OULUN YLIOPISTO
UNIVERSITY of OULU

Psychophysiological measurements in programming tasks

Guidelines for conducting EMG research

University of Oulu
Faculty of Information Technology and
Electrical Engineering
Master’s Thesis
Mikko Rantanen
30.10.2019

Abstract

Programming languages have been studied and developed throughout history of programming. There are lots of different programming languages that are being used in software development, but only core languages are taught in Universities. Programming languages usually have their own syntax, which may differ greatly from each other. Using different programming languages for same task may provoke different emotions in programmers, depending their knowledge on the language.

Research on programming and programming languages have generally focused on technical and exterior aspects. More recently, there has been some research on the programmers and their emotions during the programming tasks. This master's thesis focuses on latter and aims to provide new information of programmers experienced emotions during the programming tasks by using EMG-recordings. This master thesis' main study focus is in psychophysiology, which combines psychology to physiological research, by finding correlation between physiological activity and emotional phenomenon.

This study assessed university students experienced emotions when conducting programming tasks with C and Python programming languages. EMG measurement device was used on the test participants to record signal data from facial based muscles for smiling and frowning activity, which are linked to positive and negative emotions.

This study's results showed small differences with emotional experiences during the programming tasks, but the overall results were not statistically significant. Therefore, more research on this topic is needed for more consistent results. Additionally, this research has provided guidelines on how EMG studies are conducted on laboratory setting and suggestions for future studies.

Keywords
Emotions, programming, programming languages, psychophysiology, electromyogram.

Supervisor
PhD, University Lecturer, Mikko Rajanen

Foreword

I would like to thank PhD Mikko Rajanen and PhD Dorina Rajanen for giving me the opportunity to conduct this thesis' research and guiding through in process of creating this thesis. Additional thanks to Mikko and Dorina Rajanen for giving me opportunities to improve my research skills and giving me more confidence for working in field of research. Special thanks to my spouse, family and close friends who had patience during this process and who have motivated me throughout the writing and study process of this thesis.

Mikko Rantanen

Oulu, October 30, 2019.

Abbreviations

EEG	Electroencephalogram
EMG	Electromyogram
HCI	Human Computer Interaction
IDE	Integrated Development Environment

Contents

1. Introduction

Computer programming is activity where working computer programs are being produced. Essentially computer programming is regarded as communication between human and computer, where humans write programs that computers can understand through compiling them. Principally, programmed computer programs are producing intended actions that are written in source code of the program. (Shneiderman, 1986.)

Modern society relies heavily on computers and computer based programmable devices such as smartphones. Computers and computer related communication innovations such as the Internet. Computers have evolved highly after WWII, where they played crucial role on encoding and decoding encrypted messages. Computers have had similar purpose throughout history, to process complex mathematical calculations. These calculations have been programmed to computers through different methods, by using algorithms and native language of used computer. Programming has evolved along the computers. Programming has evolved from being abstract machine language towards human readable languages and moved focus of programming towards problem solving tasks. (O'Regan, 2012.)

Assessment of programming tasks and languages has been traditionally conducted through efficiency of programs and usage of programming languages. Continuously increasing count of programming languages makes assessing of programming languages differences and similarities difficult and therefore they are not studied in extensively. (Sammet, 1991; Shneiderman, 1986.) Additionally, emotional factors of programming tasks have been studied quite widely as part of human-computer interaction and education related studies throughout recent history. (Fritz, Begel, Müller, Yigit-Elliott & Züger, 2014; Shneiderman, 1986.) More recently, psychophysiological research has been brought as a part of research methodology in human-computer interaction (Müller & Fritz, 2015).

Studying human factors in software development has previously helped to generate ideas that have been benefited the software industry. One major example of this is the Agile manifesto, which resulted from studies of software development processes’ and tools. These ideas have helped to improve performance and productivity in field of software engineering. Previous studies of human factors in software development has indicated that more studies of software developers’ emotions and cognitive work process’ during the development process are needed in order to provide improvement for working methodology and conditions. (Graziotin, Wang & Abrahamsson, 2015a.)

This study implements multidisciplinary research by utilizing computer science, psychology and psychophysiology fields of study. These fields of study provide main concepts to this research. This study follows positivist research philosophy. This study is conducted by implementing an empirical research, more precise as an experimental research, where experiment is being conducted and randomization of study is being conducted. (Coolican, 2009.)

1.1 Motivation of this study

This thesis conducts research primarily on how different programming languages effect on programmer’s emotions, in order to compare if different programming languages features and additionally to test EMG measurement techniques' suitability to this kind of study. Secondary scope of this study is to provide guidelines for EMG studies in this context and research setting. Main focus of this study is on students of Information processing science and Computer Science, although this study was not exclusive with these study majors. Students with preliminary knowledge of programming or their study major included programming was accepted to take part to this study.

Proposition for this thesis came from PhD Mikko Rajanen of University of Oulu's INTERACT research group. Initial purpose for this master's thesis was to provide new perspective and information from different psychophysiological tool, than what was earlier used in another master thesis. Previously conducted study used EEG measurement tool in similar setting as this study. The selected tool for this study was PsychLab's EMG monitoring and measurement device and software.

This study focuses on the C and Python programming languages in more detail. Selection of these programming languages derives from University of Oulu's programming courses in Information Processing Science and Computer Science and Electrical Engineering, where these programming languages are being taught on introduction courses to programming. Selection of these languages was also used in Rajendra Desai's (2017) master's thesis.

The motivation and preliminary setting of this research was to compare results of this study with previous research of Rajendra Desai's (2017) master's thesis study with use of different psychophysiological measurement device. The previous study was used as an inspiration for research setting, research questions and references. This study differs in methodology and more precise in conduction of the experiments. Main difference between these studies are in the used psychophysiological measurement device. In Rajendra Desai's study, the used device was EEG-headband, while in this study, the used device is EMG-device. Main goal of these studies is similar as psychophysiological studies in programming languages, but the context of these studies has more difference.

This study's used programming tasks for experiments had influenced, but redesigned and rewritten for this experiments' purposes, by programming tasks that were used in Rajendra Desai's (2017) study. On this study, the previously used first task was divided into two tasks and implemented precious study's second task as third on this study. The third task of this study was enhanced further. Programming was conducted likewise on online-IDE, but in different website.

In this study, the EMG measurement tool is used for psychophysiological measurement, which is used for measurement and analyzing facial muscular activity. The used EMG measurement tool is provided by PsychLab's with utilization of their measurement software. Nature of EEG and EMG differs significantly, since they are used to gather different kind of psychophysiological data, but they may use to gain insights of same issues. EEG is used for analyzing brain waves, while EMG is used for analyzing muscular activity. (Andreassi, 2007.) Additional motivation for this research was the need to provide guidelines for conducting EMG research in University of Oulu's UX-Lab. Therefore, this viewpoint and medium of psychophysiological measurement device was chosen for this research.

1.2 Research hypotheses

Following research hypotheses are examined further in this study on subsequent sections. Following hypotheses serves as basis for this research and are based on theoretical background that is presented in this study. This thesis aims to answer following hypotheses:

> H_0: Test participants do not experience higher facial muscular activity when programming with either C or Python.
>
> H_1: Higher corrugator supercilii activity is recorded when programming with compared language.
>
> H_2: Higher zygomaticus major activity is recorded when programming with compared language.

This thesis follows subsequent structure: on Chapter two, the theoretical background of this study is being presented. Third Chapter presents the previous research on this thesis' subject. On Chapter four, the research methods and analyzation methods of this study are being presented and discussed. Chapter five presents this study's research implementation. The results of this study are presented on the sixth Chapter of this study. On Chapter seven, the results are being discussed and contrasted with previous research. Finally, the Chapter 8 consist of conclusion of this study and conducted experiments, future studies are being discussed also on the eight Chapter.

2. Theoretical background

This Chapter provides theoretical background for this study's selected topics through scientific articles and previous studies. Following sections are describing programming, emotions and psychophysiological research on theoretical level. Selected topics are described further with related subjects that are derived from the main topics.

2.1 Programming

Traditionally programming as a task requires additional work phases that are producing supplementary value for the actual task. This task process usually begins with problem statement, where programmers are looking for answer for requested program through requirements analysis and early designs of the program. Actual programming occurs after the early stages, when initial lines are drawn for suggested program. Programming tasks contains also revision of the produced source code, that may contain bugs and other errors that requires programmers' attention in order to get the final version of program to be working as expected. (Shneiderman, 1986.)

Software development is mainly conducted by humans; therefore, it is prone to human errors. For mitigating these errors, the software development process includes debugging activity, which aims to correct possible errors that software developer has been made. (Khan, Brinkman & Hieros, 2010.) Software engineers needs lots of supportive information to success in their work. Earlier studies have suggested that software engineers most relevant criteria for information comes from effortless access and its usefulness. Information needs among the software engineers are strongly based on work-related or task-related context, since retrieved information generally helps them to proceed in their work. Software engineers usually search information through document-based resources as Internet, organization's intranet, technical documentation or through other resources such as colleagues. Software engineer's duties impact on the level of their information needs. Junior level engineers usually need more practical information while senior level engineers may require more recent theoretical information (Freund, 2015.)

Human context in programming performance have been evaluated on how well the written program performs and how well it can be maintained by others, but additionally through cognitive processing of programming tasks in some studies. Psychological assessment of programmers and use of programming languages can bring improvement for programming learning methods and used programming tools. (Shneiderman, 1986.)

2.1.1 Programming languages

Programming languages are formal languages that are constructed through series of strings and symbols, forming a script that performs set action. Programming languages are widely used for creating working programs in computer. (Scott, 2005.) Programs are needed to be compiled by computers in order to make the work as expected. Successful compiling action enables the intended actions to be performed with the computer. The compiler verifies the written programming language's syntax and semantics in order to make it compliable for the computer. (O'Regan, 2012.) Programming language's syntax functions as programming languages' vocabulary and required form of the language, as a semantics of language. Correctly followed syntax provides programming language meaning that will produce indented programmed action through interpretation with computer. As an example, mathematical signs, like a plus sign, can be used for mathematical calculation within syntax to produce addition calculation through the program. In addition to syntax, programming utilizes algorithms in order to run programs logically while performing given tasks. Algorithms are used for creating programmatical loops and other programmatically following sequences, providing more operations and functionality for the program. (Scott, 2005.)

There is huge amount of used programming languages that are being used in software development. To name few programming languages, C, Java, Python are being used in software development context, since they are easy to use in contrast of older languages like ALGOL, COBOL and FORTRAN, which are among the first programming languages that has been developed and used. Many of different programming languages' syntax differ from each other, but there are lots of similarities or identical syntaxes among programming languages. This is because the new programming languages have been influenced by existent programming languages and formed sort of a programming language family trees. Programming language's characteristics can influence usage of the language when deciding where specific language is prosperous to use, for example when programming on device or machine level activity or when programming graphical user interfaces. (O'Regan, 2012; Sammet, 1991.)

C Programming language is a structured programming language, which means that the language must follow logical structure in order to work correctly. Structured languages, like C, uses loops and conditional expressions as part of its syntax. C programs are built in code blocks that are breaking down the code in smaller and more readable code segments. C programming language has advantages in operability in different computer platforms and it has cost-efficient memory handling capabilities. (Dixit, 2010.)

Python is likewise a structured language as C, but it is mainly used for scripting because of its characteristics. Python is newer language than C, and it has been influenced C language among other programming languages. (McGrath, 2014.) Python offers as much programming capabilities as C, in addition, Python contains more relaxed syntax and memory allocation solutions. These features are also making Python ideal for beginners programming language, since they support in reducing programming errors occurrences. Usage of Python as first programming language for students has been studied in Finland and in England. These studies have indicated that Python’s characteristics with syntax are factors that make it easier to learn as a first programming language. (Nikula, Sajaniemi, Tedre & Wray, 2007.)

Learning of different programming languages can be frustrating if compiler's given feedback on erroneous code is difficult to read. C programming language has been recommended as a good example of first language to learn, because it provides understandable error messages when compiling of the code fails. C programming language provides capabilities to perform complex tasks and therefore enables learner to develop complex problem-solving skills. (Kordaki, 2010.)

2.1.2 Programming skills

Learning efforts of students programming skills have been researched on several studies. Traditional focus point in programming learning assessment has been on outcomes of the programming tasks, such as efficiency and correct output of the tasks. For a contrast to traditional code generation skills of the students, the researchers have suggested to bring out programmatic way of thinking and problem-solving skills for grading methods of students in programming courses. (Blikstein, Worsley, Piech, Sahami, Cooper & Koller, 2014.)

Generally, it has been noted that beginning in learning of programming languages can be challenging for students in programming courses. Prior studies have tried to observe if the students' problems in programming learning difficulties are emerging from level of knowledge of programming languages and paradigms or missing cognitive skills that would help them to master the programming tasks. It was found that some students lacked the knowledge in fully understanding certain programming commands function rather than missing problem-solving skills. (Perkins & Martin, 1986.)

Efficient programming skills requires understanding of written source code and how to produce meaningful source code. Mastering these skills may take time from students and they might not pay enough attention for both of these skills. Source code reading skill is regarded to be crucial for programming experts. (Nikula et al., 2007.) Earlier studies have indicated that students' problems in programming are experienced within planning of the task and limitations in knowledge of the programming language. For programming experts and novices, same problems have been recorded and additionally debugging problems of the source code. (Blikstein et al., 2014.)

Teaching of simple languages with simple syntax, like Python, as a first programming language has been criticized by some researchers. Critiques have claimed that students who learn simpler syntaxes may not be able to solve harder programming problems. As a response to critics, the supporters of simple programming languages and syntaxes have argued that by learning these languages, students learn algorithmic thinking in programming that is applicable in another languages and basis of programming syntax in general. It is argued that learning one simpler language will prepare student for learning more complex languages in future and improving programming skills. (Mannila, Peltomaki & Salakoski, 2006.)

2.1.3 Reading and Programming

Reading is an activity to understand written text. Reading activity consist of construction and analysis techniques. Construction is used for making sense of written text, where analysis is used for assessing the information in the text. Context of the text can vary from printed to digital material. There have been studies that indicates that digital reading is intensive for humans' cognitive skills than reading from printed media. Additionally, readers preference over digital versus print medium can be explained partly because familiarity with the medium and partly that for certain personality types, like Drive searching or Fun Seekers, that were found from Rajanen, Salminen & Rajava's (2016) research, the digital medium does not appear sufficiently motivational. (Rajanen, Salminen & Ravaja 2015; Rajanen, Salminen & Ravaja 2016.) Software reading is reading technique in software engineering context where the reader gathers information for succeeding with software engineering task. Software reading utilizes analysis and construction reading techniques, which are applied usually to digital context of reading. In software engineering context, the purpose of reading is to find errors in the code or to understand the logic behind the code, in addition to understanding the context and succeeding with the task. (Basili, Caldiera, Lanubile & Shull, 1996.)

Software reading consist different techniques that are used for assessing the reading source. Software reading techniques consists of Unstructured, Semi-structured and Structured reading. Unstructured reading technique is most common used technique, where the reader tries to understand parts of code by reading without particular method applied. Semi-structured reading uses checklists as guidelines for the reading, therefore the readers know what kind of issues should be looked from the reading subject. Structured reading technique is conducted via several reading techniques and different perspectives that are used to assess the software. These technique combination and perspectives helps the reader to understand the read software thoroughly and see possible inconsistencies in it. (Zhu, 2016.)

Software reading differ from traditional reading, since the source code can be read from different directions, depending on individual reading preferences or if there is specific situation that requires reading the source code from different direction. Software reading is also affected from several other factors. These factors include readers ability to read the source code, representational factors, such as used naming conventions in the code and environmental factors, such as poor lighting. Readers ability to read and understand the source code has most significant effect on reading, since the readers experience with the particular programming language and programming on overall level determines how well the reader can manage the source code reading task. (Zhu, 2016.)

2.2 Emotions

Emotions have been deeply studied phenomenon in psychology. Emotions have been analyzed and mapped throughout emotions research history and researchers have suggested that there are over 100 emotions. Emotions have been categorized to be inheriting from higher level categories to wider range of emotions and emotional states. Emotional categories can be utilized when assessing different activities. (Shaver, Schwartz, Kirson & O’Connor, 1987.)

2.2.1 Emotions as a phenomenon

Emotions are psychological related activity or phenomena that are based on biological and sociological factors, which are reflecting individuals’ feelings regarding something or someone. For example, emotions can be sensed as state of happiness or fear from individuals confronted experienced. Emotions are associated with learned models and social constructions of how individuals have learnt to feel specific emotions when same specific experience is confronted and repeated, as an example fear of certain animals. Experiencing certain events and dealing with the with same emotional responses as earlier is empowering experience feeling in same situations. (Ekman, 1999.)

Evolutionary psychologists have analyzed emotions as adaptions of functions that are based on human traits. Functions or adaptions are enabling humans to adjust their environment and changes in them. Another emotional related function is to organize activity of muscles and organs through cardiovascular and respiration systems. (Oatley, Keltner, & Jenkins, 2006.)

Emotions are regarded as necessity for human’s survival and capability to cope with social environment. There has been research on human emotions that person may experience different emotions without impressing it and vice versa. These emotions have been able to be recognized through typical patterns in human nervous systems, which are reflecting certain emotions, for example emotions like anger and fear. Researchers have utilized EEG to find these emotional patterns when persons are stimulated. (Ekman, 1999.)

Lewis' (1999) model of emotional development suggests that basic emotions are acquired early stages of childhood in infant stage, in early as three months old. These emotions and feelings are empowered or discouraged through early childhood development as part of development of self-consciousness. It is suggested that emotions states can arise unconsciously without thinking of the stage that one is on at certain moment, but also through active thinking as being aware of experienced emotions. (Lewis, 1999.)

2.2.2 Emotional states

Modern psychology uses concept of mental representation to study emotions. Mental representations are also used in human-computer interaction and user experience -studies. Mental representation contains two levels of mental content. Mental representation is at first level cognitive, focusing on sensory and memory information, which are related to person's ability to observe space, colors, objects, movement, laws and other impalpable elements. On second level, mental representation focuses on emotional information, which contains valence, arousal, mood and emotional themes. Valence contains emotions that are related to positive and negative feelings. Valence is regarded as a basic content measurement of human emotion. Arousal contains feelings that can activate or deactivate person's interest towards something. Valence and arousal have been recorded to be most influential emotional states for humans. Mood and emotional themes are associated with such states as fear, rage, envious feelings and happiness. (Gomez, Zimmermann, Guttormsen, Schär. & Danuser, 2009; Saariluoma & Jokinen, 2014.)

The Dimensional framework is used to categorize human emotional states and feelings into dimensional level in order to visualize emotional states related to arousal and valence. This framework uses PAD model, where emotional states are divided into three categorical levels: Pleasure-displeasure, Arousal-nonarousal and Dominance-submissiveness. PAD models are typically used for assessing bipolar dimensions, which means that the model contains opposite emotional states. (Graziotin, Wang, & Abrahamsson, 2015b.)

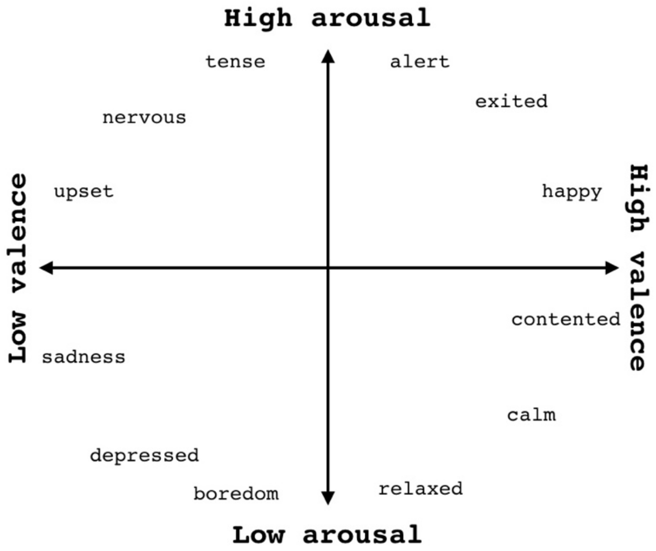


Figure 1. PAD-model.

PAD model can be used as a tool to assess emotional state associated with tasks. This example PAD model contains arousal and valence as factors for contrasting emotional states. Pleasure and displeasure are associated with attraction of the task, which also is regarded as motivational level. Arousal is associated with awareness and Dominance is associated of being in control of the situation. (Graziotin, Wang, & Abrahamsson, 2015b.)

Human cognitive actions can have affect to emotions and moods, one of these actions is attention, which controls humans' interest to its surroundings. Attention is regarded as automatic and controlled processing, which means that humans can focus their attention unintentionally or intentionally. Negative emotions are known to weaken focus controlled processing in demanding tasks, such as problem solving. It is suggested that switching between different kinds of tasks may have positive effect to attention regaining and performing more efficiently with given tasks, if current task seems to be too challenging. (Matthews & Wells, 1999.)

Analyzation of human face as a source or measurement for human emotions has been questioned, mainly if it can provide accurate enough information regarding the emotions. Theses critiques has been based on old researches, who have been inclusive and countercriticized by other researchers. There has been conducted several studies afterwards, which has been observing human facial behavior that have gained results that indicate similar facial expressions and behavior associated with similar circumstances or events among the test participants. Cultural similarities with facial behavior have been measured when showing neutral and stressful video material by utilizing FAST (Facial Affect Scoring Technique) -technique, which is used to observe three areas of human face, brow and forehead area, eyes and lower face. This technique provides measurable data for observing six different emotions of being happy, sad, surprised, anger, fear or disgusted. Studies of test participants from two different cultures, showed that similar emotions and facial behavior could be recorded when showed test material. Additional research has been conducted with similar means and has found that these emotions are universal. These tests indicate that observing human face and related behavior can provide accurate data regarding human emotions. (Ekman, Friesen & Ellsworth, 2013; Ekman & Oster, 2013.)

Emotions linkage to distinct appraisals have been studied and found that there are certain actions that can produce different levels of emotions with the subject and subjective behavior. There have been study results of Olympic athletes that have won or lost medals in Olympic games, these results have shown that medal winners levels of smiling have had variance according to medal's characteristic they have won. From, individual aspect, the emotional variation has been recorded altering if individuals have experienced life changes. (Matsumoto, Keltner, Shiota, O'Sullivan, & Frank, 2008.)

2.3 Emotions in programming

Research of programmers' personalities have emerged new trends in human-computer interaction research. Psychological profiles have been studied with psychological test that measures different personality dimensions. These tests have included effects of feelings and ways of thinking over personality type, but further emotion related study was not conducted within this research. (Shneiderman, 1980.) Modern psychological related studies in programming utilizes psychology's theories and new measurement tools of psychophysiological devices to gain more insight of these issues (Fritz et al., 2014; Müller & Fritz, 2015).

Human and psychological factor in software engineering research have been lacking in number of studies. Acknowledging fact that human and psychological factors have enabled higher performance in software industry, mainly through Agile manifestation. Software development's productivity on individual level has not been studied significantly, but researchers are suggesting that paying more attention to emotions and moods of the software developers could improve the working methodology. (Graziotin, Wang, & Abrahamsson, 2015a.)

