Final Degree Project

Biomedical Engineering Degree

"REHABILITATION PROCESS USING ELECTROMYOGRAPHY AND BIOFEEDBACK "

Barcelona, June 2021

Author: Francisco José Domingo Gil

Director/s: José Luis Parreño Catalan

Tutor: Manel Puig Vidal

ACKNOWLEDGEMENTS

I would like to express my gratitude to both my director José Luis Parreño, for trusting me and giving me the opportunity of developing the project with him, and my tutor Manel Puig for helping me, teaching me, and supporting me during the development of the project.

Moreover, to my family and friends for giving me the necessary support to carry on the project.

Thank you.

Abstract

A good rehabilitation routine is essential for the best possible recovery after an injury or to increase the quality of life of those who suffer from neuromusculoskeletal diseases. It is of particular relevance to maintain the motivation throughout all the process, for which videogames may play an essential role.

Biofeedback is a process that provides real time information from psychophysiological recordings about the levels at which physiological systems are functioning. In this project the biofeedback system has been implemented by means of a low-cost EMG system created using Arduino.

The EMG system has been developed using an Olimexino-328 microcontroller and an EMG-Shield, both from Olimex. The program was developed using Arduino IDE. To assess the quality of the signal of the prototype, it was compared to a professional EMG device, the DataLog from Biometrics Ltd. The comparison showed promising results although it could be improved by means of post-processing algorithms.

An audio-visual Biofeedback system targeting maximum strength and explosiveness of the muscles was created using Python. Flappy Bird game commands were changed to control them with the EMG low-cost prototype. The flap logic of the game is guided by a threshold fixed automatically by the software at the 60% of the maximum signal obtained by the EMG system.

The system was tested to optimize its performance and fix possible flaws. Although it is simple and further investigation may be needed, having in mind the fewer resources used, the system performance is encouraging, and a clinical trial should be performed to assess its real behaviour, usefulness, and efficiency for rehabilitation purposes.

Keywords

Rehabilitation – Arduino – EMG – Low-cost – Audio-visual Biofeedback

List of figures

List of tables

INDEX

1. Introduction

As you know skeletal muscles are the motors that allow humans to move. The mechanism of action of muscles is a very complex procedure which incorporate signal transmission along nerve fibres and across neuromuscular junctions, electrical activation of the muscle fibres, which are organized in elementary units known as motor units, which finally produce forces that act on the tendons of the muscles and allows bones movement. ElectroMyoGraphy is a technique which evaluates the health condition of all this procedure. It translates the signal into graphs or numbers which can be used for many applications such as diagnosis or treatment. [1]

Understanding EMG signals implies the understanding of muscles and the way they generate bioelectrical signals. It implies understanding how specific mechanisms and phenomena influence the signals, as well as the inverse problem, which is even more difficult, and consists of understanding how signals reflect certain mechanism and phenomena and allow their identification and description [1].

The electric signal is produced in the cortex, and travels all along the spinal cord until it arrives to the motoneuron, which transmits the signal to the skeletal muscle. Figure 1.1 shows a simplified schematic of the signal transmission [1].

There are two types of EMG: neurologic EMG and kinesiologic EMG. Each one has some advantages and disadvantages shown in Table 1.1.

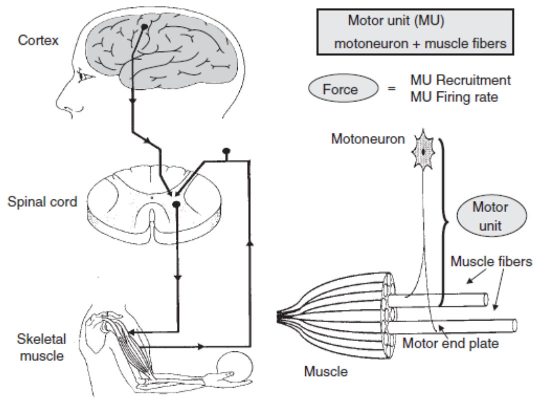

Figure 1.1 Signal transmission schematic. Extracted from [1]

	Neurologic EMG (nEMG)	Kinesiologic EMG (sEMG)
Position	Directly in the muscle	In the surface
Invasiveness	Invasive, painful. Use of needles	Non-Invasive. Use of electrodes
Complexity	Highly complex, need of high anatomy knowledge	Simple, lower knowledge needed
Precision	High precision, detects signal from small muscles, individual action potentials	Low precision, limited to surface and big muscles. Difficult to get signal from an isolated muscle

Table 1.1 Neurologic vs Kinesiologic EMG

EMG applications have been growing since the development of this technology. Numerous applications for this technology have been developed in clinical practice, such as diagnosing of neuromuscular diseases analysing and determining abnormalities or for treatment purposes which could be improving ergonomics or muscular rehabilitation, what is called *biofeedback*. *Biofeedback* is defined by the Association for Applied Psychophysiology and Biofeedback (AAPB) as a process that provides real time information from psychophysiological recordings about the levels at which physiological systems are functioning. This means that by means of sEMG, *biofeedback* measures and transforms the physiological information from muscles into visual and/or audio signals [2, 3, 4, 5].

When an injury occurs, not only bones, ligaments and muscles are hurt, but also special nerve receptors. These receptors, found in muscles, send great amount of information back to the brain. If during rehabilitation the receptors are not retrained, injury is more likely to reoccur. Is in this situation where biofeedback becomes important. It helps the patient to retrain the muscles and receptors thank to the outcome that they receive. The loop of biofeedback is shown in Figure 1.2. The brain produces a command which arrives to the effector organs, the muscles. Thanks to sEMG, the signal can be monitored, and an outcome is obtained. This raw outcome is a graph, which does not make any sense to the patient. Therefore, it must be processed and transformed in another outcome more recognizable to the patient, so the signal is transformed into an acoustic or visual feedback [5].

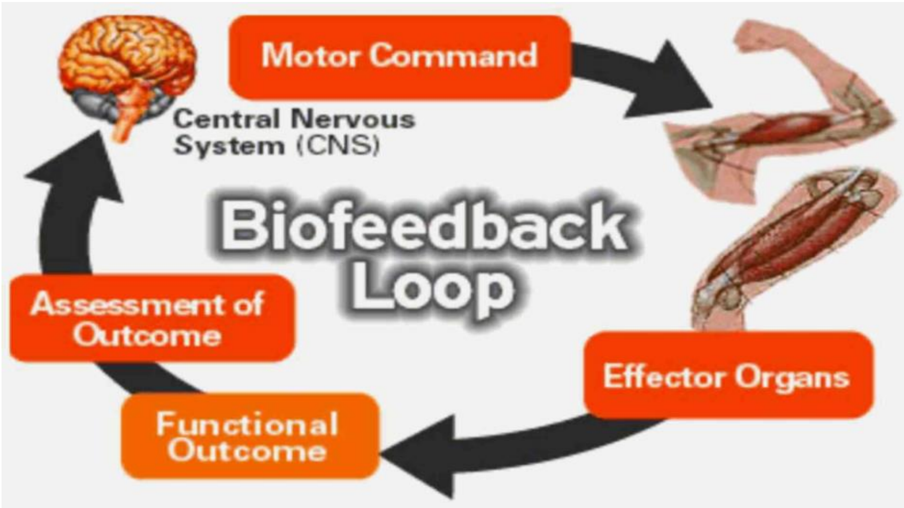

Figure 1.2 *sEMG Biofeedback loop. Extracted from [5]*

1.1. Motivation

Since very young I have suffered from different injuries myself such as elbow luxation, collarbone fracture or knee injury, therefore I have had to attend rehabilitation several times and during different stages of my life, therefore I have been able to experience myself the evolution of the technology for rehabilitation.

In addition, I have always had a great interest on technology, and my own experience showed me how technology can improve people's quality of life, but also that many times rehabilitation processes are tough and boring, which can lead to the abandonment of the process. When I saw the opportunity of developing a project in that field, I thought that it could be a great opportunity to apply my own experience and to gain knowledge.

Initially the project was focused on shoulder rehabilitation and acoustic biofeedback, but during the development of it, the director and me decided that it could be better to generalize to the rehabilitation process since biofeedback could be applied to many muscles of the body. Also, the biofeedback system was changed to a more complex type, which would be much more interesting both for me as a student and the challenge that it would suppose completing it, and for the project itself since it would make more motivating the rehabilitation process, fact that as it will be explained later, is a very important point.

1.2. Objectives

This work was based on the belief that rehabilitation procedures could be improved to make them more friendly and effective. A study in Argentina shows that exists a highly abandonment ratio of rehabilitation procedures [6]. I believe that a more entertaining rehabilitation procedure would reduce this abandonment percentage and biofeedback could have a fundamental work on it.

The main goals of this project were the following:

- Assess the use of biofeedback and EMG in rehabilitation.
- Compare the signal obtained in the low-cost EMG systems with a professional EMG system.
- Develop an EMG and biofeedback low-cost system.
- Clinical trial of the project in *Institut de Biomecànica Clínica*.

1.3. Methodology

The project was performed with the help of José Luis Parreño and it consisted of the following stages:

- Stage 1: Bibliographic research of both technologies was performed to get familiarized with the topic. It was focused on the applications that EMG and biofeedback can be used.
- Stage 2: the development of the EMG and biofeedback system was performed. This phase consisted of getting the needed material for the project and the development of the software needed for both obtaining the EMG signal and the development of the Biofeedback interface.
- Stage 3: Comparing the EMG signal of my prototype with a DataLog from Biometrics Ltd. A professional EMG system.
- Stage 4: This last phase consisted mainly in writing the present report. It consists of an introduction of both technologies, then a description of the possibilities we had for the EMG and biofeedback development and the final solution. Next the whole process of developing the system. Then the technical and economic viability, and the timeline. Finally, a conclusion of the project.

1.4. SCOPE and Limitations

As this was a final degree project, time has been an important limitation. It has been carried out from February 2021 to June 2021. The main limitation where time affected is that it would have been interested to apply this project for patients to perform the efficacy of the system, but biofeedback rehabilitation procedures need a much larger time window, which made impossible to complete this objective.

Covid-19 pandemics has been another large limitation, which forced to perform from distance this project as much as it was possible. Probably if this pandemic did not happen, we could have started the project in April 2020 and the clinical study could have been performed. This also caused that the meetings with the director had to be online and I could not attend the biomechanics laboratory where the project was intended to be performed, what would have helped me on understanding better the applications of Biofeedback.

Considering all these limitations the Scope of this project included:

- Bibliographic research of EMG, Biofeedback applications and types of biofeedback currently used.
- Development of an EMG prototype using Arduino
- Development of a Biofeedback software
- Comparing EMG signal of my prototype with the DataLog system.
- Discussion of the differences between the signals

1.5. Location of the project

The project described has been performed in collaboration with José Luis Parreño, director of the Institut de Biomecànica Clínica.

2. State of the art

To develop an EMG system and an effective biofeedback software, it has been necessary to understand at which point is the current technology and some biofeedback systems to get an idea of the software I could develop.

2.1. EMG

To correctly perform a sEMG analysis, several points are needed to be took in account. There are different variables that will lead to wrong results if they are neglected.

First of all, the electrode shape. It is defined as the shape of the conductive area of the SEMG electrodes. In literature both square and circular electrodes are reported to be used for SEMG recordings. As SENIAM project (Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles) states, not much difference in performance and pick up can be expected. As long as the total surface are for both electrodes is the same, the skin impedance of both electrodes will almost be equal, so no influence can be expected. However, the European inventor showed that circular electrodes are preferred [7].

Second, the electrode size. It is defined as the size of the conductive area of a SEMG electrode. This parameter is important since an increase of the size, is expected that the view of the electrodes increases. It can be shown that has an integrative effect on the SEMG signal, increasing the detected amplitude and decreasing the high frequency contents. In general, it is important that the size of electrodes is large enough to be able to record a reasonable pool of motor units, but small enough to avoid crosstalk from other muscles. Both the SENIAM and the European inventor recommend electrodes of 10mm of diameter maximum [7].

Third, the inter electrode distance. It is defined as the centre-to-centre distance between the conductive areas of 2 bipolar electrodes. The influence of the inter electrode distance on the recorded area and crosstalk is a relevant item. SENIAM recommend an inter electrode distance of 20mm. If the SEMG is performed in small muscles, the distance should not exceed ¼ of the muscle fibre length. This will avoid unstable recording due to tendon and motor endplate effects [7].

The preparation of the skin is a very important factor to get a good electrode-skin contact. It will help to obtain better SEMG signal and lower noise and artifacts. There are different techniques and combinations in this field, but SENIAM recommends shaving the patient, and then clean it with alcohol and allow it to vaporise so the skin is dry before electrodes placement [7].

Surface EMG devices have been developing since 1950. In that time the devices used analogic systems. From that point the systems have been continuously evolving, developing digital systems to the actuality, which most of the are portable devices with wireless sensors. One of the most advanced EMG devices is FREEMG. It is a SEMG device which uses 4G technology. It distinguishes from its competitors for the high signal accuracy, the absence of wires, the lightness and reduced size of the probe and many other properties which allows to perform analysis of any type of movement, for any muscle altering the minimum possible the movement of the subject. It is connected to the PC through a USB receiver, which can manage up to 20 sensors simultaneously. Even though, the sensors have an internal memory to ensure that the signal is recorded in case that a temporary disconnection occurs, and to allow recording in wide places where the subject can be far from the USB receiver [8].

Figure 2.1 *FREEEMG system. The sensors can be seen. Extracted from [8]*

Some of the problems of this type of cutting-edge technologies are that they are expensive and difficult to operate, which make impossible to use by non-specialists. Hence a new sEMG device, which is defined as simple to use and highly accessible, has been developed for people who are not familiar with EMG. But before using it in professional activities, the validity and reliability must be stablished. To do so, this new sEMG system, showed in Figure 2.2, was compared to the aforementioned FREEEMG. The signal was recorded in participants which performed three maximal voluntary isometric contractions for 5 seconds each with a 5-minute rest between each one. The electrodes were placed following the SENIAM protocols, and the signal was recorded at the same time. After analysing the recorded data, it was confirmed that the new system valid enough to monitor muscle activity during daily life exercise [9].

Figure 2.2 *New sEMG device.*

During or after the acquisition process, different signal filtering and processing procedures are used. First, a denoising step is performed. It can be denoising by hardware or by software. The hardware denoising improves the performance of the device using physical filters. The software denoising is done using filters or wavelet transforms. The wavelet transform is an extension to traditional Fourier transform. Its coefficients have different characteristics at each scale of noise and signal, so the idea is to remove these components generated by noise at each scale. Then the inverse transform is used to reconstruct the original signal [10].

If a comparison between analog and digital filters is started, it will not end fast. In one hand, analog filters are faster which reduces the delay to 0, but they need of several components which can reduce its accuracy due to temperatures or difference in manufacturing. Analog filters then, increase power consumption and cost. In the other hand, digital filters may be slower, but since in EMG acquisition devices, the sampling frequency is at the range of kHz, the delay is negligible. In addition, they have other advantages such as, they act exactly equal, no matter the circumstances, and allow changing the notch frequency from 50Hz in Europe to 60Hz in North America easily by changing only one parameter. Furthermore, it allows the implementation of real-time algorithm such as artifact suppression or other devices control [11].

