Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Spatial Mappings in the Home: Evaluating Targeting Techniques to Control Smart Home Devices

Master's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Sebastian Pettirsch

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Jürgen Steimle

Registration date: 02.12.2021
Submission date: 02.06.2022

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Statutory Declaration in Lieu of an Oath

Pettirsch, Sebastian

Name, Vorname/Last Name, First Name

399846

Matrikelnummer (freiwillige Angabe)
Matriculation No. (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

Spatial Mappings in the Home: Evaluating Targeting Techniques
to Control Smart Home Devices

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Aachen, 02.06.2022

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen

*Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

Para. 156 StGB (German Criminal Code): False Statutory Declarations

Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence

(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.

(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Aachen, 02.06.2022

Ort, Datum/City, Date

Unterschrift/Signature

Contents

List of Figures

List of Tables

Abstract

In today's homes, the number of controllable devices is rising rapidly. Almost all types of devices are becoming smart and controllable. Modern controls, like voice assistants, universal remotes, and smartphones are not optimized for controlling single targets in a room, making the interaction cumbersome. Therefore we propose to use a stationary and haptically explorable controller that offers the user to select individual devices in a room. Spatial mappings should be used to make the mapping of the controller—the relation between input and device—intuitive to understand.

In this thesis, we present the design, construction, and evaluation of five controllers for target selection in real-world environments. These use spatial mapping techniques of different abstraction levels. Three controllers, *Pillar Map*, *Zelda Map*, and *Cluster Map* follow an absolute approach, mapping the devices in the room to a map with buttons. The other two, *Direction Swiping Controller* and *Sun Controller* use a relative approach and map devices to user input based on the position of the target relative to the user. In a user study, we evaluated and compared the five controllers on their targeting time and accuracy. Participants selected targets in an exemplary living room using each of the controllers. The exemplary living room was designed to cover edge cases and common target patterns. The study was conducted across different positions in the room and the controllers were tested inside and outside the users' field of view. Additionally, we explored the effect the targets' positions had on the performance.

For interaction with vision, the results showed that controllers with lower abstraction levels, like the *Pillar Map* and *Cluster Map* perform better. For eyes-free interaction, the mapping should either have tactile elements that are easy to recognize, like the *Pillar Map*, or should work without visual and tactile cues, like the *Direction Swiping Controller*.

Überblick

In heutigen Haushalten steigt die Zahl der steuerbaren Geräte rapide an. Fast alle Arten von Geräten werden intelligent und steuerbar. Moderne Steuerungen wie Sprachassistenten, Universalfernbedienungen und Smartphones sind nicht für die Steuerung einzelner Ziele in einem Raum optimiert und machen die Interaktion umständlich. Daher schlagen wir vor stationäre und haptisch erkundbare Controller zu verwenden, mit denen Nutzer einzelne Geräte im Raum auswählen können. Räumliche Mapping-Techniken sollten verwendet werden um das Mapping des Controllers—also die Beziehung zwischen Eingabe und Gerät—intuitiv verständlich zu machen.

In dieser Arbeit stellen wir den Entwurf, die Konstruktion und die Evaluation von fünf Controllern für die Zielwahl in realen Umgebungen vor. Diese verwenden räumliche Mapping-Techniken auf verschiedenen Abstraktionsstufen. Drei Controller, *Pillar Map*, *Zelda Map* und *Cluster Map*, verfolgen einen absoluten Ansatz, indem sie die Geräte im Raum auf eine Karte mit Knöpfen abbilden. Die anderen beiden, *Direction Swiping Controller* und *Sun Controller*, verwenden einen relativen Ansatz und ordnen die Geräte den Benutzereingaben basierend auf der relativen Position zum Nutzer zu. In einer Nutzerstudie haben wir die fünf Controller hinsichtlich ihrer Zielerfassungszeit und -genauigkeit bewertet und verglichen. Die Teilnehmer nutzten jeden der Controller um Ziele in einem Beispiel-Wohnzimmer auszuwählen. Das Beispiel-Wohnzimmer wurde so gestaltet, dass es Randfälle und häufige Zielmuster abdeckt. Die Studie wurde an verschiedenen Positionen im Raum durchgeführt, und die Controller wurden innerhalb und außerhalb des Sichtfelds der Nutzer getestet. Zusätzlich wurde untersucht, wie sich die Position der Ziele auf die Leistung auswirkt.

Innerhalb des Sichtfelds zeigten die Ergebnisse, dass Controller mit niedrigerem Abstraktionsniveau, wie die *Pillar Map* und *Cluster Map*, besser abschneiden. Außerhalb des Sichtfelds sollte der Controller entweder taktile Elemente haben, die leicht zu erkennen sind, wie die *Pillar Map*, oder er sollte ohne visuelle und taktile Hinweise funktionieren, wie der *Direction Swiping Controller*.

Acknowledgements

First of all, I would like to thank Prof. Dr. Jan Borchers and Prof. Dr. Jürgen Steimle for examining this thesis.

I would also like to thank my supervisor Oliver for supporting me throughout the thesis with his dedication and enthusiasm for the topic.

Additionally, I would like to thank everyone at the chair, for answering questions and helping me out when I needed help.

Furthermore, thank you to everyone who participated in the user study.

A special thanks to my girlfriend Ann-Kathrin, who spent her sparse free time helping me come up with ideas, motivating me, calming me down, proofreading, and just being there for me every day.

Lastly, I want to thank my parents for financially and emotionally supporting me throughout my studies.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

The whole thesis is written in Canadian English.

The first person is written in plural form.

The singular *they* is used to refer to unidentified third persons.

Chapter 1

Introduction

Modern living rooms are increasingly equipped with smart electronic devices. Device types vary heavily, from media devices such as TVs and stereos to lamps and air conditioners to robotic vacuum cleaners. All of these devices require a way of control. In nowadays smart home setups this is often done via the user's smartphone, universal remotes, or voice control by smartphone or smart speakers.

Modern living rooms are increasingly equipped with interactive devices.

Smartphones and universal remotes are not always with the user, leading to situations where the user is unable to control their home. While voice controls are always available, they lack visibility of the available functions, requiring the user to remember all commands. This problem increases even more when targeting individual devices, as different device categories require different commands and each device a unique name. That name should also be easy to remember and fit natural speech. These commands are usually only known to the administrator of the system, which makes it hard for guests to perform tasks like adjusting the lighting. Furthermore, it is not always socially desirable to use voice assistants, for example, while watching a movie or during a conversation. This makes selecting and controlling single devices, in particular, cumbersome using current methods.

Smart Home controls lack intuitive selection methods.

Concluding, a stationary controller that allows individual devices in a room to be selected and controlled would be

Prime locations for a controller are at a couch, an armchair, or a table.

desirable. As the exact position of the controller is not clear, it should be possible to use the controller without having vision on it, to allow placements outside the field of view. Prime locations for such a controller would be places where the user frequently spends time. In a living room, these could be a couch, an armchair, or a table.

Natural mappings help users understand controllers intuitively.

To guide users in understanding a controller intuitively, natural mappings should be considered. Mappings in general are relations between controls and the devices they control. Norman described that designers should use mappings that "[take] advantage of spatial analogies" which "leads to immediate understanding" (Norman [2013], p. 22). This reduces memory load of users and makes the system accessible to users who are unfamiliar with the setup or new to the environment.

Mapping three dimensions onto a controller can be problematic.

In a living room, devices are distributed in a three-dimensional space. Controls that use spatial mapping techniques must either be three-dimensional themselves or reduce the dimensions. Reducing dimensions can be problematic, especially when multiple devices share position in one or more dimensions, such as when they are on top of each other. A controller for selecting devices in a room must be able to handle such scenarios.

We evaluated five controllers on performance for target selection in a real-world scenario.