Emotions that are experienced among programmers during their programming tasks varies from happiness to frustration. Additionally, anger and passion are also commonly being experienced when programming. These emotions reflect certain stages in programming tasks. Happiness is mainly associated with positive progress in programming tasks, where frustration reflects mainly situations where programmer gets stuck in his task. (Müller & Fritz, 2015.)

Several researchers have been studied how workplace conditions effect on emotions and moods and if they have effect on workers productivity. These researches have been conducted in software development context. Results of these studies indicated that working conditions and practices have huge impact on the productivity. (Shaw, 2004.) Only few of these have studied how actual programming effects on emotional states of the programmers and how they effect on productivity (Graziotin, Wang & Abrahamsson, 2015a).

Development in psychological studies enables researchers to observe different emotions through biometric sensors, which can record test participants skin temperature or facial expressions. Psychological research gives tools to identify flow and stuck stages among tasks which can be applied to research in different activities. These methods have been used to research programmers' emotions during programming tasks. Eye-tracking and fMRI tools have also been utilized to study how programmers understand the source code and how they use certain programming tools. (Müller & Fritz, 2015.)

2.4 Psychophysiology

The psychophysiology is empirical research methodology of human cognition related to physiological aspects. Psychophysiology is based on medical science's theoretical and empirical characteristics that provides concrete data for psychological research on human behavior. Psychophysiology is closely related to behavioral studies, that are studying human and animal behavior. Psychophysiology's main goal is to study human's cognitive skills that are linked to human physiology, mainly with brain activity's relation to movement skills and brain injuries that have effect on physiological actions. (Andreassi, 2007.)

Psychophysiology's main research interests are in emotional factors, relation of stress to physiology, cognitive task performance. Additional to physiological related behavior, psychophysiology studies personality and intelligence. Psychophysiology is applied in six different fields: Social psychophysiology, developmental psychophysiology, cognitive psychophysiology, clinical psychophysiology, applied psychophysiology and individual differences. From these fields, the applied psychophysiology is most relevant for this study, since it is used for gathering individual biofeedback in occupational, recreational and clinical environment. (Sowden & Barrett, 2008.)

Psychophysiology has also been used on research of reactions and emotions regarding digital content and media. Physiological research tends to look for reactions that are unconscious or spontaneous for the test participant itself and therefore provide more authentic content than for example with self-reporting of the participants. (Rajanen, Salminen & Ravaja 2016.) Psychophysiology methods, such as facial EMG measures, can be used to find discriminated emotional responses more effectively than self-reports (Marghescu, Salminen & Ravaja, 2011). Psychophysiology studies with digital content and media have utilized theories regarding memory and attention, which are related to how test participants or viewers of media content perceive the material. Additionally, when reading digital material, the reading device's physical features, like screen size and how far the screen is placed from the reader, has impact on reader's perception of the read material and abilities to evaluate the read material. (Rajanen, Salminen & Ravaja 2016; Ravaja, 2004.)

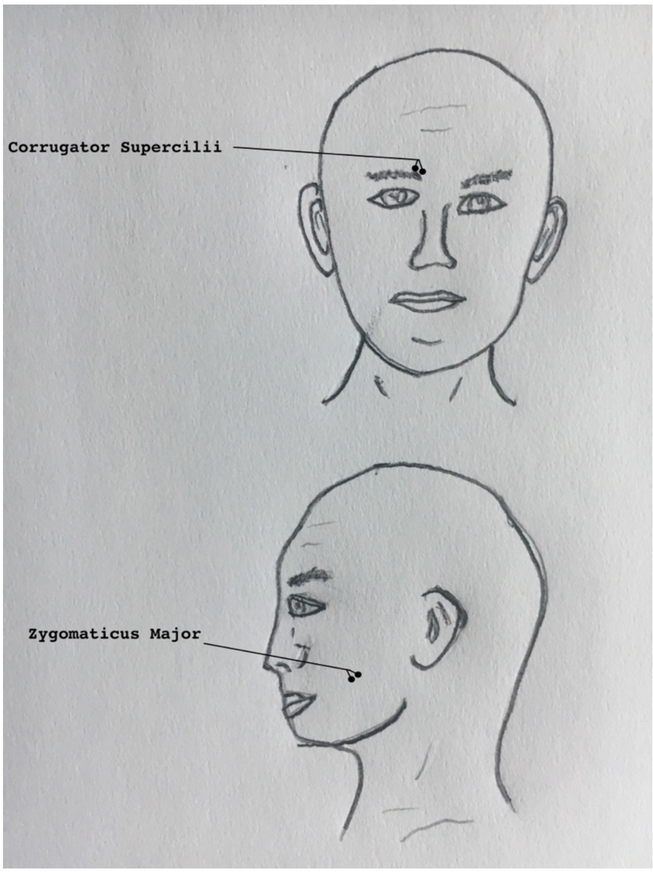
2.4.1 Electromyograms

Behavioral scientists and modern psychologists have measured humans muscle activity through electromyograms (EMG) recordings, that have indicated corresponding results in human behavior and muscle activity. The EMG is recorded from the test subject's skin surface, where electrodes are placed to monitor specific muscles that are point of interest. The muscles provide small electric amplitude when in rest or in active state, that can be monitored through the electrodes. There are different kinds of EMG electrodes, mainly electrodes that contain small needle, which is injected to observed muscle for being able to monitor its activity and surface electrodes that are placed over test participants skin with sticker. EMG recordings are traditionally conducted for observing rehabilitation of the muscles or for clinical diagnosis. (Andreassi 2007; Fridlund & Cacioppo, 1986.)

Another modern usage of EMG is related to observing different psychological behavior related to motorial behavior, such as speed of reaction, cognition and emotional expressions. History of facial expression studies derives from Charles Darwin's studies from 1872 to present days. Facial expression studies made significant results in 1970s when EMG devices were taken within to the studies. Human behavior and emotions can be measured with electrodes that are placed over test subjects face in specific places. These electrodes measure facial muscles activities and can detect for example if test subject smiles. The muscle activity produces tension that can be observed and analyzed through recording with electrodes. (Andreassi, 2007.)

Recording facial activity with EMG also requires researchers to understand psychological factors and emotional context of related facial expressions that are being measured. Additionally, EMG can be used for measure test subject's confusion and interest levels. Researchers have found relation between cheek-based zygomaticus major muscle's activity with positive emotions and eye-brow related corrugator supercilii muscle's activity with negative emotions. Studies have shown that muscles on left side of human face can be more active when recording facial activity on positive and negative related emotions. This means that muscles on left side of face can provide more spontaneous reactions. (Andreassi, 2007; Partala, Surakka, & Vanhala, 2006.)

Prior studies in facial EMG activity have shown that muscular and EMG activity is higher in corrugator supercilium muscle area than in zygomaticus major muscle area when experiencing negative emotions. Studies have shown zygomaticus major having increased EMG activity when individuals are experiencing positive emotions. Cacioppo, Petty, Losch and Kim (1986) have covered in their studies on surface EMG activity of facial activity, have shown that humans experience over emotional stimulus is varying on individually. Their study's results showed that humans are experiencing valence can be measured significantly with facial EMG, particularly from muscles of supercilium and zygomaticus major, which were observed more closely in this study. (Cacioppo, Petty, Losch, & Kim, 1986.)



Picture 1. Placement of electrodes corrugator supercilii and zygomaticus major -muscles.

Fridlund and Caioppo (1986) have studied EMG measurement and composed widely recognized guide that illustrates electrode placement for major facial muscles. On Picture 1, the placement of electrodes on corrugator supercilii and zygomaticus major muscles are shown. EMG electrode placement is usually conducted with bipolar setting with two active electrodes, usually on left side of the face. In placement of the electrodes, the muscle's mass and location should be identified for correct placement. Motorial endplate of the muscle should be identified and not to place electrodes nearby it to reduce cross interference and to get accurate results. (Fridlund & Cacioppo, 1986.)

The EMG devices electrodes are placed on test subject's facial areas that are linked to different emotions when activated correctly. Same muscles can be used with at least two to three different emotions. With EMG, these muscle actions can be recorded whether they are minimal or unnoticeable. It is suggested that researchers should know on beforehand what emotions are being recorded or observed. (Ekman & Oster, 2013.) The EMG studies have usually been conducted by applying visual stimulation of test subjects. Main studies with EMG have been consisted of showing different kinds of pictures that will stimulate test participants and see what kind of reactions can be recorded with EMG from test participants. (Andreassi, 2007.)

In this study, the corrugator supercilii and zygomaticus major are muscles of interest, since they are linked to positive and negative emotions, such as happiness and frustration. Corrugator supercillii controls eyebrow by raising and lowering it. Zygomaticus Major controls the lips' corner vertical movement. Electrodes on bipolar setting are placed 1 cm apart from its reference electrode. When recording muscle activity of Corrugator supercillii, the first electrode is placed linearly next to the eyebrow and the second electrode is placed laterally and slightly superior to the corner of inner eyebrow. Placement of electrodes to record Zygomaticus Major are placed on horizontal line that slightly diagonal. The first electrode is placed between corner of the lip and earlobe. The second electrode is placed medial and mildly inferior to first electrode. The ground electrode is preferred to be placed on test participant when conducting facial EMG studies. It is suggested that placement of ground electrode is placed on forehead, near the hairline. (Fridlund & Cacioppo, 1986.)

2.4.2 Other psychophysiology measurement techniques

Psychophysiological data can be collected with various other methods in addition to electromyography, that are categorized into eye, brain and skin -related methods. Typically, researchers have used multiple sources of psychophysiological inputs to gain more data and therefore more reliability to study results. Eye-related methods are categorized into Electrooculography (EOG). Electrodermal activity (EDA) measurement is related to skin-related psychophysiological activity. Brain-related measurement method is electroencephalogram (EEG) with specific filtering techniques. (Fritz et al., 2014.)

Eye-related methods are used for measuring cognitive load and task related studies through dilation of pupil size. Eye-movement can show different patterns on how different level of programmers are viewing the provided code. Eye-movement studies can be applied to reading studies, such as code review studies or when studying readability of code for human eyes. Eye-related measurements with psychophysiology is conducted with Electrooculography (EOG). Studies with electrooculography uses heat maps of test participants eye-movement to analyze certain points in viewed material to assess where test participants attention have belabored during the experiments. Heat maps are usually produced with compatible software with eye-tracking hardware. (Fritz et al., 2014; Mansor, & Isa, 2018.)

Electroencephalogram is mainly used to observe and measure brain activity. Electroencephalogram recordings are be obtained by using electrodes that are placed over test subjects head, typically with special lycra cap that integrated is with the electrodes. Brain activity creates alpha, beta, gamma, delta, theta, kappa, lambda and wu waves. These waves can be associated with examples such as; alpha waves are representing state of relaxation, beta waves relate to excitement, gamma waves occur when person reacts to flashing light, delta waves appear when person is in deep sleep, theta waves occur in pleasuring activities, kappa waves reflects thinking activity, lambda waves are related to visual activity and wu waves are representing movement. (Andreassi, 2007.)

Measurement of EEG is done by electrodes that are placed to head of test participants. The electrodes measure these electric waves that brains produce. The electrodes are placed in specific positions in order to measure correct brain waves. EEG measurement devices provides option to record the brain activity, which can be analyzed and searched for patterns. The EEG patterns provides information about test subjects psychological states. This includes emotional states and expressions that can be assessed through EEG waves. Positive and negative emotions have been successfully recorded through EEG and scientist have been able to locate brain activity with corresponding emotional stimulus. (Andreassi, 2007.)

Skin-related psychophysiology is recorded with Electrodermal activity (EDA) measurement devices. This technique is used to measure skin’s conductance or galvanic response on skin. EDA measurement is used to monitor physiological activity that are related to experiences of arousal, attention, emotional states, stress levels and anxiety. (Fritz et al., 2014.)

Theoretical background provides understanding to key concepts of this study and how these concepts have been defined in theoretical articles. On the next Chapter, the theoretical background is explored further with previous studies of equivalent theoretic area. Previous studies of key theoretical concepts aim to provide more comprehension on how specific theoretical background is utilized in scientific studies and findings of those studies.

3. Previous studies

In this Chapter, the theoretical background is explored further by reviewing previous studies that have been conducted with chosen topics related to this study. Previous studies are presented from higher lever to more specific level. First studies of emotions are presented and moved forward to specific programming task studies. Furthermore, the previous studies in emotions in applied fields are explored.

3.1 Emotions and HCI

Saariluoma and Jokinen (2014) have conducted research on user experience related emotions when controlling crane with computer aided tools and performing certain tasks. Their study consists of three experiments, where first two were conducted in laboratory setting and the third one was conducted on field. Participants in laboratory setting were university students and participants of the field study were professional crane operators and designers, which all were considered to be novice users in these experiments since the technology differed from their everyday technology. On the experiments, the test participants were controlling miniature crane by using game joystick and tablet computer. In addition, one experiment was conducted with gesture-based technology as controlling medium of the crane. (Saariluoma & Jokinen, 2014.)

Human computer interaction research tends to asses mainly positive and negative emotions that users' experiences. Assessing positive and negative emotions are made approachable, since there is a lot of prior research on these emotions and how they can be measured. (Fridlund & Cacioppo, 1986; Müller & Fritz, 2015.) Saariluoma and Jokinen's (2014) experiment showed that the test participants felt variety of emotions, from engagement to frustration. Results showed that competence related emotions were experienced when participants could control the crane. Instead, the frustration related emotions were experienced when participants lack ability to control the crane. Results of the study helped to map different emotional states that are experienced by novice technology users. (Saariluoma & Jokinen, 2014.)

Studying users' emotions is growing trend in human computer interaction research. Users' emotions and emotional states in use of computer applications and applications to adapt towards users' emotional states are studied in specific research area of affective computing. In affective computing, the applications are designed to reflect users' emotional states and to provide suggestions for certain actions for users according to their emotional state. Affective computing research utilizes psychophysiological research equipment and methodology to gain base knowledge of human emotions during use of computers. Used psychophysiological equipment has included EMG, EEG and EDA devices. Affective computing can provide guidelines for software developers to build more consistent and usable software applications. (Aranha, Correa & Nunes, 2019.)

3.2 Previous EMG studies

Electromyography, especially facial electromyography, has been heavily used for emotional response studies. Emotion related studies with EMG have concentrated on corrugator supercilii and zygomaticus major, since they are providing reliable results in studies correspondingly to positive and negative emotions. Visual stimulation has been highly used in psychological research to gain results regarding test participants emotional responses. (Tan, Walter, Scheck, Hrabal, Hoffmann, Kessler & Traue, 2011.)

In Tan et al. (2011) study of emotional responses over visual stimulation, the test participants showed high valence in their viewing activity. The test participants were shown 24 pictures and the participants' corrugator supercilii and zygomaticus major were monitored for arousal, valence and dominance related activity. Used baseline was one second between shown pictures. The results indicated high valence, but emotional signals of arousal were not being able to record on significant level. It was implied that arousal might be harder to achieve in laboratory setting and other biometric sensors could be more feasible, for example heart rate and skin conductance, for measuring arousal of the test participants.

Schuurink, Houtkamp and Toet (2008) have also studied corrugator supercilii and the zygomaticus major muscles' valence and arousal activity through EMG. In their study, the engagement of activity was also observed. Their study was conducted over first responder themed serious gaming experience, where test participants were finding failure mechanisms of emergency situation. Results of this study indicated that valence was experienced throughout the gaming experience increasingly, but arousal was measured decreasing from start of the game to its ending. Engagement towards gaming experience was assessed with questionnaire. In the gaming experience, the engagement was assessed through situations of three-dimensional characteristics, causality, interaction, control, communication and persuasion. Results indicated that feeling of being in control of the situation were regarded as most engaging element in gaming task. (Schuurink, Houtkamp & Toet, 2008.)

Dimberg, Thurenberg and Grunedal (2002) have studied how facial muscles react on different pictures. Goal of their study was to investigate if certain pictures that contain images that typically causes negative or positive emotions, or cause involuntary muscle activity in according regions, whether the person does not feel that way. This activity is based on preconscious processing and automatic activation which leads to certain learned facial reaction. Researchers of this study conducted three experiments, where test participants' corrugator supercilii and zygomaticus major regions activity was tested with EMG when different pictures were shown.

Dimberg, Thurenberg and Grunedal's (2002) study consisted of three experiments, which of each were conducted with total of 48 students. On the experiments, the test participants were shown pictures that were associated with positive or negative emotions. Control group of test participants were instructed to try reacting or not to react with opposite facial muscular activity when pictures were shown. Experiments showed that facial muscular reactions can be spontaneous even if person is trying to react differently as expected. The researchers concluded that persons recognize different learned shapes, colors and patterns. This research showed that persons will respond to learned patterns, such as flowers and snakes, as they have learned to response to them in form of smiling or frowning the facial muscles.

Another study found that corrugator EMG activity can be used as a valid indicator of negative emotional responses in studies, whereas zygomatic EMG activity should be interpreted with caution as an index of positive emotional responses, especially when the emotional messages are less extreme. As a further caveat, a slight increase in zygomatic activity, like as a facial grimace, may also be produced by very unpleasant stimuli. (Marghescu, Salminen & Ravaja, 2011.)

3.3 Previous studies in reading

Emotions in reading tasks have been studied in several research studies. Emotions related to reading tasks has been regarded as moment-to-moment emotions, which are activated on the reader through different scenario setting. Defined emotions in these scenarios were boredom, frustration, confusion and flow or engagement. Boredom was found on scenarios where the reader do not find relevant information from the text. Frustration occurs on scenario where the reader doesn’t completely understand the content of the text or the context is presented vaguely. When readers do not purely understand the context of the text, it will lead to emotional state of confusion. Flow state or engagement to the reading task occurs when the text is optimal for the reader and provides correct information. (Graesser, D’Mello & Stahl, 2012.)

Rajanen, Salminen and Ravaja (2015; 2016) have studied reading of newspapers in print and digital form. In their study, they were observing, if test participants experience difference in their approach motivation while reading same content from different types of medium. Test setting consisted of printed newspaper and digital version of same newspaper in tablet computer. The study was conducted with frontal electroencephalographic (EEG) along with heart rate, electromyography and electrodermal activity. The study showed that the test participants experienced higher motivation while reading printed media than digital content. It was also found that reading style had effect on approach, test participants with higher focus had higher approach towards print medium. Additionally, novice users of tablet computers had higher approach reading content from them than printed medium while experienced users had higher approach towards printed medium. (Rajanen, Salminen & Ravaja, 2015.)

Künecke, Sommer, Schacht and Palazova (2015) have studied facial muscle responses while reading. Motivation behind their research was to find out if certain words and patterns have predetermined emotional response. The research consisted EMG test where the test participants were presented wordlists, from where they marked words if they were familiar with them. Test participants were recorded with both EEG and EMG simultaneously. EMG recording was conducted by measuring corrugator supercilii and zygomaticus major muscles. Results of this study indicated that concrete words generated higher valence rating and words that were categorized as positive words decreased negative emotional rate. It was found that abstract words did not produce as high valence as concrete words. Abstract words had also decreasing effect on experienced valence.

3.4 Previous studies of emotions in programming

Following sections are presenting how different researchers have studied emotions in programming. Sections are divided based on different studies and named after the researchers. These selected studies describe how emotions are studied in field of computer science, in scope of programming activity.

Graziotin, Wang and Abrahamsson

Graziotin, Wang and Abrahamsson (2015a) have studied software developers' emotions during their development work. The study was conducted by short questionnaires that were filled every ten minutes during one and half hour of working period on software development task. The study consisted total of eight participants and they were interviewed before and after the task in addition to the questionnaire. Pre-task interviews were used for gathering information about participants background and the post-task interviews were used to evaluate the participants self-assessed productivity during the tasks. The filled questionnaire was used to measure test participants valence, arousal and dominance of and compared if they had positive correlation to test participants self-assessed productivity. Results of the study showed that test participants with expert level of programming skills had positive correlation between valence and self-assessed productivity. This indicates that they enjoyed more their work and were more productive than participants with inadequate skills. (Graziotin, Wang & Abrahamsson, 2015a.)

Khan, Brinkman and Hieros

Khan, Brinkman and Hieros (2010) have studied how emotions and moods are affecting on programmer's performance, the researchers focused on debugging activity of the programmers. The researchers conducted their study by conducting two experiments. On the first experiment, the test participants were assigned to watch five different video clips with certain emotional content to invoke individual emotions. The video material invoked emotions that are connected to valence and arousal. After each video, the test participants were assigned to debug shown a piece of software code and mark errors in it.

The test participants of Khan, Brinkman and Hieros (2010) on the first experiment were first shown a neutral video clip in order to have a reference for their debugging activity in order to see covariance between different moods and neutral mood. Total of 72 participants took part in first experiment. Results showed that emotional states and moods had effect on test participants performance on debugging tasks. High level of arousal state found to be hindering debugging activity and producing more errors on tasks, but no significant correlation was detected with high valence and debugging on this experiment.

On the second experiment, level of arousal was reduced from test participants by interventions. The first intervention took place after 16 minutes of starting of debugging activity and with second intervention after 8 minutes of first intervention. Total of 19 participants were used in this second experiment. Results of the second experiment showed that the interventions helped to reduce test participants' arousal level and improved test participants' performance with the tasks. On an overall level, this study indicated that with optimal level of arousal level, the software developers can perform from different task more effectively than on overwhelmed by mood levels. This can be interpreted into that working conditions can have huge effect on development tasks succession. (Khan, Brinkman & Hieros, 2010.)

Fritz, Begel, Müller, Yigit-Elliott and Züger

Fritz, Begel, Müller, Yigit-Elliott and Züger (2014) have studied if tasks' difficulty has effect on programmers' emotions. The researchers tested on this study whether the emotions with the tasks could be observed by utilizing psychophysiological measurement techniques. The study consisted total of 15 test participants that were professional software developers with minimum of two years of working experience and expertise in C# programming language. The psychophysiological measurement utilized in this study were eye-tracking, EDA and EEG -devices. The EDA device measures test participants heart rate.

In Fritz et al. (2014) study, the test participants tasks consisted of reading the provided source code from computer screen and to answer questions regarding them, without actually running the code. Test participants had also to utilize the think aloud technique. The researchers used screen recording and eye tracking for observing test participants movement on the screen. The results of this study showed that the researched could utilize the psychophysiological devices and techniques to measure the task difficulty. The results provided categorization for easy and difficult tasks that were experienced through the tests and that these methods could be used for improving software development process. Limitation of their study was that the researchers only tested professional level software developers. (Fritz et al., 2014.)

Müller and Fritz

Müller and Fritz (2015) have proceeded their previous study further to exanimate how emotions and programmers' experiences interact with change of programming tasks. They're also interested in their study how biometric sensors ability to record emotional changes between tasks. Their study involved 17 test participants with two tasks, which each took 30 minutes. The test tasks consist of Java programming assignments and test participants could use Internet for searching for information regarding their tasks. This was based on suggestion on how programmers would normally do with these tasks. The participants wore biometric sensors, which were: EEG headband and wrist band to record skin heat and heart rate. Participants consist of six professional software developers and eleven PhD students in computer science. Professional software developers average mean of professional experience was 7.1 years, ranging from one year to 29 years. Background information of participants were gathered beforehand of the tests. (Müller & Fritz, 2015.)

