Cognitive load design implications for the Modsy controller

Bram van Driel

University of Twente
Creative Technology
Graduation Project Thesis

Supervisor: Eggink, Wouter, dr. ir.
Critical observer: Faber, Erik, dr. ir.

July, 2021

Table of Contents

Abstract

How easy and understandable a product is to use partly determines whether people can benefit from the features of the product. If people are capable of understanding a product, they can benefit from the possibilities of that product. On the contrary, if the product is too difficult for them to understand, the people who use the product might only become irritated when using the product. The cognitive load of a product is one determinant that influences how easy and understandable the product is to use.

Cognitive load is also an important factor that influences the process of making music. Having as much cognitive load available for the creation of music might enable musicians to put more expression, creativity and fun into their music production. As a starting point of this thesis, a design for a music controller formed the basis that needed to be improved. This music controller is called Modsy which is a product that was developed by Weirdly Wired. Since the cognitive load is an important determinant for a creative process such as making music, reducing the cognitive load might help musicians with their music production. Thus the goal of this thesis is: “Decrease the cognitive load of the Modsy controller while maintaining its main functionalities”.

A literature review helped to reveal six cognitive load reducing categories. These load reducing categories can be used to reduce the cognitive load of a product. By applying these categories to the design of the Modsy, ideas were generated that could be used to reduce the cognitive load of the Modsy. Eventually, an idea was worked out that especially makes use of the load reducing category *pacing learning*. By using the knowledge musicians already have about their software, the required cognitive load of the music controller might be lowered. To do so an additional controller was built with faders as input type. This way the hardware looks more like the software in comparison to the old system. It was expected that by applying the cognitive load reducing category *pacing learning* the cognitive load of the new system would be lower than with the old system. Although no significant results could be found to confirm this hypothesis, the results of the evaluation are promising. Future work is needed to improve the prototype on some aspects and more musicians should test the new system to draw any conclusions.

Acknowledgements

I would like to thank some people for their support throughout this project. First and foremost I would like to thank Robbert-Jan and Olivier. About two years ago we started our company Weirdly Wired and decided to work on a shared passion of ours: music. With the three of us, we have developed a product that we can be proud of and we share some experiences that no one will ever take away from us. Working together on our thesis' and our company Weirdly Wired was sometimes not easy but luckily we all kept laughing.

Secondly, I would like to thank my supervisors, Wouter Eggink and Erik Faber. The feedback and ideas they gave me have helped to bring this thesis to a higher level which I am grateful for.

Glossary

The terms used in this research can have different definitions in different contexts. For this reason, a glossary is given with relevant terms and the corresponding definitions that are used in this research. Reading it from top to bottom is recommended, as some definitions make use of other definitions.

1. **Digital Audio Workstation (DAW)**

A software application used by producers and performers to record, arrange, compose, mix and master audio. This piece of software can be compared to a physical studio filled with music gear.

2. **Musical Instrument Digital Interface(MIDI)**

MIDI is a standard technical communication protocol for a variety of instruments, computers, and other audio devices. It can be used for playing, editing, and recording music.

3. **Ableton Live (Ableton or Live)**

An example of a DAW, that has multiple millions of users. This DAW is made by Ableton for both producers and performers, whereas most DAWs are only focused on producers.

4. **Device**

A device inside of Ableton Live can be an audio effect, instrument, or MIDI effect. This software instrument or effect can be dragged onto a track, from which it can either produce audio or manipulate the audio in that track.

5. **Plugin**

A software component that enhances audio-related functionality. This component is "plugged in" to a DAW to add extra functionality, for instance for digital sound synthesis or processing.

6. **Virtual studio technology(VST)**

VST plug-ins can be used inside a DAW and thus also in Ableton. VSTs come in three types: VST instruments, VST effects and VST MIDI effects. VST plug-ins can be used to enhance a DAW. Using VSTs musicians are not bound to the capabilities of a DAW. VSTs help to enrichen the possibilities of a DAW so, even more, is possible than with just a DAW on its own.

1. Introduction

The introduction starts with a general introduction to the main topic. Here the problem and its importance will be described. Secondly, the product that is the starting point of this thesis will be introduced. The second section also introduces the scope in which this thesis will take place. The third section of the introduction states the goal and research questions of this thesis. Finally, an outline of this thesis will be given to get an overall image of what will be done in which chapter.

1.1 Problem description

Thinking about something requires the working memory of a person. It is imaginable that thinking about multiple things at the same time requires more working memory of a person than thinking of one thing at a time. This can be explained by looking at studies that are available about cognitive load. Cognitive load can be defined as “the amount of ‘mental energy’ required to process a given amount of information” (Feinberg & Murphy, 2000, p. 354). Cognitive load studies suggest that mentally integrating various sources of information whilst solving a problem significantly affects task performance (Beghelli & Prieto, 2020).

Making music is a creative process and could be influenced by cognitive load. A lot of research is available about how learning academic skills is affected by cognitive load. One might think that for the creative process this is different. Redifer et al. (2019) suggest that creative thinking is influenced by cognitive factors in the same way traditional academic tasks are. They mention that the working memory capacity and the cognitive load influence creative thinking just like they influence learning academic tasks.

Designing a tool for musicians with an optimized cognitive load can help musicians with their music production. Following the theory about creative thinking of Redifer et al. (2019), it might be beneficial for musicians to have the most working memory available for the creation of music. Therefore, developers of tools that help musicians with their music production might need to think about how to manage the cognitive load of that tool. It can be beneficial for musicians to use a tool with a low cognitive load to enable the musicians to focus on their music production. This could enable them to put more creativity, expression and fun into their music production.

1.2 Introduction Modsy Controller

This section introduces the Modsy controller and its main goals. First, the reason why this product is relevant will be explained. Secondly, the product itself will be introduced.

1.2.1 The current status of music production

Music production has changed a lot over the last 20 years. The rise of computers created a new way to create and manipulate music. Where musicians used to need whole rooms of equipment to create a song, musicians nowadays can record whole albums with the use of their smartphone (Pierce, 2017). While a smartphone might be slightly inconvenient, a laptop or computer is a perfect basis for music production and most modern musicians rely on software for music creation and manipulation. The software program at the core of this music production is called a Digital Audio Workstation, or DAW for short. This software can be seen as the studio of a music producer. Inside a DAW a musician can use different digital instruments and effects that can be used within music production. A DAW also allows for the composition into a song and ways to mix, master, and finalize one's music.

Music production has become more accessible, flexible, and cheaper. However, there is also a downside to computer-based music production. This is because musicians can only use their mouse and computer keyboard to manipulate the sounds of instruments and effects. This is far from the original feeling of analogue instruments and effects. Analogue instruments and effects create a workflow that is more expressive, creative and it's a whole lot more fun. Tweaking multiple buttons at the same time, mistakes that turn out to be masterpieces, and collaborating with peers are examples of the pros of having physical control while making music.

1.2.2 The Modsy controller

This is where Modsy steps in (Weirdly Wired, 2021). Modsy is a product created by Weirdly Wired which is a start-up founded by Olivier Mathijssen, Bram van Driel, and Robbert-Jan Berkenbos: all Creative Technology students at the University of Twente. The main goal of Modsy is to create an instrument that enables musicians to put more creativity, more expression, and more fun into their music production.

Modsy creates a way for musicians to get an analogue feeling over their digital music production. Modsy is a hardware and software solution that creates an environment that allows musicians to instantly take physical control over any digital instrument or effect inside of their DAW. The controller will be connected to the DAW of a user. The controller has 32 parameters to control an instrument or effect, with a display above each of these parameters, see figure 1.

Figure 1: A render of the Modsy controller

The first step of using the Modsy controller is for the user to select a digital instrument or effect in his DAW, see figure 2-1. Secondly, the user can press the mapping button on the controller, see figure 2-2. By pressing the button the controller maps itself to the selected digital instrument or effect, see figure 2-3. All of the parameters will be linked to software parameters of the digital instrument or effect and the musician can start manipulating the sound right away. The displays above each input element show the parameter its name and value. This is essentially what the Modsy controller does;

it provides direct analogue control over digital instruments and effects and ensures that the manipulation of these parameters is fully intuitive. This creates an analogue workflow for any digital music production tool.

The unique selling points of Modsy are:

- Automatic mapping to any digital instrument or effect within a DAW
- Display for each parameter
- Enough parameters for complete analogue control

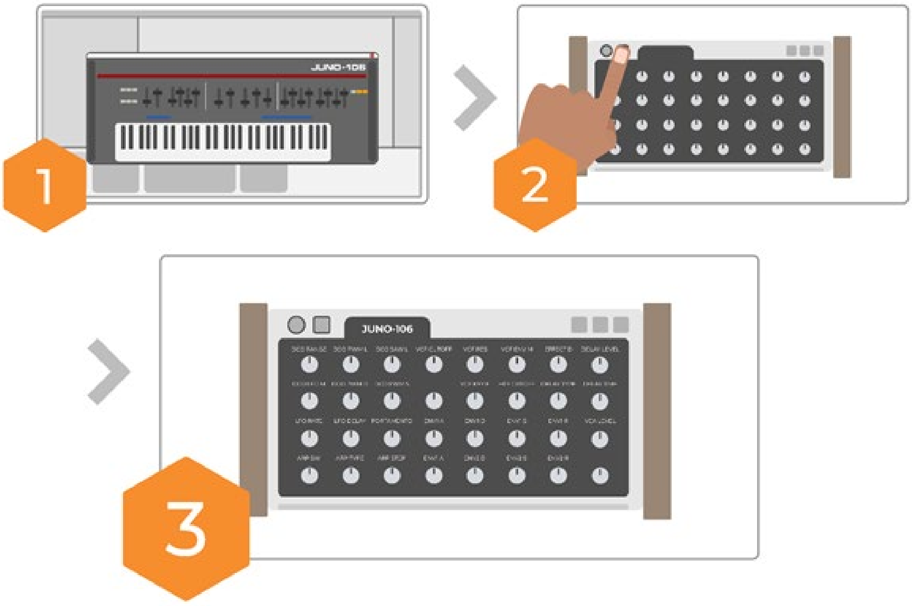

Figure 2: the mapping process of the Modsy, 1) select a device in Ableton 2) press the map button on the Modsy 3) the Modsy is mapped to the selected device

1.3 Goal and research questions

The three main goals of the Modsy controller are to enable digital musicians to put more creativity, more expression and more fun into their music production. To reach these goals the controller should be easy and understandable to use. Looking at the cognitive load can be an important step to make the controller more easy and understandable to use. This leads to the following goal and research questions.

1.3.1 Goal

Decrease the cognitive load of the Modsy controller while maintaining its main functionalities. These main functionalities will be explained in section 4.2.

1.3.2 Research questions

The main research question for this thesis is:

- **How to decrease the cognitive load for the Modsy controller?**

To answer this main research question several sub-questions are formulated. The three sub-question to formulate an answer to the main research question are:

1. How can cognitive load theory be used within the context of product design?
2. How to implement changes upon the Modsy controller to decrease the cognitive load?
3. What are the differences between the new prototype and the original Modsy controller, in terms of cognitive load, overall useability and experience of making music?

The first sub-question will be answered using literature research. The answer to question one, directly informs the direction to answer the second sub-question. There are probably many answers to sub-question two. It is important to notice that the main functionalities should be maintained. This will limit the number of possibilities already. The answer to sub-question two is an idea that can be worked out as a prototype.

The last sub-question is answered by developing a prototype and testing how this prototype differs from the original Modsy. The main topic of this thesis is the cognitive load of the Modsy controller. Reducing the cognitive load can help to make the controller more easy and understandable to use. However, reducing the cognitive load should not be a goal on its own. Therefore research question three is also about the overall useability of the controller and the experience of making music.

1.4 Outline

This study starts with background research in chapter 2. During the background research, theory about cognitive load will be studied. To test how other products apply this theory, several products will be analysed. Looking at how other products manage the cognitive load can help to spark the creativity to come up with solutions for the Modsy controller. The third chapter is about the methods and techniques that will be used during the design process. In chapter four the actual design process starts with the ideation phase. During the ideation phase, lots of ideas are generated to eventually come to a preliminary concept. This preliminary concept will then be specified more during the specification phase, which is documented in chapter five. The goal of the specification phase is to form the preliminary concept from the ideation phase into an idea that can be worked out during the next phase. Chapter six is about the realization of the prototype. The prototype that will build during the realization phase can be used to evaluate the concept and see whether it works. Chapter seven discusses how this evaluation will be done and what the results are from the evaluation. Finally, chapter eight comes back to the earlier mentioned goal, research questions and possibilities for future work. After chapter eight, all the references and appendices are stated.

2. Background research

The background research consists of two sections, first, a literature review will be done about cognitive load. The outcome of the literature review will be used for the second section of the background research, the state of the art. In the state of the art, creativity is sparked by looking at other products. This background research is meant to get a better understanding of cognitive load theory. Exploring possibilities how to apply cognitive load theory in the context of the design of a music controller helps during the ideation phase to come to an idea that could be successful.

2.1 Cognitive load and its implications for product design

The first part of the background research aims at getting a better understanding of cognitive load. The theory that is needed for this research is assessed using a literature review that consists of three sections. Firstly, a definition of cognitive load will be established and three types of cognitive load are discussed. The second section is about how guidelines can help to decrease the cognitive load of a product. Finally, a conclusion of the literature review will be given to get an overview of the results.

2.1.1 Defining cognitive load

The design of a product should enable the user to reach its goal in such a way that there is working memory left for learning how to use the product. A lot of literature about cognitive load is based on the cognitive load theory of Sweller (1988). The basis of this theory is the assumption that a person has two kinds of memory: a short-term or so-called working memory and a long-term memory. Sweller incorporates the definition of the working memory from Miller (1956).

Miller suggests that the working memory of a person can only save up to seven informational elements. This is the maximum amount of information a person can process at a certain time. Sweller (1988) suggests that in order to learn something, the information needs to be transferred from the working memory to the long-term memory. The information in the long-term memory is saved in so-called schemas. Transferring the information from the working memory to the long-term memory also uses the working memory. The size of a schema determines how much information can be processed in the working memory. Schemas are cognitive structures that allow people to recognize a problem and put that problem in a certain category. These schemas can be used to solve problems. If experts in a certain topic try to solve a problem they quickly recognize what kind of problem it is because it is in one of their schemas. By knowing which category of problem they are dealing with, they also know the best way of solving the problem(Sweller, 1988). A novice in a certain topic has less deep schemas and therefore is less likely to recognize in which problem category the current problem belongs. Since novices might not recognize the problem category it is much more difficult to solve the problem since they cannot compare the current problem to a problem from one of their schemas. In essence, people group certain problems by their similarities. Similar problems often call for a similar problem-solving technique.

To get a better understanding of how transferring information from the working memory to the long-term memory works, cognitive load theory can be used. Klepsch et al. (2017), Renkl and Atkinson (2003) and Van Merriënboer and Aryres (2005) are only a few examples of researchers that use a model with three types of cognitive load: intrinsic, extraneous and germane cognitive load. Kalyuga (2011) however, suggests a different model where the germane load is indistinguishable from the intrinsic load. Although he has good reasons to abandon the model with three types of cognitive load, there is no clear evidence for either one of the models. Kalyuga (2011) also suggests that if the model with three types of cognitive load helps to explain certain scenarios it can still be used. Debue and van de Leemput (2014), Lee and Wong (2014), Leppink et al. (2013) and Leppink et al. (2014) all reference the paper of Kalyuga (2011) but still use the model with three types of cognitive load.

This thesis will continue with the model of three types of cognitive load, generally-speaking researchers agree on the definition of the three types. Paas et al. (2003) describe the three types of cognitive load very clearly. Intrinsic cognitive load is determined by the complexity of the topic that is being learned. So the more complex a topic is, the higher the intrinsic cognitive load. Extraneous cognitive load has to do with the load needed for processing the way of presenting the information. Paas et al. refer to extraneous cognitive load as ineffective cognitive load and mention that it interferes with saving information in the long-term memory. Germane cognitive load uses the working memory for transferring the information to the long term memory (Paas et al., 2003).

To learn something there should be enough working memory left for germane load. Galy et al. (2012), Galy and Mélan (2013) and Paas et al. (2003) all suggest that the three types of cognitive load are additive, this means that intrinsic cognitive load is a basis that depends on the complexity of the material. Extraneous and germane cognitive load work in tandem, meaning that if the extraneous load uses a lot of working memory there is not so much cognitive load left for germane cognitive load. However, if the extraneous load does not use much working memory there is a lot of working memory left for germane cognitive load. This enables the learner to use its working memory for schema acquisition and automation and thus learns efficiently.

Although there is not so much literature available about cognitive load within the context of product design, cognitive load might still be a valuable concept for designers. Within the context of psychology, a lot of research has been done about cognitive load, these concepts can be used to think of ways to incorporate cognitive load theory in design. When using a product for the first time a lot of things about the product need to be learned. Depending on the type of product, cognitive load should be treated carefully. Think for example about a computer mouse which for most users is solely used to navigate through a computer. The task which is performed on the computer is the intrinsic load of the overall task. Every part of the working memory of the user that is needed to use the mouse can be seen as extraneous cognitive load. To have more working memory left for germane cognitive load the mouse should not use too much cognitive load.

2.1.2 Incorporating cognitive load theory in product design

For designers, it could be important to integrate the knowledge of cognitive load while designing. There are several ways to apply cognitive load theory in design. Mayer and Moreno (2003), Clark et al. (2006) and Van Merriënboer and Sweller (2010) all presented different guidelines or methods to manage cognitive load. Clark et al. define twenty guidelines split up into six categories that can be used to reduce intrinsic and extraneous cognitive load. Mayer and Moreno describe nine load reducing methods that are very comparable to the guidelines of Clark et al. (2006). Van Merriënboer and Sweller describe fifteen design guidelines split up into four main categories. All the different papers have comparable categories but they are not completely the same. In this section, these different categories are explained and compared.

The first category consists of guidelines about using visuals in combination with audio. In the paper of Clark et al. (2006) this section contains four guidelines that are all about using a combination of audio and visuals. Van Merriënboer and Sweller (2010) have a similar design guideline “the modality principle” but they only mention to replace written text with spoken words. They do not mention anything about the combination of audio and visuals. Mayer and Moreno (2003) describe the load-reducing method “off-loading”, which is also solely about moving some of the essential material from the visual channel to the auditory channel. They do not mention anything about the combination of audio and visuals. For designers, this category is not so easy to use. When designing some sort of instruction this is highly relevant. By replacing written text of instruction with spoken words, the extraneous cognitive load can be lowered.

The second category consists of guidelines about enabling the learner to focus on the important material. In the paper of Clark et al. (2006) this section contains two guidelines that are both about pointing the learner in the right direction to focus on the material that needs to be learned. Van Merriënboer and Sweller (2010) do not describe a guideline that can be compared to this. Mayer and Moreno (2003) describe the load reducing method “signaling”, which describes how the cognitive load can be reduced by providing cues for how to process certain material. Designers can use these guidelines by making a product that uses signals to point at important features of the product. An example of this could be the size and colour of the power button on a TV remote.

The third category consists of guidelines about eliminating unnecessary information. In the paper of Clark et al. (2006) this section contains three guidelines that describe how to design an instruction to be minimalistic. Van Merriënboer and Sweller (2010) describe this as the guideline “the redundancy principle”, which is also about eliminating information that is not needed to be explained. Mayer and Moreno (2003) describe the load reducing methods “eliminating redundancy” and “weeding”, which are comparable to the section of Clark et al. (2006). Designers can use these guidelines by making a very minimalistic product that does not have any unnecessary extras. An example of this could be a very minimalistic computer mouse. There are a lot of very complex mouses that have a lot of features but therefore also require a higher cognitive load. A very minimalistic mouse has a lower cognitive load than a very complex mouse.

The fourth category consists of guidelines about giving users mnemonics to reduce the cognitive load. In the paper of Clark et al. (2006) this section contains two guidelines that describe how user hints can help to reduce extraneous cognitive load. This category is more about bypassing the working memory and therefore this section is not mentioned in the paper of Van Merriënboer and Sweller (2010) nor Mayer and Moreno (2003). Designers can use this category by giving the users mnemonics on how to use the product, an example of this could be the letters on the keyboard which indicate their functionality. Without the letters on the keyboard, users would have to remember all the keys themselves, which requires a high cognitive load.

The fifth category is about presenting new information in parts and letting the user decide when to learn what. In the paper of Clark et al. (2006) this section contains four guidelines which are about segmenting, sequencing, and pacing learning. Van Merriënboer and Sweller (2010) describe two guidelines that relate to the section of Clark et al. (2006). The “simple-to-complex” strategy and the “low- to high-fidelity” strategy both describe strategies on how to enable the learner to learn in parts. Mayer and Moreno (2003) describe the load reducing methods “individualizing”, “segmenting” and “pretraining” which all relate to this section. Designers can use these guidelines by enabling the user to learn how to use the product in parts, or by using the already existing knowledge of the user. An example of this could be to apply often used design patterns while designing. Since most of the users are already familiar with the design patterns, the product is easier to use.

The sixth and last category is about applying techniques to practice efficiently. In the paper of Clark et al. (2006) this section contains four guidelines about how different types of practice can help to deepen schemas. Van Merriënboer and Sweller (2010) also describe various guidelines about how to practise efficiently. Mayer and Moreno (2003) do not describe ways on how to practise efficiently. Designers can use these guidelines by thinking carefully about which type of practice an introduction should contain to learn how to use the product. An example of this could be to make a video walkthrough of how to use the product and give exercises to the users.

2.1.3 Conclusion literature review

The objective of this literature review was to gain insights on how to apply cognitive load theory in design. Cognitive load can be defined as the amount of mental energy required from the brain to process certain material and consists of three categories: intrinsic cognitive load, extraneous cognitive load and germane cognitive load. To build schemas and learn how to use a product, there should be enough working memory left for germane cognitive load. Several design guidelines can help to reduce the cognitive load of a product and leave working memory for germane cognitive load. These guidelines can be split up into the following six categories: using a combination of audio and visuals, using signals, minimalistic design, provide mnemonics, pacing learning and practise styles. An overview of the guidelines for reducing the cognitive load can be seen in table 1. These guidelines can be used within the context of product design and will be used within this thesis to reduce the cognitive load of the Modsy controller.

The literature used for this review is mostly from the field of psychology and in the context of learning, which might be less relevant for product design. However, when using a product the user needs to learn a lot of new things and therefore the theories from psychology could be relevant for designers.

The influence of the given guidelines on the cognitive load was not discussed in this literature review whereas there might be big differences between them. The provided guidelines probably do not influence the cognitive load equally. There is literature available about the weight of certain guidelines within the context of learning which was beyond the scope of this literature review. No literature could be found about how much these guidelines influence the cognitive load within the context of product design.

Name	**Meaning**	**Example**
Combination of audio and visuals	Make use of the auditory and visual channels of the users.	-
Using signals	Enable the user to focus on the most important features of a product.	TV remote
Minimalistic design	Remove unnecessary and redundant material from a product.	Minimalistic computer mouse
Provide mnemonics	Help to remember the user of the functionality of certain parts of the product.	Computer keyboard
Pacing learning	Let the user learn the functionality of the product in steps instead of all at once.	Using design patterns
Practising techniques	Help the user to practice the functionality of the product efficiently.	Video walkthrough

Table 1: An overview of the six load reducing categories

2.2 State of the art

The state of the art uses the concepts from the previous section and provides examples to spark creativity for the continuation of this thesis. The state of the art consists of three sections, the first section gives examples of the categories that influence cognitive load as described in section 2.1.2. These examples can help to create a better feeling for what these categories mean. Only the first category *combination of audio and visuals* is not listed here. Since this thesis is about a musical product it does not seem practical to use audio as an aid to reduce the cognitive load.

The second section gives an overview of how other musical tools apply the six categories. These musical tools are analysed and reviewed on the load reducing methods from chapter 2.1. The review is done by looking at the products and grading them. Finally, the last section shows how the results of the state of the art can be used for the continuation of this thesis.

2.2.1 Examples of the load reducing categories

Using signals: TV-remote

Figure 3: A TV remote (LG, 2021)

A TV remote is a great example of how to use signals to reduce the extraneous cognitive load. TV remotes often have differences in button size, button forms, button colours and they make great use of groupings. For example, the power button on the remote in figure 3 is round and red. This indicates that the button is different from the buttons that can be used to navigate through the menus of the TV. The volume buttons on this TV remote are bigger than the other buttons to indicate their importance. The buttons with a number on them are used to change channels. On this remote, they are placed conveniently together to indicate that their functionality is comparable. This is an example of how groupings can help to decrease the extraneous cognitive load.

Minimalistic design: computer mouse

Figure 4: A minimalistic mouse (Microsoft, 2021)

Figure 5: A less minimalistic mouse (Logitech, 2021b)

Computer mice come in a lot of different styles and the use of the mouse determines which mouse fits the user best. Figure 4 shows a very minimalistic mouse. This mouse only has the key features of a computer mouse and therefore almost every user can use this mouse intuitively. Figure 5 shows a computer mouse with a lot more buttons and functions. This mouse is probably less intuitive to use for a lot of users. The mouse from figure 4 is a great example of how to design something minimalistic. An important thing to notice here is that the extra functionalities of the mouse in figure 5 can help the user to improve its workflow. Although the cognitive load might be higher with the mouse from figure 5 the workflow of the mouse might also be more efficient. It might be very important for designers to have a clear overview of the requirements of the user to make sure the design fits the use. With these requirements, the designer can think carefully about finding a balance between keeping enough functionality for the user and designing minimalistically.

Provide mnemonics: computer keyboard

Figure 6: A computer keyboard(Logitech, 2021a)

Computer keyboards have letters written on the keys to help the user remember what the functionality of the button is. Without the letters on the keys, the user would have to remember all the functionalities of the keys which probably increases the cognitive load. Having mnemonics greatly decreases the cognitive load of the keyboard. Figure 6 shows how a keyboard uses the load reducing category *provide mnemonics.*

Pacing learning: Monogram Creative Console

Figure 7: Monogram Creative Console (Monogram, 2021)

The Monogram Creative Console is a so-called modular controller. It is fully customizable because you can add modules yourself and therefore build your own controller. As can be seen in figure 7 several modules of the controller can be configured by the users. For novice users, the controller can be very minimalistic and for advanced users, the controller can be very advanced. There is a wide variety of modules available with very creative control possibilities. The monogram is compatible with a lot of software and can also be used for music production. This is a very good example of how to implement the cognitive load reducing category *pacing learning* in products. Users can determine for themselves how many modules they want to use, this enables the user to learn in steps and at their own pace.

Practise styles: video game Portal

Figure 8: Portal 2 (Steam, 2011)

Video games are often great examples of how to make an introduction. Valve, the developer of Portal, did a great job at making a tutorial while the players did not even know they were playing a tutorial. On an online platform, a lot of players of Portal 2 rate their practice style very high (Reddit, 2017). A review about the introduction of the game describes the introduction as the following: "When the game wasn't making you think hard about how to arrange portals to get through the next test chamber, it may well have been making you laugh." (Totilo, 2011, p.1). Throughout the whole game, new elements are introduced and apparently, players think this is a very good way for an introduction. Figure 8 shows the front picture of Portal 2, as can be seen here how the player of the game jumps through portals.

2.2.2 Examples of music products

The second part of the state of the art lists five products that can also be used for music production. For every product, a short explanation is given what kind of product this is and what it contributes to the state of the art. To look at the cognitive load of the product the six categories from section 2.1.2 are used. Using a table the products are rated on how well they apply the six categories. With the rating, motivation is given what this product does well and how they could improve on this certain category. The last product of this section is the Modsy controller itself. Also here a rating is given on how well the cognitive load reducing categories are applied

Please note:
This part of the state of the art is highly subjective. The ratings given to the examples of other music products are based on observations of the product.

Korg nanoKONTROL studio

Figure 9: Korg NanoKONTROL studio (Korg, 2021)

The korg nanoKONTROL is usefull as a mixer. This is something different from the Modsy controller since it is not possible to make any music with the mixer. Korg is a company that originates from Japan and has multiple DAW controllers on the market. The controller has eight channels with buttons, potentiometers and faders as can be seen in figure 9. The controller has names on the buttons which serve as mnemonics for the user to remember the functionality of the button. There are eight different sections on the controller which help the user to distinguish between the eight different groups on a DAW. In table 2 the controller is rated on the six load reducing categories.