In this thesis, we present target selection techniques for real-world scenarios. For this purpose, we constructed five different controllers. These make use of natural mappings in varying abstraction levels to allow the user to quickly and easily understand the controller. In a user study, we compared the following five controllers: an abstraction of a room made of pillars (*Pillar Map*), a 2D map of the room, surrounding overlaying targets with a box (*Cluster Map*), a 2D map of the room divided into three height levels (*Zelda Map*), a direction-based controller with buttons (*Sun Controller*), and a direction-based controller using swipe gestures (*Direction Swiping Controller*). As the prime locations for these controllers are typically covered with textile surfaces, textile inputs should be considered when designing the controllers. Like this, the controllers should blend in with their surroundings. To investigate the intuitiveness of the controllers, we constructed an exemplary living room environment with targets to select. Study participants selected different targets in the exemplary living room by using each of the controllers. The controllers were evaluated on their performance, meaning the time participants needed to select the correct target and how accurately the targets were selected. During the study, the effect of different positions in a room and having vision on the controller or not were also investigated.

Chapter 2

Related work

2.1 Interactive Fabric and E-Textiles

As prime locations for a stationary controller are often made out of textiles, the possibilities and limitations of textile controls should be kept in mind when designing a controller. Therefore, this section will provide an overview of how to construct textile interfaces, how they should be designed, and which types of controls can be created with textiles.

Early studies on interactive fabrics reach back to 1997. Post and Orth [1997] proposed to use conductive textiles to build flexible, wearable, and washable computers. Since then, a multitude of fabrication techniques, guidelines and use cases were identified and introduced.

2.1.1 Fabrication Techniques

Interactive fabrics can be manufactured in multiple ways. From attaching interactive elements on fabric by for example embroidering over making whole patches of fabric conductive to creating new fabrics out of conductive yarns. These fabrication techniques can be categorized into additive and constructive fabrication techniques.

Fabrication techniques for e-textiles can be categorized into additive and constructive techniques.

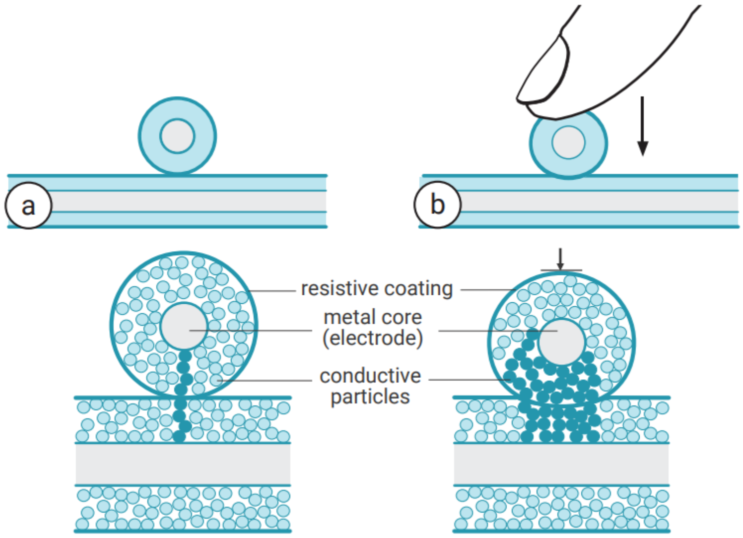


Figure 2.1: Cut-through of the yarn proposed by Parzer et al. [2018]. The conductive metal core is surrounded by a resistive coating, changing its resistance when pressure is applied.

Additive Fabrication

Additive fabrication techniques are those that enhance existing non-interactive fabrics into interactive ones. This can be done for example by embroidering the fabric with special yarn, or by directly attaching interactive elements to the fabric.

Parzer et al. and Roh proposed conductive yarn that can be used to create resistive touch sensors.

A common technique is to use conductive yarn that can be used to recognize user inputs. Parzer et al. [2018] proposed a method to create conductive yarn by coating a metal core with resistive material. Like this, a single conductive thread is created. When pressure is applied to two overlapping yarns, the resistive coating gets compressed. This increases the density of conductive particles as seen in Figure 2.1 which lowers the resistance of the coating. This change of resistance can be measured and interpreted as a touch input. This way, textile resistive touch sensors can be created, that can be used to measure positional pressure data.

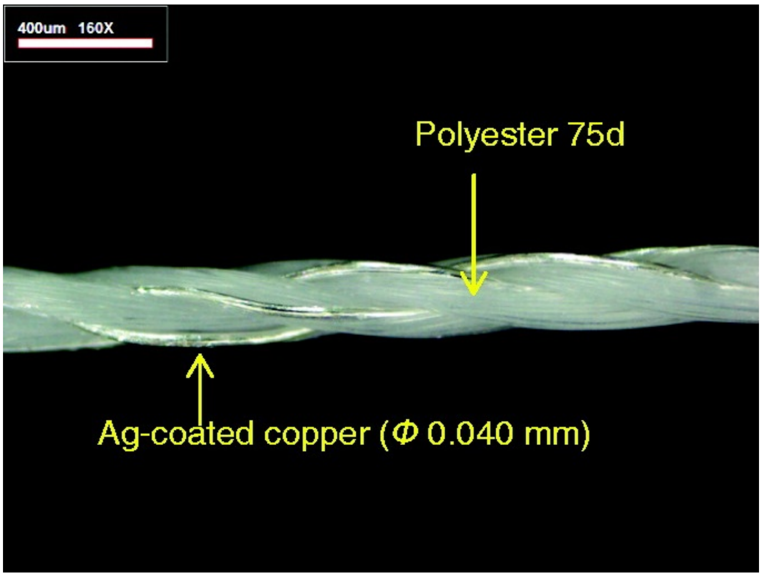


Figure 2.2: Image of the yarn proposed by Roh [2014]. Conductive copper threads are combined with non-conductive polyester threads to create a conductive yarn that can be used to recognize touch inputs.

The same functionality provides the yarn proposed by Roh [2014]. Instead of creating one thread, their approach was to combine and twist conductive and non-conductive threads. In their case, copper is used as conductive and polyester as non-conductive threads which is depicted in Figure 2.2. It can be used in a similar way to build a resistive touch sensor.

Apart from embroidery, there are also other possibilities to enhance existing fabrics. Additive fabrication techniques are especially interesting for rapid prototyping, as no new fabric needs to be produced.

Klamka et al. proposed a rapid prototyping technique by ironing interactive elements on fabric.

Exemplary for this field of research, Klamka et al. [2020] proposed a technique for creating textile prototypes by ironing interactive and conductive elements on fabric. Like this, capacitive touch sensors, sliders, and bend and pressure sensors can be built. Moreover, output elements can be ironed on the fabric. There are multiple ways to create displays, either using electroluminescending wires, LEDs or E-Ink displays. In combination, complex interactive systems can be created quickly on fabric, while keeping the flexibility and feel of the fabric.

The technique by Honnet et al. allows to make whole patches of fabric conductive by polymerization.

Lastly, Honnet et al. [2020] proposed to enhance fabric by polymerization. In the process, the fabric is put into a bowl with pyrrole and iron chloride, which leads to conductive polymer chains forming in and around the fabric. This process makes the fabric conductive while keeping its haptic and mechanical properties. It can be applied to strings, patches of fabric or even complete clothes. Stretching and touching the fabric will result in changes in the fabric's resistance. Measuring this allows the implementation of sensors for touch and deformation.

Constructive Fabrication

As an alternative to enhancing existing fabrics, new interactive fabrics can be produced. These methods are called constructive fabrication methods. Usually, these techniques involve knitting or weaving a new fabric out of conductive yarns.

KnitUI is a technique to create interactive fabric of various colours and forms by machine knitting.

KnitUI is a technique by Luo et al. [2021], that is used to fabricate interactive fabrics by machine knitting. The technique allows creating textile user interfaces with resistive touch sensing, in which shapes and colours of control elements can be adjusted to the users' needs. Additionally, the shape and colour of the fabric and input elements can be changed. Their technique works for ordinary knitting machines and therefore allows to quickly create interactive fabrics. Only minimal manual post-processing is needed, making the technique easy to use. The resulting fabric is made of conductive and non-conductive threads, enabling to create wearable and stationary textile user interfaces like depicted in Figure 2.3.

