Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Jenny Marie Fristad

Gamified Mechatronic Training System

Master's thesis in Ingeniørvitenskap og IKT
Supervisor: Amund Skavhaug
July 2023

NTNU
Norwegian University of
Science and Technology

Jenny Marie Fristad

Gamified Mechatronic Training System

Master’s thesis in Ingeniørvitenskap og IKT
Supervisor: Amund Skavhaug
July 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Norwegian University of
Science and Technology

TPK4960 - ROBOTICS AND AUTOMATION, MASTER'S THESIS

Gamified Mechatronic Training System

Author:
Jenny Marie Fristad

Supervisor:
Amund Skavhaug

2023-07-17

Preface

With a background in mechanical engineering and ICT solutions, my wish was to combine these two fields in a way that could be beneficial to others. Learning 3D printing and further deepen the knowledge about graphical user interface programming were also concrete goals for the thesis. In addition, with an understanding of how games can influence behavior and motivation, as well as a passion for creating physical devices, the focus quickly became exploring how game design concepts could be incorporated into a physical device to enhance an existing solution.

A preliminary study was conducted prior to this thesis. The main goal of this study was establishing requirements, and creating design suggestions for a mechatronic system working as an exercise tool for fine motor skills. The system was explored based on research on fine motor skills along with concepts related to gamification. This thesis further delves into this system and focuses on prototyping one of the solutions presented in this study.

I would like to thank my supervisor Amund Skavhaug for his guidance and help during this project. A thank you is also in order to Håkon Jarand Dugstad Johnsen and Lars Tingelstad for lending me an Arduino Uno board and a Raspberry Pi. They were also a huge help with giving insight and tips to the boards, and the field of mechatronics in general. Last, but not least, I would like to thank Håvard Vestad for his help in laser cutting the needed parts.

Summary

This thesis describes the creation of a prototype that works as a gamified mechatronic training system for exercising cognitive and muscular skills. The work is based on a preliminary literary study where design solutions to the system are justified, discussed, and presented. In this thesis, the designs from the prestudy are brought to life by implementing electronic components combined with a physical box that works as a training system. The prototype is made to benefit those that struggle with fine motor skills and those that could benefit from exercising and maintaining these skills. In addition, by implementing gaming logic the goal is also to further develop, or exercise logical thinking.

A game is therefore made with three levels that increase in difficulty, giving the user specific tasks that make exercising fun. This game is made as an enhanced version of the children's toy where you put different shapes into corresponding holes. The box and its physical components are customized by using 3D printing and laser cutting techniques. A display with a graphical user interface showing the game is made by sending sensor values from an Arduino Uno located within the box underneath the board with holes to the display via a Raspberry Pi. This is done in order to register the user's moves and provide correct feedback per the tasks given.

The prototype is also evaluated using a qualitative usability test mixed with inspection methods in a controlled environment. The goal is to say something about how the system feels in use and how it can be improved for further testing. Even though missing functionality and faults were discovered, feedback from the participants concludes that the system was intuitive and responsive, making it a good concept for further development.

Sammendrag

Denne oppgaven tar for seg prossessen av å lage en prototype som fungerer som et spillifisert mekatronisk system for å trene kognitive og muskulære ferdigheter. Arbeidet er basert på et litterært forstudie hvor designforslag til systemet ble argumentert, diskutert og presentert. I denne oppgaven blir tegningene fra dette forstudiet bragt til live ved å implementere elektroniske komponenter i en fysisk boks som fungerer som et treningssystem. Prototypen er laget for å være til nytte for de som sliter med finmotoriske evner og de som kan ha nytte av å trene, og opprettholde de. I tillegg, ved å implementere spillogikk er målet å videre utvikle og trene logisk tenking.

Et spill er derfor laget med tre ulike nivåer som øker i vanskelighetsgrad, disse gir brukeren spesifikke oppgaver som skal gjøre treningen mer gøy. Spillet er laget som en mer avansert versjon av den populære barneleken hvor man putter ulike klosser i hull. Selve boksen og dens fysiske komponenter er laget og tilpasset ved å bruke 3D printe- og laserkutting teknikker. Et display med et grafisk brukergrensesnitt som viser spillet er laget ved å sende sensor verdier fra en Arduino Uno under et brett med hull, til en Raspberry Pi. Dette gjøres for å registrere brukerene sine trekk, og for å gi riktige tilbakemeldinger tilknyttet de spesifikke oppgavene.

Prototypen er også evaluert ved å gjennomføre en kvalitativ brukertest blandet med inspeksjons metoder i en kontrollert setting. Målet er å kunne si noe om hvordan systemet føles i bruk, og hvordan det kan forbedres til videre testing. Til tross for at manglende funksjonalitet og feil var oppdaget under testing, var tilbakemeldingene fra testbrukerne at systemet var intuitivt og responsivt. I tillegg ble det konkludert med at systemet er et godt konsept for videre utvikling.

Contents

List of Figures

List of Tables

Listings

Chapter 1.

Introduction[1]

1.1. Motivation

This thesis is motivated by the use of mechatronics to make a difference and improve the everyday lives of individuals that struggle with independence due to a handicap. In particular, it explores the potential of a mechatronic system that can work as a new and innovative way of cognitive and muscular learning for those that struggle with fine motor skills and make training fun, rewarding, and motivating in a creative and entertaining way. To achieve this, concepts from mechanical toys and principles from computer games, such as skill level and challenges, are combined to create a gamified training system.

Making training fun by combining it with games is, however, not a new concept. Early attempts at this can be traced back to the 1980s with HighCycle by Autodesk and the Puffer by Atari [71][Figure 1.2]. However, it was Nintendo who truly made a breakthrough in the early 2000s with their Wii Sports and Wii Fit [Figure 1.1] games [71]. They introduced a fun and easy way to train at home and it was wildly popular, with Wii Sports, Wii Fit, and Wii Fit Plus having sold roughly around 120 million units combined around the world today [91]. Nevertheless, while these games simulate exercise they are not a replacement for the real thing. It has, however, been shown that energy expenditure during active play could be comparable to moderate-intensity walking [29]. Making these games a fun way to promote exercise, and a clever way of using principles from computer games to re-invent exercising, which may make it more appealing to different people.

[1]Parts of this chapter are based on the author's own prestudy [49].

Figure 1.1.: Wii Fit [1].

Figure 1.2.: Puffer by Atari [12].

Although these games and the mechatronic system presented in this thesis are very different in ways of what they want to achieve, it is the underlying concept of mixing activity with technology they have in common.

Mechatronics is the term used to describe an interdisciplinary field that combines mechanical, and electronic components to develop intelligent devices and systems that can perform a wide range of tasks. As a consequence of Moore's law [21], which predicts that the processing speed and capability of computers steadily increase with time, a general demand for automation in various industries today is growing. Mechatronics has therefore emerged as one of the fastest-growing fields in engineering, with robotics being a sub-discipline in very high demand. As a result, mechatronic devices are found more and more in everyday life, from robotic lawnmowers to anti-lock braking systems (ABS) in cars. These two wildly different examples show just a fraction of the endless possibilities for innovation the field of mechatronics has to offer, and how it can better our lives in very different ways.

1.2. Problem description

The main objective of this thesis is to create a prototype demonstrating a system for exercising cognitive and muscular skills for those that could benefit from such training. The goal is to specifically target fine motor skills and logical thinking in a fun way by creating a mechatronic gamified system. The prototype is based on the work done in a preliminary study [49] done prior to this thesis. Whereas the prestudy merely presents a conceptual system, the goal of this thesis is to bring the drawings and discussions to life by looking at more specific solutions and developing a functional prototype as a proof of concept. This includes identifying hardware and software, designing a graphical user interface (GUI), creating blocks, and putting the different parts together to create a cohesive system. The system should also be made easy to use, where the goal is to create a system for anyone without the need for specialized knowledge, rare equipment, or a substantial amount of money.

The main focus during prototyping is to make the system per the requirements developed by the author in their own prestudy. These requirements are summarized in section 2.3. Limitations to these requirements are however implemented to make the workload adapted to the limited time available. As the goal is to have a functional prototype, only the necessary parts for creating a working system are made, for instance only one board and a limited number of blocks. In addition, to showcase the ability of the board and its potential, no more than three distinct levels with increasing difficulty are needed.
To further limit the workload, and have the work focus on the objectives, the system is made with a specific user group in mind to better adapt the size and the features of the system to something more specific. For this prototype, the system is adapted to children.

In order to evaluate the success of these requirements, determine how the prototype can be improved further, and get an understanding of how it is to use, an objective for the thesis is consequently also to conduct user test evaluations.

1.3. Reading guidance

Subsequent to this introduction and the objectives of the thesis, a background chapter (**Chapter 2**) follows where the preliminary study and its findings are summarized. This chapter includes an overview of who the system is intended for, previously made systems that also targets fine motor skill, the requirements for the intended system, and lastly a summary of the discussion from the prestudy. These parts are meant to give a better understanding of how and why this system is further developed, and work as a guide for prototyping.
After this, some theory needed to understand how the system is made is presented in **Chapter 3**. Concepts like Gamification, Additive manufacturing, embedded systems, and GUI programming are explained, along with how it is used and why. The different electronic components are also presented here along with their abilities and features.
Chapter 4 dives into the work done in this thesis. In this chapter everything that has been done is explained and discussed in detail, with the blocks, box, and display. The methods that have been used are explained and justified, and the results of each part are presented.
The next chapter (**Chapter 5**) describes the evaluations that were carried out with the system. The methodology of the evaluation is presented along with an explanation of how they were conducted and who participated. The results are presented in two tables, problems that occurred are in one, and suggested solutions with a prioritized order of execution in the other. These results are then

discussed more in detail and the different problems are categorized. After that, a summary of the evaluations of how it achieved its goals is briefly discussed. Lastly, some improvements done to the system after the evaluations are presented and explained.
The last chapter in this thesis (**Chapter 6**) is a general discussion of how the entire process has been and includes reflections on how it could have been done better. In addition, a section covering potential future work is also included, before finishing with a conclusion of how the finished prototype is in accordance with the objectives of the thesis.

Appendix A is the script made for each evaluation, where the whole process is written down in order to make sure the experience is similar to each participant.
Appendix B is a collection of supplementary figures included for illustrative purposes.
Lastly, **Appendix C** includes parts of the code made for the system.

Chapter 2.

Background[1]

In this chapter, the most important parts from the preliminary study are summarized to further give a better understanding of the intended system. The ideas and discussions presented in this chapter are the overall goals for the system, and what the prototyping process is based on.

2.1. Who the system is intended for

The target user group for the system presented in this thesis is those that struggle with fine motor skills. Motor skill or motor control is the ability to move your body in specific ways in order to perform certain tasks. It can be divided into two different subgroups, fine and gross motor skills. Gross refers to when the larger muscles are in use, for example when jumping, walking, or waving. Fine motor skill is smaller more controlled movements in hands [96]. Having trouble with either one of them can have a significant impact on people's daily lives and how much assistance one might need to perform simple tasks. It is therefore very favorable to try to prevent deterioration or improve motor skills as it can help people become more independent. It can however be hard to aid everyone with the same tools, as the aid has to be adapted to different age groups and needs.

The development of motor skills starts already in infancy and further develops during childhood. This time is critical for both physical and cognitive growth and research have shown several links between the development of fine motor skill and other abilities. For example, it has been shown that fine motor skills can have a direct effect on mathematical performance [94], and difficulty with motor control in general can also affect a child's popularity amongst peers, and their self-esteem [39]. The importance of developing fine motor skills at an early stage can therefore

[1] Parts of this chapter are based on the author's own prestudy [49]

help children with their cognitive learning, and help them grow and adapt better through childhood.

Problems with fine motor skill is, however, not only found amongst children. Adults may experience damage to their fine motor skills at a later point in life. This can be due to an accident, injury, disease, age-related regression [63], or it can be an indication of certain neurodegenerative diseases [27]. Luckily it is possible to improve affected skills and develop them again with exercises specific to fine motor skills and the individual's skill level [65]. This can for example be, drawing, folding clothes, or using scissors as these activities target the smaller muscles in the hands in different ways. But doing activities you used to excel at before, with much struggle now can be very demotivating and feel like a perpetual project. Instead, giving other types of exercises where you can later see the effects of them in daily activities can be very rewarding and give motivation for continued training. The new system presented in this thesis achieves this by making training different and fun while targeting fine motor skills and training them with its use. The goal is to make the user think of exercising their fine motor skill as something they want to do, rather than something they have to do, which again could contribute to the overall motivation for its use.

As this shows there are many different people, of almost all ages, who could benefit from a system like the one presented in this thesis. The problem is making a system that can be beneficial to everyone. Fine motor skill is in itself a very general skill but it is how much each individual struggle to use that skill that defines the purpose and execution of exercises. A child just beginning to show signs of late development have more control over their fine motor skill than people who just suffered a stroke for instance. The motivation for exercising and using the system also varies, as a 40-year-old may not be motivated by the same things as a 1-year-old. But by making the system as general as possible without compromising its intended use it is however possible to make the system appealing to many different people at the same time, which is discussed and looked at later in this thesis.

2.2. Previously made systems

When it comes to systems that exercise motor skills there exist several different products on the market today. You have everything from physical games and digital solutions to a mix of them. To properly get an understanding of how a new system can be made, the individual benefits of already existing solutions are looked further into.

As mentioned in section 2.1, it is children who can benefit the most from training fine motor skills in the long run, as it would not only benefit their physical aspects but also their cognitive development. That is why there exist several different systems that are specifically designed for them on the market today.

Figure 2.1 and 2.2 show two examples of different toys that have been made for children. They are made with the intention that the skills gained by them are transferable to everyday activities that require fine motor skills. Take for instance the lacing cards toy [Figure 2.1], the purpose here is to move a string in and out of holes on a board or an object. This requires a lot of control and is the same skill used when lacing shoes. So by using this toy, a child could be able to learn the basics of tying their shoelaces without it being a complicated task. The same goes for the Peg toy [Figure 2.2] where having to be precise in the placing of the blocks, the user challenges their fine motor skills. The toys manage to take a rather complicated activity and make it simpler in order for the children to train their fine motor skills at their own level.

Figure 2.1.: Lacing cards toy [2].

Figure 2.2.: Peg toy [3].

Figure 2.3 shows one of the more popular toys when it comes to the training of fine motor skills for children. Here you are supposed to grab different shapes and get them into the corresponding holes. By doing this, a child can train their fine motor skills, but also their cognitive skills as they have to use their logic to match one shape to a hole and figure out how to put them in. With it being a specific toy for fine motor skill and also having the ability to benefit other areas, it is inspiring and can be a solid foundation for further development, which is what this thesis show.

Figure 2.3.: Box toy [4].

Other types of toys and games that are not necessarily made with the purpose of training, but rather for pure enjoyment also exist on the market today. Many of them can however be used as tools for exercising fine motor skills, for example, puzzles. They are made to be a fun activity and something challenging for people to do together. But with pieces being very small it requires fine motor skill to pick them up and put them in the right places. This way you are able to exercise or maintain your fine motor skills whilst having a good time.

Even though these types of games are not intended as exercise tools they still highlight the concept mentioned earlier with the Nintendo games [section 1.1, Figure 1.1], being able to enjoy something without realizing that you are exercising. In other words, by transforming the exercise into more of a game you could be able to take the focus of exercising away.

Even though these physical objects, or toys, are very beneficial to training fine motor skills, we live in a digital era where smartphones or tablets are, in Norway, introduced to children already in primary school [56]. The toys presented above are almost too old-fashioned for the world we live in today, where many hobbies or replacements for physical activities are found online or digitally. There exist for example many different apps you can download on your phone or tablet that mimic activities for fine motor skill exercises, puzzle apps for instance.

However, when it comes to some of these, they cannot substitute the real thing, similar to that of the Nintendo games [section 1.1, Figure 1.1]. Take for instance a puzzle app. In reality, you can grab the pieces and move them around. On a tablet, all you do is drag the piece from one place to another using just one finger. Even though that still requires some fine motor skills it does not give the same benefits as doing a puzzle in real life. Computer technology however does have other advantages as it has been shown to help support learning and is especially useful in developing skills of critical thinking, analysis, and scientific inquiry [50]. Technology is therefore able to teach and support learning differently than physical toys.

It is the use of mechatronics that gives the system in this thesis the benefits from both the physical toys and the digital solutions and lets it be a versatile multifunctional tool. One mechatronic product available today that targets fine motor skills is the Robotic rehabilitation glove from Syrebo [Figure 2.4].

Figure 2.4.: Syrebo glove [5].

It is a glove that can "*...help patients master fingers flexion and extension, reduce hand muscle tension, relieve edema and stiffness, promote rehabilitation of brain nerve injury through exercise, improve hand activity and accelerate the rehabilitation of hand function.*" As Syrebo website explains [5]. With the use of mechanical parts, the glove moves your fingers for you to achieve these results. It can also mimic movements from a different glove on another hand so that you can get your disabled hand to move in different ways. Because the use is so general, and the fact that they have made different sizes to fit smaller hands as well, makes it adaptable to almost everyone. Also, by being so general, easy to use, and small it is a great device to have at home and with you wherever you want. On the other hand, the functionality of the device is rather limited as to how the user can contribute, and learn on their own. This limits its potential to be a fun way to exercise.

Another mechatronic system that focuses more on the fun part is the Amadeo from Tyromotion, [6], seen in Figure 2.5. The goal here is more specific training of fine motor skills by being able to have movement in each finger. It is a heavily sensor-based device where it can measure how you use it and give this feedback to the user for further improvement. You move your fingers to play a game on the screen that is made to consider age, as motivation can be very individual and age-dependent.

Figure 2.5.: Amadeo from Tyromotion [6].

The glove from Syrebo has an elegant design solution when it comes to its portability, as it is rather small, but lacks in its overall functionality. The Amebo is the other way around, where it is a rather large and complex system, but a user can benefit greatly from its functionality in many ways. A system that is a combination of these two can therefore be seen as a more user-friendly and fun way of exercising, which is what the system in this thesis aims to be.

2.3. Requirements

An overall goal is to make the system as general as possible. By having adaptable skill levels and exercises, you are able to reach a broader user group, and more people could benefit from it. Generalization could also remove the need of making a specialized system for each individual, which can help with self-esteem as the users are treated just like everyone else, and not as a special case that needs to be solved. This is something that could also help with motivation for the user, and contribute to the system being something fun they would want to do, rather than something they have to do. It also makes economic sense to make it very general as it would target a larger audience, and therefore possibly also make more money.

Making the system general should however not compromise its intended use. Having a specific user group in mind, and rather add more general traits as you go along could therefore be beneficial. Based on the arguments made in section 2.1 it can be concluded that children may benefit the most from the intended system in the long run as training of fine motor skills can have other effects. Because of this, children are also a larger group as they could benefit from this type of training independently of their level of fine motor skills. With that in mind along with the system's intended use, some non-functional requirements can be presented and discussed.

- **Universal design**
  - Because fine motor skill is a characteristic known to be shared by the users of this system, it is designed with mainly this skill in mind. However, it is important to also have universal design principles in mind for the prototype to accommodate different needs and other challenges people may face. Additionally, applying universal design rules strengthens the case for the generalization of the system, as it ensures accessibility for a broad range of users. The seven principles of universal design [87] are therefore taken into consideration and discussed more later in this thesis.
- **Not feel like a test**
  - As the system aims to be a fun way of exercising and a tool for development, it should not feel like a test for the user. A testing environment can be stressful and take the joy out of doing something, as they often take place outside of familiar places, with an observing second party present. It could also contribute to the feeling of something you have to do, rather than something you want to do. By making the system intuitive and easy to use, in addition to being portable so you can use it in your own home, you make the users feel safer in its use and avoid having it feel like a testing environment.
- **Size and usage**
  - With children being mostly in mind for the use of the system, it is important to think about making it robust. It has to endure that it could be treated without the love it may deserve, as children may not be as careful and considerate as perhaps an adult would be.
    Additionally, the size of the system should not be too large as a child should be able to use it on its own. With it being a medium to small size it can easily be placed in a home, kindergarten etc. without taking up too much space.
- **Inexpensive and basic**
  - Even though robustness and durability are already taken into consideration, something can always go wrong and parts may break. It is therefore favorable to make the system in a general manner so that it could be easily reproduced. This can be achieved by using tools and parts that are common, available, and simple. In addition, by making a basic foundation it is easier to modify and further develop it at later stages, making it more versatile.

In addition to these points, the most important thing the system needs is a proper function. This should be a physical activity for the user to exercise their fine motor skills beneficially. As the user group, independent of age, are people who struggle with fine motor skills to different degrees, the activity should not be too complicated or too intricate. This also coincides with point three from the seven universal design principles [87] which states that a design should be easy to understand and intuitive to use. The toys in Figure 2.1, 2.2 and 2.3 are examples of different activities to exercise fine motor skills, the box having the most potential. With it, you can adjust and adapt to different skill levels by simply changing the size of holes and using different shapes. By also training cognitive skills the system is more versatile and appealing to children. In addition, an advantage of basing the system on something familiar is that it can help make the system seem less frightening and complex in the minds of new potential buyers and users. By having the overall concept of the system in place, more specific requirements can be looked at for the intended mechatronic playtoy.

- **A removable board with holes**
  - Having a board you can remove from the system makes it more versatile as you then can design multiple different boards that each can have different features and be adapted to different skill levels. This also opens up the possibility of making more advanced and different boards in the future.
- **Extraction of blocks**
  - With putting blocks through holes, it is natural to want a way to retrieve them again. An elegant way to extract the blocks should therefore be incorporated as this could make the system more intuitive and easy to use, as specified by design principle three of universal design [87].
- **Electrical components**
  - Electronic components are a crucial part of making it a mechatronic system. With sensors registering how the user interacts with the board you can create innovative and fun ways to use it, for instance in the form of games or challenges. It can also be a tool, making it more intuitive as it can guide the user and make sure they exercise in the right way. Speakers or a way to connect headphones can also be incorporated for giving feedback to users not dependent on sight, conforming to principle four of the 7 design principles of universal design [87].
- **Display and GUI**
  - With sensors registering how the system is being used, a display along with a GUI can be incorporated for showing the progress, as well as

relevant feedback and reward mechanisms. This could also help guide the user, as well as motivate them. It also opens up many possibilities for different designs. A GUI can be designed to better adapt the system to its user group and is an essential part of making the system fun and motivating.

Based on these requirements, Figure 2.6 illustrates what the system could look like.

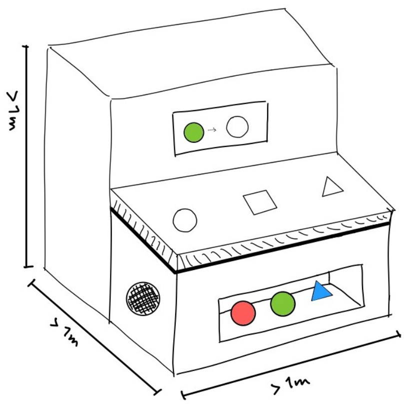


Figure 2.6.: A first draft of how the system could look.

2.4. Design suggestions

From the preliminary study [49] it is concluded that having different design solutions is favorable as physical installment can lead to unexpected problems. In addition, by having different designs you could be able to adapt to unexpected events and overcome them more easily. However, generalization in the designs is taken into consideration to eliminate most of these types of problems. With this in mind, a summary of the discussed parts from the prestudy follows.