Müller and Fritz (2015) instructed in their study the participants to watch two-minute-long fish tank –video to bring out their baseline, which helps participants to normalize their emotions and relax, studies show that baseline activity is reached after two minutes of relaxing activity. This was also conducted in change of the tasks. This method helps to record accurate biometric data. Participants were interviewed after the test to gather information about their feelings related to change of tasks. (Müller & Fritz, 2015.)

Results of study conducted by Müller and Fritz (2015) did not show major difference between software development professionals and PhD students. The test tasks were mostly completed in these two groups. Participants emotions varied from feeling of being stuck to flow-state within these two tasks. The second task was on overall-level more frustrating than the first task. Results shows that variance between emotional levels are based more on the individual level than on the overall level, which means that individuals experienced these tasks differently. (Müller & Fritz, 2015.)

On Müller's and Fritz's (2015) study, participants told at post-interview that most increase in their emotions happened when they found relevant point in the coding task. As an opposite, the difficulty to understand the code or the task decreased their emotions or progression on the task. Other increasing effects towards emotions and progression in tasks were gain from understanding of the code properly and clear steps of progression of the task. Decreasing effect in emotions and progression were gain also from difficulty to locate relevant code in the task and uncertainty of the next step in task's progression. Results indicate that programmer's emotions could be calculated or recognized due to machine learning aided classifiers and biometric devices. These devices could help recognizing situations when programmer could be needing help or when not to be disturbed, as they are in flow-state. Further indication of Müller & Fritz (2015) suggest that biometric sensors, such as eye-tracking devices could be used as refactoring tools or as an automated code smell detector, which would assist programmer when unreadable code is detected. (Müller & Fritz, 2015.)

Müller and Fritz (2015) were interested in their study to acknowledge how programmers deal with situations in their profession, when they get stuck or to avoid negative emotions in programming tasks. Participants told that their strategies when they got stuck, included switching the context, which means changing to another task, talk with colleagues or take a break from the task. Also, by setting clear goals regarding the task or by giving more time to handle the task with proper resources, were common strategies for avoiding getting stuck with tasks. (Müller & Fritz, 2015.)

Kinnunen and Simon

Kinnunen and Simon (2012) have conducted several studies analyzing university students' experiences in introduction course of programming. Goal of these studies were to find out how computer science students experience these introduction courses and why there is large number of dropouts at these courses. Students' experience over programming was gathered about student's course work, rather than their actual programming experience. Total of 18 freshmen students were interviewed for the study. Students' were interviewed every other week during the course. (Kinnunen & Simon, 2012.)

Results of Kinnunen and Simon's (2012) study revealed that programming tasks were experienced by students quite stressful and occasionally confusing. Students often found some negative aspects in their self-evaluation of their progress on programming tasks, although their tasks were conducted correctly. Students progression on tasks were mainly categorized in two stages, being "completely lost and on some track". Students being "completely lost" often got stuck with their task and waited for help from co-student or teacher's assistant. On this stage students' emotions were usually strongly negative towards their own capabilities. Instead, students that were "on some track", were confident on progressing on tasks, but they often still self-evaluated themselves negatively, depending if they succeeded with help of external resources or on their own. (Kinnunen & Simon, 2012.)

On this Chapter, the relevant studies were presented, in order to gain insight to how key concepts were studied earlier and what kind of results these studies have provided. Methods and results of previous studies provides background aspects to conduct this master's thesis' study. Next Chapter describes what research methods are used in this study and describes them in theoretical scope.

4. Research methods

In this Chapter, the relevant research methods of this study are being presented. Selected research methods focus on quantitative methods and psychophysiological research due the nature of this study. First, the basics of research methods that are relevant for this research are being discussed. Then, the data gathering of this research is being discussed and considered. Finally, the psychophysiological research method is discussed in detail. Research method concepts are presented from higher level to more detailed level.

4.1 Quantitative research

Quantitative research is common methodology in statistics science and psychology, which traditionally relies in statistical data. The quantitative research tries to find relations between theories and research hypothesis through quantities of data. Statistical analysis is used in quantitative research for making sense of the gathered data and to gain answers and support for research hypotheses' or to test theories. Psychological research tends to compare different datasets from prior research to gain support in researchers' own investigations and hypotheses. (Creswell, 2014.)

In quantitative research, a philosophical aspect of research can be applied to enhance the research legitimacy. One of these aspects are positivism, which tends to find factual data from the research settings and results. Testing of data relies to positivist research philosophy, where received results validity is regarded as an important factor of succession of the research. In positivism, the research questions or hypotheses of research are set beforehand of the research. (Coolican, 2009.)

Quantitative research methods comprise of conducting experiments, ask questions from test participants and observing test participants. On experimental research, the researcher tends to manipulate the test setting for evaluating differences in results. Other quantitative research methods tend not to manipulate the test situation and attains to understand and verify the studied phenomenon. (Clark-Carter, 2004.) Premise of quantitative research is to gather and study quantities of data. Quantitative research's data gathering methods and analyzation are serving its purpose to validate and compare variables that are presenting numeric values. In cases, qualitative data can be obtained and treated as quantitative data, if the data can be turned in numeric amounts. (Coolican, 2009.)

4.2 Survey research

Survey is common research methods in quantitative research, and survey is used to measure sample of population. Surveys can be utilized by using structured or unstructured forms, which produces different results. Surveys can contain several parts, including questionnaire, which can be used to find out public opinions of certain focus point. Survey data can be analyzed in quantitative or qualitative methods, depending on surveys initial form, in case of open-ended questions or Likert-scale questionnaire. Surveys should always follow logical form, that is clear and person who is answering can understand used language of the surveys. Surveys questions should follow certain scheme which justifies their order and makes them appear logically in canon. (Lazar, Feng & Hochheiser, 2010.)

Questionnaires among interviews and telephone surveys, have been most used forms of surveys. Questionnaires are aiming to provide information from the test subject for the researched subject. One negative side is that questionnaires may have limitation for the responders to present their opinions for the research subject, if not provided correct questions and forms. (Goddard & Villanova, 2006.)

Surveys or questionnaires can contain different scales that are used to measure answers for questions in the survey. Questions can be closed-ended, where the test participant may answer accordingly to certain scale or he must choose preferable option, for example; “yes” or “no”. Using open-ended questions in survey is efficient method to gain deeper understanding of studied phenomenon and gaining supplementary information for research questions. (Lazar et al., 2010.)

4.3 Experimental research

Psychological research aims to describe and understand human behavior and psychological process behind the behavior. Scientific experiments and statistical analysis have played key role in history of psychological research and provided suitable frameworks to gather feasible information from human behavior. Experimental research method is designed on premise of testing variabilities between groups, in order to test predetermined hypotheses of the research. The research hypotheses are derived from prior studies and theories. Hypotheses are suggesting something that will or will not happen during the experiment or research setting that involves test participants. (Davis & Bremner, 2006.)

Experimental research is one aspect of observational research’s dimensions. Observational research aims to understand human actions and behavior through interpretation of the observation. Observational research’s dimensions include theory testing, which opposes explanatory observation. Theory testing tries to understand action related to theoretical background whereas explanatory observation aims to understand process within observed actions and behavior. (Dallos, 2008.)

In experimental research, null hypothesis and alternative hypothesis are usually formed. The null hypothesis implicates that the experimental examinations do not differ from each other and the alternative hypothesis implicates exclusion of the null hypothesis. Experiments as a study method are aiming to invalidate the null hypothesis through results of the study. (Lazar et al, 2010.)

Experimental research makes possible to replicate previously conducted studies, in order to test the previous results with differentiating the sample. Experimental research can be replicated by close or differentiated replication. Close replication is conducted by repeating previous study as accurately as possible to see whether similar results are gained. Differentiated replication is conducted by repeating previous study with same research questions, varying the used experimental methods and conditions. (Wohlin, 2012.)

4.3.1 Experimental research settings

The research setting in experimental research is constructed to have variance in its results, to see if there is difference between the test participants behavior in certain cases. This is usually gained through manipulation and control over the research setting, by independent or dependent variables. The independent variables are seen as the entities that are bringing the change into research setting, whereas the dependent variables are being brought to experiment by the researcher to raise variance in the test participants. (Davis & Bremner, 2006.)

One of the data collection possibilities in experimental research is laboratory research. Laboratory setting can provide the researchers a controlled environment where they can conduct the experience research. Laboratory setting is used to test hypotheses and theories regarding real world settings. (Zaccaro & Marks, 1996.)

The results of experimental research are formalized through reliability and validity. Reliability of experiment is obtained by repetition of the test setting. Validity in experimental research needs more effort than reliability, since it measures if the test results are correct and in sense of causality reliable. The causal reliability and relations are covered through randomization in experimental research. In randomization, the test factors, like test tasks or the test equipment, are allocated by random chance, for example, by lottery, to the test participants. (Davis & Bremner, 2006; Lazar et al., 2010.)

4.3.2 Controlled experiment

In technological or software engineering field, experimental research can be conducted in human-orientation or technology-orientation. Human-orientation, test subject can be tested by test participant with two different methods, for example by two different methods are used on two sets of different source code. On contrary, technology-orientation, different tools are being used to targeted test subject, for example different tools can be used on same computer program by the test participants. It should be regarded that technology-orientation provides more control into the experiment than the human-orientation. (Wohlin, 2012.)

Controlled experiments can be conducted as controlled observations, which are usually conducted in laboratory setting. The laboratory setting provides environmental control into the experiment, although controlled experiments can be conducted in other environments as well, such as meeting rooms. Laboratory as a setting may cause anxious activity in some test participants, since it may not be normal environment that may be used in field experiments. Positive side of laboratory setting is that it provides closed environment, where daily routines can be put aside and therefore higher accuracy to results of the experiment than in field experiments. The nature of experiments determines, if there is more need for controlled or uncontrolled experiment. (Coolican, 2009.)

4.4 Humans as test subjects

Humans are regarded as individuals with individualistic characteristics, therefore studying humans has its challenges. In quantitative studies, humans as test subjects are studied in samples, that are representing certain part of population. The population is regarded as group of people that has certain similar background, like education, and sample is selected group that is representing the population in study. (Coolican, 2009.)

The nature of psychophysiological experiments can be psychologically very invasive, since aim of the experiments is to gather information about human factors. Presence of electronical measurement devices that are attached to persons can also be very invasive experiment for someone. It is suggested for the researchers to discuss with the test subjects beforehand of the experiment what are the devices and how they are functioning in order to achieve some level of calmness in test participants. Each step of the experiment should be discussed and instructed to the test participants, since they may contain physical contact. (Fridlund & Cacioppo, 1986.)

In research, one method of sampling is quota-sampling, which enables researcher to take group of people from the initial sample. Quota-sampling enables researcher to take sample of people that are fitting in predetermined characteristics, for example, certain age group. The quota-sampling is common method in market research, where researchers are finding results that are fitting for certain group of people. (Coolican, 2009.)

4.5 Data gathering

In observational research, the researchers may use audio or video recordings to gain additional data from the observation sessions. The recordings of the sessions help researchers to recall occurred events and behavior of test participants. Recordings can reveal and add new perspectives to gained results from the observation and therefore help to draw strong conclusions on the studied behavior. (Dallos, 2008.)

Psychophysiological data is true ratio level, which means that the data should be treated on interval level measurement with true zero applied. The nature of psychophysiological data is true and quantitative, but it’s analyzation may require qualitative methods for understanding the psychological concepts behind the recorded behavior. (Sowden & Barrett, 2008.)

4.5.1 Quantitative data

Quantitative data is presented in numeral form and therefore it is measurable. Quantitative data is measured by four scales, which are nominal, ordinal, interval and ratio. The nominal scale consists of two categories that can be applied to result. The ordinal scale is applicable when the resulted data can be placed in ordered. The interval scale can be applied if two points of data are comparable and to see their distinction. The ratio scale is used to gain difference between two points of data. The ratio requires the data to have true zero, which means that the data has no quantity on the point zero. (Clark-Carter, 2004.)

Quantitative data is measured by descriptive statistics and presented in suitable form. Data analysis utilizes numerical methods that describes the presented data and forms connections between the data and studied phenomenon. Analyzed data is used further to test hypotheses of the research. Statistical significance is tested from the analyzed data to see if it rejects or supports the proposed research hypotheses. (Clark-Carter, 2004.)

In statistical analysis, mean is used for calculating average values in given data sets. Mean can be used for searching differences and correlations between population parameters. Mean can also be used for measuring distance between interval of values in data sets and to provide their middle point. Mean values show if the data is normally distributed and therefore provides information of true population. Normality of distribution can be tested with Shapiro-Wilk test. Using mean as a measurement indicator has its pitfall examinations of individual level of values. Individual values may cause dissonance in the value of mean if it contains outlier value. (Coolican, 2009.)

Another method for calculating population parameters is standard deviation, which is used to calculate distribution of data. Standard deviation takes all given values in data set and calculates square root of deviation. Standard deviation disregards extreme values in the given data set. (Coolican, 2009.)

In statistical research, means and distributions provides important information about the sample and indicates what test should be conducted with the sample data. Parametric and non-parametric testing are applied to test sample's relation between different conditions. Parametric test is used, when sample is normally distributed. Non-parametric tests can be used, if sample size is considered to be insufficient or skewness of the sample is remarkable. This method is also applicable when studying sums of means. In parametric tests, Levene's test is used to test homogeneity of variances and to produce t-test for significance of provided values. Variance of normally distributed and homogenic variances are tested further with one-way ANOVA against the null hypothesis. In non-parametric tests, Wilcoxon *T* matched pairs signed ranks test is used to test differences of studied sample groups and to test H_0. (Coolican, 2009.)

Questionnaire's that use Likert-scale produces categorical data. Provided categorical data can be tested with chi-square test, which indicates if the results are associated to given proposals. Chi-square tests are used also to test null hypotheses against the results. (Coolican, 2009.)

4.5.2 Psychophysiological data

Psychophysiological data is gathered through used hardware as electronic signals that are analyzed with appropriate tools and methods. The gathered data's accuracy is tied into the used hardware and sensors. It should be noted that the psychophysiological signal data is prone to noise and disturbance. (Sowden & & Barrett, 2008.)

In psychophysiological research, the absolute threshold is commonly being searched from the signal data. The absolute threshold shows the minimum level of stimulus needed while recording the signal data. Absolute threshold can be hard to find, since humans are individual beings that behaves differently on stimuli and therefore provides different levels of data. The individual differences also have effect on level of noise, which is information loss, within the data. It should be noted that human's biomedical signal can be full of noise and therefore the absolute threshold can be hard to notice from the recorded signal, since they might be overwhelmed by the received noise. (Rose, 2008.)

Noise in biomedical signal gathering can be caused from various sources. These sources are physiological, device or testing environment based. In EMG recordings, the physiological interferences from test participant such as coughing or changing position crucially can cause noise. Testing environment can produce different interfering signals that produces noise into the recordings. Environmental interference is usually caused by display monitors, power cables and electronic devices. Environmental interference can be reduced with grounding and electromagnetic shielded cables. In more intense physiological signal recordings, such as EEG recordings, use of Faraday cage -like setting of the laboratory is recommended. (Rangayyan, 2015.)

The Absolute threshold can be tested by using trials or pilot tests where intensity of stimuli is being tested, varying from zero percent to hundred percent. The recorded data includes the threshold and the noise. This method is being called constant stimuli. Finding of the absolute threshold and estimation of noise accuracy usually needs hundreds of short trials, but for example in clinical studies, more efficient methods are being used for fast paced nature of clinical work. In these environments, the stimuli are being recorded near the maximum threshold. These test gives clear presentation where the stimulus is most intensive. (Rose, 2008.)

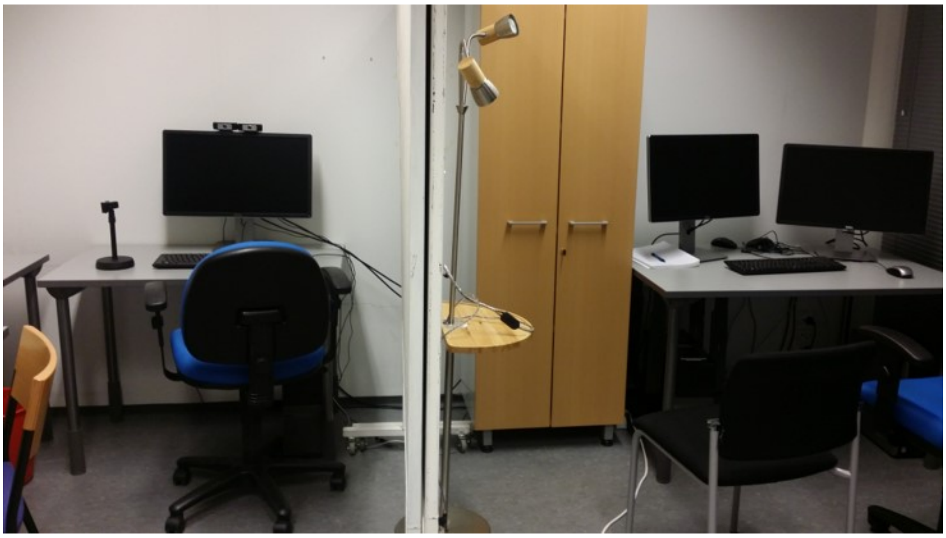
There are several other methods to study recorded stimulus. One is to study the difference threshold. In this method, the recorded stimuli are being compared with another stimuli and minimum difference is being searched between them. This method can be tested with the constant stimuli method. (Rose, 2008.)

5. Research implementation

In this Chapter, the processes of this study's experiment are presented. The analysis process of the results is presented in more detail. The laboratory setup for EMG recording and experiments are presented in detail. Used analyzation methods for EMG data of this study is presented and process flowcharts are presented for clarifying the analyzation process.

5.1 Research setting

The research setting was located on premises of University of Oulu, at INTERACT Research Unit's UX-Lab, which contains monitoring computer and test computer. The room is divided into two parts, separating the computers.



Picture 2. Picture of the UX Lab (Retrieved 2019, February 22 from https://interact.oulu.fi/site/files/FilesDorina/uxlab.jpg).

Picture 2 shows that the UX lab contained two computer stations. On right side of the Picture 2, there is computer that the test participant uses for conducting the monitoring or administrator computer of the lab is on the right side. Between these two computer stations is room divider that separates this lab space. Administration computer has two screens, but the other is actually connected to the test participant's computer for monitoring and measurement purposes. This additional screen shows copy of test participant's screen, therefore test participants actions can be monitored in real time.

Test participant's computer contains Camtasia software, which is used in this study to record the screen while the test participants conduct the programming tasks. Test participants screens were recorded for analyzation purposes, mainly to verify correct points of data in the blocks. Video analysis was used to determine, when the test participant entered to next task and then that part of data blocks was mapped to that specific task. Recordings were also used to see additional or interesting activity in the test cases. These cases are discussed further on next Chapter, where the experiment results are presented.

5.1.1 Research tools

Research tools of this study are divided into EMG measurement tool and to analyzation tools. EMG measurement tool is used for measuring signal data through electrodes that are being recorded with specific measurement hardware. Analyzation tools includes used tools that benefit research analysis process. Analyzation tools includes software that is used for handling the recorded EMG data, statistical analysis software and video recording software.

EMG measurement tool

The used EMG measurement tool in this study is PsychLab's EMG recorder provided by University of Oulu. The recording device consist of base station amplifier that contains sockets for chords for surface electrodes. The base station is connected to computer by USB cable. PsychLab device comes with suitable analyzation software, but the data can be analyzed with third party software as well. The PsychLab software provides synchronized video with the recorded physiological data. (PsychLab Hardware Manual, 2009.)

Analyzation tools

The used analyzation tools include PsychLab 8 psychophysiological analysis software and IBM SPSS statistics tool. Video recordings of this study were recorded with Camtasia 3 software. After the preliminary analysis and data processing with PsychLab 8, the results are analyzed and tested further with IBM SPSS statistics software.

The PsychLab 8 is analyzation software designed for PsychLab psychophysiology recording hardware and provides raw waveform data. The provided data is recorded continuously, therefore researchers have access to all relevant data, but also have to locate the relevant data from the recording. Recorded data is then divided into blocks by the researcher. Data blocks are used to analyze the particular set of data, also multiple sets of data can be analyzed and compared. The provided data is handled by dividing it to appropriate amount of data blocks and then smoothened and rectified. Then the data's key figures are exported in numeral format for further analysis on statistical software. (PsychLab 8 Software Manual, 2009.) In this study, IBM SPSS is used for statistical analysis software. SPSS provides different analysis methods that can be used to analyze and test the data.

Camtasia software 3, is video recording software, which provides options for different video recording methods and tools for converting video material to different format. In this study, Camtasia software is used for recording the screen of the test participants (Camtasia, 2019.) Screen recordings are used for verifying the lengths of the programming tasks. Test participants spent different amount of time to conduct specific task, therefore, it is needed to verify how much time each task takes. This time is compared to starting time of different blocks that are gained from PsychLab software. The blocks starting and ending time with recorded video helps to map correct blocks of data to specific programming tasks.

5.1.2 Test Participants

The preferred target group to this study was planned to be first- and second-year students of Information Processing Science, Computer Science and Electrical Engineering of University of Oulu. The recruitment emails were sent to mailing lists of the first-year students of these departments. Another recruitment email was sent to more general mailing list, which covered every student of Information Processing Science, Computer Science and Electrical Engineering in University of Oulu and everyone else that has been registered to that list, for example minor students.

The recruitment message emphasized that first- and second-year students would be anticipated for this test and did not out rule others participation, since these lists contains students that may not be majoring within these study subjects. Participating students were rewarded with a movie ticket for their participation in this study.

Initially, total of six students were recruited to test, but only 4 students were able to participate to the tests. From these participants, one was majoring in Biomedical Technology, one was majoring in Information Processing Science and two were majoring in Computer Science. All recruited students were master's level students. Therefore, original target group were modified, but was accepted, since these majors included programming within their curriculum.

The test participants age ranged from 24 to 54 years; average age of the test participants was 35,5 years. Three of the test participants were man and one was woman. Two of test participants had studied in current major for two years and other two had studied one year in their major. Three of the test participants had prior programming experience, two of the participants had prior experience with Python and two had prior experience with Java. Only one participant did not have any prior programming experience. One of the test participants prior programming experience in years was two years and other two participant's experience in years was one year. Two of the participants had gained their experience from previous studies, one was self-taught.

5.2 EMG setup

The EMG studies requires proper preparation of the test participant and specific placement of surface electrodes or the interested muscle. The surface electrodes require less preparation when placed, than electrodes with needles. The test participant's skin has to be cleaned with alcohol or with other purging liquid over the placement area of electrodes. Then the surface electrodes with two-sided sticker tape, that are applied with paste gel, are placed over the muscle of interest. In recording of the muscle activity with EMG, the usual setting consists of two bipolar active electrodes that are placed over the muscle of interest and one inactive electrode, which service as a ground. (Andreassi 2007; Fridlund & Cacioppo, 1986.)

There is known problem with EMG setup when recording activity of frontalis. The problem is that there are two individual muscles placed near each other on frontalis, over the eyebrows. There should be applied careful placement of the electrodes on the single muscle, preferably on lateral placement to gain more accurate results. There is change that no recording is gained if both frontalis muscles are being recorded with same pair of active electrodes. (Andreassi, 2007.)