Poupyrev et al. [2016] presented a technique to create e-textile using weaving. With conductive yarns, fabrics with capacitive touch sensors can be fabricated. These can be integrated seamlessly or combined with patterns, textures,

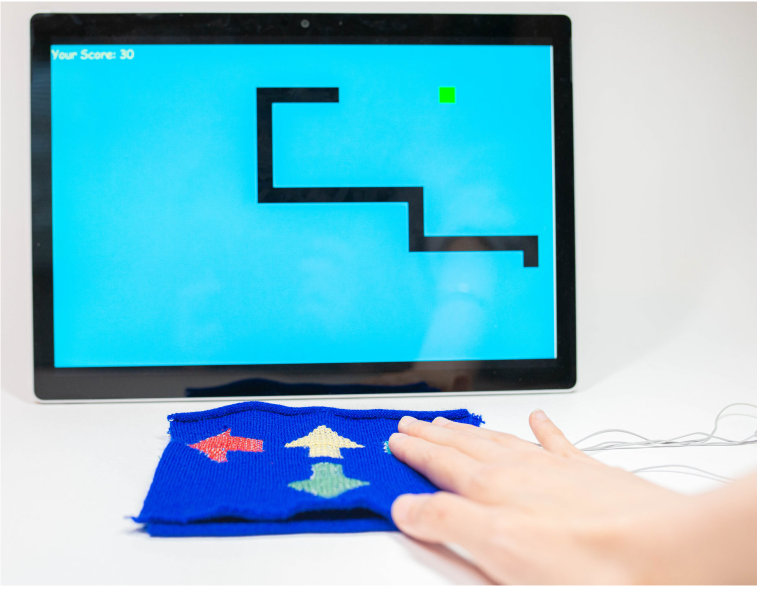


Figure 2.3: A knitted controller created with the techniques proposed by Luo et al. [2021] that is used to play snake on a connected computer.

colours, and materials to make them visible and graspable to the user. Like this, complete textile user interfaces can be created on a large scale.

2.1.2 Textile Inputs

As already mentioned, there are multiple techniques to create touch controls with fabrics. The techniques proposed by for example Luo et al. [2021] and Honnet et al. [2020] allow for the creation of resistive touch sensors. The technique by Poupyrev et al. [2016] allows to create capacitive touch sensors. This can even be further enhanced, as Wu et al. [2020] demonstrated a textile capacitive touch sensor, that works from both sides of the fabric and can even distinguish between inputs on the two sides.

Textiles can be created with capacitive and resistive touch sensors.

The tactile properties of fabric enable us to also implement less common input techniques. As fabric is flexible and stretchable, novel input techniques can be created. The fab-

Stretch and pinch sensors can be integrated in e-textiles.

rics by Honnet et al. [2020] also allow to measure stretching of the fabric which can be used to create stretch sensors. Hamdan et al. [2016] make use of the flexibility of fabrics to create a pinch sensor.

Tessutivo allows to recognize objects by conductive sensing.

Gong et al. [2019] proposed Tessutivo, a technique for inductive sensing on interactive fabrics. The technique offers the possibility to sense and recognize conductive objects that are placed on the fabric. Non-conductive objects can be made recognizable by attaching constructive strips to the bottom of the object. These can be sensed by Tessutivo to recognize the object then.

2.1.3 Textile Outputs

Dynamic textile can be created with techniques by Kwon et al. and Devendorf et al.

E-textiles can also be used as outputs to communicate with the user. Kwon et al. [2018] demonstrated how thin, weavable fibre OLEDs can be fabricated. These can light up efficiently with high luminance and can be integrated into e-textiles. Devendorf et al. [2016] presented an e-textile that is made out of conductive threads that are coated with thermochromatic paint. These can change their colour in a matter of seconds without emitting any light. With techniques like these, dynamic textile displays can be created, that could be used for future textile controllers.

Pneumatic actuators can make textile contract and expand.

Zhu et al. [2020] combined flexible textiles with pneumatic actuators. These actuators can make the fabric contract or expand. They used these movements to make a textile sleeve vibrate by repeatedly contracting and expanding it. This vibration was used to communicate with the user.

Using tendons, textile objects can change shape by pulling on a string.

Albaugh et al. [2019] demonstrated a special knitting technique, that allows the creation of shape-changing textiles. A tendon is integrated into the fabric. By pulling on the tendon, the fabric deforms, creating actuated textile objects like in Figure 2.4.

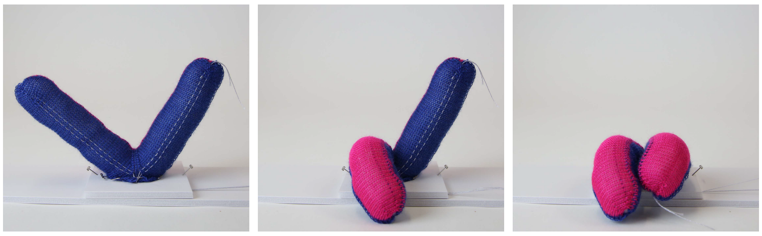


Figure 2.4: Textile V-shape with integrated tendons as proposed by Albaugh et al. [2019]. Both legs of the V can be moved independently by pulling at the attached strings.

2.1.4 Designing Textile Controllers

Due to the tactile differences between fabric and common user interfaces, knowledge about how to design textile controllers is still limited. Mlakar and Haller [2020] investigated how textile elements should be designed. Their focus was mainly on the properties buttons and icons need, to be distinguishable from each other. In their studies, a special focus was put on eyes-free interaction. The results showed that height differences are the easiest to recognize. The minimal size for tactile elements should be 13 mm to make them recognizable eyes-free. They also showed that simple shapes are easier to recognize than complex ones.

Textile elements should be 13 mm large and preferably simple. Height is the property that is the easiest to recognize eyes-free.

In a follow-up study, Mlakar et al. [2021] investigated how affordances can be used on fabrics to propose interaction to the user. Due to the rich tactile properties of textiles, affordances can be created by colours, textures and types of fabric. For example, a fabric like in Figure 2.5 A affords to be stroked in a clockwise direction. Other directions can be indicated with the textile in Figure 2.5 C. The stitched boxes are only fixed on one side, making a swiping gesture in one direction easy and in the other direction hard to perform. The fabric in Figure 2.5 B offers the affordance to be pressed and textiles like in Figure 2.5 D tend to be touched more often, due to their pleasant look.

Textile colours, shapes, and textures can be used to afford interactions.

Nowak et al. [2022] explored different designs for textile sliders. They investigated the effect of the form and textile design properties on users' preferences and performances.

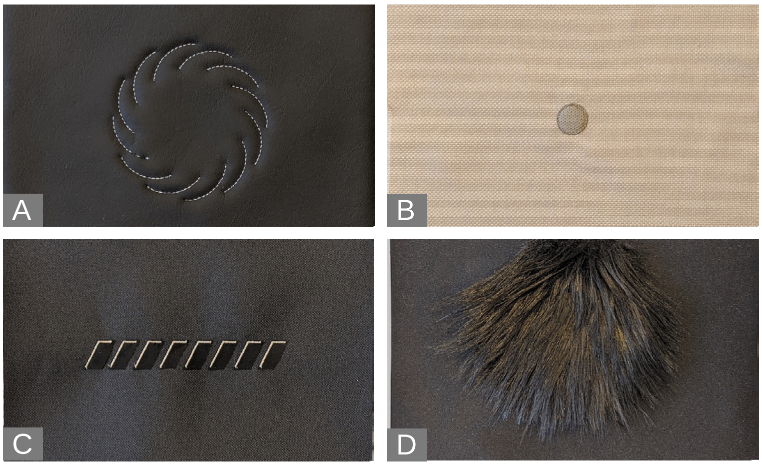


Figure 2.5: Patches of fabric that afford proposed by Mlakar et al. [2021]. The fabrics in A, C, and D afford to be stroked in a particular direction, and B affords to be pressed.

Tactile orientation points, like tick marks, helped users orientate on the slider.