When it comes to placement of the display there are three main points to take into consideration.

1. Visible to the user while the system is used.
2. Large enough so that it can be seen.
3. Be a robust part of the system to avoid breaking it.
4. Not compromise the boards and their use.

Figure 2.7 show a solution that satisfy point 3, and to some degree point 2 as you have the whole front to use. It does, however, not satisfy point 1 as the focus of the user would be on the top and therefore maybe miss important information from the screen. With sound, you could be able to catch the user's attention back to the screen. This would, however, not support principle four of the seven design principles for universal design [87] as the user would be solely reliant on sound to catch the given information. Figure 2.8 improves a little on point 1 as the screen now is easier to see, but is still not an ideal placement when it comes to being seen by the user. The design also makes the system less fragile, compromising point 3, with the screen sticking out. With these two solutions, the display would have to be a less significant aspect of the system as the user would not view the display and its contents at all times. They would however both satisfy point 4 as the display does not interfere with the boards in any way.

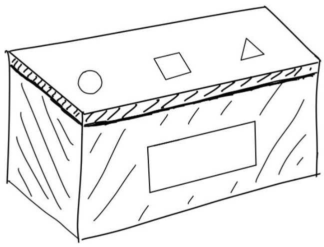


Figure 2.7.: How the system could look with the display in front.

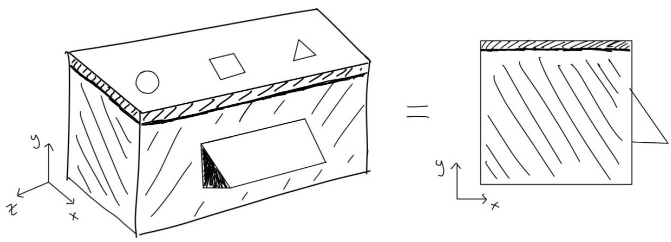
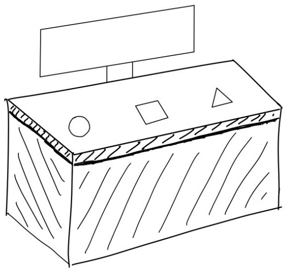
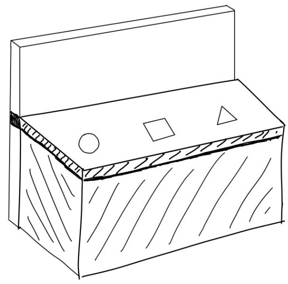


Figure 2.8.: How the system could look with the display sticking out in the front.

Regarding point 1, Figure 2.9 and 2.10 are better solutions as the user's attention already are in the eyesight of the screen at all times. With Figure 2.9 however, point 3 is not satisfied as the screen is now a loose part, which should be handled with care. Figure 2.9b would be a little more robust as it is a more integrated part, but are still more exposed and fragile than for example Figure 2.7. Point 2 however, is more than satisfied as the screen can be as large or small as you would want it to be.



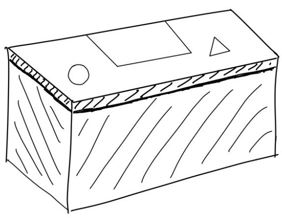
(a) Display attached on top of box.



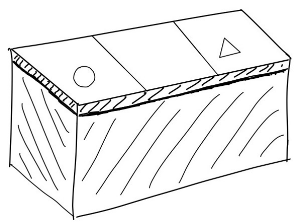
(b) A more robust attachment of display.

Figure 2.9.: How the system could look with the display above the box.

Figure 2.10 on the other hand, would eliminate the faults Figure 2.9 have with point 3 as it now is an integrated part of the system. Point 4 would, however, be an issue with both of the designs, as the display now takes up a third of the board, making it smaller and limiting its potential with the sizes and quantity of holes. Point 2 would also be compromised as the screens here would have to be as small as possible to allow the boards to have their functionality.



(a) A smaller display not dividing the board.



(b) Display dividing the board into two pieces.

Figure 2.10.: How the system could look with the display on top of the box.

Figure 2.11 shows a solution that to an extent satisfies all of the points above. The screen is placed in a better position to get the user's attention, and can be adaptable in size with a larger surface area to use. In addition, it would be a more robust part of the system and not in the way of the board. The biggest drawback of this design is that the size of the system would be larger. This can however open up for the possibility of storing additional future boards in the back.

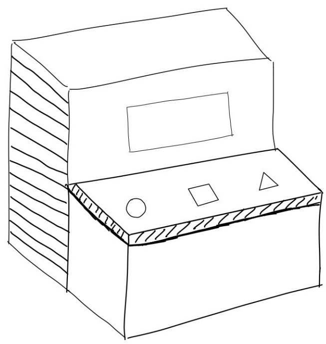


Figure 2.11.: The most optimal placement of display with an integrated solution.

With the display being one of the most essential parts of the system, its placement must be prioritized. The other features of the system are therefore looked at with Figure 2.11 in mind. Without compromising point 4 as this still applies for other parts, and having Figure 2.11 as a basis, the logical placement of for example speakers and extraction of blocks is having them on the sides, or the front of the box. The backside is ruled out as an option with generalization in mind so that the system can be placed against a wall without compromising its use. One solution is therefore to have the extraction of blocks as a hole in the front so that the user could easily grab them. With the extraction being in front, to give it the needed space, the speakers would then have to be on the sides.

The blocks are an essential part of exercising fine motor skills with the system. They would therefore have to fulfill some requirements for the training to be efficient.

- **Customizable**
  - Since the system has a very specific intention of how and why it should be used, the blocks would benefit the system if they were able to be made specifically for the holes.
- **Light**
  - For the blocks to be easily picked up they need to be fairly light so as not to make the exercise unnecessarily hard.
- **Universal design**
  - One possible solution to the game is that the blocks can have different colors to indicate where the blocks should go. If the user is short of sight, however, this would not work very well. In order to conform to

universal design, texture could be added to the blocks, where a specific texture would correspond to a color for instance.

- **Not too small, not too large**
  - Size is an important thing to consider when training fine motor skills. The smaller they are more precise fine motor skill is needed to pick them up and place them in the hole. If they are too big on the other hand, it would be a struggle for those with small hands to pick them up at all, limiting their ability to even use the system as intended. Another important point to make here is that they should not be so small that a child would swallow them.

The most important point to take away from this list is the ability to customize the blocks, as this would fulfill the other requirements as well.

An important part of many games is the use of levels. The implementation of levels and increasing difficulty could contribute to motivation and give the user a natural training progression [52]. In addition to being a device that exercises fine motor skills, it can also work to train cognitive skills simultaneously as these two skills are very closely tied to one another [42]. By increasing the difficulty in a physical and cognitive aspect level by level you can obtain a natural training progress and learning in both skills.

When it comes to actual implementation of levels there are three main aspects that need to be considered.

- **The blocks.** By changing size, adding texture, and/or changing their shape.
- **The boards**. By reflecting the shape and sizes of the blocks in different ways. For example by placing them far away from each other, having tighter holes, etc.
- **The display**. By guiding the user, and showing them how different boards can be played.

With the training of fine motor skills, grabbing the blocks is what mostly trains the muscular abilities in the hands. It is when mixing the blocks with the boards you are able to include training in both skills. For example, if the user is presented with several holes on the board and one block, it requires logical thinking to figure out which hole it belongs to. In addition to feedback, the user could be able to learn from their mistakes further practicing cognitive skills.

An intuitive thought when implementing levels and having the possibility to use more than one board is that each board can be one level. With a requirement being that the system should be as small as possible it is, however, not favorable to make too many boards. Therefore it is important when designing the different boards to make them as general as possible so that they can be used in more than one way. An example of doing this is to implement several levels in just one board. One board could then for example be level 1 to 3, the next 4 to 6, and so on.

Figure 2.12 and 2.13 show two different ways of doing this with the display being what separates the levels. In figure Figure 2.12 an easy task is illustrated where the user has to put the green ball in the green hole in order to finish the task. In Figure 2.13 a harder task can be seen, where the user has to first place the red ball in the red hole, then the blue triangle in the blue triangular hole. If the user then manages to do this in the correct order the task is completed.
This design utilizes the advantages of the different parts, as discussed, and would all in all give the most benefits to the system as a whole.

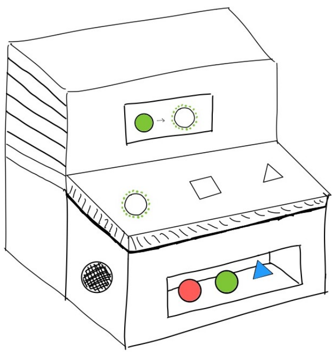


Figure 2.12.: Example of an easy level with the system.

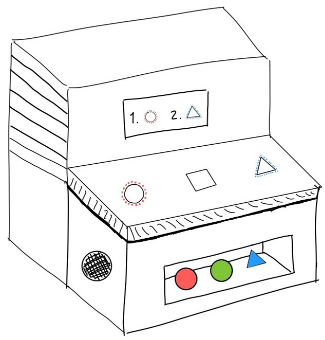


Figure 2.13.: Example of a harder level with the system.

Chapter 3.

Theory

3.1. Gamification[1]

To begin grasping the concept of game logic and implementation of it, there should first be an explanation of gamification.

Gamification is the concept of adding game elements, mechanics, and game-based thinking to non-game environments like a website, classroom, or business to create similar experiences to when playing games [54]. This could be the inclusion of levels, feedback, progress or rewards systems, etc. Gamification first entered the mainstream vocabulary around 2010, making it a relatively new concept. One of the reasons for its growing popularity lies in the belief in its potential to foster motivation [20].

For this reason, it has been very popular in the last couple of years to implement gamification in educational environments. Although research in this area is rather new, meaning nothing can be said of certainty when it comes to the effects of this long-term, promising research has been conducted. It has for instance been shown that gamification has the potential to increase student motivation and that students following a gamified version of learning get better scores on practical assignments [11]. Several other studies also conclude that gamification can have a positive effect on learning and motivation [69] [22].

However, it can also be debated whether or not there is a need for gamification in different settings. If the goal is to use it in school and learning, many studies also show that while it can contribute to something positive, the effects of it may be rather small [60]. Another important point to shed light on is the use of game elements in a system that can be used by elderly people. It is a common myth to

[1] Parts of this section are based on the author's own prestudy [49].

assume that elderly people do not play games or like to play games and therefore will not like gamification. But according to Entertainment Software Association's 2013 report "*Gamers Over 50 Study: You're Never Too Old to Play*" 48% of adults age 50 and older say they play video games [54]. Hence, implementing gamification by itself is not age-limiting for its users.
In addition, it can be debated whether it is problematic or not to mix a learning environment with something mainly thought of as a leisure activity, as it can affect how effective gamification is in its use [48].

However, the system intended in this thesis is not meant as a pure learning tool, but more as an exercise tool that can contribute to learning. Because of this, the arguments that debate the necessity of gamification are not entirely applicable to this setting. Nevertheless, the main thing to take from all of this is that gamification is proven to have, if not small, a positive effect on motivation.

Moreover, it is important to keep in mind that when it comes to principles of what motivates people there is no one correct answer as everyone is different and therefore can be motivated by different things. On the other hand, feedback and rewards systems, in general, have been shown in several settings to be a good tool for motivation [40]. In games, they are often used in how you progress through levels, and it is this way they are used in this system as well.

By implementing gamification in a new system, one could therefore argue that it can have a positive effect on its users. The goal is to implement just enough gaming logic to gain the positive effects of gamification by simulating similar experiences to playing a game, while still keeping it simple and user-friendly.

3.1.1. Feedback, Rewards, and Progress

The concept of a reward system is commonly known when it comes to motivation. In the real world, salary can for instance be looked at as a form of reward system where you get paid for a job well done. In games, these rewards are often items or tools to help you when you get to harder levels. They can also be benefits for finding something or doing something good in a game to keep you going. Take the classic Super Mario Bros game [Figure 3.1] for example. By completing a level, the reward you get is to progress to the next one, which in return gets you closer to the goal of the game, saving princess peach. In addition, you can gain rewards throughout the levels in the form of power-ups or extra lives which helps you complete the levels. All of this gives motivation for you to continue the quest of saving the princess because the game makes the quest attainable by giving you help and resources to complete it.

In addition, games can use progress as a reward in itself as they often have a

Figure 3.1.: By getting a special mushroom Mario can grow and destroy all in his path, making it easier to progress through the level [7].

story they follow or a quest that needs completion. An exercise system does not have the same opportunity for story-making in that sense. On the other hand, one could be able to make progress in itself be the reward, by having some way of indicating how far you have come in your training. The Wii Fit game [Figure 1.1] for instance, uses this. After completing a level you are able to see how much further you have come in your exercise, how many calories you have burned, etc. Because of this, you can see the progress you are making after each session with the system and allowing you to continuously see the goal, making it more reachable.

Feedback is a concept closely connected to that of reward systems, where rewards can be seen as a concrete result of getting feedback. When you do something right the game rewards you, as explained with the Mario game. However, there is also feedback for when doing something wrong, often in the form of punishments. In Super Mario Bros this can be walking into an enemy for instance. If you are tiny this punishes you with death and you have to start the level all over again.

Feedback can also be explained as the underlying concept used to make players learn how the game works, whereas rewards contribute to motivation. A big reward may give a bigger motivation in thinking the next level could be easier, while a smaller reward may not be that exciting and demotivate the player to continue. By receiving punishments or rewards, the user is told what is good and bad in the game and then learns how to play the game and progress through the levels.

3.2. Additive manufacturing

Additive manufacturing can be described as the process of creating an object by building it layer by layer [97]. It typically refers to 3D printing, a technique that started as a phenomenon in the 80s [88], where the material is layered in successive layers to create 3D models from digital files [97]. Today, a series of different materials and methods are used [37] in order to fit both a consumer market and manufacturing processes [10].

The most common method of 3D printing on a consumer level is called fused deposition modeling (FDM). It is a method that melts and extrudes thermoplastic filaments, the most common materials being PLA or ABS. This is done through a printer nozzle that builds the model layer by layer on a build plate [37]. FDM is well-suited for basic models and quick prototyping parts, but not for printing complex designs or parts with intricate features. This is because the method has very low accuracy and resolution in comparison to other plastic 3D printing methods [37]. However, FDM printing comes with many advantages and is more than adequate for this thesis in regards to the blocks, as per their design and requirements [section 2.4]. An example of an FDM printer that uses PLA is shown in Figure 3.2, which specifically shows the original Prusa i3 MK3S+ 3D printer [68].

Figure 3.2.: Prusa 3D printers in prototyping lab at NTNU [68].

One of the main advantages of FDM printing is that it allows for flexible designs. As long as you have 3D modeling software you can create models according to your very own needs and designs. For this project *Fusion 360*, which is a professional 3D CAD software by Autodesk [35], is used to create the different blocks. However, before a model can be printed from a 3D modeling software it has to be

transformed from a digital model into printing instructions, called G-code, using a software called *Slicer* [101]. The printers in Figure 3.2, along with their slicer software *PrusaSlicer* [68] were mainly used for this project, in addition to Ultimaker printers and their slicer *Cura* [95].
In addition to the flexibility of 3D printing, because of the increasing popularity of the printers, and now rather inexpensive prices [9], several tutorials on different designs, uses, and printing exist online. This makes it easy to learn, and a fun way of making your own parts and designs for different purposes.

However, certain downsides to 3D printing should be mentioned. Most affordable printers have a rather small build plate, and volume, limiting the size of models. In addition, with the FDM method where models are created by layering material, each new layer has to be supported in some way to keep the intended shape and form. If your model then has an overhang that is not supported by anything below, there is a good chance the model could be ruined [25]. To avoid the issue, support structures are usually used, where an overhang threshold is chosen in the Slicer for when the supports should be generated. These supports then have to be removed after printing, adding more post-processing work and risking damaging the model's surface. Limiting the use of supports is therefore favorable to getting better-looking models.

3.3. Embedded systems

An embedded system can be defined as a combination of computer hardware and software, designed for a specific function [18]. They usually consist of a processor, power supply, memory, and communication ports that transmit data between the processor and peripheral devices, using a communication protocol [18].
For this thesis, the electronic components that make up the mechatronic system work together as a fully functional embedded system. Its task is to support the display and its GUI. For doing this, an Arduino board, a Raspberry Pi, UART, sensors, and a display are used to create the embedded system according to the systems requirements [Figure 3.3]. Each of these, and how they are used, are explained further in the following sections and chapters.

(a) Sensors, Arduino, Raspberry Pi and display.

(b) Raspberry Pi attached on the back of the display.

Figure 3.3.: The different electronic components making up the embedded system.

3.3.1. Arduino vs. Raspberry Pi

To choose the correct hardware for an embedded system, it is important to have an understanding of how it is intended to function [30]. For this specific project, a display is needed to showcase the game, and sensors are needed to register when blocks are put into the board in order to give feedback to the game and the user. This might seem like an easy task, but the decision of choosing hardware should not be taken lightly.

As mentioned in section 3.3, an embedded system usually consists of a processor, which in this case refers to either a microprocessor or microcontroller. While they do resemble one another in that they both perform relatively similar functions, and both incorporate a central processing unit (CPU), they are very different in how they are used and what they can be used for [62]. A microcontroller is a single integrated circuit where a CPU and other necessary components are all incorporated onto the same chip [62]. As a result, microcontrollers are designed to be used for specific tasks, or any assigned tasks in loops [67], making them applicable to control electronic devices [70]. Microprocessors on the other hand, also consist of a CPU but use separate integrated circuits for memory and peripherals, instead of including them on the same chip [18]. As a result, they typically require more external connections than microcontrollers and are used for more complex tasks that typically require higher memory and more complex coding [67].
The Arduino and Raspberry Pi boards are two very popular choices for using either a microprocessor or a microcontroller in embedded projects.

Arduino

Arduino is an *"open-source electronics platform based on easy-to-use hardware and software"*, as described on their website [98]. Their boards are microcontroller based, made with the intention to simplify the process of working with them. In addition, by making them easy to learn and use, the goal is to give anyone that wishes to enhance their lives with electronics [13] an opportunity. The most used, and best-documented board in the Arduino family is the Arduino Uno [17], which is also the board chosen for this project. The board and some of its technical specifications are shown in Figure 3.4 and Table 3.1.

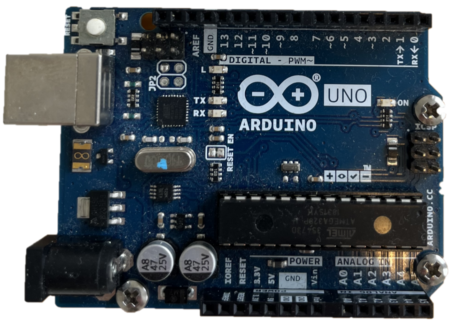


Figure 3.4.: The Arduino Uno used for this prototype.

Features	**Specifications**
Processor	ATmega328P
Processing power (clock speed)	16MHz
Flash memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Additional features	14 Digital I/O pins (6 provide PWM output)
	5V operating voltage

Table 3.1.: Technical specifications for The Arduino Uno [17].

To program the board, the Arduino integrated development environment (IDE)

is used, which can either be downloaded or used online with an account on the Arduino website.
In addition to the board's specifications and simple interface, a lot of different modules are made specifically for the Arduino, making them a great fit in most electronics projects. Concerning this thesis, the Arduino is an excellent choice for controlling sensors underneath the boards with holes to keep track of the user's moves. However, making the rest of the system with a display and corresponding GUI could be more challenging by only using the Arduino board.

Raspberry Pi

As explained by their website, Raspberry Pi is a foundation *"... with the mission to enable young people to realize their full potential through the power of computing and digital technologies."* [14]. As a way of achieving this, the foundation has developed microprocessor-based boards, simply called the Raspberry Pi. These boards work similarly to computers where they even have an official supported operating system called Raspberry Pi OS, a version of Linux. This enables the use of different programming languages for development, which as a result makes the Raspberry Pi very fit for different use. Figure 3.5 and Table 3.2 show the Raspberry Pi 4B model which is used in this project, along with some of its technical specifications.

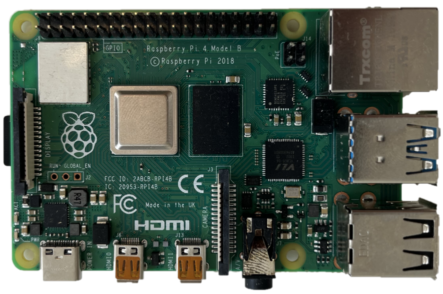


Figure 3.5.: Raspberry Pi model 4B used for this prototype.

Features	**Specifications**
Processor	BCM2711 (ARM v8)
Processing power (Clock speed)	1.5GHz
Memory	1GB, 2GB or 4GB LPDDR4 (depending on model)
Additional features	4-pole stereo audio and composite video port
	2 × micro HDMI ports (up to 4Kp60 supported)
	Micro SD card slot for loading operating system and data storage
	5V operating voltage

Table 3.2.: Technical specifications for the Raspberry Pi model 4B [77].

Since the Raspberry Pi board has more processing power, RAM, and a larger CPU than the Arduino, they are better suited for larger programs and more complex coding. This makes it an ideal platform to develop the main attraction of the system in this thesis; the display.
With Raspberry Pi also being a very popular learning platform, used for both home and industry purposes, the Raspberry Pi Foundation continuously works with updating and better the technology. Because of this, and because of the potential Raspberry Pi has for different projects, many tutorials, and guides also exist online, similar to those for Arduino, making it a user-friendly and easy tool to learn.

Best of both worlds

Even though the Arduino and the Raspberry Pi have many good qualities, they are best suited for different types of projects. As the project in this thesis requires both repetitive work with sensors and more complex programming with the display, it was decided to use both an Arduino and a Raspberry Pi to bring the system to life. This way, the Arduino handles the sensors and reads their data, while the Raspberry Pi focuses on the GUI and the logic of the system.

With sensors connected to the Arduino, the idea is to send this data to the Raspberry Pi using a communication protocol, where the GUI then reacts according to the sent data.

3.3.2. Communication is key

In Embedded systems, communication protocols are used to define how data is transmitted between devices [53]. There exist many of these protocols, but for this project, the UART protocol is used and described more in detail. UART, or Universal Asynchronous Receiver-Transmitter, which uses asynchronous serial communication to transmit data, is a simple, yet reliable way for low-power, long-distance communication between two devices [41]. It can also be described as a multi-master protocol, where all connected devices are free to send data when they want, unlike master-slave protocols (I2C, SPI, etc.) where only the master device can initiate communication [31]. The communication is done without the need for a clock signal (hence Asynchronous), and by sending bits at a predefined baud rate using two wires, RX and TX that receive and transmit the data. Both the Arduino and Raspberry Pi boards have support for UART communication, where you could either use the RX/TX pins on each device, or a USB cable between them.
By default, the Arduino uses Serial communication [78] to communicate between the board and the Arduino IDE. By connecting the Arduino to the Raspberry Pi, the Arduino IDE side of the serial communication is replaced with the Raspberry Pi board, as UART is also based on serial communication. From the Raspberry Pi side, Python and PySerial [73] is used for serial communication with the Arduino board.