Previous studies have indicated that there might be crosstalk with different facial muscles that are close to each other. This crosstalk can cause interference with recorded EMG signal. However, previous studies have shown that amount of crosstalk between facial muscles on activities such as smiling, and frowning have only small impact on recorded signal. (Rantanen, Ilves, Vehkaoja, Kontunen, Lylykangas, Mäkelä, Rautiainen, Surakka & Lekkala. 2016.)

Facial activity recording with surface electrodes may have difficulties to locate certain muscles. Therefore, it is suggested to record activity on certain area of face, one example of this is to record activity muscles around brows. This area contains muscles that can raise and lower the brows. In these experiments, the EMG signals are referred preferably as EMG activity of that certain site than specific measured muscle. (Fridlund & Cacioppo, 1986.)

It is noted that test participants are usually conscious about surface electrodes, which may affect the results and test participants may try to please the researcher. In some cases, additional electrodes are placed for misguiding the test participants to reduce intentional reactions in interest muscles. Researchers should not state to test participants which reactions are observed to reduce test participants intentional actions and reactions in recorded muscles. (Andreassi, 2007; Fridlund & Cacioppo, 1986.)

5.3 Structure of the experiments

Process of the experiment is presented on next on an overall level. This process includes flow of the experiment and description of the steps in the experiment process. EMG tasks are being presented in more detail on own process chart and the used materials are discussed in subtitles of this Chapter.

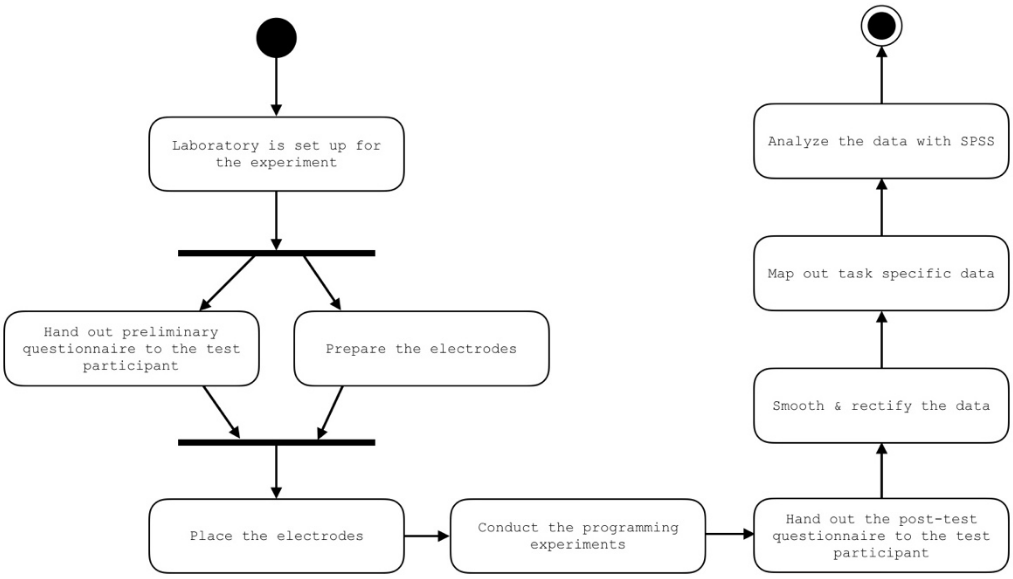

Laboratory is set up for the experiment
Hand out preliminary questionnaire to the test participant
Prepare the electrodes
Place the electrodes
Conduct the programming experiments
Hand out the post-test questionnaire to the test participant
Smooth & rectify the data
Map out task specific data
Analyze the data with SPSS

Figure 2. Process of the experiment.

The overall structure of the tests is presented above on Figure 2. At first, the laboratory is being set. This includes opening of the monitoring computer and test computer, setting up EMG device into the monitoring computer and attaching electrodes to the EMG device. When the laboratory is ready for the testing, the test participant is taken in and is handed out with preliminary questionnaire that is used to gain information about test participants background. The electrodes of the EMG device are being prepared by adding stickers caps and paste gel for receiving accurate signal. Another noise reduction method used in this study is to prepare test participants skin with antiseptic liquid for cleansing effect.

Preparation of the electrodes can be done beforehand, on setting up the laboratory step, if there are more than one researcher conducting the experiments. On this experiment, the process of experiment presented on the Figure 2 was applied, since there were only one researcher conducting the experiment and the laboratory's setting supported this process, since it lacked space where prepared electrodes could be laid down.

When the preliminary questionnaire was done by the test participant, the electrodes are placed over related muscles of the test participant according to guidelines. After this, the test participant is presented with short introduction to tests and handed out cheat sheet, which contains some help to programming tasks and terminal commands to compile source code in every task. Questionnaires, cheat sheet, introduction and programming tasks are presented in English or in Finnish depending of request of the test participant at the beginning of the test session.

5.3.1 EMG tests

The conducted EMG test consist programming tasks of two modules, programming with C and Python languages. Both C and Python programming tasks, consist of three different programming tasks that are executed in repl.it, which is browser-based programming environment (repl.it).

The test persons are given total of 15 minutes to be execute given programming tasks in the module. Between the modules, a one minute of relaxation period is being taken. Total of 30 minutes is valid for test data from the recordings. EMG tests are recorded with epoch length of 1000 milliseconds. PsychLab's (2009) manual suggested to use 500 millisecond epochs, which were also default setting of PsychLab's hardware. The used length of epochs of this study were based on overall duration of the recording. Setting of PsychLab EMG measurement hardware's screening range of epochs was set on the software for minimum value to 0µV and for maximum value was set to 450µV. These settings modified the sensitivity of measurement electrodes. The screening settings was based on PsychLab manual's recommendations (PsychLab, 2009).

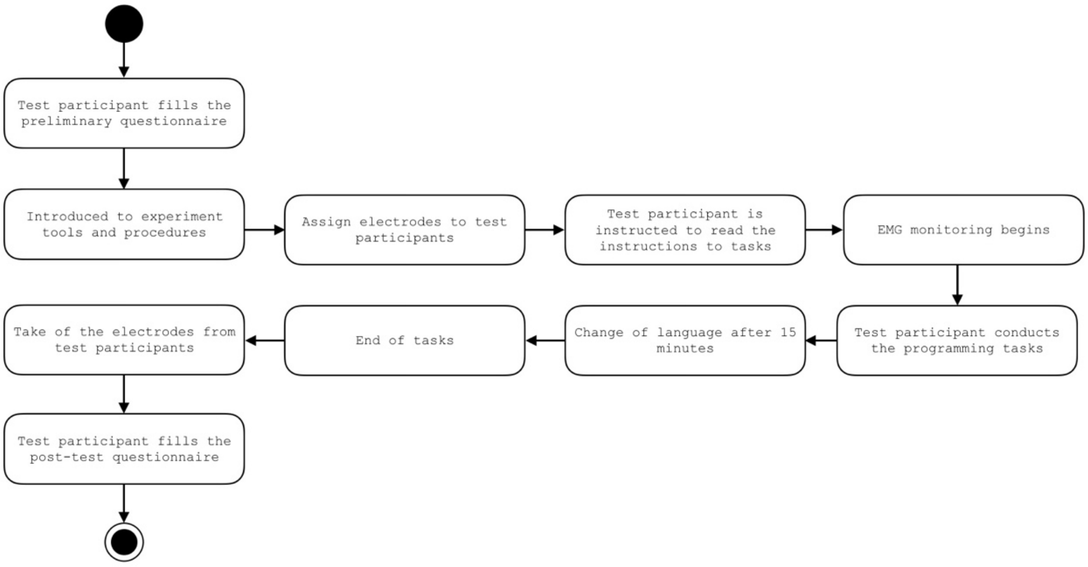

Test participant fills the preliminary questionnaire
Introduced to experiment tools and procedures
Assign electrodes to test participants
Test participant is instructed to read the instructions to tasks
EMG monitoring begins
Test participant conducts the programming tasks
Change of language after 15 minutes
End of tasks
Take of the electrodes from test participants
Test participant fills the post-test questionnaire

Figure 3. Process of the tasks.

On Figure 3, the process of task is shown in more detail. The process begins with handing out preliminary questionnaire for the test participant. The preliminary questionnaire contains questions regarding background of the test participant. When test participant has filled out the questionnaire, then the test participant is introduced to used procedures and devices of the experiment. After the introduction, test participants skin is prepared and the electrodes are placed over to the test participant on top of monitored muscles, then the test participant is ready to start the programming tasks. Progression of this experiment was partly similar to the Rajendra Desai's (2017) experiment but was improved and assessed to match this study's context in order to gain relevant data for this research.

The programming tasks begins short introduction to used programming languages and links to the tasks. Before the programming tasks are started, the baseline or period of relaxation when test participant facial muscle activity is neutralized. This relaxation occurs also when switched between the tasks. The test participant begins with either C or Python programming tasks. This will be decided by coin toss to achieve randomization for the test. Test participant has 15 minutes to conduct three programming tasks with the first given language, after that, new baseline is recorded and then another programming language's tasks are being started.

Finally, after the tests, the test participant is handed out with post-test questionnaire. This questionnaire gives additional information about test participants succession and experienced emotions during the tests. The test participants are rewarded with movie ticket when the test process is finished.

5.3.2 Analyzation of EMG tasks

Figure 2 shows process of analyzation of the EMG tasks after the test participant is handed out the questionnaire and finished it. This process begins after the test participant is brought out of the laboratory.

First step of the analysis process is to start processing of the recorded data. The recorded data is formed through epochs as continuous line of signal, that is needed to be split into smaller pieces, known as blocks. The blocks are split by finding suitable duration that is common for every recording. Finding suitable duration for blocks depends on overall duration of recorded data sets, since it is encouraged that blocks count is even on every compared dataset. In this study, the used length of blocks were 10 seconds. Even counts can be achieved by limiting the recording time for every data sets. After the split into blocks, the data was processed further by rectifying and smoothing the overall data. After the data is processed, suitable scopes of data are being obtained. (PsychLab, 2009.) In this study, data's average, standard deviation, minimum and maximum values are obtained from every block. These descriptive statistics are used in analysis process of the recordings.

Recorded EMG signal have to be treated to gain accurate results regarding the targeted action or muscle. Typically, EMG data provides information on how long the muscle has been active or how strong the level of activity has been. Raw EMG data is usually normalized for more precise and readable results that can be analyzed further. Normalization treatment can be conducted with different methods. One normalization method is to measure mean of the recorded data. This method is used for assessing recorded signal mean amplitude on given task. It is noted that mean method of normalization should be utilized if recorded EMG data is used in maximizing the reduction between test participants results variability. (Burden, 2010.)

In Partala et al. (2006) study, the EMG data was first rectified and corrected by removing irrelevant data from it, including dissonance created by eye blinks. The data was categorized according to research setting into negative, neutral and positive emotions. Statistical analyses were used to assess the data. The researchers tested the data with t-test to determine if the data was significantly different to chance rating. Paired t-test was were used for testing the pairs. The valence was tested with pairwise comparison with matched-pairs analysis. The results indicated significantly more positive relation on positive results than on negative results. Results significance is being tested to reject or accept null hypothesis (Coolican, 2009).

The surface EMG electrodes produce small impedance, since they are made of metal and are conducting electricity for measurement purposes through send signal. High level of impedance between skin and electrodes can reduce the signal strength and cause distortion in recorded waveform, which will make analyzation of the EMG recordings difficult. These high levels of impedance are managed with signal amplifiers. Other measures include using conductive gel or paste on the electrodes, since dry electrodes are causing more impedance on skin contact than paste-coupled electrodes. Another measure is to prepare point-of-measurement on skin correctly. (Clancy, Morin & Merletti, 2002.)

5.3.3 Questionnaires

The test participants are presented with two questionnaires as a part of this research. First, the test participants are given preliminary questionnaire for gathering information about test participants' background information After the EMG recording, the additional questionnaire is handed out for the test participants. The post-test questionnaire is used for gathering data regarding the conducted EMG recordings and programming tests. All questionnaires are handed out in paper format.

Background Information

Background information (Appendix A) of the test participants was gathered before of the EMG test to get additional data of the test participants. The gathered background information consists of information about test participants age, sex, study major, years of study, prior programming experience outside university studies. The questionnaire has additional questions for those participants, who answered that they have prior programming experience. The additional questions examine how the participant has studied programming, for example, in primary school, extracurricular activities or by themselves. Additional questions examine also participants experience of different programming languages and how many years they have studied particular languages. Background information form's questions are presented by descriptive and nominal scale, with open form questions. Additional questions are presented as open form questions. Likert-scale to measure test participants test participants programming experience.

After the tests

After the EMG tests, the test participants are presented questionnaire (Appendix B) with questions regarding conducted programming tasks and EMG testing. Test participants' efforts in programming tasks are assessed with Likert-scale based questions. Programming related questions also assess the test participants' experienced emotions regarding the programming tasks and difficulty of the tasks. The test participants are presented with nine predetermined emotions that are used for describing experienced emotional states during the programming tasks. This question gives the participant possibility for self-reflection. Additionally, EMG test's succession is assessed with open questions as a feedback about the tests.

5.3.4 Programming tasks

The programming tasks of this EMG study contains two modules of programming tasks. Each module contains three different tasks that are executed on selected programming languages, C and Python. The programming tasks are similar by their content, objective of the tasks is to fulfill given part of non-working source code into working and to produce given task. The given task is given on top of the code in comment block. Programming tasks of this study were based on programming tasks that were used in Rajendra Desai's (2017) master thesis, but the tasks were altered to be more suitable for this experiment. Rajendra Desai's tasks consisted too many parts in one task, therefore one of these tasks was split into two and one task was discarded of being too similar as the final task, since both were built over loop-conditions.

Reason for providing partly finished source code for the test participants is to present them problem to be solved. This also simulates refactoring and code reading situation, where the test participants may have to look up more information for their problem. Prewritten code is also more suitable solution for this study, since this method has been used previously with similar studies (Rajendra Desai, 2017).

Presented tasks are shown as whole for easier analyzation process. Here the both C and Python tasks are presented under same task heading in order to ease the comparison of syntax difficulties. Task instruction is presented only on top of the firstly presented task. Sections presented on comment blocks were deleted from presented programming tasks for the test participants. Test participants were tasked to fill in these parts. Presented code in the comment blocks are suggestions for how the program would be working as expected. Comment blocks start in C syntax with `//` and in Python syntax with `#`, additionally comments are on italics for highlighting them.

First programming tasks

Firs task in C:

```c
// Program should get name of the user as an input in terminal
#include <stdio.h>
int main(){
    char name[20];
    printf("Enter your name: ");
    // scanf(“%”, name);
    printf("Nice to meet you, %s.\n", name);
    return 0;
}
```

First task in Python:

```python
#name = input('What is your name: ')
print('Nice to meet you, ' + name)
```

The first task in these modules is to produce working program that requires given string, users name in this context, and prints it to the terminal window of the repl. This task’s difficulty is considered to be easy.

Second programming tasks

Second task in C:

```c
// Program should get two integers as an input and compare them
#include <stdio.h>
int main() {
    int x, y;
    printf("Enter a number value for X:");
  //  scanf("%d", &x);
    printf("Enter a number value for Y:");
    scanf("%d", &y);
  //  if(x > y)
        printf("X is greater than Y");
    else
      //  printf("X is smaller than Y");
    return 0;
}
```

Second task in Python:

```
x = input('Please type a number value for X: ')
y = input('Please type a number value for Y: ')
    #if(x>y):
    print('X is greater than Y')
    elif(x<y):
# print('X is smaller than Y')
```

On the second task, the object is to produce program that asks for two integers and compares them to see which number is larger. After the comparison, the result is printed to terminal window. This task is considered to moderate by its difficulty.

Third programming tasks

Third task in C:

```
// Print pyramid shape using conditional loop
#include <stdio.h>
int main(){
//    int i, j, k=0;
    for(i=1; i<=5; ++i, k=0){
        //for(j=1; j<=5-i; ++j){
            printf("  ");
        }
        while(k != 2*i-1){
            printf("* ");
        //    ++k;
        }
        printf("\n");
    }
    return 0;
}
```

Third task in Python:

```
def pyramid(rows=5):
# for i in range(rows):
        print ' ' * (rows-i-1) + '*' * (2*i+1)
#pyramid(5)
```

On the third task, the objective is to produce program that prints image of pyramid that is made of asterisk symbols to terminal window. This task has more variance between implementations with different languages in its syntax. This task is considered to be more challenging than two earlier tasks by their difficulty.

5.4 Pilot studies

Pilot studies are conducted beforehand of the actual study to test the research setting and methods. Pilot studies are used to assess the used methods in research and getting familiar with the actual research and to see, if there are any alterations needed to be made for the actual research design. Pilot studies can be used effortlessly to assess needed resources and time for the actual tests. Pilot studies includes additional test subjects, who are representing the intended population of the test subjects. (Clark-Carter, 2004.)

Pilot study was conducted prior to actual studies to determine whether the preliminary test setting is working and to determine any possible weaknesses of the study. Pilot study was conducted with supervisor of this master's thesis, who acted as test subject. Findings of the pilot study showed that recording process was quite long when recording both test modules into same file. During the pilot test, the used recording software gave error alert, that notified that the recording would not be saved. This was misleading, since the actual recording was saved on computer.

These findings of the pilot study were considered to be taken into action with the actual implementation of the study. Main improvement for the actual implementation was that the recordings of the tasks was done by the modules, producing two data sets per test participant. Therefore, analyzing C and Python tasks were made easier. Additionally, the pilot test was that the analyzation process of the results could be trained and ensured correct methodologies.

6. Results

In this Chapter, the results of this study are presented in detail. First the research setting, and the research hypotheses are revised, additionally preliminary questionnaire's results are presented to gain knowledge of test participants' background. Then the results of the EMG study and post-test questionnaire are presented.

Research hypotheses of this study are:

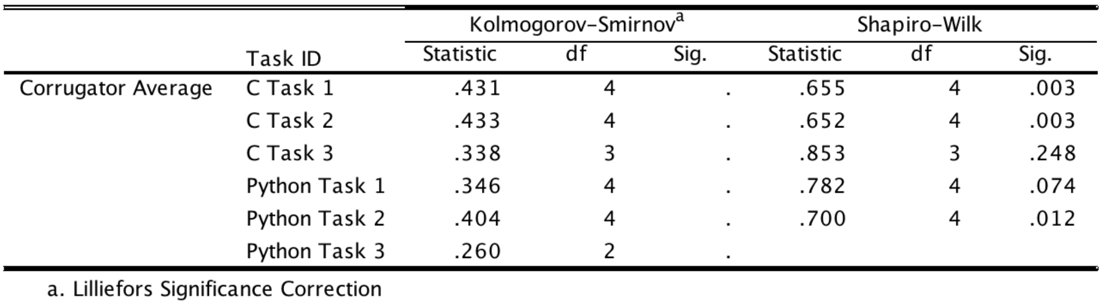
> H_0: Test participants do not experience higher facial muscular activity when programming with either C or Python.
>
> H_1: Higher corrugator supercilii activity is recorded when programming with compared language.
>
> H_2: Higher zygomaticus major activity is recorded when programming with compared language.

Research questions are assessed with the recorded EMG data, which is analyzed on following sections. Theses research questions were inspired by previous studies that had similar context and were conducted with EMG measurement tool. The research questions were formed after the studied context and to answer issues that were found previous studies. These studies indicated that certain images invoked higher corrugator supercilii or zygomaticus major activity among test participants. Aim of these research questions was to broaden the scope of these studies and to compare gained results of this study with previous studies. One goal of this experiments is to study if different programming languages invoke higher facial muscular activity than other. It is assumed in this study that the test participants do not experience higher corrugator supercilii or higher zygomaticus major activity when programming with either of tested languages.

Preliminary questionnaire was presented on previous Chapter. To Recap the results, N of this study is 4 and average age of test participants were 35.5 years. All test participants were either first- or second-year students in master level of their study program. Prior programming experience of the test participants limited into Python or Java.

6.1 EMG recordings

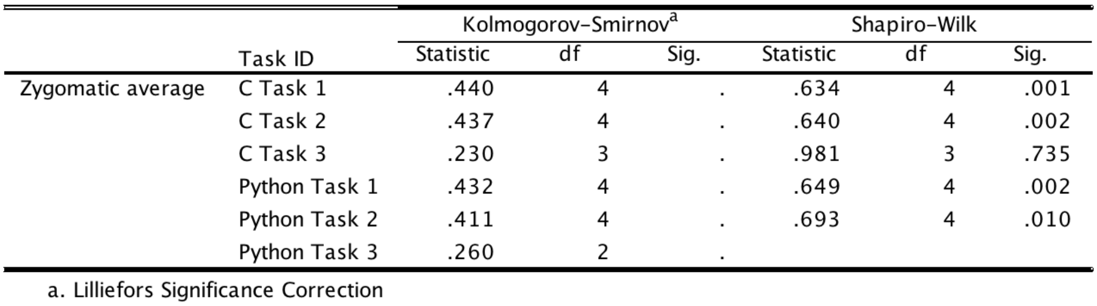
EMG data is used to asses if either of programming languages invoke higher activity on monitored facial muscles. The monitored muscles have been associated with positive and negative emotions; therefore, the results of this test can indicate if a test participant experiences these linked emotions during the programming tasks. Results of the EMG recordings are calculated into mean values and are tested in order to gain reliable results.



		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Task ID	Statistic	df	Sig.	Statistic	df	Sig.
Corrugator Average	C Task 1	.431	4	.	.655	4	.003
	C Task 2	.433	4	.	.652	4	.003
	C Task 3	.338	3	.	.853	3	.248
	Python Task 1	.346	4	.	.782	4	.074
	Python Task 2	.404	4	.	.700	4	.012
	Python Task 3	.260	2	.			

a. Lilliefors Significance Correction

Figure 4. Corrucator Supercillii normality test.



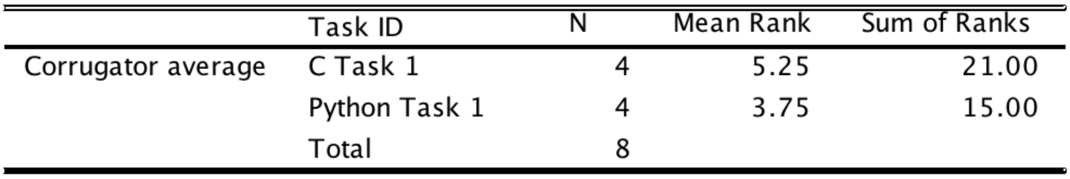
		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Task ID	Statistic	df	Sig.	Statistic	df	Sig.
Zygomatic average	C Task 1	.440	4	.	.634	4	.001
	C Task 2	.437	4	.	.640	4	.002
	C Task 3	.230	3	.	.981	3	.735
	Python Task 1	.432	4	.	.649	4	.002
	Python Task 2	.411	4	.	.693	4	.010
	Python Task 3	.260	2	.			

a. Lilliefors Significance Correction

Figure 5. Zygomaticus Major normality test.