These studies focus on the design of single input elements but do not consider the general design of composed textile user interfaces. Research in that field is still missing and should be looked upon in the future.

2.2 Mappings

Mappings are relationships between controls and the devices they control, or the actions that get performed. As Norman [2013] described, good mappings are natural mappings. These are mappings, that the user of a system can understand, without further knowledge.

Spatial analogies can be used to make mappings easy to understand.

Spatial analogies are the primary measure to make mappings easy to understand. Spatial mappings are used, when the controls are aligned similarly to the devices that are controlled by them. An example can be seen in Figure 2.6. Figure 2.6 A and B show examples of how not to map the controls. These mappings require additional labels, as without

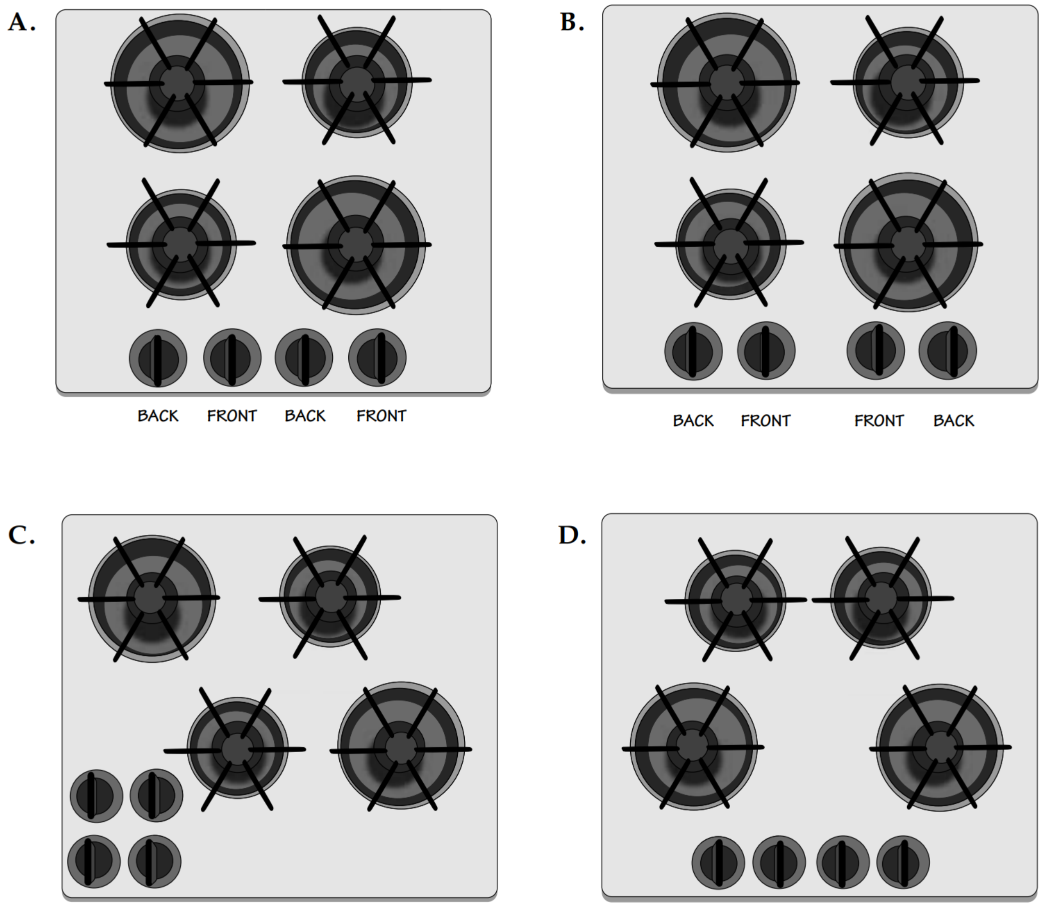


Figure 2.6: Examples for a stove control with burners using natural mappings proposed by Norman [2013]. The mappings in A and B are not intuitive, as the controls cannot be assigned to the stove burners unambiguously without labels. In C and D, the controls and burners are realigned to make the mapping understandable just by the layout.

them it would not be possible to know which controller is for which stove. Figure 2.6 C and D show how the layout could be improved to make the mapping immediately understandable. Placing the controls either in a two-by-two grid which resembles the stove burners, or realigning the burners, so that each has a unique position on one coordinate makes the mapping understandable without the necessity of labels.

The spatial mappings proposed by Norman [2013] focus

Figure 2.7: Seat adjustment in a Mercedes-Benz. The control resembles the device it controls, making the interaction with it intuitive. Image taken from Norman [2013].

Neglecting a dimension for a spatial mapping can lead to ambiguous mappings.

on mapping controllable devices to one or two input dimensions. The most natural way is to keep the spatial dimensions in the controller domain the same as in the device domain. This can be seen in Figure 2.6 C, where two dimensions are used to distribute the controllers and two dimensions to distribute the controllable devices. Alternatively, a dimension in the device domain can be neglected, reducing the dimension in the controller domain, as done in Figure 2.6 D, where the control distribution is only one-dimensional, but the device distribution is two-dimensional. To do this, the controllable devices were reorganized to give them a different spatial distribution in one dimension. Figure 2.6 A and B are examples where spatial information in one dimension is not enough to distinguish devices. Therefore the mappings are not intuitive and self-explanatory.

Another possibility to increase the intuitiveness of a mapping is to make the controls resemble the device they control. An example of this can be seen in Figure 2.7. The controller for the seat is shaped similar to the seat. Pushing the upper part of the controller to the front tilts the seat's back to the front.

Park et al. [2014] conducted studies on how mappings can be further improved and made more intuitive. Based on Norman's guidelines, they proposed feedforward as a measure for more intuitive mappings. This is done by showing the user the effect of an action before it is performed. In their study, they used a light switch which made the corresponding light shine in a dimmed state when touching the switch. While this reduces the likelihood of accidental activation, it does not lead to a quicker understanding of the underlying mapping.

Accidental activation can be avoided by feedforward.

2.3 Target Selection Methods

Studies on the selection of individual targets in spaces with high target density have been mostly focused on virtual reality and augmented reality environments. On the one hand, Montano-Murillo et al. [2020] proposed letting the user slice the 3D environment in 2D planes on which they then make a selection. Wonner et al. [2012], on the other hand, presented a technique that allows the user to make a pre-selection before narrowing it down to one target point. As these techniques rely on virtual elements for support and guidance of the user, they are not applicable to real-world scenarios.

Individual target selection techniques for virtual and augmented reality cannot be applied to real-world scenarios.

2.4 Eyes-Free Interaction

Eyes-Free interaction with user interfaces is mainly investigated with a focus on visually impaired users. Most of these studies focus on how to design digital user interfaces on a computer, like Alonso et al. [2008], who developed a framework to design hybrid user interfaces, that fulfill the requirements of visually impaired users, while staying efficient and satisfying to use for sighted users. Another example is the study by Bornschein and Weber [2017]. They analyzed the requirements blind users have for digital draw-