Cognitive load category	**Score (1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	7	The controller has a possibility of controlling eight channels and also uses groupings to indicate this to the user. However, no colours are used to make the product even more intuitive to use.
Minimalistic design	6	The controller has a lot of controls but the user only has to remember one channel to understand the other seven elements.
Provide mnemonics	8	The buttons on the controller have mnemonics on them to help the user remember the functionality.
Pacing learning	7	The controller has the same channel layout as a lot of DAWs which help users to understand the controller. This way users can use existing knowledge from their DAW to understand their controller.
Practice styles	3	Korg has some introduction videos, but no specific introduction is available for the controller.

Table 2: Rating of the Korg nanoKontrol studio according to the cognitive load categories

Ableton Push

Figure 10: Ableton Push 2 controller (Ableton, 2021)

Weirdly Wired considers the Ableton Push 2 to be a big competitor of Modsy. Looking at how Ableton handles cognitive load can give inspiration on what could make Modsy a more unique solution. The Push 2 has very advanced control within Ableton Live. It is very highly rated among producers and performers, the controller scores especially high because of its "greatly improved workflow" (Musicradar, 2016). A big con of using the Ableton Push 2 is that the required level of expertise of Ableton Live is quite high (Dashfade, 2020). By looking at the controller, see figure 10, it can easily be seen that the controller has a lot of buttons and there is no easy way of knowing which button does what. For beginners, this controller might even obstruct their creative process more than accelerating it. Looking at the comments and critiques from customers on the Push 2 can help to prevent that Modsy has the same flaws as the Push 2. Table 3 shows a rating of the Ableton Push 2 according to the cognitive load reducing categories.

Cognitive load category	**Score (1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	5	No use of groupings however it does use different knob sizes to indicate which buttons are important.
Minimalistic design	4	The controller provides a lot of functions which makes it difficult to understand for beginners.
Provide mnemonics	4	Some buttons have text to indicate what they do but the main functions do not.
Pacing learning	7	The controller uses a lot of functionality from Ableton Live, if a user is familiar with this the controller becomes a lot easier to understand.
Practice styles	5	Ableton provides lots of videos on how to use the controller however there are no creative introductions.

Table 3: Rating of the Ableton Push 2 according to the cognitive load categories

MP Midi Controller

Figure 11: MP Midi controller(MP Midi, 2021)

The MP Midi controller is made by a startup and is not officially released yet. The MP Midi controller shows a lot of potential by its use of a big screen. On the website of the MP Midi controller, there are several product pictures with some minor differences in them. Figure 11 shows how the controller potentially is going to look like. The screen provides every knob with a small label that indicates which parameter that button is controlling, this way the user always knows which knob is mapped to which parameter. This is a great example of how mnemonics can help to improve a music production controller. Although people are very enthusiastic about the use of a screen not everybody is convinced that one big screen is the perfect solution. Since this controller is not officially released yet there is no feedback from customers on how intuitive this screen is to use. Table 4 shows a rating of the MP Midi controller according to the cognitive load categories.

Cognitive load category	**Score (1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	1	No use of groupings or different knob sizes
Minimalistic design	7	The controller has encoders and they do what you expect them to do. Although there are 32 knobs it is still manageable to understand what they do.
Provide mnemonics	9	Every knob is provided with a label to indicate what the knob is controlling. This helps the user to understand the functionality of every button.
Pacing learning	7	The graphics on the screen are probably familiar to the user. Therefore most functionalities of the controller are already known by the user. However, the layout of the controller is different from the interface of the DAW.
Practice styles	-	There is no official release yet so it is unknown how MP Midi is going to help their users with practising.

Table 4: Rating of the MP Midi controller according to the cognitive load categories

Akai professional LPD8

Figure 12: Akai professional LPD8 (Akai Professional, 2021)

Akai is a very big company and they have made a lot of products for digital musicians. The LPD8 is a great example of how a very simple and cheap controller also has a lot of advantages. The LPD8 which is shown in figure 12 controller is especially rated for its simplicity and ease of use, therefore it can especially help novel musicians a lot. This controller falls in a completely different price range than the Modsy controller and only offers simple control to users. Because of its simplicity, the controller has a low cognitive load but the amount of control the controller offers is quite limiting. For the Modsy controller, such a simple design would probably not work because it does not offer the same amount of control which is one of the main assets of Modsy. Table 5 shows a rating of the Akai professional LPD8 according to the cognitive load categories.

Cognitive load category	**Score(1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	4	No use of groupings but there are differences in knob sizes.
Minimalistic design	10	Only 16 parameters can be controlled and therefore it is easy to remember which button does what.
Provide mnemonics	5	The small buttons at the bottom left do have mnemonics to help the user however the other buttons do not.
Pacing learning	1	No use of pacing learning
Practice styles	5	Akai provides videos to help the user get started. There is no creative introduction.

Table 5: Rating of the Akai professional LPD8 according to the cognitive load categories

Mine S

Figure 13: Mine S (Special Waves, 2021)

The Mine S controller is a product made by Special Waves which is a start-up based in Italy. The Mine S, which can be seen in figure 13, is a modular controller just like the Monogram Creative Console. A difference with the Monogram Creative Console is that the Mine S has a fixed grid that can be used. The controller is not officially released yet so how well this controller works is hard to judge. The Mine S uses pacing learning very well since musicians can add new elements of control themselves once they are ready to do so. Looking at the Mine S and the Monogram Creative Console it seems like that letting a user decide when to add a new element of control is a very good option to decrease the cognitive load. This would fall under the category of pacing learning since the user can learn several parts of the product in stages. Table 6 shows a rating of the Mine S according to the cognitive load categories.

Cognitive load category	**Score (1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	6	The colours can be used to make groupings depending on the user's preferences.
Minimalistic design	7	Depending on the configuration the controller can be very simple. However, because of the fixed grid, some elements cannot be placed very intuitively.
Provide mnemonics	1	The controller does not provide any actual mnemonics.
Pacing learning	8	The controller is not as modular as the Creative Console because of its fixed grid. However, within this grid, it is possible to completely customise the type of control and therefore it allows users to learn how the controller works step by step.
Practice styles	-	There is no official release yet so it is unknown how Special Waves is going to help their users with practising.

Table 6: Rating of the Mine S according to the cognitive load categories

The current Modsy design

Figure 14: Modsy controller (Weirdly Wired, 2021)

The current Modsy design is established through elaborate user testing and several design iterations. Through the whole design process of the controller, the design of the controller has changed drastically multiple times, the current design of the controller can be seen in figure 14. Most of the decisions that the Weirdly Wired team made were not based on lowering the cognitive load. A more in-depth analysis of the current design of the Modsy controller will be performed in section 4.2. Table 7 shows a rating of the Modsy controller according to the cognitive load categories.

Cognitive load category	**Score (1-10)**	**Motivation**
Using visuals in combination with audio	1	No use of audio
Using signals	8	The LEDs on the controller can be used to make groupings. Also, the size of the buttons can be changed by the users. This helps to indicate which parameters are more important than others.
Minimalistic design	2	The controller has a lot of input elements and is not so minimalistic.
Provide mnemonics	8	The controller has small screens to indicate the functionality of every input parameter. The top bar however does not have any mnemonics on its usage.
Pacing learning	5	A user can decide not to bind one of the input parameters to the software. This would mean the screen stays black and the user cannot use this knob. This decreases the cognitive load and once the user wants to bind more parameters to the controller this is possible. With this configuration, the user can learn how to use the controller in steps to lower the cognitive load.
Practice styles	1	Currently, there is no introduction.

Table 7: Rating of the Modsy controller according to the cognitive load categories

2.2.3 Conclusions state of the art

Looking at other music products and thinking carefully about the implementation of the categories it is possible to conclude that category one *using visuals in combination with audio* is not relevant within the context of a music product. It is probably very annoying for a musician to hear other audio than the audio the musician is producing and therefore this category is excluded from the rest of this thesis. Also, category six, *practise styles,* will not be investigated further upon in the rest of this thesis. This category is highly relevant for music products and having a good introduction can help music controllers to be easier to understand. A different thesis is about how to design an easy and engaging introductory tutorial for the Modsy (Berkenbos, 2021). Therefore, during the rest of this thesis, this category will be excluded.

Seeing how other products deal with cognitive load problems helps to think of creative ways to decrease the cognitive load. The six categories of how to reduce cognitive load provide a good foundation to assess the cognitive load of a product. Looking at how other musical products implement the six categories can help to think of interesting ways to implement these changes in the Modsy controller. Another interesting thing to look at is the trade-off between the cognitive load and the functionality of the product. Modsy is a complex product with a lot of features. If a device has a lot of features that a user needs to think about, the cognitive load of the product is likely to be higher. The challenge is to design something that helps users to understand how the controller works without removing any functionalities.

Looking at the assessment table of the Modsy it can be seen that the controller already scores quite high on the categories *provide mnemonics* and *using signals.* The current design already applies these categories quite good and thus these categories are possibly less interesting to look at. Especially the categories *minimalistic design* and *pacing learning* are categories that can be applied better. During the ideation phase, it is important to keep this assessment of the Modsy controller in mind to come up with something that has the highest chance of reducing the cognitive load.

3. Methods and techniques

This section explains the different methods and techniques that are used during this thesis. First section 3.1 gives an overview of the creative technology design process. This is the method that will be followed during this thesis. After section 3.1, the techniques that will be used during this thesis will be explained one by one. The order in which the techniques are presented here is also the order in which the techniques will be used during this thesis.

3.1 Creative technology design process

The creative technology design process(Mader & Eggink, 2014) is a method to design innovative products, applications and services. This design process starts with a divergence phase where the design space is opened up and defined. During the divergence phase, many ideas are generated and the main goal is to foster the creativity of the designer. After the divergence phase, the convergence phase is meant to reduce the design space. Using requirements and available knowledge these reductions are meant to eventually lead to a certain solution (Mader & Eggink, 2014). The whole design process consists of four phases that will be explained in this section. After the four separate phases are explained, figure 15 gives an overview of the four design phases.

3.1.1 Ideation

The design process starts at the ideation phase. The starting point of the ideation phase is often a design question but for Creative Technology the starting point can also be an already existing piece of technology (Mader & Eggink, 2014). During the ideation phase ideas are generated for the to-be-designed product. Creativity for these ideas may come from a flash of inspiration or related work. The result of the ideation phase is a basic concept that will be worked out further in the specification phase (Mader & Eggink, 2014). It is very usual to draw up functional and non-functional requirements during the ideation phase that can be tested in the evaluation phase.

3.1.2 Specification

During the specification phase, the endpoint of the ideation phase is used and further developed upon. Typically for Creative Technology is the high number of low fidelity(lo-fi) prototypes that are built (Mader & Eggink, 2014). Often these prototypes already include some electronics and some sort of microcontroller. These prototypes can be tested with users or by the designer himself. After the testing phase, it is likely that the whole process of making prototypes is repeated, until a final design is reached (Mader & Eggink, 2014). The outcome of the specification phase is a well worked out product idea. Also during the specification phase, functional and non-functional requirements can be drawn up which can be tested in the evaluation phase.

3.1.3 Realisation

The realisation phase uses the specifications that are drawn up earlier to build a hi-fi prototype. The models used for the Creative Technology realisation phase are proven methods from engineering design (Mader & Eggink, 2014). The models are mostly linear which allows the designer to evaluate wrong decisions and go back in the design process (Mader & Eggink, 2014). During the realisation phase, a big part of the evaluation already takes place. If there are any malfunctioning parts of the system these can be eliminated to achieve a higher quality prototype. Eventually, the realisation phase leads to a hi-fi prototype that can be used to test with potential users.

3.1.4 Evaluation

During the evaluation phase, the non-functional requirements from the ideation and specification phase can be tested upon users. The prototype build in the realization phase is used to see whether the product meets the user requirements and if the earlier described problem is solved. The evaluation is also the place to reflect upon the design process.

3.1.5 Overview of the design phases

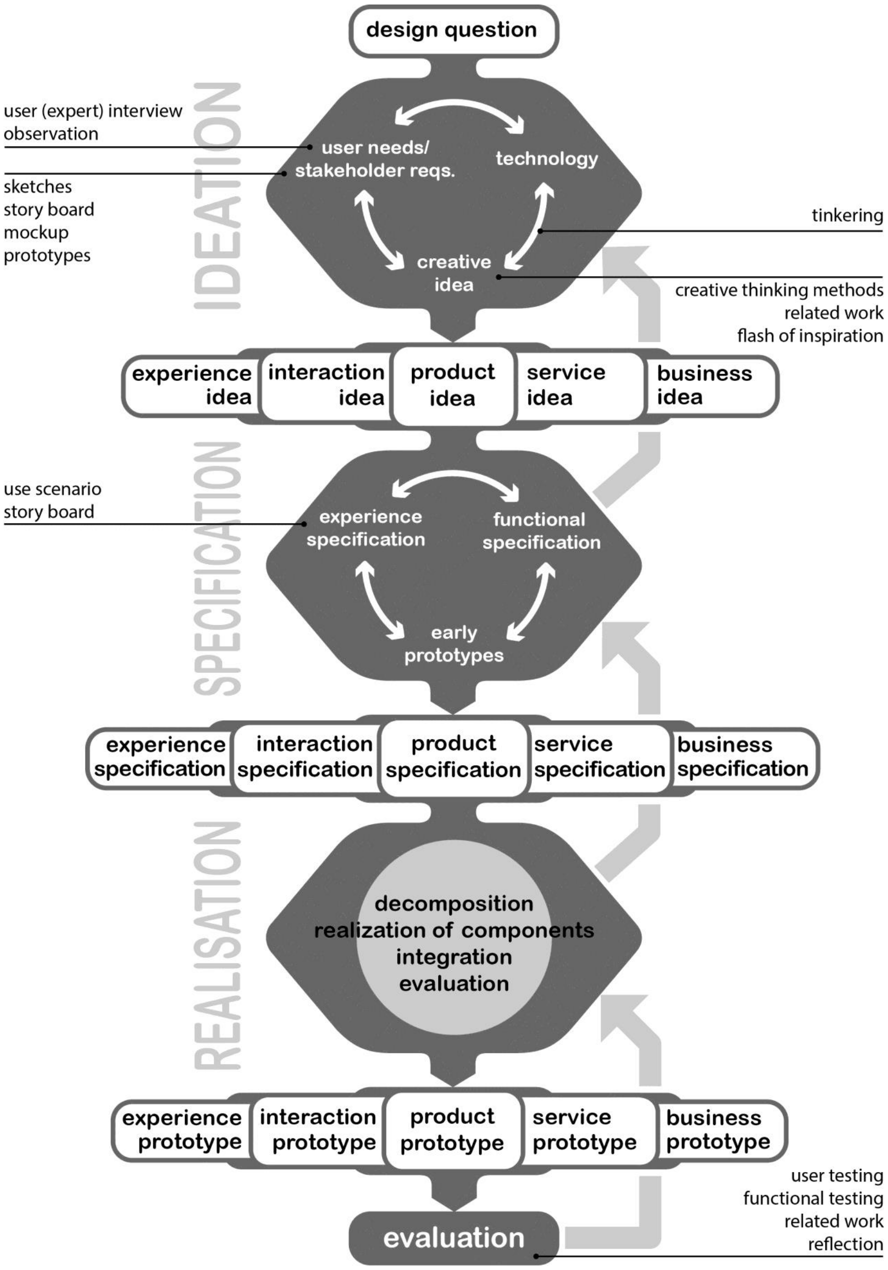

Figure 15: The four design phases of creative technology (Mader & Eggink, 2014)

3.2 Stakeholder analysis

A stakeholder analysis is meant to identify all the parties who affect the project or who are affected by the project. Freeman (1984) describes two types of stakeholders: those who affect and those who are affected by a decision or action. It is important to get an idea of the stakeholders because this can help to see which parties influence the project (Crosby, 1991). The stakeholder analysis also helps to get a better understanding of all the people who have an interest in the project. By acknowledging the people with interest in the project the decision-makers can think of the effects of their decision more accurately. To get a quick overview of the stakeholders they are displayed in a table and a grid matrix. In this grid the interest is displayed on the x-axis and the power is displayed on the y-axis, see figure 16 for how this would work.

Figure 16: Illustrative figure of stakeholder grid matrix (Roseke, 2019)

3.3 People, activities, context and technologies(PACT) analysis

A PACT analysis helps to see how a system should work from a user's view. A PACT analysis consists of four parts that all contribute to the overall image of the system (Nayanathara, 2020). The four parts of the PACT analysis are explained in this section.

3.3.1 People

The characteristics of the people that are going to use the system are explained here. This could be in the form of physical characteristics but also psychological characteristics or social differences can be explained here (Nayanathara, 2020). For some products, the typical characteristics of users can be very important to write down since it might limit the design possibilities. Drawing up personas can help to get a better feeling of a typical user.

3.3.2 Activities

The activities users are likely to perform with the to-be-designed product are elaborated here. Of course, the activities users are going to perform with the product differs a lot. For most users, however, the typical activities are comparable to one another. In the activities section of the PACT analysis, the activities are explained for which the product will be designed (Nayanathara, 2020).

3.3.3 Contexts

The context in which the people using the product conduct their activities is explained here. Most products can be used in a wide variety of contexts. The contexts in which the activities are most likely to be performed are explained here (Nayanathara, 2020).

3.3.4 Technologies

Most products consist of hardware and software components. For the technology section of the PACT analysis, especially the interaction between the user and the technology is important. The technology section explains how the technology will be seen from a user's point of view. Therefore, especially the input and output from the system is explained here (Nayanathara, 2020).

3.4 Group brainstorm strategy

To get more creative it can be beneficial to brainstorm in a group context (George, 2007; Hoever et al., 2012). The method of the group brainstorms used for this thesis is based upon the Walt Disney creative thinking strategy (Dabell, 2018). The main question that will be asked during the group brainstorms is: "How to improve the design of the Modsy controller using the four load reducing categories? (from section 2.1)". Before the brainstorms, the current design, which will be explained in section 4.2, of the controller will be shown and shortly explained. Also, the participants of the brainstorm will be informed about the four load reducing categories.

The Walt Disney creative thinking strategy consists of three phases: the dreaming phase, the realising phase and the criticising phase (Dabell, 2018). The participants of the group brainstorm will be led through this process to eventually come to ideas. The three phases of the brainstorm are explained below.

Dreaming

During the dreaming phase, it is all about generating as many ideas as possible. The viabilities or weak points of the ideas do not matter. An idea will be written down on a paper or whiteboard and a big circle is drawn around the idea. The colour used for writing down the idea also had a meaning during the dreaming phase. During the background research and state of the art four load reducing categories were identified. These four load reducing categories are also introduced to the participants of the brainstorm session. The four categories are assigned a colour that can be used to write down the ideas. See figure 17 for how this will be presented to the participants. An example and a short explanation of the categories are also printed out on paper to help the participants remember what the categories are. See Appendix A for the printed paper.

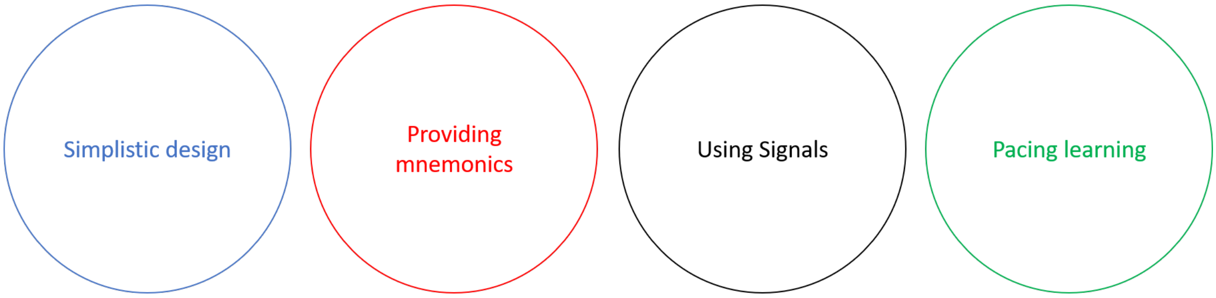

Figure 17: colours according to the load reducing categories

Realising

After a lot of ideas have been generated, the realising phase helps to think about how to achieve the 'dream' from the dreaming phase. Thinking more about practical solutions for the ideas generated in the dreaming phase helps to see which ideas can be executed. Practical solutions for the ideas generated in the dreaming phase are written down on a sticky note. The sticky note will be placed on the circle of the idea from the dreaming phase. It is possible to have multiple sticky notes for one idea.

Criticising

The last phase is about looking for possible vulnerabilities of the ideas generated in the dreaming phase and realising phase. Looking for problems, difficulties and unintended consequences of the ideas helps to see which ideas have a high potential. The vulnerabilities of the ideas are written on a sticky note with a different colour than the ones used for the realising phase. The vulnerabilities are also placed on the circle of the idea from the dreaming phase. It is possible to have multiple sticky notes for one idea.

3.5 User scenario

A user scenario is a method for telling a story about how a user interacts with a product, service or website (Costa, 2020). For this thesis, the user scenario will be used to describe how a regular user would interact with the prototype. In this case, it will be in the form of a short story where a persona uses the system for the first time. His thoughts and actions are written down in the story which helps to get a better feeling of how the prototype is going to be used.

3.6 MoSCoW analysis

A MoSCoW analysis can help to prioritize certain product features (ProductPlan, 2020). The requirements of the system are analysed and based on how important they are for the system scaled on a four-point scale. The term MoSCoW is an acronym of the first letter of each point on the scale.

The M stands for "must-have", these are essential product features. If the eventual product does not meet the requirements from the must-have this is a big loss for the product (Agile Business, 2014).

The S stand for "should-have", these are product features that are important but not vital. Although the should-have requirements will probably be painful the leave out, the product is still a viable solution for the design problem (Agile Business, 2014).

The C stands for "could-have" these are product features that are wanted or desirable but less important. If the could-have requirements will be left out of the product it will have less impact on the eventual solution compared to the should-have requirements (Agile Business, 2014).

The W stands for "won't have this time", these are product features that will not be delivered. Possible "wont-have" requirements might be terminated due to time or money constraints (Agile Business, 2014).

Within this thesis, the MoSCoW analysis also distinguishes between functional and non-functional requirements. Functional requirements(FR) are requirements of the system that can be tested during the realisation phase of the project. The non-functional requirements(NFR) need to be tested with users to confirm whether they are met.

3.7 Use case scenario

A use case scenario describes how a user interacts with the system seen from a system point of view (Larson & Larson, 2004). All the actions a user performs on the system can be described chronologically. To visualise all these actions a diagram can help to get a clear image of what is going on at which moment. In the diagram, multiple user-actions and the responses of the system are visualized. The use case scenario especially helps to get a better feeling of all the possible actions users have. It can help to see how the system should react to different inputs from the user. For this thesis, the use case scenario describes how the system acts based on the earlier described user scenario.

3.8 System Usability Scale(SUS)

A SUS is a common way of testing the useability of a system using a questionnaire. The SUS was originally developed as a replacement for long questionnaires that took a lot of time from the participants. The SUS consists of ten statements that all have to do with one of the following three categories: system effectiveness, system efficiency and user satisfaction (Brooke, 1995). The participants need to rate how much they agree with one statement on a five-point Likert scale (McLeod, 2008). The Likert scale ranges from Strongly disagree to Strongly agree. The ten statements of the SUS can be seen below.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

To get a result out of the SUS, the scores of the ten statements need to be converted to a number. To do so the Likert scale is converted to a number. So if the participants answered a statement with "Strongly disagree", this would be a one, and "Strongly agree" would be a five (Brooke, 1995).

The odd-numbered statements are positive things about the system. So the higher the given score the better the system is. The even-numbered statements are negative things about the system. Here the opposite applies, the lower score the better the system is. To convert these Likert scores to a number, two things need to happen:

- For each odd-numbered statement, the actual score is: $(Likert\ score - 1) * 2.5$
- For each even-numbered statement, the actual score is: $(5 - Likert\ score) * 2.5$

This gives a score from 1-10 for every statement. By summing up the ten statements a score comes out of the SUS. The score can range from 0 to 100 (Brooke, 1995). Research has proven that a SUS score higher than 68 is above average and a SUS score lower than 68 is below average. For this thesis, two SUS scores are compared and thus this number of 68 will not be used.

3.9 Rating Scale Mental Effort(RSME)

A common way to ask participants about the cognitive load of a system is the RSME. Paas (1992) introduced the use of RSME. After the introduction of this scale, it became a widely adopted instrument to assess the mental workload of persons while using a product (Ghanbary Sartang et al., 2016; Widyanti et al., 2013). Participants of a user test are asked to rate the amount of mental effort on a nine-point Likert scale ranging from “Very, very low mental effort” to “Very, very high mental effort”. The scale contains nine anchor points on which participants can rate a system. Figure 18 shows how the rating scale looks like. By comparing the results of two tests, conclusions can be drawn to see which system has a higher cognitive load.

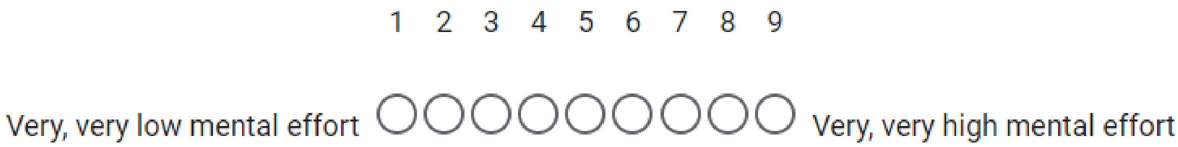

Figure 18: Likert scale for mental effort

4. Ideation

The starting point of the ideation phase in this case is a design question. This design question is: "How to decrease the cognitive load of the Modsy controller?". The ideation phase is meant to find possibilities for answering this design question. The endpoint of the ideation phase is a somewhat vague idea that needs to be worked out during the specification phase. In this thesis, this is called the preliminary concept.

To start the ideation phase a stakeholder analysis is performed. This is an instrument to identify all the people involved in the thesis. Mapping out all stakeholders helps to get a better understanding of the implications of the decisions made during the whole thesis. Before it is possible to think of improvements to the current Modsy system, it is important to have a clear image of how the current system works. Therefore, all the product features of the current Modsy are explained in the product features section of the ideation phase. This system will then be analysed using a PACT analysis. The PACT analysis helps to understand all the things that are important to know about the current product besides the product features. Once the current system is completely explained, it is time to start thinking of possible improvements. This will be done by two types of brainstorms. During these brainstorms, the design space is constantly diverging and converging. The outcomes of the brainstorm are then evaluated into a preliminary concept, which is the starting point of the specification phase. Finally, the preliminary concept is summarised and formed into system requirements. This will be featured in the form of a MoSCoW analysis.

4.1 Stakeholder analysis

The first part of the ideation phase is a stakeholder analysis. How a stakeholder analysis works is explained in section 3.2. The stakeholders are split up into three categories, namely product developers, users and advisers. The stakeholder analysis helps to identify people with interest and influence in the product. First, the stakeholders are presented and a short explanation will be given about their role during this thesis. The stakeholders are also presented in a table and a graph to create a schematic overview. Finally, a conclusion will be given, with points that need to be taken into account during the later stages of this thesis.

4.1.1 The stakeholders

This section lists all the stakeholders and their interests and influence on this thesis.

Product developers

- Weirdly Wired

The team of Weirdly Wired are the product developers of the Modsy. The results of this thesis can be used by them to eventually improve the design of the Modsy. During the process of this research, the Weirdly Wired team will often be consulted to debate ideas. Because the Weirdly Wired team has already done a lot of research about the product, they possess a lot of knowledge and expertise about the product. The interest and influence of the Weirdly Wired team are both high.
Interest: high
Influence: high

Users

- Beginner music producers and performers

The Modsy is meant for music producers that are a bit more experienced. Since the controller is an addition to a music production setup and is quite expensive, Weirdly Wired expects that the controller will be too expensive for beginners. Therefore it is expected that the interest of beginner music producers and performers in the product is lower. It is not beneficial for Weirdly Wired to focus on beginner music producers, and thus the influence of this stakeholder is also low.
Interest: low
Influence: low

- Intermediate music producers and performers

This group has a lot of experience of working inside a DAW and might have already bought one or two other controllers before they will buy the Modsy. To gain more control, intermediate music producers and performers can have a high interest in the Modsy controller. The product design should enable intermediate music producers to use the controller without any problems. The design process is focused on making sure the product fits its target audience. Therefore the influence of this group is also high.
Interest: high
Influence: high

- Advanced music producers and performers

These stakeholders also have a lot of experience of working inside a DAW and might even produce music professionally. Although this group of users probably already owns a lot of hardware, Modsy might still offer them something new. Just like intermediate music producers and performers, this group has a high interest in the product because it offers them more control over their DAW. Since these users are already quite familiar with other hardware tools, it should be easy for them to use the controller. Without having to learn too much, they should be able to get acquainted with how the controller works. The influence of this group is high. The design of the controller should enable advanced music producers and performers to use the controller without too much effort.
Interest: high
Influence: high

Advisors

- Wouter Eggink and Erik Faber

Wouter and Erik are the supervisors of this thesis. They are both professors at the University of Twente and they can help and give advice where needed. By doing so, they have a medium interest and medium influence on the project. These stakeholders also impose boundaries on this thesis. One very important boundary is the time frame in which this thesis should be finished. Over the whole design process, the time constraints have high consequences.
Interest: medium
Influence: medium

4.1.2 Stakeholders overview

This section shows the stakeholders in two types of overviews. First, table 8 presents the stakeholders, their interests and their influence on the project. Secondly, the information from table 8 is presented in a graph in figure 19. In this graph, the interest of the stakeholders is displayed on the x-axis and the influence of the stakeholders is presented on the y-axis.