The 50hz signal is one of the main signals that must be removed from any acquired data. It can be done using analog or digital filter. In this case, by the reasons explained before, a comb digital filter was used. This comb filter was designed as a high pass filter and fixing a sampling frequency of 50hz, that Is the frequency to be filtered. A Butterworth and Chebyshev filter was used. The filter was designed using Matlab FDA filter tool. First, the filter zeros, poles and gain were determined as if it was an analog lowpass filter. Then, by bilinear transformation, the analog filter was transformed into a digital filter, and then to a zero-pole-gain form again. It gives as result the filter coefficients as floating-point numbers [11].

In EMG recordings other types of noise appear such as artifacts. Artifacts in EMG are different errors in the signal that can come from different sources such as cable movements or relative movements of the muscle to the EMG sensor, for example at the beginning and the end of a contraction. This error must be removed to obtain the clearest possible signal, an option is using digital high pass filters, since artifacts are in the range of 0-20Hz, and a cut-off frequency at that range can be fixed. After an evaluation of the filter using different cut-off frequencies, the optimal signal was obtained with f_c=60Hz with a Chebyshev filter. In addition, two lowpass filters were applied. One of them removes the high frequency noise after the high pass, and the other performs a smoothing to get a DC output signal. The f_c was fixed at 531Hz for removing high frequency, and 3.1Hz for the smoothing step [11].

2.1.1. EMG applications

EMG applications have been growing during last years as consequence of the development of new technologies such as machine learning and artificial intelligence. Many of the new uses of EMG are related to gesture recognition. The increase in computing power has brought the presence of many computing devices in the daily life of human beings. Hand Gesture Recognition, from now on HGR, models are human-computer systems that determine which gesture was performed. HGR models need acquiring big amount of data using different sensors such as inertial units, gloves, and SEMG sensors. All of them have limitations, for examples gloves cannot be used by amputees, and SEMG generate noisy data. Even though, all these sensors collect date related to a real movement, EMG also extracts the intention of the movement, therefore it might be useful with amputees who cannot execute the movements. As it has been explained, EMG record the electrical activity of skeletal muscles which have two types of contractions: static and dynamic. Each of them can be modelled using mathematical models, and the combination of the two of them results on an EMG model. However, the mathematical models are not used in HGR as consequence of the parameter estimation difficulty [12,13,14].

Therefore, machine learning is used. It can obtain a solution using different techniques. By applying ML, a wide range of possibilities for myoelectric control opens, and it allows using EMG signals to control prostheses or drones. In conventional amplitude-based control, one EMG channel us used for controlling only one function of the device, and when it exceeds a predefines threshold, this function is activated. For many applications HGR models are required to work in real time so one the user performs an action, the system gives him a response, but it is important that this response is fast enough to be perceived as instantaneous [12].

An example of this application is Myo Armband. It is a method for gesture control which is equipped with EMG and Inertial sensors. The inertial sensor uses an accelerometer, a gyroscope, and a magnetometer to obtain the maximum precision. With the combination of these two different sensors, a mobile robot can be controlled [15]. The following figures show the connection between the two devices:

Figure 2.3 *Myo ArmBand. The bracelet includes the sensors Extracted from [15]*

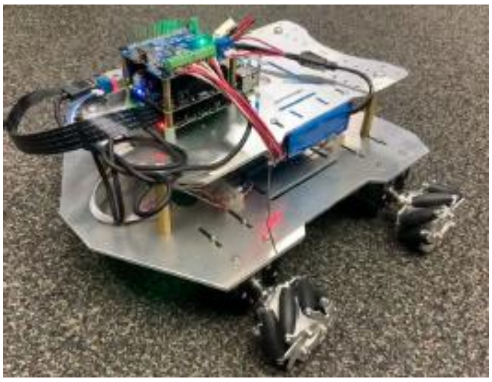

Figure 2.4 *Mobile robot controlled with the MyoArmBand Extracted from [15]*

The user receives feedback from the device with two different indicators: LEDs and a haptic feedback vibration. When a movement is produced the armband detects the electrical signal, and comparing the signal detected by the different sensors, it can be classified in 5 predefined gestures which are shown in Figure 2.5 [15].

Figure 2.5 *Predefined gestures that the armband detects. Extracted from [15]*

Each of these gestures will produce a movement to the robot, which can be go forward, backwards turn left, turn right, or stop. In this article it is shown that the robot control successfully exhibited real-time human-robot interaction through hand gesture using a low-cost SEMG device. However, it also has some drawbacks. For example, it cannot be used for medical applications such controlling robotic prostheses since these need six degrees of freedom axes, and the Myo has a smaller number of degrees. In addition, the accuracy and precision could differ between users or in cases where the muscles have impaired development [15].

Apart from these complex applications, EMG can be used for many rehabilitation purposes to perform and study and evaluation of the process. Incidence of surgery in shoulder lesions affecting the rotator cuff have increased a 238% from 1995 to 2009 in the United States. Many reports indicate failure after surgery, in most cases within the first 3 to 6 months post-surgery. Different factors affect the outcome, such as the size of the injury, the tissue quality, or the location, but postoperative rehabilitation protocol is one of the most important factors. After a rotator cuff surgery, it is important to improve range of motion in the early stages, but without overloading surgical repair which could lead to failure. The gold-standard for evaluating muscle activation and therefore, loading in limb, is EMG. It is important to categorize the activation into different levels, to control when it could be harmful to the rotator cuff. Different reports indicate that an activation greater than 15% could be harmful, therefore it is important to control muscle activation at any moment of the rehabilitation process. To do so, a control group was used which performed different exercises and the professionals assessed the percentage of activation that every exercise caused. Then, the exercises which had a lower activation than 15% were in the users. There were four types of exercises: actives, passives, active-assisted with the non-affected limb, active- assisted with bars. The three first cases showed activation below the 15% of the maximum, but the last one, showed higher. If EMG were not used, the exercise would still be used in the rehabilitation process, which could lead to a bad recovery [16].

2.2. Biofeedback

Since biofeedback can be used to train a specific function of the body and improve its voluntary control, several medical applications have appeared. It is often used to train neuromotor control during rehabilitation processes. Neuromotor control consists of selecting the right fiber types and activate them with precise timing. For such purposes, different biofeedback systems can be used. One of the simplest systems is drawing a graphic line. Initially the line is horizontal and is asked to the patients to raise the line without altering the pace or direction of the studied movement. After the trial, the data recorded using sEMG was processed and it was shown that with an appropriate training paradigm, most skeletal muscle groups that are voluntarily activated could be trained to increase their coherence [17].

Based on the above, biofeedback can be used for simple purposes such as muscular rehabilitation after an injury, to more complex such as training the walking ability on people with nerval damage. Robotic-assisted body weight sported treadmill training (RABWSTT) has been used in gait rehabilitation for people with neurological conditions, It reduces fatigue of both the patient and the clinician what enables prolonged walking training. In addition, it provides guidance in the lower limbs movement, and this type of extensive exposure to a task-specific repetitive training helps promote reorganization of the primary motor cortex and functional outcomes can be improved in patients with neurological conditions like spinal cord injuries. In this case a Lokomat system was used for RABWSTT, which is a system used for gait training and rehabilitation, and it was combined with an EMG-biofeedback system. It is important that during the training, the patient has an active participation, because in the contrary, it will not improve as much as he should. That is why biofeedback is used. In this case, an audio feedback was generated if the muscle activation was less than 30% of maximal recruitment to encourage active participation during the stance phase of the gait cycle. In most neurological diseases high repetition of task-specific training with proper sensory feedback are essential elements for neuroplasticity after spinal cord injury. EMG-biofeedback systems enhance muscle contraction, and it has been proven that visual and audio feedback can promote muscle recruitment, increase muscle performance, and promote better improvement in lower limbs performance in people with neurological conditions. It was also used to monitor muscle contraction and enhance modulation in central nervous system without any adverse effect, what indicates that it is a feasible and safe way to promote participation of subjects during training. So, the use of EMG-biofeedback during RABWSTT is a valid treatment for promoting independent walking ability what can help improve independence in daily activities in people with SCI and enables them to enhance their walking endurance that promotes social re-integration [18].

Figure 2.6 *Lokomat RABWSTT System. Extracted from [18]*

Biofeedback has been used in children with spastic cerebral palsy (CP) which can suffer from muscle imbalance in the elbow joint and associated reaching movement incoordination. Different treatments such as neurodevelopmental treatment, strengthening or resistance training have been sued, but outcome results have been variable. These methodologies do not provide accurate or quantified biofeedback about muscle activation imbalance as it would in EMG biofeedback trainings, and in addition, undesirable compensatory movement cannot be controlled. Therefore, a hybrid model of EMG-VR system has been recently developed to provide accurate biofeedback and motivation to restore muscle imbalance between the triceps and biceps during elbow reaching movements in children with CP. This system is designed to provide a real-time visual feedback about triceps and biceps muscle activation patterns, during functional VR games. These games were designed to improve self-motivation, so they were enjoyable and improve functional strength [19].

Figure 2.7 *Elbow extension target activity Extracted from [19]*

This system provides real-time audio-visual biofeedback information about muscle activation which enabled the patient to acquire the motor task more consciously at the cortical motor control level, which is essential for biofeedback training. The game consisted of a blowing balloon VR game which was provided for triceps activation. The VR system provides a fun experience and improves motivation for children to perform elbow extension more implicitly minimising undesirable compensatory movement patterns. This game was used for increasing triceps muscle contraction while inhibiting biceps activity. It was compared the immediate effects of EMG biofeedback and EMG-VR feedback, and the latest methodology showed superior improvements [19].

***Figure 2.8** EMG-Biofeedback activity Extracted from [19]*

Biofeedback has been used for other type of diseases such as disorder that affect the pelvic floor. In such cases, biofeedback is used along with EMG sensors. This therapy is designed to improve abdominal push effort and its coordination during defecation, anal sphincter tone at rest and voluntary contraction of it. In this case a visual biofeedback display is used, seeing a visual color-coded graphic of anal canal and rectal pressure, patients' objective is to squeeze the anal sphincter muscles, without co-contracting the brace muscles and maintaining chest breathing. Using this therapy, patients which have increased rectal sensation can be trained to tolerate larger rectal volumes and patients with hyposensation can recognize lower rectal volumes and contract the pelvic floor muscles in response to rectal distension [20].

Another very common disease is low back pain. It can have different origins such as heavy lifting or a bad posture at your workplace. Therefore, health professionals often prescribe postural training as preventative measure to reduce forces on the lumbar spine when lifting. This therapy may lead to short-term changes in behaviour, but the long -term effects are uncertain. In this case biofeedback was provided by two inertial sensors attached to the lumbar spine. These inertial sensors measured the range of motion of the lumbar spine, and when the 80% of range of motion was achieved, an audio biofeedback was produced by a software. The 80% threshold was set because when it is exceeded, passive loading on the lumbar spine significantly increased. When the results were compared with a non-biofeedback group, a positive outcome was determined. The differences of flexion in the same task at the end of a 20 minute session the non-biofeedback group performed almost full flexion for lifting tasks, while the biofeedback group only 64% [21].

Biofeedback devices can be used for postural training to those that hold a static posture during prolonged computer work and can develop neck pain. The device used for the detection of the posture was Lumo Lift which is attached to the skin below the mid-clavicle. The subjects performed the same task with and without the biofeedback system, and after analysing the results, it is shown that once more, biofeedback task giver better results, however, long-term effects are not investigated [22].

Biofeedback-based videogames can be also used on improving balance challenges in autism spectrum disorder which a relatively high number of individuals struggle with postural stability. A visual-biofeedback system was implemented through a videogame, to perform a balance training that improves postural stability. Commercial videogames may not provide sufficient biofeedback to improve balance performance in autism, hence a prototype was developed with a simple interface and the possibility of hardening the difficulty to personalize the game depending on the capabilities and improvements of the subject. Existing devices such as Microsoft Kinect and Nintendo Wii balance board were used since these allow monitoring of balance and posture. To improve motivation, a training with the designed videogame and genuine Nintendo games was developed. After a six-week period training, participants improved balance times in the prototypes game. On average, participants almost doubled the amount of time that they were capable of standing at one foot. As conclusion, these findings suggest that visual-based biofeedback training improves balance in autism [23].

3. Analysis of the market

3.1. EMG

There are several EMG devices in the market, each of them has different properties and functions, hence it is important to evaluate the application that it will be destined for and then decide which fits better.

Delsys is a company which has different sEMG systems depending on the need of each client. It is mainly focused on research groups and hospitals. They offer systems with EMG and inertial sensors such as Trigno Research+, which can detect signal from up to 32 wireless sensors simultaneously and allows integration with third-party software plugins. There are different sensor depending on the application, for example, for gait analysis, fingers, face, motor control or dynamic movements. Each of them has a different sampling frequency, for example the gait analysis has a maximum sampling frequency of 4370Hz, and the face sensor up to 2222Hz [24].

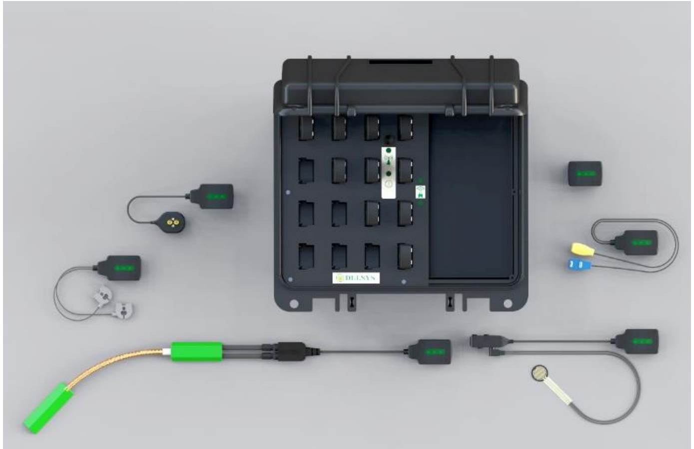

Figure 3.1 *Trigno Research+ EMG System Extracted from [24]*

Another system which also uses inertial sensors is Ultium EMG from Noraxon and also has up to 32 EMG channels available, but in this case the maximum sampling frequency is 4000Hz. However, the software allows to select high pass or low pass filters with predefined cut-off frequency. It allows connecting other sensors such as a hand grip dynamometer which measures grip strength. The combination of EMG recording with the dynamometer allows a better understanding of the injury. Noraxon's software is one of the most popular part of the system since it is extremely user-friendly and offers extensive reporting options [25].

Figure 3.2 *Noraxon Dynamometer Extracted from [25]*

A system which offers a very different point of view is Sierra Summit from Cadwell. It can be used with surface electrodes or needles. This system is the first EMG system which offers a fully integrated ultrasound system, what improves safety margins for the needles' injection. It also allows deep nerves stimulation. Compared to the Noraxon and Delsys systems, it offers only 12 EMG channels and does not include inertial sensors. It is highly customizable, fact that allows building the system that covers the needs that the therapist needs [26].

Figure 3.3 *Sierra Summit EMG System. Extracted from [26]*

A totally different approach is made by Myontec. This company designed textile products which incorporate EMG sensors. Its products are destined to athletes in therapy or in training session with a professional. They have different products such as a shirt or shorts, which have an 8 channel EMG device. Its latest product is a belt which is used for measuring low back EMG. This belt was developed for identifying factors behind low back pain, therefore can be used by athletes but also by workers who do repetitive lifting tasks [27].