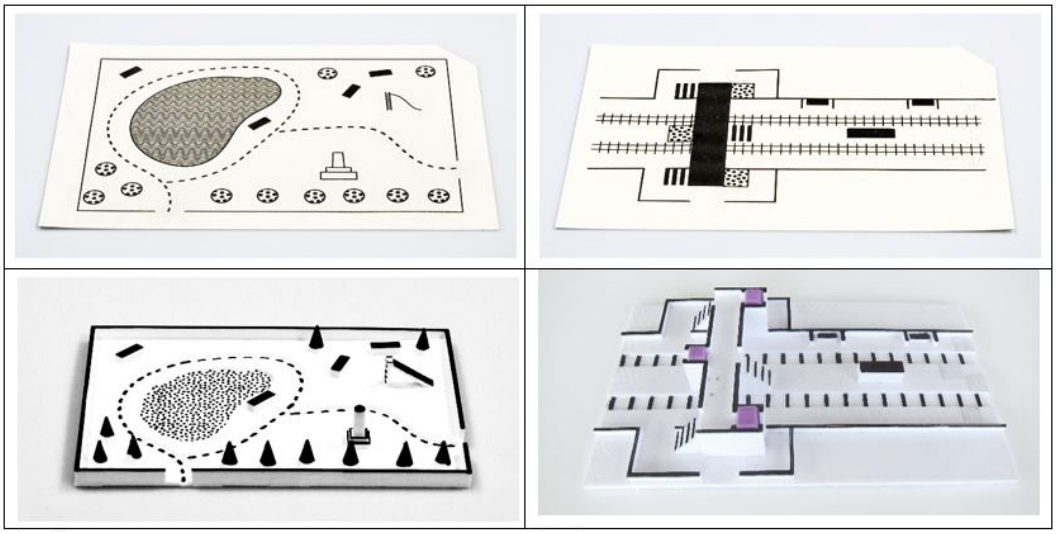


Figure 2.8: Tactile maps by Holloway et al.. A park and a train station were mapped using a two-dimensional and a three-dimensional approach. For study participants, orientating on 3D maps was easier than on 2D maps.

ing tools and provide these as a prioritized list. Some of the proposed points are more general and also relevant for non-digital user interfaces, like having an easy and intuitive to learn interface, but most requirements focus especially on the task and on digital user interfaces.

Three-dimensional tactile maps help understanding an environment.

Holloway et al. [2018] investigated how maps for visually impaired users should be designed. They compared 2D tactile maps to 3D tactile maps. An image of the maps that were compared can be seen in Figure 2.8. Their results showed, that 3D maps were preferred by users and led to an easier understanding of the environment. On the one hand, 3D icons, like the trees in the maps on the left side and stairs on the maps on the right side, were easier to recognize than their corresponding 2D icons. This led to an easier understanding of the map. On the other hand, the 3D map showed improved short-term recall, as participants felt that it was easier to build a model of the environment shown by the map.

Palivcová et al. [2020] conducted studies on the effect of 3D tactile maps on elder adults with vision impairments. As many of these have age-related visual limitations, they often lack experience in reading tactile maps and do not have well-developed tactile acuity. Studies showed, that tactile maps helped participants to build spatial knowledge of the environment.

During the research, we could not find further studies on how individual targets can be selected in real-world environments. Therefore, we believe that further research is necessary. We will investigate how controllers to select such targets should be designed.

Chapter 3

Controller and Room Design

This chapter describes the design and fabrication process of the controllers and the exemplary room used during the user study. Firstly, requirements for the controllers will be outlined. Afterwards, the general concept and the construction process of each controller will be explained. Lastly, the requirements and the design for the exemplary living room will be presented.

3.1 Controllers

3.1.1 Requirements

The controllers need to be three-dimensional or provide methods to reduce the dimensions.

In a living room environment, devices are distributed in three dimensions. As the controllable devices have fixed positions and cannot be redistributed, corresponding controllers either need to be three-dimensional or provide reliable methods to reduce the dimensions. Otherwise, problems, like described in Section 2.2, could occur. The more devices should be covered by the controls, the harder it becomes to find an unambiguous mapping using a two-dimensional controller. Devices could be placed on top of each other or in the same direction relative to the user, making them indistinguishable depending on the neglected dimension.

The controllers' concepts need to be applicable to all types of rooms.

The controllers are constructed specifically for the exemplary room used in the study. It is therefore not required for the controller to be dynamically adjustable to changes in the room. While this is a desirable feature for an actual controller, it is not necessary for the prototypes used during the study, as the room will not change. On the other hand, the concept for each controller should be applicable to all types of rooms and device distributions.

All controllers should also be usable without having vision on it. Therefore, all required information to use the controller should be haptically explorable.

Measurements should be done electronically.

Each controller has some sort of input modality that lets the user select each target in the room individually. Spatial mappings are used to make the mapping of the controller easy to understand, as proposed by Norman [2013]. In the study, the targeting time and accuracy of the user on each controller will be measured. To reduce measuring inaccuracy, controller input recognition and time measurements should be done electronically. Therefore, each controller should be able to communicate the user's input to a computer directly or via a microcontroller, like an Arduino.

3.1.2 Concepts

Five spatial mapping techniques will be investigated in this thesis.

We decided on five different concepts for the controllers, each using a different spatial mapping technique and fulfilling the requirements. The concepts follow either an absolute or relative spatial approach. Relative spatial mappings are dependent on the controller's position. Devices are mapped to controls based on the position relative to the controller. Absolute spatial mappings do not depend on the controller's position. Devices are mapped to controls based on their position in the room. The concepts we considered follow different abstraction levels of the mapping. The higher the abstraction level, the less the controller rep-

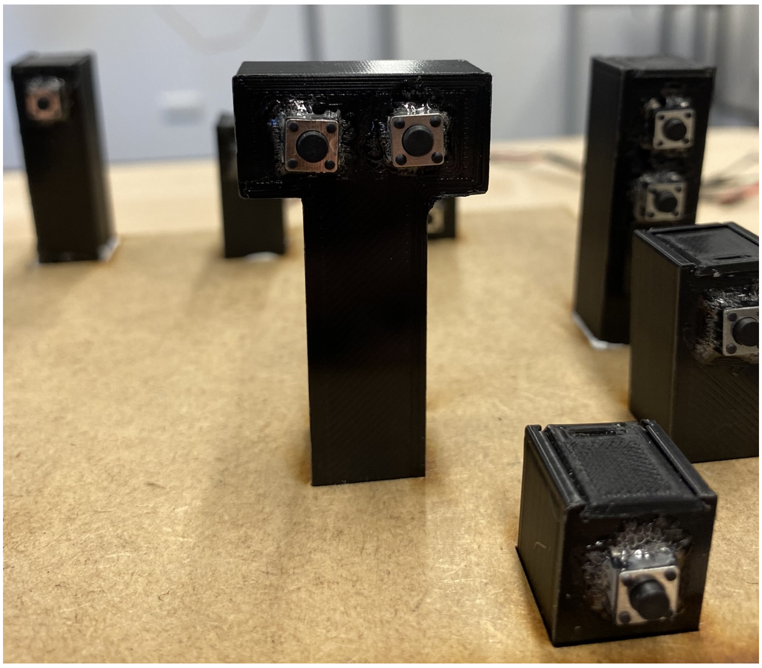


Figure 3.1: A close look at the pillars used for the *Pillar Map*. In the centre is a pillar that represents two devices close to the ceiling.

resents the room. This makes it easier to use the controller in other rooms without adjusting the controller.

Pillar Map

The first approach is to construct a 3D abstraction of the room using pillars. Each device is represented by a pillar. The height of a pillar scales according to the height of the device in the room. Devices that are on top of each other or close to each other are put on one pillar. A pillar has buttons at the position of the devices which can be pressed by pinching the pillar. In Figure 3.1 a pillar representing two devices placed near the ceiling can be seen.

Pillar Map is a three-dimensional controller, abstracting the room with pillars.

The controller includes information about each device's position in the room, as well as its height. As the controller is three-dimensional, no reduction of dimensions is necessary

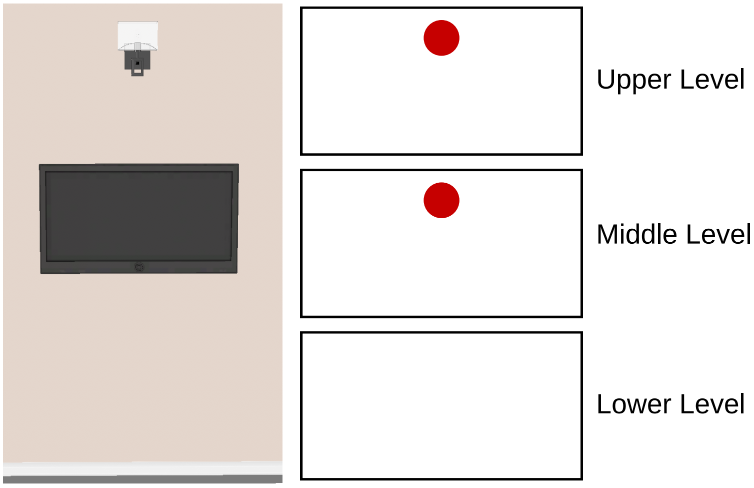


Figure 3.2: Example of the *Zelda Map*. The lamp is mapped to the upper level, as it is placed above head height, the TV is mapped to the middle level, as it is placed between hip and head height.

and every device can be mapped to the controller without problems.