Stakeholder	Interest	Influence
Beginner music producers and performers	Low	Low
Intermediate music producers and performers	High	High
Advanced music producers and performers	High	High
Wouter Eggink and Erik Faber	Medium	Medium
Weirdly Wired	High	High

Table 8: Overview of stakeholders presented in a table

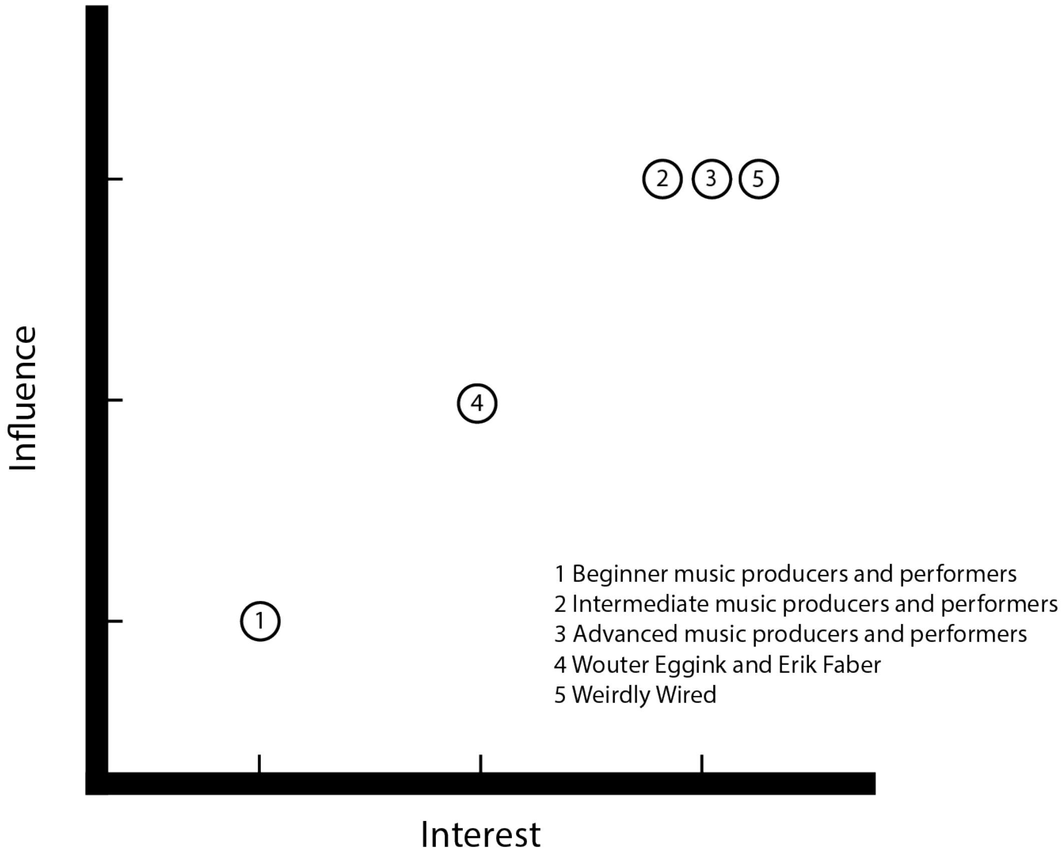

Figure 19: Overview of stakeholders presented in a graph

4.1.3 Conclusions stakeholder analysis

From the stakeholder analysis, there are some important points to take away for the continuation of this thesis. Firstly, it is important to notice that the two main target audiences are intermediate and advanced music producers and performers. Thus the design does not have to be focused on beginner music producers. Secondly, the time frame in which this thesis has to be finished needs to be taken into account. There is not so much time available and this will probably affect the design process. Finally, it is very important to keep the vision from Weirdly Wired clear. The product that will be built can only help Weirdly Wired if the design process stays aligned with their vision.

4.2 Product features

This section explains how the current Modsy system works and what all the features of the complete system are. It is necessary to have a clear understanding of the current system before it is possible to improve it. Modsy consists of two parts, a hardware part and a software part. The hardware part is in the form of a physical controller. The hardware controller is connected to the computer using a USB-C cable. The USB-C cable provides power to the controller and is used to transfer data from the controller to the computer and vice versa. First, the hardware of the controller will be explained.

4.2.1 Modsy controller

The controller is divided into two sections, the play section and the control section. The control section is surrounded by a blue line in figure 20. The control section gives musicians direct control over their DAW. In the play section, all the fun happens, this is where the actual music is made, the play section is surrounded by a red line in figure 20.

Figure 20: Modsy controller with indications for control and play sections

The control section

This section zooms in on the control bar and its functionalities. The control bar can be used for three main functionalities. The first and most important function of the control bar is mapping the controller to a device inside the DAW. This can be done by using an encoder, a pushbutton and two small screens. The encoder, or so-called selector, is used to scroll through the devices within the DAW and is located on the top left of the controller, see the blue section of figure 21. The pushbutton, or so-called mapping button, is used to map the controller to the selected device and is located on the right of the mapping button, see the red section of figure 21. A LED inside the mapping button gives feedback to the user whether the push was registered by the Modsy controller. If users want to map the Modsy to a new device, first they would scroll through their devices by using the selector. Once the wanted device is selected, the user can press the mapping and the Modsy is mapped to the selected device. To give feedback to the user, two small screens are used for this mapping process. These screens are located next to the mapping button on the controller, see the yellow section in figure 21. The first screen is used to show the user which device is being selected within the DAW. The second screen is used to show the user to which device the Modsy is currently mapped.

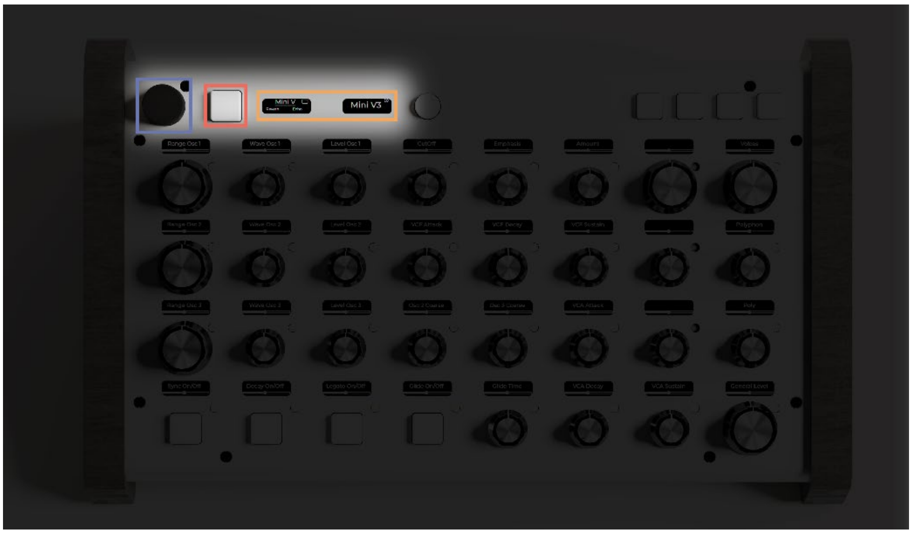

Figure 21: Modsy controller with highlighted mapping section

The second functionality of the control bar is the page button. The page button is a push button that is located directly next to the small screen which shows the device that is being mapped to the Modsy, see figure 22. The page button can be used if a musician wants to control more parameters than 32. Because the Modsy controller has 32 knobs this would be the maximum amount of parameters that can be controlled by the Modsy. Using the page button even more than 32 parameters can be controlled. Once the page button is pressed the Modsy maps itself to the next page of parameters. So the selected device stays the same but 32 different parameters can be controlled. The screen on the left of the page button gives feedback to the user about the currently selected page.

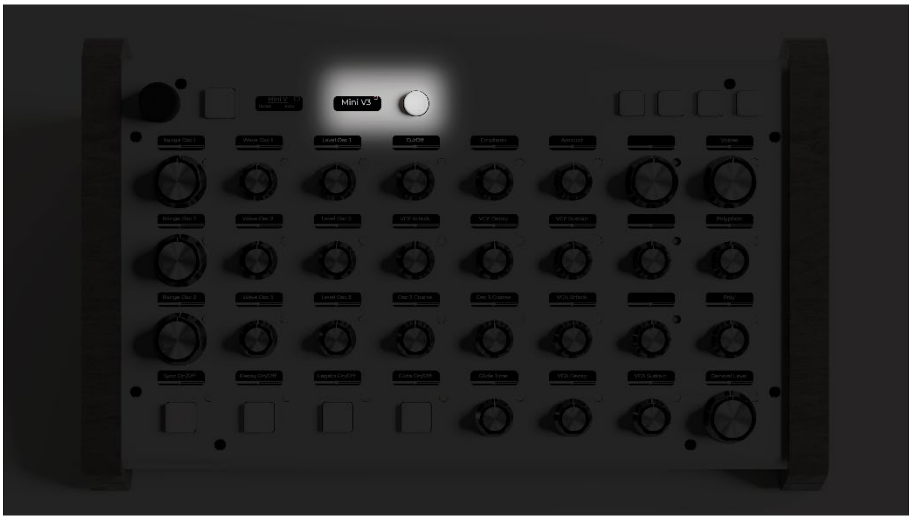

Figure 22: Modsy controller with highlighted page section

The function buttons are four push buttons that are located at the top right of the controller, see figure 23. Users can use these buttons to perform certain actions inside their DAW. Examples of actions that can be triggered by the function buttons are: play/pause, record or undo.

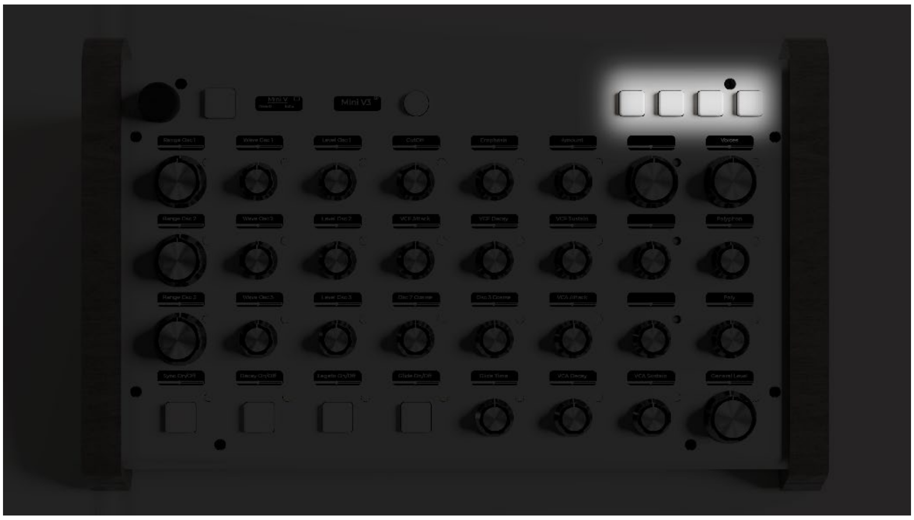

Figure 23: Modsy controller with highlighted function buttons

Play section

The play section of the controller can be used to adjust parameters from the device that is being selected in the control section. The controller has 28 potentiometers and four pushbuttons that are positioned in an eight by four grid, see figure 24. The potentiometers can be used to adjust parameters with an input ranging from zero to 127. The pushbuttons on the controller can be used for parameters that are either on or off(zero or one).

The potentiometers and pushbuttons are all accompanied by a screen and a LED. Both of these features are implemented in the current controller to decrease the cognitive load of the controller. The screen is used to show which parameters can be controlled by the respective knob. So if a potentiometer can be used to control the master volume of a device the screen would have the letters “Master volume” on it. These screens are a great example of the load reducing category *provide mnemonics* to the user to lower the cognitive load.

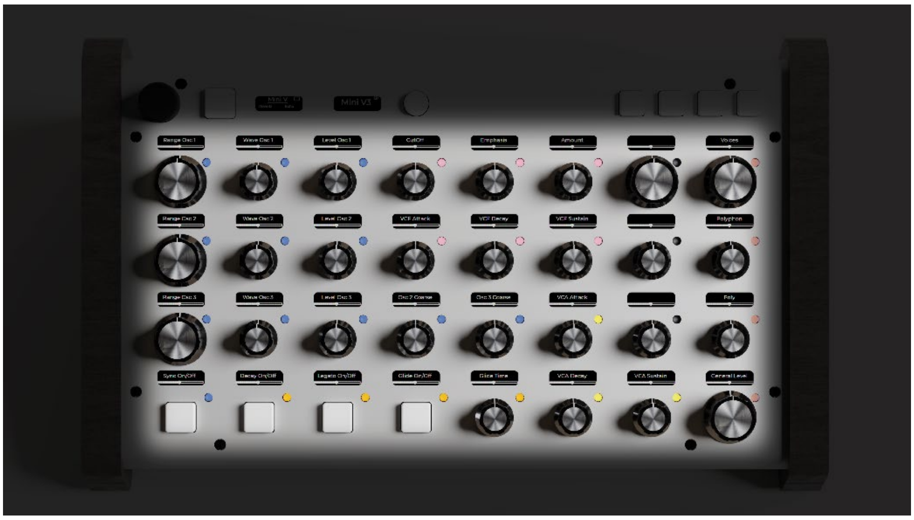

Figure 24: Modsy controller with highlighted play section

The LEDs can be used to make groupings. It is very likely that the device that is being controlled also has some groupings. See figure 25 for an example of such a device. Here there is a clear distinction between, for example, oscillator A (OSC A) and oscillator B (OSC B). It is possible to give all the LEDs accompanying a knob that is used for OSC A a blue colour and LEDs accompanying a knob that is used for OSC B a red colour. This is also beneficial for the cognitive load. The LEDs are an example of *using signals* to decrease the cognitive load.

Figure 25: Serum, an example of a device in a DAW (XFerreccords, 2021)

4.2.2 Modsy software

The software accompanying the controller can be used to adjust the controller to the personal needs of musicians. The Modsy software lives inside a DAW and can be seen as some sort of plugin. Currently, the Modsy software only works within Ableton. Inside the Modsy software, users can determine which parameters they want to have control over on the controller. It is also possible to customize the text that is displayed on the screen and the colour of the LED accompanying the knob. Also, the function buttons can completely be customized in the Modsy software. DAWs have lots of features that can be triggered and users can determine themselves which of these features are controlled using the function buttons. See figure 26 for a schematic representation of how the current Modsy system works.

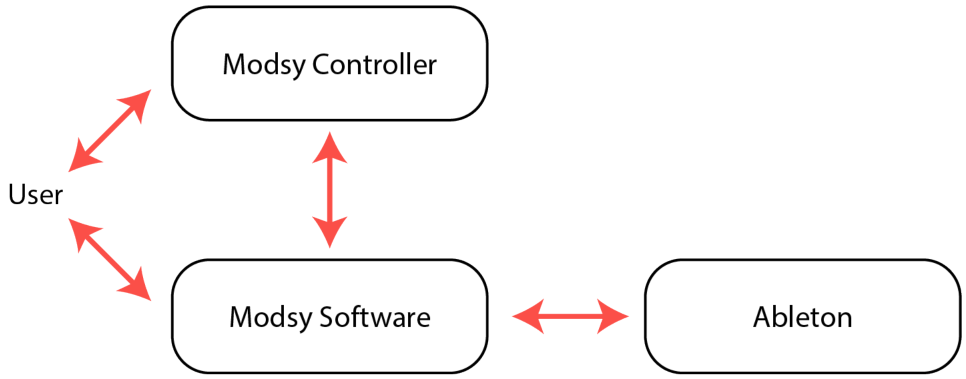

Figure 26: Schematic of Modsy system

4.3 PACT analysis

In this section, a PACT analysis, as explained in section 3.3, of the Modsy system will be performed. Parts of the PACT analysis were made together with two other researchers (Berkenbos, 2021; Mathijssen, 2021). The PACT analysis helps to identify important characteristics of the system and things to take into account while designing.

4.3.1 People

Modsy focuses on advanced music producers and performers. Most of the users will therefore already have some knowledge about their DAW and will probably already own some sort of controller. It is important to know that the typical user of the Modsy probably recognizes often used design patterns, because of their experience with other musical products. One probable reason for them to buy the Modsy is to gain even more control over their music. Because of the price of the Modsy users expect a product that does not have any flaws and is easy to work with. Most users know their way around technology and are not afraid to take on a new challenge. To get more insights on what kind of users the Modsy will have four personas will be described.

Boris, male, professional producer, 28 years old, Amsterdam, DJ/Producer

Figure 27: A representation of the persona Boris (Karras et al. & Nvidia, 2019)

Boris is a 28-year-old DJ and producer. He has an apartment in Amsterdam together with this girlfriend. Boris is signed to an electronic music label based in Amsterdam and can utilize a studio owned by this label. Next to this Boris has different music production equipment in his apartment. This equipment consists of good speakers, a synthesizer and a keyboard. In the studio, he has different synthesizers, drum computers and mastering equipment at his disposal. Boris has been making music for a long time and has always been a digital music producer. Musical concepts are mostly created in his apartment, the finalizing of the concepts into actual songs is done from the studio of this label. Boris his main drivers for music production are income, enjoyment, and emotional release. Boris has signed a deal with his label which states that he has to release at least 1 album each year, and he has been able to deliver on this promise. Boris is very active on social media, mostly for the promotion of his music. Furthermore, Boris has a large social circle within Amsterdam with many friends active in the music industry, this results in a lot of local collaboration and support. Because of his experience with musical technology he knows his way around new technologies quite fast. Commonly used design patterns can help Boris to get acquainted with new technologies quicker. Boris loves devices that are plug-and-play and does not like to do a whole setup before he can use its gear. Since he needs to eventually deliver music for his label he does not want to spend too much time learning completely new features. Boris uses almost all features within his DAW to fully customize his music.

Josh, male, advanced producer, 32 years old, Berlin, software developer

Josh is 32 years old and has been producing music since his childhood years. He started out playing the guitar but later moved on to digital music production in Ableton. He has studied computer science at the University of Delft and now works a full-time job as a software developer in Berlin. He lives alone in a moderate apartment in the city of Berlin. Josh mostly partakes in music production during evenings and the weekend. Josh his main drivers for music production are enjoyment, emotional release, and personal development. He does not release his music often, but occasionally releases some of his work on Soundcloud or his Spotify page. Josh is a member of a Facebook group where he talks with other like-minded musicians. Within this group, he is not very active but will engage occasionally. Josh can finance his musical activities well since he has a stable income. His setup is minimalistic but of good quality, with a proper mixer, good speakers, a MIDI keyboard, and a few instruments. Because of Josh's job, he thinks systematically and generally wants to understand the whole logic behind technologies. He is very curious about how systems work and does not mind spending a whole evening figuring out all the features of a system. Since Boris has been a producer for a long time he mostly uses the original features from Ableton and has a basic routine for creating music. When Ableton has a new updated he is excited to see what new things they have to offer but usually, he does not incorporate these new features in his workflow.

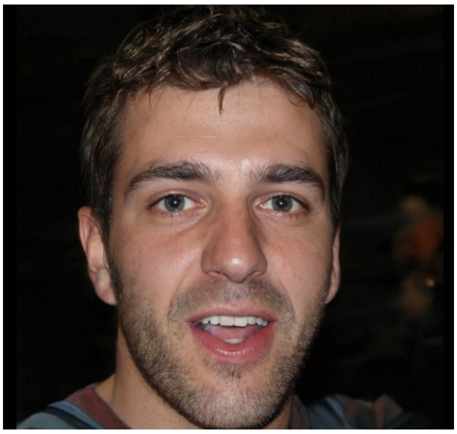

Figure 28: A representation of the persona Josh (Karras et al. & Nvidia, 2019)

Nina, female, advanced producer, 21 years old, Utrecht, Sound Design student.

Figure 29: A representation of the persona Nina (Karras et al. & Nvidia, 2019)

Nina is a student sound design at the HKU in Utrecht. In this program, she specialises in sound for video content. She works part-time in a sound design agency and makes music herself as well. She has a lot of contact with young producers and DJ's and is trying to build a portfolio for herself, which includes being active on social media and doing promotional work. Since she wants to distinguish herself from the crowd she is always looking for creative ways to make music. This includes utilizing old instruments or sampling old records. Her budget is not big which means she is very particular about what she invests her money in. When she buys something new she needs some time to adapt to the new technology. She likes to work with her computer but is not so affiliated with hardware. Because of her minimal budget, she is willing to spend a bit of time learning the new technology.

Robert, male, advanced producer and performer, 56 years old, Hoogeveen, Electrical engineer

Robert is 56 years old and has been performing music for a long time. He started in bands around the age of 16 and has been making music ever since. This passion has progressed parallel to this professional career in engineering. The music Robert produces is mostly psychedelic rock and synth wave music. Roberts main drivers for music production and performance are emotional release and enjoyment. He makes a small amount of money from his music performances, but this is not the main drive to make music. Robert records music with a band and uses different instruments in this process. Robert has always been very interested in the mixing and mastering of the audio and has a small mixing studio in his garden. This studio holds a variety of guitars, effects, mixing boards, and some synthesizers. Robert mostly uses Ableton for the finalizing of his recordings and occasionally uses software instruments or effects in the process. Robert usually understands the hardware of devices easily because of his background as an electrical engineer. If the software accompanying the hardware is difficult to understand Robert is quickly turned down of a device. He likes devices that look like synthesizers from the days he was younger.

Figure 30: A representation of the persona Robert (Karras et al. & Nvidia, 2019)

4.3.2 Activities

This section describes with what kind of activities the Modsy controller will be used. There are three types of activities in which the Modsy serves as a tool for musicians. While using the Modsy controller it is a common practice to switch between the type of activity that is being done. It is very likely that while creating a song the described activities are performed multiple times.

Sound design

The Modsy controller is perfectly suited to design all different kinds of sounds. Because the controller has 32 parameters that can be mapped to Ableton, musicians have all the control they need over their software. Sound design especially consists of listening to the same sound over and over again and tweaking small parts of it until it is perfect.

Sound automation

Sound automation enables musicians to let the computer automate certain parts of music production. Recording in a DAW how a parameter should behave over time is called automation. An example of a parameter that can be automated is the volume of a certain sound. If musicians want the volume of the guitar to change over time this can easily be done by automating the guitar volume. See figure 31 for an example of how this would look like in a DAW.

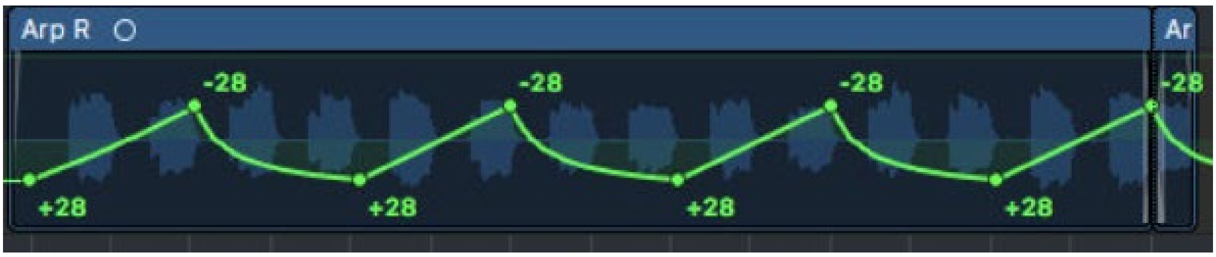

Figure 31: An arp that is automated in a DAW (IZotope, 2018)

Ableton control and automation

Modsy offers a lot of control within Ableton. By using an encoder it is possible to switch between the selected device. If a musician is using a certain device but then wants to switch to a different Ableton device this can be done by using the encoder. This process is both relevant for musicians who are designing a certain sound but it can also be very useful for recording automation.

4.3.3 Contexts

This section looks at in which contexts the Modsy especially is going to be used. Modsy can be used in two main contexts. One user of the Modsy can use the Modsy in both scenarios.

Music producing

Music production often happens in a studio where producers lock themselves in for a couple of hours to create new music. While creating music, often producers listen to the same piece of music multiple times to try out new things.

Music performing

When performing music, Modsy can also be used. Music performance can perform their songs or songs of other producers. The main usage of Modsy for performance would be to be able to live tweak sounds and effects to adjust the music that is being played.

4.3.4 Technologies

Modys is a combination of hardware and software. The hardware communicates with the software and the two combined offer a very strong package. This section especially looks at the inputs and outputs of the system. A more elaborate explanation of all the features of the Modsy has already been given in section 4.2.

Modsy controller

The hardware of Modsy is a physical controller with buttons, screens and LEDs. The input of the user to the controller is by turning the knobs and tapping the buttons. The controller offers feedback to the user by use of screens and LEDs. The screens on the controller represent the functionality of the knobs to the user. On the controller, there are also LEDs present. These LEDs can be used to make groupings. Another output point of the controller is the Modsy software.

During this thesis, the Modsy system will be changed to decrease the cognitive load. The prototype that will be developed is also a technology with which users of the Modsy will interact. How this technology is going to look and in which way it will interact with the user will be explained in the coming sections.

Modsy software

The software of Modsy lives inside a DAW and can be used to change the physical controller. The controller gives direct input to the software. The values of the buttons are communicated with the software which then communicates this with the DAW. Users can communicate with the software by using the graphical user interface. The input of the software is thus the Modsy controller and the input directly from the user communicated via mouse and keyboard. The output of the software is a signal that goes to the controller and the user interface.

4.4 Brainstorm

Now that a clear image is created of how the current Modsy system works, it is time to look into how this system can be improved. The first step into creating a new idea for the Modsy controller is done by brainstorming. First, two group brainstorms will be performed to generate a lot of ideas, this is a diverging phase. After the two brainstorms, the results will be analysed and rated on the feasibility of an individual brainstorm. During the individual brainstorm, the main goal is to converge all the ideas generated during the group brainstorm into a more or less feasible concept.

4.4.1 Group brainstorms

Both group brainstorms are performed with two participants excluding the researcher. The researcher does not actively participate in the group brainstorm. The participants of the brainstorm are led to the process of the Walt Disney brainstorming technique as explained in section 3.4. To explain the current concept a picture of the current Modsy design is presented to the participants, which can be seen in Appendix B. The participants are also informed about the four load reducing categories: *using signals, minimalistic design, provide mnemonics* and *pacing learning*. Every load reducing category and an example of the category is printed out on paper to help the participants remember what the categories are. How these papers look like can be seen in Appendix A.

Group brainstorm outsiders

First, a group brainstorm with outsiders helps to get completely new ideas. The participants that were selected for this brainstorm were two persons with a basic understanding of the Modsy, however, who do not know all the ins and outs. It was convenient that the participants already had a basic understanding of the Modsy since therefore not everything needed to be explained. However, it is also good that they do not know every detail about the system. Since they do not know all the limitations of the system, it is easier to create very outstanding ideas. In Appendix C pictures of the three different stages from the brainstorm can be seen. Based on how relevant the ideas are related to the cognitive load, the most interesting ideas of this brainstorm are summarized below.

Fewer Inputs
When having fewer inputs the cognitive load of the controller will likely go down. This concept uses the load reducing category *minimalistic design*.

An AI which helps the user with mapping the parameters
This AI would need to recognize where the user is in the progress of making music and with that the hardware controller would adapt itself to the user needs. This way it is possible to make certain 'more important' parameters stand out of other parameters. This helps the users by giving signals to which parameters are probably the most important. This concept uses the load reducing category *using signals*.