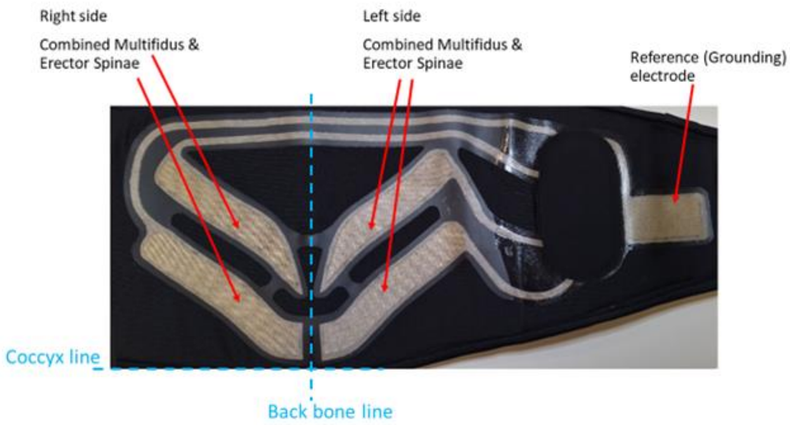

Figure 3.4 *Myontec EMG belt. It includes an shcematic of how it works. Extracted from [27]*

Even though lately many EMG wireless sensors have appeared, they were not designed for aquatic sports. Cometa systems offers the first waterproof EMG sensors. This approach is really interesting for assessing possible injury risk in swimmers. It also has an inertial sensor on board and up to 16 EMG channels with accelerometers. Its sampling frequency can be fixed up to 2000Hz. In addition, the EMG systems from Cometa have been developed to have a fixed delay from the acquisition of the signal to the analysis output. This delay can be removable when used with other EMG systems and it is so short that does not affect in any way real time monitoring of the signals [28].

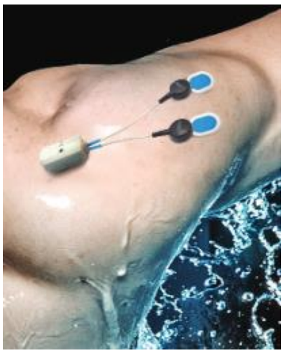

Figure 3.5 *Cometa waterproof EMG System. Extracted from [28]*

3.2. Biofeedback

One of the biggest companies in the biofeedback systems field is ThoughtTech. They offer measure instrumentation but what it is more important the biofeedback systems. They offer different hardware and software packages depending on the needs of the customer. Its EMG biofeedback systems use up to 3 EMG sensors which are connected to a computer. The signal received by the computer is used for controlling the games that they developed [3, 29].

Figure 3.6 *ThoughtTech Biofeedback software Extracted from [29]*

They have partnered with Zukor interactive, which have different videogames depending on the muscle rehabilitation purpose: strength, resistance, explosiveness, and other muscular capabilities. They offer games from different fields such as sports, cars or animals what helps them to increase the public and maintain the motivation among their patients. In the actuality they are developing some VR games, which will improve the immersion and the training even more. In its offer there is a tool which is used for balance analysis and training [30].

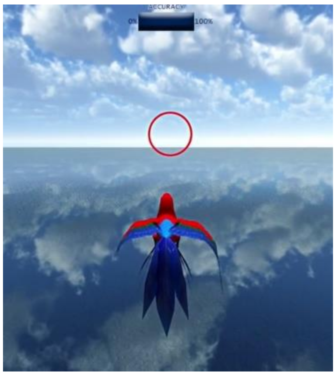

Figure 3.7 *Zukor Interactive Biofeedback game. Extracted from [30]*

Figure 3.8 *Zukor Interactive Balance platform. Extracted from [30]*

Accelerated Care Plus designed a sEMG tool which is used in the treatment of dysphagia. The tool is named Synchrony and enables using VR activities to visualize therapeutic swallowing activity. The different activities included in the tool are work-rest cycles, the diver, designed to target tongue base retraction, endurance, and specificity of lingual movement; the Kangaroo designed to target strength, duration, and progressive resistance; and the bow and arrow which is designed to target skill-based training and timing. This tool is continuously improving thanks to the feedback received from final users and dysphagia researchers [31].

Figure 3.9 *Kangaroo activity, designed to target strength, duration, and progressive resistance. Extracted from [31]*

4. Conception engineering

In the following section, the possible hardware used, the possible programming environment, the type of biofeedback system and the method for developing the system is discussed. The different options are shown in Table 3.1. The final solution is described into further detail in the detailed engineering section.

	Studied solutions
EMG Hardware	- Individual components, using resistances, capacitors, and operation amplifiers. - Integrated circuits and microcontrollers.
Arduino vs DataLog	- Record signal at different times. - Record signal at the same time.
Type of Biofeedback system	- Visual (LED). - Acoustic (Sound). - Complex (Game).
Biofeedback programming environment	- Python. - C
Biofeedback development method	- Integral development of the game - Downloading a game and adding biofeedback needed features

Table 4.1 *Studied solutions for the hardware, programming environment, type of biofeedback system, and biofeedback development method.*

4.1. Study of solutions

4.1.1. EMG Hardware

For the EMG system there were two options, make us of protoboard and individual components or using integrated circuits and microcontrollers such as Arduino.

The use of a protoboard and individual components is a more complex solution. It would have been needed a highly study of all the properties of an electronic circuit, such as the gain to get a signal, the frequencies in which the EMG signal works, and therefore a complex study of all the components needed to perform the different filters needed. An advantage of this solution is that it has the option of including an Isolation phase, what would give the prototype the possibility of passing official tests and laws such as EN-60601, to be certified as a medical device.

In general, it would have made the project much larger, and taking in consideration the limitations explained before, it would have difficult the development of the biofeedback system.

To test all these components and the correct implementation of the circuit it would have been needed many other instruments such as oscilloscopes and waveforms generators, that would have made more difficult the task due to covid limitations, or buying a all-in-one instrument such as Analog Discovery 2, which would have make a lot more expensive the project.

Arduino is an Open-source electronic prototyping platform which enables users to create interactive electronic objects. In this case I was needed of a basic microcontroller board and a board which integrated the EMG circuit. This solution provided me the possibility of dedicating more time to the biofeedback system, which is more important in the project. Once I received the two components needed, I only had to develop the software, which was not difficult thanks to Arduino software that provides an easy environment to control this microcontroller.

4.1.2. Arduino vs DataLog signal acquisition possibilities

Comparing the two different signals is difficult, since it is important to record the same amount of data, with the same sampling frequency, and the acquisition points as near as possible. I had two options. Recording the signal at different times allows to use the electrodes exactly at the same point, but other problems appear. It is impossible to reproduce the same exact signal two different times, and it is impossible to superpose the two signals because the contraction would take place at different times.

Recording the signal at the same time has the opposite advantages and disadvantages. The electrodes cannot be at the same point, but the signal is the same and it allows me to superpose them in order to compare the signals from the different devices.

4.1.3. Type of Biofeedback system

The biofeedback system could have been implemented using mainly a visual, an acoustic or a more complex system.

Visual biofeedback can be applied using binary systems such as a simple LED, that is powered-on or powered-off, a digital system showing numbers depending on the activation level or a graph that shows continuously the signal. It has the disadvantage that the patient needs to be in front of a screen, therefore when the biofeedback is applied to rehabilitation procedures in which the patient must move along the room, it makes difficult to perform it [32].

Acoustic biofeedback is based on an individual sound or multiple sounds that could be used for playing a song. It can also be binary, which indicates if the activation threshold has been reached. An advantage of this modality is that is can be used in daily activity, allowing to give feedback to the patient only when is needed, to make sure that he is focused on his daily tasks. It also solves the problem of visual biofeedback because the sound can be listened even though the patient is far from the feedback source [32].

Complex biofeedback makes use of both previous modalities. This system is generally implemented in different types of games, fact that makes the rehabilitation procedure more entertaining and motivating. The main disadvantage is that it is more complex to carry out since a game must be programmed [32].

4.1.4. Biofeedback programming environment

Python has large library of built-in functions library which facilitates programming. For my purpose, it has *pygame* library which facilitates programming a game and its controls. In general, it is a simple language and environment, and it is easy to learn and write and read.

C is a more complex language, with a limited number of built-in functions. The syntax is harder; therefore, it is more difficult to learn, write and read.

4.1.5. Biofeedback development method

There were two possibilities for developing the game, the entire development of it, or downloading a project already done and implementing the functionalities needed for the biofeedback application.

The full development of the game is translated into the need of great knowledge of the programming environment and library needed. It is a long project with several difficulties which would lead to a high time consumption.

Using an already existing game and changing the controls is a simpler solution. It also needs the understanding of the library and the program itself but reduces a lot the complexity of all the procedure.

4.2. Proposed solution

Considering the above the integrated circuit and using microcontroller has been the option for developing the EMG system. It is a very simple method which reduces time consumption. The main disadvantage is that it does not include an isolation phase, therefore it cannot be recognized as a medical device, but since we are doing a prototype and we are not testing it in patients it is not that important. In addition, in case 9that we finally decided to test in a clinical trial, an isolation phase could also be implemented by means of an isolation amplifier.

The approach used for comparing the signal from different devices was recording at the same time. It is more important to record the same signal, even though this implies slightly different acquisition points than recording it at the exact same points. The electrodes can be placed at a very short distance therefore obtaining signals from very near points.

The biofeedback system chosen was the *complex biofeedback*. As I already said there is a high abandonment of rehabilitation. One of the main reasons is that patients consider it boring. By using a game, it would make the treatment procedure more interesting and motivating, which occasionally would lead to reduce rehabilitation dropping. As shown in [19, 31] in which virtual reality exercises are used as biofeedback for swallowing disorders treatment, virtual reality exercised, and games in general enhance the treatment outcome. This is possible because as the patient improves the clinician can manipulate the game and progressively make the tasks more therapeutically challenging. In addition, when talking about muscle rehabilitation therapy, it requires muscle exercises to be carried out on a repetitive basis, typically for 15-30 minutes periods, 2 to 5 times a week. The repetitive nature of this therapy, as already has been said, can cause motivation problems with the subject and as consequence, adherence to these types of rehabilitation programs can be low which leads to a worse rehabilitation outcome [33]. In addition, this complex biofeedback opens a wide range of therapeutic possibilities. Depending on the necessity a game or another can be used, being each one developed for different purposes such as resistance, maximum strength, or control [31].

The programming environment and language elected was Python. It is a more affordable solution, since *pygame* library includes all the necessary functions for developing a game. It is more flexible, and while C is more dedicated to hardware control program, Python allows better solution for developing software. In addition, I am much more familiar with Python language than with C, which will facilitate the development of the game.

Deciding if I was going to develop the whole game or going to download a project and change the controls, was mainly based in two factors: time and my current programming skills. Even though I am much more familiarized with Python, I had not never made use of *pygame* library, so it would become a challenge to develop the game. It would also imply a higher time consumption since first I would have need to understand the library and functions available and the implement them to develop the game. But downloading the game and changing the controls would have been a very simple solution. Therefore, I decided to join both solutions, I looked for a project that was explained on a video, and I followed the tutorial for developing it. Finally, I only needed to change the controls and add a pop-up window. This solution brought me the opportunity of getting in contact with *pygame* library, but without consuming a lot of time.

5. Detailed engineering

This section will be divided in three parts. First, the EMG system development, then the signal comparison will be explained, and finally the biofeedback system implementation.

5.1. EMG system

The EMG system was developed using Arduino-like microcontrollers. In my case I was needed of two main components: the basic board and the EMG-shield.

The basic board could have been any which fitted in the EMG shield. Therefore, first I looked for the appropriate EMG shield. After an extensive search on Internet, I found Olimex EMG-Shield. This shield allows Arduino like boards to capture EMG signals. In the distributor's web it indicates that this shield opens new possibilities to experiment with biofeedback and shows some projects, so it seemed perfect for my purpose. The datasheet of the product included some important information such as the total gain, which is 2848, and the filters used. Some interesting filters which the shield includes are two high-pass filter and a 3rd order *Besselworth* filter [34,35].

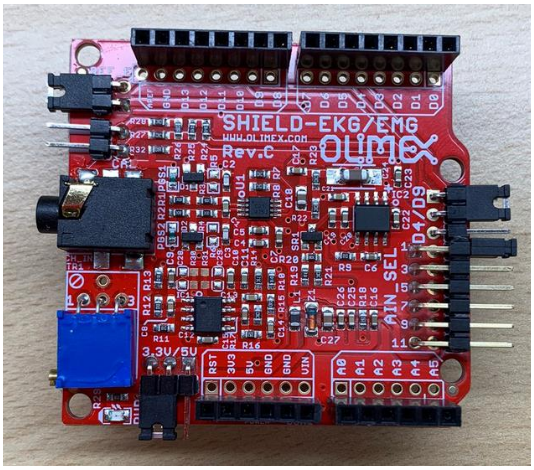

Figure 5.1 *Olimex EMG Shield. Personal source*

Once I decided the shield for the EMG, I had to get the main board. The shield's datasheet recommends some boards, so I decided to use one of the indicated to obtain the best possible signal. The final board was the Olimexino-328 from Olimex [35].

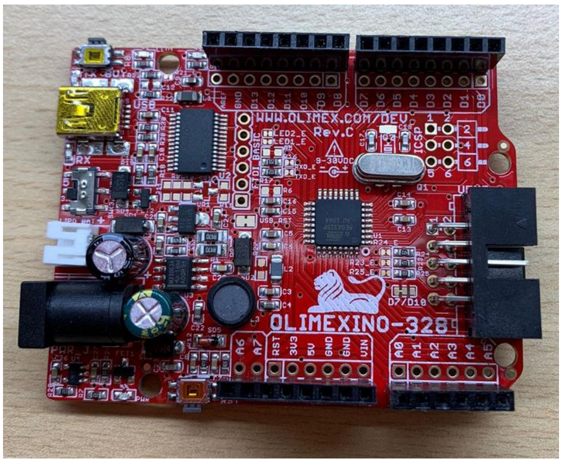

Figure 5.2 *Olimexino-328 board. Personal source*

For the EMG I needed the cable electrodes and electrodes. Olimex also provides the cable electrodes needed for its EMG shield. It has the positive and negative poles and a ground cable. The electrodes can be bought in many medical devices stores but I did not have to buy them since my project director Jose Luis Parreño provided me the electrodes that I needed.

Figure 5.3 *Cable electrodes in the upper part and electrodes in the lower part. Personal source*

The assembling of the whole system is easy. The EMG-Shield fits perfectly on the Olimexino-328, so I only had to insert one on top of the other, and finally connect the cable electrodes to the EMG shield. The final assembled system is shown in Figure 5.4.