Zelda Map

Zelda Map is a two-dimensional map divided into three height levels.

The second controller represents a two-dimensional map of the room. The perspective for the map is a top-down view. Devices are marked with dots on the map which can be pressed to select the device. Due to the reduction of dimensions, there are devices that would be mapped to the same spot on the map when devices are positioned exactly above each other. To avoid this problem, the idea for the Zelda Map is to divide the room into three height levels and have one map for each level. For the thresholds between levels, we decided to use roughly the hip and head height of a human. The results are three two-dimensional maps, one with all devices below hip height, one with all devices between hip and head height, and the last one that has all devices above head height. These maps are placed behind each other, with the map of the lowest level closest to the user and the highest map the furthest away.

Figure 3.3: Example mapping of the *Cluster Map* showing how a floor lamp (button 1), TV (button 2), and speaker (button 3) are mapped to the controller.

An example is shown in Figure 3.2. The lamp is positioned above head height and therefore mapped to the upper level of the *Zelda Map*. The TV is positioned between head- and hip height. As a result, it is mapped to the middle level. The lower level is left empty, as there is no device.

Cluster Map

Cluster Map represents the room as a two-dimensional map, grouping devices on top of each other into a cluster.

A similar concept is used for the Cluster Map. Again a two-dimensional, top-down map of the room will be used as a basis for the controller. Each device is marked by a dot, that can again be pressed by the user to select it. Instead of dividing the map into three levels, one map is used. In cases where multiple devices are on top of each other, these are grouped into a cluster. A cluster is surrounded by a box. Inside this box, the devices are ordered from lowest to highest, with the highest device in the room being placed furthest away from the user on the map and the lowest device in the room mapped closest to the user on the map. Thanks to the box around the cluster, devices that are above each other can be differentiated from devices that are next to each other.

In Figure 3.3 an example mapping is depicted. As the TV

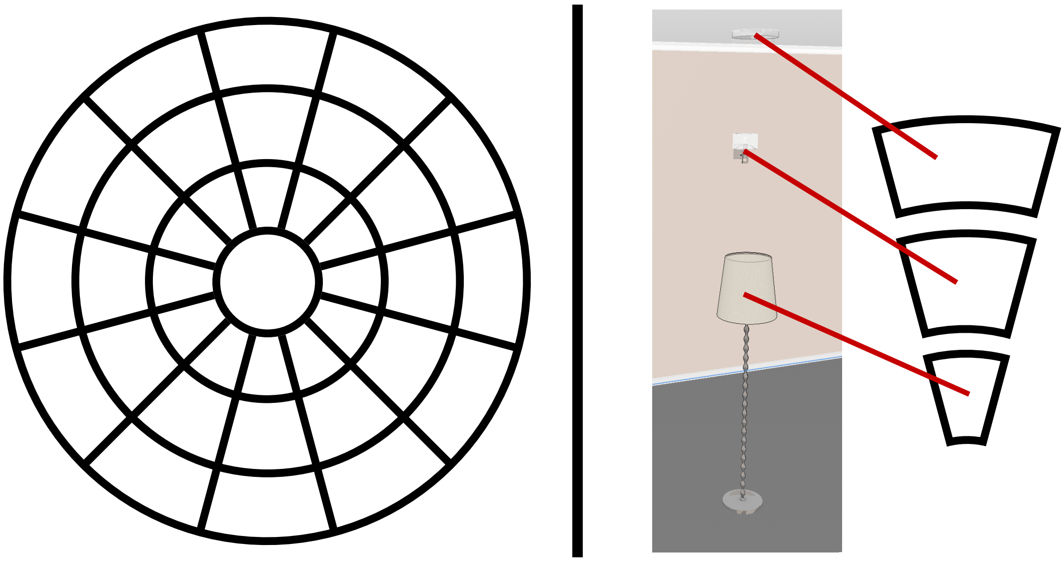


Figure 3.4: A schematic view of the *Sun Controller* on the left side and an example mapping on the right side. Devices in one direction are mapped to the controller based on their perspective height.

and the speaker are positioned at the same place, but above each other, they are grouped into a cluster for the mapping. The TV is assigned to button 2, the speaker to button 3. The floor lamp to the left is separated from the cluster and mapped to button 1.

Sun Controller

The Sun Controller maps targets to buttons depending on their direction and perspective height.

Different from the approaches before, the *Sun Controller* uses a relative spatial mapping. The controller is round and separated into twelve equally sized sections which are further divided into three buttons as displayed on the left side of Figure 3.4. Each section corresponds to a direction. The buttons in each section can be pressed to select a device in that direction.

The Sun Controller is divided into twelve directions.

This controller consists of twelve different directions. Front, back, left and right need to be easy to identify which leads to a design with directions dividable by four. Second, every button should be large enough to be pressed without pressing the buttons next to it, but the overall controller should not be too large. Therefore we set the maximum diameter to 15 centimetres. The twelve directions offer a good balance between size and number of buttons. Moreover, it is already a common way to communicate directions in twelve directions by giving the directions on a clock.

If there are multiple devices in one direction, the assignment depends on the perspective height of the devices. The device that is seen topmost from the user's perspective will be mapped to the outer button. The lowest device will be assigned to the inner button. Each device is mapped to exactly one button. If there are less than three devices in one direction, outer buttons are assigned first. This means, that the innermost button in a direction is only assigned to a device when there are three devices in that direction.

An example can be seen on the right side of Figure 3.4. As the floor lamp is the lowest from the user's point of view, it is mapped to the inner button. The ceiling lamp is the highest from the user's perspective and is therefore mapped to the outer button. The remaining wall lamp, which can be seen between the other two lamps in the picture, is mapped to the middle button.

As there are three buttons per direction, up to three devices per direction can be selected directly. If more than three devices are in one direction or devices at the same perspective height are too close to each other to be assigned to different directions, multiple devices are assigned to one button. The button can then be pressed multiple times to iterate through these devices. As some sort of confirmation is required for the study, the selection is confirmed when the user does not make any further input for one second.

Direction Swiping Controller

The Direction Swiping Controller maps devices to swiping gestures based on their direction.

The last controller also follows a direction-based approach. Instead of having buttons, directions should be selected by swiping. Devices can be selected by performing a swiping gesture in the direction of the device. If there is more than one device in a direction, swiping marks all devices in that

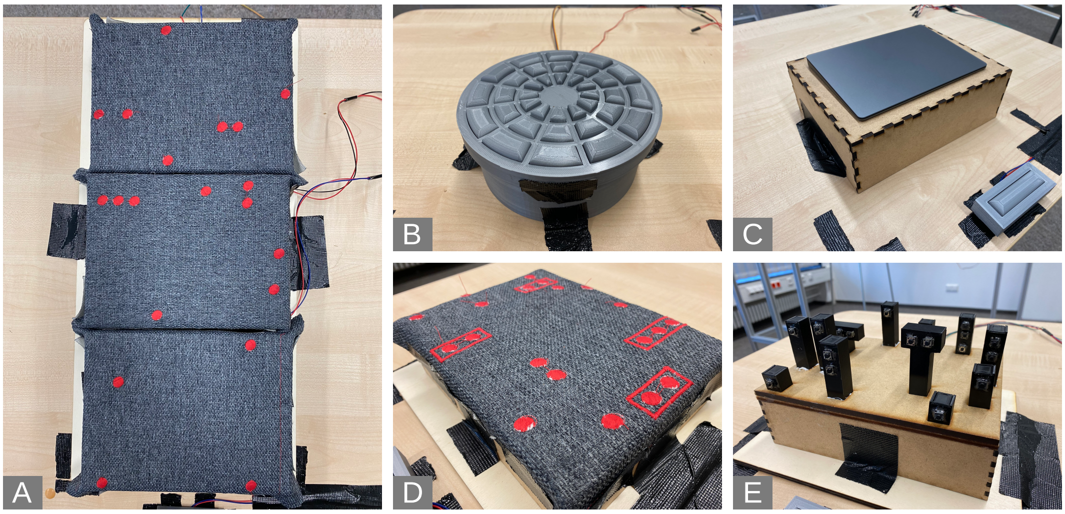


Figure 3.5: All five controllers used in the user study. The controllers are *Zelda Map* (A), *Sun Controller* (B), *Direction Swiping Controller* (C), *Cluster Map* (D), and *Pillar Map* (E).

direction. It is then possible to iterate through these devices in that direction by pressing on the controller. The selection is confirmed when no further input is made for one second.