Since first and second tasks group's p-value is greater than .05 and third task's groups are unequal as shown on Figures 4 and 5, and therefore normal distribution is not assumed on individual tasks. Therefore, Mann-Whitney test is used for comparing language differences on results. This method is used for comparing each task and language group with equal method.

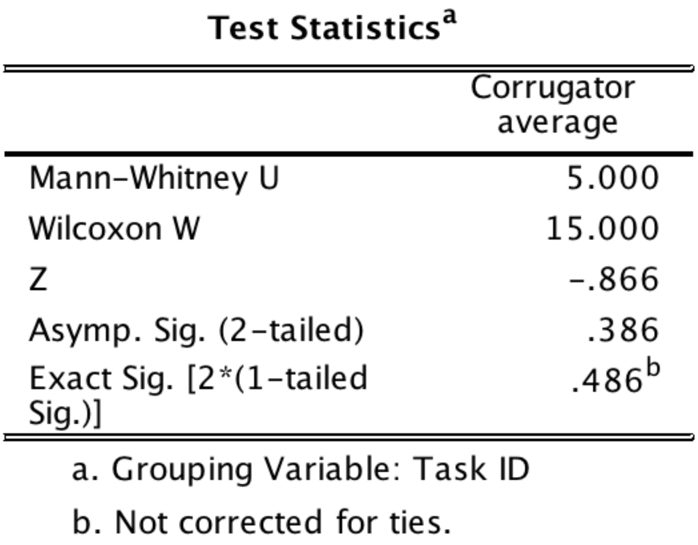
First Programming task



	Task ID	N	Mean Rank	Sum of Ranks
Corrugator average	C Task 1	4	5.25	21.00
	Python Task 1	4	3.75	15.00
	Total	8		

Figure 6. Mean ranks of average corrugator supercilii activity on first task.

Figure 6 indicates higher corrugator supercilii activity on test participants at C tasks, which indicates active frowning activity on test participants facial muscular activity during the first C programming task. This shows that test participants may experience more negative related emotions on C language tasks than on Python language tasks.



Test Statistics[a]

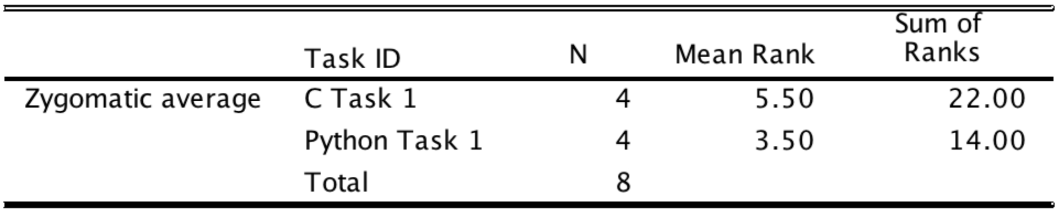
	Corrugator average
Mann-Whitney U	5.000
Wilcoxon W	15.000
Z	-.866
Asymp. Sig. (2-tailed)	.386
Exact Sig. [2*(1-tailed Sig.)]	.486[b]

a. Grouping Variable: Task ID

b. Not corrected for ties.

Figure 7. Significance test on corrugator supercilii activity on first task.

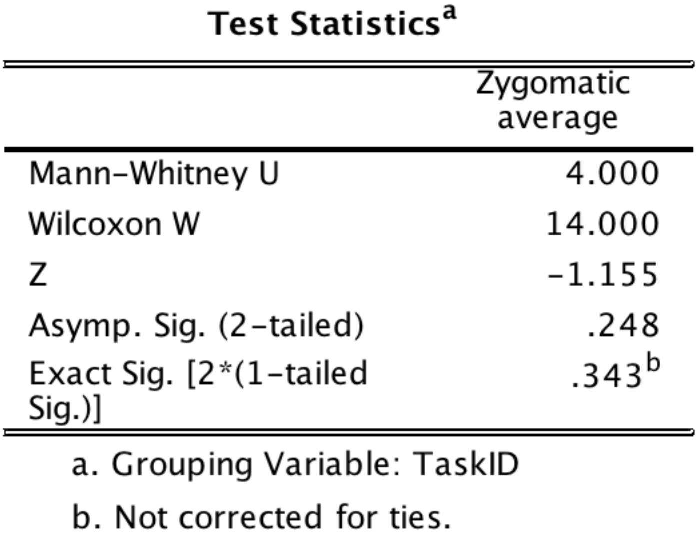
The corrugator supercilii's activity on first task is not statistically significantly higher on either C language or Python language programming tasks (*U* = 5, *p* = 0,386) as show on Figure 7. Therefore, null hypothesis on first tasks cannot be rejected. On an overall level, corrugator supercilii activity does not differ when programming with either C or Python, but in this experiment, more corrugator supercilii activity was recorded with C language. Therefore, results of this task indicate that test participants experienced more negative related emotions when programming with C than Python, but the results are not generalizable.



	Task ID	N	Mean Rank	Sum of Ranks
Zygomatic average	C Task 1	4	5.50	22.00
	Python Task 1	4	3.50	14.00
	Total	8		

Figure 8. Mean ranks of average zygomaticus activity on first task.

Figure 8 shows that C programming language activated also zygomaticus maximus more intensively on first task than with Python language. This could be presumed that test participants experienced more valence when programming with C than with Python.



Test Statistics[a]

	Zygomatic average
Mann-Whitney U	4.000
Wilcoxon W	14.000
Z	-1.155
Asymp. Sig. (2-tailed)	.248
Exact Sig. [2*(1-tailed Sig.)]	.343[b]

a. Grouping Variable: TaskID

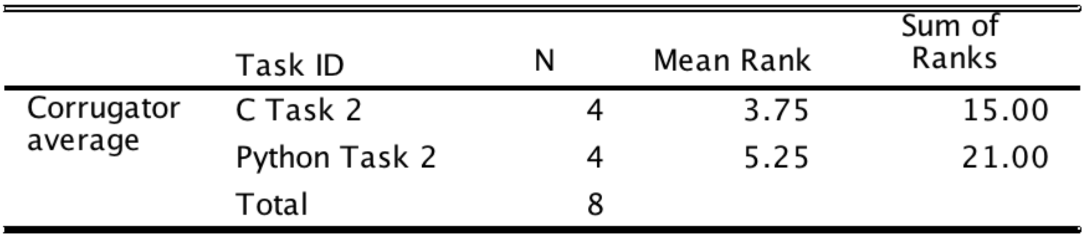
b. Not corrected for ties.

Figure 9. Significance test on zygomaticus activity on first task.

The zygomaticus major's activity is not statistically significantly higher on either C language or Python language programming task (*U* = 4, *p* = 0,248), which is shown on Figure 9. This shows that test results were not statistically significant. Therefore, null hypothesis cannot be rejected.

The first experiment indicated that C programming invoked more facial muscle activity on both corrugator supercilii and zygomaticus major, this indicates that the test participants produced stronger facial reactions when programming with C language than when programming with Python language. Null hypothesis was that either of programming languages would not produce stronger facial muscular activity than other and was not rejected. Therefore, it cannot be said that either of those languages produces strictly stronger facial muscular activity than other. This task was considered to be easiest task and was completed by every test participant on each language.

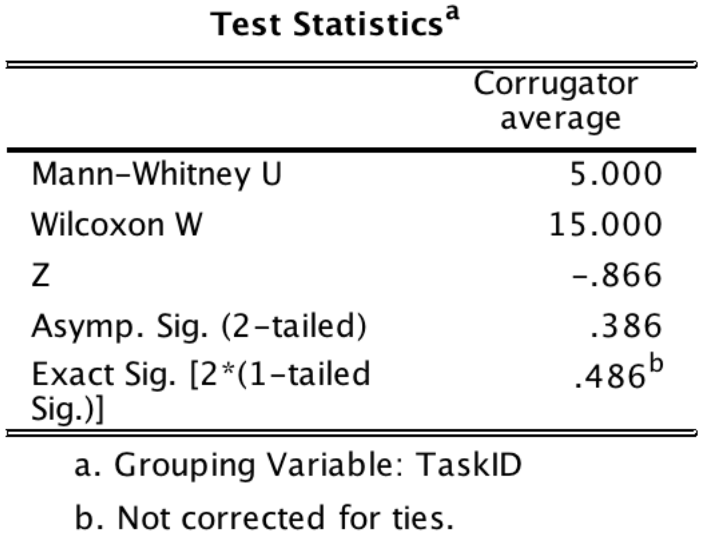
Second programming task



	Task ID	N	Mean Rank	Sum of Ranks
Corrugator average	C Task 2	4	3.75	15.00
	Python Task 2	4	5.25	21.00
	Total	8		

Figure 10. Mean ranks of average corrugator supercilii activity on second task.

Figure 10 shows that the test participants had higher corrugator supercilii activity when performing Python tasks, which indicates active frowning activity on test participants facial muscular activity during the experience. Therefore, this could indicate that test participants experienced more negative related emotions on Python language version of this task than on C language version of this task.



Test Statistics[a]

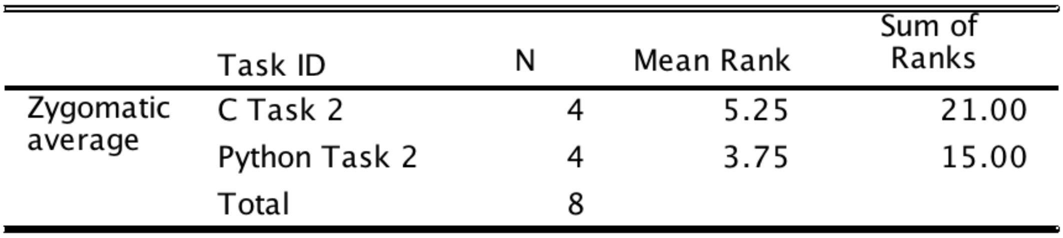
	Corrugator average
Mann-Whitney U	5.000
Wilcoxon W	15.000
Z	-.866
Asymp. Sig. (2-tailed)	.386
Exact Sig. [2*(1-tailed Sig.)]	.486[b]

a. Grouping Variable: TaskID

b. Not corrected for ties.

Figure 11. Significance test on corrugator supercilii activity on second task.

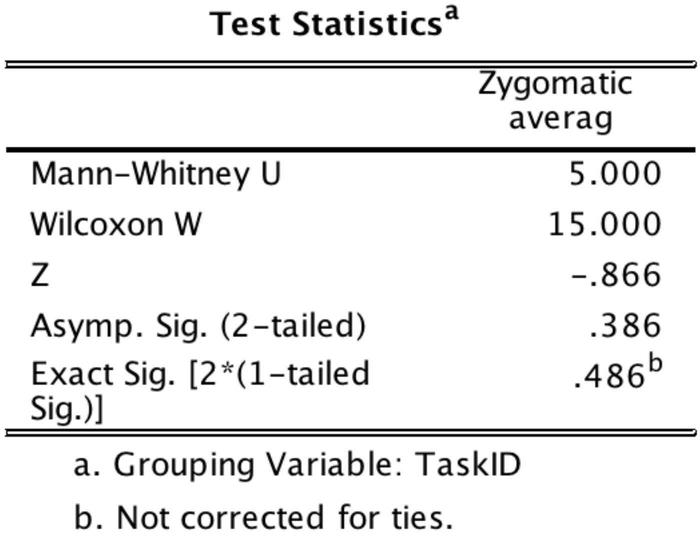
Results of the corrugator supercilii's activity on second programming task is shown on Figure 11. Results of this task indicates that corrugator supercilii's activity is not statistically significantly higher on C language programming tasks than on Python language tasks (*U* = 5, *p* = 0,386). Therefore, the null hypothesis cannot be rejected on second programming tasks.



	Task ID	N	Mean Rank	Sum of Ranks
Zygomatic average	C Task 2	4	5.25	21.00
	Python Task 2	4	3.75	15.00
	Total	8		

Figure 12. Mean ranks of average Zygomaticus activity on second task.

Figure 12 shows that test participants had higher zygomaticus maximus activity on second task when programming with C language. more This implies that test participants experienced more valence related emotions when programming with C than with Python.



Test Statistics[a]

	Zygomatic averag
Mann-Whitney U	5.000
Wilcoxon W	15.000
Z	-.866
Asymp. Sig. (2-tailed)	.386
Exact Sig. [2*(1-tailed Sig.)]	.486[b]

a. Grouping Variable: TaskID

b. Not corrected for ties.

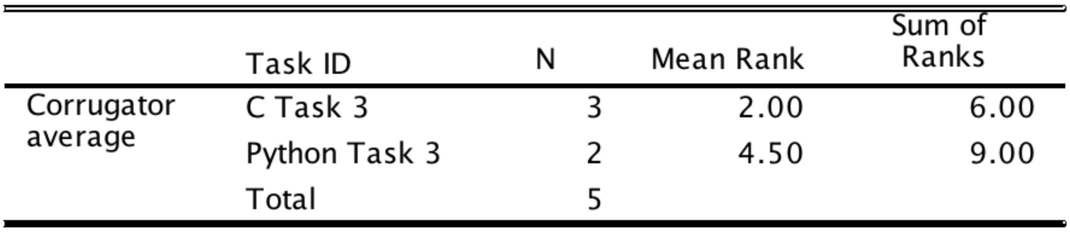
Figure 13. Significance test on Zygomaticus activity on second task.

The zygomaticus major's activity on second task is not statistically significantly higher on C language programming tasks than on Python language tasks (*U* = 5, *p* = 0,368) as indicated on Figure 13. Therefore, the null hypothesis cannot be rejected.

The second experiment indicated that C programming invoked more intense facial muscular activity than programming with Python. This would indicate that C language would invoke more muscular activity that is associated with positive and negative emotions. Testing of the results eventually showed that the null hypothesis cannot be rejected, therefore statistically significant results were not gained from this task.

Interestingly, all test participants spent more time testing their code implementation longer than on first task. Test participants conducted simple software testing by trying input different numbers to required fields of the code, whether they would produce correct answer on comparison to incorrect inputs. Nature of this task could have invoked test participants interest to test their code. Test participants tested their code more on the first language than on second language. This could be probably explained by familiarity of the task from the first iteration of the tasks.

Third programming task

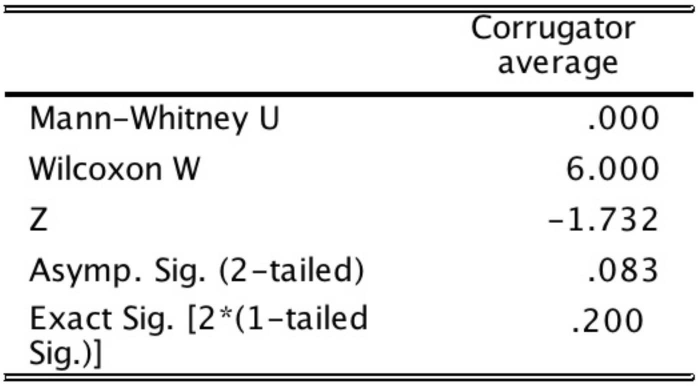


	Task ID	N	Mean Rank	Sum of Ranks
Corrugator average	C Task 3	3	2.00	6.00
	Python Task 3	2	4.50	9.00
	Total	5		

Figure 14. Mean ranks of average corrugator supercilii activity on third task.

Figure 14 shows that test participants count is not even on this experiment. Only three of four test participants had time left to attend C programming task and only two of four test participants had time left to attend Python programming task during the whole experiment. On this task, corrugator supercilii's activity was recorded to be higher on Python programming language tasks than on C programming language tasks.

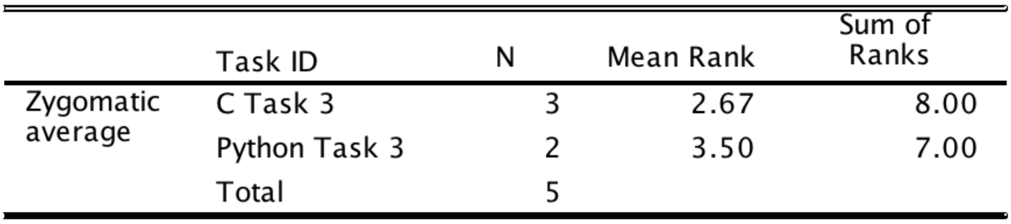
This could be difficulty of Python task, which was set to rather difficult, if the programmer were not familiar with use of functions. Python task's structure was vaguer than C programming task. Results of this task implies that test participants experienced more negative related emotions when programming with Python than with C. Only one test participant was able to accomplish this task with Python, other participants did not have enough time to accomplish this task during the experiment. C task was not completed by any of the test participant during the experiments.



	Corrugator average
Mann-Whitney U	.000
Wilcoxon W	6.000
Z	-1.732
Asymp. Sig. (2-tailed)	.083
Exact Sig. [2*(1-tailed Sig.)]	.200

Figure 15. Significance test on corrugator supercilii activity on third task.

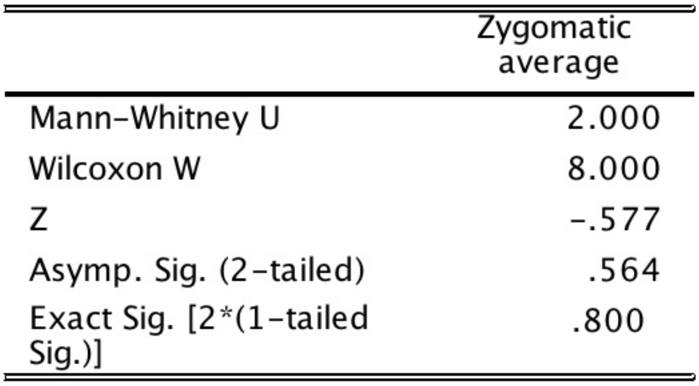
Corrugator supercilii's activity on third programming task is not statistically significantly higher on C language programming tasks than on Python language tasks (U = 0, p = 0,083) as shown on Figure 15. Therefore, null hypothesis cannot be rejected.



	Task ID	N	Mean Rank	Sum of Ranks
Zygomatic average	C Task 3	3	2.67	8.00
	Python Task 3	2	3.50	7.00
	Total	5		

Figure 16. Mean ranks of average zygomaticus activity on third task.

Figure 16 shows that test participants had higher activity on zygomaticus major on Python language than on C language. Level of mean does not differ greatly, since Python tasks mean is 3.50 and C language mean is 2.67, compared to corrugator supercilii activity, which is on Python language 4.50 and on C language 2.00. It should be noted that number of participants differ on this experiment, which has effect on mean values. This is interesting, since test participants experienced higher corrugator supercilii activity and zygomaticus major activity on Python language rather than activating only other, similarly to earlier two tasks.



	Zygomatic average
Mann-Whitney U	2.000
Wilcoxon W	8.000
Z	-.577
Asymp. Sig. (2-tailed)	.564
Exact Sig. [2*(1-tailed Sig.)]	.800

Figure 17. Significance test on zygomaticus activity on third task.

Figure 17 shows zygomaticus major's activity on third task and indicates that it is not statistically significantly higher on C language programming tasks than on Python language tasks (*U* = 2, *p* = 0,564). Therefore, the null hypothesis cannot be rejected on third tasks.

Overall of tasks

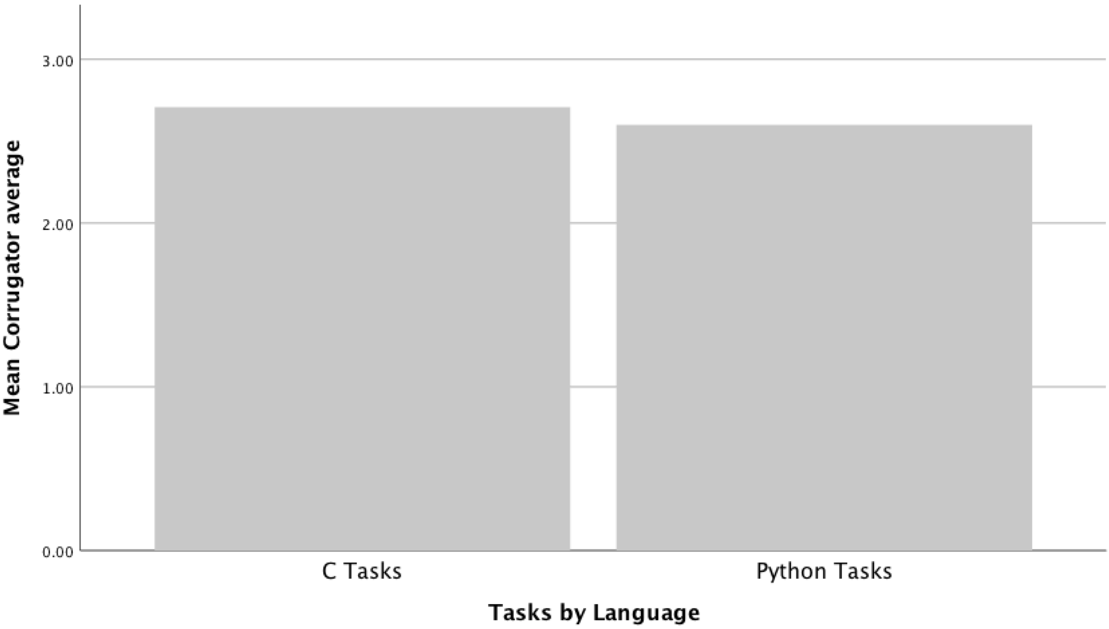
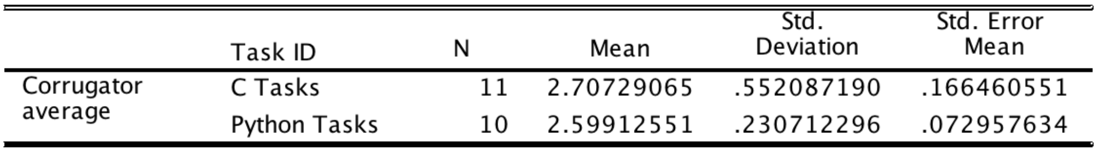


Figure 18. Graph of corrugator supercilii's activity means on all tasks.

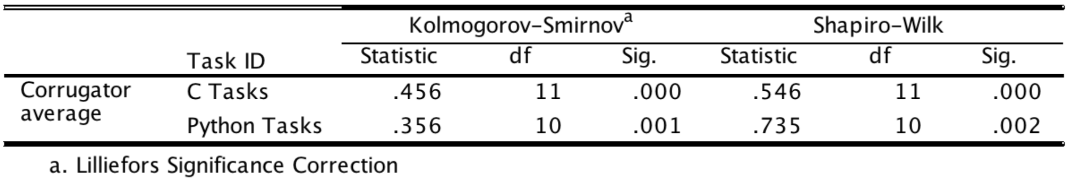
On Figure 18, mean corrugator average of corrugator supercilii's is presented on graph format. This Figure 18 shows that there is low variance between C programming language and Python programming language tasks. This indicates that on an overall level, there were only slight difference between these two languages when recorded corrugator supercilii activity.



	Task ID	N	Mean	Std. Deviation	Std. Error Mean
Corrugator average	C Tasks	11	2.70729065	.552087190	.166460551
	Python Tasks	10	2.59912551	.230712296	.072957634

Figure 19. Mean average of corrugator supercilii's activity on all tasks.