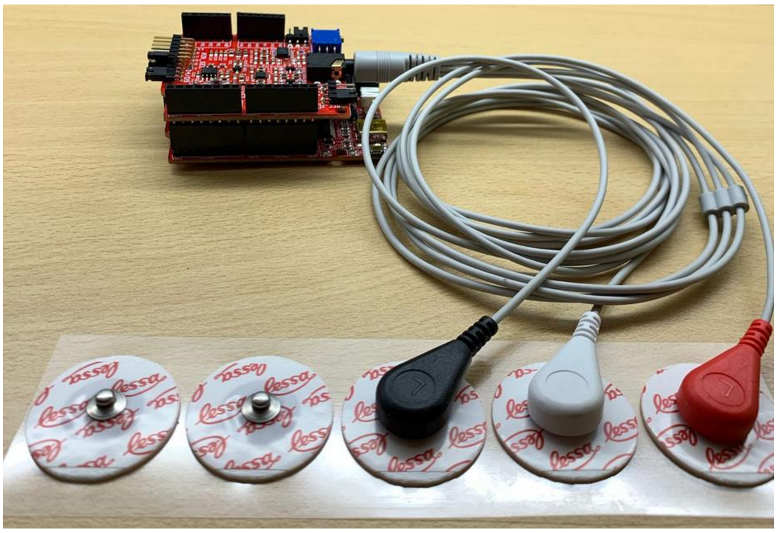

Figure 5.4 *Assembled EMG System. Personal source*

Once the system was assembled, I needed to write the code which will acquire the signal. The Olimexino-328 board is compatible with Arduino IDE, what makes easier the programming step. An important point in this phase is to fix a sampling frequency because in case that it is not fixed, each signal point could have different temporal separation, what will lead to a wrong recording signal. The maximum sampling frequency that the system allows is 256Hz, but due to some problems that I will explain in the Biofeedback section, I decided to finally fix it at 120Hz. In this step it has not been performed any filtering or rectification signal, only the baseline has been corrected. In figure 5.5 the acquired signal can be seen. The amplitude of the signal has not been modified either. Knowing that a common EMG signal is between -3mV and 3mV, some processing steps may be important, but at this point I want to show the raw signal, which is the original signal without any filtering or post-processing step.

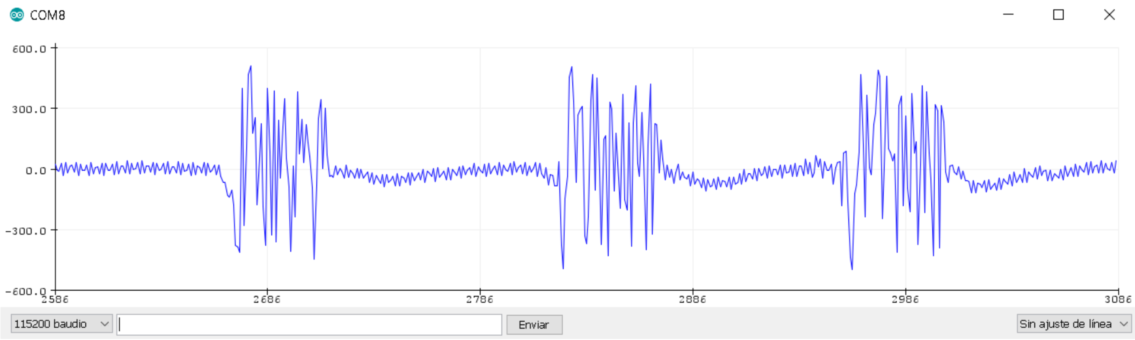

Figure 5.5 *Raw EMG signal in Arduino IDE. Personal source*

The code used for recording the signal can be found in Appendix 1.

5.2. Arduino signal and DataLog signal comparison

To compare the signal, it should be as similar as possible. Therefore, I connected and recorded the signal in my biceps at the same time, as shown in Figure 5.6. Since I want to compare the same signal, the electrodes from the two different devices must be as near as possible one from the other.

I connected the positive and negative poles to the electrodes located in the muscle, and the ground, which acts as the reference, in my elbow bone because it should be placed far from the EMG detecting surfaces, on an electrically neutral tissue.

Figure 5.6 *Electrodes placed for Arduino and DataLog signal recording at the same time. Personal source*

When comparing two signals it is important that they have the same sampling frequency. The DataLog software allows to set sampling frequencies up to 1000Hz, but my prototype only allows 256Hz. The DataLog software has a list of predefined sampling frequencies, in which the higher that my prototype can achieve was 100Hz. So, I selected it, and in the Arduino script I changed the interval which controls the sampling frequency.

Figure 5.7 *DataLog EMG Device from Biometrics Ltd. Personal source*

Arduino reads real time signals, hence I had to store the values in a file, which will allow me to compare them later. To do so I wrote a code in Python which reads the signal and stores it in an array, which I can save in a *txt* file. For the connection between Arduino and Python, I made use of the library *Serial* which has functions for detecting the Arduino board and reading its signal in real time. I fixed how many samples I wanted to have, and I started the recording. Code can be found in Appendix 2.

At the same time, I started recording the signal in the DataLog software. This software allows to select some parts of the signal or make some post-processing steps, but as I wanted to compare the raw signal, I did not make use of them. The signal was saved in a text file.

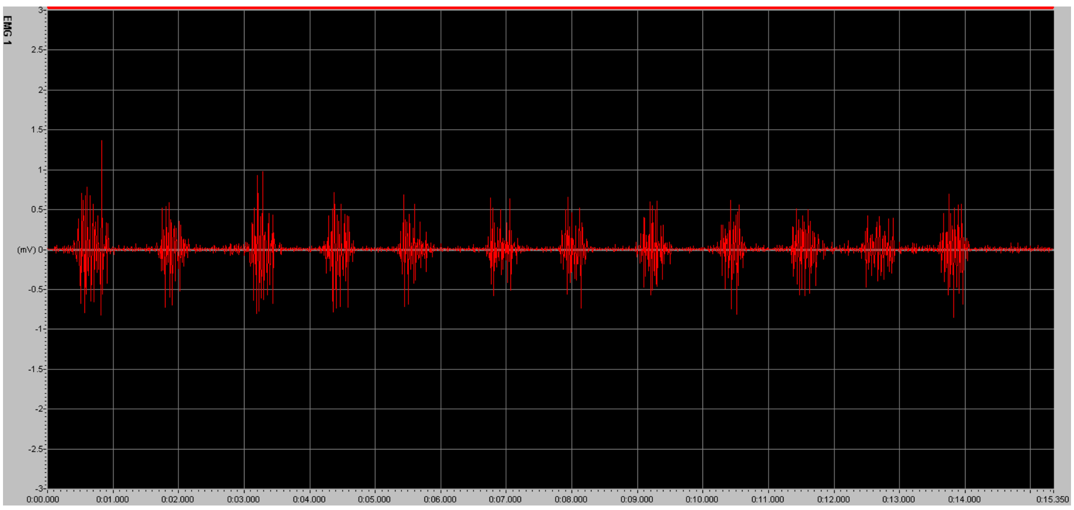

Figure 5.8 *DataLog EMG signal seen in its own software. Personal source*

For comparing the signals, I had to write some code. In this case I made use of Python in Anaconda's platform. First, I had to open the csv file with Arduino data, and save the values in an array, and do the same with the text file from the DataLog.

Once I had the two arrays, I had to delete some values from the beginning of both signals, and from the end of the DataLog signals, because they had different length as consequence of the time span between when the Arduino signal recording and the DataLog signal recording started and stopped. I used the *pop* function which is included in Python.

The next step was normalizing the intensities of both signals. The maximum value of my prototype signal was 491.0, which was very larger than the DataLog signal, with a maximum value of 0.91, as it can be seen in Figure 5.9 & Figure 5.10.

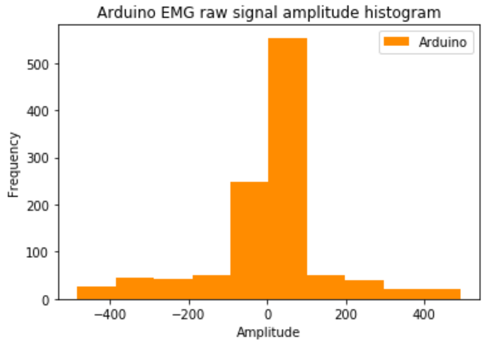

Figure 5.9 *Arduino EMG signal histogram before normalizing intensity. Personal source*

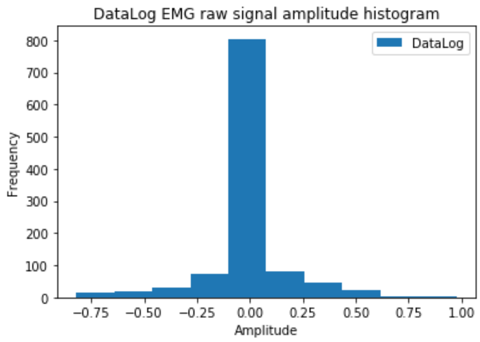

Figure 5.10 *DataLog EMG signal histogram before normalizing intensity. Personal source*

The selected method was applying an intensity normalization using z-score. Z- scores were calculated for each array of EMG signal using their respective means and standard deviations, thus obtaining amplitude values constrained between the two same limits.

$$A' = \frac{A_{a,d} - \mu_{a,d}}{\sigma_{a,d}}$$

Equation 1. Z-score formula. A' *refers to normalized intensity, a to Arduino and* d *to DataLog*

After the normalization step the resulting histogram is shown in Figure 5.11

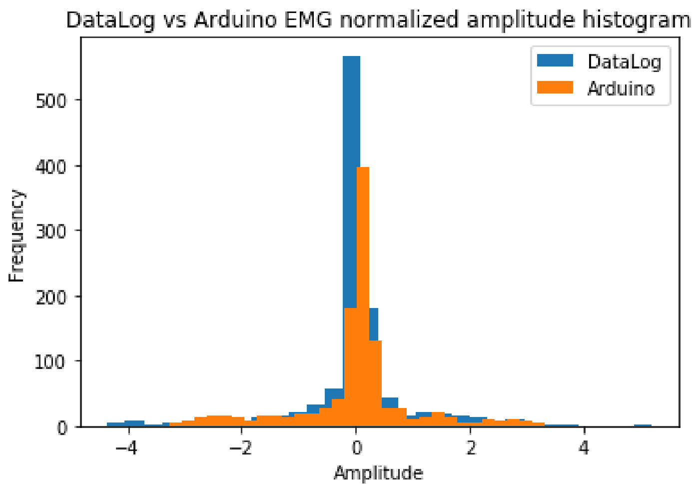

Figure 5.11 *DataLog vs Arduino EMG normalized amplitude histogram. Personal source*

This histogram clearly shows that the signals have now the same amplitude values. If we focus on the Arduino histogram, we can see that the values which have higher frequency are greater than 0. This fact indicates that the baseline is not perfectly corrected, and that the signal at muscle relaxation could be improved by applying digital filters. These two facts can be seen in Figure 5.12 too.

Once both signals were normalized, I plotted them using *matplotlib* library. The resulting plot is shown in Figure 5.12**.**

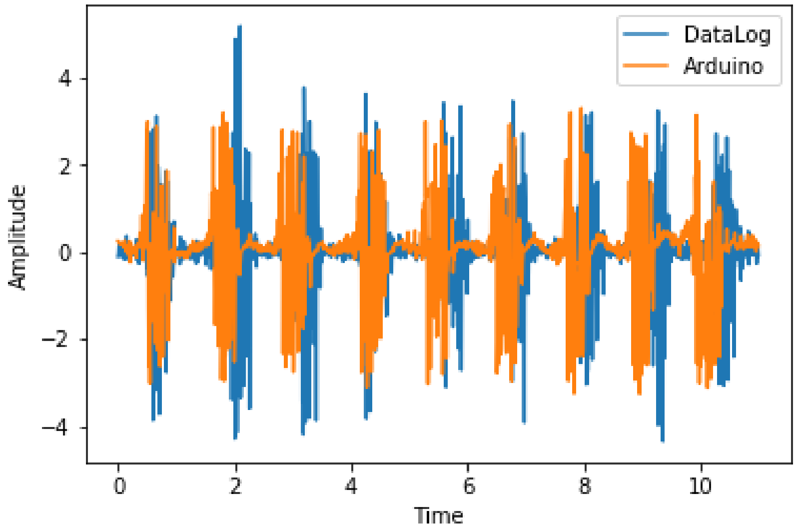

Figure 5.12 *Arduino vs DataLog signal. High amplitude corresponds to biceps voluntary contraction. Personal source*

As it can be seen the signal is really accurate, taking in account that the Arduino signal do not have any type of filter and the DataLog signal has a digital filter and a more exactly baseline correction. The differences in amplitude of each point may be consequence of the different position of the electrodes, which can cause different signal recordings.

But to have more exactly comparison I performed a T-test of both signals before the normalizing step and after it. The t-test was performed thanks to the *stats* package included in *scipy* library, which has a t-test function. The p-value before the normalization step was $p=0.86$, and after the normalization step $p=0.99$. In statistics, in general, a $p>0.1$ shows that there are not significative differences between two arrays of data. In my case, both values are much higher than 0.1, and really close to 1, which will mean that it is the same data. Once more, this p-values show the good behaviour of the Arduino prototype compared to the DataLog.

After comparing graphically both signals and performing a T-test, it can be concluded that the signals are similar. Probably if some post-processing steps were performed, it would give better results, especially in the resting regions.

All the code can be found in Appendix 3.

5.3. Biofeedback system

Once I decided the type of biofeedback which I wanted to implement, I had to think which game could be motivating and entertaining. I thought that a known game would make more interesting the rehabilitation process. My first option was Super Mario Bros, which is a classic game, and many people know. But it has some problems: even though it is a simple game, it has many controls which I cannot implement, since I only have an EMG system, so I discarded this option. Then I looked for some game that has only one control, and I remembered Flappy Bird. In this game the player controls a bird, which must fly between columns of green pipes without hitting them. It has only one control, which originally was taping the smartphone screen, and in my case could consist of setting a threshold, and once the EMG signals overcomes it, the bird would eventually fly. This type of games is useful in rehabilitation for targeting maximum strength and explosiveness, since it has to activate the muscle as much as the patient can and as fast as possible.

For creating a game, it is very important to understand how it works. There are two main concepts: First, any game basis is an image that is continuously updating, and second there is a game logic that indicates which will be the following image.

I used *Visual Studio 2019* which is a development environment that allows using Python for programming. I had to install Python package, and some Python libraries needed for developing the game. The most important one was *Pygame.* It is a cross-platform set of Python modules designed for writing video games. It is free; therefore, anyone can create open source or commercial games using it. For installing this module, I had to open the command prompt on windows and write *pip install pygame*. I looked for a Python project which included a video-tutorial so I can step-by-step learn to create the game, and later change the controls.

Pygame has two main functions, init() and quit(). They start and close the game respectively, and between each of them the game logics must be written. The main element is the display surface. It is the canvas where the game will be seen. The width and height of the display surface must be indicated. This command is followed by the game loop which will include all the games logic and the canvas update. The game loop must include another loop which checks if any event has taken place. An event is anything that happens in the program, could be a mouse movement, a key click, or a signal read.

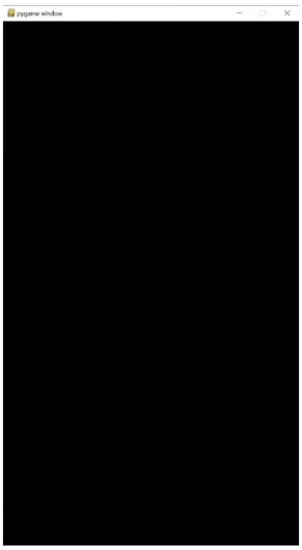

Figure 5.13 *Empty Display Screen. Personal source*

Another key factor in a game is the number of times in a second that it is updated. This is called frames per second (fps), and it affects the speed of the game. It is important to fix a maximum value, because in the contrary, the game will update the image depending on the difficulties for processing it, and in case that it is a very complex game, it can change from one point to another, affecting the speed of the player. Flappy Bird is a simple game, so it will run fluent in most computers. Therefore, I set the fps at 120, meaning that in one second the image will be update 120 times. In case that the computer is not powerful enough, the game will have as many fps as it can.