Possibility to put the hardware together yourself
If a user can put the hardware together and determine how and where certain elements are placed the user can learn how the controller works in smaller steps. Users would be able to determine how the controller is going to look and use their logic for putting it together. This concept uses the load reducing category *pacing learning.*

Motorised faders or potentiometers
Motorised faders or potentiometers can be used to make the hardware always look the same as the software. With the current Modsy concept the hardware is not always synchronized with the software when a new device is mapped to the controller. By using motorised faders or potentiometers this problem would be solved since they can just slide themselves to the correct position. This concept uses the load reducing category *provide mnemonics.*

Use former knowledge of the software
Since advanced and professional music producers and performers are the main target audience of the Modsy controller they will have some knowledge of the music software. If the hardware uses concepts from the software the users will have pre-knowledge which makes it easier to understand the controller. This concept uses the load reducing category *pacing learning.*

Group brainstorm Weirdly Wired team

The second group brainstorm that will be performed is with the Weirdly Wired team. The Weirdly Wired team consist of three persons including the researcher. The researcher will not participate actively. So the brainstorm will be done by two persons. The Weirdly Wired team has a lot of technical knowledge and knows all the ins and outs of the Modsy controller. The ideas generated during the Weirdly Wired brainstorm might be less outstanding however the ideas might be more feasible and relevant for the controller. Based on how relevant the ideas are related to the cognitive load the most interesting ideas of this brainstorm are summarized below. In Appendix D the pictures containing all the ideas can be found.

Modsy virtual reality experience
Using virtual reality would offer a lot of possibilities to decrease the cognitive load. Within the virtual reality experience, it is possible to use a lot of the load reducing categories.

Form/colour of the input knob is coupled to a certain type of parameter
A lot of VSTs have similarities between them which can also be used for the hardware of the controller. If users can use the same form or colour for comparable parameters it is easier for users to know which parameters they are adjusting. This concept uses the load reducing categories *using signals* and *provide mnemonics.*

Modular system that can be elaborated
With a modular system, users can determine themselves what kind of input they use for which parameter. It might be possible to determine where pushbuttons, potentiometers and faders are on the controller. This concept uses the load reducing categories *using signals* and *pacing learning.*

Make sections in the controller with form and colour
By dividing the controller into different sections users are hinted to which parameters do what. Here it is also possible to use the former knowledge of the users to make the groupings. If in the software certain elements are grouped they can also be grouped on the hardware. This concept uses the load reducing categories *using signals* and *pacing learning.*

4.4.2 Individual brainstorm

The goal of the individual brainstorm is to analyse the group brainstorm and use the data to design a concept that can be worked out. There were a few things that played a role during the brainstorm. Firstly, the vision of Modsy and Weirdly Wired has a big impact on the decision of where the new product needs to go. As explained in section 1.2.2, Modsy tries to add an analogue feeling to digital tools. A lot of the ideas generated during the group brainstorms did not follow this concept. Concepts like “a VR experience” or “an AI that helps the user”, are eliminated because they are not aligned with the aim of the Modsy and the principles that Weirdly Wired stand for.

Another thing to take into account while designing the prototype is that the main functionalities of the Modsy should be kept. This means that the Modsy itself should maintain its functionality, and for that, it is not possible to simply remove some of the input parameters to reduce the cognitive load.

The last thing that heavily limits the design space for this thesis, is that the Modsy itself already has some excellent solutions to lower the cognitive load. The sole purpose of the LEDs is to indicate groupings to help the user remind which parameters are paired. This makes use of the cognitive load reducing method *using signals.* Another great example of reduction of cognitive load is the screens. The screens make use of the load reducing method *provide mnemonics*. Since the Modsy controller already applies these two load reducing categories, it would probably be more beneficial to choose other load reducing categories. Therefore, concepts that are especially about the categories, *provide mnemonics* and *using signals* might have less impact on the cognitive load.

To illustrate all these considerations a word web has been made. In this word web, the four load reducing categories are presented in four colours. Ideas that were generated during group brainstorms are visualized and some of them are more explicitly explained. In the word web, arrows indicate a relation between the idea and a certain category.

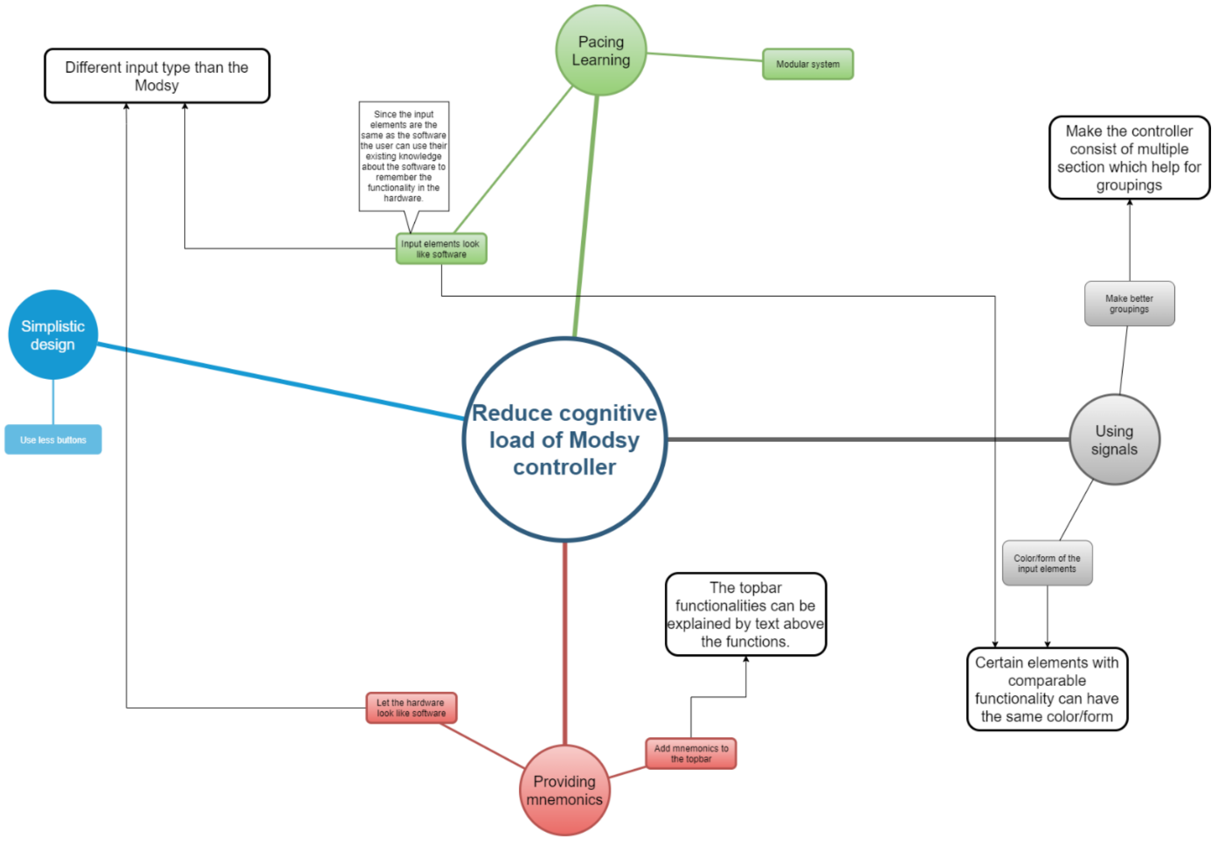

Figure 32: Word web of individual brainstorm

Looking at the word web in figure 32, four ideas stand that could help to reduce the cognitive load of the controller. The ideas that are connected to the category *using signals* are both possibilities to make groupings. Since the Modsy already has LEDs to indicate groupings it is questionable whether these ideas can reduce the cognitive load of the controller. The idea that is connected to the category *providing mnemonics* seems like an idea that could help the Modsy controller. However, for this idea, it is questionable whether it would result in a reduction of cognitive load. Adding more text makes the design less minimalistic which could increase the cognitive load. Also, the topbar functionalities do not have much to do with the music itself. They are especially for controlling functionalities inside the DAW.

The last concept "different input types than the Modsy" makes use of two categories: *providing mnemonics* and *pacing learning*. This concept shows potential to reduce the cognitive load of the Modsy controller and is, therefore, the chosen idea to work further upon. The next section explains how this concept works, and how the cognitive load of the Modsy can be decreased by using this concept.

4.5 Preliminary concept

The current idea is a controller that can be used in addition to the original Modsy controller. The new controller should work very well alongside the original controller and therefore it would make sense to use several concepts from the original controller again. The strong points in terms of cognitive load from the original controller can therefore be implemented in the new controller directly. The additional controller should have the same type of screens to provide mnemonics to the users about which parameter can be controlled with that input. Also, the LEDs to indicate groupings should be implemented in the same way. See figure 33 for a schematic of how this system would look like. Although this figure does not include the LEDs and screens, it does help to illustrate the system. In figure 33 faders are used as input types to illustrate how this concept works. However, more research is needed to confirm whether faders are the correct type of input. In section 5.1 a VST analysis helps to determine which kind of input type and how many inputs seems appropriate.

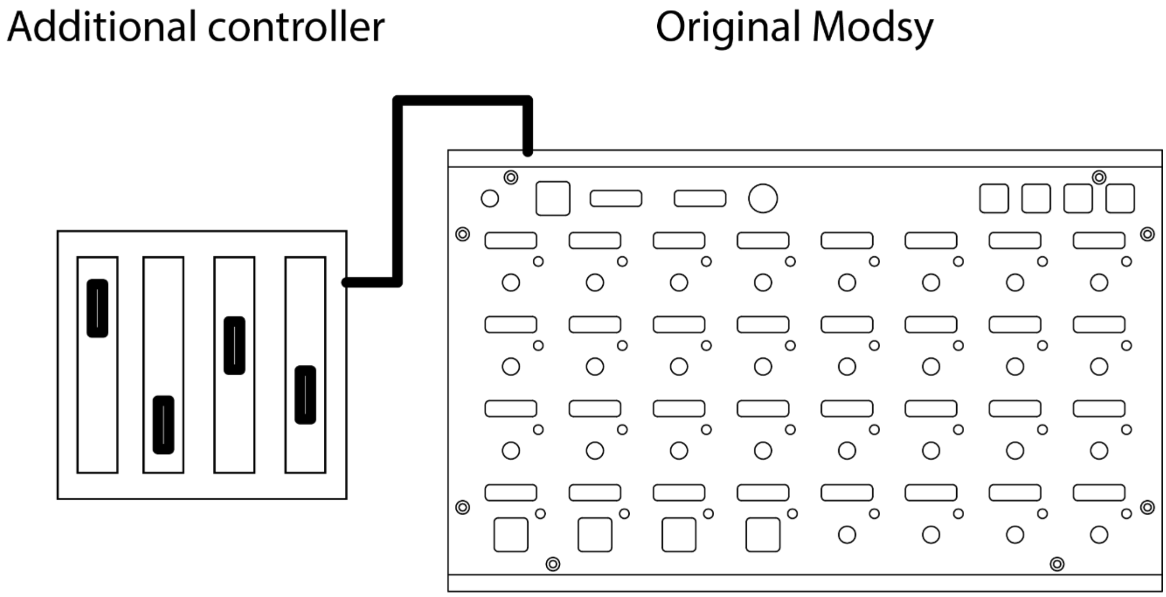

Figure 33: Preliminary concept of the additional controller

A controversial point of having an additional controller is that the total amount of inputs will increase. A positive effect of adding more controls is that this gives musicians, even more, control over their DAW. This might enable musicians to put more creativity, more expression, and more fun into their music production. However, the main goal of this thesis is to decrease the cognitive load of the Modsy controller and it seems like adding more controls does not contribute to this. Looking at the cognitive load design rules from section 2.1.2, adding more controls would go directly against the guideline *minimalistic design.* However, this concept does contribute to several other design guidelines that do contribute to lowering the cognitive load.

This concept of an additional controller especially applies the load reducing method *pacing learning*. By using different input elements than the current design of the controller, users can use their prior knowledge of the instrument or effect to understand the hardware. If, for example, an instrument uses faders for certain control elements it would make sense to also use faders on the hardware for this. The main target audience for the Modsy controller is intermediate and advanced music producers or performers. These users have some knowledge about their DAW which can be used to make the Modsy controller easier to use. If a musician wants to adjust a certain parameter, he might think about the software and how he would adjust this parameter here. Using his knowledge of the software it is easier to remember where this certain parameter is on the hardware. This is also why the new controller should have a different type of input than the original Modsy controller. Which type of inputs should be used will be determined during the specification phase.

A second possibility of how the additional controller can reduce the cognitive load is to use commonly used design patterns for instruments and effects on the hardware. An example of this is that a lot of instruments use faders for the attack, decay, sustain and release (ADSR) of for instance a filter. The Juno-106 (see figure 34) has, for example, two envelopes with an ADSR. On the instrument, the ADSR can be controlled by faders. In figure 34, the first envelope is indicated with a blue arrow and the second envelope is indicated with an orange arrow. One of the main reasons why faders are used for this is that in this way the faders look like the waveform of the sound. If the ADSR is always mapped to the input parameters on the additional controller, users instantly know where they can find the ADSR and thus the cognitive load is lowered. Using faders for the ADSR also comes with the benefit that musicians can instantly see the waveform of the sound on their physical controller. Using potentiometers the waveform is not as graphically as with faders. Although this does not contribute to lowering the cognitive load this might contribute to increasing the overall experience of using the Modsy.

Another possible improvement of the new concept compared to the original controller, is that it is possible to make two very distinct groups of controls. If a certain instrument or effect has two very distinct types of parameters, this is easily imitated on the hardware by having an additional controller. One of the two groups can be mapped to the additional controller and the other group of parameters can be mapped to the original controller. Using groupings makes use of the design guideline *using signals.*

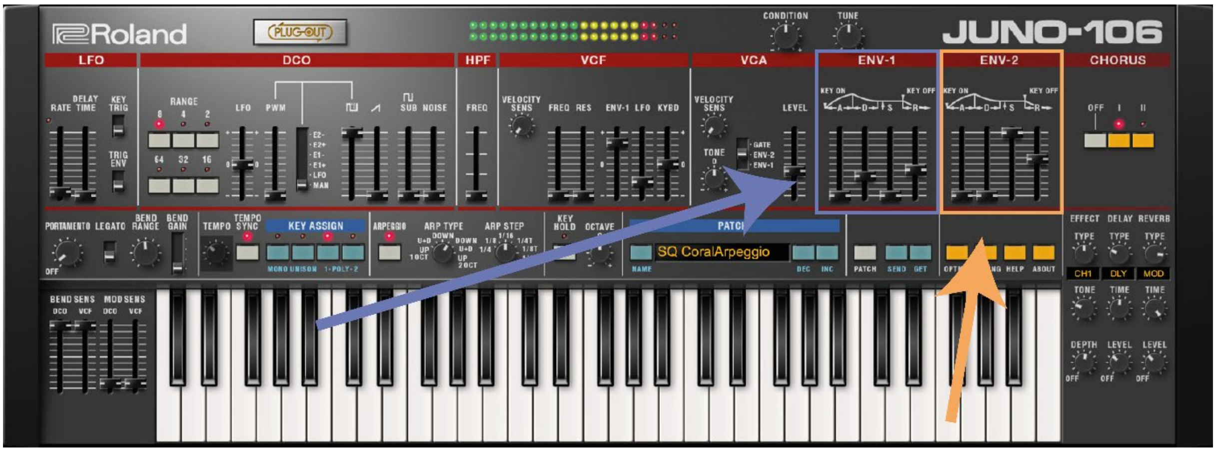

Figure 34: Juno-106 VST

4.6 User scenarios

As part of the ideation phase, this section walks through a possible usage of the new system through the eyes of a user. This helps to get a better understanding of the user interactions and experiences with the new system, as explained in section 3.5. In this case, the persona Boris, from section 4.3.1, will use the Modsy and the additional controller to produce some music. This section is written from the eyes of Boris and thus the "I" person refers to him.

I just came cycling from home to my studio. When walking in the hallway of the building where my studio is based, I already hear the sounds made by my fellow producers. I walk to my studio quickly; I am hyped to make some music! Today I feel like making a summer-vibe tune and I know just the synthesizer that fits this sound. Although I do not possess this physical synthesizer, I do have a very good software copy of the physical instrument.

I quickly startup my computer and launch Ableton. When using a software synthesizer, I always use my Modsy controller to gain more control over my synthesizer. I do not really enjoy twisting software buttons, although I have used this synth for years without using any physical device. To start and get into the feeling of the music, I just want to play around with the sounds of the synthesizer. This is exactly what I enjoy the most about working with the Modsy. Just play around a bit, to create a new sound and music line, that I eventually can record by using Ableton. Today I am extra excited to use the Modsy since yesterday my new device arrived. This device has different types of inputs to add to the Modsy controller which should make it easier to use... Well, we will see about that...

First I plug the additional controller in the original Modsy. I directly notice that the LEDs and screens blink: good news! It works! Since I already know how to use the original Modsy controller, I don't think I need to follow the Modsy tutorial again. The first thing I normally do when using the Modsy is just map it to my Juno-106, which is the synthesizer I love the most. But who knows, this device might work out of the box...? Thus, I select the Juno using my original Modsy controller and press the mapping button on the Modsy. I see the screens and LEDs changing and directly notice a different mapping is loaded than normally. The ADSRs of both of the envelopes of the JUNO are mapped to the new controller, interesting...!

Let's make some music! At the start, I need to figure out how it works, because my ADSR is not on the original Modsy controller anymore. But I enjoy how easy it is to adjust. This external device makes more sense! When I was using the software without this new part, I always had to use faders for the ADSR. I love how this additional controller offers the opportunity to work with faders on the hardware. I familiarize myself quite quickly with the new setup. I feel how it adds to my creativity and it is convenient to work with! While using the new controller, I tinker on the summer-vibe I was aiming for and experience the flow in which this new controller brings me. I am satisfied with its possibilities!

To enrich my summer-vibe tune I also need some sounds from my Diva synthesizer. I select my Diva synth and press the mapping button on the Modsy controller once again. I notice that the Modsy maps itself, and the additional controller has the ADSRs of the Diva. Wow, this thing works well! I start playing with the Diva instantly and notice that knowing where the ADSRs are is very *convenient. Since I just used the Juno with the ADSRs on the additional controller it is easy to remember where I can find them this time.*

This is where the user scenario stops. It seems that the additional controller can help a person like Boris very well.

4.7 System requirements

To summarize the preliminary concept system requirements are drawn up. These system requirements are easy to check afterwards and help to keep a clear image of what the current idea is. The system requirements are made using a MoSCoW analysis as explained in section 3.6. “The controller” in this case means the additional controller.

4.7.1 Must have

- FR1: The controller can be used in addition to the original controller.
- FR2: The controller must work within Ableton Live.
- FR3: The new controller must use different types of input elements than the original controller.
- FR4: The controller must have some way of providing mnemonics for the input parameters.
- FR5: The controller must have some way of indicating groupings.
- FR6: The controller must have good communication with the original controller.
- NFR1: The use of the controller must reduce the cognitive load for the Modsy.
- NFR2: The use of the controller must help musicians to make music in a more fun, creative and expressive way.

4.7.2 Should have

- FR7: The controller should have a casing around it to support all the elements inside it.
- NFR3: The input elements of the controller should not have any noticeable delay.
- NFR4: The use of the controller should offer musicians more control over their software.

4.7.3 Could have

- FR8: The controller could have some way of synchronizing it with the software.
- FR9: The controller could have the possibility to be used standalone.
- FR10: The controller could have some way of putting it at an angle to be more pleasant to use.

4.7.3 Won’t have this time

- No “Won’t have this time” requirements

5. Specification

During the specification phase, the preliminary concept from the ideation phase will be used. The concept of the additional controller will be elaborated further upon and details about the controller will be worked out. To start, the type and amount of inputs will be determined. This will be done by looking at the software program that forms the basis of digital music production, a VST. Once this has been established the layout of the controller will be worked out. To get a clear image of how the new system should work the system architecture will be explained. Using the architecture the system requirements are once again listed. These system requirements are the starting point for the evaluation phase.

5.1 VST analysis

In this section, the type of input that the new controller should have and the number of inputs will be determined. To do so popular VSTs will be analysed, for an explanation of this term see the glossary. The goal of the additional controller is to make the hardware look more like the software. This way the musicians can use their knowledge of the VST to their advantage. Therefore, it makes sense to have a critical look at these instruments or effects before determining which kind of inputs and how much of these inputs should be on the controller.

In total 19 VSTs are analysed which are the top VSTs according to two blogs (LANDR, 2020; Production Music Live, 2018). By looking at the VSTs there are only three types of input: faders, turning knobs and on/off buttons. The original Modsy controller has potentiometers to manipulate turning knobs and pushbuttons to manipulate on/off buttons. However, currently, the sound can only be controlled by mapping them to a potentiometer. This causes that the fader on the software needs to be controlled by a potentiometer on the hardware, which is not optimal. Therefore, the new controller should have faders as input type.

To determine how many faders the new controller should have, a more in-depth analysis is performed. During this VST analysis, the goal is to find a percentage of how much of all the input parameters are faders on the software. Since the goal of the new concept is to make the hardware look like the software, it would make sense to use this same percentage on the hardware. By using the percentage of faders on the software, the number of faders on the hardware can be calculated.

For all the analysed VSTs, table 9 shows five characteristics. The first column shows the name of the VST. The second column is about whether the VST has a fixed layout. If the VST does not have a fixed layout, the configuration of the VST depends on the user which is why these VSTs are not taken into consideration for this analysis. The third column is about whether the VST has faders at all. If the VST does not have any faders at all, the percentage is set at 0%. The fourth column shows the total amount of inputs of the VST and the fifth column shows the number of faders of these inputs. The last column shows the total percentage of faders in the VST. This is calculated in the following way: $\frac{\#Slider}{\#Total\ inputs} * 100\%$. The last row of table 9 shows the average percentage of faders of the analysed VSTs.

VST name	Fixed layout	Fader	Total inputs	Fader	Percentage
Lethal Audio	No				N.A.
Omnisphere	Yes	No			0.0
Massive	Yes	Yes	39	11	28.2
Serum	Yes	No			0.0
FM8	No				N.A.
Juno	Yes	Yes	80	28	35.0
Emulator II V	Yes	Yes	38	11	28.9
Vocoder V	Yes	Yes	49	18	36.7
OB-Xa V	Yes	No			0.0
Jup-8 V	Yes	Yes	78	26	33.3
Stage-73 V	No				N.A.
Kontakt 6	No				N.A.
Reveal Sound Spire	No				N.A.
u-he Hive	Yes	Yes	62	16	25.8
REFX Nexus2	Yes	No			0.0
Spire	Yes	Yes	54	10	18.5
u-he Diva	No				N.A.
Reaktor 6	No				N.A.
Sylenth1	Yes	Yes	38	15	39.5
					20.5

Table 9: VST analysis

Looking at table 9, it turns out that about 20.5% of the total amount of inputs from the analysed VSTs are faders. The current design of the Modsy controller has 32 input parameters. The new design will be used in addition to the Modsy. The analysis revealed that it would make sense to have about 20% faders. By adding eight faders to the 32 input parameters of the Modsy controller, the total amount of faders would be 40. In this case, eight faders are 20% of the total amount of input parameters the Modsy offers, see equation 1. Therefore the controller should have eight faders.

$$\frac{8}{32+8} * 100\% = 20\% \qquad [1]$$

It is important to notice that eight is not the maximum amount of parameters that can be mapped to the additional controller. On the original Modsy, the page button enables musicians to map more parameters to the controller than 32. By pressing the page button the next page of parameters is loaded on the Modsy which gives musicians control over 32 more parameters. Musicians can decide themselves how many pages with parameters they want to have. The additional controller will use this same concept. Once the page button on the original Modsy is pressed, eight new parameters can be loaded on the additional controller. This way it is possible to have control over more than eight parameters.

To conclude, the additional controller should have faders as input type to mimic the software. The most appropriate number of faders is eight, this way the hardware looks the most like the software in terms of input parameters. This number is difficult to determine and it could be possible that more research would help to choose a more appropriate number. However, as section

4.5 already explained, it would be very logical to map two times an ADSR to the controller. Since two ADSRs have eight parameters it seems logical to make the controller have eight faders.

5.2 Controller layout

Now that the amount and type of inputs are determined the layout of the controller can be made. The layout of the controller is very important for the cognitive load. To think about how to apply the load reducing categories the best, lo-fi prototypes are made in Adobe Illustrator. In figure 35 and figure 36 these prototypes can be seen. The prototypes from figure 35 show three possible layouts for the controller. The advantage of prototypes A and B from figure 35 is that they both have two distinct groups of faders. In prototype A, two groups can be made by using the two vertical strips. In prototype B this is also possible however now the groups are horizontal. The advantage of prototype C is that the original Modsy controller also has a grid with eight parameters in the x-direction. The disadvantage of prototype C is that it is more difficult to make two distinct groups. Looking at VSTs it is very likely that the ADSR will be mapped to the controller, as explained in section 4.5. Just like the Juno 106, there are a lot of VSTs that have two ADSR sections and thus it would make sense to divide the controller into two groups. Most VSTs indicate the ADSR with four faders that are next to each other. To make the controller look most like the software layout B from figure 35 makes the most sense.

All the prototypes from figure 35 also have a control bar at the top of the controller. The advantage of having the control bar is that it is possible to completely control the music using the additional controller alone. However, since the additional controller will always be used in parallel to the original Modsy controller the control section is not necessary. Adding more buttons will most likely increase the cognitive load since the design will only be made more complex. Following the load reducing category *minimalistic design,* the cognitive load will be lower without the control section. Therefore figure 36 shows the layout without the control section, this will therefore be the layout of the additional controller.

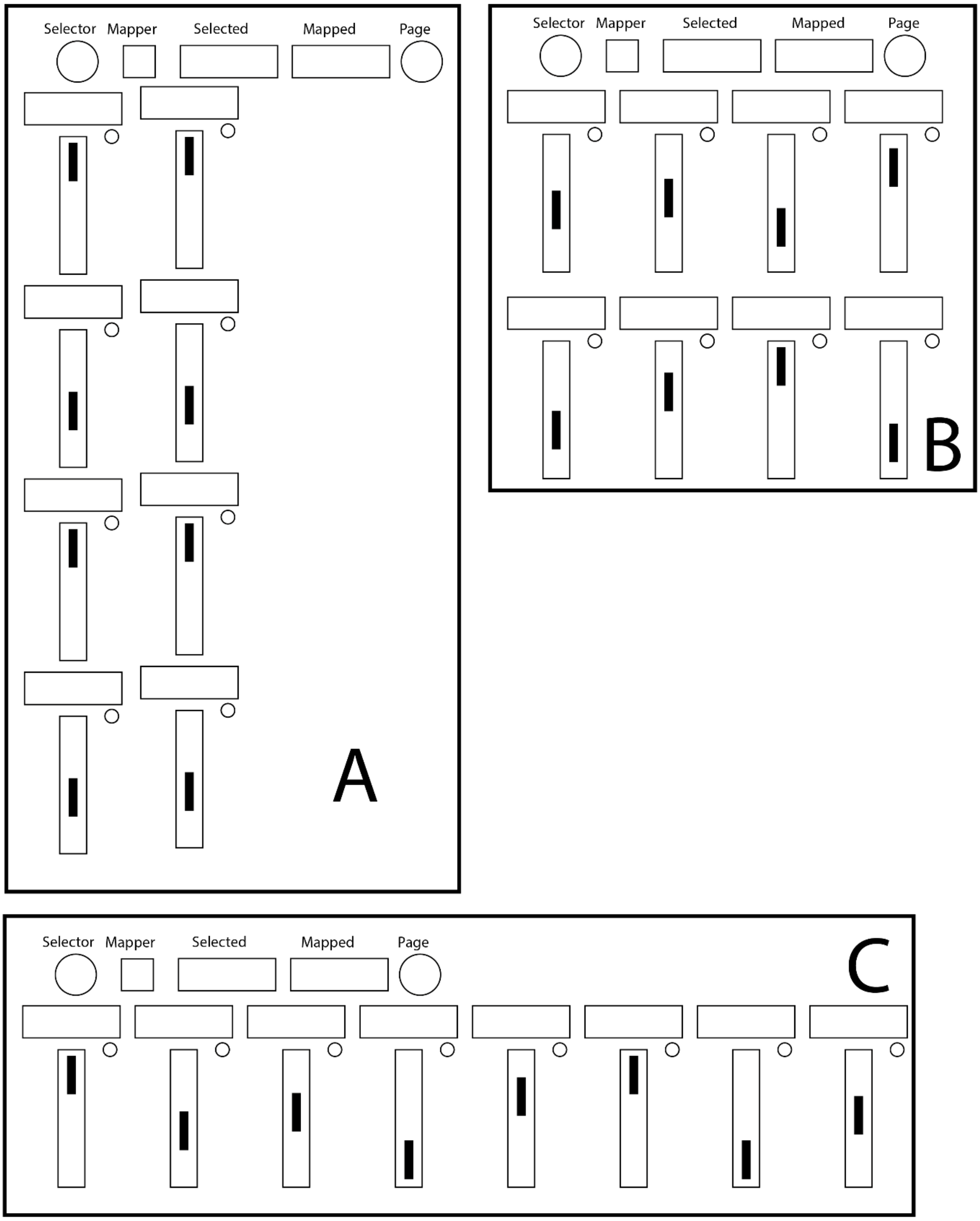

Figure 35: lo-fi prototypes of the layout of the additional controller

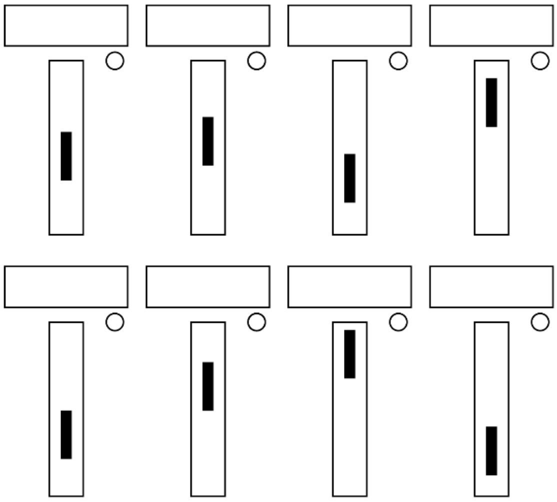

Figure 36: Layout of the additional controller

5.3 System architecture

This section will give an overview of how the system should work and how the new controller should work in combination with the current Modsy. Ideally, the additional controller can be connected to the original Modsy. This connection would then power the additional controller and can be used for the transfer of data between the additional controller and the original Modsy. The information that needs to be transferred to the fader controller is the colours of the LEDs and the names that need to be displayed on the screens. This information comes from the computer and will be passed through to the additional controller. The data that goes back to the original controller is only the values of the faders. These values will in their turn be passed on to the computer. Figure 37 gives a schematic overview of how the whole system operates.