On more detailed level, Figure 19 shows that C language tasks have slightly higher level of corrugator supercilii's activity (*M* = 2.70) than Python language tasks (*M* = 2.59). Figure 19 shows also that corrugator supercilii activity's standard deviation was larger on C language tasks than on Python language tasks (*SD* = 0.55; *SD* = 0.23). This shows that on C language tasks variance differentiated more from mean values than Python languages.

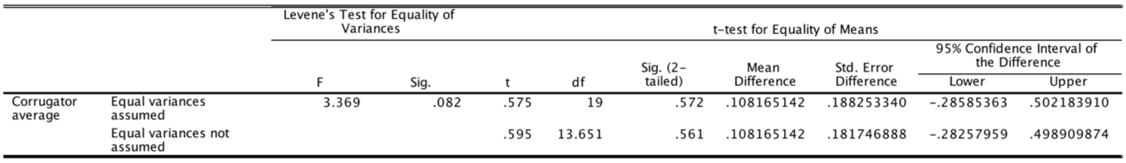


		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Task ID	Statistic	df	Sig.	Statistic	df	Sig.
Corrugator average	C Tasks	.456	11	.000	.546	11	.000
	Python Tasks	.356	10	.001	.735	10	.002

a. Lilliefors Significance Correction

Figure 20. Normality of corrugator supercilii activity.

On Figure 20, normality of corrugator supercilii's activity on C language and Python language tasks is shown. In this case, Shapiro-Wilk's test is used since the sample size is smaller than 2000 participants. Normality test shows that on an overall level C language tasks (*p* = .000) and Python language tasks (*p* = .002) are normally distributed.



		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Corrugator average	Equal variances assumed	3.369	.082	.575	19	.572	.108165142	.188253340	-.28585363	.502183910
	Equal variances not assumed			.595	13.651	.561	.108165142	.181746888	-.28257959	.498909874

Figure 21. Equality of variances on corrugator supercilii activity.

Levene's test on Figure 21 shows that corrugator supercilii's activity's variances are not equal between C programming language and Python programming languages. Null hypothesis cannot be rejected since *p* value is greater than 0.05 (*p* = .082). This means, that on an overall level, corrugator supercilia activity is not higher on either C or Python programming language than on other.

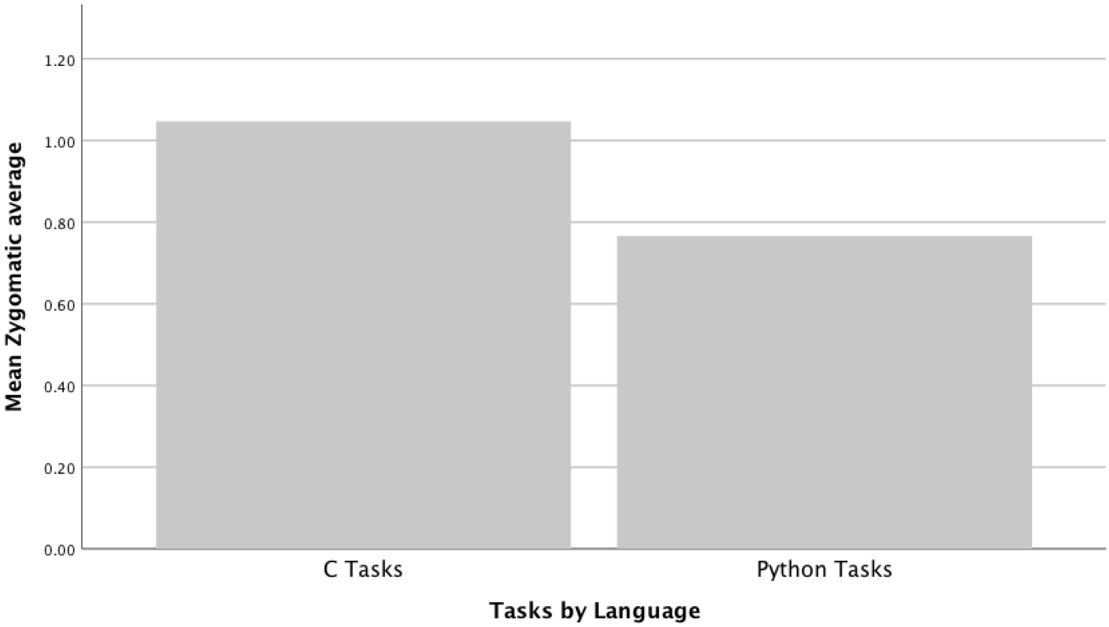
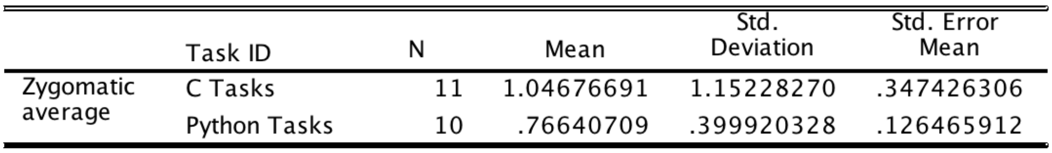


Figure 22. Graph of zygomaticus major's activity means on all tasks.

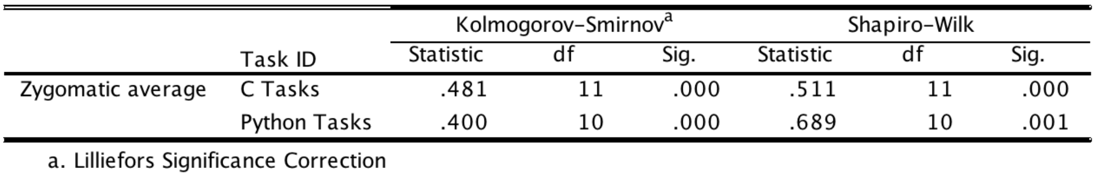
Figure 22 presents graph of means of zygomaticus major's activity on every task. This Figure 22 shows that there is variance between C programming language and Python programming language tasks on zygomaticus major activity on sum of all test cases. This indicates that on an overall level, there is some difference between C and Python when measuring zygomaticus activity. It is noticeable that recorded mean values are lower on zygomaticus activity than on corrugator supercilii activity.



	Task ID	N	Mean	Std. Deviation	Std. Error Mean
Zygomatic average	C Tasks	11	1.04676691	1.15228270	.347426306
	Python Tasks	10	.76640709	.399920328	.126465912

Figure 23. Mean average of zygomaticus major's activity on all tasks.

Evaluating zygomaticus major's means in more detailed level, the Figure 23 shows that C language tasks have a higher level of zygomaticus major activity (*M* = 1.04) than Python language tasks (M = 0.76). Figure 23 shows also that corrugator supercilii activity's standard deviation was larger on C language tasks than on Python language tasks (*SD* = 1.15; *SD* = 0.39). This means that on C language tasks, variance from average mean value differentiated more than on Python values, which in this case were more consistent.

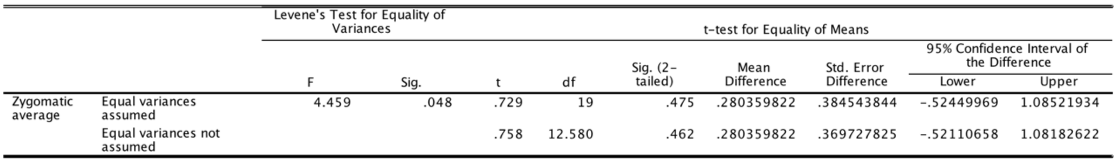


		Kolmogorov-Smirnov^a			Shapiro-Wilk		
	Task ID	Statistic	df	Sig.	Statistic	df	Sig.
Zygomatic average	C Tasks	.481	11	.000	.511	11	.000
	Python Tasks	.400	10	.000	.689	10	.001

a. Lilliefors Significance Correction

Figure 24. Normality of corrugator supercilii activity.

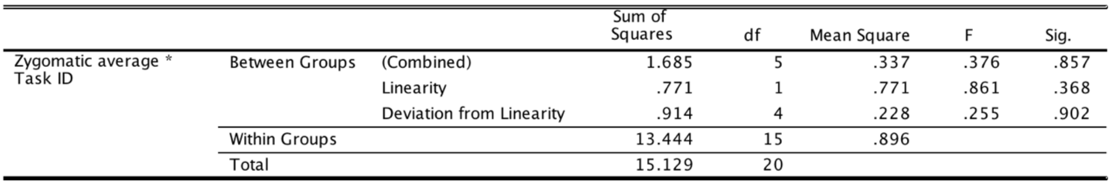
On Figure 24, normality of zygomaticus major activity on C language and Python language tasks is shown. Shapiro-Wilk's test is used since the sample size is smaller than 2000 participant count and same test is used on equivalent test of corrugator supercilii's activity. Normality test shows that on an overall level C language tasks (*p* = .000) and Python language tasks (*p* = .001) are normally distributed.



		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Zygomatic average	Equal variances assumed	4.459	.048	.729	19	.475	.280359822	.384543844	-.52449969	1.08521934
	Equal variances not assumed			.758	12.580	.462	.280359822	.369727825	-.52110658	1.08182622

Figure 25. Equality of variances on Zygomaticus major activity.

Variance's test of homogeneity on Figure 25 shows that zygomaticus major activity's variances are equal between C programming language and Python programming languages. Null hypothesis can be rejected since *p* = .048 (*p* < .05). This shows that on an overall level, zygomaticus major activity is not varying on different programming languages based on these findings. This finding is statistically significant. Since variances were small between two groups in zygomaticus major activity, the variances should be tested with one-way ANOVA.



			Sum of Squares	df	Mean Square	F	Sig.
Zygomatic average * Task ID	Between Groups	(Combined)	1.685	5	.337	.376	.857
		Linearity	.771	1	.771	.861	.368
		Deviation from Linearity	.914	4	.228	.255	.902
	Within Groups		13.444	15	.896		
	Total		15.129	20			

Figure 26. ANOVA zygomaticus major activity on all tasks.

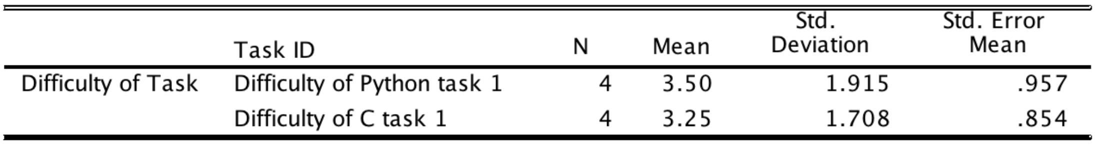
On Figure 26, the variance in zygomaticus major activity between C language and Python language is tested. The results show that there is small variation between means of groups. Results shows after all that null hypothesis cannot be rejected since *p* = 0.857 (*p* > 0.05). In conclusion, variance between groups is not statistically significant and it cannot be implied that there is significant difference in zygomaticus major activity between C and Python languages.

6.2 Post-experiment questionnaire

In supplement of EMG data, the post-test questionnaire is used to gain additional knowledge of the mindset of the test participants. Post-test questionnaire aims to gain knowledge regarding test participants experiences of the actual testing. Post-test questionnaire contains questions about difficulty of tasks, difficulty of programming language syntax and pleasantness of programming language.

On post-test questionnaire with difficulty and syntax of programming tasks (Appendix B), the answers to questions are categorized as from very challenging to very easy. Categories were encoded into numeral from where “very challenging” was given value of 1 and “very easy” was given value of 5.

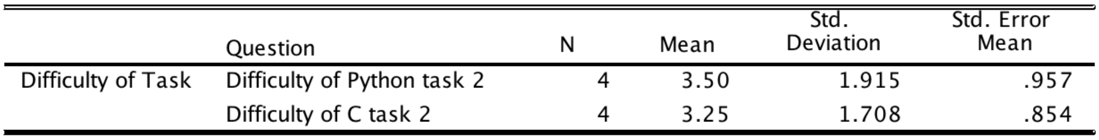
On the following figures, experienced difficulty of tasks is analyzed in detail. Means of different programming language groups are compared in aspect of difficulty of the tasks. Task difficulty is assessed as a whole after individual comparison. After comparison of means, the validity of results is tested.



	Task ID	N	Mean	Std. Deviation	Std. Error Mean
Difficulty of Task	Difficulty of Python task 1	4	3.50	1.915	.957
	Difficulty of C task 1	4	3.25	1.708	.854

Figure 28. Difficulty of the first task.

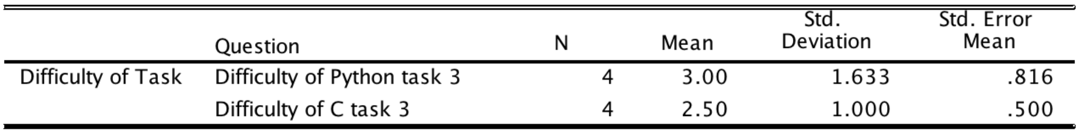
First task was experienced with Python language was experienced to be slightly easier than with C language as shown on Figure 28. Test participants answered on questionnaire that mean of Python is closer to 5, which indicates very easy task (*M* = 3.50; *M* = 3.25).



	Question	N	Mean	Std. Deviation	Std. Error Mean
Difficulty of Task	Difficulty of Python task 2	4	3.50	1.915	.957
	Difficulty of C task 2	4	3.25	1.708	.854

Figure 29. Difficulty of the second task.

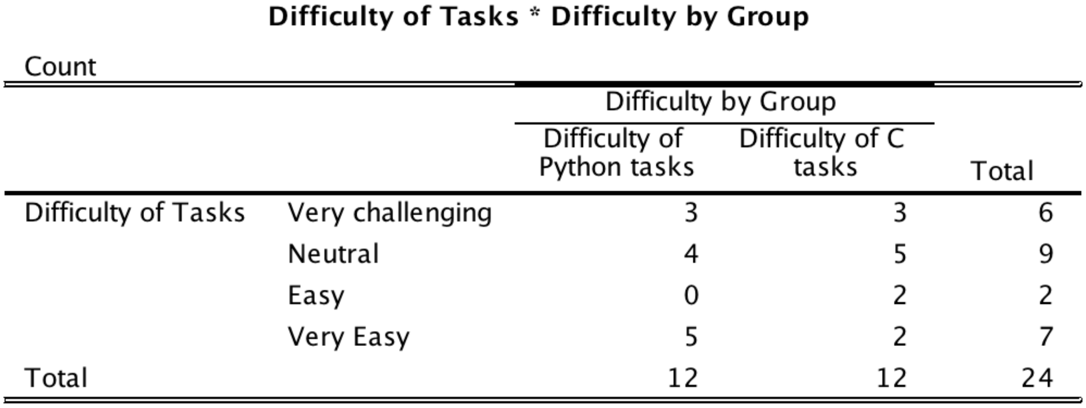
Figure 29 presents second tasks experienced difficulty of programming languages according to post-test questionnaire. Second task was also experienced to be slightly easier with Python language than with C language (*M* = 3.50; *M* = 3.25).



	Question	N	Mean	Std. Deviation	Std. Error Mean
Difficulty of Task	Difficulty of Python task 3	4	3.00	1.633	.816
	Difficulty of C task 3	4	2.50	1.000	.500

Figure 30. Difficulty of the third task.

Third task was experienced to be easier with Python language than with C language (*M* = 3.00; *M* = 2.50). On this task, Python language's mean was 0.5 points smaller than on first and second tasks results, and C language's mean was 0.25 points higher than on first and second task as show on Figure 30 on above. Interestingly, every test participant answered to this question, even though all of them did not complete this task on EMG tests and questionnaire did not explicitly inform the user to fill every question of post questionnaire.



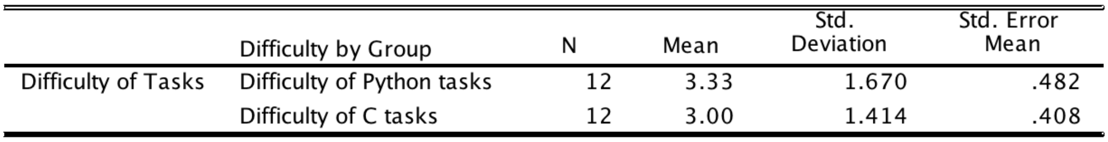
Difficulty of Tasks * Difficulty by Group

Count

		Difficulty by Group		
		Difficulty of Python tasks	Difficulty of C tasks	Total
Difficulty of Tasks	Very challenging	3	3	6
	Neutral	4	5	9
	Easy	0	2	2
	Very Easy	5	2	7
Total		12	12	24

Figure 31. Difficulty of programming language.

Figure 31 presents summary of how difficulty of programming tasks was experienced by language. In this Figure, the individual questions of difficulties are summed into total and are used in further tests. The post-test questionnaire (Appendix B) had question regarding overall difficulty of tasks but summarizing individual questions' results provides more consistent view on tasks' difficulty. More precise overall difficulty could be achieved in this approach, since the test participants have first assessed the tasks on an individual level and then the results are summed together. Interestingly, no one answered that programming tasks were "challenging". Only presented four choices were answered by the test participants.

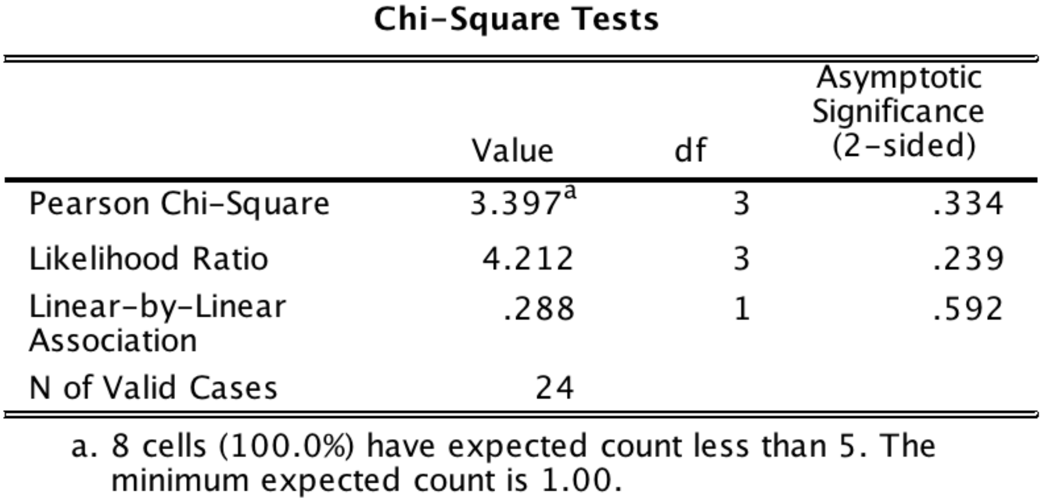


	Difficulty by Group	N	Mean	Std. Deviation	Std. Error Mean
Difficulty of Tasks	Difficulty of Python tasks	12	3.33	1.670	.482
	Difficulty of C tasks	12	3.00	1.414	.408

Figure 32. Difficulty of tasks.

On Figures 31 and 32, the number of cases (N) is 12, since there are two languages compared and N of the experiment is total of four test participants. Each task has its own question with is multiplied with two, which is number of languages that were studied and then multiplied with four, which is number of test participants answering to this post-test questionnaire.

On an overall level, Python tasks were experienced to be slightly easier on in contrast of the C language tasks (*M* = 3.33; *M* = 3.00) as shown on Figure 32. Difference between means of each task combined seems to be close, differing only 0.33 points. Standard deviation differs also between groups, results of Python tasks had generally larger deviation than C tasks (*SD* = 1.670; *SD* = 1.414).



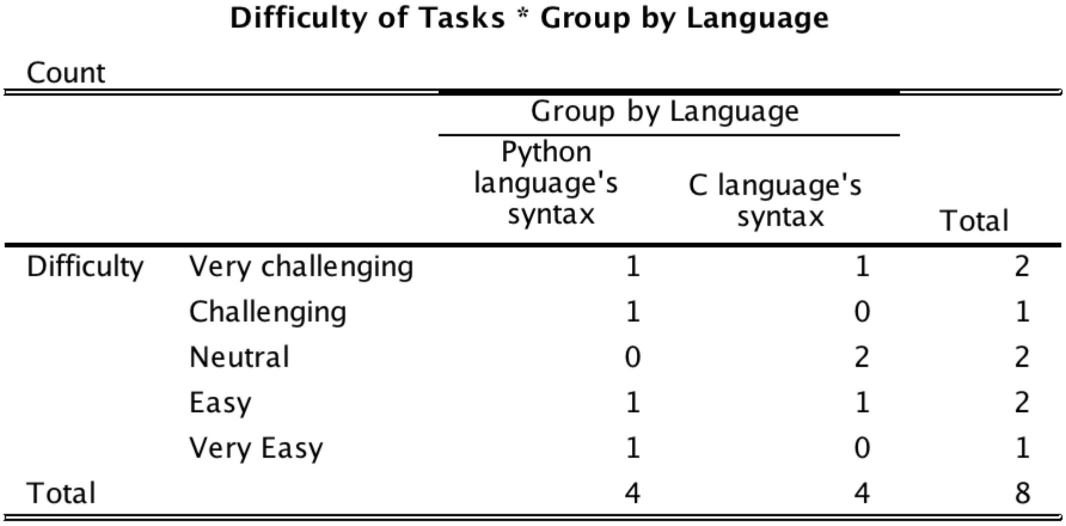
Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	3.397[a]	3	.334
Likelihood Ratio	4.212	3	.239
Linear-by-Linear Association	.288	1	.592
N of Valid Cases	24		

a. 8 cells (100.0%) have expected count less than 5. The minimum expected count is 1.00.

Figure 33. Chi-Square tests of language difficulty.

The relationship between programming language and task difficulty cannot be verified by statistical methods when analyzing results of this questionnaire. On Figure 33, the N is 24 since count of all answers are summed together. Relationship between experienced difficulty and programming language is not statistically significant p = .334 ($p > 0.05$).



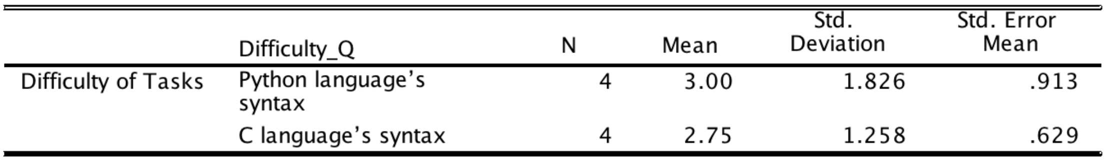
Difficulty of Tasks * Group by Language

Count

		Group by Language		
		Python language's syntax	C language's syntax	Total
Difficulty	Very challenging	1	1	2
	Challenging	1	0	1
	Neutral	0	2	2
	Easy	1	1	2
	Very Easy	1	0	1
Total		4	4	8

Figure 34. Difficulty of programming language syntax.