To transfer data between the two boards, a USB cable was used, as this is the tidier solution. To establish the connection between the two boards, a guide from The Robotics Back-end was followed [31], where parts of the code seen in Listing C.1 and C.2 are taken from this guide and adapted to the specific project. Briefly explained, the Arduino board is first detected using the terminal on the Raspberry Pi, and the baud rate is chosen to be 9600, a common baud rate to use and adequate for this project. The connection is then established, and the data is sent from the Arduino in a list to the Raspberry Pi where the elements are stored in variables in the Python program on the Raspberry Pi. The communication was established similarly in the final code, but modified to fit the logic and the rest of the code.

3.3.3. Display

For the system to work as intended in accordance with the requirements, a display with a GUI is needed. Either a display connected to the hardware, or a separate display, for instance, a tablet or smartphone could be used. To not make the process of creating the system longer than needed, and since a decent display compatible with Raspberry Pi was found rather cheap, this solution was chosen for the project. In addition, by using a display directly connected to the Raspberry Pi, the only step needed to create the display with its required functionality would be to program the GUI and run the program on the Raspberry Pi.

The display used in this project is a 7" Raspberry Pi touch display [79] seen in Figure 3.6.

Figure 3.6.: 7" Raspberry Pi touch Display with DSI cable and wires.

The Raspberry Pi is attached to the back of the display with bolts and connected to the display via a DSI (Display serial interface) cable and powered using Vcc (Voltage common collector) and ground pins. As seen from Figure 3.6, the display also has mounting holes that can be used to attach the display to the physical system. A hole to make room for the Raspberry Pi would then be required in the physical box for the display to be attached properly. The size of the display compared to the system can be seen in Figure 4.18.

3.3.4. Sensors

For the thesis work, some kind of sensor is needed to register the user's moves and update the GUI accordingly. Some relevant, common sensors used in electronic projects, and how they are used can be seen in Table 3.3.

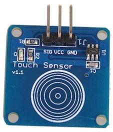
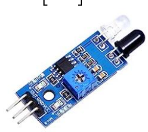
Type of sensor	**Usage**
Touch [33] 	Used to detect and record physical touch. Works as a switch where pressure is applied to an area, opening the electrical circuit and allowing the current to flow through it [92].
Light [76] 	Detects and measures the intensity of light, usually electromagnetic radiation in a wavelength range from ultraviolet to far infrared [19].
Ultrasonic [76] 	Measures distance of an object by emitting ultrasonic sound waves and converting the reflected sound into electrical signals [26].
Color [85] 	Uses external means of emitting light and then analyses the reflected light of an object to determine its color [16].
IR [32] 	Emits infrared radiation which is reflected by an object, and registered by a receiver [44]. Usually used to determine the proximity of an object.

Table 3.3.: Overview of the most common sensors applicable for the thesis work.

The sensors presented in Table 3.3 are some of the many sensors that exist on the market today which could be used in different ways for this project. A touch sensor could for example be implemented where the user has to touch somewhere to be able to drop the block into a hole. Or it could be used underneath the hole where the block then lands, indicating that it has been inserted. A light sensor could be used as a block inserted into a hole would deprive the environment around the sensor of light, implying something is blocking the sensor. By having each block distinct colors, a color sensor could detect different colors and let the system know which block had been inserted. Ultrasonic and IR sensors could both be used to detect the proximity of the blocks registering when a block is near the sensor.
Because of available technology and simplicity in implementation, IR sensors were used for this project. One sensor for each hole was placed underneath the board

and used for detecting the inserted blocks.

3.4. Programming

As Python is a high-level, interactive, object-oriented programming language [36] that is used in everything from desktop GUIs to business applications [15], it was chosen as the language for this project. For the IDE, Thonny [89] was chosen for simplicity, as it was pre-installed on the Raspberry Pi. The Arduino on the other hand, uses a variant of C++ in the Arduino IDE.

To detect time-critical events happening outside the main program, polling and interrupts are common techniques in embedded systems programming. Polling can be described as a protocol where the processor continuously checks if an event has happened. Interrupts on the other hand is a hardware mechanism where a interrupt signal is sent if the device needs attention [28]. While polling is more explicit and straightforward, interrupts are better for longer and more complex code.

For this thesis, the main program is the GUI while the Arduino sending sensor values is an outside event, and hence polling is used in order for the GUI to properly receive sensor values at correct times. As the code is not overly complex, and with only the sensor values needing surveillance, polling is more than adequate.

3.4.1. GUI

A graphical user interface is a digital interface that consists of graphical components such as buttons, icons, and menus for people to interact with [46]. GUIs have been around since 1981 when Xerox released the first GUI in a consumer product [46]. Today, GUIs are all around, from smartphones to coffee machines, where they work to better the user experience of the product. For this thesis, a GUI is made as an important part to guide and inform the user of the system.

Python is the chosen programming language for the work as it is very compatible with creating GUIs. Many different Python GUI frameworks exist, TKinter for instance is a popular choice as it is the only built-in framework to the Python standard library [24]. Another popular framework is PyQt or PySide [58], which is used for this project. Both PyQt and PySide [58] work similarly, where code written for one can often be used as it is with the other by changing the imports from one to the other [58]. For this project, PySide is used, along with Qt Designer [59] to better create the GUI. Qt Designer is a graphical editor compatible with PySide where You can lay out graphical components such as buttons, labels, and widgets, and then edit their functionality in the code, simplifying the process of creating a GUI.

As PySide and Qt Designer were new and unknown tools prior to this thesis, guides from Python GUIs [74] were used to learn and set up the interface. Listing 3.1 shows how the Main file for the interface looks like and is based on the code from the Python GUIs guides.

```
1 import sys
2 import PySide2.QtWidgets as QtWidgets
3 from PySide2.QtWidgets import QApplication, QWidget
4 from motorskillgame import game
5
6 app = QtWidgets.QApplication(sys.argv)
7 window = game()
8 window.showMaximized()
9 window.show()
10 app.exec_()
```

Listing 3.1: Main file for the GUI, where the application is defined and executed.

In Listing 3.1, an instance of the PySide2.QtWidgets.QApplication class is first created in line 6. This application controls the main control flow and settings. Next, the window of the interface is defined and displayed. The window made in line 7 is an instance of another self-constructed class where the game itself is defined. In the last line, the main loop is entered and the application is executed. To properly close the application an exit command should also be included, this is implemented in the game class as a close event [see Listing C.5].

If another loop is introduced in the code while the GUI is executed, since this also is a loop, the GUI will freeze until the additional loop completes. Having sensor values read in a loop from the Arduino could therefore invite problems to the application. The solution was to use threads. In computing, threads can be defined as a separate flow of execution [45], where it is defined with simple instructions that can be executed independently of other code. This makes it possible to have the loop from the Arduino run at the same time as the GUI, without causing the application to freeze. How threads are used specifically in this program can be seen in Listing C.4, where the thread is defined [function begin in line 122 in Listing C.4] and started [line 20 in Listing C.4], beginning the logic of the game. The thread is then closed at the end of the main logic function in each game by implementing a return at the end of the function [can be seen in line 119 in Listing C.4]. The threading class in python [45] is used for the threads. Alternatively, the PySide.QtCore.QThread class in PySide could be used.

Chapter 4.

Prototyping process

This chapter examines the approach and procedure for the thesis, with discussions of how and why the following methods have been used to progress and create the finished prototype. The work is split into three distinct parts dealing with different aspects of the system. They are the:

1. Blocks [section 4.1]
2. Box with board(s) [section 4.2]
3. Display [section 4.3]

4.1. Blocks

As mentioned in section 3.2, 3D printing was used to make the blocks to create and customize them according to their requirements [section 2.4]. Because 3D printing was something the author had never done before, the process of making these blocks started early in order to have an iterative design process to learn and adjust the blocks underway.

When it comes to the shapes of the blocks, their only requirement is their ability to be grabbed. For simplicity, three manageable shapes are made to begin with as the author's experience with 3D printing is limited. A Sphere, cuboid, and triangular prism were chosen as they have very simple designs, making the process of printing smoother. Their corresponding shapes on the board are then a circle, square, and triangle, which are also fairly simple shapes to implement.

Since FDM 3D printing materials can come in a variety of colors [37], they can be chosen freely by the creator[1] . In accordance with universal design, it can be favorable to then adapt the colors to those that are color-blind. The most common

[1] Amongst the available material. Or else you would have to order special colored material.

color blindness is having difficulty separating red and green colors, whereas a less common is trouble with blue and yellow [93]. Since IR sensors are used in the system [subsection 3.3.4], black also has to be avoided as black objects may not be registered by the sensors. Avoiding red and green in combination, and black altogether is therefore prioritized for the blocks. The final color combination for the blocks was therefore red, blue, and white, as these were the best-suited, available PLA colors.
In addition to color, texture could also be added to the blocks easily with 3D printing in order to further conform to universal design, but as this is more advanced, only color is prioritized for now.

4.1.1. Cuboid

Figure 4.1 show the 3D model of the cuboid made in Fusion 360. The 3D printed version, with only adjustments in size, can be seen in Figure 4.2. The printed cuboid is 7x7x3.5cm. These measurements were chosen a bit at random because it was a first test, mainly for the look and feel, but also to get a better understanding of 3D printing. The 3D model was made hollow in Fusion360 without a bottom, and a lid printed separately. In Figure 4.1 you can also see small indents at the bottom of the print, this was to have a better surface to adhere the lid on if that were to be necessary.

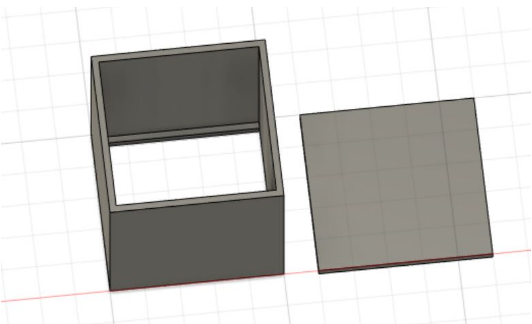


Figure 4.1.: 3D model of cuboid in Fusion 360.

Figure 4.2.: First 3D print of cuboid.

Having no infill was mainly done to make the printing time shorter, but by having hollow blocks you are also able to adjust the weight of the blocks by putting different things in them as desired. In addition, you can be able to put different things in them so they make distinct sounds when you shake them. With the system or display then replicating that sound, it can be used to indicate what

block is supposed to go in next. This could make it more suitable for those that do not rely on sight to recognize the different blocks and use the system.
However, to achieve this, you would have to print more than one part for each block, thus having to attach the parts at some point. An easy and quick way to do that would be glue, but since ingesting glue can be dangerous, no glue should be visible or on the outside of the blocks.
However, the focus for the blocks in this thesis is rather on the easiest and fastest way to produce them as their sole function for this work is to go into holes in a board. These points are, however, valuable for further work with the system in making it more versatile.

As seen in Figure 4.2, the cuboid is almost the same size as the author's hand. But as stated in the objectives for the thesis [section 1.2], the user group to have in mind for designing the system is children, which usually have smaller hands.
As stated in section 2.1, motor skills start developing in infancy, and further develop during childhood where exercising the skills can have other benefits. As a consequence, young children do not benefit the same as someone older when it comes to the exercise of fine motor skills, and/or cognitive learning. Children between 1-4 years can therefore be seen as a good age to adapt the system to, as they are still in development, but old enough to grasp the concept and be able to use it [96]. Since they usually have smaller hands than that of adults, the blocks need to reflect this. They can not be larger than what the child can grab, but at the same time not small enough to swallow, as this would make them a choking hazard.

The approximate size of the hands for children aged 1-4 is 10cm in length and 13cm in circumference [83]. This would make the width of their hand ∼5.5cm by taking half of 13 and subtracting ∼1cm for the breadth of the hand. Based on this, an estimate of how large the palm of their hands are can be seen in Figure 4.3, where the size of the palm is set to be half of the total length of the hand minus 1 cm.[2] This reference square is further used for the size of the blocks, as it indicates a size that can be considered maximum for the blocks while a child is still able to grab them.

[2] These are estimations based on a general sized hand to get a rough estimate of the size of the blocks, and can not be described as entirely accurate

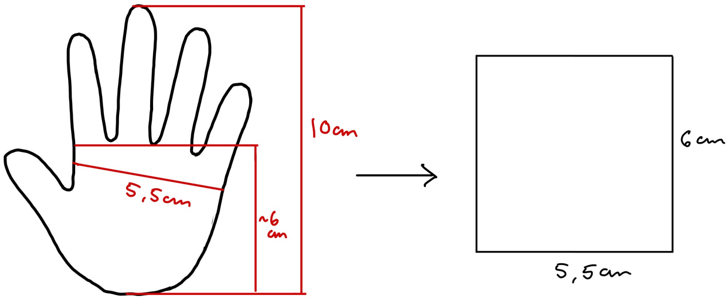


Figure 4.3.: Approximate hand size of children aged 1-4.

For deciding the minimum size of the blocks, a small part test fixture [82] is used as a reference. This is a cylinder, with an inner diameter of 3.17cm. An approximation of the size of a fully expanded throat of a child under three years old. The absolute minimum size of the diagonals to the sides of the cuboid is therefore 3.2cm, as this is just enough for it not to fit in the part test fixture cylinder. Based on this minimum and maximum size, the first print of the cuboid could be considered a little too big and should be made smaller in the next iteration of printing.

For the second print, a bottom was included, but no lid. This was done to not print unnecessary parts, as a lid could easily be printed later. This cuboid, in comparison to the first print and the author's hand, can be seen in Figure 4.4.

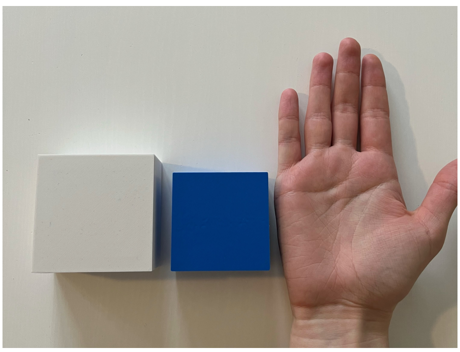
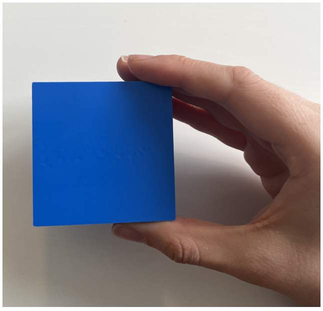
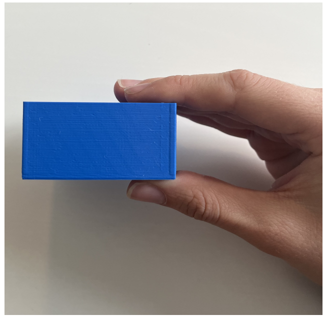


Figure 4.4.: First and second 3D print compared to a hand.

This time the base surface of the cuboid was 6x6cm, just within the scope of the maximum size [Figure 4.3]. This was to see how big the maximum would look and feel. To compensate for the large base and not make the cuboid unnecessarily large, it had a height of 3cm. Even though the cuboid felt a little too big when grabbed, having a height of 3cm makes it possible to grab the cuboid from the side instead of the top or bottom, making it easier. Figure 4.5a and 4.5b shows the two different ways the cuboid block can be grabbed, where Figure 4.5b is the easiest way a smaller hand would grab the cuboid. This shows that with a base plate as large as the maximum size it can still be used by adjusting the height.



(a) Cuboid grabbed from the front.



(b) Cuboid grabbed from the side.

Figure 4.5.: How the cuboid blocks can be grabbed.

Two more prints were made to see how it would look with a smaller base plate and height. Figure 4.6a and 4.6b show the printed blocks. The base plate for both is 4.5x4.5cm, but 4.6b has a height of 4.5cm making the block a cube, whereas Figure 4.6a has a height of 3cm, the same as with the second print [Figure 4.5a and 4.5b].

(a) 4.5x4.5x3cm cuboid print.

(b) 4.5x4.5x4.5cm cube print.

Figure 4.6.: The two different red blocks.

Having a base plate of 4.5cm instantly felt better in size when it was held, in regards to how easy it would be to grab by smaller hands. Consequently, 4.5cm is also implemented in the other blocks as their base size. For the height of the block, the lowest cuboid [Figure 4.6a] felt more comfortable when holding as it is better adapted to smaller hands.

The smaller the blocks are, the more fine motor skill is needed to grab them. A size between the minimum and maximum is therefore also seen as favorable, as they add a little more challenge to the exercises, and as a consequence, make completing the exercise more rewarding [52]. To further add complexity if needed, the different blocks could be printed hollow, without a lid, opening up the possibility for a child to grab it from within, which requires even more precise fine motor skills as the walls are very thin. Figure 4.6a and 4.6b can therefore be good candidates for use, both with and without a lid, but they should be tested with children to get an idea of the use. For now, the 4.5x3cm cuboid in Figure 4.6a is used further with the system.

4.1.2. Sphere

After having the cuboid sorted out, the sphere was printed, with a diameter of 4.5cm. To print the sphere in its entirety, supports were needed as it then had a larger overhang than 45°, seen in Figure 4.7. However as discussed in section 3.2, supports can create unfortunately ugly models or ruin them, if the supports are removed in the wrong way.

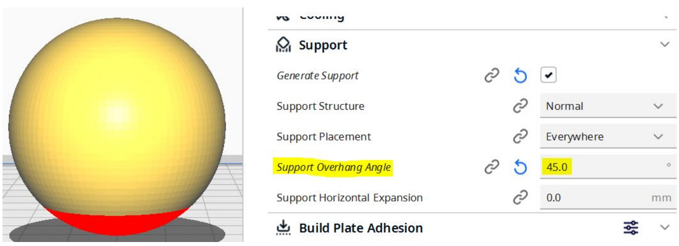
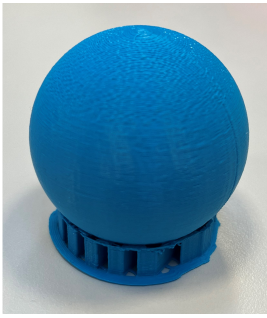
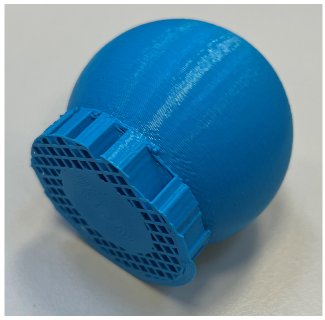


Figure 4.7.: Image from Ultimaker's slicer software Cura. The red area requires support as the overhang value is more than the threshold of 45° here.

Two identical spheres in terms of size, with a diameter of 4.5cm, were therefore created to see what the difference of having supports made to the physical model. The printed sphere with supports can be seen in Figure 4.8. The removal of supports was first done with a normal pair of needle-nose pliers, but as the print was rather small there was a limit of how much they could remove. The result after the initial removal can be seen in Figure 4.9a. Sandpaper was further used to sand down the rest, with the result seen in Figure 4.9b.



(a) Supports seen from the front.



(b) Supports seen from the side.

Figure 4.8.: Sphere printed with supports.

(a) Supports removed with pliers.

(b) Rest of supports sanded down.

Figure 4.9.: Sphere print with supports removed.

The other printed sphere was made to avoid supports. This was done by cutting the model in half, essentially making two hemispheres [Figure 4.10], that needs to be attached in order to create the sphere.
Even though sanding down the supports removed the excess pieces and sharp edges, it still shows on the model compared to the hemisphere print. It is however mostly a problem with aesthetics, as the sanded-down area only feels a little coarser than the rest of the print, not affecting the overall use of the sphere. The biggest downside of printing this way is the extra work needed after printing, where specific tools are needed in order to get a good result. In addition, you are not able to later decide the infill, controlling weight or sound as discussed in subsection 4.1.1.

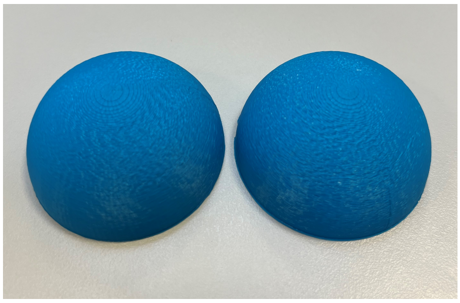
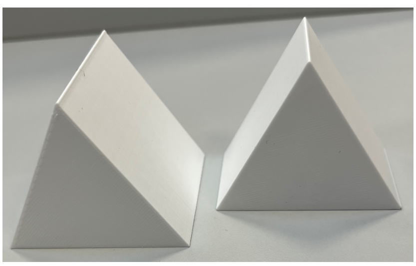


Figure 4.10.: Two halves of a whole sphere.

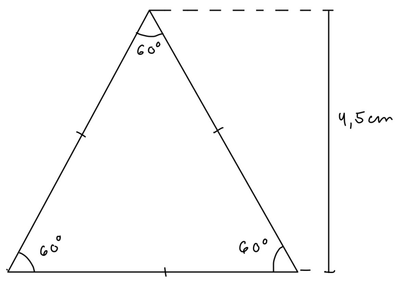
By printing hemispheres, you have the possibility of creating a hollow sphere. This would, however, require more work with difficulty attaching the pieces with a smaller surface area, in addition to removing supports from inside the hemispheres. By making the halves solid, as done in Figure 4.10, the only work needed after printing is attaching the pieces, where the most permanent and easily available solution is glue. With this solution, however, the sphere becomes more prone to breaking with rough use due to its vulnerability in the attached area.

4.1.3. Triangular prism

Because of the arguments mentioned with the cuboid [subsection 4.1.1], the height of the prism was set to 4.5cm. For simplicity, the geometry was made to be an equilateral triangle with 60° angles, illustrated in Figure 4.11b. As with the cuboid print, two different prints with lengths 4.5cm and 3cm were also made for the triangular prism, seen in Figure 4.11a, to get a feel of the size. Both were printed in their entirety, as no supports were needed and making them hollow was not a priority.



(a) The two triangular prism prints.



(b) Dimensions of the prism face.

Figure 4.11.: Triangular prism print and its dimensions.

With the same arguments made for the cuboid print, the smaller-sized prism feels better when holding as it is more adapted to smaller hands, and is therefore also easier to grab from different angles. The only possible issue regarding this print is the edges, which can appear sharp. However, printed PLA can be sanded down, as explained in subsection 4.1.2 with the removal of supports. This way sharp edges can be easily removed if they were to be a concern.

4.2. Box

With drawings on paper of how the system should look [Figure 2.6], one of the first things that was done in this thesis was to make a 3D model based on the drawings. This was done with simple sketching and extrusion techniques in Fusion 360, with the result seen in Figure 4.12.

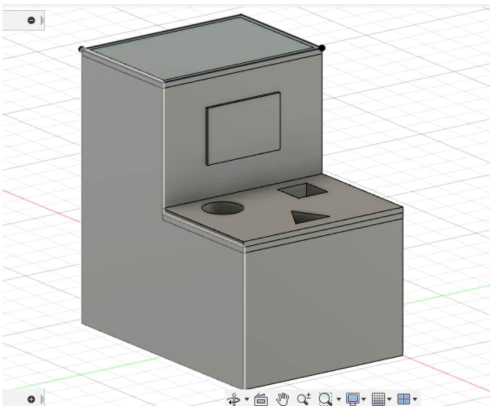
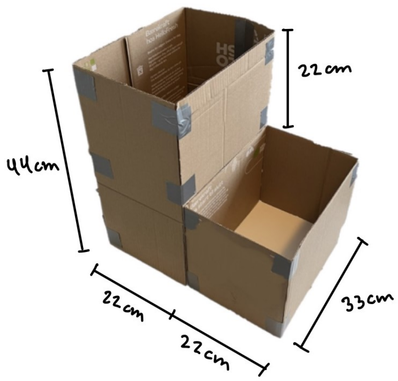


Figure 4.12.: First rough draft of a 3D model of the box made in Fusion 360.

