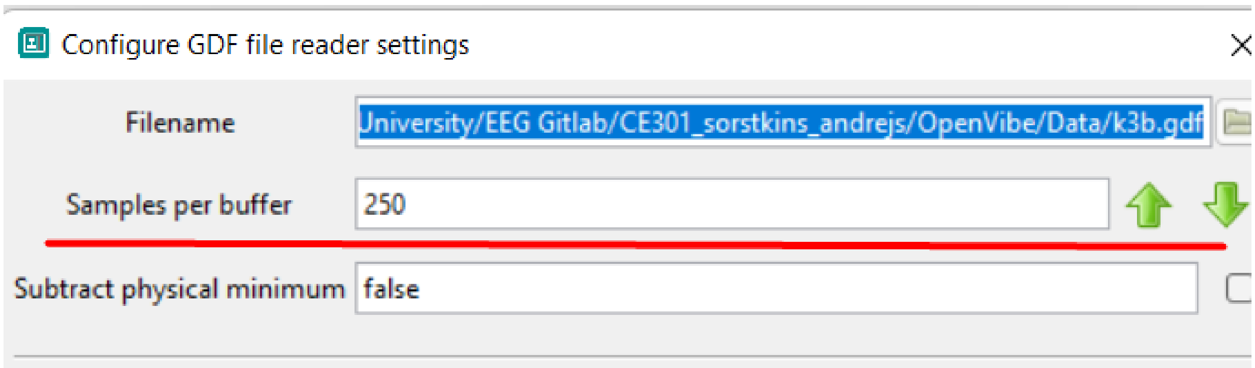


Figure 35 GDf configuration

While CSV file was used it was no needed to specify sample rate because program was just reading file line by line, bet GDF file give access to 60 electrodes and sample rate is crucial for optimal rate of program work.