3.1.3 Construction

DTS61K button that is used in all button-based controllers.

First of all the construction of the button-based controllers will be described. The only controller, that is not button-based is the *Direction Swiping Controller*. As it should be possible to recognize inputs on the controller with a computer to make accurate measurements, we decided to use DTS61K buttons as interactive elements on the controllers. These buttons are connected to an Arduino Mega 2560 which recognizes when which button is pressed. How the recognition of the pressed button works is explained in Section 3.1.5. For the *Sun Controller*, *Cluster Map*, and *Zelda Map*, the buttons are directly placed on a 150 mm by 200 mm wooden base. The *Zelda Map* uses three of these bases. The bases are made out of 3 mm MDF plates with holes for the body of the button and 1 mm thick finboard with holes for the buttons' pins. These two layers were glued

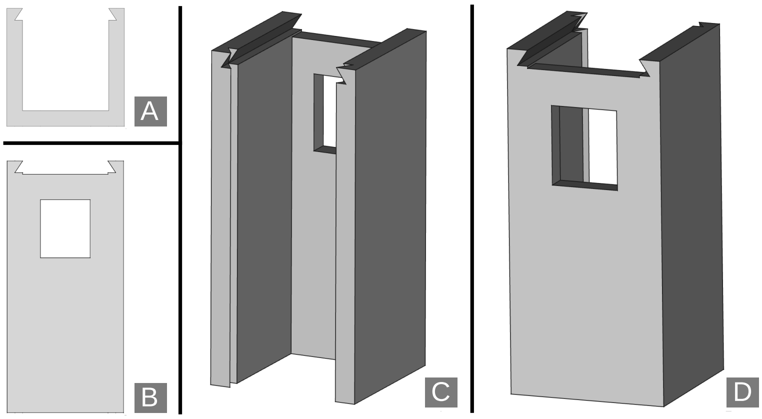


Figure 3.6: 3D Model of the pillars used for the *Pillar Map*. The view from below can be seen in A. B is the view from the front of the box. C and D are isometric views from the front and back of a pillar.

onto each other using wood glue. For the *Cluster Map*, an additional layer of finboard is glued around button clusters to raise these areas and make them recognizable without looking at the controller. On the bottom of the base, wires are soldered to the buttons' pins, which are then connected to the Arduino.

The button-based controllers have a wooden base with holes for the buttons' pins.

For the *Pillar Map*, the buttons are not placed on the wooden base, but in the pillars that resemble devices in the room. These pillars are 3D-printed using an Ultimaker S5 3D printer. The 3D model of a pillar for a single button is depicted in Figure 3.6. The buttons fit tightly into the large hole in the front of the pillars and are glued additionally to make them withstand pressure from user input. The top and back sides of the pillar can be closed with a lid that fits into the ledges of the pillar. The lid allows access to the pins of the pillar's button to solder wires onto them. The pillars are then put into a 150 mm by 200 mm wooden base, again made out of a 3mm MDF plate and 1mm finboard. The MDF plate has holes the size of the pillars (15 mm by 15 mm). The finboard has holes of the size 11 mm by 11 mm, which are large enough to pass the wires through. As some pillars were not standing stable and were wiggling a bit when touched, these are glued with a hot glue gun onto the base. The finally assembled controller can be seen

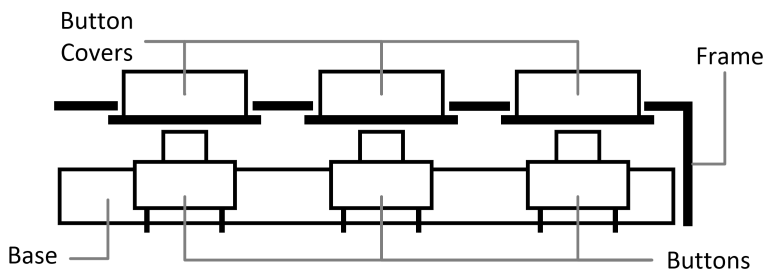


Figure 3.7: Schematic view of how the cover is placed on the *Sun Controller*. Buttons are placed in a base with the buttons' pins reaching through the base. The covers are used to increase the size of the surface that can be pressed. The frame holds the covers in place.

in Figure 3.5 E. All wood was cut using an Epilog Fusion M2 40 laser cutter.

The *Sun Controller* has a 3D-printed frame and button toppings to increase their size.

For the *Sun Controller*, we decided to fabricate toppings for the buttons to increase the size of the buttons. Therefore, we 3D-printed an additional frame and button covers. A schematic view can be seen in Figure 3.7. Covers are placed on top of each DTS61K button. When the user presses on a button cover, the button below will be pressed. The covers are held in place by the frame that surrounds the controller. The resulting controller is depicted in Figure 3.5 B.

An *Apple Magic Trackpad 2* is used as *Direction Swiping Controller*.

Lastly, we needed a controller to recognize swiping gestures. To minimize the effort for prototype creation, we decided to use an *Apple Magic Trackpad 2* as the controller. This trackpad is connected to a computer which runs a python program that recognizes clicks and swiping gestures on the trackpad. The swiping gestures can be distinguished between eight directions, up, down, left, right, and the diagonal directions in between. When a gesture or click is recognized, a command is sent to the connected Arduino which then handles the input. The selection of direction can be changed until the input was confirmed. This allows for input corrections which are not possible with the other controllers, but the design of the *Direction Swiping Controller* offers to make corrections easily and immediately, therefore it would be unnatural to forbid them. An image of the *Direction Swiping Controller* can be seen in Figure 3.5 C.

The controllers *Cluster Map* (cf. Figure 3.5 D) and *Zelda Map* (cf. Figure 3.5 A) are covered with textiles, as we want to use the controllers on textile surfaces in the future. The textile cover is made out of grey sofa fabric. Above each button of the controller, the fabric is marked with red dots that are embroidered on the fabric. Below each dot, a layer of embroidery foam is embroidered under the fabric to make the buttons easier to recognize eyes-free by raising the button from the fabric. As Mlakar and Haller [2020] investigated, height differences are the easiest differences to recognize eyes-free. For the *Cluster Map*, clusters are marked by a red box that is embroidered onto the fabric using a raised satin stitch. The box is raised by the stitch additionally to the finboard, resulting in a height difference of more than 1.6 mm, which should be noticeable without looking at the controller according to Mlakar and Haller [2020]. All embroidery was done with a BERNINA 880 computerized embroidery machine.

The *Cluster Map* and *Zelda Map* are covered with sofa fabric.

For all other controllers, the technical limitations of the prototype made it impossible to cover them with fabric. For the *Sun Controller* fabric needed to be either put between or on top of the button cover and frame. When put between, the buttons often were stuck after pressing, making the controller hard to use. When put above, the buttons are hard to recognize when touching the fabric. The *Direction Swiping Controller* does not work reliably with fabric on top. For the *Pillar Map*, we decided against a textile cover, as we would have needed to fabricate one for each pillar separately.