The goal of this was not necessarily to get a finished 3D model but rather to see how the system would look in its entirety, and get a better understanding of what to do next. Because of this, shape and size were the two most important features to focus on with this first model. An extraction hole, the correct size of the display, and a way of attaching the board to the box were neglected as they all depend on the dimensions of the shell, and could be easily added or adjusted later.
In deciding the initial dimensions, the size of a standard laptop was used as a reference (roughly 33x22cm). This was chosen because of the familiarity of the size, as it does not feel excessively small or large in real life.

In order to further study the shape and size of the 3D model, a cardboard model based on Figure 4.12 was made next. Figure 4.13a and Figure 4.13b show how it turned out with measurements, and the size compared to a regular laptop.



(a) Size of cardboard model with dimensions.

(b) Size of cardboard model compared to a laptop.

Figure 4.13.: Cardboard prototype.

Since the system is supposed to be used with smaller people and in different settings, the size should reflect this. However, with children, it is important to consider that they usually prefer the ground when they play, and not sitting by a table on a chair. Because of this, the main intended way for the system to be used is on the ground. Both having the box on the ground and on a table was however tried.

When placed on the ground the cardboard model does not feel too large, as 22cm is more than enough for a small child to reach over. However, when it was placed on a table the size it felt a little too big. With 22cm from the table and up, in addition to the distance from legs to table, it is a bit too tall in order to be comfortably used.
To reduce the height of the box a few things need to be considered. As per the drawing in Figure 2.6, an extraction hole need to fit onto the front of the box, being minimum the height of the blocks, 4.5cm. Because children have smaller hands and can more easily reach into tighter spaces, the hole does not need to be excessively tall. A placeholder height for the size could therefore be roughly 8cm, as this would be enough for the hole to function as it should.
In addition to the hole, the thickness of the board with holes and the electronics underneath needs to be considered. A space of 10cm from the hole and up could therefore be implemented, to be on the safe side. With these estimates, the front has to be at least 18cm tall. The box can therefore be reduced by 4cm with good

conscience, making the system 40cm tall instead of 44cm.

By cutting the cardboard model 4cm and making an extraction hole 8cm tall, the modified cardboard model can be seen in Figure 4.14. The display was here only attached by duct tape to get a feel of the system in its entirety.

Figure 4.14.: Cardboard prototype with changes.

4.2.1. Board

The only requirement for the placement of the different shapes on the board is to be able to fit the sensors underneath, with enough space between the holes. The design chosen can be seen in Figure 4.14.
The shape is one thing, but the size of the holes compared to the blocks was also looked into. To not make it overly complicated to insert a block into a hole, 5mm was added to the holes compared to the sizes of the blocks. This way some wiggle room is created making it easier for the blocks to go into the holes. In addition, two main options were considered when deciding the size of the blocks compared to the holes.

Option 1
Having each block fit into all holes, with feedback mechanisms.

If you can fit more blocks into the same hole without any feedback mechanisms, as with the common box toy [Figure 2.3], the cognitive and logical learning potential is limited. The user potentially then never learns what the right block for each hole shape is, as they all seem to match. On the other hand, by having a display as guidance, this can be avoided by displaying error messages when the wrong block is inserted into the wrong hole. The user is then able to learn from their own mistakes, maximizing the learning potential of the system.

Option 2
Having each block only fit into their respective holes, not the others.

If one block only fits into one hole, logical thinking is more necessary with this option, where the user then has to correctly match the block with its shape. A block not fitting into a hole works as instant feedback to the user forcing them to try something else, making them learn what the wrong and correct solution is. A trial and error method could therefore also apply to this option but without the need for external feedback.

Both options promote logical thinking and can be considered good options for the intended system. However, option 1 would be difficult to implement in its entirety as IR sensors are used. As they have no way of detecting shapes or colors, the move would be considered correct as long as the right sensor is blocked, independent of what was blocking it. Option 2 is therefore more realistic because the IR sensors would then only detect the right blocks.

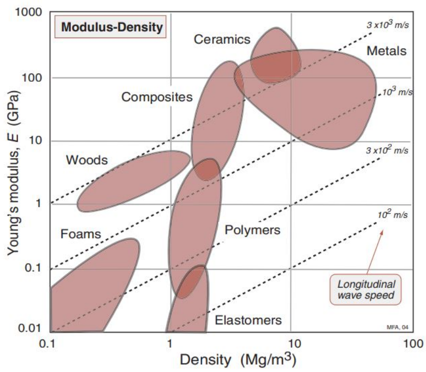
Despite this, a mix of the two options was implemented, not on purpose. Option 2 was the intention to make, but it was discovered late in the process that an error was made with the final cut of the board. Changing the design that late in the process would require making new blocks and a new board, demanding more work than it was worth at this stage. A closer look at the board and the blocks should therefore be looked into in future work.

4.2.2. Material

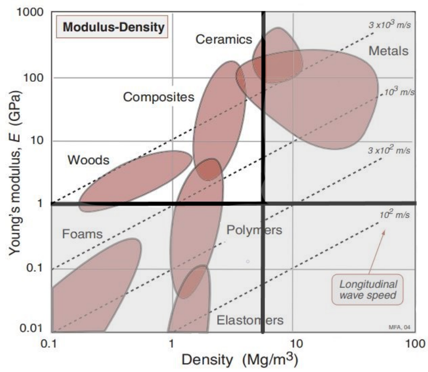
With measurements for the system decided and looked at with a cardboard model, it is possible to create it in a more permanent, and sturdy material. With material selection, there are different properties, attributes, and features to consider with each material to choose the best suited. Material selection can be crucial for designs that have requirements of for example strength, durability, chemical, or electrical properties. In the system presented in this thesis, where the size is limited and the usage is more for aesthetics than function, many of these requirements can be ignored when choosing material. Requirements of cost, availability, density, and stiffness become more important and are the biggest factors that are looked into in this chapter.

Material property charts are good tools for choosing materials as they condense a large amount of information into a compact accessible form and reveal correlations between material properties that can help in checking and estimating data [61]. Figure 4.15 shows a chart consisting of the different material classes in bubbles that are marked by the shaded regions. Density and Young's modulus are represented on the axis saying something about the weight and stiffness of the different materials.

As per the requirements for the system, it needs to be robust and easy to use in everyday settings, in other words, it needs to be stiff and fairly light. Materials best suited for these two requirements can be found in the upper left square, shown in Figure 4.15b. This indicates that woods, composites, some polymers, and metals are best suited.



(a) Material property chart.



(b) Chart with modifications.

Figure 4.15.: Material property chart [61, p. 48].

In Figure 4.15, woods and composites are separated but in reality, they are both classified as hybrid materials, consisting of two or more materials assembled in such a way as to get attributes the materials alone can not achieve. Hybrid materials are generally very good because you combine materials to get specific qualities. The downside to many of them, however, especially composites, is that they can be very expensive as they usually are difficult to manufacture.

Other than stiffness and density, one of the most important requirements for material in this thesis is availability and cost. Based on Figure 4.15 and this requirement, the different materials that are further discussed are, therefore, polymers, metal, and wood.

Polymers

There are two main categories of polymers, natural and synthetic [66], but for this use, synthetic polymers are the most interesting. In this category, you find different types of plastics that can easily be shaped as they can have large plastic deflections [61, p. 29]. One of them is PLA which is used for 3D printing [section 3.2]. Other plastics can also be used, but they are not considered here as they

are much harder to come by and design yourself. Contradicting the requirement that the system should be easy and inexpensive to make and replicate.
In comparison to other plastics and most polymers, PLA is more environmentally friendly as it is made from renewable sources and can remain environmentally friendly if the correct way of disposal is used [100]. 3D printing is also becoming more common and is a remarkable way of creating, as discussed in section 3.2. This makes it a great choice for creating the shell of the system.

However, as mentioned in section 3.2, the printing plate is rather small, making it difficult to print the entire surfaces of the box. The box would therefore be printed in smaller pieces that later would have to be attached. With the same arguments made for the blocks, specifically for attaching the hemispheres in subsection 4.1.2, with the blocks being more prone to breaking, this solution would not be favorable to create a robust system. In addition, correct disposal of PLA is harder to achieve and requires more work, making it less beneficial if it often breaks and needs replacement.

Metal

In prototyping and making a larger system where durability and strength are required, metal is often the first choice. This is because they have superior durability, are heat resistant, and are very ductile, making it easy to predict their behavior under different loads and conditions. It is even possible now to 3D print metals, enabling more complex geometries with metals than what regular manufacturing processes can do [37]. But with great power, sadly also comes great cost as 3D printing metals can be very expensive and hard to come by as specialized tools and resources are required. Another disadvantage of using metals for this specific system is the presence of sharp edges that can be a safety issue in handling and usage, especially for children. They can also be very challenging to modify once they are shaped, making them a bad choice for having flexibility in design iterations.
Even though metal is great for constructing robust and enduring prototypes, they are not that suited for this system and its prototyping process as they fail to meet the requirements of flexibility and low cost.

Wood

Woods are a natural composite consisting of cellulose, lignin and other polymers [61, p. 297], making them a sturdy and light material with significant strength and stability traits. Different techniques can be used to create composite materials consisting of wood, enhancing their attractive qualities. Plywood for instance, is made as a laminate where layers of wood are glued together with the wood grain rotated up to 90 degrees relative to each other in each layer, making the material stronger in multiple directions [84]. Woods, in general, can also easily be cut, shaped, and joined, without the need for very specialized tools and equipment, allowing for quick adjustments and iterations during a prototyping process. Additionally, they have a natural appeal and warm aesthetic, making them suitable for use with frequent human interaction.

As wood is the material with the most benefits per the non-functional requirements for the box, and easily available with the work on this thesis, this material was decided to use for this prototype. Other materials can however be considered in future work.

4.2.3. Laser cutting

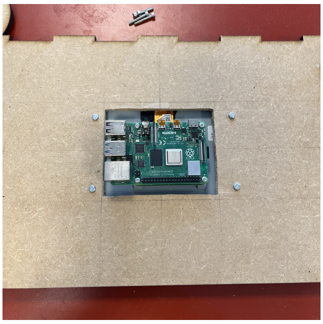
Wood can easily be modified according to specific designs, but that can also be a time-consuming task, especially with detailed work. To work efficiently and save time, a laser cutter was used to create the shell of the system. Each side of the box was modified with finger joints in Fusion360 before they were cut, making it easier to piece the sides together afterward. Neither the extraction hole nor a hole to attach the display was cut in this process to first get a feel of the box in its entirety before adding details, the same procedure as with the cardboard model. The result and laser cutting machine can be seen in Figure 4.16.

(a) Box with board. **(b)** Box without board. **(c)** Laser cutting machine.

Figure 4.16.: Laser cut box and machine.

With using finger joints the model can be smoothly assembled and disassembled without the need for more permanent attachment methods, making it simple to modify the box and the different plates during prototyping without affecting other parts. The board is the only plate that does not have finger joints and is only placed on top of the side and front panels, as seen in Figure 4.16a and 4.16b. Due to time constraints and the scope of the work concerning this thesis, a more secure placement was not prioritized. It should however be looked better into in future work with the system, especially if it is tested with children.

To make a hole for the display, the area between the attachment points for the bolts was measured. A square with these dimensions was then drawn in the middle of the front plate before cut out and the display fastened with bolts. Figure 4.17 shows how this looks, with the entire square not cut out as the display fit nicely, without the need of making it larger.



(a) Display seen from behind.

(b) Display seen from the side lying on a table.

Figure 4.17.: Display attached to the front panel of the box.

The extraction hole was made similarly by first drawing a 14x8cm opening at the bottom of the plate. It was then cut out using only a saw and some sandpaper. The corners of the hole were also rounded to avoid sharp edges. In addition to these changes, a last important detail made was a hole in the side of the box for the power supply cord to the raspberry pi. The entire box with these details and the attached display can be seen in Figure 4.18.

Figure 4.18.: The finished prototype.

4.3. Display

The process of creating a GUI for the display is done in three distinct steps. First, a conceptual design of the GUI is created with the interface design software Figma [34] to visualize the requirements. Second, the designs from Figma and the logic implied by them were programmed using a text-based user interface (TUI) approach, where the goal was to understand the sensors and the communication between the Arduino and the Raspberry Pi. Lastly, a GUI was made based on the visual designs from Figma, and the code from the TUI. This GUI is then evaluated by others in user testing [see chapter 5]
It is important to note that the choices in design made here are largely based on design conventions, which are commonly known guidelines that are culturally learned. Hence the conventions followed in this work are based on the author's own experiences with interaction design, and theory from a course at NTNU [86].

4.3.1. Figma

Three aspects should be considered when creating the conceptual design for the display, and laying the foundation for the use of the system. Firstly, as the blocks and board in this prototype are fixed in size, the display should be the main aspect to increase difficulty with levels, as explained in section 2.4. Secondly, to get a well-integrated system, the display should also be the main source of information, and work as a guide for the user. Lastly, the display needs to implement feedback to the user to contribute to the overall motivation for its use.

To implement these aspects in an organized and user-friendly way, six design principles are taken into consideration [38, ch. 1.6.3]. Following is a summary of what the different principles entail. Before explaining how the game and levels are designed.

- **Visibility**
  - How visible different functionality is. The more visible something is, the easier it is for the user to understand its use, and figure out the next move.
- **Constraints**
  - Refers to ways in restricting user interactions. This is used to better guide the user on what to do next.
- **Consistency**
  - Refers to creating similarity in the design, where similar elements are

used for similar tasks, and the system has similar operations throughout.

- **Affordance**
  - What different functionality implies with its look and design.
- **Feedback**
  - Same definition as before, but in GUI design more specifically it is providing information to the user about what actions have been taken and what state the system is in.

For the user to be in control of when the system starts, and to have a soft start to the game, a main menu was first created. As the intention of the system is to later develop more boards with different functionality, a main menu was also made to function as a unifying element between the different boards. Where each board and the corresponding game is started from this menu. Its design in Figma can be seen in Figure 4.19.

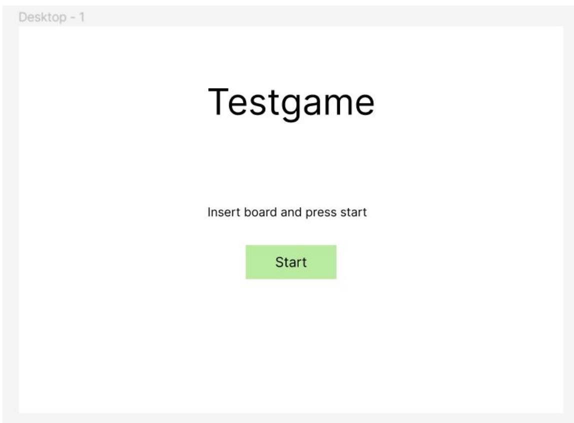


Figure 4.19.: Main menu.

From Figure 4.19 "Testgame" is seen at the top, a placeholder name for the system. Underneath is text saying to "Plug in a board and press start", and lastly a colored square with text saying "Start". With text saying to press start, as well as having a colored square behind the word Start, the intention is to suggest the affordance of a button to the colored square where pressing start starts the game. The color surrounding "Start" can also be noted here. It is used as feedback to signalize to the user something about its state. Green implies that it can be pushed, as green often is a color associated with success [64]. If the button was red on the other hand, it could signalize that the button was invalid or not pushable, as red is associated with error or failure. If multiple boards are implemented, the "Start" button could be colored red as long as no board is inserted, and change to green when a board is detected, providing feedback to the user that the board is accepted and they can start the game. Red and green are also used actively further in the designs with these meanings. For this prototype it is adequate, but if color blindness should be accommodated, green could be swapped out with blue as the colors have similar qualities [64].

After having pressed "Start" the users are taken to a level select page, seen in Figure 4.20, where they can choose the different levels.

Figure 4.20.: Level selection menu where level 1 is completed, level 2 has not been played yet, and level 3 is unavailable.

The same color design as the main menu is also used in Figure 4.20 to bring consistency to the system, implying that the level boxes are buttons as well and can be pressed. The color red is used to signalize that the buttons can not be pressed, the grey is used to signalize that the level has not been played yet, while the green indicates that the level is completed. With implementing this type of constraint, the goal is for the system to guide the user in which order to play the different levels. In addition, the intention with the colors is to have them change according to what levels the user has completed.
Another aspect that can be seen in Figure 4.20 is the number of levels included. As per the limitations of this work [section 1.2], only three levels are implemented in this prototype. This is done to limit the workload and show the basics of intended functionality in an organized way. This way, by also making it very general, it should be easy to further build on this prototype by making more levels and implementing other features.

When a level is pressed from the level select page they are taken to the different levels and the game starts. To have consistency in the system with the different levels, a base [Figure 4.21] is first created where different elements are later added. These elements contribute to making the levels distinct and more complex as they progress, with mainly logical thinking being challenged, and not fine motor skills as much.

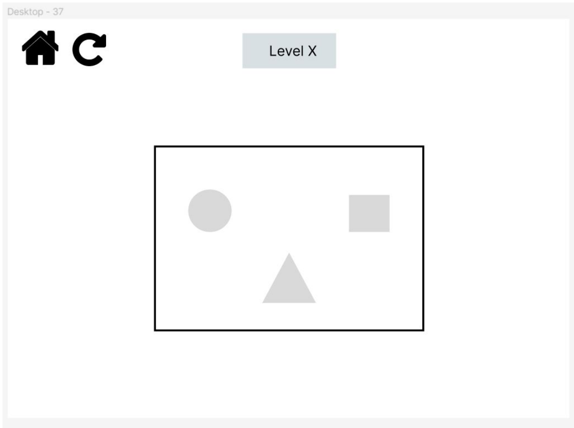
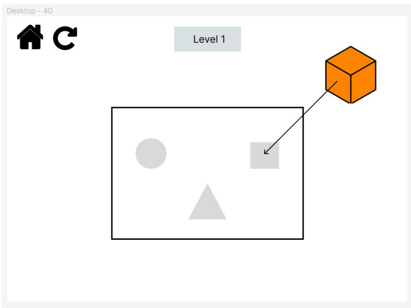


Figure 4.21.: Base for the levels.

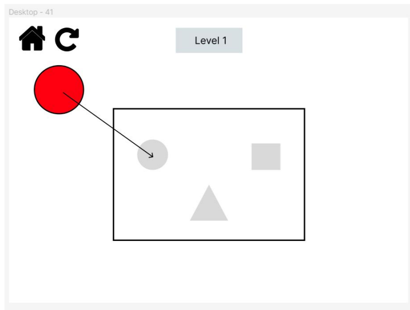
To make the system more user-friendly and intuitive, two functionalities were added to the base. A way of exiting the level, and a way of restarting it. To represent them, design conventions were used as symbols to strengthen their affordance, respectively a house and a bent arrow, as seen from Figure 4.21. In addition, "Level X", is used as feedback to indicate what level the user is on. The intention with the box around it is to change color to green when the level is complete, keeping the consistency of color established earlier, and providing feedback to the user of the systems state. If a level is selected again after it is completed, the "Level X" box stays green to indicate to the user that the level has already been completed. The X is further substituted by 1,2 or 3 depending on which level is played. However, by using a box around the text with color, the "Level X" box could for some users be confused as a button as its design is consistent with the button design established on the previous pages. Nevertheless, to keep with the consistency of the design, the "Level X" box could be made into a button, where pressing it for instance returns you to the level select page. By implementing this alternative home button, the user then has two different ways of returning to the level select page, making it adaptable to different thoughts and use, and keeping the button design consistent in the system.

As implied in section 2.2, the overall task of the game is to put blocks in their corresponding holes on the board. Each level is then implemented as tasks related to this. To convey this to the user, the square with figures inside, in Figure 4.21 is made to represent the physical board with its corresponding holes. To further indicate what blocks should be put into the board, and in which order, figures representative of the actual blocks are added to the base, along with arrows from the figure to its correct hole [illustrated in Figure 4.22, 4.24 and 4.25]. Each level is then made distinct, and gradually more difficult by using feedback, constraints, and adding elements to Figure 4.21.

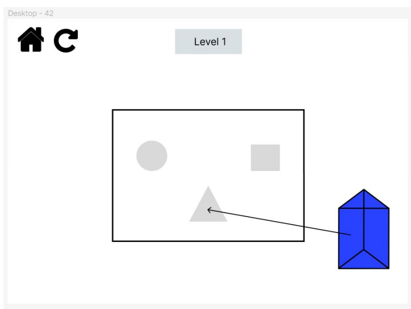
Since level 1 aims to be the easiest, and work as an introduction of the system to the user, it should be very intuitive how to play. This is done by implementing feedback and constraints to the base [Figure 4.21]. The level is further divided into three different steps the user has to do to complete the level, which can be seen in Figure 4.22.



(a) First step of level 1 with cube.



(b) Second step of level 1 with sphere.



(c) Third step of level 1 with triangular prism.

Figure 4.22.: The design of level 1.[3]

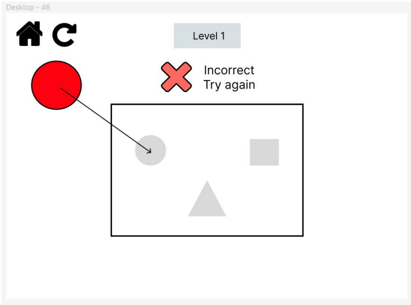
Because the Figma was made before the blocks, the color of the different shapes

[3]The icon used for the triangular prism and cube in these figures, as well as all other figures in this thesis is from the Noun Project [8]

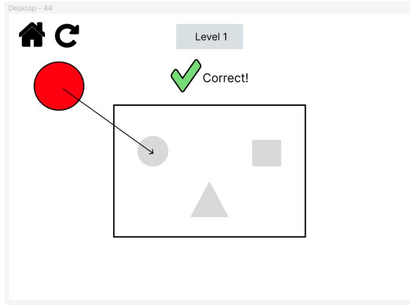
does not correspond to the actual blocks. They were only chosen at random to illustrate the point that the blocks were going to have color.

When the user enters level 1, Figure 4.22a is what greets them. For the user to progress, the page indicates that the user has to put the cube into the square hole. When that is done the page changes to Figure 4.22b, and the user then has to put the sphere into the circle hole. The last step is for the user to put the triangular prism into the triangle hole, as seen in Figure 4.22c, before completing the level. By only giving the user one block at a time on the display, their choices of action are limited, and they do not have to guess what their move is in any way.