Step 2: Specify channels on channel selector

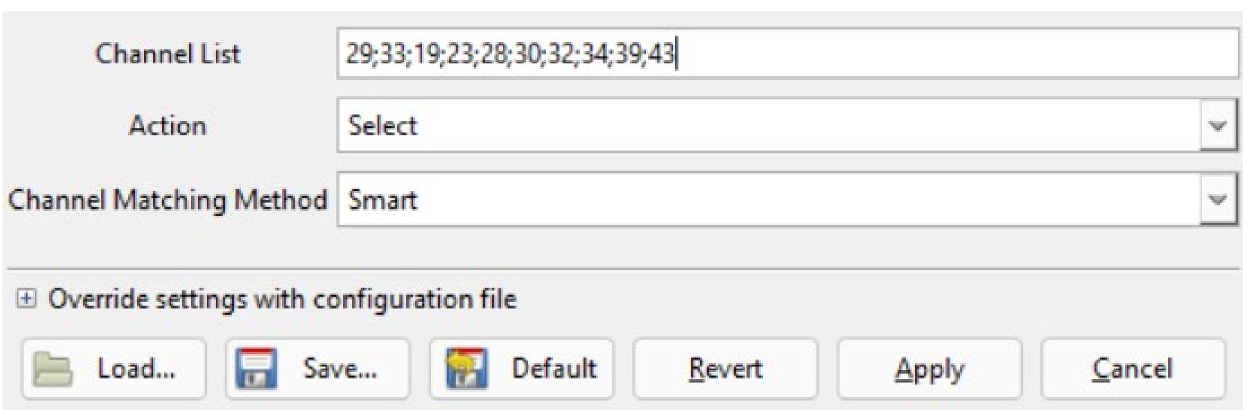


Figure 36 Channel Selector Configuration

Step 3: Add 2 more classes to Stimulation Based Epoching

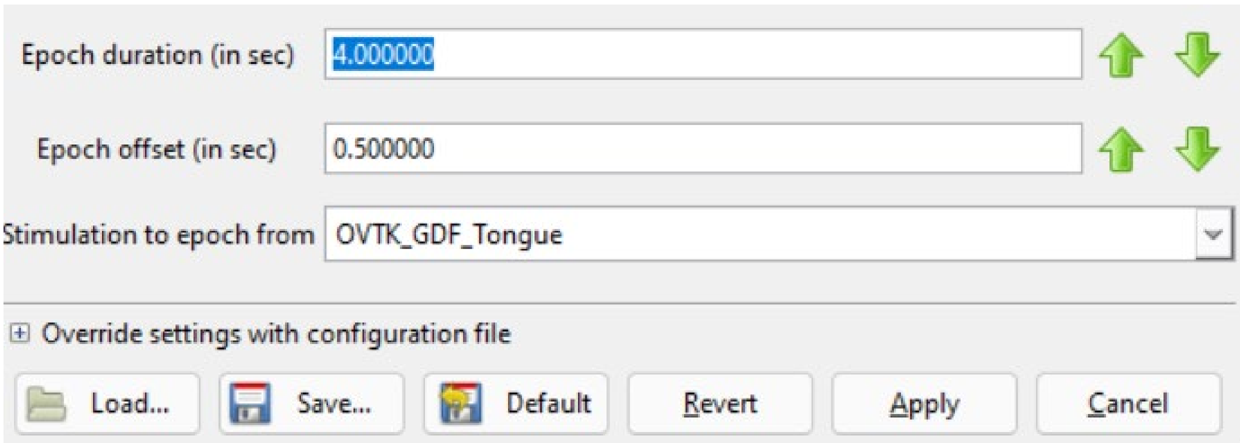


Figure 37 Stimulation Based Epoching Configuration, Tongue

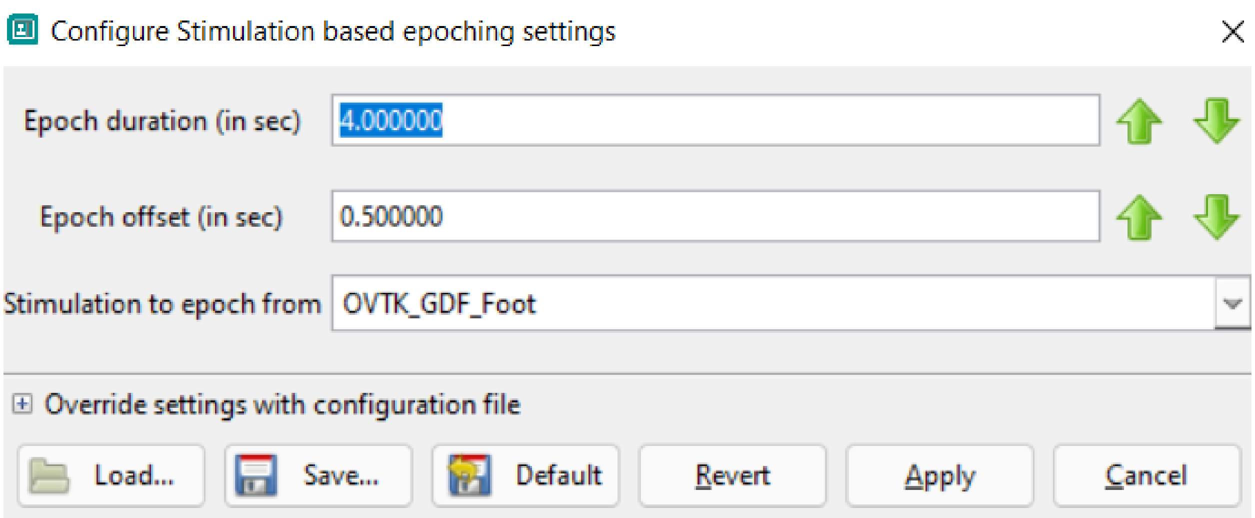


Figure 38 Stimulation Based Epoching Configuration, Foot

Step 4: Extrapolate learning algorithm

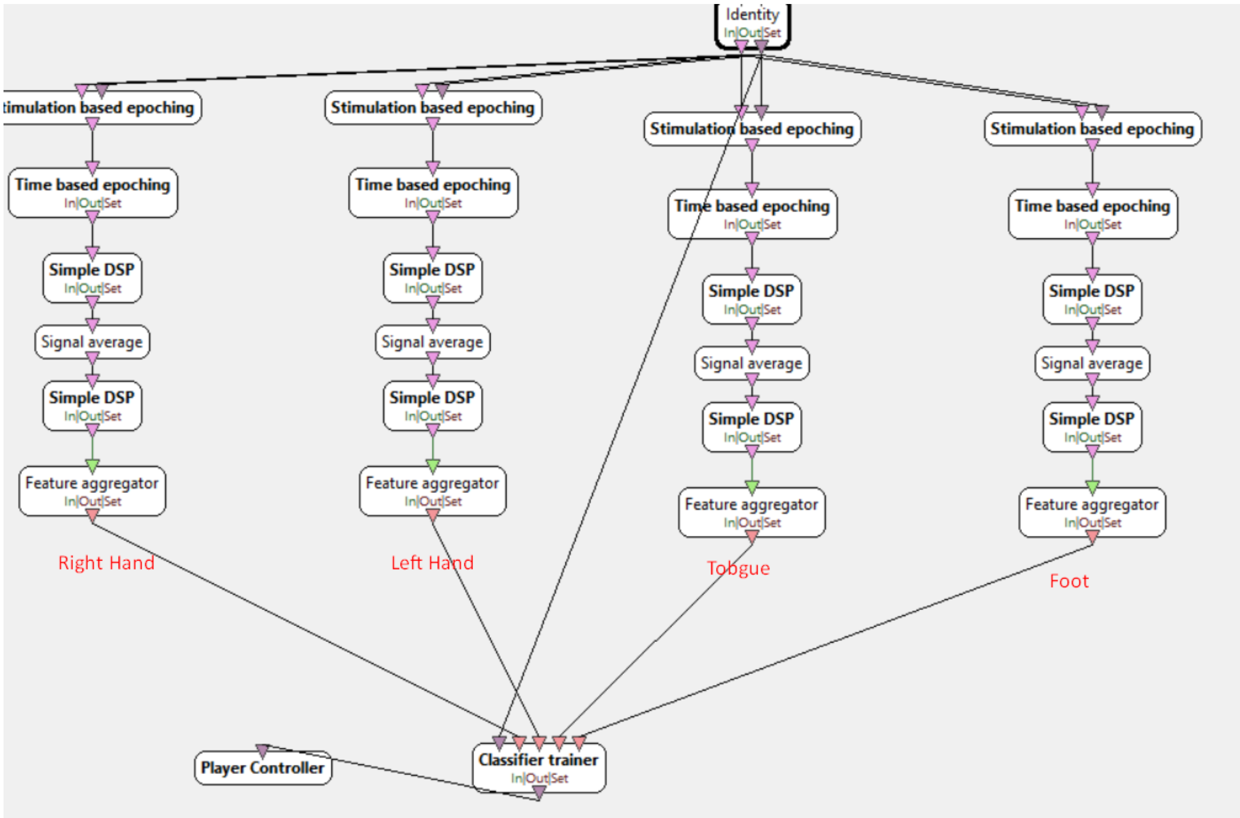


Figure 39 Learning Algorithm for 2 data set view in OpenVibe text editor

Step 4: Add 2 more classes to LDA

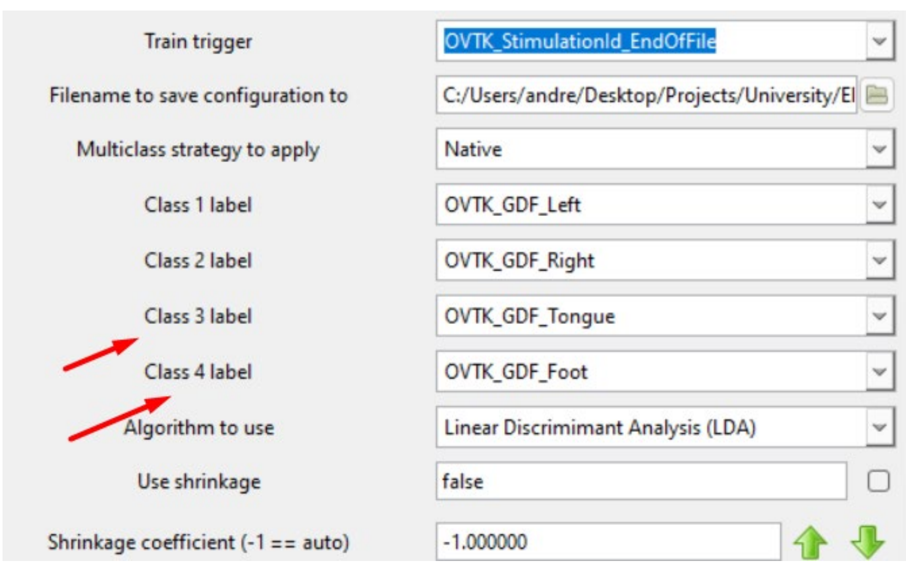


Figure 40 LDA configuration

3.5.3 Algorithm use

Algorithm use was exactly like in first data set. Configuration file for LDA was changed on trained one from 4 classes classifier

3.6 Implementation – interconnection of both systems

The lab streaming layer (LSL) is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access as well as optionally the centralized collection, viewing and disk recording of the data. [5]

LSL4Unity library was used from GitHub to interconnect OpenVibe and Unity.

```
namespace Assets.LSL4Unity.Scripts.Examples
{
    /// <summary>
    /// Just an example implementation for a Inlet recieving float values
    /// </summary>
    public class ExampleFloatInlet : AFloatInlet
    {
        public string lastSample = String.Empty;

        protected override void Process(float[] newSample, double timeStamp)
        {
            // just as an example, make a string out of all channel values of this sample
            lastSample = string.Join(" ", newSample.Select(c => c.ToString()).ToArray());

            // Debug.Log(
            //     string.Format("Got {0} samples at {1}", newSample.Length, timeStamp)
            //     );
        }
    }
}
```