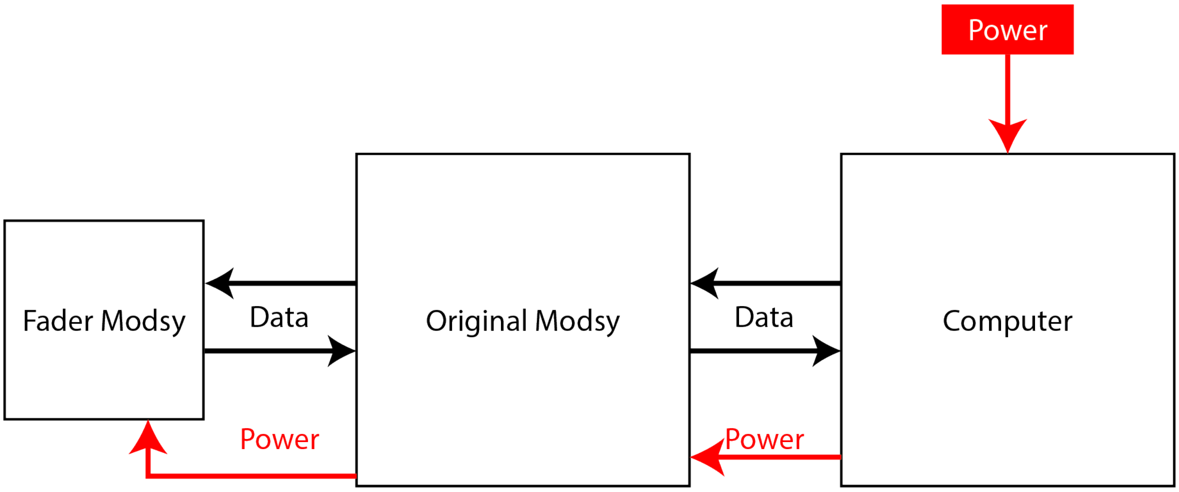

Figure 37: A schematic overview of the complete system including the additional controller

5.4 Use case diagram

This section relates to the user scenario from section 4.6. Boris is going to do the same action with the system as in section 4.6 however, now these actions are explained from a system's point of view. Only the interactions with the Juno are explained. For the Diva and every other VST, the process is the same, only the names on the screens are different. First, the actions the system undergoes are explained step by step with text. The actions are also displayed in a diagram to get a clearer image of all the actions.

To start, Boris powers on the Modsy. To do so he plugs a USB-c cable in his Modsy and into his computer. He sees the Modsy is working since all the LEDs and screen power on. This action is displayed in the second row of the diagram in figure 38.

Secondly, Boris plugs the additional controller into his Modsy controller. Once he does this the screens and LEDs on the additional controller turn on. This action is displayed in the third row of the diagram in figure 38.

To select the Juno, Boris turns the selector on the original Modsy. The Modsy needs to know which VST is selected inside Ableton and thus requests the name of the VST. The computer answers this request with the VST name. The name of the VST is displayed to Boris using a screen in the control section. This action is displayed in the fourth row of the diagram in figure 38.

Now that the Juno is selected Boris maps the Modsy controller to the Juno. To do so, he presses the mapping button on the original Modsy controller. Afterwards, the Modsy requests the parameters that need to be displayed on the screens of the Modsy and the additional controller. The computer answers this request with the parameters that need to be displayed on the Modsy and the additional controller. First, this information goes to the original Modsy which displays all these parameters on the screens. The names that need to be displayed on the additional controller are sent through to the additional controller. The displays of the additional controller are updated accordingly. The information that is showed on the screens of the additional controller and the Modsy controller is visible to Boris. This action is displayed in the fifth row of the diagram in figure 38.

Boris now starts twisting parameters on the additional controller to manipulate the sound. Boris manipulates the sound by moving a fader up and down. The value of the fader is displayed on the screen according to that fader directly. The value of the fader is also sent to the original Modsy. The Modsy directly sends this value to the computer which manipulates the software parameter. By changing the software parameter the sound is changed. This change in sound is audible for Boris. This action is displayed in the sixth row of the diagram in figure 38.

Instead of changing a parameter on the additional controller Boris now twists a parameter on the Modsy controller. Again the value of the fader is directly shown on the screen according to that fader. The value of the fader is also sent to the computer. The computer adjusts the software parameter which manipulates the sound. This change in sound is audible for Boris. This action is displayed in the last row of the diagram in figure 38.

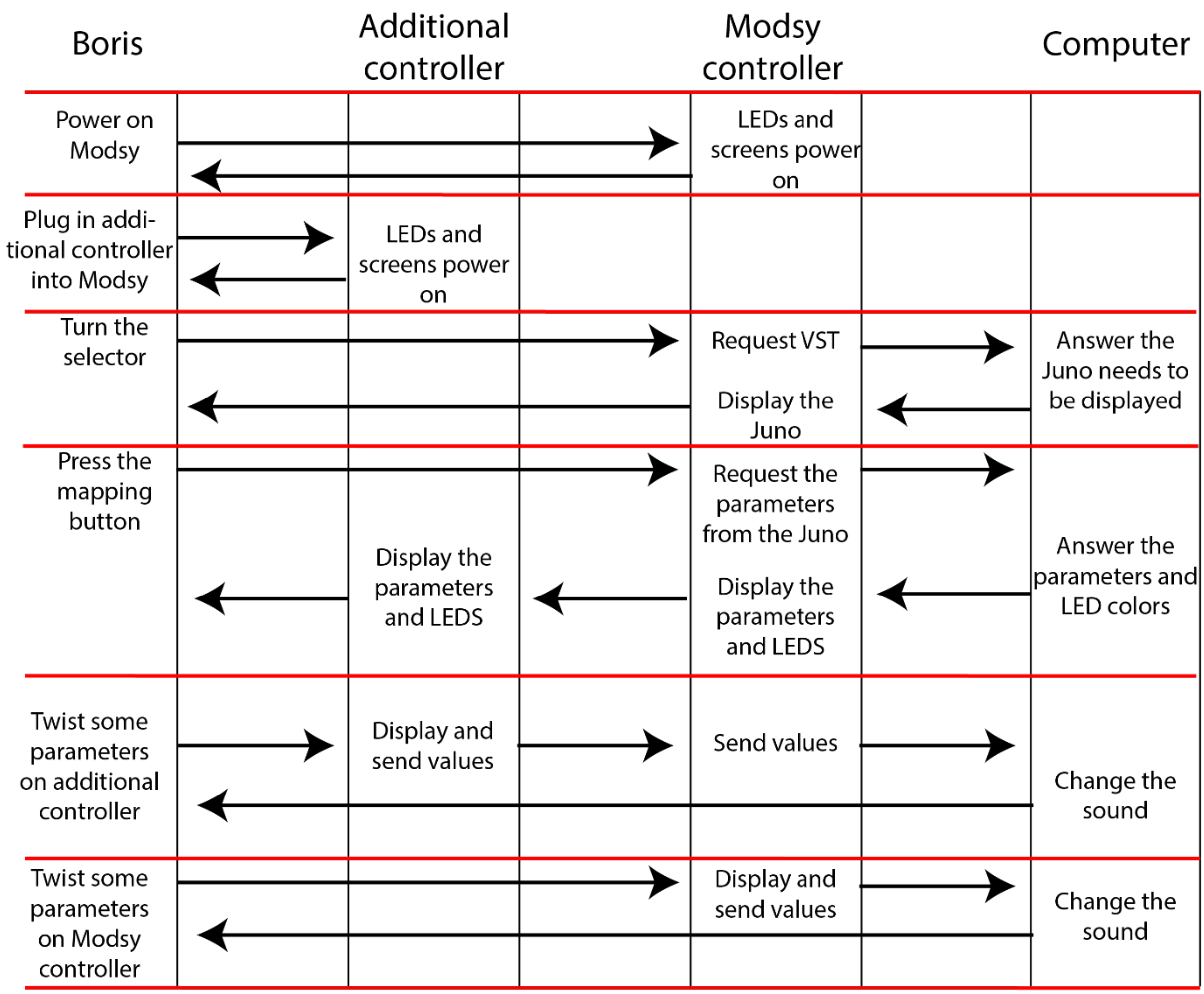

Figure 38: Use case diagram

5.5 System requirements

The system requirements are mostly the requirements from the ideation phase, however, now they are more precise and refined so they can be used during the realisation phase. Also, some of the requirements are changed from “could have” to “won’t have this time” due to time constraints. “The controller” in this case means the additional controller.

5.5.1 Must have

- FR1: The controller can be used in addition to the original controller.
- FR2: The controller must work within Ableton Live.
- FR3: The controller must have eight faders.
- FR4: The controller must use the same type of screens as the original Modsy controller
- FR5: The controller must use LEDs to indicate groupings like the original controller.
- FR6: The controller must have good communication with the original controller.
- NFR1: The use of the controller must reduce the cognitive load for the Modsy.
- NFR2: The use of the controller must help musicians to make music in a more fun, creative and expressive way.

5.5.2 Should have

- FR7: The controller should have a casing around it to support all the elements inside it.
- NFR3: The input elements of the controller should not have any noticeable delay.
- NFR4: The use of the controller should offer musicians more control over their software.

5.5.3 Could have

- No “Could have” requirements

5.5.4 Won’t have this time

- FR8: The controller won’t have a way of being used standalone.
- FR9: The controller won’t have some way of synchronizing it with the software.
- FR10: The controller won’t have sides to put the controller into an angle that is convenient to use.

6. Realisation

During the realisation phase, the controller will be developed. The documentation of the realisation phase is divided into four parts. Firstly, the hardware of the controller will be shown and explained. In the second section, the software that determines how this hardware can be used will be explained. Once the hardware and software are explained the build process of the prototype will be shown. Finally, a picture and some remarks about the prototype are listed. This final prototype will be used for the evaluation phase.

6.1 Hardware

In this section, the hardware of the controller will be explained. First, all the individual parts of the controller will be listed and it will be explained how they are connected. Secondly, an overview of the electronic system will be given. In the third section, the design of the casing will be shown. Finally, a list of components gives a quick overview of all the components that are used for the prototype.

6.1.1 Electrical components

This section gives a short explanation for all the electrical components that are used.

Arduino

The brain of the controller is an Arduino Mega 2560(Arduino, 2021a). Arduinos are small computers that can be used for prototyping something with electronics. As can be seen in figure 39, the Arduino has pins that can be used to connect other electronic devices. The Arduino itself can be connected to the computer using a USB cable. This way the Arduino is powered and it is possible to communicate with the Arduino. Via the software of Arduino themselves, code can be uploaded to the Arduino. Using this code the Arduino can operate other devices.

For the additional controller, two main functionalities of the Arduino are used; the digital communication pins and the analogue input pins. The digital pins are used to write data to the devices that are connected to these pins. The analogue input pins are used to read analogue sensors and turn this analogue value into a digital value.

Figure 39: Arduino Mega 2560

Faders

The faders(Otronic, 2021) can be used by the users as input to the system. Using the faders the parameters of the music on the software are changed. The faders used for the controller have a travel of 4.7 cm. The faders are connected to the power and ground directly. To measure the resistance of the fader an analogue output pin is connected to the Arduino. The Arduino can read the resistance and convert this to a digital value. This value is used as input for the music software. See figure 40 for the output pins of the fader. The fader has a power pin(VCC), an output pin(OUT) and a ground pin(GND).

Figure 40: Pin layout of the fader

LEDs

The LEDs(Elegoo, 2014) used for this system need to be able to display all. Common anode LEDs can be used to change the colour of the LEDs to all possible colours. Since the LEDs are common anode the 5 volts(5V) connection of the LED is shared for the red, green and blue parts of the LED (Techmirtz, 2017). The output of the LED is one pin for every colour so, one red pin, one green pin and one blue pin. These outputs of the LED go into the Arduino and can be used to change the colour of the LED. The LEDs have a resistor in between for every colour of the LED. So for the red, green and blue parts of the LED one resistor is required. The resistor is added to the circuit to make sure the current over the LED is not too high, otherwise, the LED would malfunction. See figure 41 for the pin layout of the LED. The LED has a pin for controlling the red LED(Red), a power pin(5V), a pin for controlling the green LED(Green) and a pin for controlling the blue LED(Blue).

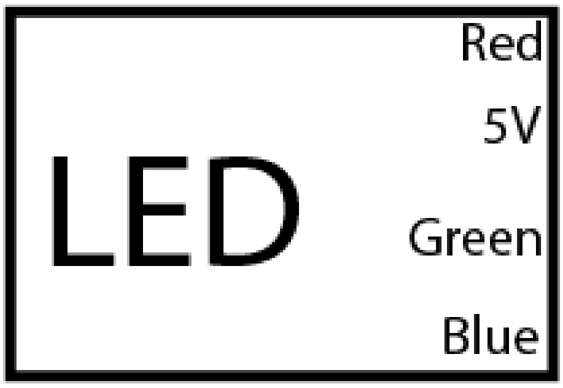

Figure 41: Pin layout of the LED

Screens

The additional controller must have eight screens (AZDelivery, 2021a) for helping the users to remember which parameter is mapped to which fader. The screens are organic light-emitting diode(OLED) screens and have a display area of 128 x 32 pixels. To connect the screens to the power, the VCC pin and the GND pin are used. The screens also have two pins that can be used to communicate with them. See figure 42 for the pin layout of the screen. To communicate with the screen the I^2C protocol is used. How this protocol works will be explained in the coming section. The screen has a pin for the data line of the I^2C protocol(SDA), a pin for the clock line of the I^2C protocol(SCK), a power pin(VCC) and a ground pin(GND).

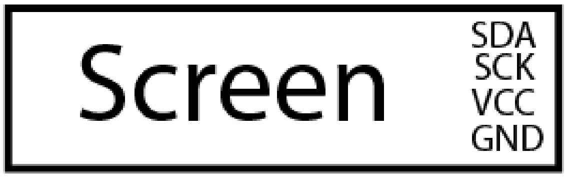

Figure 42: Pin layout of the screen,

I^2C protocol

The I^2C protocol can be used to control multiple devices easily from one main device. The protocol uses one 'master' device(Arduino) and multiple 'slave'(screens) devices. The master device is used to control the slave devices. The main advantage of using the protocol is that it only takes up two wires from the Arduino to set up the connections to multiple devices. The data line(SDL) is used to send data to the device. The second line that is used is the clock line(SCL). This line is used to synchronize all the data transfers over the I^2C bus.

With the protocol, every slave gets all the information that is sent over the channel. This information includes an address for which slave the information is meant. The first thing every slave does is checking whether the address that was sent with the information is the same as his. If this is the case, the slave uses the information to change the content on the screen. If the address does not match, the slave does not use the information (Sparkfun, 2015).

The address from such a slave is a property from the device itself. This cannot be changed afterwards. The screens that are used for the controller all have the same I^2C address. Because the screens do not have a different address it is not possible to send information to the individual screens using I^2C only. A multiplexer offers the solution for this problem.

A multiplexer(Mouser, 2021a) is a device that uses I^2C to send certain information to the wanted device. Using the data line and clock line the multiplexer only has two wires connected to the Arduino for handling the data for up to eight devices. Figure 43 shows a schematic of the inputs and outputs of the multiplexer. The VCC line is the power line and is connected to a 5V power source. The GND is connected to the according ground of that power source. The SDA and SCL lines are used to send information from the Arduino to the multiplexer. The SDx and SCx inputs of the multiplexer can be connected to the slave devices. The first information to send to the multiplexer is which of the devices connected to the multiplexer the information should go to, this can be an integer ranging from 0 to 7. All the information that is sent after this will be used only by the wanted device. This way only two pins on the Arduino can be used to operate eight devices with only one multiplexer.

For the implementation of the system, all the eight inputs(the SDx and SCx pins) of the multiplexer are connected to a screen. The screens also have a SDA and SCL line which can be connected to the multiplexer accordingly. By selecting the correct output line the wanted screen can be selected and changed. By using the multiplexer two wires from the Arduino are enough to send data to eight screens. The SDA and SCL lines of the multiplexer have a pull-up resistor connected to the 5V line. This is done to prevent a floating state of the data line.

Figure 43: Pin layout of the multiplexer

6.1.2 Electronic circuit

The section explains how the electronic circuit of the new piece of technology works. All the devices are connected to a power source in parallel. This way the voltage on all the devices is constant. The working voltage of the system is 5V. All the individual components operate on this voltage which is convenient since this way all the components can be connected to the same power source. Figure 44 shows how compartment one is connected. The black box on the left is a compartment of a screen, LED and a fader. This configuration is present in the system eight times. So the actual system has eight screens, eight LEDs and eight faders. An overview of how all the components are connected to the Arduino is available in Appendix E. An overview of how the screens are connected to the multiplexer is available in Appendix F. Figure 45 shows which compartment is where on the layout.

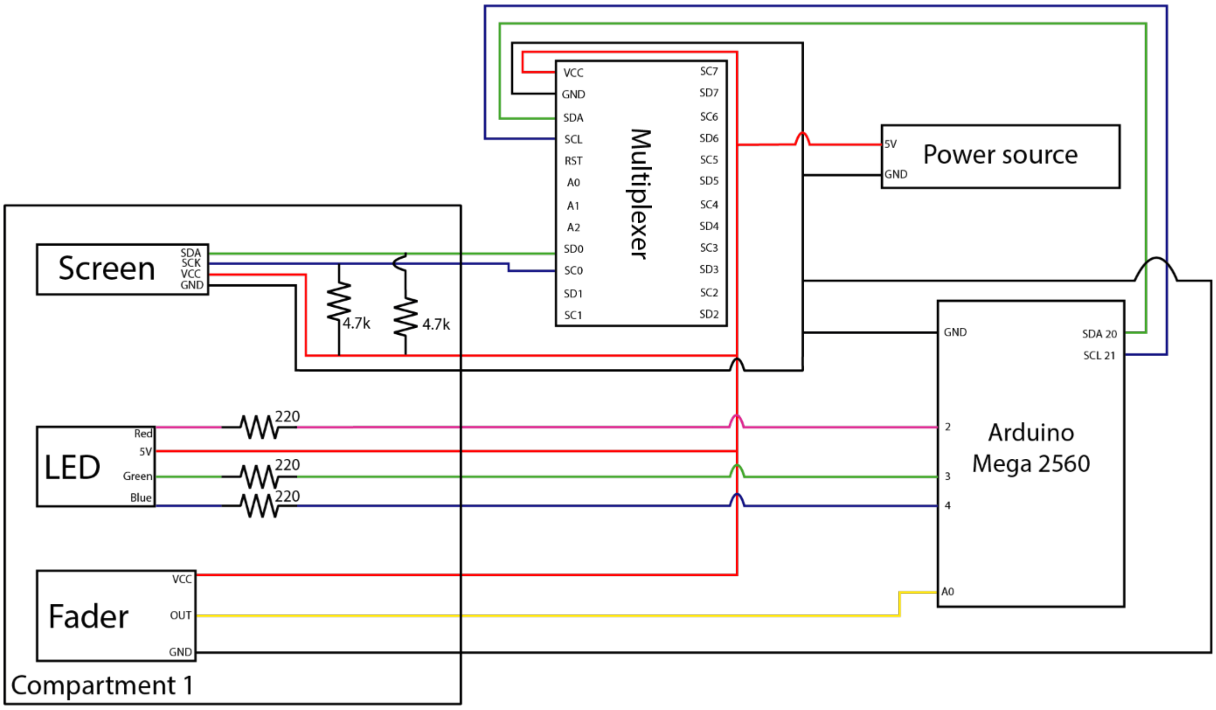

Figure 44: Connections of compartment one

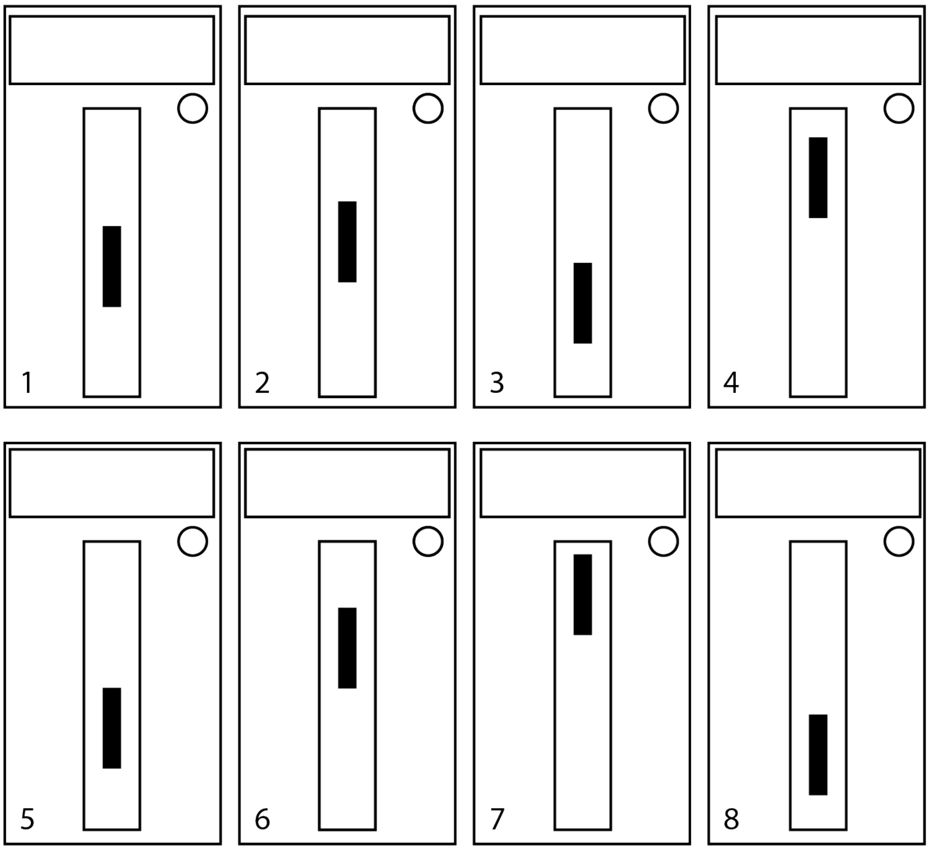

Figure 45: Explanation of the compartments of the additional controller

Circuit current draw

In this section, the current draw of the circuit will be calculated. By knowing how much current the system will draw an appropriate power source can be chosen. The whole system operates on a supply voltage of 5V. To calculate the current draw of the system the required currents from the separate components can be added up. First, the current from the screens is calculated. One screen has a maximum operating current of 27 mA (AZDelivery, 2021a). Since eight screens are used the total current draw of the screens equals 0.216 A, see equation 2.

$$I_{screens} = I_{screen} * 8 \quad [2]$$

$$I_{screens} = 27 * 10^{-3} * 8 = 0.216\ A$$

The second part of the system that draws current is the LEDs. As explained in section 6.1.1 every LED has a separate part for every colour. The current draw of one colour is about 20mA, so the total for one LED is 60mA. The total current draw for all the LEDs will then be 0.420 A, see equation 3.

$$I_{LEDs} = I_{LED} * 8 \quad [3]$$

$$I_{LEDs} = 60 * 10^{-3} * 8 = 0.480\ A$$

Since the rest of the circuit does not draw any current the total current draw is 0.696. See equation 4.

$$I_{total} = I_{screens} + I_{LEDs} \quad [4]$$

$$I_{total} = 0.216 + 0.480 = 0.696\ A$$

To provide the needed power the USB-A port of a computer will be used. The maximum output current of USB-A ports is one ampere which will thus be enough for the system.

6.1.3 Casing

To house all the electronics of the controller a casing is designed. The casing has holes for the faders, LEDs and screens. The design of the casing is made using SolidWorks. See figure 46 for a render of the design of the casing. This design was realised by using a 3d-printer. The printer that printed the casing is a Prusa i3 MK3S+ printer (Prusa, 2021). The material used for 3d-printing the casing is PETG which is plastic. All the electronics are underneath the cap of the casing so the testers cannot touch the actual circuit.

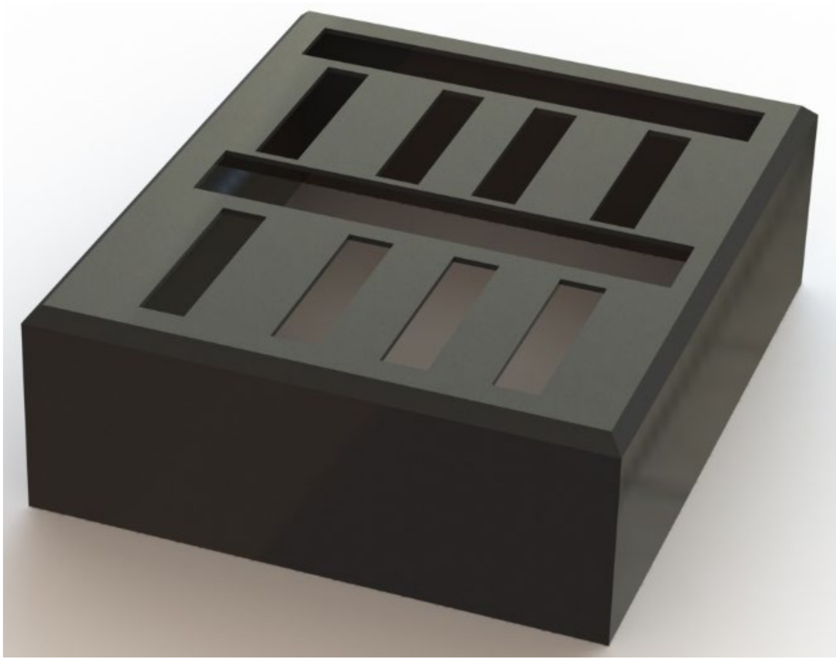

Figure 46: Render of the casing of the fader controller

6.1.4 Component overview

This section lists all the components that are used to build the prototype.

- Arduino Mega 2560 (Arduino, 2021a)
- USB 2.0 (Arduino, 2021b)
- Micro USB (AlleKabels, 2021)
- 12 printed circuit boards (AZDelivery, 2021b)
- 8 RGB LEDs (Elegoo, 2014)
- 8 OLED displays (AZDelivery, 2021a)
- 24 Resistors 220 Ω (Mouser, 2021b)
- 16 Resistors 4.7 kΩ (Mouser, 2021c)
- 1 Adafruit multiplexer(Mouser, 2021a)
- Jumper wires
- 3d-print plastic PETG (123-3D.nl, 2021)
- Soldering tin
- Fader caps (Aliexpress, 2021)

6.2 Software

To make the Arduino do the correct things code can be uploaded to the microcontroller. This section explains globally how the code works. The code for all components will be explained shortly to understand the basic behaviour of the controller. Appendix G contains the raw code that was uploaded to the Arduino.