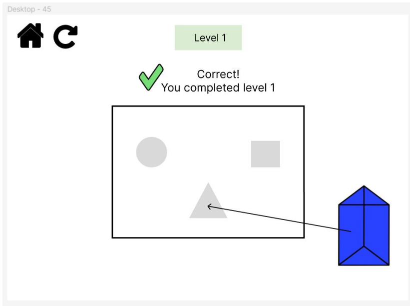
To make it even clearer for the user what the right and wrong moves are, feedback is also included. Feedback for an incorrect move is shown in Figure 4.23a, correct in Figure 4.23b, and feedback for when the level is completed can be seen in Figure 4.23c. The symbols and the corresponding text for correct and incorrect moves show up if the user either does right, or wrong per the displayed page. The hope is also to have the feedback show up and disappear after a couple of seconds. This way, the feedback pops up each time a move is made, clarifying to the user what consequence each action has. If the labels stay on the screen until the next block is inserted, and for instance, two incorrect moves are made after one another, the user does not get any new feedback on the screen. The incorrect message from the last move only appears to be trailing or falling behind. This behavior could be confusing and not provide the user with the information they need to understand the levels.



(a) Feedback when the user inserts the wrong block.



(b) Feedback when the user inserts the correct block.

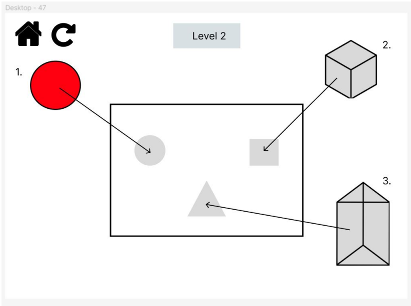


(c) Feedback when level 1 is completed.

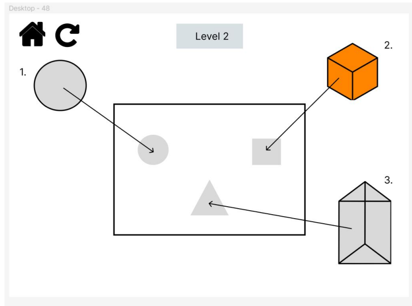
Figure 4.23.: Feedback for correct and incorrect moves, illustrated with level 1.

The symbols regarding the feedback, seen in Figure 4.23, are also chosen regarding design conventions, with red and green added to further enhance their meaning. When the level is completed, in addition to the "Level 1" box turning green, a different text appears. By doing both, the intention is to make it more clear to the user that they have completed the level, and not just correctly placed the block.

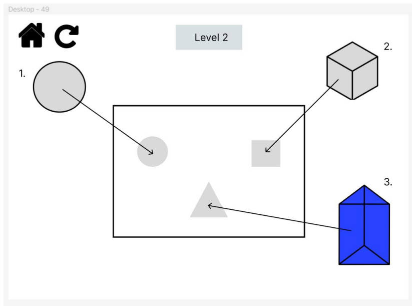
After level 1 is completed the user then can choose to play level 2 by going back to the level select page, where the "Level 2" button is now grey [Figure 4.20]. Similarly to level 1, level 2 is also divided into three steps, but now in a different way, as seen in Figure 4.24.



(a) First step of level 2.



(b) Second step of level 2.



(c) Third step of level 2.

Figure 4.24.: The design of level 2.

The base [Figure 4.21] is also present in this level, as can be seen from Figure 4.24, but the level is notably different than that of level 1. Where level 1 only showed one block at a time, level 2 shows all of the blocks at the same time, but the order is made clear by indicating the next block with color and a number. Having more elements on the page is meant to force the user to study the page a bit before starting with the level. In addition, to increase the complexity of the level from level 1, two new features are implemented. The order of how you put in the blocks is changed, with the sphere going in first instead of the cube, further enhancing the need to study the level before starting. And, as punishments in games can introduce an element of risk and challenge, consequently leading to an increase in satisfaction when the goal is achieved [51], a larger consequence is added when you make a mistake. If the user inputs the wrong block into the wrong hole, the incorrect message Figure 4.23a pops up, in addition to the page changing back to Figure 4.24a, implying that the user has to start the game over. In level 1 the consequence of making a mistake was not being able to progress to the next block, but in level 2 they are forced to do it all over again, demanding more of the user to complete the level.

After the user has completed level 2, the level 3 button on the level select page becomes grey, and available to press. Unlike levels 1 and 2, level 3 is not divided into three steps. Here all of the information about the level, throughout the level, is given on one page, sen in Figure 4.25.

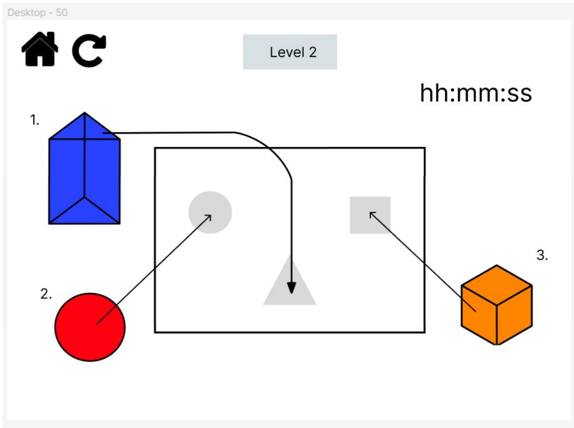


Figure 4.25.: Design of level 3.

To gradually increase the complexity again, the features added in level 2 are also implemented here, along with three new features. As with level 2, the order of the blocks has changed again, with the triangular prism now being the first block to insert. The added feature of this is the placement of the figures on the screen. In levels 1 and 2 each figure is placed on the same spot, despite their order. By moving them, the goal is to make the user study the level even more so than in level 2, where they now have to understand the new order and what block goes into which hole with the arrows. The second added feature to this level is the visibility of the figures. In level 1, only one figure and one colored block were visible on the screen at a time. In level 2, all of the figures, but only one colored block were seen. In level 3, all of the figures and all of the colored blocks are seen at the same time, further adding to the user's need to study the level before starting. The order is now only made clear by the use of numbers, and not color as with the other levels. But as level 2 shows [Figure 4.24], the order is here denoted by the use of color and numbers. The goal of this is to prepare the user for level 3, with numbers slowly being incorporated, and not added as an entirely new feature in level 3. The third new feature of level 3 is a timer, seen in the upper right corner in Figure 4.25. By adding a timer, stress is added to the level as the player has to complete the level before the time runs out, or else they have to start over. This is implemented to increase the difficulty with using fine motor skills, as the user has to be in control of their hands and movement to complete the level on time. This timer can further be implemented with a scoreboard for instance, showing the user how far they have gotten with their fine motor skill

exercises by how long it took them to complete the level. A similar feature to that of the Wii Fit game for instance, where the progress of the user is tracked to motivate improvement [see subsection 3.1.1].
In addition to these features, the consequence of making a mistake from level 2 is also implemented in this level. As all of the blocks are visible on the screen in level 3, it is difficult to show the user what the order is when a mistake is made, as no blocks change color. To solve this, level 2 introduces the user to the consequence of going back to the start when making a mistake, teaching them this feature in the hope that the user intuitively will assume the same for level 3. However, if the user manages to do level 2 without mistakes, level 3 suddenly becomes even more complex as they then have to guess the order after making a mistake.

After having completed level 3, the game is finished, and each of the buttons on the level select page turns green. They are however free to play the different levels as many times they would want.

4.3.2. Programming

After the different levels in Figma were designed, and the different features of the game were determined, the TUI was made with Thonny and Python. This was done to understand how the logic from Figma could be implemented with code. And understand how communication between the Arduino and the Raspberry Pi worked with the sensors.

First, the communication between the Arduino and Raspberry Pi was established as described in subsection 3.3.2, and tested with a few lines of code with the sensors. With an established connection and the sensors working, the logic of the different levels was programmed in Thonny and the Arduino IDE [Listing C.3 and C.2]. The game is then played by instructions in the output terminal and by blocking the correct sensors. An example of output from this code can be seen in Figure 4.26.

```
>>> %Run 'level1&2.py'
 Serial ok
 level 1, 2 or 3? or quit (q):2
 You have to complete level 1 before doing the others
 level 1, 2 or 3? or quit (q):1
 Sensor 1 is blocked, Correct!
 Sensor 2 is blocked, Correct
 Sensor 3 is blocked, Correct

 You completed level 1!
 Want to do it again?
 y or n:n
 level 1, 2 or 3? or quit (q):3
 You have to complete level 2 before doing level 3
 level 1, 2 or 3? or quit (q):2
 Sensor 2 is blocked, Correct!
 Sensor 1 is blocked, Correct
 Sensor 3 is blocked, correct

 You completed level 2!
 Want to do it again?
 y or n:n
 level 1, 2 or 3? or quit (q):3
 level3
 level 1, 2 or 3? or quit (q):q
 closed
>>>
```

Figure 4.26.: Output when Listing C.3 is run, and played without errors.

In the code [Listing C.3] and in Figure 4.26, sensor 1 corresponds to the square hole, sensor 2 to the circle, and sensor 3 to the triangle. What can also be noted from the code and its output is the missing functionality of level 3.
Since the only new logic level 3 introduces is a timer, whereas the rest is a combination of the logic already made for levels 1 and 2, the level is not written intentionally to save time. A timer is assumed to not affect the structure made for levels 1 and 2 [as seen in Listing C.3] so it is assumed that the code does not require much extra work, and is therefore rather saved for the GUI.

The logic intended for levels 1 and 2 from the Figma explained earlier can also be seen in Figure 4.26. Constraints are put in place by not being able to choose a red level, in this case, level 2 or 3 if you have not completed level 1. And the order of what block to put in, or in this case, the order of which sensor to block, are different. In addition, feedback is given to the user for doing correct and incorrect moves. Only feedback from correct moves is shown in Figure 4.26, but the incorrect message can be seen in Listing C.3, along with the logic for the game and the levels.

Level 1 and 2 are both defined, as indicated from Listing C.3, with each level in its corresponding function, Level1, and Level2, with two variables as input. The level1 and level2 variables are used to control the sensors and their order. In level 1, if the first sensor is covered correctly level1 is set to 1, if the second sensor then is covered after, level1 is set to 2, and then 3 if the last sensor is blocked. In level 2 it is also used to implement the feature from the Figma of starting over when a mistake is made. This is done by setting the level2 variable to 0 when the user blocks the wrong sensor, making the loop start from the beginning. The Correct1 and Correct2 variables are used to indicate what level is completed in the main loop of the code.

4.3.3. GUI

As explained in subsection 3.4.1, to create a graphical user interface, PySide and Qt Designer was used with Python and Thonny on the Raspberry Pi. In addition, to make the code more readable and organized, the different levels were implemented in separate classes, named level1, 2 and 3 accordingly. A game class was also made, called game, and referenced to in the levels as motorgame. In this class, the overall logic of the game is controlled, and the communication with the Arduino is opened. The communication was first tried to be opened in each level, to avoid having it open when it did not need to, but after troubleshooting for far too long, it was moved to the game class. In addition to this, the game class also holds the functionality of the main menu and the level select page in the GUI. Lastly, there is a main class [subsection 3.4.1], that runs the entire program.

The same order as for the Figma was followed when making the GUI, with the main menu first, then the level select page, and lastly the three different levels. The first change made was the button design. To further add visibility to the buttons they were made to be wider and better fit the screen. In addition, to make their design more approachable and inviting, the corners were rounded [57]. This was done to the "Start" button on the main menu and the buttons on the level select page. The "Level X" info box on the top of each level [seen in Figure 4.21] however, was not made longer, and the corners were only rounded a little. By doing this, the goal was to make the buttons and the information box more distinct to limit the confusion about it being a button. The functionality of the info box being an alternative home button is however still implemented, just in case. The different new designs can be seen in Figure 4.27.

(a) Button design for the buttons on the home menu and level select page.

(b) Design of info box, illustrated with level 1.

Figure 4.27.: The design of the buttons and the info box in each level.

In addition to this design, feedback was added to the buttons. A pushable button is indicated by it getting a brighter shade in color, and getting a black border when

it is hovered over by a mouse. If the button is pressed, it turns to a darker shade than the original, and the border is added now as well. An illustrated example can be seen in Figure B.1. Not pushable buttons, like the red ones [Figure 4.20] and the info box mentioned above, did not get feedback added to them. This is to further imply the constraints they have and strengthen the affordance of the buttons. Furthermore, if an attempt is made to press a red button, a message appears on the screen [Figure B.2] for 3 seconds by using a timer from the Threading module [45], instructing the user to complete either level 1 or level 2 first, depending on which button was pressed.
Functionality was also included to consider completed levels. For instance, If levels 1 and 2 had been completed and then level 1 was played again, by pressing the home button, the user would be taken to the level select page where the Level 1 and Level 2 buttons are green, and the Level 3 button grey. In other words, the system would recall the state of the game independently of levels being played multiple times. Except for these additional features, the main menu and the level select page are the same as those from the Figma designs, with their intended functionalities from the TUI.

When it comes to the levels of the game the design of the board and blocks used in the GUI were made separately using the free online tool Photopea, an advanced image editor. They were then saved as jpg or png files, and added to Qt designer in a stackedWidget, making it possible to switch between pages in the code according to the state of the game. These designs were made similar to those in the Figma, with only minor adjustments. Firstly the positioning of the blocks on the display was lowered a bit. This was done for the images from Photopea to be shown satisfactorily and visibly, and better showcase the feedback messages.
In addition, the colors of the different blocks were changed to correspond to the actual blocks. An example of these changes can be seen in Figure 4.28, which show how level 3 looks in the GUI [level 1 and 2 are illustrated in Figure B.3 and B.4].

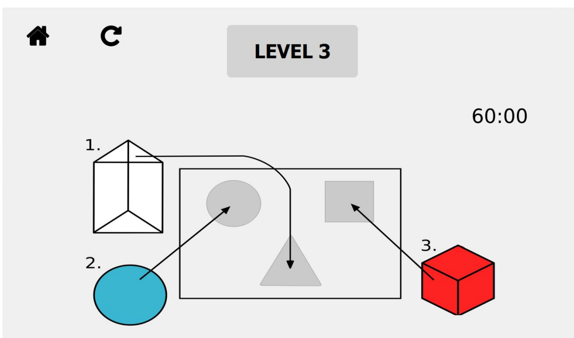


Figure 4.28.: Level 3 in the GUI.

The feedback of each level presented in subsection 4.3.1, was also included for each of the levels in the same place as they had in the Figma [Figure 4.23]. The messages for correct and incorrect moves were altered a bit to fit the screen better but still convey the same messages [Figure B.5].

After the GUI was created, it was quickly tested to see if the feedback mechanisms worked as they were supposed to with the sensors. A couple of small errors were then discovered. The initial setup of the Arduino code was sending data to the Raspberry Pi at all times with a delay of only 500, as seen in line 27 in Listing C.2. Because of this, if you held over the correct sensor for too long, the GUI would register it as correct, but immediately after also register it as incorrect. Providing the correct feedback mechanism first, and then immediately after the incorrect feedback message. For level 1 this was not that severe, but in levels 2 and 3, where you were taken to the start of the level each time you made a mistake, this was crucial to figure out before user testing. As an optimal solution, was not found in time, the temporary solution became to change the delay on the Arduino [line 27 in Listing C.2], from 500 to 1500, to slow it down before the next sensor value was transmitted. Hopefully giving the user enough time to insert a block before it is registered as wrong.

In addition, a problem related to getting the messages to disappear after a few seconds on the display was discovered as well. In step 2 of level 1 [Figure B.3b] and level 2 [Figure B.4b], the wrong feedback messages showed up. If the first block was put in correctly, when the second block then was put in correctly, the incorrect feedback message first showed up on the screen, before switching to the correct feedback message, and then moving on to the last step. As this behavior was very confusing, to not have this fault in the user test, a temporary solution was to let the incorrect messages stay on the screen after they appeared. In some way, this also fixed the issue of showing an incorrect message before the correct message. With this solution, the display now showed the right messages with each move, independent of previous correct or incorrect moves. However, the incorrect messages stayed on the display until a correct move was made. The correct messages, on the other hand, disappeared after about 2 seconds of being on the screen. The system was then tested with this functionality, and further problem-solving was looked into after testing [section 5.6], as getting the messages to disappear after a few seconds is the intended way for them, as mentioned in subsection 4.3.1.

Except for these faults, level 1 and level 2 were implemented with the same logic presented in the Figma and TUI. For level 3 on the other hand, as explained in subsection 4.3.2, it was coded in its entirety with the GUI. As expected, with the logic being a combination of levels 1 and 2, no major trouble occurred in the making of the basic functionalities. As most of the kinks with other functionality also had been worked out in the making of levels 1 and 2 with the GUI, the logic was simply copied over to work in level 3 with minor adjustments. The only thing missing to make level 3 the way it was intended, was adding the timer functionality. The Time module [90] in Python was used for this purpose. The timer and its design were made entirely in the code, and added from there to a label made in the Qt Design application. Its logic can be seen in Listing 4.1, where line 1 is defined at the beginning of the Level3 class, while the rest is written in the main logic function in the Level3 class.

```
1   self.starttime = time.time()
2
3   self.countdown = round(60-(time.time()-self.starttime),0)
4           motorgame.label_5.setText(str(self.countdown))
5           time.sleep(0.01)
6           if self.countdown == 0:
7               break
```

Listing 4.1: Snippet of code from the Level3 class.

As seen from Listing 4.1, the timer is initially added with 1 minute, or 60 seconds. This can easily be changed by substituting 60 with the time preferred. 1 minute was however chosen for the testing to showcase the timer, and give the participants enough time to both play the level and reflect on its functionality. From Listing 4.1 it can also be seen that nothing particular happens when the timer runs out. It simply breaks the loop and the user is not able to receive feedback when inserting blocks at this point. The reason for this was that the making of levels 1 and 2 took longer than anticipated, because of problems in PySide and Qt Designer. In addition, there was no time to figure out how to sync the refresh button to the timer. Consequently, when the refresh button is pressed, the timer just keeps counting down, but the level is restarted.

Chapter 5.

Evaluation of prototype

By having a working prototype, it is necessary to evaluate the results for assessing, and implementing, further improvements, as stated by ISO in ISO 9421:210 [43]. With an evaluation, you can obtain a better understanding of user needs, gain feedback that can be used to improve the designs and assess the requirements in whether or not they have been achieved.

A first assessment of the current prototype is therefore done as an opportunistic evaluation [38, ch. 13.3.5]. The overall goal of this process is to gather more information about the system as a whole to later improve, and further test the system. A research question can therefore be specified as the following:

How does the system feel in use, and how can it improve for further testing with the target user group?

In the following sections, the evaluation and its methods are justified and designed to obtain an answer to this research question.

5.1. Evaluation methodology

To receive valuable feedback from the participants regarding the research question, the entire system, not specific parts, should be tried. As the functionality of the system is rather limited in what it can do, this is possible without using an enormous amount of time or resources. This is further done by mixing inspection methods [38, ch. 13.3.3] with a qualitative usability test [55], in controlled environments [38, ch. 7.6.2]. However, When it comes to the participants, the intended user group is not used in this evaluation, even though it usually is recommended in usability testing. This is because getting children to partake in an evaluation can be a long and demanding process, requiring more time than what is possible for the scope of the work in this thesis. Furthermore, by using adults who have no trouble with fine motor skills you are quickly able to assess if the system is made too complex for a child, based on the adults' level of difficulty.

Usability testing is a method where the main goal typically is to determine the usability of a system with the intended user group [38, ch. 14.2]. The approach involves collecting data using a combination of methods. For this evaluation, observation of the participants and interviews are used. Through those, the goal is to obtain feedback about the functionality, usability, design, and aesthetics of the system as a whole. The focus is specifically on how the participants perceive the interaction between the board and the display.

Inspection methods, on the other hand, are typically used *"...to predict behavior and identify usability problems, based on knowledge of usability, users' behavior, the contexts in which the system will be used, and the kinds of activities that users undertake."*[38, ch. 13.3.3]. Therefore, these methods often use people with knowledge of interaction design and the typical behavior of users to get the most valid results.

It is important to note that the findings from such evaluations are good indications rather than valid results. For the results to be valid, the participants would have to be representative and it should be tried out in settings representative to its context of use [38, ch. 13.6.2]
However, since the purpose of this evaluation is not to say for certain how the system is, but rather give some indications of how the prototype is at its current state, these methods and their respective results are more than adequate.

5.2. The participants

When choosing participants for an evaluation, 5 is a recommended amount [47]. Fewer is however also acceptable in the use of quick feedback. Having many participants could result in more representative results, but would also be more time-consuming, which is not favorable with the work of this thesis. Regarding this evaluation, a few people should be more than sufficient with the goal of the evaluation in mind.
As fewer people are used, to strengthen the validity of the test results, one person with more knowledge regarding children's behavior, and one with more familiarity with interaction design are used in the evaluation. In addition to them, 3 adults are also chosen to give feedback.
The background of these three participants should, however, be considered. They are all people in their 20s who study or have studied engineering practices and are therefore quick to learn new and unfamiliar technology. This can affect how

they interact with the system and make the results less valid as it is the opposite of how the intended user group would use the system.

5.3. The evaluation

To get feedback on the system in its entirety, thus answering the research question, the participants are presented with the system individually and are asked to finish each of the levels, whilst thinking out loud. In other words, 5 separate, but identical evaluations are performed with the current prototype.
By giving the participants little to no information in advance about the system, the goal is to assess how intuitive the system is to start and use without needing help. The think-aloud method is used to give the observer a better understanding of the thought process behind [38, ch. 7.6.2] their actions. In addition, explaining how and why they navigate through the levels and use the features gives a better indication to the observer how hard or easy they find it to use.
With this method, in addition to observing performance and asking interview questions after they have tried the system, functionality, and usability are evaluated. Design and aesthetics on the other hand are evaluated only with questions before and after the evaluation. Asking the participants question regarding design before they test the functionality is done intentionally to solely get reflections on the appeal and design of the system. This can give some indications of how the design initially feels, and give feedback on how it can improve to make it seem more inviting and interesting to new users. After the participant has completed the levels, they are asked again what their thoughts on the design are. This time it is done to get feedback on how the design and functionality work together and how it contributes to the system overall.

To create a trusting and safe environment for the people that partake in the evaluations, and to minimize apprehensiveness in a rather formal setting, some precautions are taken. Before each evaluation begins the participants are told that they have the option to abort the test at any time if they feel discomfort. They also get information regarding who the observer is, what equipment is used, and how the evaluation is conducted. To bring more credibility to the evaluation, each participant is treated the same way [38, ch. 7.6.2]. A script for how each evaluation is conducted, including the interview questions, can be found in Appendix A.

5.4. Results

From observing the participants with the system, and supplementing with interview questions, problems regarding their experiences are observed. These problems are what the evaluations aims to identify, as they can be used to further develop the system and its functionality. They are presented in Table 5.1, along with an ID and a measure of severity, either low, medium, or critical. These measures are used as an indication of how serious a problem is to the system and its requirements, and how they should be prioritized further in the development. Deciding what measure each problem receives is based on three distinct factors:

1. The number of participants that discovers a problem.
    - A problem being encountered by multiple participants implies that it is not limited to a specific individual, but is rather a general issue in the system.
2. Who discovered what problem.
    - As mentioned in section 5.2, one person who has experience with children, from now on denoted as user 1 (U1), and one with experience in interaction design, user 3 (U3), are among the participants in these evaluations. As they are considered to be the "experts", their opinions and feedback regarding their respective fields are considered to be more valuable than that of the other participants.
3. How the problem affects the entire system.
    - Problems occurring during testing preventing the participant from completing their task are considered more critical than issues related to design. A system that does not work can not be tested properly later.