Figure 41 Part of LSL library used

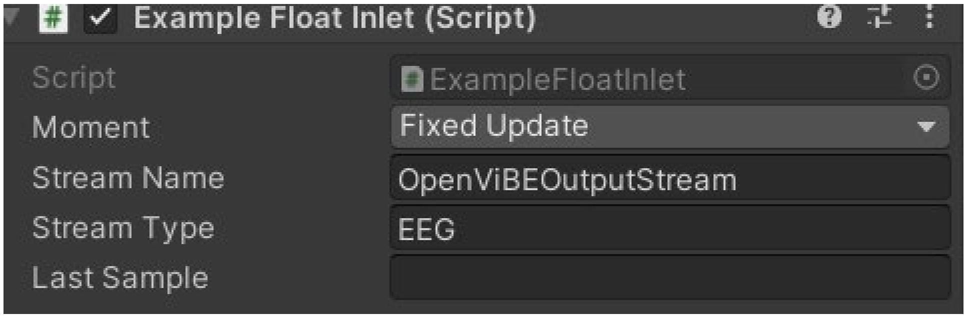


Figure 42 Initialization of data transfer, _inlet hold transferred data (probability of event occurred)

```
audioSource = GetComponent<AudioSource>();
_inlet = (ExampleFloatInlet) FindObjectOfType(typeof(ExampleFloatInlet));
if (_inlet)
    Debug.Log( message: "LSL object found: " + _inlet.lastSample);
else
    Debug.Log( message: "No LSL object could be found");
```

Figure 43 Stream Set Up in Unity Components

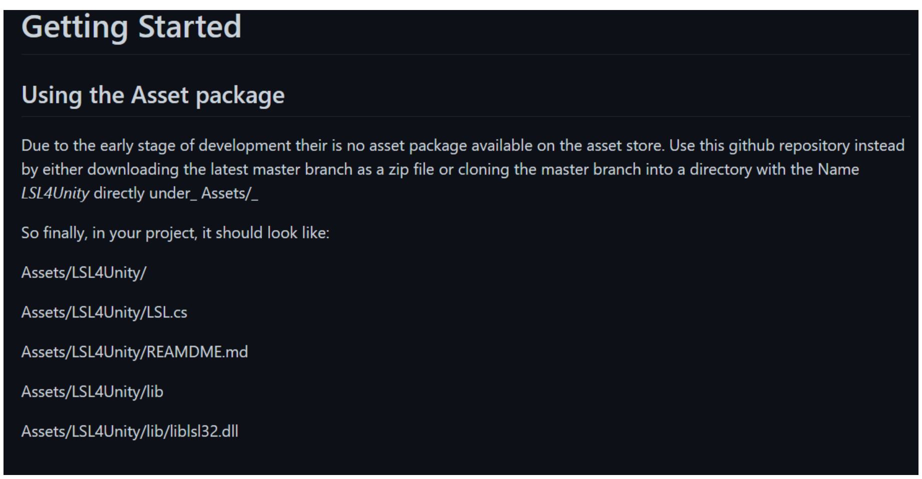
3.7 Challenges during the project development

This project was very complicated for me personally to develop. All areas of the project were new for me, and I didn't have any background to handle them (e.g., game development models, biological background).

Firstly, I spend a lot of time on reading Neuroscientific background of a studied topic in advance on September. I learned a lot about brain anatomy, functions, and overall, a lot about myself. This self-exploration was main driving force for me to continue to develop this project.

I learned completely new programming language C# and game development engine during self-education to handle all problems.

Big challenge for me was to use open-source libraries for interconnecting OpenVibe with Unity. The main problem was lack of documentation or precisely absence of it



Getting Started

Using the Asset package

Due to the early stage of development their is no asset package available on the asset store. Use this github repository instead by either downloading the latest master branch as a zip file or cloning the master branch into a directory with the Name *LSL4Unity* directly under_ Assets/_

So finally, in your project, it should look like:

Assets/LSL4Unity/

Assets/LSL4Unity/LSL.cs

Assets/LSL4Unity/REAMDME.md

Assets/LSL4Unity/lib

Assets/LSL4Unity/lib/liblsl32.dll

Figure 44 All documentation for LSL at a time of development

I am literally using reverse engineering methods to understand how to use this library, by going line by line through source code.

The biggest critical point for me was second term. I have had 3 modules during second term (It was Languages and Compilers, Network Security, Evaluation Algorithm). All these 3 modules were consuming a lot of my time and energy to make them good and make all notes, assignments, tests, labs in time. Also, pressure was coming from uncertainty about future. This is final year for me in university, therefore I started job search (making several CVs, covers letters, going to interview) and start finding potential options and places for living and overall global political environment make big impact on me due to fact that my relatives is living very close to Ukraine. I was thinking that “I am juggle with 10 bolls at the same time on high-speed train” (this analogy is very good describe second term)

I was changing initial plan of development very dynamically during second term due to the above reasons.

Initial ideas were to use Sound Imaginary for controlling TV and Lamp states and record electric signal from hand to make switch for Motor Imaginary to Sound Imaginary processing bet this idea was postponed and changed on full motor imaginary system. 3 commands for controlling character movement and 1 command for controlling TV and Lamp on/off functionality.

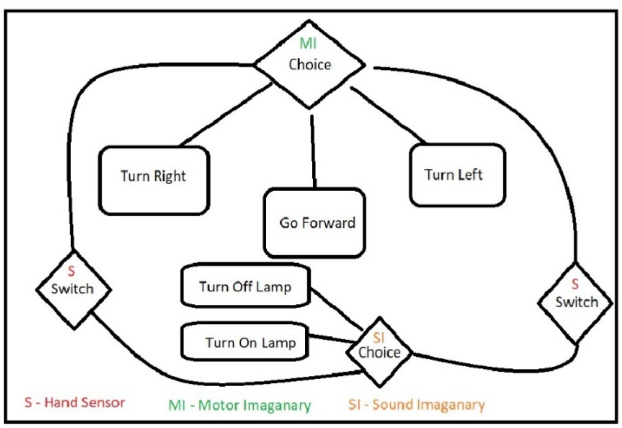


Figure 45 Initial System Diagram

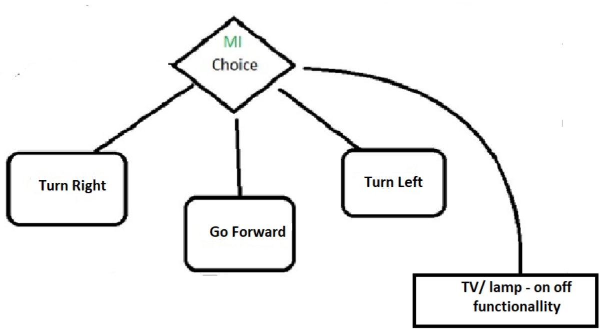


Figure 46 Modified Plan

I was planning to rewrite OpenVibe code to python to make complete build of a program. This idea also was postponed

Also, management of a project in Jira take big hit due to lack of time in second term. I am literally didn't have time to create and manage tasks. I was making notes mainly on my phone on notes app.