Once the screen and the game loop have been created and the fps fixed, the images that will be in our screen must be designed. The images that are continuously changing are called surfaces. Using these surfaces, will make the elements in the screen move, giving the fluent motion which characterizes games. There is a function which makes this possible.

The main elements of Flappy Bird are the bird and the pipes which it must avoid. The basic logic of this game is checking for collisions between the bird and the pipes, but screen do not have these options, therefore rectangles are going to be used. With rectangles it is possible to check if two rectangles have collided, so if the bird and the pipe as a rectangle are modelled as rectangles, collision events will be possible to be checked. Before creating the collisions' function, all the elements needed will be created which are the bird and it's control, and the pipes.

The bird has two movements, falling or flapping. There will be a common variable for both movements, *bird_movement*. For the falling movement gravity will be implemented. Since surfaces have the origin at the top-left of the screen, and the positive side is below the origin, as shown in Figure 5.14, to make the bird fall, the gravity, which has a constant value, has to be summed to *bird_movement* inside the game loop. The movement effect for the rectangles is created moving its horizontal and vertical centre. For the bird's case, since it is moving up and down, it only changes the vertical component.

.

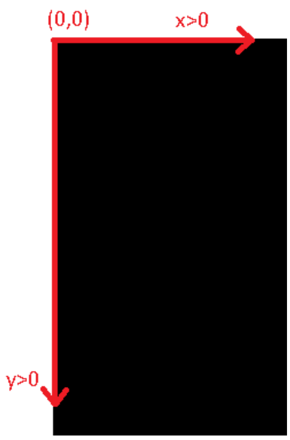

Figure 5.14 *Diagram which shows origin and direction of increasing values in pygame surfaces. Personal source*

The flapping movement was originally controlled using the keyboard, but for the biofeedback application I had to change it. In this case, since I am using EMG and it measures the muscle activity, I can use it to control the flapping of the bird. In the game loop, an *if* loop is used which will compare the EMG signal, read by Python from Arduino, using the same function that in the EMG section, with a fixed threshold. When the signal is higher than the threshold the code will execute some lines, which will make the bird jump. These code lines use the same bird movement variable than the gravity, but in this case, the value must be negative, as explained in the diagram. It is important that before summing the value to *bird_movement*, its value is reseted and setted to 0, because if it is not done, it will have positive value as consequence of gravity continuously summing, the bird will not be able fly and when eventually, gravity is counteracted, each jump will have different height.

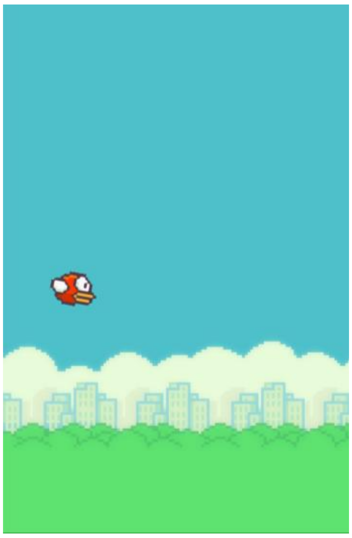

Figure 5.15 *Screen with background and bird implemented. Personal source*

Now it is time to add the pipes. For the pipes it has been performed similar as for the bird, using surfaces and rectangles, but in this case, the pipes must move to the left, and must spawn regularly after a constant time. So, list of rectangles for the pipes and a timer have been used. In this timer it must be indicated what it must happen, for this case creating pipes, when a fixed time has passed. Then, a function which creates the pipes, locates it and makes them move is needed. Since eachs pipes has to have different height, a list with different heights is needed. In the function which creates the pipe, a random function which picks a height from the list and uses it for generating the new pipe is used. This will generate the bottom pipe, but this game also has a top pipe at the same position, so it is necessary to subtract, from the bottom pipe, the distance that will separate both pipes, and finally it must be flipped.

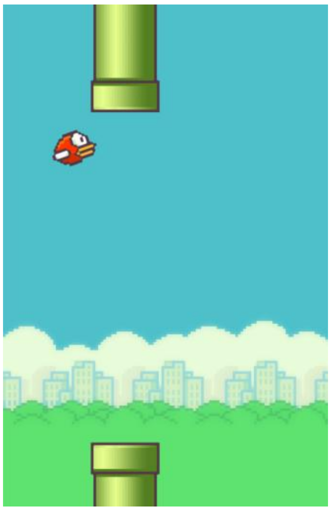

Figure 5.16 *Screen with background, bird and pipes implemented. Personal source*

The next step is adding the collisions. Since rectangles are being used, there is a function which checks if there is a collision between two rectangles. It returns *True* or *False* if there has been a collision or not, respectively. We also want to check if the bird has touched the limits of the screen, so we need another condition. This last collision is implemented by checking if the bird has surpassed the limits of the screen.

The collision's logic will be used for controlling if the game is active or not. When a collision happens, the function will return a *False* value, which will make the game over screen to appear, and while a collision does not take place, it returns *True.* Since the active game variable has two different values, which correspond to *True*, if the game is active and *False* if it is not, what happens when the game is active or inactive must be controlled. When it is active, all the game's logic, such as variables, bird's movement, collisions, and pipes spawn, must be controlled. When it is inactive, it will appear the *Game Over* screen all the variables will be reset and cleaned and when the signal overcomes the threshold, the variable will be changed to *True,* and the game will start again.

Finally, some details were added such as the scoreboard and sounds for flapping or after correctly passing a pipe.

Figure 5.17 *Game Over screen. Personal source*

For setting the aforementioned threshold, it was necessary a pop-up window with an input box. This will make easier changing the threshold, because without it, the therapist would have to change it in the code, which will make difficult to use the program. In this pop-up window there are also the instructions of the game for the patient. T*kinter* library has been used, which has a function that opens a pop-up with an input box and saves the values in a variable. This function saves strings, which cannot be used for comparing numbers, hence it must be transformed into float. It has been also implemented a loop that checks if the introduced value is a correct number. In case that it is not, another *pop-up* window appears, which remembers that the introduced in the box must be a number.

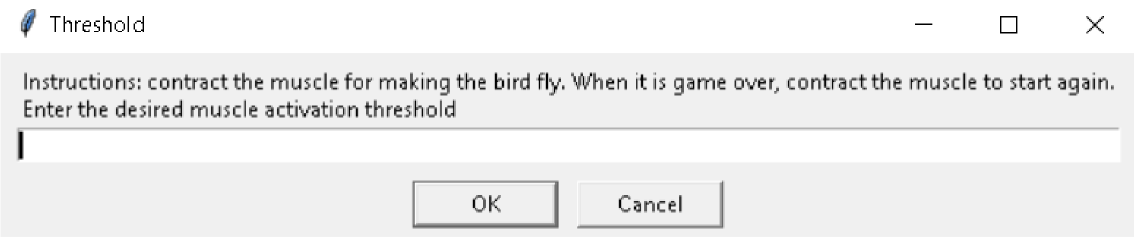

Figure 5.18 *Pop-up box for fixing the threshold. Personal source*

The signal was read using the *Serial* library. It also reads strings, so it had to be transformed into floats. When the code was finished, it was tested, but it had some delay problems, which caused that the signal was read seconds later, making impossible to play the game. To solve it a command that clears Arduino's buffer had to be used. If the buffer is not cleaned after some time, all the previous values are stored, this makes the microcontroller to go slower. The sampling frequency and the frames ratio was changed too since it also made the program to go slower.

6. Experimental validation

To properly work with the EMG and Biofeedback system, it is important to evaluate the whole system and try to fix possible problems that it must use the best possible product when it is necessary. The most important point for a useful therapy, is setting the correct threshold for each case, it should be reachable, but it should suppose an effort to the patient.

In the first version, before running the Flappy Bird code, the Arduino code was run, and the serial plotter windows was opened. Then a contraction was performed in order to see the maximum achievable value. This value was used to establish the threshold. This method was slow and had small precision, therefore some changes were added to the code. At the beginning of the code, an array saves the data from Arduino. During the time that the data is being saved, the subject must perform some contractions. The maximum value recorded will be used for setting the threshold. From my experience during the validation of the system, a 60% of the maximum value is a good threshold. If it is too low, it will detect lower peaks and the bird would fly unintentionally. If it is too high, a very high contraction should be needed, and it will be frustrating for the subject.

During the validation of the system some changes in the code were performed. Initially the pipes were very close, which made very difficult to play with the EMG system. They were separated to make the game more achievable. The spawn time of the pipes was also adjusted.

In addition, the fps were adjusted. With a high level of fps, the game was very fast, what increased a lot its difficulty, and taking in account that it has therapeutical purposes, it may be motivating but not frustrating. The fps adjustment also helped removing the delay problem indicated at the Detail Engineering section, which was one of the main problems that it had in the first version. As the fps in Python were adjusted, the sampling frequency in Arduino too, to avoid delay problems explained in the Detailed Engineering section.

Other problems cannot be resolved due to the limitations that Python and *pygame* has. The bird and the pipes are created using rectangles; therefore, they cannot have its original shape. The consequence of this is that if the bird passes very near of the pipe, a corner of the rectangle collides with the pipe, but visually it does not collide, and it is detected as a normal collision, so the loss screen appears.

7. Project execution schedule

In this section the timing of the project will be explained using different tables and diagrams. The documentation times has not been taken in account.

7.1. Work Breakdown Structure

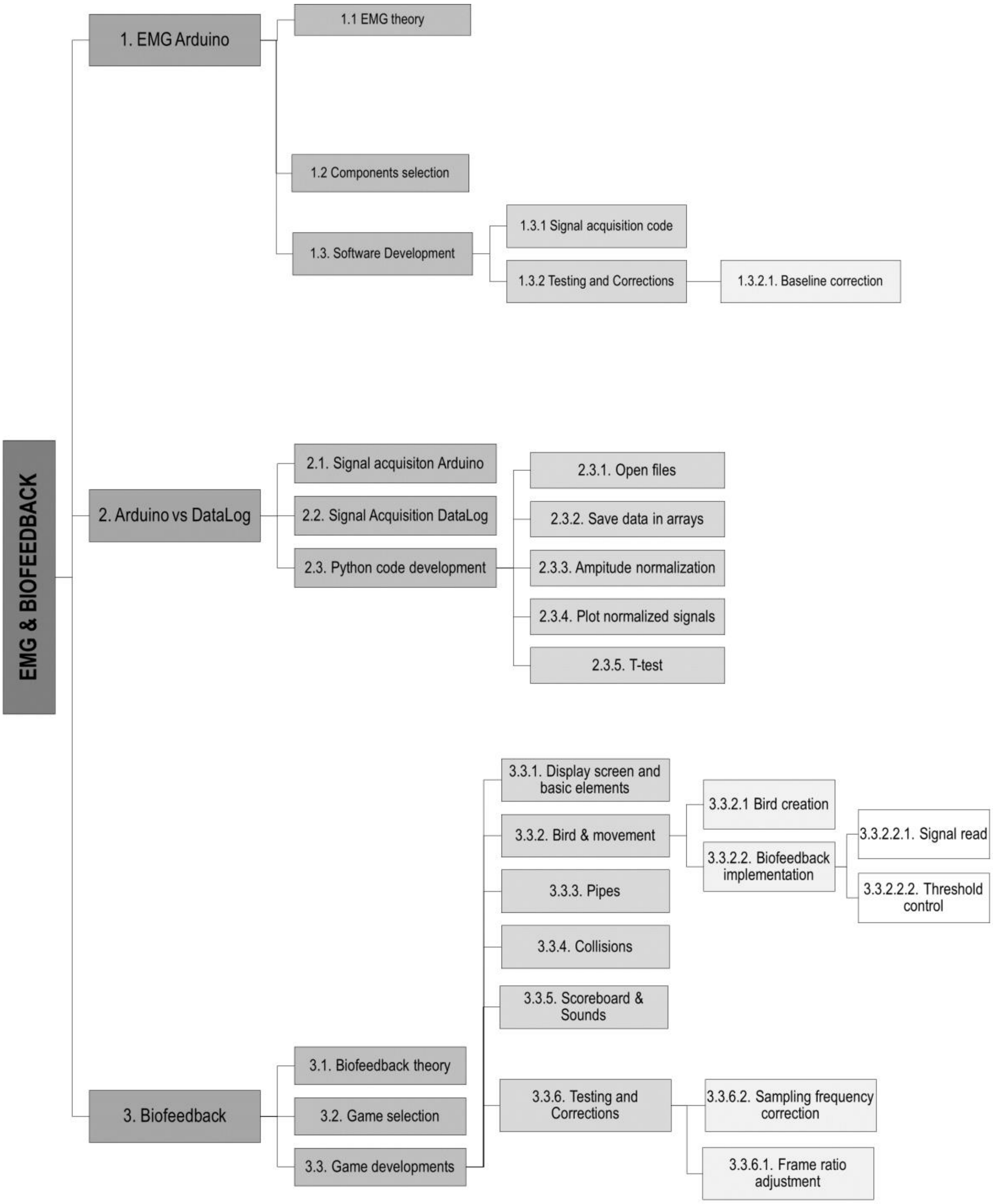

Figure 7.1 *WBS of the project. It has 3 main parts. Personal source*

7.1.1. WBS Dictionary

The following table includes the activities present in the WBS with its description.

Nº	Name	Description
1.	*EMG*	It includes all the necessary steps for correctly developing the EMG.
1.1.	*EMG theory*	Introduction to the EMG, how it works, and electrodes placement.
1.2.	*Component's selection*	Study of the components needed and final selection
1.3.	*Software development*	Development of the program needed for reading the EMG signal.
1.3.1.	*Signal Acquisition code*	Development of the functions needed for acquiring the EMG Arduino signal.
1.3.2.	*Testing and Corrections*	Testing of the acquisition program and correct possible errors.
1.3.2.1.	*Baseline correction*	The baseline was not at the 0. In this step it was corrected.
2.	*Arduino vs DataLog*	In this phase, the steps needed for comparing Arduino prototype and the DataLog signals have been developed.
2.1.	*Signal Acquisition Arduino*	To acquire Arduino's signal, it was needed to develop a Python program which saved the values in a CSV file.
2.2.	*Signal Acquisition DataLog*	Using DataLog's software the signal was acquired at the same time than the Arudino one. The signal was saved in a text file.
2.3.	*Python code development*	To compare the signal a Python code was needed.
2.3.1.	*Open files*	It was needed to open the CSV and TXT files.
2.3.2.	*Save data in arrays*	The values must be saved in two different arrays.
2.3.3.	*Amplitude normalization*	The ampitudes were different, therefore it was needed to normalize the intensity.
2.3.4.	*Normalized signals plotting*	The normalized signals were plotted. Two different plots were used, first an Histogram to see the normalized values, and then a Time vs Ampitude plot.
2.3.5.	*T-test*	A t-test was performed to obtain a p-value which indicated a statistical comparison result.
3.	*Biofeedback*	The biofeedback system was developed.
3.1.	*Biofeedback theory*	Introduction to biofeedback. Needed to select the type of biofeedback and which muscle characteristics were wanted to train.