In addition to the problems, solutions to each of them are presented in Table 5.2. They are based on statements from the participants and the author's knowledge of the system. They are in addition each marked with a priority based on the level of severity of their respective problems from Table 5.1, and how the related problem affects the functionality and the requirements of the system. The priority is meant as a guide for what and how things should be changed in the system before a potential next evaluation.

Table 5.1.: Problems identified by the different participants during the evaluations.

ID	Identified by					Problem	Severity
	U1	U2	U3	U4	U5		
P1	X	X	X	X	X	Size of the system when placed on a table feels a little big.	Medium
P2	X	X	X	X		Box is very plain, not fun to look at.	Medium
P3	X		X	X		Edges are perhaps a little too sharp. A child might fall and hurt themselves on it.	Low
P4	X	X	X	X	X	Difficult to collect the blocks after they are put through the holes.	Medium
P5	X		X		X	Display is perhaps a little too small, trouble with pressing the smaller buttons.	Medium
P6	X	X	X	X	X	Sensor does not register the block fast enough as it is inserted into a hole.	Critical
P7	X	X	X	X	X	No logical way of progressing after having completed level 1.	Critical
P8	X	X	X	X	X	Missing elements for it to truly feel like a game.	Critical
P9		X	X		X	Reset buttons does not function properly.	Medium
P10			X		X	Colors on the screen can be hard to see if the display is looked at from certain angles.	Medium
P11			X			Alternative home button does not work properly.	Low
P12			X		X	Shape of triangular prism on screen is a little confusing and mismatched	Low
P13			X	X	X	Triangular prism is duller as it does not have color.	Low
P14			X	X		When restarting level 3, not intuitive what the order is now.	Medium
P15	X		X	X	X	Missing feedback and visibility of timer in level 3.	Critical
P16				X		Children may not have the same associations with symbols as we do.	Low
P17	X			X		The game can be exited by accident.	Medium

Table 5.2.: Suggested solutions to the identified problems from the evaluations.

ID	Solution	Priority
S1	Could be made more compact, for example by placing the holes closer to each other making the board smaller, and thus also the box.	12
S2	Adding color would bring more life to it, and make it more fun.	11
S3	As the box is made of wood it is easy to sand the edges down.	16
S4	A slide of some sort inside the box can be made to lead the blocks to the front of the extraction hole. Similar mechanism to that of a pool table.	9
S5	Either a new, larger display can be used, or buttons can be removed from the display altogether. By removing them they can instead be implemented as physical buttons on the box.	5
S6	Error in code logic that needs to be looked further into.	1
S7	Implementing better feedback when a level is finished, for example with a "Next level" button in the interface	2
S8	Implementing sound as feedback, maybe animations, point systems, or achievements.	4
S9	Error in code that needs to be fixed.	6
S10	Using brighter colors and having bigger contrast between the background and grey colors on board.	10
S11	Error in code that needs to be fixed.	13
S12	Have the shape reflect the actual block better by printing a new block or changing the shape on the display.	15
S13	Add a bright color to a new block, or paint the existing block.	14
S14	Better feedback regarding what state the system is in.	7
S15	Make the timer more visible with larger and bold writing. Perhaps add a ticking sound, and color change to the timer when it is close to running out.	3
S16	Have an adult help them in the beginning, helping them learn what the different symbols mean.	17
S17	Create more constraints to the window, making sure no kids can accidentally exit the game.	8

As seen from Table 5.1, 17 different problems, all reflecting the research question, being either about how the system feels in use [P4, P6, P7, P9, P10, P11, P12, P14, P15], or how it could improve for the intended user group [P1, P2, P3, P5, P8, P13, P16, P17], are defined. The majority of the problems are further categorized with a medium severity, covering 47.1% of the total problems. With only 23.5% and 29.% considered critical and low respectively.
As seen from the two tables, the problems with critical severity in Table 5.1 are prioritized first in Table 5.2, then the medium, and lastly the problems with low severity. Each of the solutions is then prioritized from highest to lowest within the severity categories. P6 is prioritized first and P8 last in the critical severity category, P5 first and P1 lowest in the medium category, and lastly, P11 is highest while P16 is at the bottom of the list in the low category.

5.4.1. Highest and lowest prioritized problems, P6 and P16

The highest and lowest prioritized problems are P6 and P16 respectively. As the system relies on sensors for giving feedback to the user, a sensor not working properly [P6] is a huge problem for the system. As explained in subsection 4.3.3 this problem was already known to some extent before testing, but its severity was first seen during these evaluations. The problem was discovered quickly with the first participant, and to avoid the issue further, this person and the other participants were asked to hold the blocks over the sensor a little longer before letting go. As adults, they have the patience to do this to test the entire system, but smaller children may not behave similarly and this is therefore an issue that must be fixed before it can be further tested.
On the other end of the scale, we find P16, which was identified by only one participant, U4, in connection with P7. As all of the participants struggled with further progression after level 1, they were told they had to go back and select the next level. That made everyone intuitively press the house symbol as they all associated it with going back to the main menu, in this case, the level select page. It was at this point U4 commented on how a child might not associate a house with the same thing we adults do. This problem is prioritized last as it could be solved by fixing P7, which is seen as more severe, or by having an adult help the children initially with the system to learn to associate the house with choosing new levels.

5.4.2. Problems affecting the box, P1 and P4

As seen from Table 5.1 P1 is a problem concerning one of the main requirements for the system, that the box should not be too big, and was commented on by all of the participants. The severity based on this alone should therefore be critical

by the reasoning made earlier.
An important point here is that in the evaluations all of the participants sat on a chair with the system on a table in front of them, and not on the ground which is preferred, as reasoned for in section 4.2. When this was explained to the participants, almost all of them ended up re-evaluating the size and made a new statement that it felt more right. Two of the participants, however, still commented that it could be made a little more compact, where U5 specifically suggested S1 in Table 5.2. Despite this, P1 was lowered to medium severity, as the majority of the participants retracted their original statements. In addition, S1 got the lowest priority in the medium category as it does not affect the overall use of the system.

During the evaluations it was discovered that the extraction hole was too small, resulting in P4 [Table 5.1]. This was however not surprising as the hole is made for smaller hands. Because of this, the front panel was simply removed for the other participants after U1. Despite this, all of the participants still struggled a bit with collecting the blocks as they did not end up in front of the box by themselves. This forced the participants to bend down to look and reach underneath the box to collect them. This is not an ideal solution, and S4 is, therefore, favorable to implement before further testing.

5.4.3. Design and aesthetic related problems, P2, P3, P10, P13 and P12

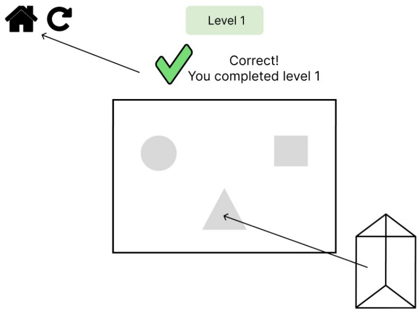
As these problems do not affect the use of the system, they are prioritized lower, as seen in Table 5.2. Despite of this P2 is still prioritized over P1. This is based on the possible effect P2 can have on the desirability of the system [72], as it can affect the user experience. In addition, the majority of the participants commented on its appeal and how it felt very plain, the opposite of what the system aims to be, which is fun. By implementing the simple solution of S2 you could give the users an entirely different experience, and hopefully also make the system seem more fun in its use.

P10 on the other hand was only discovered by two of the participants, but is also an important problem to consider as it affects the user experience as well, with the user having to face the screen at a certain angle to see the colors. This can, however, be a result of the chosen display, and could be solved with S5, by swapping out the screen. S10 would however also solve this problem. Since brighter colors appeal more to the still-developing eyes of children [75], it can be a good idea to make all the colors in the system brighter and have more contrast. An idea presented by U5 is to implement more contrast between the board and the grey colors on the display, the board on the screen could be made to match the color it has in real life for instance. This would also make it even more clear that the users have to put the blocks through actual holes in the box.

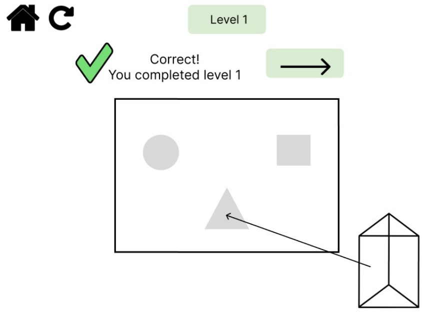
P13 is also considered an aesthetic issue but does not impact the system or the user experience as much as P2 and P10. This problem is therefore given a low severity and a priority of 14. It is nevertheless higher prioritized than P3, which is a safety issue. U1 commented on the edges [Table 5.1], but they also concluded that it was not that sharp and should not be an issue, especially since it can be sanded down very easily. S13 is, therefore, higher in priority as this solution requires a little more work, with finding color and painting or printing a new block, and should be started on before P3.
P12 is also prioritized higher than P3 because of this, but also very low as it is a problem regarding aesthetics. During the evaluations, U3 and U5 commented on the shape of the triangular prism on the display, saying it was not as accurate as the other shapes, but good enough for them to understand which block was meant. The black line on the figure [seen in Figure 4.28] was especially commented on as confusing. The shape on the display should therefore be changed to better reflect the actual block to create a more accurate image of the system.

5.4.4. Feedback issues, P7, P8, P14 and P15

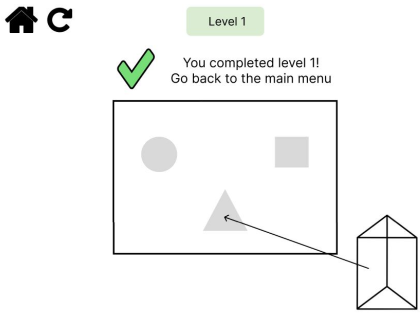
All of these problems have solutions regarding feedback as seen from Table 5.2, and the problems can therefore be seen to affect functionality. Because of this, each problem is marked with high severity, critical or medium, and their corresponding solutions with high priorities. During the evaluations, there was only one occasion each participant required help from the observer to progress. This was when level 1 was completed, as a natural progression was not intuitive for the participants. U4 and U3 however, managed to eventually reason that the house symbol was the most intuitive button to press, but both commented that a "Next level" button could be useful. The fact that adults struggled with this indicates that it would be very difficult for a child to figure out the same, as their cognitive skills are not fully developed [99]. S7 is therefore considered to be priority number two in Table 5.2, as this solution needs to be implemented for the users to naturally progress through the levels on their own. Luckily the solution is simple; implement more feedback. Three different design suggestions can be seen in Figure 5.1.



(a) With an arrow.



(b) With a button.



(c) More explanatory text.

Figure 5.1.: Feedback suggestions to when a level is completed, with the current design on level 1 as an example.

As the design of the levels currently have a "Correct!" message pop up if you do something right, this can easily be changed to have more informative text, as illustrated in Figure 5.1c. By giving a very clear message of what the next move is, it should be intuitive for the users what to do. However, the downside to this solution is that it takes up a lot of space, and requires the user to be able to read properly, which should not be taken for granted with smaller children. In that case, they would have to use the system with an adult to understand what to do. The two other suggestions, therefore, illustrate possible solutions that do not take up too much space and should be more intuitive, even for smaller children, with using figures and color instead of text. Figure 5.1a illustrates that an arrow could pop up when the user completes the level, indicating to the user where to press. The last solution Figure 5.1b shows S7, which was suggested by some of the participants. This way you can still choose to press the home button and go back to the level select page, or press the arrow button, jumping straight to level 2.

Another problem [P8] all of the participants commented on was the lack of audio-visual feedback for the system to be more fun, and feel more like a game. U4 and U2 also commented that other game mechanics could be included, such as points or achievements, but this was more suggestions than something they felt was missing. The participants specifically commented on sound as an addition to the "Correct" and "Try again" labels, where some suggested sound on both, and others only sound on correct moves. As the best way of teaching someone a new skill is with positive reinforcement [81], only adding a sound when something is done correctly can enhance the positive experience of the game, and hopefully as a result, make it more rewarding and fun for the users. As clapping is seen as a sign of approval [80], this sound could for instance be used when something is done correctly.

As with P6, problems regarding the timer added on level 3 were also known to some extent before the evaluation [subsection 4.3.3]. The participants then highlighted the concern with the timer during the evaluations as the majority of the participants commented on it and its lack of feedback [P15]. Some of the participants did not even see the timer to begin with. As level 3 is heavily based on this timer, not being able to register the timer is seen as problematic. This makes it a problem with high severity and needs to be fixed for the users to fully experience level 3. The problem has however gotten a lower priority than P7. This is because P15 does not hinder the participant in their tasks the same way P7 did, making it a less critical problem to be solved.

The last problem requiring better feedback is P14. This is a harder problem to categorize than the others regarding feedback, as the point of level 3 is to be more challenging in its logical thinking. As discussed in subsection 4.3.1, the design and progress of level 3 is made intentionally for the user to have to use more logical skills. And a timer is used to further stress the user forcing them to think fast. From the evaluations, both the timer and the design of the level were commented on by what their intentions were [subsection 4.3.1] as a good thing. Only the feedback of the systems state came up as a problem.
When U3 and U4 tried level 3 they wondered what would happen if they put the wrong block into the board midway through the game. As nothing happened on the display with the block or board when they did, they both questioned the order of what block to put in next. They both commented out loud that they assumed they would start from the beginning, and by trying a new block they quickly understood that this was the way. However, they both still commented that this was hard to intuitively know, even though U3 had made a mistake in level 2, seeing what the intended order was [subsection 4.3.1]. Because only two participants discovered this problem, making the results not representative, it was given a medium and not critical severity. The solution still got a rather high priority, as two adults did not find the order intuitive, implying that children might find it even harder. S14 should therefore be looked at, and perhaps also

tested with the current level to better conclude the level's difficulty.

5.4.5. Problems with the display, P5 and P17

Even though P17 was discovered by two participants, it has gotten a medium severity, and a priority of 8 because of its effects on the system. Since the display in this prototype is a touchscreen, a child who likes to press buttons may be able to exit the game in its entirety, by accident. This would mean the system, in its current state, would have to have an adult present to watch the child and possibly restart the game when it is exited. This is not favorable, and solutions to this should be implemented before further testing. This can for instance be done as S17 suggests, with implementing constraints to the game making it impossible to exit the game. An example could be to implement an if-statement in the code, where the game can not be exited before all of the levels are completed.

P5 is a different problem but was identified by three of the participants, implying its severity. Therefore, its solution is favorable to look at in order to solve P5, but it can also be used to solve P17. By using a different screen without touch, you could remove the need for buttons on the display. Instead, they could be implemented as physical buttons on the box. Since they can be made to accommodate for lack of fine motor skill with their size, they can be easier to press, and as a result, make the system easier to use. A button to exit or turn off the game can then be implemented with a fail-safe. For instance, a home button equivalent to that in the current design can turn off the system if it is pressed a certain way for a certain amount of time, making it more difficult for a child to accidentally exit the game.
This solution does however require a lot of extra work, as changing the display will affect other areas of the system.

5.4.6. Issues in code, P9 and P11

As seen from Table 5.1, both P9 and P11 are regarding functionality, but one is categorized as medium, and the other low. This is because neither are large problems, but can be fixed if needed. During the evaluations, the majority of the participants observed the reset button and tried to press it at different times, with the hope that the game would restart. In each level, it worked when it was pressed in the middle of a level that had not been completed yet. It also worked in the middle of a level if it was completed, and then played again, but only in levels 2 and 3, not 1. In addition to this, the participants observed that the reset button did not work immediately after a level had been completed. This implies there is an error in the code that needs to be fixed.
P11 was discovered similarly with U3 commenting on it that *"It looks like the*

other buttons, so maybe this can be pressed as well". As the button intends to work as an alternative home menu, with nothing happening when it was pressed, also implies an error in the code. P9 is however considered more critical than P11, as a reset button is more likely to be pressed and used with intention than an alternative home menu, implied by the number of participants who discovered the two problems.

5.5. Summary of the evaluations

With each problem that occurred during testing listed in Table 5.1, along with their respective solutions in Table 5.2, the meaning is not to implement all of these solutions at once. They are rather meant as suggestions for specific aspects that should be further looked into, and how the system could be tested with other solutions. For example, P1 implies a change of the entire box, requiring a great deal of work, as many other aspects of the system then also have to be changed. It can, however, be done as simple as making a cardboard model in a new size and comparing it to a smaller child, only testing this aspect and not the entire system. As the research question states, the evaluations aim to answer how it *could* improve for further testing, not necessarily how it should.

In addition to the improvements, the evaluations also aimed to answer how the system felt in use, to assess how other people interacted with it, and to see if it could have potential with the intended user group. Each participant was therefore asked what they liked about the experience, and the system in general [Appendix A] after the evaluation. Even though they discovered a lot of problems and saw the need for improvements, they all expressed how they liked the system in that it was intuitive, very responsive, and that it seemed like a good concept for further development.
U1 however, after having tried the system expressed their concern about how 1-4-year-old children could be able to figure everything out by themselves. However, they also said that the system was exciting, with a lot of potential for the intended system. U1, therefore, suggested that the system should either be adapted better to older children or have this system work as an educational toy in interaction with an adult rather than a toy, to maximize its potential.

Although the participants were not representative of the user group, or proper experts were used, the evaluations can be considered successful in what they aimed to achieve. Feedback on the initial prototype was given, and many aspects that were not considered during prototyping were discovered, helping the prototype for further development at a later stage. In addition, the author received valuable feedback on how the system was to use, confirming that the prototype was on the right intended path.

5.6. Improvements made to the system

With the feedback from the evaluations, the GUI was looked further into to see if some kinks could be worked out.

As mentioned in subsection 4.3.3, and section 5.4, some problems in the system were already known before the evaluations.
The issue with the feedback messages, discussed in subsection 4.3.3, in that they did not disappear after a few seconds was first looked at. After a while, it was discovered that the mistake was simply a forgotten AND in one of the elif statements in both levels 1 and 2. The AND statement was included in the TUI version in levels 1 and 2, as can be seen in lines 27 and 67 in Listing C.3, but was forgotten in the GUI code. When the AND statement was added in level 1, and a similar statement in level 2, along with code to remove the feedback message after an incorrect move in each level [Example of the Level1 class can be seen in Listing C.4], the game worked as intended.

Since P6 was given the highest severity, and priority from the results tables [Table 5.1 and 5.2], this was also looked into. As it turned out, this was fixed quite easily by changing the Arduino code in Listing C.2. The change made was to add the If statement seen in Listing 5.1 before the sensor values are transmitted to the Raspberry Pi [after line 21 in Listing C.2]. This statement ensures that no data is sent unless one of the sensors is blocked. This solved the problem of the user having to hold in front of the sensor for some time before the feedback message showed up. Now, the feedback is instantaneous if the sensor is registered as blocked. The added delay of 1500 [subsection 4.3.3] was however kept, to give the user some time to insert the block into the hole, before it sends more sensor values to the raspberry pi.

```
if (SensorState1 == 0 || SensorState2 == 0 || SensorState3 == 0) {
```

Listing 5.1: If statement added to Listing C.2.

P9 and the reset button were also looked into after the evaluations. The problem regarding not being able to restart level 1 mid-game if the level previously had been completed, was easily fixed. This was done by adding functionality to the specific reset buttons connected to each completed level, that had initially been forgotten. The added line of code in level 1 can be seen in line 28 in Listing C.4, where a similar line was added in the two other levels as well. The issue with not being able to restart the game immediately after it had been completed however was more complicated, but a solution was eventually discovered. When the reset button was pressed, the already implemented logic was to show the first step in the level, for level 3 nothing changed visually, and set the level variable to 0, jumping back in the main logic function [See Listing C.4 for the variables in level1. They are used similarly in the other two levels]. The problem with this was that the thread closes when a level is completed, making it impossible for the program to jump back in the loop with the sensors and give feedback for moves. For the reset button then to work when a level is completed, the thread also needs to be started again when the button is pressed in addition to setting the variable and showing the first page. An If statement was therefore made in the Reset function [Can be seen in lines 32-43 in Listing C.4] for each level, where the thread was started if the level had been completed.[1]

In the evaluations, the alternative home button did not work when it was pressed and was therefore assumed to be faulty altogether. However, it was discovered after the evaluations when fixing other code, that the button function did work, but only if it was pressed after a level was completed. When this was further looked into, it was discovered that the "Level X" info box did not have button functionality in Qt Designer but was rather only a label when a level had not been completed. This was then fixed, by making the box in Qt designer a button, and by adding its functionality with one line of code to each level [Seen in line 24 in Listing C.4 for level 1, a similar line was added to the other levels]. After that was done, the alternative home button worked just as intended on each level.

Another problem, not discovered before the evaluations, nor during, was discovered afterward when the other issues were looked into. As explained in subsection 4.3.3, feedback regarding the color of the "Level X" box on top of each level was implemented. In addition, functionality to keep track of the levels that had been completed was implemented. Or so was thought. As it turned out, an issue with this was found. The original intention of the levels was to be able to play green levels as many times as the user would want. This also implies that the user is able to replay level 1 after it is completed, before level 2 or 3 is done for instance. The issue found can be illustrated in Figure 5.2.

[1] A level is considered complete when the Correct variable is set to 1, if not the variable is 0.

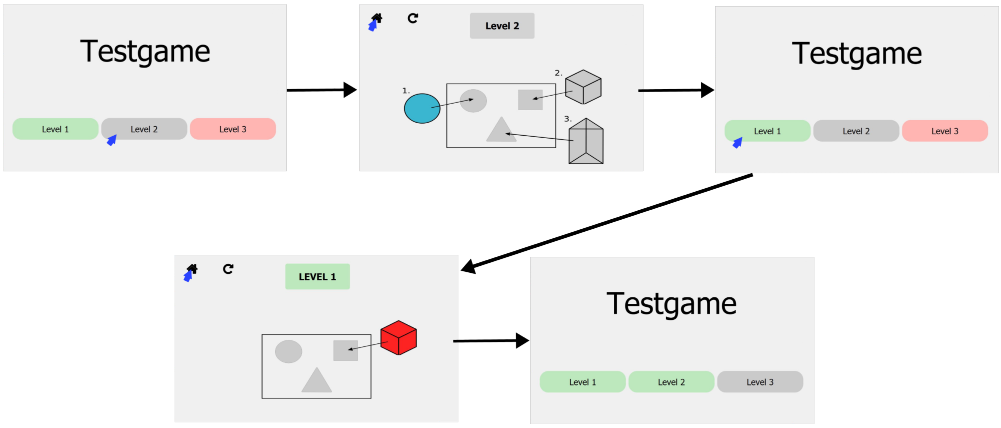


Figure 5.2.: The process showing how the error was discovered. The blue mouse indicate what is pressed. If the buttons are pressed in this order, without completing a level, the wrong level select page is illustrated at the end.

Figure 5.2 indicates that if you enter the next playable level, then go back to a completed level, when you then press the home button you are sent to the wrong level select page. This was then tried to fix, but due to time constraints, a solution was not found. A possible solution to avoid this problem however could be implemented. For example by not allowing the user to replay the different levels before each of them are complete. This would however limit some of the potential the system has, with the user not being able to practice a level multiple times before accessing the next one.