Overall, I somehow manage to solve nearly all problems and projects is working correctly.

4. Product Evaluation

The main parameters for product evaluation is efficiency of BCI algorithm

4.1 Evaluation of BCI Algorithm

Cross validation technique was used to assess efficiency of BCI algorithm.

Cross-validation is a method for evaluating an analytical model and its behavior on independent data. When evaluating the model, the available data is divided into N parts. Then, the model is trained on N−1 parts of the data, and the rest of the data is used for testing. The procedure is repeated N times; in the end, each of the N pieces of data is used for testing.

The result is an estimate of the effectiveness of the chosen model with the most uniform use of the available data.

Algorithm was unchanged during all process of learning with both data sets.

4.1.1 Data Set 1 – Evaluation

Number of examples for each class: 980

Number of classes for evaluation: 2

Class list: right hand signal, left hand signal

Number of cross validation tests: 7

```
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 1 / 7 (performance : 53.928571%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 2 / 7 (performance : 84.285714%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 3 / 7 (performance : 83.214286%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 4 / 7 (performance : 88.928571%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 5 / 7 (performance : 64.285714%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 6 / 7 (performance : 70.357143%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 7 / 7 (performance : 65.357143%)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Cross-validation test accuracy is 72.908163% (sigma = 11.894
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Cls vs cls      1     2     3
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  1:   71.4  28.6   0.0 %, 980 examples
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  2:   25.6  74.4   0.0 %, 980 examples
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  3:   -1.$  -1.$  -1.$ %, 0 examples
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Training set accuracy is 77.091837% (optimistic)
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Cls vs cls      1     2     3
time 463.195 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  1:   75.9  24.1   0.0 %, 980 examples
```

Figure 47 Cross-validation result after training for data set 1

Cross validation test demonstrate that test accuracy is approximately 73%, therefore algorithm produce correct probability of event occurred in 7 out of 10 times in average for 2 class data sets. It is relatively very good result if take in account that we are working with raw data which was record on surface of the head without any surgical interventions. Optimistic result for this data set and algorithm is approximately 77%

4.1.2 Data Set 2 – Evaluation

Number of examples for each class: 2295

Number of classes for evaluation: 4

Class list: right hand signal, left hand signal, tongue signal, foot signal

Number of cross validation tests: 7

```
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> k-fold test could take quite a long time, be patient
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 1 / 7 (performance : 60.183066%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 2 / 7 (performance : 63.844394%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 3 / 7 (performance : 52.286585%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 4 / 7 (performance : 45.690313%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 5 / 7 (performance : 59.832317%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 6 / 7 (performance : 63.386728%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Finished with partition 7 / 7 (performance : 43.445122%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Cross-validation test accuracy is 55.524075% (sigma = 7.786449%)
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Cls vs cls      1     2     3     4
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  1:   79.6  12.7   5.3   2.4 %, 2295 examples
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  2:   19.9  80.0   0.1   0.0 %, 2295 examples
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  3:   19.3   2.1  30.4  48.2 %, 2295 examples
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer>   Target  4:   16.0   2.8  49.0  32.2 %, 2295 examples
[ INF ] At time 3949.000 sec <Box algorithm::(0x02e67945, 0x5ea8d309) aka Classifier trainer> Training set accuracy is 60.021786% (optimistic)
```

Figure 48 Cross-validation result after training for data set 2

Samples set was 2.3 times big for each class in data set 2 bet number of recognized classes have increased twice from 2 till 4 which dramatically increase classification complexity. Cross validation test accuracy falls down on 17.4% from 72.9 till 55.5 percent. Optimistic result for second data set with 4 classes is about 60%. Overall, it is quite good result for such basic algorithm with completely raw data from head scalp.

5. Project Management

There was used several tools and programs for Project Management. It is Jira, Gitlab, OneDrive, Samsung Notes (don’t underestimate this tool)

Protocol for management tasks in Jira was Kanban. Tasks was split by weeks by using Epic Links. Each task was assigned deadline and acceptance criteria which it must feet to be accepted.

5.1 Jira – Term 1

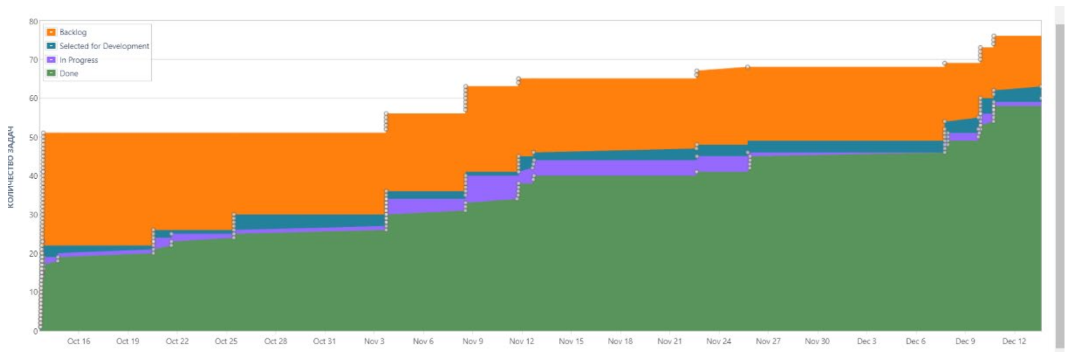


Figure 49 Cumulative Flow Diagram, Term 1

A lot of work was done during first term. Nearly all elements of research part were done during first term and a lot of part of Unity Development. There had been done about 80 tasks. Initially, it was visible that second term will be very intense due to big number of heavy modules, therefore it was planned to make all “heavy parts” of a project during first term and this decision was very successful for a project completion.

5.1.1 Epic Links

All tasks during first term were divided into weeks by using epic links.

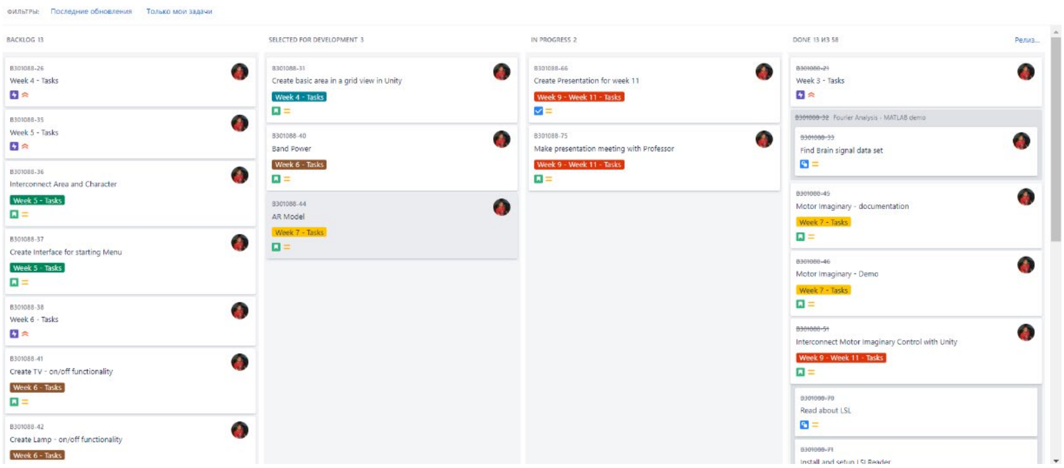


Figure 50 Kanban board structure

Epic link of week was considered finished when all tasks at this week was done. This management decision give freedom to see how different parts of project is moving towards the end and what parts of a project requires extra attention. Some big tasks were subdivided on subtasks to solve it (It was done this way, because I have initial idea, bet didn't know steps to achieve this idea, this subdivision was helping me to solve big problem by small bits).