3.2.	*Game selection*	A game was selected as the biofeedback system. The biofeedback game used was selected in this phase.
3.3.	*Game development*	The game was needed to be developed
3.3.1.	*Display screen and basic elements implementation*	The display screen and the basic elements such as the background or loops were implemented.
3.3.2.	*Bird & movement*	The bird and the variables and loops needed for its movement were created.
3.3.2.1.	*Bird creation*	Create the elements needed for the bird
3.3.2.2.	*Biofeedback implementation*	The biofeedback control was added in this step.
3.3.2.2.1.	*Signal read*	It was needed to read the signal from Arduino and use it for controlling the bird.
3.3.2.2.2.	*Threshold control*	It was needed a threshold that made the bird fly. A pop-up box was used.
3.3.3.	*Pipes*	The pipes which the bird must avoid were added in this step.
3.3.4.	*Collisions*	Since the game consists of avoiding the pipes, collisions were between the bird and the pipes and screen limits were added in this step.
3.3.5.	*Scoreboard & Sounds*	The scoreboard was added to make the game more challenging. Some sounds were added too.
3.3.6.	*Testing and corrections*	The biofeedback system was tested, and some corrections were made. Initially it had delay, which made impossible playing.
3.3.6.1.	*Frame ratio adjustment*	The frame ratio was adjusted to reduce delay.
3.3.6.2.	*Sampling frequency correction*	The sampling frequency was also adjusted, to fit it with the frame ratio.

. ***Table 7.1*** *Dictionaries for each of the activities carried out in the project and references in the WBS.*

7.2. Task sequence matrix

Prior to the time analysis, we will determine the time associated with each activity. The times are calculated in hours and for two people, given that the team is made up of two people, each activity will be associated with certain hours that must be completed between the two members.

The pessimistic time corresponds to the time that, in the worst case, we can take to do that activity. The normal time is the estimated time it would take to carry out the activity and finally the optimistic time is how long it would take us to carry out the activity under favourable conditions. By including the 3 values in the following formula we can calculate the final time:

$$Time_{expected} = \frac{T_{optimist}+T_{pessimist}+4 \cdot T_{normal}}{6}$$

Equation 7.1 *Used for calculating the Expected time of each activity*

The order used to arrange the task sequence matrix has been done according to the activities in alphabetical order. Each activity has its corresponding number, associated to a dictionary and its previous and consequent activities. Finally, we have the durations of the activities and the final and initial times that will help us to represent the GANTT and PERT.

Number	Activity	Pessimist (h)	Normal (h)	Optimist (h)	Expected (h)
1.1.	A	38.00	27.00	20.00	27.67
1.2.	B	5.00	3.00	1.00	3.00
1.3.1.	C	20.00	10.00	5.00	10.83
1.3.2.	D	14.00	6.00	3.00	6.83
2.1.	E	2.00	1.00	0.50	1.08
2.2.	F	2.00	1.00	0.50	1.08
2.3.1.	G	1.00	0.50	0.25	0.54
2.3.2.	H	1.00	0.50	0.25	0.54
2.3.3.	I	2.00	1.00	0.50	1.08
2.3.4.	J	1.00	0.50	0.25	0.54
2.3.5.	K	2.00	1.00	0.50	1.08
3.1.	L	33.00	25.00	18.00	25.17
3.2.	M	6.00	3.00	1.00	3.17
3.3.1.	N	1.00	0.50	0.25	0.54
3.3.2.1	O	10.00	8.00	4.00	7.67
3.3.2.2.1.	P	6.00	4.00	2.00	4.00
3.3.2.2.2.	Q	4.00	2.00	1.00	2.17
3.3.3.	R	24.00	16.00	12.00	16.67
3.3.4.	S	38.00	30.00	18.00	29.33
3.3.5.	T	7.00	5.00	3.00	5.00
3.3.6.	U	30.00	18.00	2.00	17.33

***Table 7.2** Used to calculate the final time for each activity defined in the WBS.*

Number	Activity	Predecessor	Consequent	Duration	Start	End
1.1.	A	... start	B,F	27.67	0.00	27.67
1.2.	B	A	C	3.00	27.67	30.67
1.3.1.	C	B	D	10.83	30.67	41.50
1.3.2.	D	C	E,P	6.83	41.50	48.33
2.1.	E	D	G	1.08	48.33	49.42
2.2.	F	A	G	1.08	27.67	28.75
2.3.1.	G	E,F	H	0.54	49.42	49.96
2.3.2.	H	G	I	0.54	49.96	50.50
2.3.3.	I	H	J	1.08	50.50	51.58
2.3.4.	J	I	K	0.54	51.58	52.13
2.3.5.	K	J	… end	1.08	52.13	53.21
3.1.	L	… start	M	25.17	0.00	25.17
3.2.	M	L	N,P,Q	3.17	25.17	28.33

3.3.1.	N	M	O,R	0.54	28.33	28.88
3.3.2.1.	O	N	S	7.67	28.88	36.54
3.3.2.2.1.	P	D,M	U	4.00	48.33	52.33
3.3.2.2.2.	Q	M	U	2.17	28.33	30.50
3.3.3.	R	N	S	16.67	28.88	45.54
3.3.4.	S	O,R	T	29.33	45.54	74.88
3.3.5.	T	S	U	5.00	74.88	79.88
3.3.6.	U	P,Q,T	… end	17.33	79.88	97.21

***Table 7.3** The first column corresponds to the number associated to each WBS activity, in the second column the letter associated to the activity, followed by the previous activities and then the consequent ones. In the last 3 columns we find the duration of each task, its start time and its ending time.*

7.2.1. GANTT

The GANTT diagram is used for showing when each task starts and ends.

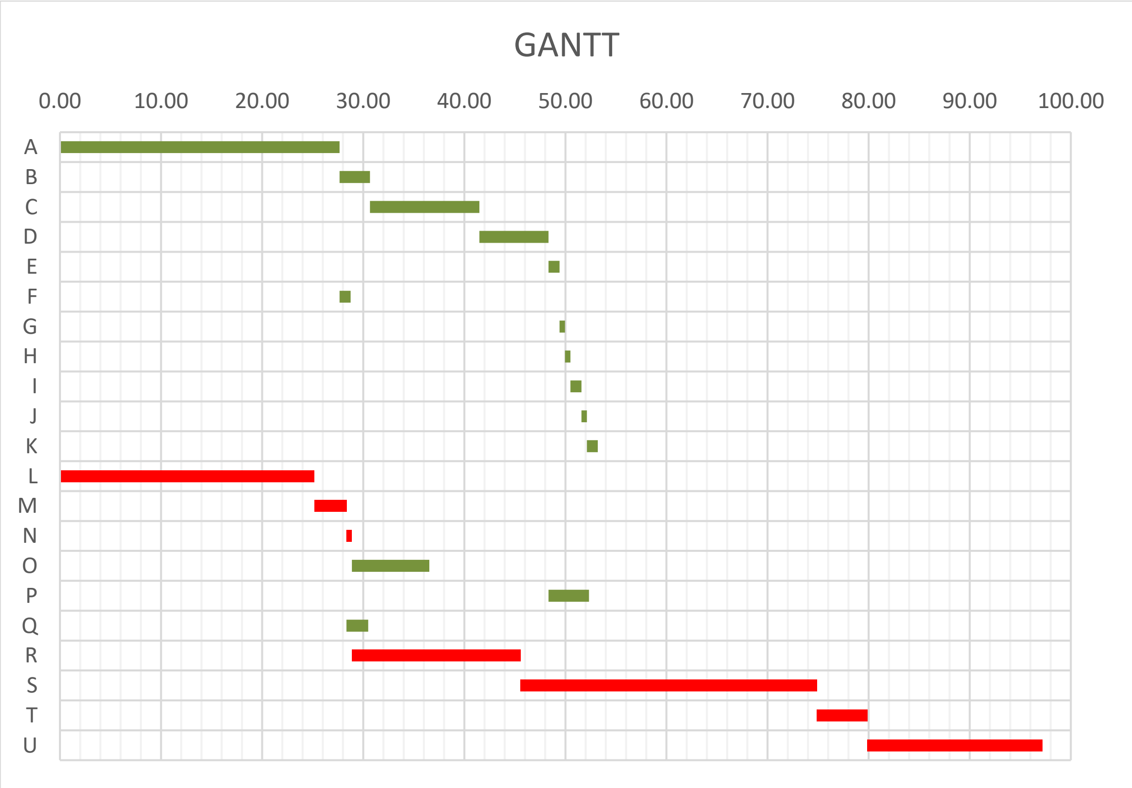

***Figure 7.2** The GANTT chart gives an idea of the start and end dates of each task and is important for monitoring specific tasks. The red bars indicate the activities from the critical path. Personal source*

7.2.2. PERT

Next, the PERT diagram associated with the sequence matrix is presented. Each node has 3 values which correspond to the number of that node, the Early and the Last times of that activity, respectively. The dashed line indicates fictitious activities, and each line has an associated activity of the matrix. The critical path is marked in red.

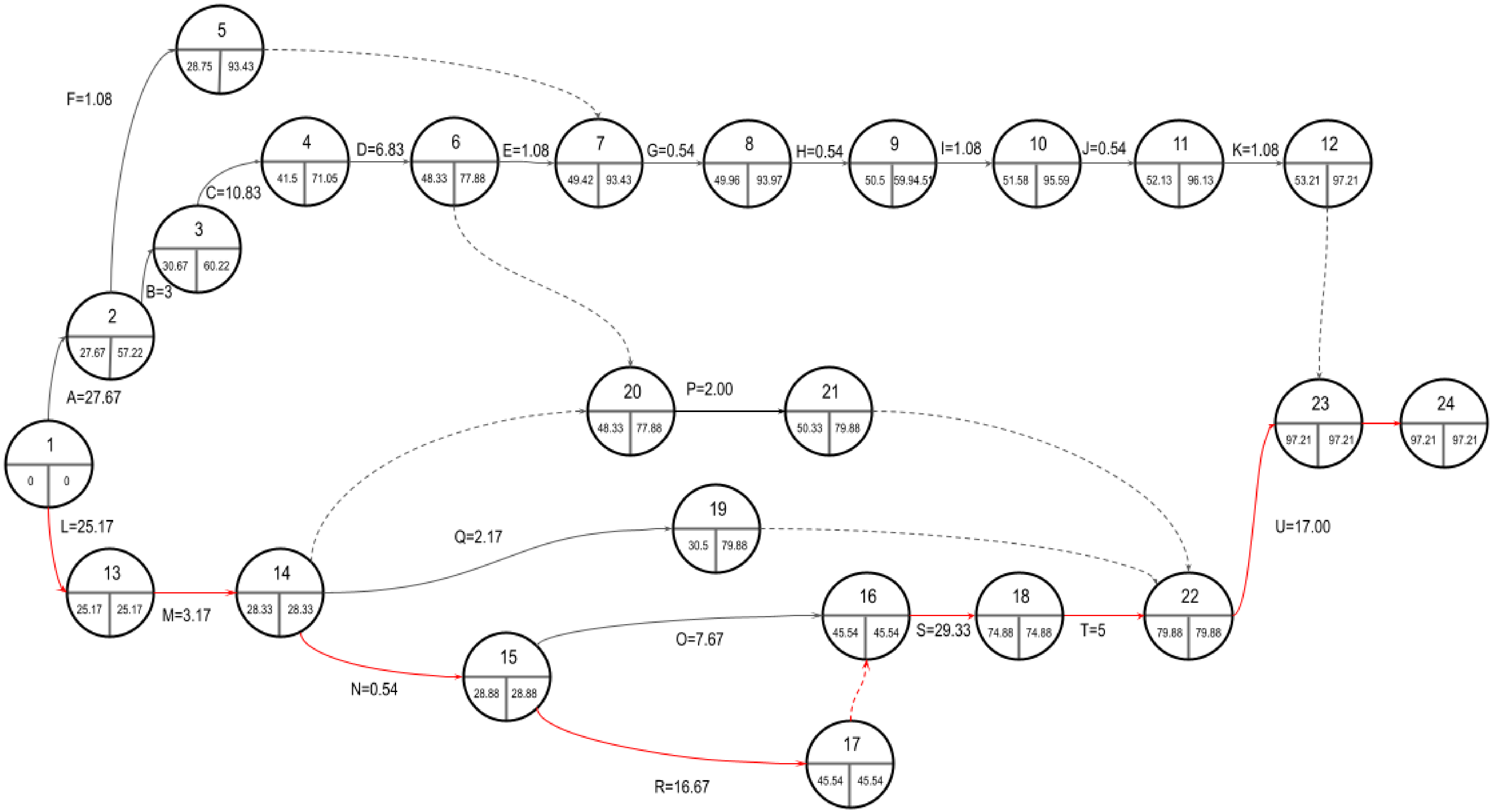

Figure 7.3 *PERT diagram of the project. Each node includes the number, the Early and Last times. The red lines indicate the critical path.. Personal source*

8. Technical viability

In this section the SWOT analysis will be performed. It will be focused on the elements that have affected the project during its development, and the elements that would affect the project if it was continued as a commercial device. First a diagram with the factors will be shown, and then, each section will explain how each element affected or could affect the project.

A SWOT analysis consists of two different factors: internal and external factors. The internal factors are the strengths and weaknesses, and the external factors the opportunities and threats.

Strengths	Weaknesses	Opportunities	Threats
An isolation phase could be added easily.	Quality of the signal may be low.	Increase of muscle atrophy and sarcopeny due to COVID restrictions.	COVID-19 pandemics
Friendly interface, both for the patient and the therapist.			
Reduces boring and repetitive tasks and promotes motivation.	It is a prototype, can not be commercialized or used in clinical trials.	Demographic aging.	USA cold wave.
Low cost device. Simplicity of the design.	Already existing technology.	Possibility of entering in developing economies	Multinationals.

Table 8.1 *SWOT matrix of the project.*

8.1. Strengths

As it will be explained in the economic viability section, this device is a low-cost system, what is an important factor during the COVID19 pandemics, which supposed a problem for several private institutions, as could be IBC. The reduced cost of all the elements needed to carry out the entire project facilitated its development.

In addition, it has a friendly and easy-to-use interface, both for the therapist and the patient. This facilitates the integration on rehabilitation processes of different injuries or problems. This is related to the next point: it promotes motivation. As it has been seen there is a high ratio of abandonment of the rehabilitation process, due to repetitive and boring routines, but with the designed biofeedback system, it could reduce this problem, specially between the younger population.

Furthermore, it has a very simple design, which is an important factor in case that in a future, we would like to continue the project as a commercial device.

Even though, it does not include an isolation phase, it could be easily implemented using isolation amplifiers and other simple components, which would be necessary in case that a clinical trial was performed.

8.2. Weaknesses

The quality of the signal is lower compared with professional EMG systems already existing in the market. The lower precision and sensibility of this system could be a problem in cases that the prototype was used in patients with severe muscular atrophy. In these cases, the electrical activation of the targeted muscle could not be bigger enough to discriminate the signal noise and the real muscle activity.

This device is only a prototype, which has not an isolation phase, therefore, it could not fulfil EN-60601, which establishes the "General requirements for basic safety and essential performance" of medical electrical equipment. Hence it could not be used for clinical trials or as a commercial device.

Furthermore, EMG and Biofeedback are not new techniques and is an already existing technology which makes hard to stand out from other products.

8.3. Opportunities

COVID-19 has affected daily life in many aspects. During the strict quarantine that most countries imposed to their citizens to reduce the breakthrough, many people reduced drastically its physical activity and augmented sedentarism. Recent studies indicates that the consequences of these two elements provoked several muscle losses, which lead in some cases to sarcopenia [36,37,38]. These same articles indicate that rehabilitation may be needed in most cases to recover the lost, and in this situation, the EMG biofeedback system could have a great opportunity in rehabilitation centres.