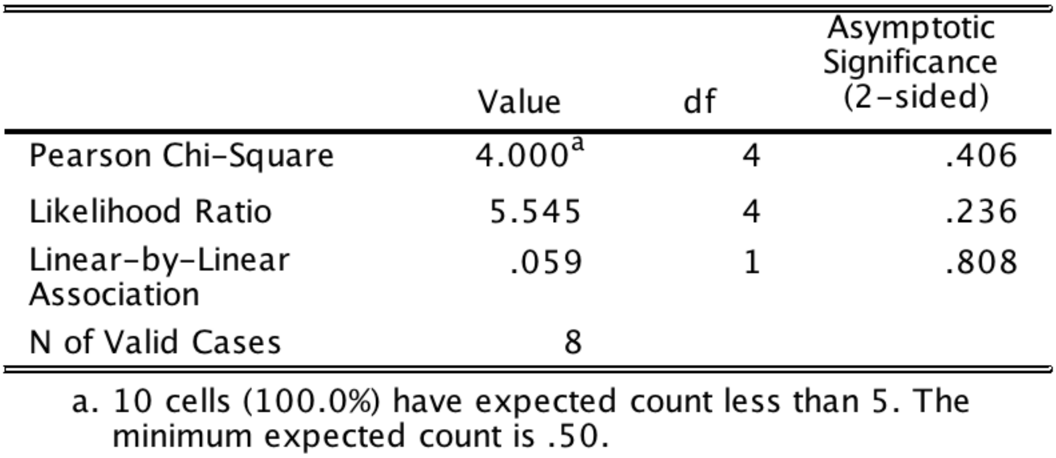
Test participants division of answers to question regarding experienced understanding of syntax programming languages syntax on Figure 34.



	Difficulty_Q	N	Mean	Std. Deviation	Std. Error Mean
Difficulty of Tasks	Python language's syntax	4	3.00	1.826	.913
	C language's syntax	4	2.75	1.258	.629

Figure 35. Means of programming language syntax.

On an overall level, Python languages syntax were experienced to be slightly understandable than C language syntax (*M* = 3.00; *M* = 2.75) as shown on Figure 35. Difference between means of each task combined seems to be close, differing only 0.25 points. Standard deviation differs also between language groups, results of Python language syntax had generally wider deviation than C language syntax (*SD* = 1.82; *SD* = 1.25).



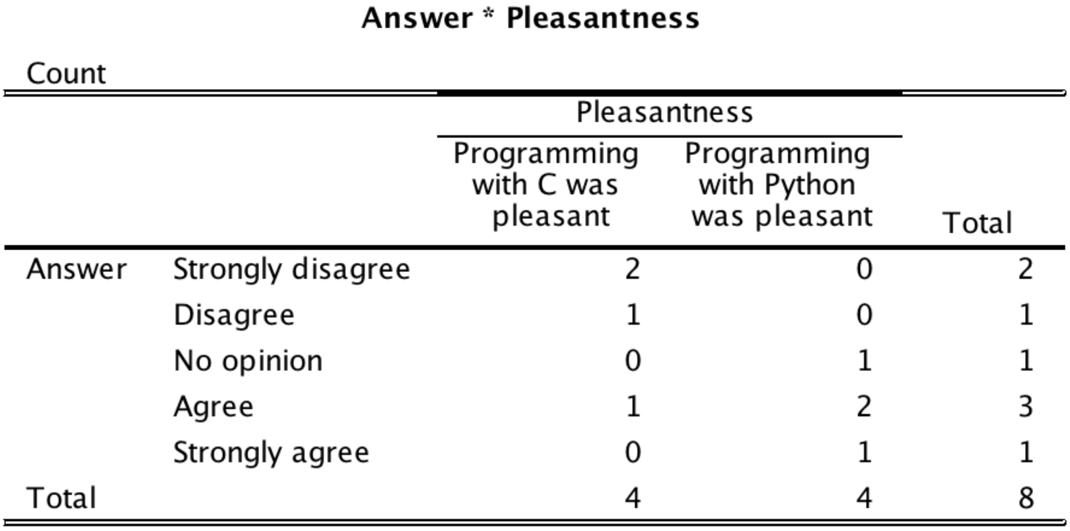
	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	4.000[a]	4	.406
Likelihood Ratio	5.545	4	.236
Linear-by-Linear Association	.059	1	.808
N of Valid Cases	8		

a. 10 cells (100.0%) have expected count less than 5. The minimum expected count is .50.

Figure 36. Chi-Square tests of syntax difference.

The relationship between programming language syntax and task difficulty cannot be verified by statistical methods when analyzing results of this questionnaire. Relationship between experienced difficulty and programming language is not statistically significant *p* = .406 (*p* > 0.05) as shown on Figure 36.

On post-test questionnaire with pleasantness of programming languages (Appendix B), the answers to questions are categorized as from strongly disagree to strongly agree. Strongly disagree was given value of 1 and strongly agree was given value of 5.



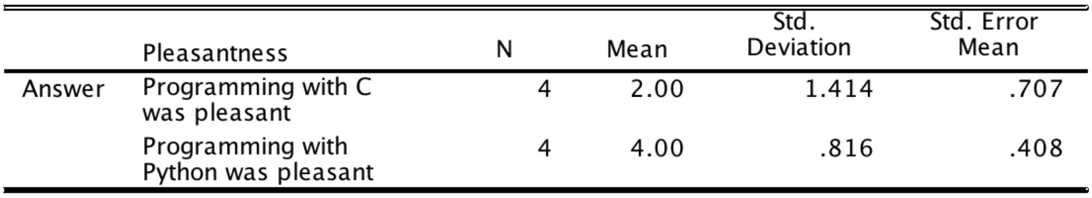
Answer * Pleasantness

Count

		Pleasantness		
		Programming with C was pleasant	Programming with Python was pleasant	Total
Answer	Strongly disagree	2	0	2
	Disagree	1	0	1
	No opinion	0	1	1
	Agree	1	2	3
	Strongly agree	0	1	1
Total		4	4	8

Figure 37. Pleasantness of programming language.

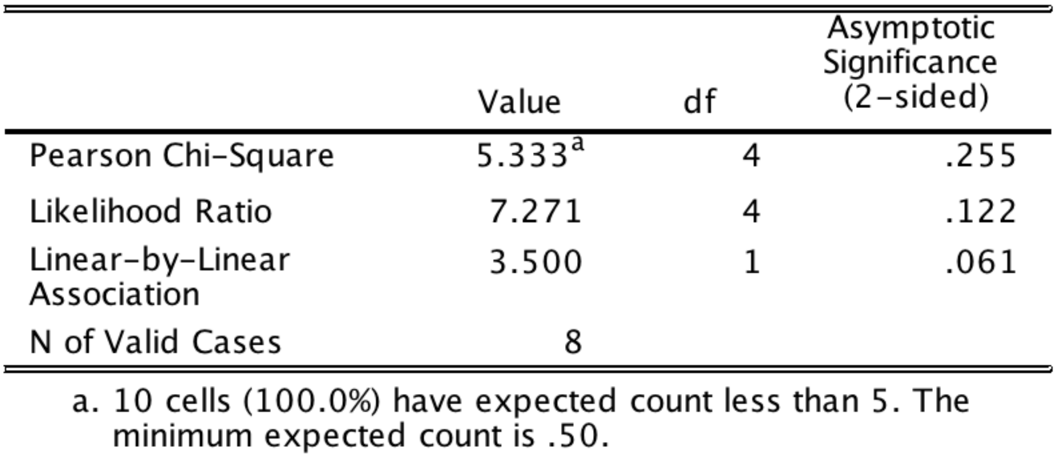
Test participants division of answers to question regarding experienced pleasantness of programming language on Figure 37.



	Pleasantness	N	Mean	Std. Deviation	Std. Error Mean
Answer	Programming with C was pleasant	4	2.00	1.414	.707
	Programming with Python was pleasant	4	4.00	.816	.408

Figure 38. Pleasantness of programming language means.

Figure 38 shows means of answers on pleasantness of programming languages. Results show that Python was experienced to more pleasant than C (*M* = 4; *M* = 2). Standard deviation on is higher on C tasks than on Python tasks answers (*SD* = 1.41; *SD* = 0.81). This indicates that experiences with Python were more consistent.



	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	5.333[a]	4	.255
Likelihood Ratio	7.271	4	.122
Linear-by-Linear Association	3.500	1	.061
N of Valid Cases	8		

a. 10 cells (100.0%) have expected count less than 5. The minimum expected count is .50.

Figure 39. Chi-square tests of pleasantness.

The relationship between programming language and pleasantness cannot be verified by statistical methods when analyzing results of this questionnaire. Relationship between experienced pleasantness and programming language is not statistically significant p = .255 (p > 0.05) as shown on Figure 39.

These presented results are discussed further in the next Chapter. Results are compared to previous studies and what kind of key points is perceived from these results. Furthermore, some issues with the research are pointed out on the next Chapter.

7. Discussion

In this Chapter, the results of this study that were presented in previous Chapter are summarized and discussed further. Results of this study are compared to previous studies that were presented earlier in this study as continuity of theoretical background. Finally, found issues with this experiment are presented at the end of this Chapter.

7.1 EMG results

The individual tests on each task showed how mean values of corrugator supercilii's and zygomaticus major's activity differed between programming languages. Individual tests were combined to see the language differences altogether. Combined test results were analyzed further for providing more concrete answer to research hypotheses. It should be noted that since means of results are analyzed, the results are implying generalization of selected facial muscle's activity on specific programming languages.

On individual level of tasks, the corrugator supercilii's activity on the first task was higher on C language version of the task than on Python language version. Zygomaticus major's activity was also recorded to be higher on C language task than on Python language task. This would indicate that C language invoked more positive and negative related emotions among the test participants in general. In contrast of null hypothesis, C language would produce more activity on corrugator supercilii with C language was presented, but results had to be rejected, since results were not statistically significant. Zygomaticus major's activity was shown to be higher on C language but was not supported by the results. Therefore, null hypothesis was not rejected on zygomaticus major activity.

On second task, corrugator supercilii's activity was higher on Python language and zygomaticus activity was higher on C language. Results indicated that the test participants experienced more negative related emotions on Python language task and more positive related emotions on C language tasks on a general level. This showed that results supported H_1 and H_2, but both were rejected, since the results were not statistically significant.

On third task, count of test participants differed, since three of four had time to start the task on C language and only two of four participants had time to start the task on Python language. Only one participant completed the third task and with both languages. Both corrugator supercilii's and zygomaticus major's activity on third task was higher on Python language tasks than on C language tasks. This indicated that Python language task invoked more intense facial muscular activity than C language task. This also would suggest that test participants experienced more positive and negative associated emotions when programming with Python on this task on a general level. Both results were rejected, since the results were not statistically significant.

The results did not support the null hypothesis of C language's production of higher corrugator supercilii's activity. Instead, the results supported the null hypothesis, that Python language invokes higher zygomaticus major activity. Both results were rejected, since the results were not statistically significant.

The results showed individual differences on different kind of tasks. Results could have supported some of research hypothesis, but were rejected, since they were not statistically significant. This means that more test cases should be implemented in order to gain more results and to have a change to produce statistically significant results. Results were combined as overall results in order to test the research questions.

On overall results, the corrugator supercilii activity had little variance between C and Python language tasks. When tested, the results showed that the results were not statistically significant. The overall results showed that there was more variance on means of zygomaticus major's activity between C and Python. Results indicated that there was statistically significant variance between groups in zygomaticus major’s activity. Tests of the results showed that there was small variation between the groups variance, but the results were not statistically significant. Therefore, it cannot be said if C language or Python language invokes more intense zygomaticus major activity on facial muscles in general. As for conclusion of the EMG results, more test should have been conducted to gain statistically significant results and to see possible differences between languages more clearly, since there already was slight indication of significance on group-based comparison of the results.

7.2 Post-test questionnaire

On Figures 31 and 32 on page 57, N is 12, since there are two languages compared and N of the experiment is four. Each task has its own question with is multiplied with two, which is number of languages that were studied and then multiplied with four, which is number of test participants answering to this post-test questionnaire. On Figure 33, the N is 24 since count of all answers are summed together.

On an overall level, difficulty of Python tasks was experienced to be slightly harder than C tasks. Relation between difficulty of task language could not be verified, since the results were not statistically significant. Python language's syntax was also experienced to be slightly more difficult than C language's syntax. No conclusion of this results can be implied, since the results were not statistically significant.

Pleasantness of the programming languages gave controversial results, since the indicated that programming with Python was more pleasant than programming with C. Again, more tests should be conducted in order to gain more reliable results, since the results of this question were not statistically significant. All questionnaire results and most of EMG test results indicated shortage of test participants in order to gain more reliable and statistically significant results.

7.3 Comparison of results

In this study, psychophysiological activity in programming tasks was measured with EMG device, but no statistically significant results were achieved. Neither of C or Python programming languages indicated difference in positive or negative emotions activation during the programming tasks. Additional questionnaire provided some insight regarding test participants opinions of the test tasks. This study did not observe closely test participants reading activity or information seeking behavior during the tasks, which could be one viewpoint for further studies.

In comparison of Rajendra Desai's (2017) master's thesis' results, where test participants experienced Python programming task with more positive related emotions than C programming task. As on contrary to Rajendra Desai's results, this study showed that there was higher activity on zygomaticus major during C programming tasks, which indicates higher level of positive emotions within C. Unfortunately, this study's results were not statistically significant, therefore, results could not be generalized as they were on the other study. Rajendra Desai's study used EEG-measurement which differs greatly from EMG-measurement as a technique as described in theoretical background, yet there are some aspects that can be compared between one and other. One of these as aspects are experienced positive and negative emotions during the programming tasks.

Rajendra Desai's (2017) used also a post-test questionnaire to gain additional information about test participants experienced emotions. Rajendra Desai's post-test questionnaire's results complemented the results of EEG-measurement results by supporting test participants felt emotions of programming with Python to be more positive experience than programming with C. In this study, the opposite results were achieved. In this study, test participants answered that Python was easier and more pleasant than C, even thou EMG results showed otherwise. This would have provided really interesting outcome of this study, but the results were again statistically insignificant and therefore further analyzation was not conducted.

One aspect that emerged from Rajendra Desai's post-test questionnaire was that the questions were leading towards in favor of Python. This may have resulted in way that the test participants answered also in favor of Python for being easier to understand and easier to write. On this study as an improvement to Rajendra Desai's study, the post-test questionnaire's questions were written in format that would not lead the test-participant in predetermined conclusion, but rather let test participants to answer as they felt. Post-test questionnaire's questions were formatted to be on a more generalized level.

On Müller and Fritz (2015) study, the test participant experienced higher level of frustration during the second task of the experiment. This is predictable result in experiments and similar result was expected in this study. There were some indications of higher level of corrugator supercilii and zygomaticus major activity on second and third task compared to first task. On both of these researches, first task was expected to be easy for the test participants and second task to be harder. This research had additional task that was considered to be harder than second task. This setting would suggest that there would have been noticeable change in activity as in Müller and Fritz experiment. Results of this study did not support the alternate hypotheses by not providing statistically significant results.

Additional aspect from Müller and Fritz (2015) study was that, the test participants told that when they got stuck with programming task, they would seek assistance from colleague or from Internet. Additionally, in their study, the test participants told that they switch between tasks, in order to avoid stacking negative emotions with programming.

In contrast to this study, the test participants followed each task accordingly and did not skip between tasks. Test participants completed the given tasks in given number and moved to next task only after they had completed the task or felt that they could not contribute to it anymore. Fixed timing used for task may have pressured test participants to conduct the tasks as fast as possible and therefore not to skip between tasks.

Previous studies in reading have shown that familiar words produce more valence than unfamiliar words. In this study, this method could have been utilized if there would have been enough test participants with previous skills in both Python and C programming languages. In this kind of study, familiar programming language syntax could have been used for analyzing valence in test participants reading.

Results of this study did not produce quite high level of facial muscular activity, which was seen on average values of the blocks. This could indicate that provided tasks were not visually attractive enough, and therefore might not have provide enough stimulus. Additionally, the focusing to task is harder monitor with EMG setting alone. It would be interesting, if larger set of erroneous code would produce intense facial muscular activity.

7.4 Issues with research

Questionnaire design should be considered, since post-test questionnaire (Appendix B) enables test participant to answer to the question by “no opinion”, which is placed in the middle of the answer sheet. Possibility of answering “no opinion” is valid, but its placement should be considered when designing questionnaires. In this questionnaire, it was placed on the middle, since it acted as neutral field, but it could have been placed on leftmost or rightmost edge of the questionnaire, since it does not provide clear answer from the test participant.

Questionnaire consisted of some additional questions, but the results of the questionnaire were set to handle only programming language issues. This could be remarked as limitation of the study, but these results were selected to limit the scope more into programming languages rather than on how tests were planned and how the test participants were able to conduct them.

Test participants spent different amount of time on each task, which determined whether the test participant had enough time to complete every task. This also had effects on recorded EMG scores, duration spent on one task increases number of analyzed test scores. This had some effect on means of data. Described issues of the study will be discussed in following Chapter further with suggestions to future studies.

8. Conclusion

This Chapter concludes this master’s thesis study by providing summary of the study. Summary will conclude the results and their relation to the topics of this study in contributions sections. In addition to the summary, this Chapter includes limitations of the study and suggestion for future studies of similar topics. Limitations of this study and future studies provides complementary information for this study. Main complementary comes from contributions to the practice and from the future studies section, where enhanced study methods are discussed.

8.1 Contributions to theory

This study's main focus points were in programming skills with C and Python programming languages, emotions, psychophysiology and quantitative research. The focus points are derived into emotions in programming tasks and psychophysiological research in scope of programming. This study's research questions, or hypotheses aimed to provide answers to these scopes by conducting EMG study on facial muscles. In more detail, the studied programming languages were tested to see if either of the languages would produce more significant facial muscular activity than other on a specific facial muscle. After the experiment, the results were compared to see whether there was any difference between the activity.

Additionally, the test participants answered to questionnaire after the test for gaining additional information about test participants experiences during the experiment. Results of this study indicated that there was some facial muscular activity among programming tasks that could be linked to emotional reactions, but the results were not statistically significant. This means that the results did not indicate whether one of the studied would have provided more intense facial reaction than other when conducting the programming tasks. This will be discussed further on limitations of this study.

Theoretical background and previous studies covered experiments of the professional and novice programmers’ skills and experiences. Programming skills of this study’s test participants were divided evenly to novice and to advanced level. This guaranteed that level of programming tasks used in experiments were suitable. This was also shown when the test participants struggled or succeeded from tasks. Main succession occurred on first task, but the second task produced more struggle to complete the tasks among the test participants.

8.2 Contributions to research practices

This study’s secondary goal was fulfilled by providing guidelines for future EMG studies in setting of University of Oulu’s UX-Lab. This study discussed widely on theoretical background for psychophysiological studies in scope of Information processing science. In addition to theory, practical instructions were discussed and guides for setup the UX-Lab for testing purposes.

Chapter five of this study, the research implementation Chapter, explains thoroughly the used research process and how EMG measurement research can be conducted in setting of University of Oulu's UX-Lab. Research implementation Chapter provides sort of step-by-step guide on EMG research in this research's setting, utilizing process diagrams for visual aid of the overall processes. Process diagrams were drawn to provide precise progression of the tests and EMG measurement activity.

Described processes and points of interests were assessed by reading previous studies in psychophysiological and EMG research and were designed to suit this research setting and laboratory. Getting to know the research setting and use of pilot test helped to design the research and improve the research process before the actual research was conducted. Carefully designed research process helped to solve and mitigate possible issues with conducted research that were encountered in pilot test. Pilot test and actual research results' observations are discussed further on next section.

In this study, the used length of recorded EMG data blocks were 10000 milliseconds. Suggested epoch length were 500 milliseconds, which would have produced more detailed results, since the comparison of data values in certain epochs are clearer. For future studies, it would be suggested to use default length of epochs. Additionally, better data processing tools such as MATLAB could be used when rectifying and smoothening the EMG signal data and to gain more precise results. Additionally, using of the provided PsychLab's software was not as easy as expected and it had some misleading errors that were encountered during the pilot test of this study.

In addition to usage of epoch length of 500 milliseconds, it could be also convenient to record task specific data set. This can be conducted by record only one task at time. This kind of set of recordings would make analyzation process easier, since researchers does not have to detach different task from the whole recording, but to analyze single data file at time.

8.3 Limitations of the study

Significant limitation of this study came from lack of test participants; however, the results of this study can be considered as indicative results and provides interesting points for further studies. There may have been several issues, which may have caused lacking in participants, for example scheduling of the tests. One issue of this study occurred during the recruiting of the test participants. This master's thesis indented primary aim was to study students on bachelor level of studies, mainly first- and second-year students of information processing science and computer science.

Test participants that were gained for this study were master level students in relevant fields of study. Master level students were not limited out of the study essentially but were included in case enough participants would not be achieved. Since only master level students responded to call for study, the study's focus was altered towards students and programmers on an overall level. This is was handled quite easily, since there were some preparations done beforehand when the test experiments were arranged. Main mitigation method was to give possibility for advanced students to participate to this study, in case of not enough bachelor level students would not participate, as describe on above. Another mitigation method was to add relevant background theory and previous studies regarding students programming skills and experiences.

One limitation factor in this study's test participant count could have been raised by timing period of the experiments. The experiments were conducted in end of November and beginning of December. This time-period aligns to ending of semester which may be busy time for students. On the other hand, provided schedule for the experiments were quite flexible, since timetable included hours from 8 to 16, with possibility to negotiate even later times. There could have been arranged additional test sessions in further times after holidays, but there were another external factor limiting additional studies, which came from tight in schedule of the researcher.

8.4 Future studies

There has been quite a lot of studies regarding programming languages and programmers throughout history of programming. This study provides setting for more research implementations for programming language studies and for study of psychological factors in programming. More research that could be implemented by using lessons learned from this study could provide more understanding for psychological factors behind programming as activity or psychological factors in programming languages. Interesting results could be achieved by combining different psychophysiological measurement instruments, in order to gain relevant data regarding studied phenomena.

Since C and Python covers only small part of all existing programming languages, it would be interesting aspect for future studies to test programming languages that differ more greatly in their syntax or lexicon. Additionally, further studies, in context of programming courses, the pain points of programming tasks could be studied in more detail. Previous studies could be used as premise of the study and utilize emotional measurements to identify which tasks are most difficult for students to complete. Additionally, different programming tasks could be analyzed through physiological measurements to see whether there are some unexpected points that are hard for students. Results of that study could be utilized to enhance programming tasks and benefit the learning experience of students.

One thoroughly discussed limitation of this study was lack of participants. One aspect that more participants would have enabled could be that test participants' previous knowledge of selected programming languages would have included as part of the study. Unfortunately, this study's test participants did not have enough prior knowledge to selected languages for this aspect to be taken into account in this study. Indicators between prior knowledge of the programming language and emotions could be studied in depth. It would be interesting to see if familiarity of programming languages reduces intensity of psychophysiological signals, compared to unfamiliar languages.

On replication or applicating of this study, it would be interesting to find more concrete and precise reactions, researcher should utilize additional measurement and monitoring tools among the EMG-device. One suggestion is to use device with heart rate measurement capabilities if available, for measurement heart rate of the test participant. This would give more precise results when determining stressful situations. Results of this study did not show great variation in results. As discussed, the programming activity may not be feasible to monitor solely on EMG-device, since EMG related facial muscles are heavily related on visual studies. More feasible results with EMG-device could be achieved in user experience studies.