5.1.2 Tags

Tags was assigned to all tasks to simplify search process for different groups of work

Used tags:

- Managing – all management work
- Development – all code related or development work
- Research – all research or self-education work

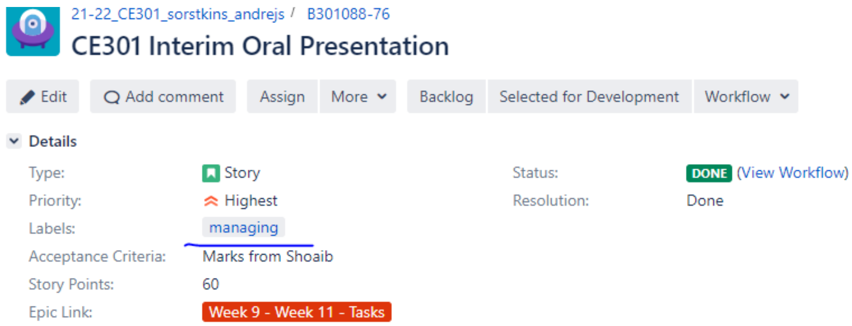


Figure 51 Example of Label or Tag

5.1 Jira – Term 2

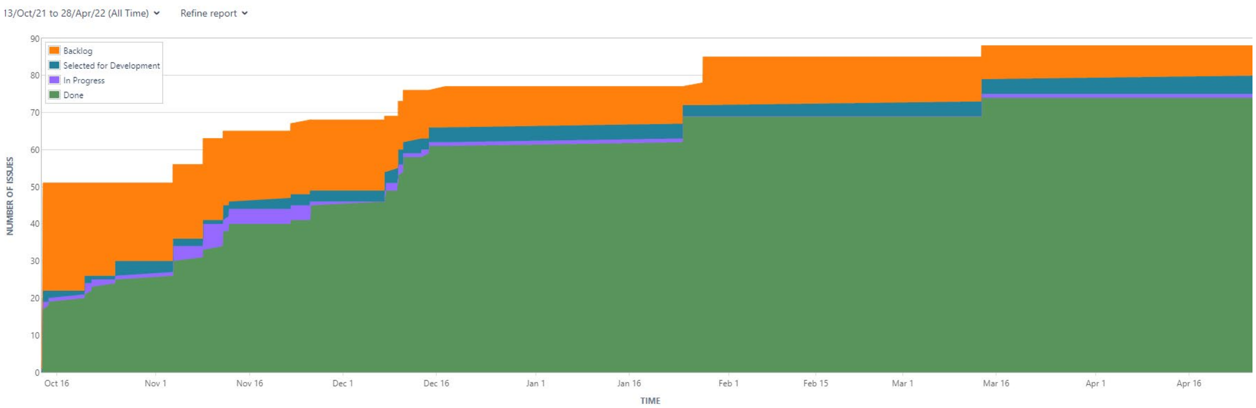
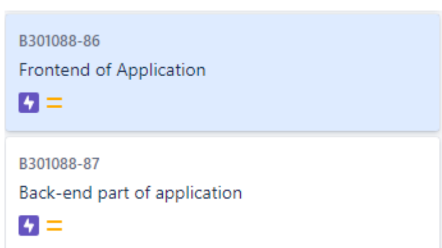


Figure 52 Cumulative Flow Diagram of management during year time

As was discussed in challenges sub chapter second term was very heavy for management because I literally didn't have time for it. The way of management is changed a lot due to this.

Firstly, I refused idea to use week approach for making tasks and change it to group approach. There was 2 main group of tasks (or epic links):

- Fronted of Application – All Unity related code
- Back-end part of application – All OpenVibe and MATLAB code



Estimation of tasks timeline was also very hard due to fact that I was doing everything at the same time switching constantly from one task to another.

Overall, there was done about 90 tasks during this period in Jira. Some of the tasks haven't been recorded in Jira. Overall, I estimate amount tasks done to 110-120.

5.2 Gitlab

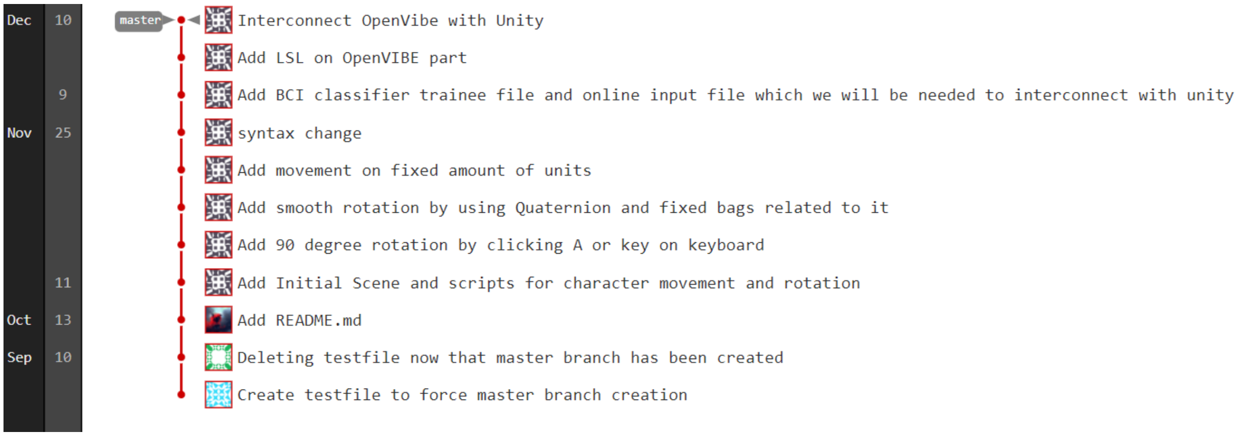


Figure 53 Engagement Map -term 1

Most project related files were stored in Gitlab due to version control system. It is visible on engagement map that main part of development started from 11 of November. It is happened this way because first month and a half I spend on research of all relevant topics (Neuroscience, 3D engines for creating virtual room). Commits rate haven't been fast during all project due to fact that it takes me a lot of time to make every new commit due to project complexity and time for exploring given topic for a commit.

There were several problems with making commits into Gitlab, firstly Unity project weight is several gigabytes due to assets size, therefore it was nearly impossible to store it on Gitlab. I was stored on Gitlab all scripting and scene related part for Unity, and all data files and OpenVibe script files.