Besides, as United Nations indicated in a 2017 report, global population is ageing during last ears, and the forecast continues in this line [39]. Aging can lead to sarcopenia, which is treated with promoting physical exercise, but to have a good recovery, in several cases rehabilitation is proposed. Even though the elderly is not used to videogames, with friendly interfaces and easy games, it could be also used for this purpose [40, 41].

Resource-poor countries could also be a target to this device in case that it becomes commercial, since as it has been said, it is a low-cost system, which could be destined to countries which have the need of rehabilitation processes using EMG and Biofeedback but cannot afford a typical device which can cost up to thousands of euros.

8.4. Threats

COVID-19 affected negatively the development of this project too. First, as it has been said in the limitation section, it had to start later than it was planned due to quarantine. This delay In the beginning along with the limitation of physical therapies, produced that the implementation of an isolation phase could not be done, and as consequence clinical trial has not been performed, which may has been an important point.

In addition, the components of the EMG system were ordered in Mouser Electronics the first week of February, which has the warehouse in Texas, USA. This year's winter in USA has been extremely cold, what caused that during the month of February the shipment of packages was cancelled, and the order arrived with 3 weeks of delay. Even though, it did not really affected the development of the project, since during this time, all the research was performed.

There are several companies and multinationals already established in the market with faithful clients. This could affect a future business, since it could be difficult to get a space in the market for the device and it may be conditioned to differentiation through price.

9. Economic viability

In this section the theoretical cost of developing the project is assessed. The budget includes the cost of human resources, the cost of the hardware and software needed for developing the project.

First, considering the salary of the study is 20€/h, and the total amount of hours needed for developing the project, considering 290 hours dedicated, make a total of 5800€. The hours are divided as follows: 53h for the educational stage which includes the research done to correctly develop the project. 112, hours have been dedicated to the development of the project, and finally 125 hours to edition of the project documentation.

The hardware, which consists of the Olimexino-328 main board, the EMG-Shield, the cable electrodes and the electrode make a total of 79,15€. The DataLog from Biometrics would have cost x€ in total considering 30€/day of renting. The laptop used 900€, and the electricity cost, considering a power consumption of 500W/h, 290h and a price of 0.1315€/kWh, makes a total of 19.07€.

The software used have been Visual Studio 2019, which is free as student, Python and the needed libraries are also free and finally Arduino IDE which is free too.

The following table summarizes the costs:

ITEM	QUANTITY	UNITARY COST	TOTAL COST
HARDWARE			
Olimexino-328	1	24.99€	24.99€
Olimex EMG-Shield	1	20.24€	20.24€
EMG cable electrodes	1	11.28€	11.28€
EMG electrodes	60	3.05€ pack of 30	6.10€
DataLog	5 days	30€/day	150€
DataLog Batteries	2	1€	2€
SOFTWARE			
Visual Studio 2019	1	1€	Free
Python	1	1€	Free
Arduino IDE	1	1€	Free
OTHER			
Electricity	145kW	0.1315€/kWh	19.07€
Laptop	1	900€	900€
Student hours	290h	20€/h	5800€
TOTAL			6933.68€

***Table 9.1** Cost analysis of the project*

Comparing a EMG professional device with the one developed in this project, it clearly has a much more higher cost, which can cost more than 7.000€ [42].

If we compare the EMG devic with other projects on the same line, all of them move through similar values. For example, a low-cost EMG device, using Myoware, the same project would cost around 100€ [43]. Only the cost of the components is being compared.

10. Normative and legal aspects

This project has consisted of developing an EMG device and biofeedback system for rehabilitation process. Since these are medical devices, it is important that they follow strict laws that ensure patient's security at any moment.

The EMG system is a medical electrical equipment, it must follow EN-60601-2-40:2019, which regulates "Particular requirements for the basic safety and essential performance of electromyographs and evoked response equipment" [44]. The prototype does not follow this statement since it does not have an isolation phase, but it could be added externally.

The Olimex Shield has not passed this standard, as its datasheet indicates, due to the high cost that it would involve. Therefore, the device has only been tested on myself, using my laptop disconnected from the electrical power to minimize risks. This is the main reason why it is only a prototype and cannot be used for medical purposes right now, even though an isolation phase could be added which would allow the device to pass the standard.

In case that in a future the device was tried to be commercialized, different regulation should be considered. For example, it should pass the Medical Devices Regulation which outlines different quality and risk management obligations. It should be tested to pass IEC 61000 which ensures electromagnetic compatibility. It also should pass one of the main regulations, ISO 13485, which stablishes specific requirements for the fabrication, installation, and maintenance of healthcare products, such as a quality control system implementation.

11. Conclusions and future steps

As it has been mentioned in previous sections of this document, the main objective of this final degree project was to design a protype with affordable components that could be understood as an alternative to cutting-edge technologies. Because of that, the aim was never to achieve the precision and sensitivity of currently available EMG systems; but to offer a device that may solve a real need for many therapists which cannot afford the latest technologies. It is important to note that even though the project has been planned in an academic context and tried to be developed with the fewer possible resources, it could easily be a product development project brought off by a healthcare company. As such, we would like to remark the success of the enterprise as a whole, with almost all the objectives fulfilled satisfactorily. In this section we present a summary of each objective and future steps towards their completion.

An initial literature study to understand the importance of a good rehabilitation therapy and how biofeedback can be implemented to maximize its outcome yielded many more applications than the expected. However, despite having great potential as a rehabilitation tool for many injuries and diseases, further investigation and development of biofeedback systems are needed to create the most customizable routine as possible to maintain motivation during the rehabilitation period and ensure function recovery.

The main aim of this project was the development of a low-cost EMG-Biofeedback system. The final prototype of the EMG system shows promising results as an alternative to high-cost devices. To assess the possibilities of a real application for therapy purposes, the prototype's signal was compared to a professional EMG signal; and it was deemed of sufficient quality to exist as an economically viable alternative for those who do not need or cannot afford the latest technologies. Nevertheless, a post-processing step may be useful to improve even more the final signal by applying digital filters and a better baseline correction algorithm. Additionally, it would be strongly recommended to add an isolation phase to ensure the security of the patients during therapy.

A second objective was to provide a motivating and engaging rehabilitation process to facilitate function recovery in an optimal time span. The resulting biofeedback game performs adequately and may be understood as an initial step in the development of a whole biofeedback environment to promote motivation in the patients, especially for the youngest ones. It is a simple game which targets maximum strength and explosiveness, two main capabilities of muscles, which are important in daily life and may be recovered after any injury.

Finally, the last goal was intended as a clinical trial with different patients to assess the performance of the prototype in a clinical setting. Such a study was not feasible due to different problems that affected the timings of the project: To begin with, COVID-19 produced a delay on the start of the work; which was planned to start in March 2020. Since it was not possible, and due to the academic nature of the project, we thought that it could be carried out when the health situation was better, but this was never the case. Another obstacle was the legal aspects required to perform a clinical trial. The device must follow the legal statements mentioned in the Normative section, which at the moment it does not, and the isolation phase implementation and future evaluation by the official authorities must be performed previous to patient testing.

To conclude with, we would like to highlight that the project was completed in a small time frame which included setbacks and delays due to shipping and purchase of the components. This was not a big problem, but it highlights the wide range of possibilities to provide a system in a short window of time even with customized features, which highlights the feasibility and possible commercialization of the project, which would help biofeedback to become more accessible as a rehabilitation tool and thus provide different patients and therapists at different acquisitive level the option to follow a rehabilitation regime with highly motivating custom exercises.

12. References

[1] Moritani, T., Stegeman, D., & Merletti, R. (2005). Basic Physiology and Biophysics of EMG Signal Generation. In Electromyography (pp. 1–25). https://doi.org/10.1002/0471678384.ch1

[2] Ferreira, C., & Augusto, R. (2012). Application of Surface Electromyography in the Dynamics of Human Movement. Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges. https://doi.org/10.5772/52463

[3] Florimond, V. (2010). Basics of Surface Electromyography Applied to Physical Rehabilitation and. Thought Technology Ltd, 1(March), 1–50.

[4] Cram, J. R. (2005). Biofeedback Applications. Electromyography, (1), 435–451. https://doi.org/10.1002/0471678384.ch17

[5] EMG biofeedback, indication, contraindication, electrode placement sites. (n.d.). Retrieved Mar 26, 2021, from https://biofeedback-neurofeedback-therapy.com/emg-biofeedback/

[6] Adrián, F. (2007). Análisis de la adherencia a los tratamientos kinésicos. Revista Del Hospital J. M. Ramos Mejía, 1–31. Retrieved from http://www.hospitalramosmejia.info/r/200701/272.pdf

[7] Hermens, H. J., & Freriks, B. (n.d.). The State of the Art on Sensors and Sensor Placement Procedures for Surface ElectroMyoGraphy: A proposal for sensor placement procedures Deliverable of the SENIAM project editors. Retrieved from http://www.seniam.org/

[8] Home | BTS Bioengineering. (n.d.). Retrieved March 27, 2021, from https://www.btsbioengineering.com/

[9] Hun Jang, M., Jin Ahn, S., Woo Lee, J., Rhee, M.-H., Chae, D., Kim, J., & Jun Shin, M. (2018). Validity and Reliability of the Newly Developed Surface Electromyography Device for Measuring Muscle Activity during Voluntary Isometric Contraction. https://doi.org/10.1155/2018/4068493

[10] Wu, J., Li, X., Liu, W., & Jane Wang, Z. (2019, July 12). SEMG Signal Processing Methods: A Review. Journal of Physics: Conference Series, 1237(3). https://doi.org/10.1088/1742-6596/1237/3/032008

[11] Roland, T., Amsuess, S., Russold, M. F., & Baumgartner, W. (2019). Ultra-Low-Power Digital Filtering for Insulated EMG Sensing. Sensors (Basel, Switzerland), 19(4), 959. https://doi.org/10.3390/s19040959

[12] Jaramillo-Yánez, A., Benalcázar, M. E., & Mena-Maldonado, E. (2020). Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review. Sensors (Basel, Switzerland), Vol. 20. https://doi.org/10.3390/s20092467

[13] Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., & Aszmann, O. C. (2014). The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. IEEE Transactions on Neural Systems and Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society, 22(4), 797–809. https://doi.org/10.1109/TNSRE.2014.2305111

[14] Shi, W.-T., Lyu, Z.-J., Tang, S.-T., Chia, T.-L., & Yang, C.-Y. (2018). A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study. Biocybernetics and Biomedical Engineering, 38(1), 126–135. https://doi.org/https://doi.org/10.1016/j.bbe.2017.11.001
[15] Mohamed, E., Tantawi, K. H., Pemberton, A., Pickard, N., Dyer, M., Hickman, E., … Nasab, A. (2020). Real time gesture-controlled mobile robot using a myo armband. Proceedings of the International Conference on Industrial Engineering and Operations Management, 59, 2432–2437.
[16] Edwards, P. K., Ebert, J. R., Littlewood, C., Ackland, T., & Wang, A. (2017). A Systematic Review of Electromyography Studies in Normal Shoulders to Inform Postoperative Rehabilitation Following Rotator Cuff Repair. Journal of Orthopaedic & Sports Physical Therapy, 47(12), 931–944. https://doi.org/10.2519/jospt.2017.7271
[17] Comaduran Marquez, D., von Tscharner, V., Murari, K., & Nigg, B. M. (2018). Development of a multichannel current-EMG system for coherence modulation with visual biofeedback. PLOS ONE, 13(11), e0206871. https://doi.org/10.1371/journal.pone.0206871
[18] Cheung, E. Y. Y., Yu, K. K. K., Kwan, R. L. C., Ng, C. K. M., Chau, R. M. W., & Cheing, G. L. Y. (2019). Effect of EMG-biofeedback robotic-assisted body weight supported treadmill training on walking ability and cardiopulmonary function on people with subacute spinal cord injuries - a randomized controlled trial. BMC Neurology, 19(1), 140. https://doi.org/10.1186/s12883-019-1361-z
[19] Yoo, J. W., Lee, D. R., Cha, Y. J., & You, S. H. (2017). Augmented effects of EMG biofeedback interfaced with virtual reality on neuromuscular control and movement coordination during reaching in children with cerebral palsy. NeuroRehabilitation, 40, 175–185. https://doi.org/10.3233/NRE-161402
[20] Cheung, E. Y. Y., Yu, K. K. K., Kwan, R. L. C., Ng, C. K. M., Chau, R. M. W., & Cheing, G. L. Y. (2019). Effect of EMG-biofeedback robotic-assisted body weight supported treadmill training on walking ability and cardiopulmonary function on people with subacute spinal cord injuries - a randomized controlled trial. BMC Neurology, 19(1), 140. https://doi.org/10.1186/s12883-019-1361-z
[21] Narayanan, S. P., & Bharucha, A. E. (2019). A Practical Guide to Biofeedback Therapy for Pelvic Floor Disorders. Current Gastroenterology Reports, 21(5), 21. https://doi.org/10.1007/s11894-019-0688-3
[22] Kuo, Y.-L., Wang, P.-S., Ko, P.-Y., Huang, K.-Y., & Tsai, Y.-J. (2019). Immediate effects of real-time postural biofeedback on spinal posture, muscle activity, and perceived pain severity in adults with neck pain. Gait & Posture, 67, 187–193. https://doi.org/https://doi.org/10.1016/j.gaitpost.2018.10.021
[23] Travers, B. G., Mason, A. H., Mrotek, L. A., Ellertson, A., Dean 3rd, D. C., Engel, C., … McLaughlin, K. (2018). Biofeedback-Based, Videogame Balance Training in Autism. Journal of Autism and Developmental Disorders, 48(1), 163–175. https://doi.org/10.1007/s10803-017-3310-2
[24] Delsys – Wearable Sensors for Movement Sciences. (n.d.). Retrieved May 28, 2021, from https://delsys.com/
[25] Home | Noraxon USA. (n.d.). Retrieved May 28, 2021, from https://www.noraxon.com/