6.2.1 LED code

To set the colour of the LED, the red, green and blue values will be sent to the according ports. The output ports of the Arduino can be set to the LED value. By setting the red, green and blue pin every colour can be made. Since the colour of the LED does not change during the program the colours are set during the setup of the program.

6.2.2 Screen code

The screens of the controller work using the I^2C protocol. Therefore before changing the name displayed on the screen, first the correct screen has to be selected. Once the correct screen has been selected the value of the screen can be sent. Just like for the LEDs the screens are set at the start of the program during the setup. The names that need to be displayed are hardcoded in the Arduino code. This means that with the current prototype the names on the screen will not change depending on the selected instrument. For the evaluation, this does not matter since only one instrument will give a good indication of whether the concept could work.

6.2.3 Fader code

The code for the faders is very simple. The only thing that needs to happen is to read out the value of the fader. The value of the fader that is read out is compared to the previous value of the fader. If the readout value differs from the previously known value, the system will be updated. The new value will be sent to the computer indicating that the parameter in the music software needs to be changed. This signal is sent to the computer using the MIDI protocol which will be explained in the coming section.

6.6.3 MIDI protocol

The communication between the controller and the computer happens using the MIDI protocol (The MIDI Association, 2021). MIDI is used very often for communication between instruments and the computer. An Arduino can be used to send and receive MIDI messages but for this application, MIDI will only be used to send information from the Arduino to the computer. Since other instruments use the MIDI protocol as well, it is well integrated within Ableton.

6.3 Build process

This section explains how the controller was built. The build process went in small steps, from compartment one to compartment eight. The LEDs, screens and faders are all soldered on empty PCBs using soldering tin. For every compartment, the LED was soldered on first. Secondly, the screen was soldered on the PCB and finally the fader. Once one of the compartments was finished, a small test was conducted to check whether all the components were working properly. Figure 47 shows the prototype after everything was soldered together.

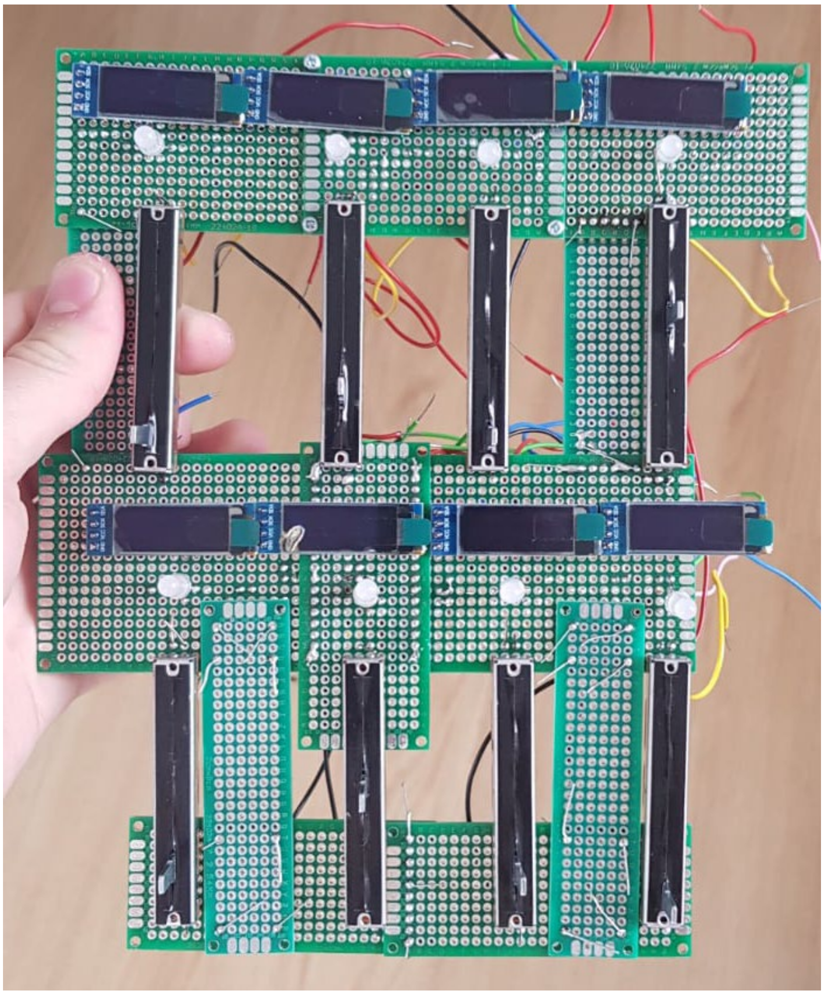

Figure 47: All the components soldered together

Once everything was soldered together, the connections to the Arduino and the multiplexer needed to be made. To do so, two new empty PCBs were used. The multiplexer is soldered directly on the PCB, and the wires from the screens are connected to the pins. An overview of how the screens are connected to the multiplexer can be seen in Appendix F. Figure 48 shows one of the PCB boards that was used with the multiplexer soldered on it. For compartments five, six, seven and eight another PCB board was used in the same way.

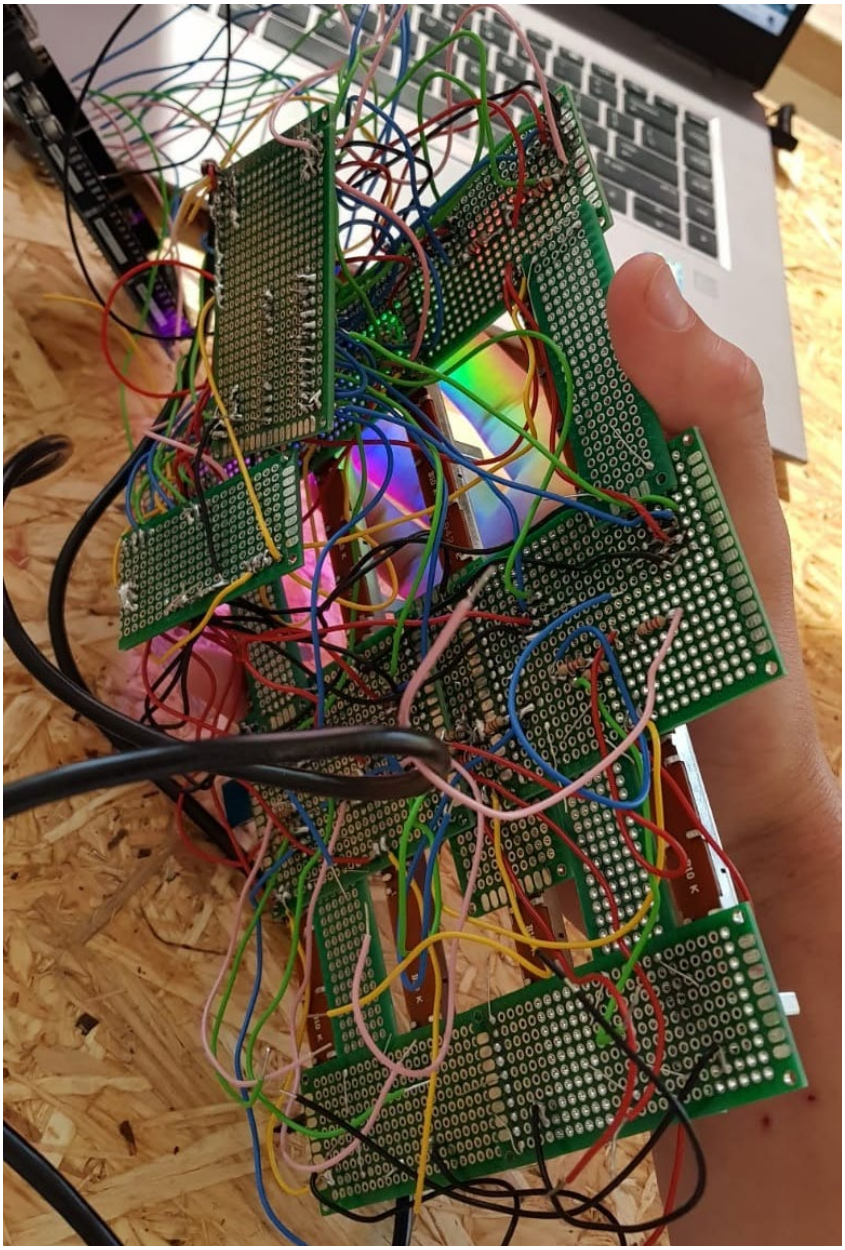

Figure 48: The soldered multiplexer on an empty PCB

These two PCB boards were used to connect all the eight compartments to the Arduino. Before moving everything into the casing, first, a final check was done whether everything was working properly. Unfortunately, two of the eight screens did not work consistently and thus compartments one and six are excluded. See figure 49 for how the prototype looked during the test.

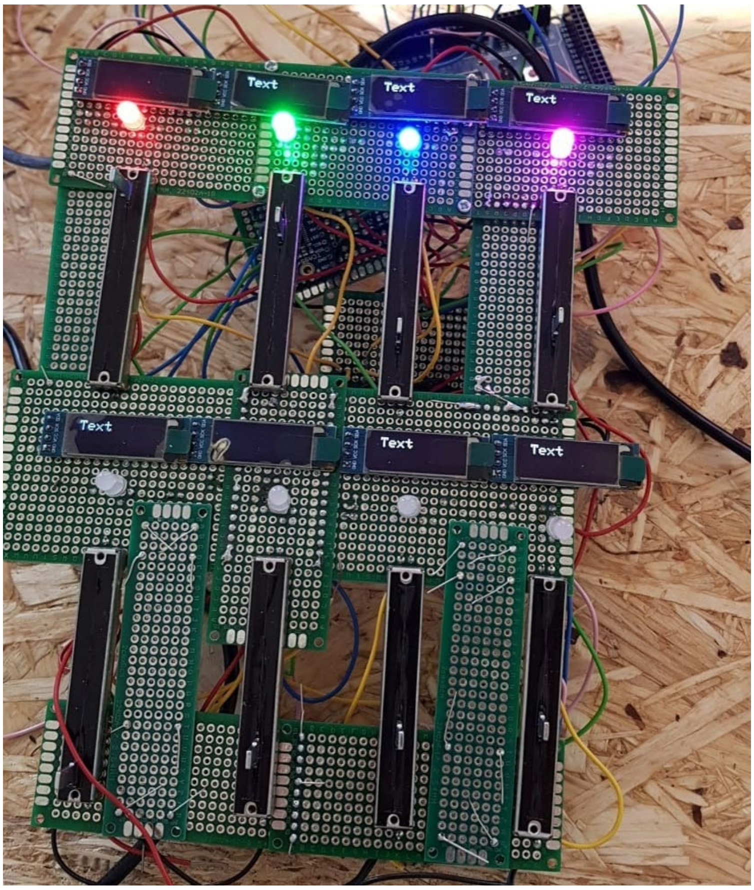

Figure 49: Testing the prototype

After everything was checked, the casing could go around it. The design process of the casing was adjusted on the measures of the current prototype so everything fitted in. Everything was fixed in the casing using tape. This made sure the whole construction was quite robust and could be tested with users. The next section shows the prototype that will be tested with users.

6.4 Final prototype

Figure 50 shows the controller that will be used for the evaluation. The prototype has some minor malfunctionings. As mentioned in the previous section, two compartments of the controller are not fully functioning. To prevent that these two compartments could cause any unclarities during the evaluation, all the components of compartments one and six were turned off. During the evaluation, it is important to keep in mind that the controller only consisted of six compartments.

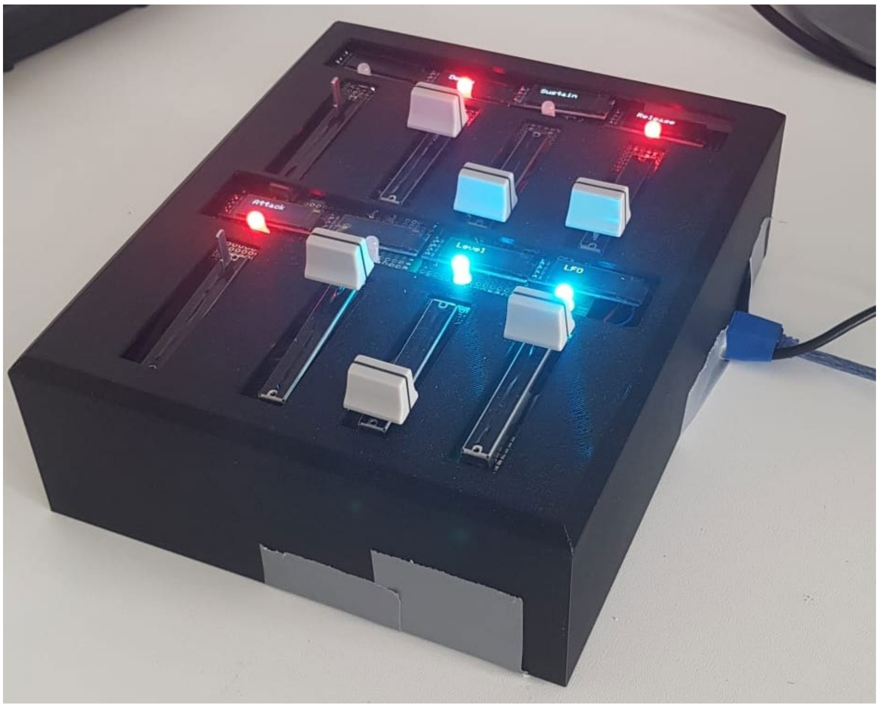

Figure 50: Hi-fi prototype of the fader controller

7. Evaluation

The evaluation phase aims to study the success of the product that was built and to look for possible improvements. The requirements that were drawn up during the ideation and specification phase, can be used to evaluate the system. Some of the requirements can be checked by looking at the system and evaluating how successful things were implemented. Other requirements can only be checked by performing user tests.

During the evaluation, it is important to keep in mind that the development of the additional controller is still in a very early stage. Therefore the main goal of the evaluation is to see whether the new system shows the possibility of becoming more understandable and intuitive to use. Another goal of the evaluation is to find possible improvements to the system. The evaluation will thus not point out one controller with an optimal cognitive load. Rather it will point out if the new system could have a positive impact on the cognitive load.

The evaluation section by explaining the test procedure of the user tests. Secondly, some remarks about the participants that did the user tests will be made. The third section is about the results of the user tests. The gathered data will be shared and the feedback from the users will be discussed. To determine the success of the prototype the requirements that were set up during the ideation and specification phase will be evaluated. Finally, the requirements will also be discussed to form new requirements. These requirements can be used used to improve the additional controller in the future.

7.1 Test procedure

This section explains how the user testing was performed and which kind of assessments were conducted. Testing the setup was done using two tests: 1) testing the original Modsy setup and 2) testing the new setup with the additional controller. Every user was asked to test both setups and to fill in the same questionnaire about both setups. The questionnaire consisted of three parts: SUS, RSME and a part about the overall experience of making music. The SUS and RSME were assessed following proven methods as explained in sections 3.8 and 3.9. The part about the overall experience of making music, however, is a subjective evaluation of the user. Here only one question was asked: "How would you rate the experience of making music using this system?". The participants could answer this question on a ten-point Likert scale ranging from very bad to very good.

During the tests, users were asked to play with both of the systems for ten minutes. For now, the system with only the original Modsy will be called system A, the test setup for this system can be seen in figure 51. The system with the original controller in combination with the additional controller will be called system B, the test setup for this system can be seen in figure 52. One very important aspect for assessing the cognitive load is the learning effect. If users play with system A for about ten minutes, it is probably much easier to play with system B afterwards. If system A would be used first, this would result in unfair test results. To counter this effect two tests were performed (both of the tests are written down below). Ideally, 50% of the user tests were performed using test A and 50% using test B. However, due to an odd number of participants, this was not the case.

Test A

1. Read the information brochure and signing an informed consent form, see Appendix H and I.

2. Make music using system A for ten minutes.
3. Fill in the questionnaire about system A.
4. Make music using system B for ten minutes.
5. Fill in the questionnaire about system B.
6. Short interview about the systems.

Test B

1. Read the information brochure and signing an informed consent form, see Appendix H and I.
2. Make music using system B for ten minutes.
3. Fill in the questionnaire about system B.
4. Make music using system A for ten minutes.
5. Fill in the questionnaire about system A.
6. Short interview about the systems.

Figure 51: Test setup for system A

Figure 52: Test setup for system B

7.1.1 Corona precautions

During this thesis, the Corona pandemic and all the measures that were taken to prevent people from getting ill, had a strong influence on the opportunities to organize live events, such as were required for the evaluation. However, during the evaluation phase of this project (June 2021), the Dutch government was slowly loosening the Corona measures. This resulted in having the opportunity to meet in small groups at the time of testing. To decrease the chance of spreading the virus, only one participant was invited to test the setup at all times. Testing the prototype happened in a controlled environment where it was easily possible to keep one and a half meters distance. Furthermore, the test setup was cleaned with disinfectant between each user test. Additionally, the researcher conducted a Corona self-test every morning on a testing day, which all resulted in a negative Corona test result.

7.2 Participants

This section displays some remarks about the participants of the user tests. As mentioned in the introduction of chapter 7, the goal of the evaluation is to see whether the additional controller could potentially decrease the cognitive load of the Modsy and to find possible improvements for the system. Therefore qualitative research helps to gain insights into what participants think of the prototype. Research has shown that five participants can be enough to find about 80% of the useability problems of a system (Alroobaea & Mayhew, 2014; Jakob Nielsen, 2012). Therefore the tests described in section 7.1 are conducted with five participants. The participants who tested the systems had a ranging level of expertise in terms of music production. Some of the participants had a basic level of expertise but there were also experts in the field of music production. This resembles the target audience of the Modsy controller quite well which is important for such a user test. Looking back at the personas as described in section 4.3.1 the participants of the user tests are quite in line with the personas. One important difference between the participants and the personas is that among the personas there was one female. The participants of the user test were only men. It is not likely that this difference has a high impact on the results. Three of the five participants followed the test procedure as described in Test A. Two of the five participants followed the test procedure as described in Test B.

7.3 User testing results

This section explains how the user tests went and what the results of the tests are. In total five participants tested the system. These participants were all digital musicians who have experience in making music in Ableton Live. Participants one, three and five performed test A, participants two and four performed test B. All the tests were completed successfully. The results of the tests will be explained in the coming section. First, the results of the SUS analysis will be discussed.

7.3.1 SUS analysis

The SUS statements were answered for system A and system B, as explained in section 7.1. The scores of the SUS analysis are calculated using the calculation as explained in section 3.8. Since five participants filled in the questionnaires, there are five SUS scores for both systems. The scores of the SUS are displayed in table 10. A high SUS score means the system is very useable.

	Modsy standalone	**Modsy with additional controller**
Test 1	77.5	77.5
Test 2	85	72.5
Test 3	80	90
Test 4	92.5	87.5
Test 5	72.5	92.5
Average	81.5	84

Table 10: SUS scores of user tests

Looking at the average scores of both tests the SUS analysis points out that the additional controller has a higher useability score. However, the difference in score between the Modsy standalone and the additional controller is very small. This, in addition to the fact that this test was only done with five participants, means the test does not prove any significant better system. Although the average of the SUS analysis seems to be in favour of the additional controller, it is impossible to say anything about the difference in useability between the two systems.

7.3.1 RSME analysis

The same participants that filled in the SUS also answered a question about the cognitive load of the system. Here an RSME analysis was used. The RSME analysis can be used to monitor how high the cognitive load is of a system based on the opinion of users, as explained in section 3.9. A very low RSME score means that participants thought they did not need much mental effort to use the system. Table 11 shows the results of the RSME analysis.

	Modsy standalone	**Modsy with additional controller**
Test 1	5	5
Test 2	1	2
Test 3	3	2
Test 4	1	2
Test 5	4	2
Average	2.8	2.6

Table 11: RSME scores of users tests

The results of the RSME are in line with the results of the SUS analysis. The average score of all the tests shows that on average the additional controller seems to have a lower required cognitive load than the Modsy standalone. The difference, however, is really small and because of the low number of participants, it is impossible to say anything about the difference in cognitive load between the two systems.

7.3.3 The experience of making music

Besides the proven SUS and RSME methods, the questionnaire also had one question about the experience of making music. This question was added to gain insight into how the participants perceived the overall process of making music using systems A and B. Table 12 shows the results of this question. The answers to the questions could range from one, which means a very bad experience of making music, to ten, which means a very good experience of making music. Thus a high score on the questionnaire means that the participant had a very good experience of making music using the system.

	Modsy standalone	**Modsy with additional controller**
Test 1	8	8
Test 2	9	9
Test 3	7	8
Test 4	4	8
Test 5	9	10
Average	7.4	8.6

Table 12: Experience of making music scores of users tests

It seems that adding the additional controller to the system improves the overall experience of making music. The average score of the system with the additional controller is much higher than without the additional controller. On the contrary to the two previous tests, the scores of the individual tests are also in favour of the additional controller. Three persons gave the system with the additional controller a higher score than the Modsy standalone. Two persons gave the same score for both systems. Again, it is important to mention that this test was only conducted with five participants thus the results are not significant.

7.3.4 Participants interview

After the participants used both systems and filled in the questionnaires, a short unstructured interview was held. The researcher asked the participants if they had any remarks about one of the systems. All the participants of the user testing were Dutch, therefore, the interviews were also held in Dutch. The raw written out interview can be seen in Appendix J. This section highlights some of the interesting things the participants said during the interview. The selection of the quotes is based on how relevant the quotes are for this thesis. Some of the remarks did not have anything to do with the cognitive load and thus these are not selected. The quotes as displayed here are translated by the researcher.

- “I directly assumed the sliders from the Juno are also the sliders on the hardware.”
- “It is very nice that there is more space for other parameters to control.”
- “I can imagine that the use of faders is very logical if it is the same as the VST.”
- “It would even be handier if motorized faders were used.”
- “Unconsciously, the experience with faders might be better for some parts of the music production than with potentiometers.”
- “I think the faders are a good addition if the ADSR is always mapped to it. In that case, I directly know these parameters are there.”

- “It is nice to see the waveform of the ADSR directly on my hardware. Faders are almost always used for the ADSR so it makes much sense that this is the same now.”

All these positive comments about the additional controller suggest that the system could be an improvement compared to the Modsy standalone. However, one of the participants was less positive about the new system. Some of the quotes from the interview with him are displayed below.

- “It would be easier to have everything on one controller.”
- “If the functionality of the button stays the same, for me there is no added value in having faders instead of potentiometers.”
- “In the way, I use my instruments it does not matter if I use faders or potentiometers.”

Concluding from the interviews, it seems that having an additional controller next to the Modsy could be an improvement for the whole system. Looking at the results, it is also important to notice that not all participants were enthusiastic about the new system. One of the five participants said he did not see any added value in having the additional controller next to the normal Modsy.

7.4 Evaluation of requirements

This section reflects on the requirements that were set during the ideation and specification phase. Every requirement will be evaluated and a short explanation will be given why a certain requirement is reached or not. Some of the requirements can be reflected upon by looking at the prototype alone, some other requirements can be reflected upon by using the feedback of the user tests. “The controller” in this case means the additional controller.

7.4.1 Must have

- FR1: The controller can be used in addition to the original controller.

The controller currently needs to be plugged into the computer instead of into the Modsy controller. For the future implementation of the additional controller, it should be possible to plug the additional controller into the Modsy itself. The usage of the controller does work fluently with the original Modsy already. This requirement is partly reached.

- FR2: The controller must work within Ableton Live.

The controller does work within Ableton Live. This requirement is reached.

- FR3: The controller must have eight faders.

The controller does have eight faders. However, due to technical failures, only six of the faders were used during the evaluation. For future implementation, this should be fixed. This requirement is partly reached.

- FR4: The controller must use the same type of screens as the original Modsy controller

The controller does use the same type of screens as the original Modsy controller. This requirement is reached.

- FR5: The controller must use LEDs to indicate groupings like the original controller.

The controller does use LEDs to indicate groupings, just like the original controller. This requirement is reached.

- FR6: The controller must have good communication with the original controller.

The controller did not communicate with the original controller. For the evaluation, it seemed like there was communication between the two controllers, however, this was not the case. To test the device it did not matter that there was no active communication between the two controllers however for future implementation this should work. This requirement is not reached.

- NFR1: The use of the controller must reduce the cognitive load for the Modsy.

Although the user tests suggest that the required cognitive load of the additional controller was lower than for the original Modsy there is no significant proof. In the future, more research should look at if the additional controller has a lower required cognitive load to prove any significance.

- NFR2: The use of the controller must help musicians to make music in a more fun, creative and expressive way.

Although the user tests suggest that the experience of making music was better with the additional controller, it is difficult to say if the controller enables musicians to put more fun, expression and expression in their music. In the future, more research should look at if this requirement is reached.

7.4.2 Should have

- FR7: The controller should have a casing around it to support all the elements inside it.

The controller does have a casing around it. The elements in the casing should be more fixed in place to make sure the controller is bump proof. This requirement is reached.

- NFR3: The input elements should not have any noticeable delay.

The users did not notice any delay. This requirement is reached.

- NFR4: The use of the controller should offer musicians more control over their software.

The additional controller did offer users more control over their software. Users explicitly mentioned that they appreciated having more control over their software. However, no significant proof could be found that supports this statement. This requirement is partly reached.

7.4.3 Could have

- No “Could have” requirements

7.4.4 Won’t have this time

- FR8: The controller won’t have a way of being used standalone.
- FR9: The controller won’t have some way of synchronizing it with the software.
- FR10: The controller won’t have sides to put the controller into an angle that is convenient to use.

7.5 Discussion requirements

Most of the requirements from the specification phase are met. From the “Must have” requirements FR2, FR4 and FR5 are reached. FR1 and FR3 are partly reached and FR6 is not reached at all. The reason why these three requirements were not reached is especially due to time constraints. Due to the low number of participants in the user tests, no significant result could be found to prove NFR1 and NFR2. However, looking at the results of the user testing, it seems that the additional controller does have the potential of reaching these two requirements.

All the “Should have” requirements are reached. Looking back at the requirements it makes more sense to move FR7 to the “Must have” requirements. Although it does not do anything regarding the cognitive load, it is impossible to use such a product if it is not sturdy enough. NFR3 was reached; there did not seem to be any difference in delay between the original Modsy and the additional controller. However, if the additional controller would be connected to the Modsy instead of the computer this could change.

NFR4 is partly reached. This requirement is partly a functional and non-functional requirement. Since faders were added to the system, more parameters could be controlled using the Modsy. Assessing the requirement like this is a way of testing the functional part of the requirement. However, it is also important to question whether adding more controls helps musicians during the process of making music. Although adding more controls means more parameters can be controlled, it is questionable if musicians are actually in need of this many types of input. After one of the user tests, one of the participants explicitly mentioned that he liked the fact that it was possible to control more parameters by adding the additional controller. Of course, this is no significant proof that all musicians want this.

7.6 Final requirements

Looking back on the evaluation of the requirements from the previous sections, this section draws up new requirements. These requirements can be used for future work to improve the additional controller. Most of the requirements from the specification phase are still the same. The coming sections list all the requirements that could be implemented in a future version of the additional controller. In section 7.7 the changes compared to the requirements from the specification phase are discussed. “The controller” in this case means the additional controller.

7.6.1 Must have

- FR1: The controller can be used in addition to the original controller.
- FR2: The controller must work within Ableton Live.
- FR3: The controller must have eight faders.
- FR4: The controller must use the same type of screens as the original Modsy controller
- FR5: The controller must use LEDs to indicate groupings like the original controller.
- FR6: The controller must have good communication with the original controller.
- FR7: The controller must have a casing around it to support all the elements inside it.
- FR11: The elements inside the casing must be fixed in place properly.
- NFR1: The use of the controller must reduce the cognitive load for the Modsy.
- NFR2: The use of the controller must help musicians to make music in a more fun, creative and expressive way.

7.6.2 Should have

- FR9: The controller should have motorized faders to synchronize the hardware with the software.
- NFR3: The input elements should not have any noticeable delay.
- NFR4: The use of the additional controller should offer musicians more control over their software.

7.6.3 Could have

- FR8: The controller could have a way of being used standalone.
- FR10: The controller could have sides to put the controller into an angle that is convenient to use.

7.6.4 Won’t have this time

- No “Won’t have this time” requirements

7.7 Remarks final requirements

This section discusses the changes of these final requirements compared to the requirements of the specification phase. To start, FR8 and FR10 are moved from “Won’t have this time” to “Could have”. During the participant interviews, FR8 did not come across, which makes it questionable if this is a functionality that needs to be implemented. The users did not mention that it would be useful to make use of the additional controller standalone. However, to make sure this is not an important feature of the controller, this should be tested in future research. The participants also did not mention anything about the angle of the controller. Thus for FR10, the same conclusion can be drawn as for FR8.