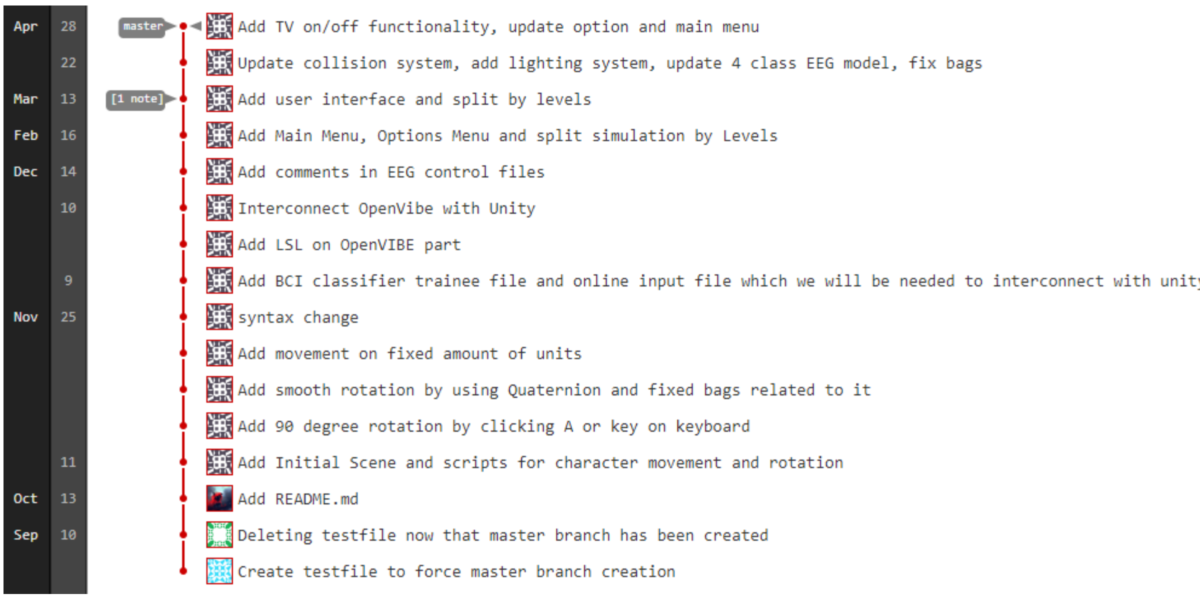


Figure 54 Engagement Map -all year

Overall, all tasks during year time were done and stored in Gitlab, It was very useful tool for version control of all C# scripts

5.2.1 Git ignore

There were big complications to import projects files to Gitlab due to assets file size. Solution for this problem was to go through all file structure of Unity project and write ". gitignore" file

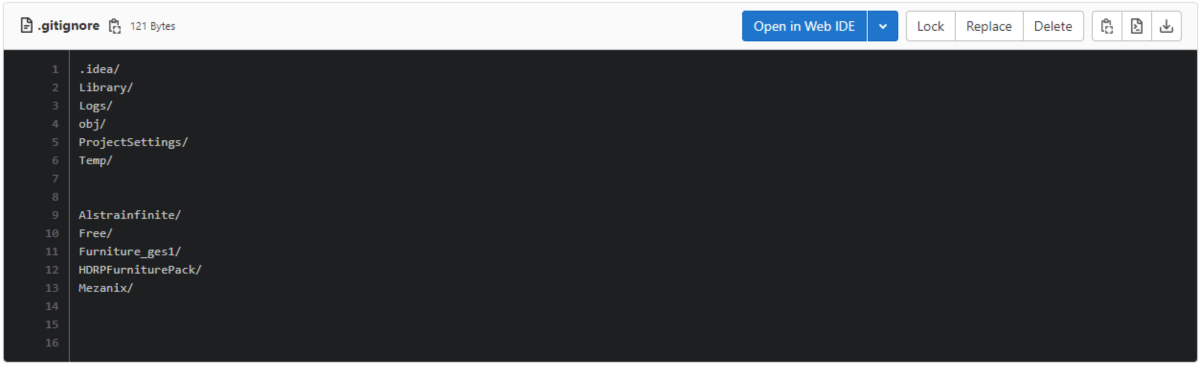


Figure 55 Gitignore file

5.3 OneDrive

I was storing all documents (research paper, presentations), assets for all of it and all project in OneDrive. It was done due to fact that university computers were restarted environment after each login, and it was much easier to access OneDrive than redownload several gigabytes of repository every time