[26] Neurodiagnostic and Neuromonitoring Solutions: EEG EMG IONM PSG. (n.d.). Retrieved May 28, 2021, from https://www.cadwell.com/
[27] Wearable Technology | Myontec. (n.d.). Retrieved May 28, 2021, from https://www.myontec.com/
[28] Cometa Systems | Wireless EMG and IMU solutions. (n.d.). Retrieved May 28, 2021, from https://www.cometasystems.com/
[29] Biofeedback - Thought Technology Ltd. (n.d.). Retrieved May 28, 2021, from https://thoughttechnology.com/biofeedback/
[30] Zukor Interactive :: Games. (n.d.). Retrieved May 28, 2021, from http://zukorinteractive.com/games.php
[31] Galek, K., Bice, M., & Hobbs, A. (n.d.). Surface Electromyography Device Assists in Rehabilitation of Swallowing Disorders. Today's Geriatric Medicine. Retrieved from https://www.todaysgeriatricmedicine.com/news/ex_021717.shtml
[32] Evolution of Biofeedback in Physical Medicine and Rehabilitation | Musculoskeletal Key. (n.d.). Retrieved May 12, 2021, from https://musculoskeletalkey.com/evolution-of-biofeedback-in-physical-medicine-and-rehabilitation/
[33] Lyons, G. M., Sharma, P., Baker, M., O'Malley, S., & Shanahan, A. (2003). A computer game-based EMG biofeedback system for muscle rehabilitation. Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, 2, 1625–1628. https://doi.org/10.1109/iembs.2003.1279682
[34] SHIELD-EKG-EMG - Open Source Hardware Board. (n.d.). Retrieved February 8, 2021, from https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/open-source-hardware
[35] SHIELD-EKG-EMG bio-feedback shield USER'S MANUAL. (2014). Retrieved from https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/resources/SHIELD-EKG-EMG.pdf
[36] Kirwan, R., McCullough, D., Butler, T., Perez de Heredia, F., Davies, I. G., & Stewart, C. (2020). Sarcopenia during COVID-19 lockdown restrictions: long-term health effects of short-term muscle loss. GeroScience, 42(6), 1547–1578. https://doi.org/10.1007/s11357-020-00272-3
[37] Narici, M., Vito, G. De, Franchi, M., Paoli, A., Moro, T., Marcolin, G., … Maganaris, C. (2021). Impact of sedentarism due to the COVID-19 home confinement on neuromuscular, cardiovascular and metabolic health: Physiological and pathophysiological implications and recommendations for physical and nutritional countermeasures. European Journal of Sport Science, 21(4), 614–635. https://doi.org/10.1080/17461391.2020.1761076
[38] Moro, T., & Paoli, A. (2020). When COVID-19 affects muscle: effects of quarantine in older adults. European Journal of Translational Myology, 30(2), 219–222. https://doi.org/10.4081/ejtm.2020.9069
[39] World population ageing, 2017 :highlights. (2017). Retrieved from http://digitallibrary.un.org/record/3799351/files/WPA2017_Highlights.pdf
[40] Saggini, R., Carmignano, S. M., Cosenza, L., Palermo, T., & Bellomo, R. G. (2017). Rehabilitation in Sarcopenic Elderly. In Frailty and Sarcopenia - Onset, Development and Clinical Challenges. https://doi.org/10.5772/intechopen.69638

[41] Dionyssiotis, Y. (2019). Sarcopenia in the Elderly. European Endocrinology, 15(1), 13–14. https://doi.org/10.17925/EE.2019.15.1.13
[42] New and Used EMG Machines, IOM Machines, and other neurodiagnostic equipment. (n.d.). Retrieved June 3, 2021, from http://www.neurosupply.com/NeuroExchange.htm
[43] Fuentes Del Toro, S., Wei, Y., Olmeda, E., Ren, L., Guowu, W., & Díaz, V. (2019). Validation of a Low-Cost Electromyography (EMG) System via a Commercial and Accurate EMG Device: Pilot Study. Sensors (Basel, Switzerland), 19(23), 5214. https://doi.org/10.3390/s19235214
[44] UNE-EN 60601-2-40:2019 (Ratificada) Equipos electromédicos. Pa... (n.d.). Retrieved June 8, 2021, from https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0061354

Appendixes

Appendix 1- EMG Arduino Code

```
#define INTERVAL 8333 //Used to fixing a sampling frequency
const int analogInPin = A0; //variable that fixed the analog inpput
pin
uint32_t lastMicros = 0; //variable that saves the last time in
microseconds

void setup() {
  Serial.begin(115200); //starts the serial at 115200 baudios
}

void loop() {
  if (micros()-lastMicros>INTERVAL){ //checks if the sampling time has
passed
    lastMicros=micros(); //saves the last time
    float sensorValue = ((analogRead(analogInPin)-float(509)));
//reads the signal and subtracts 508 to remove the baseline
    Serial.println(float(sensorValue)); // //prints the read value
}
  }
```

Appendix 2- Save Arduino signal in a txt file

```
#Import needed libraries
import serial
import numpy as np

serialArduino=serial.Serial("COM8",115200,timeout=1) #Read signal from
Arduino
arduino_data = [] #create a list
while len(arduino_data)<=1215: #While loop for reading the signal
      try: #tries to:
            data =
float(serialArduino.readline().strip().decode())#read the data, divide
the lines and decode it
            arduino_data.append(data) # Append the date to the list
      except: #if it do not can
            data=0
            arduino_data.append(data)
      print (arduino_data)
np.savetxt('bicepsard100nonorm.csv',arduino_data,delimiter=',')   #save
the data in a txt file
```

Appendix 3- Arduino signal and DataLog signal comparison

```
#Import needed libraries
import numpy as np
import matplotlib.pyplot as plt
import csv
from scipy import stats

dadesString = []
# Open the file
with open("bicepsard100nonorm.csv", "r") as iFile:
    sinReader = csv.reader(iFile, delimiter = ",",
quotechar='"')#indicate the delimiter character used
    for fila in sinReader:
# Save the data in a list
        dadesString.append(fila)
```

```
# Transform from string to float
dadesFloat = [[float(valor) for valor in Row] for Row in dadesString]
emgard=[]
for element in dadesFloat:
    emgard.append(element[0])

#The same but for txt files
with open("jutnos2.txt", "r") as inputFile:
    datalog = []
    for dada in inputFile.readlines():
        datalog.append(dada)
    datalog2=[]
    for element in datalog:
        tmp = element.strip()
        datalog2.append(float(tmp))


time=np.linspace(0,11,1096) #create time vector used for the graph

#remove first 120 values from both signals
for i in range(120):
    emgard.pop(0)
    datalog2.pop(0)

#remove last values from Datalog signal to make it as long as arduino
signal
for i in range (len(datalog2)-len(emgard)):
    datalog2.pop(len(emgard))

#plotting datalog histogram before normalizing
plt.hist(datalog2)
plt.legend(['DataLog'])
plt.ylabel('Frequency')
plt.xlabel('Amplitude')
plt.title('DataLog EMG raw signal amplitude histogram')


#plotting Arduino histogram before normalizing
plt.hist(emgard,color='darkorange')
plt.legend(['Arduino'])
plt.ylabel('Frequency')
plt.xlabel('Amplitude')
plt.title('Arduino EMG raw signal amplitude histogram')


#notmalize amplitudes
emgard_norm=(emgard-np.mean(emgard))/np.std(emgard)
datalog_norm=(datalog2-np.mean(datalog2))/np.std(datalog2)

#plotting histogram after normalizing
plt.hist(datalog_norm,bins=30)
plt.hist(emgard_norm,bins=30)
plt.legend(('DataLog','Arduino'))
plt.ylabel('Frequency')
plt.xlabel('Amplitude')
plt.title('DataLog vs Arduino EMG normalized amplitude histogram')


#plotting signal vs time from datalog and adruino to compare the
signal.
plt.plot(time,datalog_norm)
```

```
plt.plot(time,emgard_norm)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.legend(('DataLog','Arduino'))
plt.title('DataLog vs Arduino EMG signal')


#T-test performance
print(stats.ttest_ind(emgard,datalog2))
```

Appendix 4- Flappy Bird EMG-Biofeedback

```
#import needed libraries
import pygame, sys, random

import serial
import time

import tkinter as tk
from tkinter.simpledialog import askstring
#Create dialog for asking the threshold
#root = tk.Tk()
###THIS PART WAS IN THE FIRST VERSION, BUT IN THE EXPERIMENTAL
VALIDATION IT WAS CHANGED
#root.withdraw()
#try: #used for creating the pop-up which asks for the threshold
 #   threshold = float(askstring("Threshold", "Instructions: contract
the muscle for making the bird fly. When it is game over, contract the
muscle to start again.\nEnter the desired muscle activation
threshold"))

#except:
 #   threshold = float(askstring("Threshold", "Enter a valid value, it
must be a number.\nEnter the desired muscle activation threshold"))
###END OF THE CHANGED PART

serialArduino=serial.Serial("COM8",115200,timeout=1)
def readsignal(): #function which reads the signal
    try: #tries to:
        signal = float(serialArduino.readline().strip().decode())
#read the signal, decode it and save it as a float
        serialArduino.reset_input_buffer() #clears the input buffer
        serialArduino.reset_output_buffer() #clears the output buffer
    except:#if it does not can
        signal = 0
    return signal #returns the signal

data=[] #array that saves data coming from Arduino
while len(data)<400:
    val=readsignal()
    data.append(val) #adds the data at the end
print(max(data)) #prints the maximum value
threshold=max(data)*0.6 #sets the threshold

def draw_floor(): #function which draws the floor
      screen.blit(floor_surface,(floor_x_pos,450))#places the floor at
the desired position
      screen.blit(floor_surface,(floor_x_pos + 288,450))#places a
second floor 576 pixels at the right of the first floor to make the
movement effect and make it infinite

def create_pipe():
```

```
	random_pipe_pos = random.choice(pipe_height)#picks a random height from the pipe list
	bottom_pipe = pipe_surface.get_rect(midtop = (350,random_pipe_pos)) #places the bottom pipe at the desired poosition
	top_pipe = pipe_surface.get_rect(midbottom = (350,random_pipe_pos - 225))#places the top pipe at the desired position. The 225 indicates the separation with the bottom pipe
	return bottom_pipe,top_pipe #returns the pipes

def move_pipes(pipes):#function for moving horitzontally the pipes
	for pipe in pipes:
		pipe.centerx -= 5 #move sthe pipe 5 pixels to the left
	visible_pipes = [pipe for pipe in pipes if pipe.right > -25]#makes visibles the desired pipes
	return visible_pipes #returns the visible pipes

def draw_pipes(pipes): #draws the pipes
	for pipe in pipes:
		if pipe.bottom >= 512:
			screen.blit(pipe_surface,pipe)
		else:
			flip_pipe = pygame.transform.flip(pipe_surface,False,True) #flips the top pipe to orient it correctly
			screen.blit(flip_pipe,pipe)

def check_collision(pipes):  #function that checks collisions between the bird and the pipe
	global can_score
	for pipe in pipes:
		if bird_rect.colliderect(pipe): #if there is a collision
			death_sound.play()
			can_score = True
			return False

	if bird_rect.top <= -50 or bird_rect.bottom >= 450:
		can_score = True
		return False

	return True

def rotate_bird(bird): #function that rotates the bird to simulate better the flight movement
	new_bird = pygame.transform.rotozoom(bird,-bird_movement * 3,1)
	return new_bird

def bird_animation(): #function that changes the bird image for simulating the wings movement
	new_bird = bird_frames[bird_index]
	new_bird_rect = new_bird.get_rect(center = (50,bird_rect.centery))
	return new_bird,new_bird_rect

def score_display(game_state): #function that controls the score
	if game_state == 'main_game':
		score_surface = game_font.render(str(int(score)),True,(255,255,255))
		score_rect = score_surface.get_rect(center = (144,50))
		screen.blit(score_surface,score_rect)
	if game_state == 'game_over':
```

```
            score_surface = game_font.render(f'Score: {int(score)}'
,True,(255,255,255))
            score_rect = score_surface.get_rect(center = (144,50))
            screen.blit(score_surface,score_rect)

            high_score_surface = game_font.render(f'High score:
{int(high_score)}',True,(255,255,255))
            high_score_rect = high_score_surface.get_rect(center =
(144,425))
            screen.blit(high_score_surface,high_score_rect)

def update_score(score, high_score): #function that updates the high
score
      if score > high_score:
            high_score = score
      return high_score

def pipe_score_check():#function that controls score summing
      global score, can_score

      if pipe_list:
            for pipe in pipe_list:
                  if 47 < pipe.centerx < 52 and can_score:
                        score += 1
                        score_sound.play()
                        can_score = False
                  if pipe.centerx < 0:
                        can_score = True

#Definition of variables
pygame.init() #inits the game
screen = pygame.display.set_mode((288,512)) #sets the screen size
clock = pygame.time.Clock()
game_font = pygame.font.Font('04B_19.ttf',20)

# Game Variables
gravity = 0.18
bird_movement = 0
game_active = True
score = 0
high_score = 0
can_score = True
bg_surface = pygame.image.load('assets/background-day.png').convert()

floor_surface = pygame.image.load('assets/base.png').convert()
floor_x_pos = 0

bird_downflap = (pygame.image.load('assets/redbird-
downflap.png').convert_alpha())
bird_midflap = (pygame.image.load('assets/redbird-
midflap.png').convert_alpha())
bird_upflap = (pygame.image.load('assets/redbird-
upflap.png').convert_alpha())
bird_frames = [bird_downflap,bird_midflap,bird_upflap]
bird_index = 0
bird_surface = bird_frames[bird_index]
bird_rect = bird_surface.get_rect(center = (100,256))

BIRDFLAP = pygame.USEREVENT + 1
pygame.time.set_timer(BIRDFLAP,200)
```

```
pipe_surface = pygame.image.load('assets/pipe-green.png')
pipe_list = []
SPAWNPIPE = pygame.USEREVENT
pygame.time.set_timer(SPAWNPIPE,2000)#timer which controls when spawns
a new pipe

pipe_height = [200,250,300,350,400] #list which controls height of the
pipes

game_over_surface =
(pygame.image.load('assets/message.png').convert_alpha())
game_over_rect = game_over_surface.get_rect(center = (144,256))

flap_sound = pygame.mixer.Sound('sound/sfx_wing.wav')
death_sound = pygame.mixer.Sound('sound/sfx_hit.wav')
score_sound = pygame.mixer.Sound('sound/sfx_point.wav')
score_sound_countdown = 100
SCOREEVENT = pygame.USEREVENT + 2
pygame.time.set_timer(SCOREEVENT,100)
while True: #the game loop is included below
      signal1=readsignal() #reads the signal
      for event in pygame.event.get(): #when an event occurs
            if event.type == pygame.QUIT: #if the event is close
                  pygame.quit() #stops the game
                  sys.exit()
            if abs(signal1)>threshold and game_active==True: #if the
signal is bigger than the threshold and the game is active
                  bird_movement = 0 #restarts the movement variable
                  bird_movement -= 7 #makes the bird fly
                  flap_sound.play()
            if abs(signal1)>threshold and game_active==False:#if the
signal is bigger than the threshold and the game is inactive
                  game_active = True #activates the game
                  pipe_list.clear() #clears the pipe list
                  bird_rect.center = (50,256) #resets variables
                  bird_movement = 0
                  score = 0

            if event.type == SPAWNPIPE: #if the event is spawning a
pipe
                  pipe_list.extend(create_pipe()) #adds a pipe to the
pipe list

            if event.type == BIRDFLAP: #if the event is flapping
                  if bird_index < 2 :#if the index is lower than 2
                        bird_index += 1 #sums 1
                  else:
                        bird_index = 0 #if not restarts the variable

                  bird_surface,bird_rect = bird_animation()

      screen.blit(bg_surface,(0,0))

      if game_active:  #if the game is active
            # Bird
            bird_movement += gravity
            rotated_bird = rotate_bird(bird_surface)
            bird_rect.centery += bird_movement
            screen.blit(rotated_bird,bird_rect)
```

```
        game_active = check_collision(pipe_list)

        # Pipes
        pipe_list = move_pipes(pipe_list)
        draw_pipes(pipe_list)

        # Score
        pipe_score_check()
        score_display('main_game')
    else:
        screen.blit(game_over_surface,game_over_rect)
        high_score = update_score(score,high_score)
        score_display('game_over')


    # Floor
    floor_x_pos -= 1 #moves the floor
    draw_floor()
    if floor_x_pos <= -288: #if the floor is at the left of the
screen
        floor_x_pos = 0 #restarts the position


    pygame.display.update()
    clock.tick(120) #this controls the maximum frames per second
```