FR9 from the specification phase is moved from “Won’t have this time” to “Should have”, and is specified more in detail. During the specification phase, this requirement was: “The controller won’t have some way of synchronizing it with the software.”. During the user interviews, a remark was placed that the faders could be motorized faders. Motorized faders are normal faders but with the ability to move themselves using a small motor. Therefore using motorized faders it is possible to make sure the hardware is synchronized with the software by moving the faders. Therefore FR9 is specified more to: “The controller should have motorized faders to synchronize the hardware with the software.”.

One functional requirement is added to the final requirements. During the evaluation phase, it was noticed that the elements inside the casing were not fixed in place well enough. This resulted sometimes in unexpected behaviour of the electronics and this is possibly also the reason why two of the faders did not work. Therefore FR11 is added as a “Must have”.

8. Conclusions and future work

This section consists of two parts, the conclusions and future work possibilities. The first part will discuss the research questions that were formulated at the start of this thesis. The second section looks at possibilities for future work.

8.1 Conclusions

In this section, the goal and the research questions will be discussed. The goal, as stated in the introduction, was: “Decrease the cognitive load of the Modsy controller while maintaining its main functionalities.”. To study this goal, one main research question was formulated: “How to decrease the cognitive load for the Modsy controller?”. This question will be answered by answering the sub-questions first. After these sub-questions, the main research question will be answered. The last section will discuss whether the goal was reached.

The first sub-question was answered during the literature review. This sub-question was: “How can cognitive load theory be used within the context of product design?”. Multiple articles about cognitive load helped to get a better understanding of cognitive load and its implications for design. Using literature, six categories that influence the cognitive load were identified. The six categories are: “Combination of audio and visuals”, “Using signals”, “Minimalistic design”, “Provide mnemonics”, “Pacing learning” and “Practicing techniques”. In the state of the art, these categories were used to take a look at how other products apply these six categories. By applying these six categories, cognitive load theory can be used within the context of product design. This is also the answer to the first sub-question: Cognitive load theory can be used within the context of product design, by applying the six load reducing categories.

The second sub-question was more or less answered during the ideation and specification phase. The sub-question was “How to implement changes upon the Modsy controller to decrease the cognitive load?”. During the ideation phase, brainstorms helped to generate ideas. A lot of these ideas do show potential to help to decrease the cognitive load of the Modsy controller. Some of the ideas were combined to make the biggest improvement to the Modsy controller. The final concept from the ideation and specification phase was an idea about an additional controller that could be realised during the realisation phase. In the design of this prototype, the knowledge that digital musicians already have about their music software was used to reduce the cognitive load of the controller. This makes use of the load reducing method *pacing learning*. The concept of the additional controller is thus a possible answer to sub-question two. To see if this concept helped to decrease the cognitive load, user tests were conducted. During the user tests, no significant proof could be found to confirm this new concept. However, the results of the user tests do show that the additional controller has the potential of improving the Modsy controller. Thus a possible answer to sub-question two is “The use of an additional controller with faders next to the Modsy controller might decrease the cognitive load of the Modsy”.

The last sub-question was: “What are the differences between the new prototype and the original Modsy controller, in terms of cognitive load, overall useability and experience of making music?”. This question was answered during the evaluation phase. User tests helped to identify differences between the original Modsy and the Modsy in combination with the additional controller. Since the user tests were only conducted with five participants, the results of the user tests are not significant. However, the results can help to get insights into what musicians think of the system and whether it shows the potential of improving the Modsy controller. The three aspects of this sub-question will be discussed separately in the coming sections.

Firstly, the cognitive load of the controller will be discussed. The results of the user tests show that the new prototype had a slight decrease in terms of cognitive load, which is positive. However, due to the low number of participants and the fact that the decrease was so small, the result is not significant. But it is good to notice that there was not a big increase in terms of cognitive load. Since more input parameters were added to the Modsy it is interesting to see that the cognitive load did not differ that much. The prototype especially applied the load reducing category *pacing learning*. However, by adding more input parameters to the controller, the design of the device got less minimalistic. Therefore the load reducing category *minimalistic design* was applied worse on the new system. Possibly the load reducing categories *pacing learning* and *minimalistic design* compensated for each other. The decrease in terms of cognitive load due to *pacing learning* might have compensated for the increase in terms of cognitive load due to *minimalistic design*. However, to confirm this conclusion more research is needed.

Secondly, the overall useability of the system will be discussed. Looking at the user tests, the new prototype had a slight increase in terms of useability, which is positive. However, these results are not significant and thus no conclusions can be drawn from these results. The user tests were combined with a short unstructured interview. The participants made very interesting comments. Most of the participants were positive about the additional controller. One of the participants explicitly mentioned that by adding more faders, he was able to get more control over his music. This seems like an increase in terms of the useability of the system. For others, it did not matter much what kind of inputs they had. Overall, no conclusion can be drawn in terms of the useability of the system. But the slight increase in terms of useability does show that the new system has the potential of increasing the useability of the Modsy. To confirm this, more research is needed.

The third aspect that was tested during the user tests was the overall experience of making music. Here a bigger difference between the original Modsy and the Modsy in combination with the additional controller was noted. The new system was rated with an 8.6 compared to a 7.4 for the original Modsy. Due to the low number of participants, this result is not significant. It is, however, a very interesting result. Apparently, users rated the new system much higher than the old system. Future research could find significant results for this, which would be a great achievement. If the new system is so much more convenient to work with, it would be very interesting to work this system out to a product.

To answer sub-question three the results of the user tests can be combined. The question was: “What are the differences between the new prototype and the original Modsy controller, in terms of cognitive load, overall useability and experience of making music?”. For all three aspects, no significant results could be found. But the additional controller does show the potential of a decreased cognitive load, an increased system usability and an increased user experience. To find significant results for this conclusion, more research is needed.

Now that the sub-questions are answered the main research question can be answered. The main research question was: “How to decrease the cognitive load for the Modsy controller?”. Several ideas to decrease the cognitive load of the Modsy controller were generated during the ideation phase. These ideas were combined into one main concept. This is the concept of the additional controller where especially *pacing learning* could help to decrease the cognitive load. User tests showed no significant proof that this concept has a lower cognitive load compared to the original system. So no final answer to the main research question can be given, however, the concept of the additional controller could be a possibility to decrease the cognitive load of the Modsy controller.

Finally, the goal of this thesis will be evaluated. The goal as stated at the start of this thesis was: "Decrease the cognitive load of the Modsy controller while maintaining its main functionalities.". Due to the low number of participants during the user tests, no final decision can be given whether the additional controller in combination with the Modsy has a decreased cognitive load compared to the original Modsy. During this thesis an addition controller was build and therefore all the main functionalities are still in the design. This system has the potential of having a lower cognitive load, however, more research is needed to confirm this.

8.2 Future work

The last section of this thesis gives possibilities for future work. These recommendations are mainly based on things that were noted during the design process, or things that could not be done due to the time constraints of this project.

The most important thing that future work should test is whether the new system is a significant improvement compared to the current Modsy design. The user tests during this thesis did show that the additional controller does have the potential to improve the current Modsy design. However, due to the low number of participants, no significance could be proven. A possibility for future work could therefore be to do the same user tests, however, now with more participants to find significant results.

Another possibility to increase the quality of the evaluation of the additional controller is to do more elaborate user tests. Currently, the user tests were about one hour due to the time constraints of this thesis. In the future, it would be a good test to see what users think of both products after using them for about a week. This gives users a better chance to test both systems more elaborately. This also allows using a more accurate way of testing the cognitive load of the system. Besides the RSME method, there are other more ways of measuring the cognitive load. Examples of cognitive load testing methods are the tapping test (Albers, 2011), the NASA-TLX scale (S. Hart, 2006; S. G. Hart & Staveland, 1988) or secondary tasks tests (Sweller, 2018).

Future work could also look at improving the prototype to make it more accurate compared to the final product. The current prototype still had quite some flaws which made it an unfair way of testing. Only six compartments with a LED, screen and fader were fully functioning which should be eight. This could have some effect on the cognitive load since the participants of the user tests now only had to focus on six of the eight input parameters. However, it is also possible that these inaccuracies increased the cognitive load of the system. The participants had to think about the malfunctionings of the prototype, which could have distracted them from focusing on the music. To ensure a better build quality of the prototype, it is recommended to make a printed circuit board(PCB) externally. This way it is possible to design the electronics using software to make sure no mistakes can be made. In the current prototype, everything was soldered together. This caused some instabilities in the connections and led to the fact that two compartments were malfunctioning. By using a PCB it is also much easier to fix everything in place properly so the product is more sturdy.

Finally, the product needs to work fluently with the software and the Modsy controller. Currently, participants of the user tests could only play with one type of instrument inside their DAW. Since the duration of the test was so short most of the participants did not even notice this, however, with longer tests this must work. On the same note, the additional controller should be easy to connect to the original Modsy. In the current design, the additional controller was connected to the computer. For this short test, this did not matter however, if the user tests would be longer this could get annoying for the participants. Therefore, future work should make sure these things are work properly.

Bibliography

123-3D.nl. (2021). *123-3D Filament zwart 1,75 mm PETG 1 kg (Jupiter serie) - conducted on 2021-06-07*. https://www.123-3d.nl/123-3D-Filament-zwart-1-75-mm-PLA-1-kg-Jupiter-serie-i1800-t7316.html?mkwid=sAj30GflU_dc%7Cpcrid%7C445479947650%7Cpkw%7C%7Cpmt%7C%7Cslid%7C%7Cprid%7CPF_DFP11000_8718237049205_&pgrid=101846517005&ptaid=pla-928234287019&gclid=Cj0KCQjwx

Ableton. (2021). *Ableton Push 2 - conducted on 16-04-2021*. https://www.ableton.com/en/push/tech-specs/

Agile Business. (2014). MoSCoW Prioritisation. In *DSDM Agile Project Framework Handbook*. https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation

Akai Professional. (2021). *LPD8 - conducted on 16-04-2021*. https://www.akaipro.com/lpd8-lpd8

Albers, M. (2011). Tapping as a measure of cognitive load and website usability. *Proceeding SIGDOC '11 Proceedings of the 29th ACM International Conference on Design of Communication*. https://doi.org/10.1145/2038476.2038481

Aliexpress. (2021). *Fader caps - conducted on 06-07-2021*. https://nl.aliexpress.com/item/32846132060.html?aff_fcid=92407a4083d1403ebb25bd7e1b3fc250-1625564899377-02057-Y7bAZbY&cv=565442&aff_fsk=Y7bAZbY&af=282199&aff_platform=aaf&afref=https%3A%2F%2Fprijsbest.nl%2F&sk=Y7bAZbY&aff_trace_key=92407a4083d1403ebb25bd7

AlleKabels. (2021). *Micro Kabel USB 2.0 - conducted on 06-07-2021*. https://www.allekabels.nl/powerbank/15477/1275763/micro-kabel-usb-2.0.html?gclid=CjwKCAjw_o-HBhAsEiwANqYhp9SfLt2HjSwyccA1x3njMJUNOK6muo_kGr52Zz5ftt2c5LjBCRF6vhoC8RcQAvD_BwE

Alroobaea, R., & Mayhew, P. J. (2014). How Many Participants are Really Enough for Usability Studies? In *Proceedings of 2014 Science and Information Conference, SAI 2014*. https://doi.org/10.1109/SAI.2014.6918171

Arduino. (2021a). Arduino Mega 2560 Rev3 | Arduino Official Store - conducted on 06-07-2021. In *Arduino*. https://store.arduino.cc/arduino-mega-2560-rev3

Arduino. (2021b). *USB 2.0 cable - conducted on 06-07-2021* (p. 500). https://store.arduino.cc/usb-2-0-cable-type-a-b

AZDelivery. (2021a). *OLED Display - conducted on 21-06-2021*. https://cdn.shopify.com/s/files/1/1509/1638/files/0_91_Zoll_Display_Datenblatt_AZ-Delivery_Vertriebs_GmbH_1edf9d8c-04f4-4a8a-9f88-4774d7194198.pdf?v=1622442951

AZDelivery. (2021b). *PCB Board Set Hole Grid Plate Hole Grid Board PCB - conducted on 06-07-2021*.

Beghelli, A., & Prieto, P. (2020). Too many attributes!: Diminishing the cognitive load of metaphor generation in product design. *Proceedings of the 6th International Conference on Design Creativity, ICDC 2020*, 352–359. https://doi.org/10.35199/ICDC.2020.44

Berkenbos, R.-J. (2021). *Modsy tutorial: an engaging and easy to understand introductory tutorial*.

Brooke, J. (1995). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry*, *November 1995*, 207–212. https://doi.org/10.1201/9781498710411-35

Clark, R., Nguyen, F., Sweller, J., & Baddeley, M. (2006). Efficiency in Learning: Evidence-Based Guidelines to Manage Cognitive Load. *Performance Improvement*, *45*. https://doi.org/10.1002/pfi.4930450920

Costa, R. (2020). *How to design user scenarios with examples*. https://www.justinmind.com/blog/how-to-design-user-scenarios/

Crosby, B. (1991). Stakeholder Analysis : A Vital Tool for Strategic Managers. *U.S Agency for International Development*, *2*, 1–6.

Dabell, J. (2018). *The Disney Creativity Strategy*. https://johndabell.com/2018/11/10/the-disney-creativity-strategy/

Dashfade. (2020). *Is Ableton Push 2 Good For Beginners?*

Debue, N., & van de Leemput, C. (2014). What does germane load mean? An empirical contribution to the cognitive load theory. *Frontiers in Psychology*, *5*(SEP), 3389. https://doi.org/10.3389/fpsyg.2014.01099

Elegoo. (2014). *RGB Common Anode - conducted on 06-07-2021*. https://www.elegoo.com/blogs/arduino-projects/elegoo-3mm-and-5mm-led-kit-tutorial?_pos=1&_sid=f5b2473f6&_ss=r

Feinberg, S., & Murphy, M. (2000). Applying cognitive load theory to the design of Web-based instruction. *IPCC/SIGDOC 2000: Technology and Teamwork - Proceedings, IEEE Professional Communication Society International Professional Communication Conference and ACM Special Interest Group on Documentation Conference*, 353–360. https://doi.org/10.1109/IPCC.2000.887293

Freeman, R. E. (1984). *Strategic management : a stakeholder approach*. Pitman.

Galy, E., Cariou, M., & Mélan, C. (2012). What is the relationship between mental workload factors and cognitive load types? *International Journal of Psychophysiology*, *83*(3), 269–275. https://doi.org/10.1016/j.ijpsycho.2011.09.023

Galy, E., & Mélan, C. (2013). Interactions between cognitive load factors on working memory performance in laboratory and field studies. *Working Memory: Developmental Differences, Component Processes and Improvement Mechanisms*, 97–113.

George, J. M. (2007). 9 Creativity in Organizations. *The Academy of Management Annals*, *1*(1), 439–477. https://doi.org/10.1080/078559814

Ghanbary Sartang, A., Ashnagar, M., Habibi, E., & Sadeghi, S. (2016). Evaluation of Rating Scale Mental Effort (RSME) effectiveness for mental workload assessment in nurses TT -. *JOHE*, *5*(4), 211–217. https://doi.org/10.18869/acadpub.johe.5.4.211

Hart, S. (2006). Nasa-task load index (Nasa-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 50). https://doi.org/10.1177/154193120605000909

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, *52*(C), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Hoever, I. J., van Knippenberg, D., van Ginkel, W. P., & Barkema, H. G. (2012). Fostering team creativity: Perspective Taking as Key to Unlocking Diversity's Potential. *Journal of Applied Psychology*, *97*(5), 982–996. https://doi.org/10.1037/a0029159

IZotope. (2018). What Is Mix Automation? Everything You've Been Too Afraid to Ask. In *iZotope*. https://www.izotope.com/en/blog/mixing/what-is-mix-automation.html

Jakob Nielsen. (2012). How Many Test Users in a Usability Study? In *Nielsen Norman Group* (p. 1). https://www.nngroup.com/articles/how-many-test-users/

Kalyuga, S. (2011). Cognitive Load Theory: How Many Types of Load Does It Really Need? *Educational Psychology Review*, *23*(1), 1–19. https://doi.org/10.1007/s10648-010-9150-7

Karras et al., & Nvidia. (2019). *This person does not exist*.

Klepsch, M., Schmitz, F., & Seufert, T. (2017). Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load. *Frontiers in Psychology*, *8*(NOV), 3389. https://doi.org/10.3389/fpsyg.2017.01997

Korg. (2021). *nanoKONTROL Studio - conducted on 26-05-2021*. https://www.korg.com/nl/products/computergear/nanokontrol_studio/index.php

LANDR. (2020). *The 10 Best VST Instruments Every Producer Needs*.

Larson, E., & Larson, R. (2004). Use cases: what every project manager should know. In *PMI® Global Congress*.

Lee, E. A. L., & Wong, K. W. (2014). Learning with desktop virtual reality: Low spatial ability learners are more positively affected. *Computers and Education*, *79*, 49–58. https://doi.org/10.1016/j.compedu.2014.07.010

Leppink, J., Paas, F., Van der Vleuten, C. P. M., Van Gog, T., & Van Merriënboer, J. J. G. (2013). Development of an instrument for measuring different types of cognitive load. *Behavior Research Methods*, *45*(4), 1058–1072. https://doi.org/10.3758/s13428-013-0334-1

Leppink, J., Paas, F., van Gog, T., van der Vleuten, C. P. M., & van Merriënboer, J. J. G. (2014). Effects of pairs of problems and examples on task performance and different types of cognitive load. *Learning and Instruction*, *30*, 32–42. https://doi.org/10.1016/j.learninstruc.2013.12.001

LG. (2021). *Full Function Standard TV Remote Control - conducted on 15-04-2021*. https://www.lg.com/us/tv-audio-video-accessories/lg-AGF76631064-tv-remote-contol

Logitech. (2021a). *Logitech Keyboard K120 - conducted on 16-04-2021* (pp. 1–2). https://www.logitech.com/en-us/products/keyboards/k120-usb-standard-computer.920-002478.html

Logitech. (2021b). *Logitech MX Master 3 Wireless Mouse - conducted on 15-04-2021*. https://www.logitech.com/en-us/products/mice/mx-master-3.910-005620.html

Mader, A., & Eggink, W. (2014). A design process for Creative Technology. *Proceedings of the 16th International Conference on Engineering and Product Design Education: Design Education and Human Technology Relations, E and PDE 2014*, *September*, 568–573.

Mathijssen, O. (2021). *Playfulness of the Modsy*.

Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, *38*(1), 43–52. https://doi.org/10.1207/S15326985EP3801_6

McLeod, S. (2008). Likert Scale Likert Scale Examples How can you analyze data from a Likert Scale ? *Simply Psychology*, 1–3. https://www.simplypsychology.org/likert-scale.html

Microsoft. (2021). *Designer Bluetooth Mouse Microsoft Hardware - conducted on 15-04-2021*. https://www.microsoft.com/accessories/en-ww/products/mice/designer-bluetooth-mouse/7n5-00001

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81–97. https://doi.org/10.1037/h0043158

Monogram. (2021). *Monogram Creative Console - conducted on 16-04-2021*. https://monogramcc.com/shop/products/master-console

Mouser. (2021a). *2717 Adafruit - conducted on 06-07-2021*. https://eu.mouser.com/ProductDetail/adafruit/2717/?qs=FXKxjuMTmBk%252bXQsxVhda7A%3D%3D&countrycode=DE&currencycode=EUR

Mouser. (2021b). *CFR-25JT-52-220R - conducted on 06-07-2021*. https://eu.mouser.com/ProductDetail/yageo/cfr-25jt-52-220r/?qs=KUIzHt%2Fe91lrctWTReofaw%3D%3D&countrycode=DE&currencycode=EUR

Mouser. (2021c). *CFR16J4K7 - conducted on 06-07-2021*. https://eu.mouser.com/ProductDetail/te-connectivity/cfr16j4k7/?qs=DDevMFOh4stbW1%2FdnOryBQ==&countrycode=DE&currencycode=EUR

MP Midi. (2021). *MP Midi Controller - conducted on 16-04-2021* (Vol. 0, p. 25). https://mpmidi.com/

Musicradar. (2016). *Ableton Push 2 review*. https://www.musicradar.com/reviews/tech/ableton-push-2-632478

Nayanathara, R. (2020). *Pact Analysis*. https://bootcamp.uxdesign.cc/pact-analysis-3ac5fbe8817

Otronic. (2021). *Fader potmeter 10kOhm - conducted on 06-07-2021*. https://www.otronic.nl/a-60958306/weerstanden/slider-fader-potmeter-2-kanaals-75mm-10ko/?gclid=CjwKCAjw_o-HBhAsEiwANqYhp_EsoZZEAj59WXQkc0U5OL-lOSc5KyRw4D8UDISkU9-c68IApCBuwxoCheAQAvD_BwE

Paas, F. (1992). Training Strategies for Attaining Transfer of Problem-Solving Skill in Statistics: A Cognitive-Load Approach. *Journal of Educational Psychology*, *84*(4), 429–434. https://doi.org/10.1037/0022-0663.84.4.429

Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational Psychologist*, *38*(1), 1–4. https://doi.org/10.1207/S15326985EP3801_1

Pierce, D. (2017). *Steve Lacy Produced That Hot Kendrick Lamar Track Using Only His iPhone*. https://www.wired.com/2017/04/steve-lacy-iphone-producer/

Production Music Live. (2018). *Top 10 Most Popular VST Instruments*. https://www.productionmusiclive.com/blogs/news/top-10-most-popular-vst-synthesizers

ProductPlan. (2020). What is MoSCoW Prioritization? | Overview of the MoSCoW Method. In *ProductPlan*. https://www.productplan.com/glossary/moscow-prioritization/

Prusa. (2021). *Kit Original Prusa i3 MK3S - conducted on 13-07-2021*. https://shop.prusa3d.com/es/impresoras-3d/180-kit-original-prusa-i3-mk3s.html

Redifer, J. L., Bae, C. L., & DeBusk-Lane, M. (2019). Implicit Theories, Working Memory, and Cognitive Load: Impacts on Creative Thinking. *SAGE Open*, *9*(1). https://doi.org/10.1177/2158244019835919

Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational Psychologist*, *38*(1), 15–22. https://doi.org/10.1207/S15326985EP3801_3

Roseke, B. (2019). *3 Types of Stakeholder Matrix*. https://www.projectengineer.net/3-types-of-stakeholder-matrix/

Sparkfun. (2015). I2C Tutorial. In *Web Page*. https://learn.sparkfun.com/tutorials/i2c

Special Waves. (2021). *Mine S - conducted on 16-04-2021*. https://special-waves.com/

Steam. (2011). Portal 2 - conducted on 07-05-2021. In *Steam*. https://store.steampowered.com/app/620/Portal_2/

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257–285. https://doi.org/10.1016/0364-0213(88)90023-7

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review*, *22*(2), 123–138. https://doi.org/10.1007/s10648-010-9128-5

Sweller, J. (2018). Measuring cognitive load. *Perspectives on Medical Education*, *7*(1), 1–2. https://doi.org/10.1007/s40037-017-0395-4

Techmirtz. (2017). *Using Common Cathode and Common Anode RGB LED with Arduino*.

https://www.hackster.io/techmirtz/using-common-cathode-and-common-anode-rgb-led-with-arduino-7f3aa9

The MIDI Association. (2021). *Midi home - conducted on 07-07-2021*. https://www.midi.org/

Totilo, S. (2011). *The Excellent Beginning of Portal 2 - conducted on 07-05-2021*. https://kotaku.com/the-excellent-beginning-of-portal-2-5763043

Van Merriënboer, J. J. G., & Sweller, J. (2010). Cognitive load theory in health professional education: Design principles and strategies. *Medical Education*, *44*(1), 85–93. https://doi.org/10.1111/j.1365-2923.2009.03498.x

Von Merriënboer, J. J. G., & Aryres, P. (2005). Research on cognitive load theory and its design implications for E-learning. *Educational Technology Research and Development*, *53*(3), 5–13. https://doi.org/10.1007/bf02504793

Weirdly Wired. (2021). *Modsy Controller*. https://modsycontroller.com/

Widyanti, A., Johnson, A., & de Waard, D. (2013). Adaptation of the Rating Scale Mental Effort (RSME) for use in Indonesia. *International Journal of Industrial Ergonomics*, *43*(1), 70–76. https://doi.org/https://doi.org/10.1016/j.ergon.2012.11.003

XFerreccords. (2021). *Serum: Advanced Wavetable Synthesizer - conducted on 13-05-2021*. https://xferrecords.com/products/serum

Appendix A: The four load reducing methods presented to the participants of the user tests

Using Signals

Designers can use these guidelines by making a product that uses signals to point at important features of the product. The buttons on a remote have different sizes and colours, this helps the user to remember the importance of certain buttons.

Minimalistic design

Make a product as simple as possible without too many things on the side that make the product more complicated. The mouse on the left is very complicated and has a lot of features, this might increase the workflow for a professional however for a beginning user of the mouse it can be very hard to use because of its complexity. The mouse on the right is very simple and is less complex to use.

Provide mnemonics

Give the users hints on the functionality of the product. The letters on a keyboard provide the user with hints on the functionality of the keyboard. The user can simply look at the keyboard to know which button does what.

Pacing learning

Enable a user to learn how to use the product in parts. Using design patterns is also a way of pacing learning since the user learned the design pattern beforehand. The example below gives the user the option to add more components to the controller. This way the user can solely use parts of the controller and once these components are familiar with the components it is possible to add more parts.