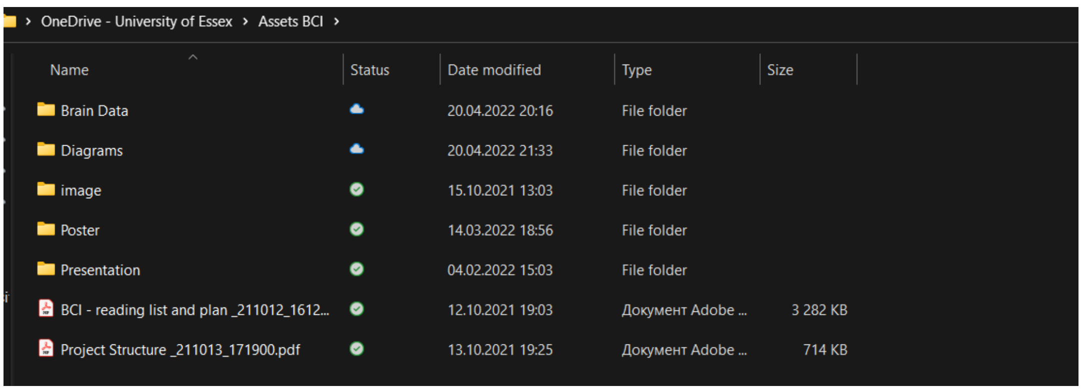


Figure 56 Assets for BCI project in OneDrive

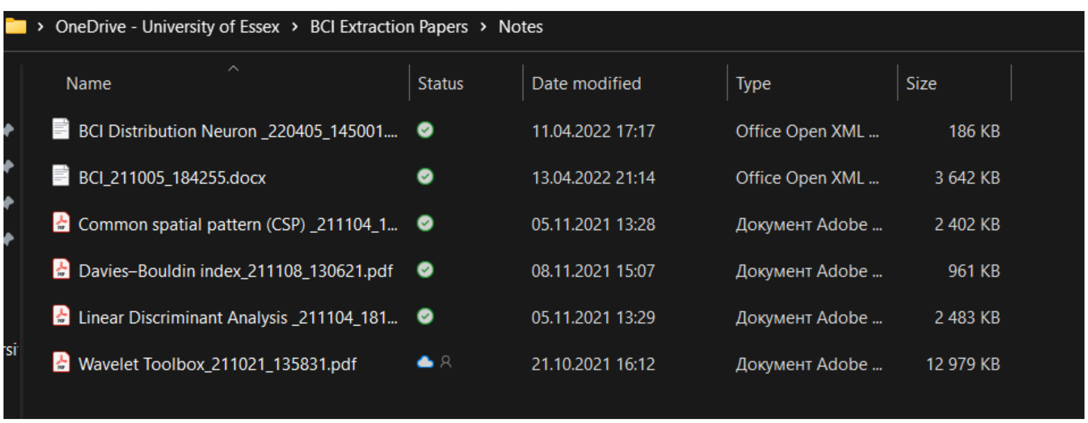


Figure 57 Notes on different research topic in OneDrive

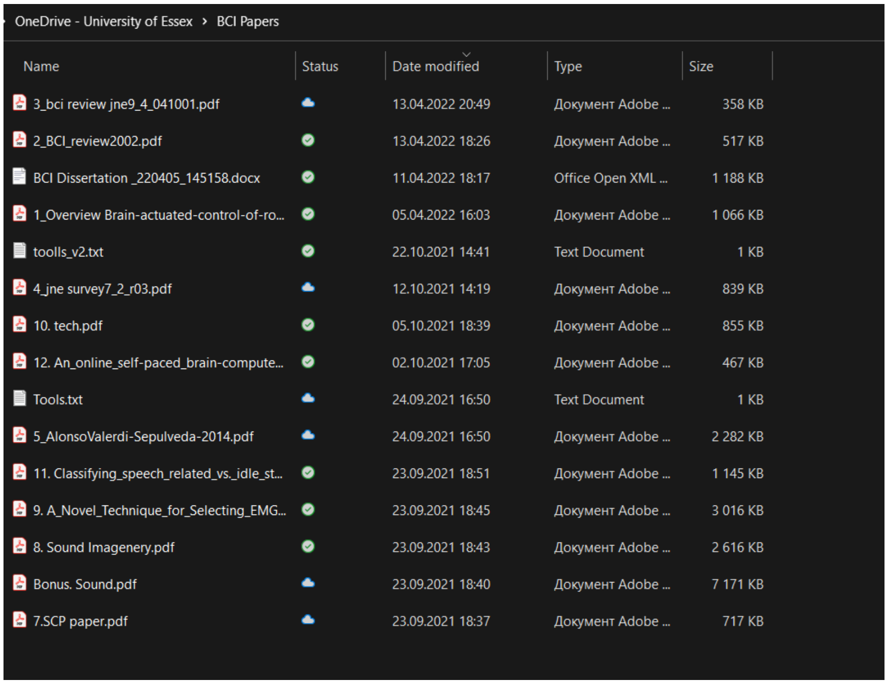


Figure 58 Some research Papers about BCI in OneDrive

5.4 Samsung notes

A lot of thinking and planning about project was done in Samsung Notes App on my tablet, I was using it like notebook for all my sketches and drafts about different concepts in BCI and Unity. It was nearly always first step before entering data to Jira or starting to write any report or code.

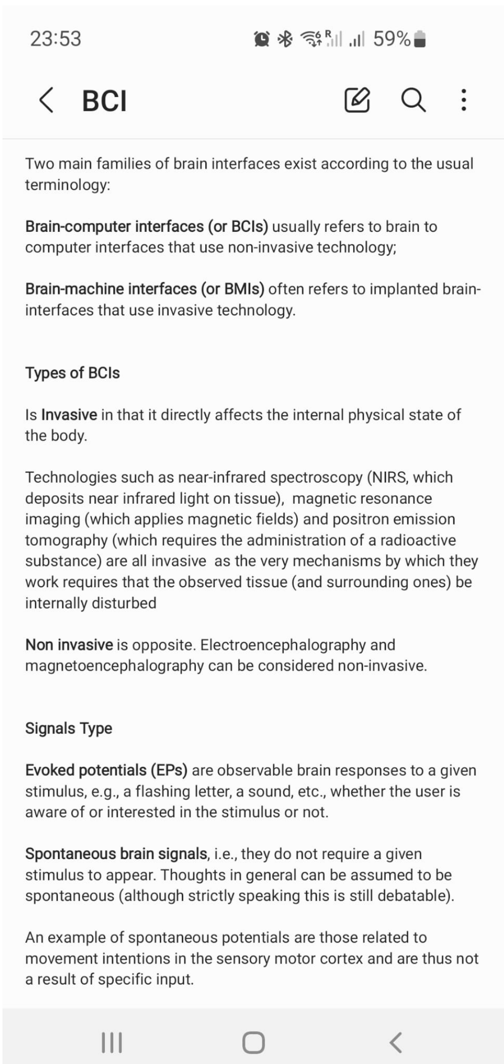


Figure 59 Notes about BCI

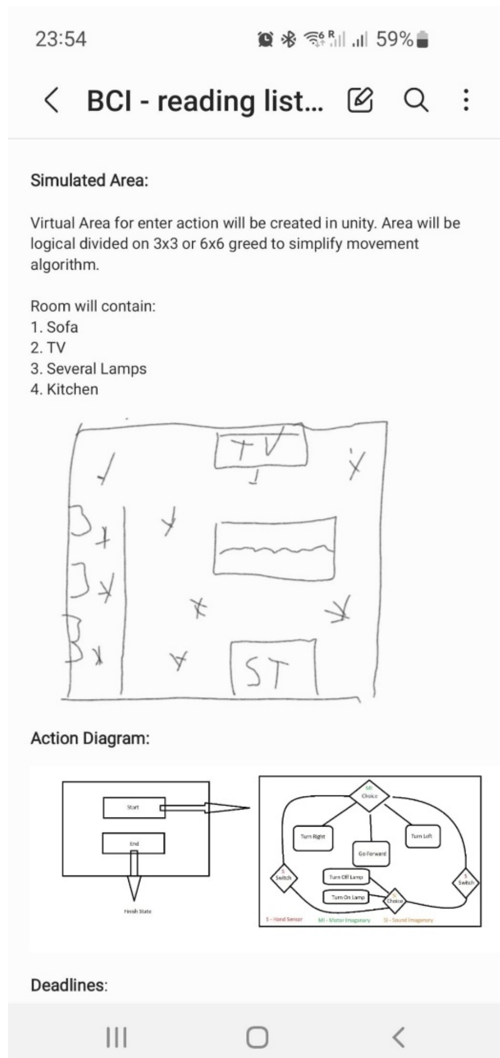


Figure 60 My drafts about room design

6. Conclusions and Future Directions

A lot of work was done to make this project alive. There is list of completed goals:

6.1 Summary of achievement

6.1.1. Achieved tasks:

Research Part:

1. Explore Fourier Analysis
2. Explore C# syntax and code writing
3. Explore CSP
4. Explore LDA (Linear Discriminant Analysis)
5. Explore Davies–Bouldin index
6. Explore. The brain–computer interface cycle
7. Explore Neuronal mechanisms underlying control of a brain–computer interface
8. Explore Covert sound production task – Sound Imaginary
9. Explore A Novel Technique for Selecting EMG-Contaminated EEG Channels
10. Explore Unity Documentation
11. Explore OpenVibe Documentation
12. Explore GDF file format
13. Explore Unity level system
14. Explore Unity UI interface development
15. Explore validation technique

Development Part:

1. Created Virtual Room in Unity
2. Create Virtual Character and scripts for character movement and rotation
3. Updated movement part in a greed layout manner
4. Add smooth rotation by using Quaternion
5. Create BCI classifier and trainee file in OpenVibe and LDA
6. Interconnect OpenVibe and Unity with LSL
7. Interconnect right and left-hand imagination with character rotation in Unity
8. Create Object Collision system
9. Create Main Menu
10. Add Control of frontal movement
11. Add TV and Lamp – on/off functionality
12. Update room decoration
13. Create 4 class BCI data extraction algorithm

6.1.2. Failed or not done tasks:

1. Sound Imaginary Signal processing and extraction
2. Switcher for controller
3. Muscle Signal interconnection
4. I haven’t rewrite OpenVIBE code on python to create complete build of the program

6.2 Future direction of this project

This project has had research-based nature due early stage of overall development *of its components.* BCI is very young field of Computer Science, computer technology and neuroscientific understanding is only reach base level of understanding and extraction of data from a brain. It was mention in a paper that non-invasive technology can get second alive step when BCI acquisition technology will decrease in cost and size to make it more commercially real product.

Nowadays, non-invasive BCI based systems is mainly used only in research lab to get better understanding of brain functionality and improve existent algorithm for data extraction from a brain. I can assume that non invasive BCI based systems will be more commercially and algorithmically ready when be based on Magnetoencephalography and when MEG will decrease in size and cost due to fact that it gives very good accuracy rate of measurements.

Another field which very rapidly growing nowadays is invasive based systems. Such type of systems can give nearly perfect accuracy rate if be implemented safely for final consumer. One of the main advantages that such systems is very compact to use. From another side, there are big number of debates about moral topic of such technologies, will they be saved for use from a cyber security perspective and will the have any bad long-term effect, all such questions will be needed to answer in research laboratories by different groups of scientists and research institutions before this technology will se commercial live.

6.2.1 Potential evaluation of this project

I think this project is a good example of a project which can evolve into master degree and after that to doctor degree thesis due to fact that it based on front level of current technologies and human understanding of human brain and methods to work with him.

My aim for this project if I or someone else will decide to continue to develop it in master and doctor degree level is:

First step can be to finish initial idea to use 2 different technologies for data extraction from a brain. It is sound imaginary and motor imaginary, it was visible in evaluation chapter that accuracy rate is falling down when more classes added to algorithm, this problem can be partially avoided by splitting different commands between different parts of a brain.

For example, 2 technologies can be used for data extraction from 2 different parts of the brain

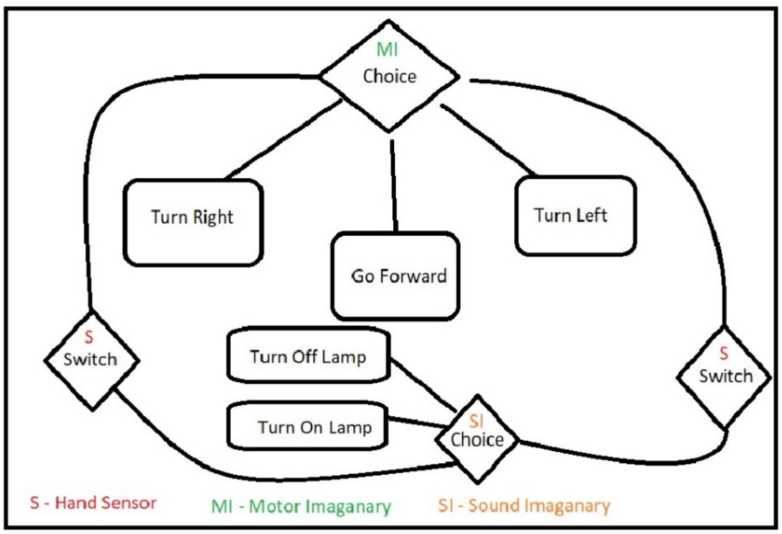
Movement can be controlled by **Motor Imaginary**:

- Go Forward - person thinking about left leg
- Turn Right - person thinking about right hand
- Turn Left - person thinking about left hand

Game Object Control can be controlled by **Sound Imaginary**:

- Turn On Lamp - person tell inside himself "Lamp On “
- Turn off Lamp- person tell inside himself "Lamp Off “
- Turn off TV- person tell inside himself "TV Off “
- Turn ON TV- person tell inside himself "TV On"

Switching mechanism between them can be electrical signals from hand muscles, e.g., person interconnect his middle and big finger together.



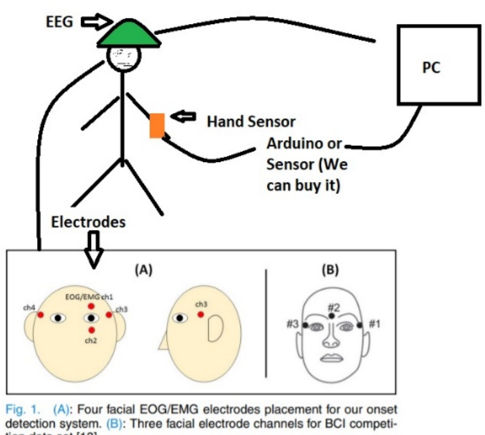


Figure 61 Example of potential system

Such approach can decrease computational complexity from system and increase accuracy rate for all system.

Other aspects of the project are to improve classification algorithm in a way of adaptation to a person individual brain signature. It can be done by implementation of evaluation algorithms or by training of neural networks to solve this task.

Potentially solution of this questions can be founded in near future and I am very proud that take part in this project. It helps me much better to understand myself and overall effort which people make to implement new technologies into our day-to-day basis.

References

[1] Brain–computer interfaces for communication and control, Jonathan R. Wolpawa, b, Niels Birbaumerc,d, Dennis J. McFarlanda, Gert Pfurtschellere, Theresa M. Vaughana, Accepted 2 March 2002

[2] Pfurtscheller G and Lopes da Silva F H 1999 Event-relatedEEG/MEG synchronization and desynchronization: basicprinciples Clin. Neurophysiol. 110 1842–57

[3] Gazzaniga M S 2004 The Cognitive Neurosciences III 3rd edh (Cambridge, MA: MIT Press)

[4] Unity Documentation, Lightmapping, https://learn.unity.com/tutorial/configuring-lightmaps#:~:text=Lightmapping%20is%20the%20process%20of,a%20relatively%20low%20computational%20cost.

[5] LSL Documentation, GitHub, https://github.com/sccn/labstreaminglayer

[6] GDF - A GENERAL DATAFORMAT FOR BIOSIGNALS VERSION V2.5, Alois Schlögl, Institute for Human-Computer Interfaces, Graz University of Technology, Austria, Institute of Science and Technology, Klosterneuburg, Austria

[7] BioSig Library, http://biosig.sourceforge.net/

[8] Brain-actuated Control of Robot Navigation, Francisco Sepulveda BCI Group - School of Computer Science and Electronic Engineering, University of Essex, United Kingdom

1.