As mentioned with the last problem, due to time constraints further issues were not looked into after testing. However, the hope is that each problem has been thoroughly discussed, and different possible solutions have been mentioned in this chapter to better guide future work with the system.

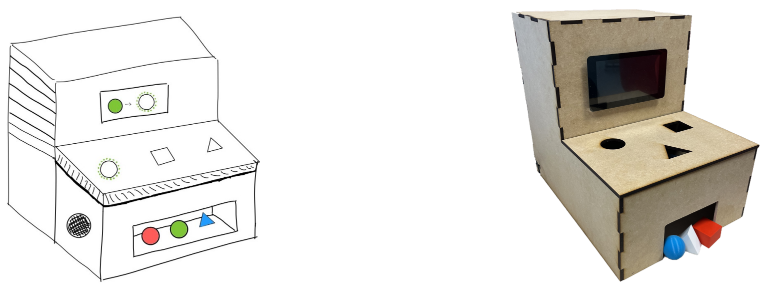
Chapter 6.

Discussion and conclusion

As the various steps in the prototype process have been discussed and justified, this chapter provides an overall reflection on the work done. In addition, a discussion about potential future work is included before finishing the thesis with a conclusion of the work done.

6.1. Discussion

In this thesis, a working prototype of a conceptual system has been developed. From the start of this process, the goal was to bring the drawings and discussion from the preliminary study [49] to life. Figure 6.1a shows the drawing presented as the most suitable for further development in the prestudy. Figure 6.1b shows how the prototype turned out with the work done in this thesis.



(a) Sketch from prestudy.

(b) Prototype made for this thesis.

Figure 6.1.: How the system was intended to look compared to how the prototype turned out.

Aesthetically they resemble one another very much. The main visual difference between these two figures is the backside of the box, where Figure 6.1a has additional stripes across the backside. One feature discussed in the prestudy was the possibility of developing multiple different boards to make the system more versatile. To then accommodate the requirements of not making the system overly complex and large, these boards were proposed to be stored inside the box itself. This is indicated by the lines on the backside of the drawing in Figure 6.1a. But in order to limit the workload, as explained in section 1.2, creating multiple boards were not the priority as only one board was needed to showcase the system and its potential. The box is, however, as seen in Figure 6.1b, made large enough in the back for possible storage if this were to be looked into at a later stage.
The other noticeable difference between the drawing and prototype is the speakers seen in Figure 6.1a. Sound was discussed throughout the prestudy as an important feature to implement in prototyping. However, since the other elements of the prototyping process took longer than anticipated, this was not prioritized as the system still worked without sound. It was, however, made clear by the evaluations that sound is indeed a significant feature that should be implemented in a future iteration of the prototype. Nonetheless, this would not be an issue to implement with the current prototype as the Raspberry Pi has an audio port [Table 3.2] that is just waiting to be used.

The prestudy [49] also highlights the potential for creating a system to benefit a wide range of users. However, the main way of doing this is explained by using multiple boards. With only having one board, also making the system versatile and appealing to a general mass would be a time-consuming task as more aspects of the game would have to be considered. Therefore, to simplify the prototyping process and make sure a working prototype was eventually made, children were chosen as the main intended user group for the system, as explained in section 1.2. The specific functionality of the system was therefore adapted to this user group during the work. For creating the blocks the group was further narrowed down to children aged 1-4 as explained in subsection 4.1.1. Because of this, the participants during the evaluations were told that the intended user group was children, and more specifically those aged 1-4. In addition, they were told that it was meant mostly as an independent toy for the children. Because of this U1 suggested how the system could either be adapted to older children or rather work as an educational toy in interaction with an adult [section 5.5]. As the intended purpose of the system was that children were able to use it without help, the last suggestion contradicts this. However, children aged 1-4 would require help with most systems as they are very early in development and their skills are limited. It could therefore be more beneficial, and simpler, to rather have older children as the main intended user group for the system to work as a toy. The same system could then also be used for smaller children with an adult for educational purposes, as suggested by

U1, to maximize its use.

When it comes to the creation of the system certain aspects need to be further discussed. As the prestudy mainly considers designs, and the purpose of the system, the specific functionality was created from scratch during this work. This made the process more time-consuming than first anticipated as 3D printing, PySide, and the Raspberry Pi had to be learned. All of these aspects were however chosen as they were considered beginner friendly and many tutorials and guides exist online. Consequently, the chosen hardware and software might not have been the best fit for this system, and could possibly have been looked better into. However, the time available for this thesis was limited, so picking a familiar language and hardware was also intentional for the process to go faster and smoother.
Even though PySide and the UART protocol had their disagreements throughout the development, a functioning game was created and tested at the end. The main reason for many of the problems with PySide, and why this took longer than anticipated was however most likely due to lousy debugging in Thonny. Is there one thing that has been learned from this experience, it is to use a better IDE when developing more complex code. The idea of downloading a better IDE was in addition first thought of when the code was half written, making downloading another seem more of a hassle than what it was worth.

When it comes to the design of the game two different functionalities were left out intentionally in the creation of the GUI. The first is a way to go back to the main menu. This was not done because only one board was used for this prototype. As explained in subsection 4.3.1, the purpose of the main menu is to work as a unifying element of the game with multiple boards. With only one board, the need to go back to the main menu is therefore eliminated. The other feature missing is a way to quit the application or shut it down in the GUI. The way to close the application as it is now is to close the window exiting the GUI and then shut down the Raspberry Pi. Adding functionality for this in the GUI was deemed unnecessary for the objectives of the thesis work and the evaluations. This feature is more relevant to the later stages of prototyping, or if starting and closing the system is specifically tested later.

6.2. Future work

Based on the discussions throughout this thesis there are two main ways for the system to be further developed. Because the objectives for this thesis are based on limitations from the preliminary study, the first way is to further develop the system according to these requirements. The main focus would intuitively then be making multiple boards and adapting the system accordingly. In addition, as the system is made very general, by making it adaptable to different boards only the imagination sets limits for their design. Having blocks and holes is not the only thing that could exercise fine motor skills in a fun way. For instance, a version of the lacing cards, or peg toy [Figure 2.1 and 2.2] could be implemented, or something else entirely.

The second way to further enhance the system is to examine the work accomplished in this thesis, addressing any problems that have arisen during its development, and then making necessary improvements.
For instance, a way of attaching the board to the system should be looked further into. For this prototype, the board was simply placed on the box, as explained in subsection 4.2.3. This was an adequate solution for this work, but could prove to be a problem if the system is further tested with children for instance as they could be more reckless in using it.
Another aspect to consider is the size of the holes on the board compared to the blocks. As mentioned in subsection 4.2.1, when the blocks and board were created two different options were considered for their sizes. As also mentioned in this subsection, neither of them was followed, but rather a mix of them was implemented. This was unfortunately due to a mistake, as multiple things were tried to be done at the same time and the different options were forgotten when the board was made. But despite this, no major problems regarding this were seen in the evaluations, but it is still considered a key aspect to further examine. It could especially prove useful if the system is further tested with the target user group.
In addition, as a result of the evaluations, another sensor and perhaps a different communication protocol could be looked into. Especially with the threads that were implemented as a consequence of using UART and polling, which resulted in many error messages throughout development. Other sensors that could work with the current prototype are listed in Table 3.3, where the color sensor may be the best second option. Other communication protocols that can be especially recommended are I2C and SPI. Because they are multi-master protocols, they can make the system more flexible, and in the future better suited to handle multiple boards and/or perhaps a different display.

The evaluations also contributed to finding potential future work for the prototype. Some of the problems were however solved, as explained in section 5.6, but there are many problems left, as seen in Table 5.1. A suggestion of how these problems should be prioritized is indicated by Table 5.2. The problems that were solved during this work did, however, not follow this exact order. This was simply because only code problems were prioritized at the end, due to limited time.

The main objective of this thesis is to create a prototype that can work as a system for exercising cognitive and muscular skills for those that could benefit from such training. The goal is to specifically target fine motor skills and logical thinking in a fun way by creating a mechatronic gamified system. The prototype is based on the work done in a preliminary study [49] done prior to this thesis. Whereas the prestudy merely presents a conceptual system, the goal of this thesis is to bring the drawings and discussions to life by looking at more specific solutions and developing a functional prototype as a proof of concept. This includes identifying hardware and software, designing a graphical user interface (GUI), creating blocks, and putting the different parts together to create a cohesive system. The system should also be made easy to use, where the goal is to create a system for anyone without the need for specialized knowledge, rare equipment, or a substantial amount of money.

6.3. Conclusion

The prototype, as seen in Figure 6.1b, and the discussions throughout this thesis show that a prototype working as a gamified mechatronic training system has been developed. A board with holes along with corresponding blocks were made to target fine motor skill, and levels were implemented in a display to challenge logical thinking. Different parts of the prototype were made separately before it was all put together to create a cohesive system working as a proof of concept from the preliminary study on which the prototype is based upon. Figure 6.1 show that the drawings from the preliminary study have been brought to life, with the same physical components present in both figures.

From the evaluations of the prototype, it was concluded that the system was intuitive and responsive, and the participants were excited about the system's potential for the intended user group. However, due to more work than first anticipated and troubleshooting along the way, there are still many aspects that could be further improved on the prototype. Both from the prestudy and the results obtained in the conducted evaluations. Nevertheless, ways of further improving the prototype have been included in this thesis to work as a guide and give suggestions in the hope of making further development go smoother.

All in all, the process of creating a prototype based on drawings and ideas in a prestudy, as well as the objectives for this thesis can be considered a success. A working system was made and tested, and encouraging feedback was received from the conducted evaluations. From the work done, it can be concluded that the intended system, and working prototype, show promising potential of working as a gamified training system for cognitive and muscular skills.

Bibliography

[1] Amazon.com. URL: https://www.amazon.com/Wii-Fit-Game-Balance-Board-Nintendo/dp/B000VJRU44 (visited on Nov. 10, 2022).

[2] From Melissa & Doug brand. Amazon.com. URL: https://www.amazon.com/Melissa-Doug-Lace-Trace-Activity/dp/B00007ISY4?ref_=ast_sto_dp&th=1&psc=1 (visited on Dec. 19, 2022).

[3] From Battat brand. Amazon.com. URL: https://www.amazon.com/Battat-Toddler-Stacking-Therapy-Pegboard/dp/B088NG7TF1?ref_=ast_sto_dp (visited on Dec. 19, 2022).

[4] From Zerodis brand. Amazon.com. URL: https://www.amazon.co.uk/Zerodis-Intellectual-Educational-Colorful-Geometry/dp/B07D5J9HQL (visited on Dec. 19, 2022).

[5] Syrebo.com. URL: https://www.syrebo.com/showroom/robotic-rehabilitation-gloves-for-hand-dysfunction-innovative-product-for-hand-function-.html (visited on Dec. 14, 2022).

[6] Tyromotion.com. URL: https://tyromotion.com/en/products/amadeo/ (visited on Dec. 14, 2022).

[7] NintendoLife. URL: https://www.nintendolife.com/features/talking-point-new-super-mario-bros-is-15-years-old-and-its-still-fantastic (visited on Nov. 10, 2022).

[8] Cube made by Bravo JP. Triangular prism made by Wahhab A. URL: https://thenounproject.com/ (visited on Apr. 5, 2023).

[9] *3D Printer Test 2023*. TekGuide.no. URL: https://www.tekguide.no/3d-printer/ (visited on June 27, 2023).

[10] *3D Printing as a Production Technology*. 3DPrinting.com, Dec. 2019. URL: https://3dprinting.com/3d-printing-use-cases/3d-printing-as-a-production-technology/ (visited on June 26, 2023).

[11] Dominguez A, Saenz-de-Navarrete J, De-Marcos L, Fernández-Sanz L, Pagés C, and Martinez-Herráiz JJ. "Gamifying learning experiences: Practical implications and outcomes". In: *Computers & education* 63 (2013), pp. 380–392.

[12] Ohlheiser A. *Atari nearly introduced the world to fitness gaming 30 years ago.* The Washington post, 2014. URL: https://www.washingtonpost.com/news/to-your-health/wp/2014/12/22/puffer-the-great-atari-exercise-bike-that-never-was/ (visited on Nov. 10, 2022).

[13] *About Arduino.* Arduino, Sept. 2021. URL: https://www.arduino.cc/en/about (visited on June 28, 2023).

[14] *About us.* Raspberry Pi. URL: https://www.raspberrypi.org/about/ (visited on June 28, 2023).

[15] *Applications for Python.* Python. URL: https://www.python.org/about/apps/ (visited on July 4, 2023).

[16] *Arduino Based Color Detector.* ElectronicsHub, Apr. 2017. URL: https://www.electronicshub.org/arduino-based-color-detector/ (visited on July 4, 2023).

[17] *Arduino Uno Rev3.* Arduino. URL: https://store.arduino.cc/products/arduino-uno-rev3?_gl=1*11up3lo*_ga*NDQwODAwMTE3LjE2ODY3MjY4Mzg.*_ga_NEXN8H46L5*MTY4NzkONTY5NS44LjEuMTY4NzkONjc3NS4wLjAuMA (visited on June 28, 2023).

[18] Lutkevich B. *Embedded system.* TechTarget, Dec. 2020. URL: https://www.techtarget.com/iotagenda/definition/embedded-system (visited on Feb. 7, 2023).

[19] *Basics of Phototransistor.* ElectronicsHub, Mar. 2019. URL: https://www.electronicshub.org/basics-of-phototransistor/ (visited on July 4, 2023).

[20] Dichev C and Dicheva D. "Gamifying education: what is known, what is believed and what remains uncertain: a critical review". In: *International journal of educational technology in higher education* 14.1 (2017), pp. 1–36.

[21] Tardi C. *What Is Moore's Law and Is It Still True?* Investopedia, Mar. 2023. URL: https://www.investopedia.com/terms/m/mooreslaw.asp (visited on Apr. 20, 2023).

[22] Gómez-Carrasco CJ, Monteagudo-Fernández J, Moreno-Vera JR, and Sainz-Gómez M. “Effects of a gamification and flipped-classroom program for teachers in training on motivation and learning perception”. In: *Education Sciences* 9.4 (2019), p. 299.

[23] *CloseEvent*. Answer on stack overflow, describing how to override the close event. Stack Overflow. URL: https://stackoverflow.com/a/9249527 (visited on May 5, 2023).

[24] Amos D. *Python GUI Programming With Tkinter*. Real Python, Mar. 2022. URL: https://realpython.com/python-gui-tkinter/ (visited on July 4, 2023).

[25] Chakravorty D. *3D Printing Supports – The Ultimate Guide*. All3DP, Feb. 2023. URL: https://all3dp.com/1/3d-printing-support-structures/ (visited on June 27, 2023).

[26] Jost D. *What is an Ultrasonic Sensor?* Fierce Electronics, Oct. 2019. URL: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor (visited on July 4, 2023).

[27] *Dexterity tests*. Physiopedia. URL: https://www.physio-pedia.com/Dexterity_Tests (visited on Nov. 10, 2022).

[28] Alciatore DG. *Introduction to Mechatronics and Measurement Systems*. 5th ed. McGraw-Hill Education, May 2018.

[29] Graf DL, Pratt LV, Hester CN, and Short KR. *Playing active video games increases energy expenditure in children*. Vol. 124. 2. American Academy of Pediatrics, 2009, pp. 534–540.

[30] Odunlade E. *Selecting the Right Microcontroller for your Embedded Systems Solution*. Nerd For Tech, Dec. 2019. URL: https://medium.com/nerd-for-tech/selecting-the-right-microcontroller-for-your-embedded-systems-solution-ce4ef7ecb906 (visited on June 28, 2023).

[31] Renard E. *Raspberry Pi Arduino Serial Communication – Everything You Need To Know*. The Robotics-backend. URL: https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/ (visited on Feb. 20, 2023).

[32] *Easy Electronics 2 Pcs IR Proximity Sensor for line follower and Obstacle sensing Robots - IR Sensor*. Amazon.com. URL: https://www.amazon.in/Electronics-Proximity-follower-Obstacle-sensing/dp/B074DVNJDF (visited on July 4, 2023).

[33] *ePro Labs TTP223 Ttp223 Based Capacitive Touch Sensor Module for Arduino*. Amazon.com. URL: https://www.amazon.in/ePro-Labs-TTP223-Capacitive-Arduino/dp/B01F5B0L12 (visited on July 4, 2023).

[34] *Figma*. URL: https://www.figma.com/ui-design-tool/ (visited on Apr. 15, 2023).

[35] *Fusion 360*. Software. Autodesk, 2013. URL: https://www.autodesk.no/products/fusion-360/overview?term=1-YEAR&tab=subscription.

[36] *General Python FAQ*. Python. URL: https://docs.python.org/3/faq/general.html#what-is-python (visited on July 4, 2023).

[37] *Guide to 3D Printing Materials: Types, Applications, and Properties*. Formlabs, Apr. 2021. URL: https://formlabs.com/blog/3d-printing-materials/ (visited on June 26, 2023).

[38] Sharp H, Preece J, and Rogers Y. *Interaction Design: Beyond Human-Computer Interaction*. 4th ed. 1.6.3. John Wiley & Sons Inc, 2019.

[39] Sigmundsson H. *Disorders of motor development (clumsy child syndrome)*. Journal of neural transmission, Supplementum, (69), 2005. URL: https://pubmed.ncbi.nlm.nih.gov/16355603/ (visited on July 5, 2023).

[40] Arnold HJ. “Effects of performance feedback and extrinsic reward upon high intrinsic motivation”. In: *Organizational Behavior and Human Performance* 17.2 (1976), pp. 275–288.

[41] *How do you choose between SPI, I2C, and UART for embedded communication?* Linkedin. URL: https://www.linkedin.com/advice/0/how-do-you-choose-between-spi-i2c-uart-embedded (visited on July 3, 2023).

[42] *How to Improve Fine Motor Skills*. Special tomato. URL: https://www.specialtomato.com/improve-fine-motor-skills (visited on Dec. 19, 2022).

[43] International Organization for Standardization. ISO 9241-210:2019. *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*. International Standard. Geneva, Switzerland: ISO, 2019.

[44] *IR Sensor Working and Applications*. Robu.in, May 2020. URL: https://robu.in/ir-sensor-working/ (visited on July 4, 2023).

[45] Anderson J. *An Intro to Threading in Python*. Real Python, Mar. 2019. URL: https://realpython.com/intro-to-python-threading/ (visited on July 4, 2023).

[46] Juviler J. *What Is GUI? Graphical User Interfaces, Explained.* HubSpot, Aug. 2022. URL: https://blog.hubspot.com/website/what-is-gui (visited on July 4, 2023).

[47] Nielsen J. *How Many Test Users in a Usability Study?* Nielsen Norman Group, June 2012. URL: https://www.nngroup.com/articles/how-many-test-users/ (visited on July 6, 2023).

[48] Skaug JH, Husøy AI, Staaby T, and Nøsen O. *Spillpedagogikk : dataspill i undervisningen*. nob. 1. utgave. Bergen: Fagbokforlaget, 2020. ISBN: 978-82-450-3204-8.

[49] Fristad JM. *Mechatronic Playtoy for Improving Various Skills.* Unpublished prestudy. Trondheim: Norwegian university of science and technology, Dec. 2022.

[50] Roschelle JM, Pea RD, Hoadley CM, Gordin DN, and Means BM. "Changing how and what children learn in school with computer-based technologies". In: *The future of children* (2000), pp. 76–101.

[51] Cieślak K. *Feedback in games - how to design rewards and punishments?* Try evidence, June 2021. URL: https://tryevidence.com/blog/feedback-in-games-how-to-design-rewards-and-punishments/ (visited on June 21, 2023).

[52] Cieślak K. *Game difficulty level - how to choose it properly.* Try Evidence, Sept. 2021. URL: https://tryevidence.com/blog/game-difficulty-level-how-to-choose-it-properly/ (visited on June 15, 2023).

[53] Girgis k. *Communication protocols in embedded system.* Linkedin, Mar. 2023. URL: https://www.linkedin.com/pulse/communication-protocols-embedded-system-keroloes-girgis/ (visited on July 3, 2023).

[54] Kapp K. “Gamification: Separating fact from fiction”. In: *Chief Learning Officer* 13.3 (2014), pp. 45–52.

[55] Moran K. *Usability Testing 101*. Nielsen Norman Group, Dec. 2019. URL: https://www.nngroup.com/articles/usability-testing-101/ (visited on June 12, 2023).

[56] Havenstrøm KE. *Det digitale skiftet i klasserommet: – Aldri vært morsommere å være lærer*. Mar. 2022. URL: https://utdanningsforskning.no/artikler/2022/det-digitale-skiftet-i-klasserommet--aldri-vart-morsommere-a-vare-larer/ (visited on Dec. 17, 2022).

[57] Bar M and Neta M. *Humans Prefer Curved Visual Objects*. Vol. 17. 8, pp. 645–648.

[58] Fitzpatrick M. *PyQt5 vs PySide2*. Python GUIs, Mar. 2023. URL: https://www.pythonguis.com/faq/pyqt5-vs-pyside2/ (visited on July 4, 2023).

[59] Fitzpatrick M. *Qt Designer Manual*. qt.io. URL: https://doc.qt.io/qt-6/qtdesigner-manual.html (visited on July 4, 2023).

[60] Sailer M and Homner L. “The gamification of learning: A meta-analysis”. In: *Educational Psychology Review* 32.1 (2020), pp. 77–112.

[61] Ashby MF. *Materials Selection in Mechanical Design*. 3rd ed. Oxford: Butterworth-Heinemann, 2005, pp. 28–30, 46–78, 297.

[62] *Microcontroller vs Microprocessor - What are the Differences?* TotalPhase, Dec. 2019. URL: https://www.totalphase.com/blog/2019/12/microcontroller-vs-microprocessor-what-are-the-differences/ (visited on June 28, 2023).

[63] *Motor skills*. Workplace testing, May 30, 2020. URL: https://www.workplacetesting.com/definition/3712/motor-skills (visited on Nov. 10, 2022).

[64] Babich N. *Using Red and Green in UI Design*. Medium, Jan. 2019. URL: https://uxplanet.org/using-red-and-green-in-ui-design-66b39e13de91 (visited on June 16, 2023).

[65] Bayona NA, Bitensky J, Salter K, and Teasell R. “The role of task-specific training in rehabilitation therapies”. In: *Topics in stroke rehabilitation* 12.3 (2005), pp. 58–65.

[66] *Natural vs Synthetic Polymers*. Carnegie Mellon University. URL: https://www.cmu.edu/gelfand/lgc-educational-media/polymers/natural-synthetic-polymers/index.html (visited on June 8, 2023).

[67] Utekar O. *Microprocessor vs. Microcontroller: Key Differences*. Moonpreneur. URL: https://moonpreneur.com/tech-corner/microprocessor-vs-microcontroller/ (visited on June 28, 2023).

[68] *Original Prusa i3 MK3S+ 3D printer*. Prusa research. URL: https://www.prusa3d.com/product/original-prusa-i3-mk3s-3d-printer-3/ (visited on June 27, 2023).

[69] Buckley P and Doyle E. "Gamification and student motivation". In: *Interactive learning environments* 24.6 (2016), pp. 1162–1175.