Appendix B: Topview render of the Modsy

Figure 53: Current Modsy design

Appendix C: Pictures of group brainstorm outsiders

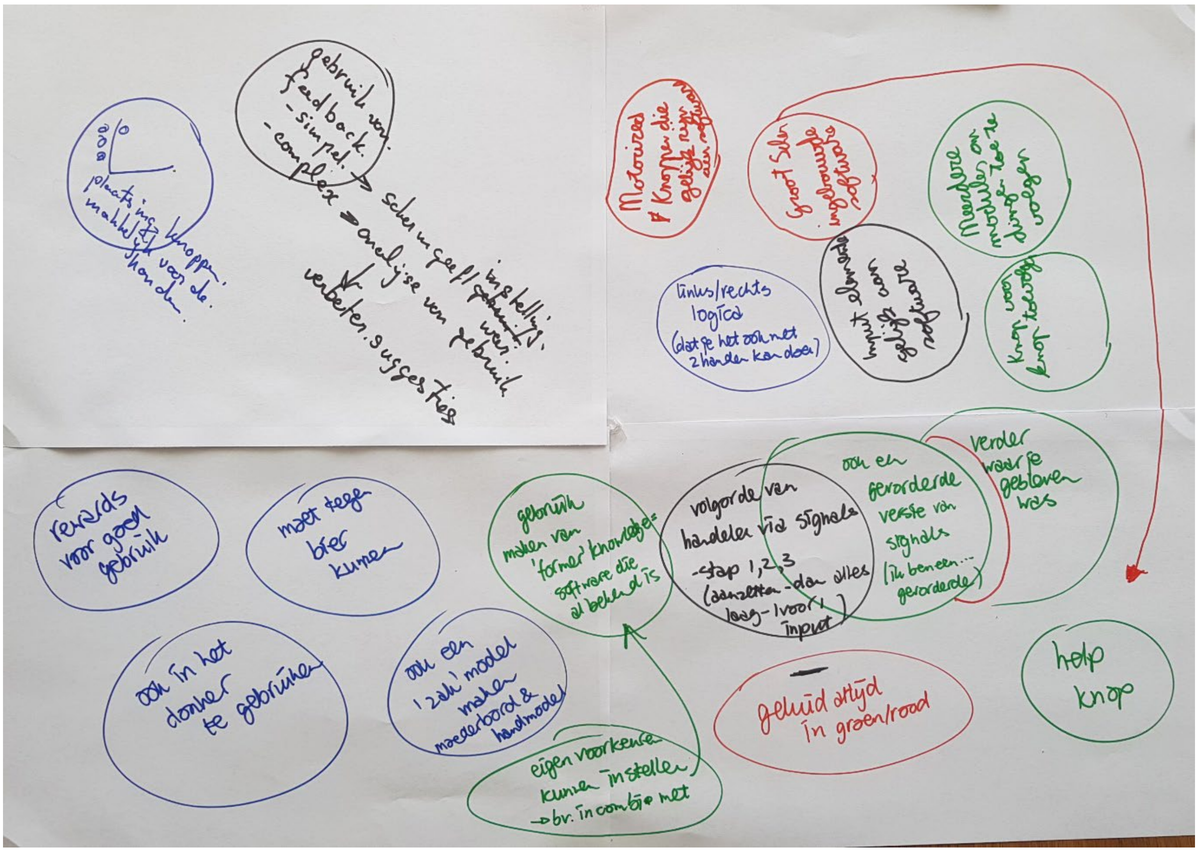

Figure 54: Dreaming phase of outsiders group brainstorm

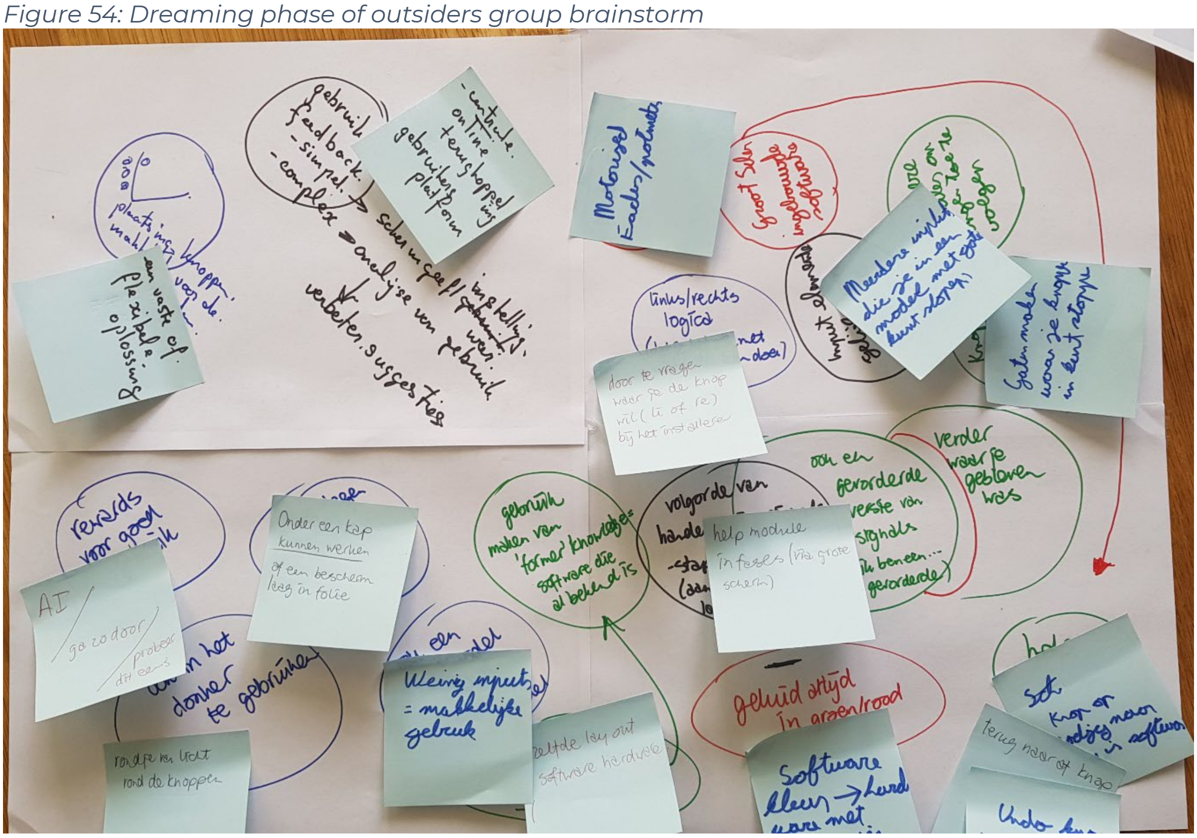

Figure 55: Realisation phase of outsiders group brainstorm

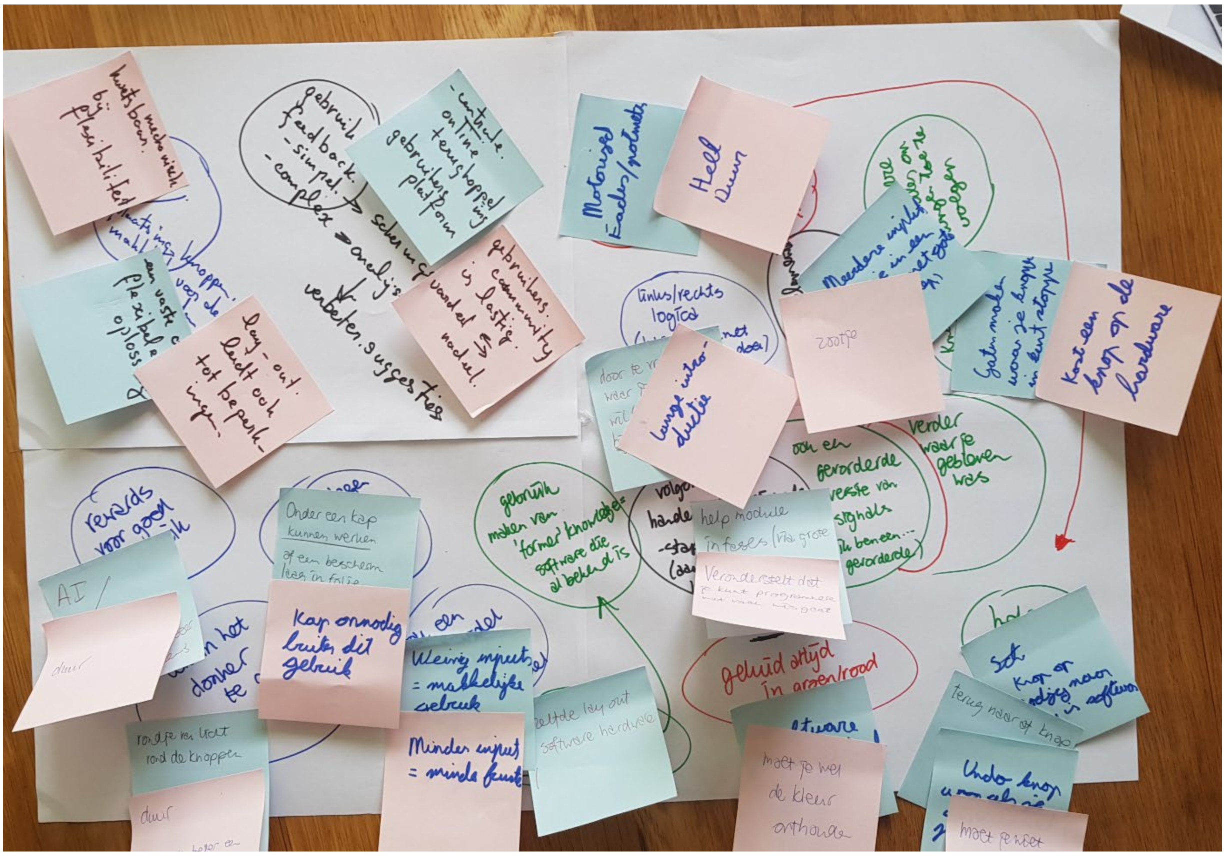

Figure 56: Criticising phase of outsiders group brainstorm

Appendix D: Pictures of group brainstorm Weirdly Wired

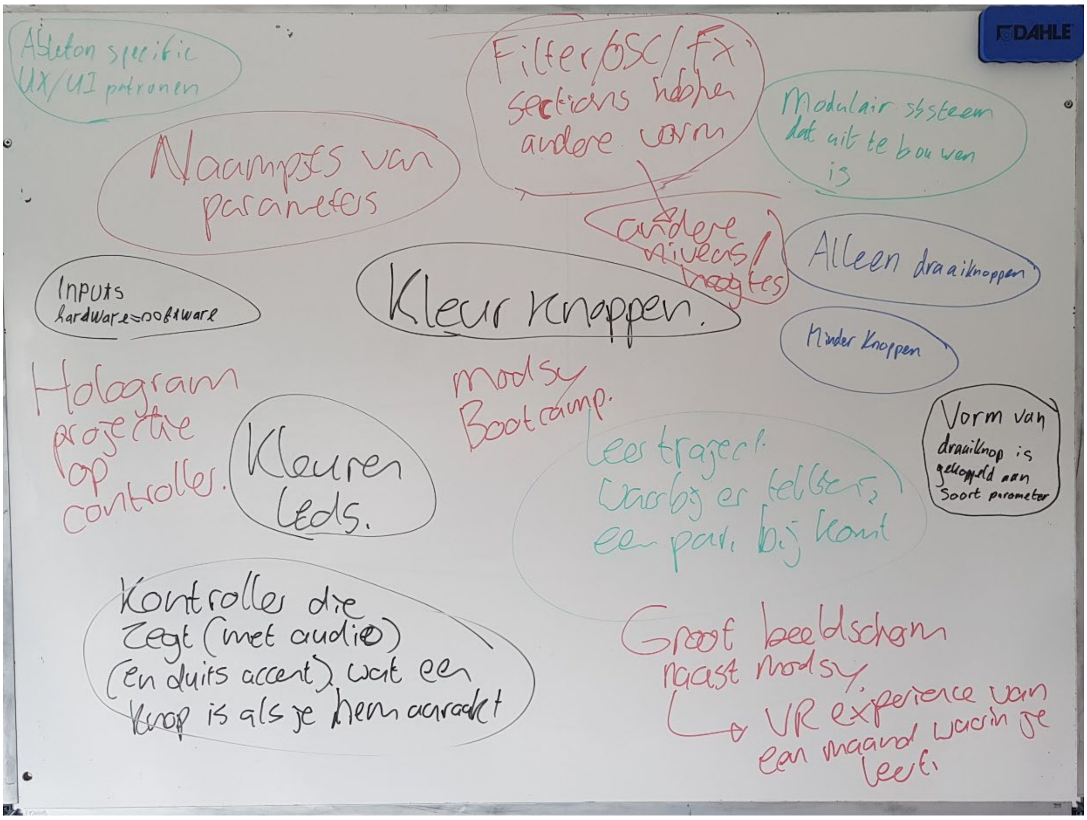

Figure 57: Dreaming phase of Weirdly Wired group brainstorm

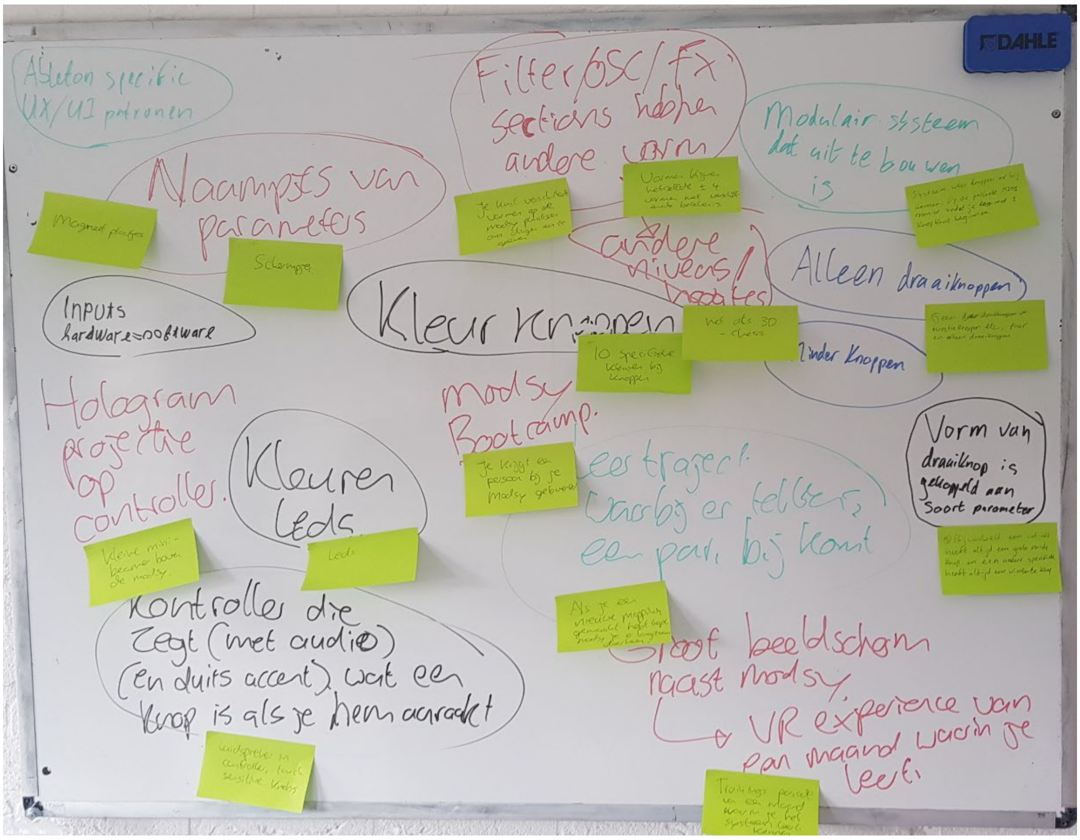

Figure 58: Realizing phase of Weirdly Wired group brainstorm

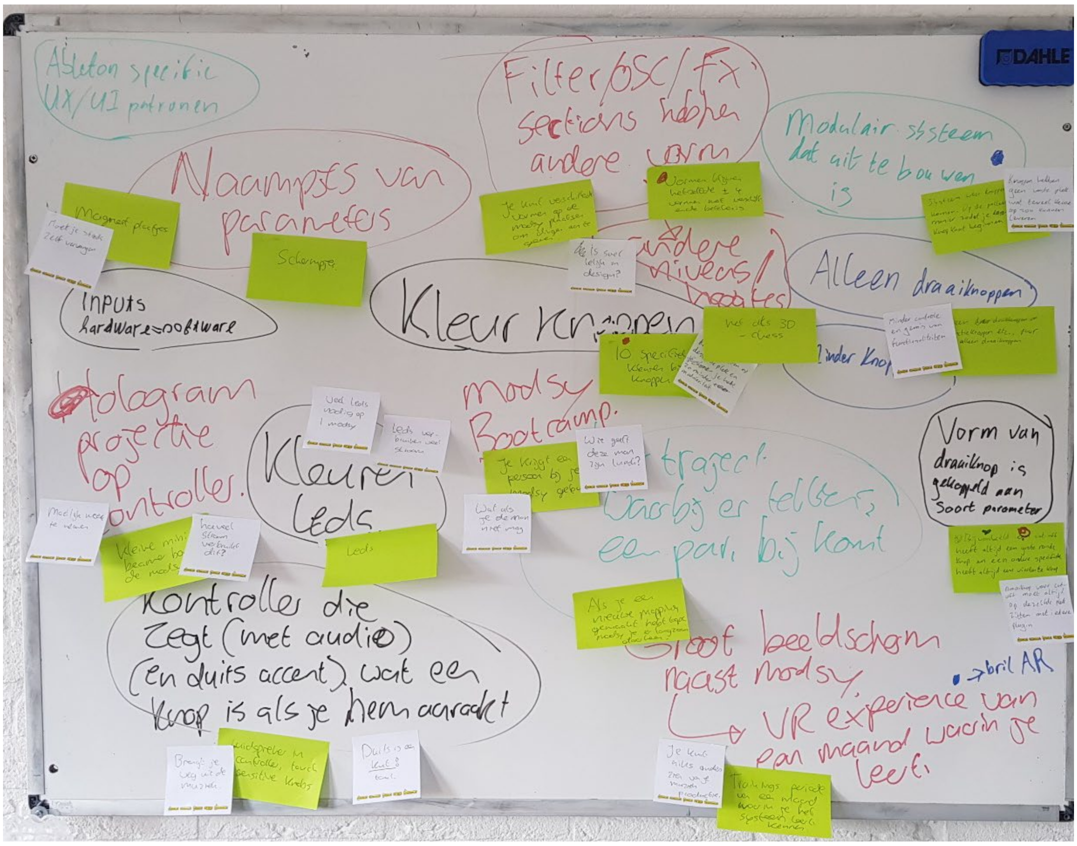

Figure 59: Criticising phase of Weirdly Wired group brainstorm

Appendix E: Pin layout Arduino

Component	Arduino Pin
Led - 1 - Red	2
Led - 1 - Green	3
Led - 1 - Blue	4
Led - 2 - Red	5
Led - 2 - Green	6
Led - 2 - Blue	7
Led - 3 - Red	8
Led - 3 - Green	9
Led - 3 - Blue	10
Led - 4 - Red	11
Led - 4 - Green	12
Led - 4 - Blue	13
Led - 5 - Red	14
Led - 5 - Green	15
Led - 5 - Blue	16
Led - 6 - Red	17
Led - 6 - Green	18
Led - 6 - Blue	19
Led - 7 - Red	23
Led - 7 - Green	24
Led - 7 - Blue	25
Led - 8 - Red	26
Led - 8 - Green	27
Led - 8 - Blue	28
Fader 1	A0
Fader 2	A1
Fader 3	A2
Fader 4	A3
Fader 5	A4
Fader 6	A5
Fader 7	A6
Fader 8	A7
Multiplexer SDA	20
Multiplexer SCL	21

Table 13: Pin layout Arduino

Appendix F: Pin layout multiplexer

Component	Multiplexer pin
Screen - 1 - Data	SD0
Screen - 1 - Clock	SC0
Screen - 2 - Data	SD1
Screen - 2 - Clock	SC1
Screen - 3 - Data	SD2
Screen - 3 - Clock	SC2
Screen - 4 - Data	SD3
Screen - 4 - Clock	SC3
Screen - 5 - Data	SD4
Screen - 5 - Clock	SC4
Screen - 6 - Data	SD5
Screen - 6 - Clock	SC5
Screen - 7 - Data	SD6
Screen - 7 - Clock	SC6
Screen - 8 - Data	SD7
Screen - 8 - Clock	SC7

Table 14: Pin layout multiplexer

Appendix G: Arduino Code

```
/**************************************************************************
  This program was made by Bram van Driel for the graduation project for
Creative Technology
  The last edit to this program was made on 29/06/2021

  The parts of the program that include changing the screens are based on
code from Adafruit Industries
  Large parts of this program are based on code written by Weirdly Wired
  Weirdly Wired has permitted to reuse this code for this project.

**************************************************************************/

//including the needed libraries
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <MIDI.h>

//setting up the screens
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET     4
#define SCREEN_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define TCAADDR 0x70

// Create and bind the MIDI interface to the default hardware Serial port
MIDI_CREATE_DEFAULT_INSTANCE();

//the value of the faders is updated only if it exceeds a certain
threshold.
int threshold = 3;
//to check for the threshold the previous value of the fader needs to be
saved
//this is updated once the new value has a difference of more than the
threshold.
int prevVal[6];

//the faders are connected to the following analog input pins.
int faderPorts[] = {1, 2, 3, 4, 6, 5};

//Not all the screens are connected, this array keeps track of which are
connected.
int connectedScreens[] = {1, 2, 3, 4, 6, 7};

//Initialise the MIDI channels that will be used to send the information
int MIDIChannels[] = {102, 103, 104, 105, 106, 107};

//Here the on-screen names are determined
String screenNames[] = {"Decay", "Sustain", "Release", "Attack", "Level",
"LFO"};

//Delay Code for screens
int period = 500;
unsigned long time_now = 0;
bool valueTimer = false;
```

```
void setup() {
  Wire.begin();
  MIDI.begin();
  Serial.begin(9600);

  //initiate the displays
  for (uint8_t t = 1; t < 8; t++) {
    if (t != 5) {
      tcaselect(t);
      display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    }
  }
  setDisplayNames();
  setColor(1, 255, 0, 0);
  setColor(2, 255, 0, 0);
  setColor(3, 255, 0, 0);
  setColor(4, 255, 0, 0);
  setColor(7, 0, 255, 255);
  setColor(8, 0, 255, 255);
  for (int i = 0; i < 6; i++) {
    prevVal[i] = autoMapValue(analogRead(faderPorts[i]));
  }
}

void loop() {
  updateFaders();

  if (valueTimer == true) {
    checkTimer();
  }
}

//Here the input of the faders is read and the systems updated accordingly
void updateFaders() {
  for (int i = 0; i < 6; i++) {
    int currentVal = autoMapValue(analogRead(faderPorts[i]));
    if (abs(currentVal - prevVal[i]) > threshold) {
      prevVal[i] = currentVal;
      MIDI.sendControlChange(MIDIChannels[i], currentVal, 1);
      tcaselect(connectedScreens[i]);
      drawProgressbar(currentVal);
      valueTimer = true;
      time_now = millis();
    }
  }
}

//Used for selecting the correct screen
void tcaselect(uint8_t i) {
  if (i > 7) return;
  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
}

//Write a text to the screen
void setScreenText(String text) {
  display.clearDisplay();
  display.setCursor(10, 0);
  display.println(text);
  display.display();      // Show initial text
```

```
}

//The colour of one LED can be changed using this method.
void setColor(int LED, int red, int green, int blue)
{
  //Set the according pins to the wanted color
  int redPin = 3 * LED + 2;
  int greenPin = 3 * LED + 3;
  int bluePin = 3 * LED + 4;

  //since using common anode the values need to be reversed.
  red = 255 - red;
  green = 255 - green;
  blue = 255 - blue;

  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}

//Here the displays are said to display to correct name
void setDisplayNames() {
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  for (int i = 0; i < 6; i++) {
    tcaselect(connectedScreens[i]);
    setScreenText(screenNames[i]);
  }
}

//The incoming analog values rage from 0 to 1023. For Midi these should be
0 to 123
int autoMapValue(int analogValue) {
  int mapperoutput;

  mapperoutput = map(analogValue, 0, 990, -1, 128);

  if (mapperoutput < 0) {
    return 0;
  } else if (mapperoutput > 127) {
    return 127;
  } else {
    return mapperoutput;
  }
}

//This is for drawing the Progressbar of the parameters
void drawProgressbar(int progress) {
  int xpos_bar = 0;
  int ypos_bar = 0;
  int width_bar = 120;
  int height_bar = 30;

  int percentage = int(map(progress, 0, 127, 0, 100));
  //Calculate the bar value for the value of the parameter
  float bar = ((float)(width_bar - 1) / 100) * percentage;

  display.clearDisplay();

  //For the display bar
  display.drawRect(xpos_bar, ypos_bar, width_bar, height_bar, WHITE);
```

```
  display.fillRect(xpos_bar + 2, ypos_bar + 2, bar , height_bar - 4,
WHITE); // initailize the graphics fillRect(int x, int y, int width, int
height)

  //For the progress text
  if (percentage >= 0) {
    display.setCursor((width_bar / 2) - 10, ypos_bar + 5 );
    display.setTextSize(2);
    display.setTextColor(WHITE);
    if ( percentage >= ((width_bar / 2) - 20)) {
      display.setTextColor(BLACK, WHITE); // 'inverted' text
    }
    display.print(percentage);
    display.print("%");
    display.display();
  }
}

//For changing the screen from progressbar to the name
void checkTimer() {
  //If more than period seconds have past since the last value change it
goes back to showing the screen.
  if (millis() > time_now + period) {
    valueTimer = false;
    setDisplayNames();
  }
}
```

Appendix H: Information brochure

Additional controller Controller user test

Background

One very important aspect of products is whether it is easy for users to use the product. An important factor is the cognitive load of the product. The cognitive load can be seen as the working memory of a person. Following the theory about creative thinking of Redifer et al. (2019), it might be beneficial for musicians to have the most working memory available for the creation of music. Therefore designing a product with a low cognitive load can be beneficial for the process of making music. This research looks at how to reduce the cognitive load of the Modsy controller to increase the overall user experience of the Modsy controller.

Research procedure

Through the means of a user test, feedback will be gathered to test and improve the prototype made for this research.
In this letter, it is explained what it means for you to take part in this research. You can decide whether to partake in this research. If you have any questions, don't hesitate to contact Bram van Driel (b.j.vandriel@student.utwente.nl).

Participation

Participation is entirely voluntary and you can quit this research at any moment without stating the reason why. Permission for participation only has to be granted once. The total duration of the test will be about half an hour(30 minutes).

What happens during the test?

As mentioned, this user test is meant to gather feedback and to test the prototype. You will be asked to make music using a music controller and to give feedback about the product. During the test, you can ask questions if there are things unclear about how to use the controller.

During the test, you will be asked to test two setups: one set up with an additional fader controller and one setup without the fader controller. Both setups can be used to control the same instrument however the setup with the extra faders offers more control.

After the test, several questions will be presented to you to analyse what you thought about the controller. The goal of these questions is to see whether the current prototype shows the possibility of decreasing the cognitive load and whether it might be possible that this prototype increases the overall experience of making music. Please note that this test is only meant for finding out whether the current prototype has the potential of decreasing the cognitive load and increasing the user experience, this test is not meant to show any significant proof.

Which data will be collected?

The questions that are answered at the end of the test will be stored. There will not be any personal data among this. There will no data be stored from the input you give to the controller. The questioned that are asked to you after the test will either be using a digital form or an interview. In case of an interview written notes will be used to store the data.

How will the data be stored?

The data will be stored and anonymously processed according to the GDPR guidelines. All data will be stored for a minimum of two months after completion of Bachelor Thesis, but in suiting cases for an undetermined time, appropriate to the current guidelines from the Vereniging van Universiteiten (VSNU).

Who has access to this data?

The answers to the questions will be only accessible for people that are involved in this research. A list of names of people that have access to this data can be requested from Bram van Driel.

How will the data be used?

The data that is being gathered during the test will be analysed and used for a Bachelor Thesis. The results of this test will be used to make claims or to reinforce claims and validate the prototype. Anonymous interview quotes or data from the interview can be used for this Bachelor Thesis.

Will any personal data be made public?

No

More information and independent advise

Would you like independent advice about participating in this research or do you have a complaint? Then you can contact the Ethics Committee (ethicscommittee-cis@utwente.nl). This committee consists of independent experts from the University of Twente and is available for questions and complaints regarding this research.

For any questions, you can contact the researcher, Bram van Driel (b.j.vandriel@student.utwente.nl) or the supervisor Wouter Eggink (w.eggink@utwente.nl).

References

Redifer, J. L., Bae, C. L., & DeBusk-Lane, M. (2019). Implicit Theories, Working Memory, and Cognitive Load: Impacts on Creative Thinking. SAGE Open, 9(1). https://doi.org/10.1177/2158244019835919

Appendix I: Consent form

INFORMED CONSENT

About

The University of Twente researches cognitive load design implications for the Modsy controller. You permit to participate in a user test in which statements and information can be used for this research. More information about this research can be found in the information brochure.

Lead researchers:

Bram van Driel (Supervisor: Wouter Eggink)

Contactinformatie

For any questions, you can contact the researcher that is present at the user test, Bram van Driel (b.j.vandriel@student.utwente.nlmailto:) or the supervisor Wouter Eggink (w.eggink@utwente.nl). You can also contact the Ethics Committee (ethicscommittee-cis@utwente.nl). This committee consists of independent experts from the university and is available for questions and complaints regarding this research.

Research: Cognitive load design implications for the Modsy controller

☐ I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions or perform the tasks and I can withdraw from the study at any time, without having to give a reason.

☐ I understand the reason for this research, methods are explained and I have had the option to ask questions.

☐ I hereby give upfront consent to participate in this research

☐ I permit using my statements during this user test in text for research purposes

A list of names of people that have access to this data can be requested from Bram van Driel. All data will be stored for a minimum of 10 years, but in suiting cases for an undetermined time, appropriate to the current guidelines from the Vereniging van Universiteiten (VSNU).

Date: Place:

Name: Paragraph participant:

Appendix J: Raw interview

Test 1

- Ging er zelf direct vanuit dat de sliders van Juno ook de sliders waren op de hardware
- Enorm logisch in gebruik
- Heel tof dat er nu meer ruimte is voor andere dingen
- De sliders hebben goede weerstand

Test 2

- Ik kan me goed voorstellen dat het gebruik van faders logischer kan zijn als het op de VST ook zo is.
- Zou tof zijn als het motorized is dan is het nog makkelijker in gebruik.

Test 3

- Onbewust is de ervaring misschien toch beter met sliders al helemaal voor dingen als de ADSR omdat die toch vaak in de software is.
- Ik denk dat de faders een goede toevoeging zijn als de ADSR er altijd naar gemapped is. Dan weet ik dat die parameters daar zitten.
- Het is fijn in gebruik dat je de waveform kan zien van de ADSR. Daar worden sliders eigenlijk altijd voor gebruikt dus het is logisch dat dat nu ook zo is.

Test 4

- Ik vind het sowieso een heel tof product.

Test 5

- Alles in 1 controller is makkelijker
- Als de functies hetzelfde zijn maakt het voor mij niks uit
- De manier waarop ik Juno zou gebruiken maakt het niets uit of het sliders.
- Ik kan me voorstellen dat het heel chill is als je veel met de ADSR bezig bent en die is daar altijd naar gemapped dat het logisch.