Eye-movement tracker could be used in reading-oriented studies with EMG. Eye-tracking device enables researchers to gain information of test participants eye movement and to see where on the screen the test participants are looking at while facial muscular activity is recorded. This would help to map certain facial muscular reactions to certain parts of the code. Eye-movement's provision of heat maps of areas where the test participant has spent most of time during the experiment, in addition of facial EMG activity, would probably provide interesting results when tested for example on different kinds of functions, loops and conditional expressions on the code.

One example could be refactoring or code review task, where the test participants are only instructed to read given code samples and to find errors in them. This experience could produce more rewarding results, since facial EMG studies have previously focused more on text reading and image viewing activity. This is more suitable focus of the study as presented previously on various examples of EMG studies in the theoretical background of this study. It would be beneficial that the used code samples are large enough to produce facial muscular activity. It would be interesting to see whether number of lines in code would produce more activity than smaller samples of code or lesser number of lines in code.

New similar research would be beneficial to be conducted in order to gain feasible information on how programming activity is performed in scope of psychology. Additionally, affective computing could be added to the scope of new research, since psychophysiological research could benefit research on users and programmers. Understanding emotions in software development could increase developer’s performance and it would help in creating new working methods and paradigms in field of software development.

References

Andreassi, J. L. (2007). *Psychophysiology: Human behavior and physiological response* (5th ed.). Mahwah (NJ): Lawrence Erlbaum Associates (LEA).

Aranha, R. V., Correa, C. G., & Nunes, F. L. S. (2019). Adapting software with affective computing: A systematic review. *IEEE Transactions on Affective Computing*.

Basili, V., Caldiera, G., Lanubile, F., & Shull, F. (1996). Studies on reading techniques, In Proceedings of the Twenty-First Annual Software Engineering Workshop, SEL-96-002, pp.59-65.

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. Journal of the Learning Sciences, 23(4), 561-599.

Burden, A. (2010). How should we normalize electromyograms obtained from healthy participants? what we have learned from over 25years of research. Journal of Electromyography and Kinesiology, 20(6), 1023-1035.

Cacioppo, J. T., Petty, R. E., Losch, M. E., & Kim, H. S. (1986). Electromyographic activity over facial muscle regions can differentiate the valence and intensity of affective reactions. *Journal of Personality and Social Psychology, 50*(2), 260-268.

Camtasia (2019, May 3). Retrieved from https://www.techsmith.com/video-editor.html.

Clancy, E. A., Morin, E. L., & Merletti, R. (2002). Sampling, noise-reduction and amplitude estimation issues in surface electromyography. *Journal of Electromyography and Kinesiology, 12*(1), 1-16.

Clark-Carter, D. (2004). Quantitative psychological research : A student's handbook (2nd ed ed.). Hove: Psychology Press.

Coolican, H. (2009). *Research methods and statistics in psychology* (5th ed ed.). London: Hodder Education.

Creswell, J. W. (2014). Research design: Qualitative, quantitative, and mixed methods approaches. Thousand Oaks, CA: SAGE Publications.

Dallos, R. (2008). Observational Methods. In Breakwell, G. M., Breakwell, G. M., Rose, D., Barrett, M., Fife-Schaw, C., Davis, A., . . . O'Sullivan, D. *Research methods in psychology* (3rd ed ed.). Los Angeles: SAGE.

Davis, A. & Bremner, G. (2006). The Experimental Method in Psychology in (Breakwell et al., 2006) *Research methods in psychology* (3rd ed ed.). Los Angeles: SAGE.

Dimberg, U., Thunberg, M., & Grunedal, S. (2002). Facial reactions to emotional stimuli: Automatically controlled emotional responses. Cognition and Emotion, 16, 449–471.

Dixit, J. B. (2010). *Programming in C, third edition* (3rd ed.). New Delhi: Firewal Media.

Ekman P. (1999). Basic Emotions. In Abramson, L. Y., Averill, J. R., Bekerian, D. A., Bentall, R. P., Berkowitz, L., Bradley, B. P., . . . Öhman, A. (Eds.), *Handbook of cognition and emotion*. Chichester: John Wiley & Sons.

Ekman, P., Friesen, W.V., & Ellsworth, P. (2013). Does the face provide accurate information? In Ekman, P. (Eds.) *Emotion in the Human Face*. (2nd ed.). (pp. 56-97). Los Altos (Calif.): Malor Books.

Ekman, P., & Oster, H. (2013). Review of research, 1970-1980. In Ekman, P. (Eds.) *Emotion in the Human Face*. (2nd ed.). (pp. 147-173). Los Altos (Calif.): Malor Books.

Freund, L. (2015). Contextualizing the information-seeking behavior of software engineers. Journal of the Association for Information Science and Technology, 66(8), 1594.

Fridlund, A. J., & Cacioppo, J. T. (1986). Guidelines for human electromyographic research. *Psychophysiology, 23*(5), 567-589.

Fritz, T., Begel, A., Müller, S. C., Yigit-Elliott S. & Züger, M. (2014). Using psycho-physiological measures to assess task difficulty in software development. In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014).

Goddard, R., D. & Villanova, P. (2006). Designing Surveys and Questionnaires for Research. In Leong, F. T. L. & Austin, J. T. (Eds.), *The psychology research handbook: A guide for graduate students and research assistants* (2nd ed.). Thousand Oaks, Calif.: Sage Publications.

Gomez, P., Zimmermann, P., Guttormsen Schär, S. & Danuser, B. (2009). Valence Lasts Longer than Arousal: Persistence of Induced Moods as Assessed by Psychophysiological Measures. Journal of Psychophysiology, 23, 7-17.

Graesser, A. C., D'Mello, S., & Stahl, K. (2012). Moment-To-Moment Emotions During Reading. Reading Teacher, 66(3), 238–242.

Graziotin, D, Wang, X, and Abrahamsson, P (2015a), Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. *Journal of Software: Evolution and Process*, 27, 467–487.

Graziotin, D., Wang, X., & Abrahamsson, P. (2015b). Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. In Proceedings of the 7th International Workshop on Social Software Engineering (SSE 2015). ACM, New York, NY, USA, 25-32.

Khan IA., Brinkman W., & Hierons RM. (2010). Do moods affect programmers' debug performance? *Cognition, Technology & Work,* 13, 245-258

Kinnunen, P., & Simon, B. (2012). My program is ok--am I? computing freshmen's experiences of doing programming assignments. *Computer Science Education*, 22(1), 1-28.

Kordaki, M. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. Computers & Education, 54(1), 69-87.

Künecke, J., Sommer, W., Schacht, A., & Palazova, M. (2015). Embodied simulation of emotional valence: Facial muscle responses to abstract and concrete words. *Psychophysiology*, 52 (12), pp. 1590-1598.

Lazar, J., Feng, J. H. & Hochheiser, H. (2010). Research methods in human-computer interaction. Chichester: Wiley.

Lewis M. (1999). The Role of the Self in Cognition and Emotion. In Abramson, L. Y., Averill, J. R., Bekerian, D. A., Bentall, R. P., Berkowitz, L., Bradley, B. P., . . . Öhman, A. (Eds.), *Handbook of cognition and emotion*. Chichester: John Wiley & Sons.

McGrath, M. (2014). *Python*. Leamington Spa, Warwickshire, U.K.: In Easy Steps.

Mannila, L., Peltomaki, M., & Salakoski, T. (2006). What about a simple language? analyzing the difficulties in learning to program. *Computer Science Education, 16*(3), 211-227.

Mansor, A. A. B., & Isa, S. M. (2018). The impact of eye tracking on neuromarketing for genuine value-added applications. *Global Business and Management Research, 10*(1), 1-11.

Marghescu, D., Salminen, M. & Ravaja, N. (2011) Media experience elicited by print and tablet news: A psychophysiological perspective. Next Media Report. D1.0.2.1.

Matthews, G. & Wells, A. (1999). The Cognitive Science of Attention and Emotion. In Abramson, L. Y., Averill, J. R., Bekerian, D. A., Bentall, R. P., Berkowitz, L., Bradley, B. P., . . . Öhman, A. (Eds.), *Handbook of cognition and emotion*. Chichester: John Wiley & Sons.

Matsumoto, D., Keltner, D., Shiota, M.N., O'Sullivan, M. & Frank, M. (2008). Facial Expressions of Emotions. In Lewis, M., Lewis, M., Haviland-Jones, J. M., & Barrett, L. F. *Handbook of emotions* (3. ed ed.). New York: Guilford Press.

Müller, S. C., & Fritz, T. (2015). Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. InSoftware Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on (Vol. 1, pp. 688-699). IEEE.

Nikula, U., Sajaniemi, J., Tedre, M., & Wray, S. (2007). Python and roles of variables in introductory programming: Experiences from three educational institutions. Journal of Information Technology Education, 6, 199-214.

Perkins, D.N. & Martin, F. (1986). Fragile Knowledge and Neglected Strategies in Novice Programmers. In Soloway, E. & Iyengar, S. *Empirical studies of programmers: Papers presented at the First Workshop on Empirical Studies of Programmers, June 5-6, 1986, Washington, DC.* Norwood (NJ): Ablex.

Oatley, K., Keltner, D. & Jenkins, J. M. (2006). *Understanding emotions* (2nd ed.). Malden (Mass.): Blackwell.

O'Regan, G. (2012). *A brief history of computing* (2nd ed.). London: Springer.

Partala, T., Surakka, V., & Vanhala, T. (2006). Person-independent estimation of emotional experiences from facial expressions. *Interacting with Computers,* 18(2) 208-226.

PsychLab Hardware Manual (2009).

PsychLab Software Manual PsychLab 8 (2009).

Rajanen, D., Salminen, M., & Ravaja, N. (2015). Psychophysiological responses to digital media: frontal EEG alpha asymmetry during newspaper reading on a tablet versus print. In Proceedings of the 19th International Academic Mindtrek Conference, pp. 155-162.

Rajanen, D., Salminen, M., & Ravaja, N. (2016). Reading a Newspaper on Print versus Screen: A Motivational Perspective. In System Sciences (HICSS), 2016 49th Hawaii International Confer- ence on (pp. 630-637). IEEE.

Rajendra Desai, A. (2017). EEG-based evaluation of cognitive and emotional arousal when coding in different programming languages. University of Oulu.

Rangayyan, R. M. (2015). *Biomedical signal analysis* (Second edition.). Hoboken (N.J.): Wiley.

Rantanen, V., Ilves, M., Vehkaoja, A., Kontunen, A., Lylykangas, J., Mäkelä, E,, Rautiainen, M., Surakka, V., & Jukka Lekkala, J. (2016). A survey on the feasibility of surface EMG in facial pacing. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, 2016, pp. 1688-1691.

Ravaja, N. (2004). Contributions of psychophysiology to media research: Review and recommendations. Media Psychology, 6(2), pp. 193-235.

repl.it (2018, January 15). Retrieved from https://repl.it.

Rose, D. (2008). Psychophysiological Methods. In Breakwell, G. M., Breakwell, G. M., Rose, D., Barrett, M., Fife-Schaw, C., Davis, A., . . . O'Sullivan, D. *Research methods in psychology* (3rd ed ed.). Los Angeles: SAGE.

Saariluoma and, P., & Jokinen, J. P. P. (2014). Emotional dimensions of user experience: A user psychological analysis. International Journal of Human - Computer Interaction, 30(4), 303.

Sammet, J. E. (1991). Some approaches to, and illustrations of, programming language history. *Annals of the History of Computing, 13*(1), 33-50.

Schuurink, E. L., Houtkamp, J., & Toet, A. (2008). Engagement and EMG in serious gaming: Experimenting with sound and dynamics in the levee patroller training game. 2nd International Conference on Fun and Games; Eindhoven, 139-149.

Scott, M. L. (2005). Programming Language Pragmatics. Elsevier Science.

Shaw, T. (2004.) The emotions of systems developers. *Proceedings of the 2004 conference on Computer personnel research Careers, culture, and ethics in a networked environment - SIGMIS CPR '04*, 2004; 124.

Shaver P, Schwartz J, Kirson D, O'Connor C. (1987.) Emotion knowledge: Further exploration of a prototype approach. *Journal of Personality and Social Psychology*, 52(6):1061–1086.

Shneiderman, B. (1986). Empirical Studies of Programmers: The Territory, Paths and Destinations. In Soloway, E. & Iyengar, S. *Empirical studies of programmers: Papers presented at the First Workshop on Empirical Studies of Programmers, June 5-6, 1986, Washington, DC*. Norwood (NJ): Ablex.

Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems*. Cambridge, Mass.: Wintrop.

Sowden, P. & Barrett, P. (2008). Psychophysiological Methods. In Breakwell, G. M., Breakwell, G. M., Rose, D., Barrett, M., Fife-Schaw, C., Davis, A., . . . O'Sullivan, D. *Research methods in psychology* (3rd ed ed.). Los Angeles: SAGE.

Tan, J. -., Walter, S., Scheck, A., Hrabal, D., Hoffmann, H., Kessler, H., & Traue, H. C. (2011). Facial electromyography (fEMG) activities in response to affective visual stimulation. Paper presented at the IEEE SSCI 2011 - Symposium Series on Computational Intelligence - WACI 2011: 2011 Workshop on Affective Computational Intelligence, 45-49.

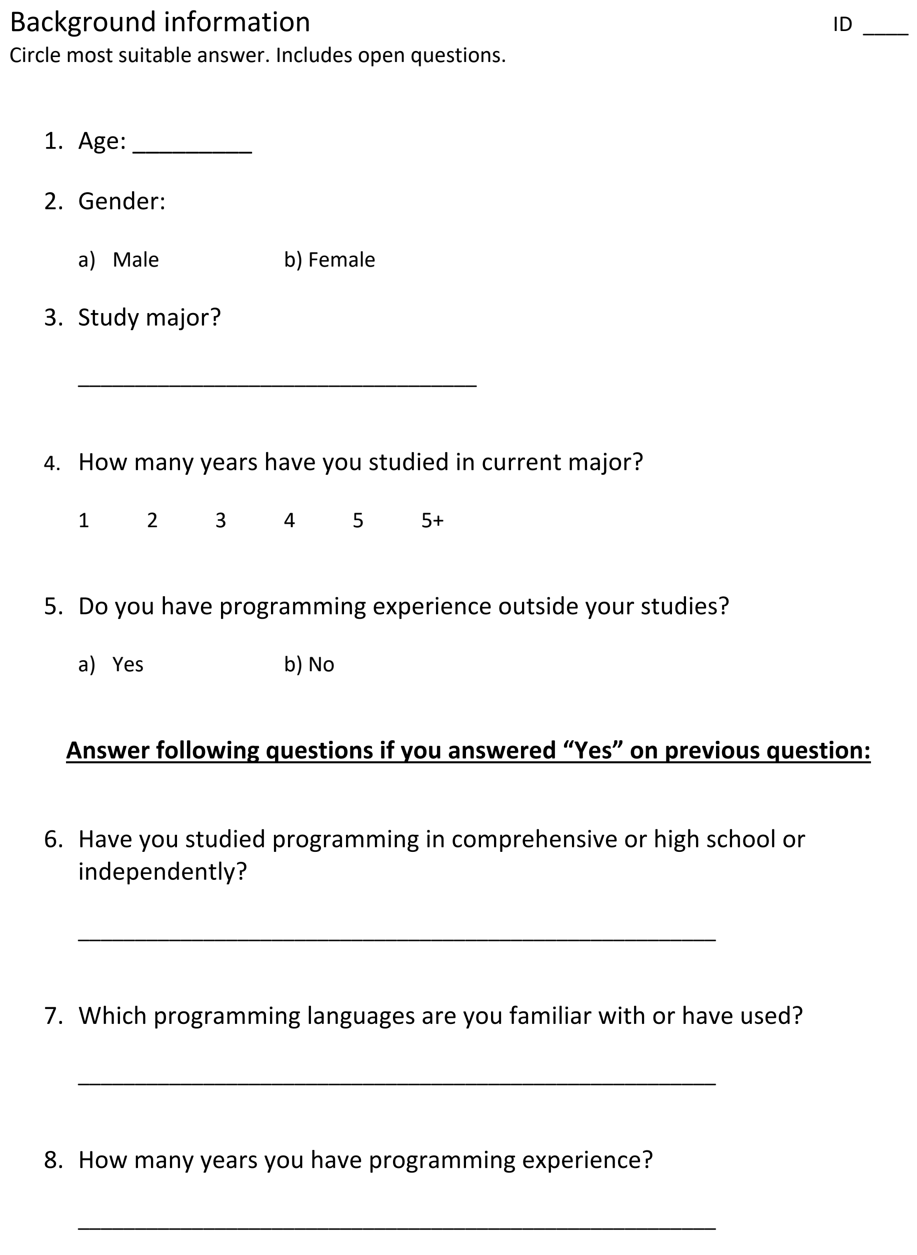
UX Lab (2019, Febryary 22). Retrieved from https://interact.oulu.fi/site/files/FilesDorina/uxlab.jpg.

Wohlin, C. (2012). *Experimentation in software engineering*. Berlin ; New York: Springer.

Zhu, YM. (2016). *Software Reading Techniques: Twenty Techniques for More Effective Software Review and Inspection*. Apress.

Zaccaro St, J. & Marks, M. (1996) Collecting Data From Groups. In Austin, J. T., & Leong, F. L. *The Psychology Research Handbook : A Guide for Graduate Students and Research Assistants*. Thousand Oaks, Calif: SAGE Publications, Inc.

Appendix A. Background information



Background information

ID ____

Circle most suitable answer. Includes open questions.

1. Age: _________
2. Gender:

   a) Male b) Female

3. Study major?

   ____

4. How many years have you studied in current major?

   1 2 3 4 5 5+

5. Do you have programming experience outside your studies?

   a) Yes b) No

Answer following questions if you answered “Yes” on previous question:

6. Have you studied programming in comprehensive or high school or independently?

   ____

7. Which programming languages are you familiar with or have used?

   ____

8. How many years you have programming experience?

   ____

Appendix B – Post-test questionnaire

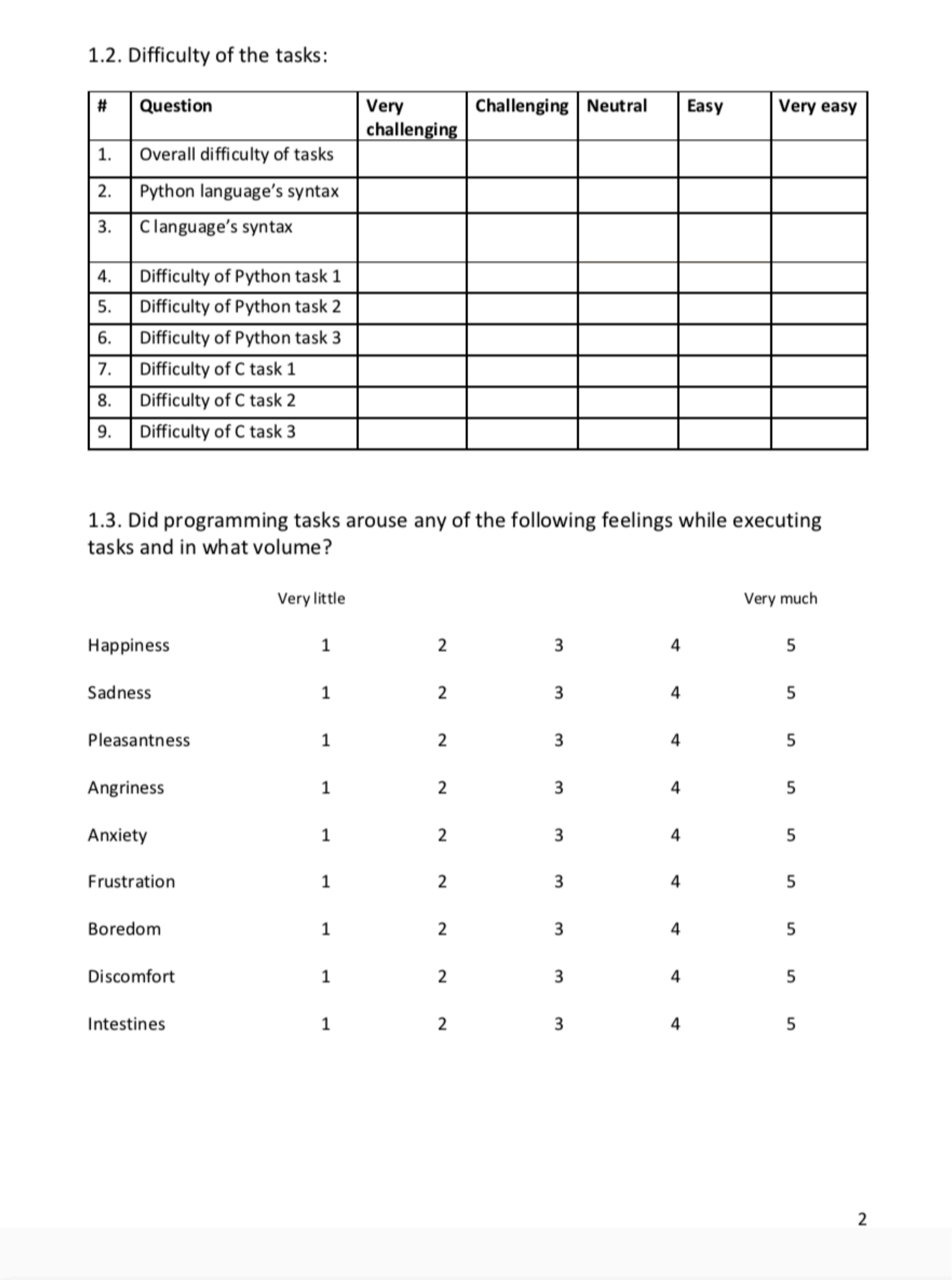
End Questionnaire

ID ______

1. Programming tasks

1.1 Functionality of tasks and opinions regarding tasks. Mark most suitable choice.

#	Question	Strongly disagree	Disagree	No opinion	Agree	Strongly agree
1.	Instructions were clear					
2.	Tasks were interesting					
3.	Enough time was reserved for the tasks					
4.	My performance went well					
5.	I found relevant information for the tasks					
6.	Switching between programming languages was easy					
7.	Programming with C was pleasant					
8.	Programming with Python was pleasant					
9.	Cheat sheet was useful					



1.2. Difficulty of the tasks:

#	Question	Very challenging	Challenging	Neutral	Easy	Very easy
1.	Overall difficulty of tasks					
2.	Python language's syntax					
3.	C language's syntax					
4.	Difficulty of Python task 1					
5.	Difficulty of Python task 2					
6.	Difficulty of Python task 3					
7.	Difficulty of C task 1					
8.	Difficulty of C task 2					
9.	Difficulty of C task 3					

1.3. Did programming tasks arouse any of the following feelings while executing tasks and in what volume?

	Very little				Very much
Happiness	1	2	3	4	5
Sadness	1	2	3	4	5
Pleasantness	1	2	3	4	5
Angriness	1	2	3	4	5
Anxiety	1	2	3	4	5
Frustration	1	2	3	4	5
Boredom	1	2	3	4	5
Discomfort	1	2	3	4	5
Intestines	1	2	3	4	5

2. Personal opinion and comments of this test:

Thank you for participation!