[70] Fromaget P. *Is Raspberry Pi a Microcontroller or a Microprocessor?* RaspberryTips.com. URL: https://raspberrytips.com/is-raspberry-pi-a-microcontroller/ (visited on June 28, 2023).

[71] Kobek P. *How video games have helped us workout since the 1980s*. TheGamer, Mar. 2020. URL: https://www.thegamer.com/video-games-workout-exercise-history/ (visited on Nov. 10, 2022).

[72] Morville P. *User Experience Design*. June 2004. URL: https://semanticstudios.com/user_experience_design/ (visited on June 20, 2023).

[73] *pyserial*. Python. URL: https://pypi.org/project/pyserial/ (visited on July 6, 2023).

[74] *Python GUIs*. Tutorials for GUI programming. URL: https://www.pythonguis.com/ (visited on July 4, 2023).

[75] Pancare R. *How Do Bright Colors Appeal to Kids?* Apr. 2018. URL: https://sciencing.com/do-bright-colors-appeal-kids-5476948.html (visited on July 6, 2023).

[76] Teja R. *What is a Sensor? Different Types of Sensors and their Applications*. ElectronicsHub, Apr. 2021. URL: https://www.electronicshub.org/different-types-sensors/ (visited on July 4, 2023).

[77] *Raspberry Pi 4*. Raspberry Pi. URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (visited on June 28, 2023).

[78] *Raspberry Pi Arduino Serial Communication – Everything You Need To Know*. Arduino, June 2019.
URL: https://www.arduino.cc/reference/en/language/functions/communication/serial/ (visited on July 3, 2023).

[79] *Raspberry Pi Touch Display*. Raspberry Pi.
URL: https://www.raspberrypi.com/products/raspberry-pi-touch-display/ (visited on July 3, 2023).

[80] Riggio Ph.D RE. *Why We Clap: The Psychology of Applause.* Psychology Today, Jan. 2022.
URL: https://www.psychologytoday.com/intl/blog/cutting-edge-leadership/202201/why-we-clap-the-psychology-applause (visited on July 6, 2023).

[81] *Reinforcement and Punishment*. University of central Florida.
URL: https://pressbooks.online.ucf.edu/lumenpsychology/chapter/operant-conditioning/ (visited on July 6, 2023).

[82] *Small Parts for Toys and Children's Products Business Guidance.* CPSC.gov is an official website of the United States government. Consumer product safety commission.
URL: https://www.cpsc.gov/Business--Manufacturing/Business-Education/Business-Guidance/Small-Parts-for-Toys-and-Childrens-Products (visited on June 1, 2023).

[83] *Standard hand and wrist sizes for crochet patterns*. Dora Does. URL: https://doradoes.co.uk/crochet-resources/charts-technical-stuff/hand-and-wrist-sizes-for-crochet/ (visited on June 5, 2023).

[84] Mou SU. *What is Plywood? Types of Plywood.*
URL: https://civiltoday.com/civil-engineering-materials/timber/203-plywood (visited on June 8, 2023).

[85] *TCS3200 Color Sensor Module*. Addicore. URL: https://www.addicore.com/TCS3200-Color-Sensor-p/ad307.htm (visited on July 4, 2023).

[86] *TDT4180 - Menneske-maskin-interaksjon*. Course at NTNU.
URL: https://www.ntnu.no/studier/emner/TDT4180#tab=omEmnet (visited on July 7, 2023).

[87] *The 7 principles of universal design.* Centre for Excellence in Universal Design.
URL: https://universaldesign.ie/What-is-Universal-Design/The-7-Principles/ (visited on Dec. 15, 2022).

[88] Team Thomas. *The Evolution Of 3D Printing*. Thomas publishing company. URL: https://blog.thomasnet.com/evolution-of-3d-printing (visited on Dec. 15, 2022).

[89] *Thonny*. URL: https://thonny.org/ (visited on July 4, 2023).

[90] *time — Time access and conversions*. Python. URL: https://docs.python.org/3/library/time.html (visited on July 6, 2023).

[91] *Top selling title sales units*. Nintento, Sept. 30, 2022. URL: https://www.nintendo.co.jp/ir/en/finance/software/wii.html (visited on Nov. 10, 2022).

[92] *Touch sensors: What it is, How it works, Arduino Guide*. Seeedstudio, 2021. URL: https://www.seeedstudio.com/blog/2019/12/31/what-is-touch-sensor-and-how-to-use-it-with-arduino/ (visited on July 4, 2023).

[93] *Types of Color Blindness*. NIH, June 2023. URL: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness/types-color-blindness (visited on June 9, 2023).

[94] Fischer U, Suggate SP, and Stoeger H. "The implicit contribution of fine motor skills to mathematical insight in early childhood". In: *Frontiers in Psychology* 11 (2020), p. 1143. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01143/full (visited on Nov. 22, 2022).

[95] *UltiMaker S5*. UltiMaker. URL: https://ultimaker.com/3d-printers/s-series/ultimaker-s5/ (visited on June 27, 2023).

[96] Higuera V. *How to Help Your Child Develop Fine Motor Skills*. Healthline, Nov. 2018. URL: https://www.healthline.com/health/fine-motor-skills (visited on Jan. 24, 2023).

[97] *What is 3D Printing?* 3DPrinting.com, 2012. URL: https://3dprinting.com/what-is-3d-printing/ (visited on June 26, 2023).

[98] *What is Arduino?* Arduino, Feb. 2018. URL: https://www.arduino.cc/en/Guide/Introduction (visited on June 28, 2023).

[99] *What is cognitive development in early childhood*. upskillist. URL: https://blog.upskillist.com/what-is-cognitive-development-in-early-childhood/ (visited on June 6, 2023).

[100] *What is PLA? (Everything you need to know)*. TWI. URL: https://www.twi-global.com/technical-knowledge/faqs/what-is-pla#IsitEnvironmentallyFriendly (visited on June 8, 2023).

[101] Team Xometry. *Slicer in 3D Printing: Definition, Features, and How it Works*. Xometry, Sept. 2022. URL: https://www.xometry.com/resources/3d-printing/what-is-a-slicer-in-3d-printing/ (visited on June 27, 2023).

Appendix A.

Script for evaluating the prototype

1. Introduce the observer
2. Explaining the purpose of the test, how it will be conducted, and who the target user group is.
   - The goal is to obtain feedback about their experiences in using the system.
   - It is the system that is tested, not the participant.
   - They will not receive help during the evaluation as the goal is for them to figure out the system on their own.
   - The session is recorded in order to remember the details of the test, but is not to be used in any other way.
3. Explain the rights to the participant
   - They can abort the test at any time
   - The data collected are anonymous and are only used in order to improve the system.
4. Explain the think-aloud method.
5. Ask if the participant has any questions before the evaluation starts
6. Ask them some initial questions about design and aesthetics
   - What are your initial thoughts about the design and material choice for the system?

- Is there something you immediately wished was included in the system or do you feel something is missing design-wise?

7. Run the system and observe the participant.
8. Ask questions regarding the evaluation and functionality, usability, design, and aesthetics
    - Is there anything you see now that should be included design-wise?
    - What is your overall impression of the system?
    - Is there anything you feel is missing in functionality?
    - What did you like about it, and the experience?
    - Do you think it could be useful for those that struggle with fine motor skills, and how?
9. Ask if the participants have anything else they want to add regarding the system.

Appendix B.

Supplementary figures of the GUI

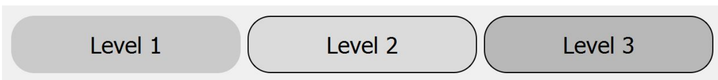


Figure B.1.: Illustrative example of how feedback is implemented into the button design. First shows the normal button, then when it is hovered over, and lastly when it is pushed. The same is done for the green buttons.

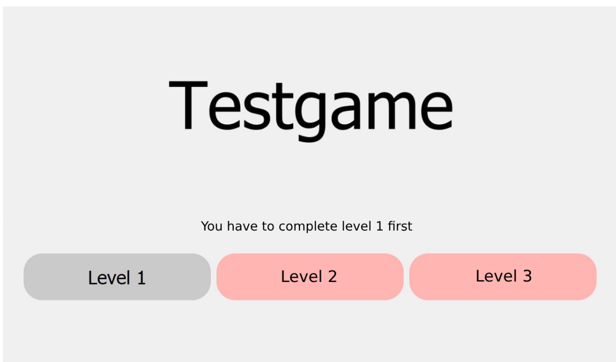
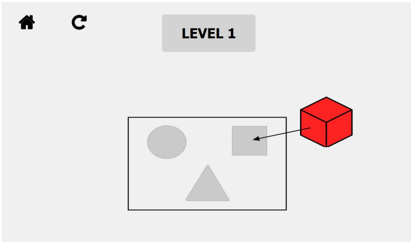
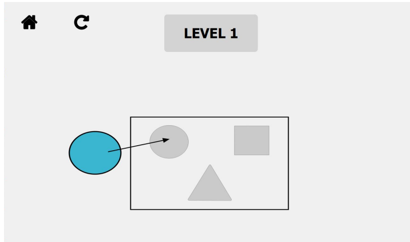


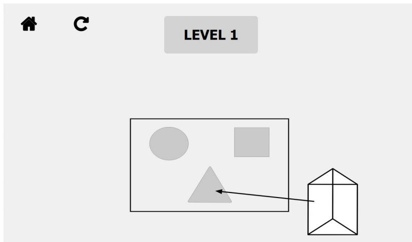
Figure B.2.: Feedback when a red button is pressed. This shows the feedback for when the red level 2 button is pressed. If the red level 3 button was pressed it would say level 2 instead of level 1 in the text.



(a) First step of level 1 in the GUI.

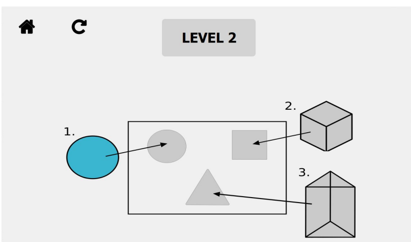


(b) Second step of level 1 in the GUI.

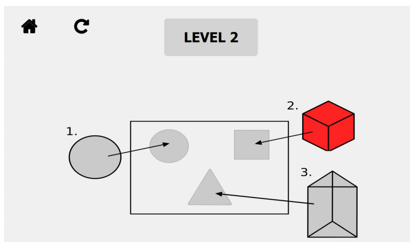


(c) Third step of level 1 in the GUI.

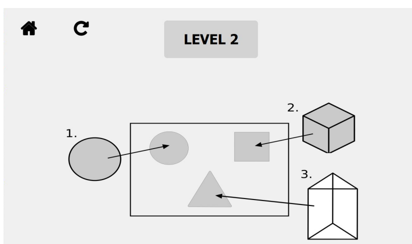
Figure B.3.: The design of level 1 in the GUI



(a) First step of level 2 in the GUI.



(b) Second step of level 2 in the GUI.

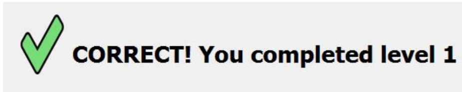


(c) Third step of level 2 in the GUI.

Figure B.4.: The design of level 2 in the GUI

(a) For correct move.

(b) For incorrect move.



(c) For a completed level, here illustrated with level 1.

Figure B.5.: Feedback messages in the GUI

Appendix C.

Code

```
#!/usr/bin/env python3
import serial
import time

ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1.0) #open serial communication with arduino (port,
      baudrate 115200)
time.sleep(3) #Seconds, enough time for the arduino to have the serial ready to communicate
ser.reset_input_buffer() #Data arrives in a buffer, resetting the input buffer
print("Serial ok")

try:
    while True:
        time.sleep(0.01) #Loop will be executed 100 times pr second
        if ser.in_waiting >0: #Checking if something is sent from the arduino
            line = ser.readline().decode('utf-8').strip('\n\r')
            sensordatasplit = line.split(',')
            ProxSensor1 = sensordatasplit[0]
            ProxSensor2 = sensordatasplit[1]
            ProxSensor3 = sensordatasplit[2]
            if ProxSensor2=="0":
                print('something is blocking sensor 2!')
            if ProxSensor1=="0":
                print('something is blocking sensor 1!')
            if ProxSensor3=="0":
                print('something is blocking sensor 3!')


except KeyboardInterrupt: #Ctr + C
    print("Serial communication closed")
    ser.close() #Closing the communication
```

Listing C.1: Testing the sensors, and communication between Arduino and Raspberry Pi.

```
#define ProxSensor1 2
#define ProxSensor2 4
#define ProxSensor3 8
//1 = not blocking, 0= blocking
int SensorState1 = 1;
int SensorState2 = 1;
int SensorState3 =1;

void setup() {
  Serial.begin(9600);
  while (!Serial){
  }
  pinMode(ProxSensor1, INPUT);
  pinMode(ProxSensor2, INPUT);
  pinMode(ProxSensor3, INPUT);
}

void loop() {
  SensorState1 = digitalRead(ProxSensor1);
```

```
  SensorState2 = digitalRead(ProxSensor2);
  SensorState3 = digitalRead(ProxSensor3);
  Serial.print(SensorState1);
  Serial.print(",");
  Serial.print(SensorState2);
  Serial.print(",");
  Serial.println(SensorState3);
  delay(500);
}
```

Listing C.2: Initial code for the Arduino

```
#!/usr/bin/env python3
import serial
import time

ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1.0) #open serial communication with arduino (port, baudrate 115200)
time.sleep(3) #Seconds, enough time for the arduino to have the serial ready to communicate
ser.reset_input_buffer() #Data arrives in a buffer, resetting the input buffer
print("Serial ok")

IsLevel1Finished = 0
IsLevel2Finished = 0

def Level1(Correct1,level1):
    while Correct1==0:
        time.sleep(0.01)
        if ser.in_waiting >0: #Checking if something is sent from the arduino
            line = ser.readline().decode('utf-8').strip('\n\r')
            sensordatasplit = line.split(',')
            ProxSensor1 = sensordatasplit[0]
            ProxSensor2 = sensordatasplit[1]
            ProxSensor3 = sensordatasplit[2]
            if ProxSensor1 =="0" and level1 == 0:
                print("Sensor 1 is blocked, Correct!")
                level1=1
            elif ProxSensor1=="0" and level1==1:
                print("Not correct, try again")
            elif (ProxSensor2=="0" or ProxSensor3=="0") and level1==0:
                print("Not correct, try again")

            if ProxSensor2 =="0" and level1==1:
                print("Sensor 2 is blocked, Correct")
                level1=2
            elif (ProxSensor1=="0" or ProxSensor2=="0") and level1==2:
                print("Not correct, try again")
            elif ProxSensor3=="0" and level1==1:
                print("Not correct, try again")

            if ProxSensor3 =="0" and level1==2:
                print("Sensor 3 is blocked, Correct")
                print()
                time.sleep(1)
                print("You completed level 1!")
                level1=3

            while level1==3:
                print("Want to do it again?")
                userInput = input("y or n:")
                if userInput=="y":
                    level1=0
                else:
                    Correct1=1
                    level1=0
                    return Correct1

def Level2(Correct2,level2):
    while Correct2==0:
        time.sleep(0.01)
        if ser.in_waiting >0: #Checking if something is sent from the arduino
            line = ser.readline().decode('utf-8').strip('\n\r')
            sensordatasplit = line.split(',')
            ProxSensor1 = sensordatasplit[0]
            ProxSensor2 = sensordatasplit[1]
            ProxSensor3 = sensordatasplit[2]
            if ProxSensor2 =="0" and level2 == 0:
                print("Sensor 2 is blocked, Correct!")
                level2=1
```

```
            elif (ProxSensor1=="0" or ProxSensor3=="0") and level2==0:
                print("Not correct, try again")
            elif ProxSensor2=="0" and level2==1:
                print("Not correct, try again")
                level2=0

            if ProxSensor1 =="0" and level2==1:
                print("Sensor 1 is blocked, Correct")
                level2=2
            elif (ProxSensor2=="0" or ProxSensor1=="0") and level2==2:
                print("Not correct, try again")
                level2=0
            elif ProxSensor3=="0" and level2==1:
                print("not correct, try again")
                level2=0

            if ProxSensor3=="0" and level2==2:
                print("Sensor 3 is blocked, correct")
                print()
                time.sleep(1)
                print("You completed level 2!")
                level2=3

            while level2==3:
                print("Want to do it again?")
                userInput = input("y or n:")
                if userInput=="y":
                    level2=0
                else:
                    Correct2=1
                    level2=0
                    return Correct2

def Level3(Correct3,level3):
    print("level3")

try:
    while True:
        time.sleep(0.01) #Loop will be executed 100 times pr second
        UserInput_level = input("level 1, 2 or 3? or quit (q):")
        if IsLevel1Finished ==0 and IsLevel2Finished ==0:
            if UserInput_level == "1":
                IsLevel1Finished = Level1(0,0)
            elif UserInput_level =="2" or UserInput_level =="3":
                print("You have to complete level 1 before doing the others")
            elif UserInput_level =="q":
                print("closed")
                ser.close()
                break
            else:
                print("invalid number")

        elif IsLevel1Finished ==1 and IsLevel2Finished ==0:
            if UserInput_level=="1":
                Level1(0,0)
            elif UserInput_level=="2":
                IsLevel2Finished = Level2(0,0)
            elif UserInput_level=="3":
                print("You have to complete level 2 before doing level 3")
            elif UserInput_level =="q":
                print("closed")
                ser.close()
                break
            else:
                print("Invalid number")

        elif IsLevel1Finished==1 and IsLevel2Finished==1:
            if UserInput_level=="1":
                Level1(0,0)
            elif UserInput_level=="2":
                Level2(0,0)
            elif UserInput_level=="3":
                Level3(0,0)
            elif UserInput_level =="q":
                print("closed")
                ser.close()
                break
            else:
                print("Invalid number")
```

```python
except KeyboardInterrupt: #Ctr + C
            print("Serial communication closed")
            ser.close() #Closing the communication
```

Listing C.3: Text-based user interface for level 1 and 2

```python
import time
import threading
from Level2Game import Level2Game

class Level1Game:
    def __init__(self, motorgame):

        #Sets the right widgets for the start of level 1
        motorgame.MainWindows.setCurrentWidget(motorgame.Level1)
        motorgame.level1Pages.setCurrentWidget(motorgame.page)
        motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)

        #Defining variables
        global level1
        global Correct1
        self.Correct1=0
        self.level1 = 0

        #Begins the thread
        self.begin(motorgame)

        #Buttons are defined
        motorgame.buttonHome.clicked.connect(lambda:self.Home(motorgame))
        motorgame.Level1NotCompleteHome.clicked.connect(lambda:self.Home(motorgame))
        motorgame.buttonHome_4.clicked.connect(lambda:self.Home1Complete(motorgame))
        motorgame.Level1CompleteHome.clicked.connect(lambda:self.Home1Complete(motorgame))
        motorgame.buttonReset.clicked.connect(lambda:self.Reset(motorgame))
        motorgame.buttonReset_4.clicked.connect(lambda:self.Reset(motorgame))


    #What happens if the restart buttons are pressed, the game is restarted
    def Reset(self,motorgame):
        if self.Correct1==0:
            motorgame.level1Pages.setCurrentWidget(motorgame.page)
            motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
            self.level1=0
        else:
            motorgame.level1Pages.setCurrentWidget(motorgame.page)
            motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
            time.sleep(1)
            self.Correct1=0
            self.level1=0
            self.begin(motorgame)

    #Changes to Level select page if the home menus are pressed
    def Home(self,motorgame):
        motorgame.MainWindows.setCurrentWidget(motorgame.Home)
        motorgame.levelButtonPages.setCurrentWidget(motorgame.Page_LevelSelect)
        self.Correct1=1

    #Changes to the correct Level select page if the home menus are pressed in a completed level
    def Home1Complete(self,motorgame):
        print(motorgame.level2Game)
        motorgame.MainWindows.setCurrentWidget(motorgame.Home)
        if motorgame.level2Game ==0:
            motorgame.levelButtonPages.setCurrentWidget(motorgame.Page_Level1Complete)
            self.Correct1 = 1
        elif motorgame.level3Game==0:
            motorgame.levelButtonPages.setCurrentWidget(motorgame.Page_Level2Complete)
            self.Correct1 = 1
        else:
            motorgame.levelButtonPages.setCurrentWidget(motorgame.Page_Level3Complete)
            self.Correct1=1

    #Function with the main logic of the game where the sensors are handled
    def func(self,motorgame):
        while self.Correct1==0:
            time.sleep(0.01)
            if motorgame.ser.in_waiting >0: #Checking if something is sent from the arduino
                line = motorgame.ser.readline().decode('utf-8').strip('\n\r')
                sensordatasplit = line.split(',')
                ProxSensor1 = sensordatasplit[0]
                ProxSensor2 = sensordatasplit[1]
```

```
                ProxSensor3 = sensordatasplit[2]

                if ProxSensor1=="0" and self.level1 ==0:
                    print("1 blocked")
                    motorgame.messageWidgets.setCurrentWidget(motorgame.CorrectWidget)
                    time.sleep(1)
                    motorgame.level1Pages.setCurrentWidget(motorgame.page_2)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
                    self.level1=1
                    time.sleep(1)
                elif ProxSensor1 == "0" and self.level1 == 1:
                    motorgame.messageWidgets.setCurrentWidget(motorgame.IncorrectWidget)
                    time.sleep(2)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
                elif (ProxSensor2=="0" or ProxSensor3 =="0") and self.level1==0:
                    motorgame.messageWidgets.setCurrentWidget(motorgame.IncorrectWidget)
                    time.sleep(2)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)

                if ProxSensor2 =="0" and self.level1==1:
                    print("2 Blocked")
                    motorgame.messageWidgets.setCurrentWidget(motorgame.CorrectWidget)
                    time.sleep(1)
                    motorgame.level1Pages.setCurrentWidget(motorgame.page_3)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
                    self.level1=2
                    time.sleep(1)
                elif (ProxSensor1=="0" or ProxSensor2=="0") and self.level1==2:
                    motorgame.messageWidgets.setCurrentWidget(motorgame.IncorrectWidget)
                    time.sleep(2)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)
                elif ProxSensor3=="0" and self.level1==1:
                    motorgame.messageWidgets.setCurrentWidget(motorgame.IncorrectWidget)
                    time.sleep(2)
                    motorgame.messageWidgets.setCurrentWidget(motorgame.blankWidget)

                if ProxSensor3 =="0" and self.level1==2:
                    print("3 Blocked")
                    motorgame.messageWidgets.setCurrentWidget(motorgame.CompletedWidget)
                    time.sleep(1)
                    self.level1=3
                    motorgame.level1Menu.setCurrentWidget(motorgame.menulevel1Complete)
                    self.Correct1=1

        motorgame.ser.reset_input_buffer()
        return

    #Starts the thread
    def begin(self,motorgame):
        t1 = threading.Thread(target=self.func, args=(motorgame,))
        t1.start()
```

Listing C.4: Level1 class

```
    #Close event, if the user exit the window
    #Based on the code from https://stackoverflow.com/a/9249527
    def closeEvent(self, event):
        can_exit= True
        if can_exit:
            print("Serial closed")
            self.ser.close()
        else:
            event.ignored()
```

Listing C.5: Code snippet from the game class showing the close event [23]

NTNU

Norwegian University of
Science and Technology