All controllers are raised by 5cm above the table they will be placed on. For this, we put a box below each controller. For the button-based controllers, this is necessary, as the wiring of the buttons is below the controller. The *Direction Swiping Controller* was also put on a box so that all controllers have the same height. This was done to avoid having targeting differences in the user study due to a longer way to the controller. For the *Sun Controller*, we 3D-printed a ring that is placed below the controller, for all other controllers a box was made out of MDF plates that are glued together using wood glue.

The controllers are placed on boxes to avoid breaking the wiring below.

3.1.4 Data Acquisition

An Arduino is used to measure and collect data.

As already mentioned, measurements should be done automatically to reduce measurement inaccuracies. As we already use an Arduino to read input from the controllers, this Arduino is also used to make measurements and collect data. To do this, the Arduino will also control the rest of the experiment during a study so that it has access to all necessary time stamps to make calculations. Because the Arduino cannot directly persist data, it is connected to a computer and communicates all relevant events with timestamps to it. The computer runs a Python program that reads the communication over the serial comport and persists the received data. The same program is used to recognize swiping gestures and clicks for the *Direction Swiping Controller* and was written for the purpose of this study.

A Python program is used to persist measured data.

3.1.5 Button Recognition

In the beginning, we planned to use an Arduino Uno for input recognition and controlling the targets. Later on, we used an Arduino Mega 2560 instead due to storage capacity issues, but at the time of planning, this was not intended. Therefore we designed the button recognition to work with the limitations of an Arduino Uno.

As the Arduino Uno only has 14 digital I/O pins, it is not possible to use these to recognize when which button is pressed. The minimum requirement for the Arduino was to recognize 32 buttons at once as this is the number of buttons on the *Sun Controller*. For that reason, we decided to use analog pins for input recognition. Multiple buttons can be recognized by only one analog pin. This is done by connecting each button to the pin with different resistances between button and pin. This way, the analog value that can be read on the Arduino is different for each button that is pressed. With this method, only one pressed button can be recognized at a time, as the read value always corresponds to the button with the lowest resistance between button and pin. This does not lead to any problems for the planned prototypes, as no controller requires multiple buttons to be

Multiple buttons are recognized on one analog pin by connecting each button with different resistances to the Arduino.

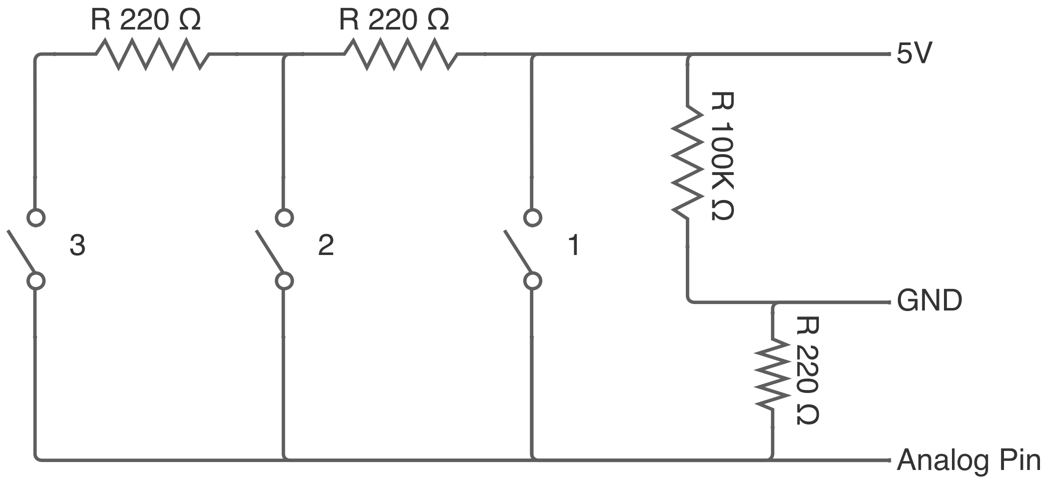


Figure 3.8: Exemplary button recognition circuit with three buttons. When button 1 is pressed, the analog pin is connected to 5V without resistance. For button 2, the resistance is 220 Ω. For button 3, it is 440 Ω. When no button is pressed, 5V is connected to ground with 100K Ω resistance.

pressed at once.

To reduce the number of different resistors needed, we built a circuit as shown in Figure 3.8. Since all resistors are connected in series, each resistor can have the same resistance. With increasing resistors, the difference in voltage decreases for each additional resistor. This results in a decreased accuracy for recognizing buttons that are connected to the end of the circuit. We experimented with how many buttons we could identify confidently and our tests showed, that we could differentiate about 14 buttons. Therefore, each button-based controller was split into three separate circuits, requiring three analog I/O pins on the Arduino.

3.2 Exemplary Living Room

3.2.1 Requirements

Participants will be required to test the controllers in an en-

Targets need to be marked and controlled by a computer.

vironment similar to modern living rooms. In the room, targets should be distributed, which participants should select using a controller. To show which target should be selected, it needs to be possible to mark single targets in the room. The participant should also receive feedback for their actions, allowing them to learn the mapping of the controller. To automatize the process, targets need to be controllable by a computer. This also increases consistency between studies. As the mappings only rely on spatial relations, it is not necessary to use real devices in the room. Only spatial information is required.

Targets should be arranged in common patterns and need to cover edge cases of the proposed controllers.

The room used in the user study should include target patterns, that are commonly observed in modern living rooms. It should also cover problematic target patterns and edge cases, to test the controllers' performances in these scenarios. Targets with similar spatial positions should be included, as these seemed to be especially problematic to distinguish on controllers. Therefore, targets that are in the same place, but at different heights should be included, as well as targets, that are positioned in the same direction of the user. At the same time, targets should be distributed in a balanced manner, so approximately equal amounts of targets should be in the front, back, left, and right of the room.

3.2.2 Targets

WS2812B LED strips are used as targets.

As already mentioned before, targets should be distributed in the room, which need to be marked and controlled by some sort of computer. For this we decided to use LED strips, as they can be controlled by an Arduino easily, using the Adafruit NeoPixel library. As we already use an Arduino for data acquisition and reading user input, the same Arduino can be used to control the LEDs. To reduce the number of wires required, multiple LED strips will be connected in series. Therefore, LEDs need to be addressable individually, as otherwise, every LED strip that is connected in series would always light up at the same time. For this reason, we decided to use WS2812B LED strips. These only need to be connected to one digital I/O pin, 5V supply

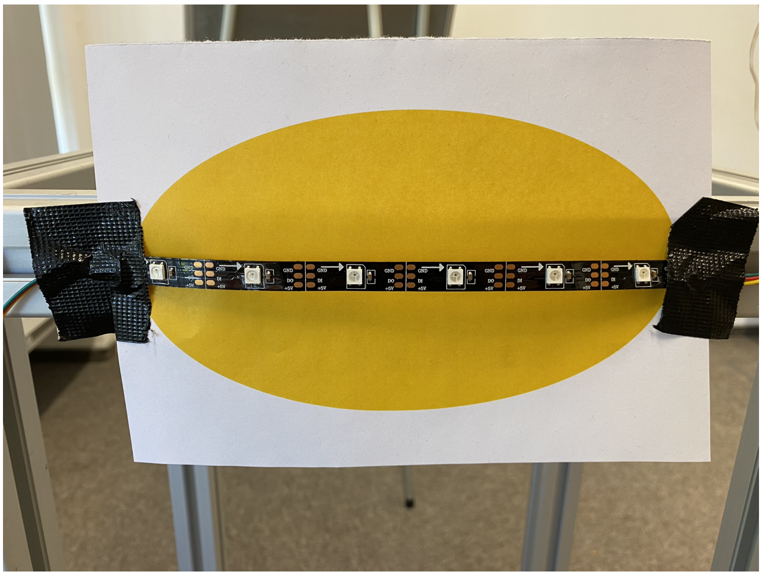


Figure 3.9: Targets as they were distributed in the study room. A LED strip is taped on a paper with a yellow background to make the target recognizable even if it is switched off.

voltage, and to the ground. Additionally, they can be cut and then soldered back together, allowing to connect multiple small LED strips with wires in series. All this makes them easy to connect to the Arduino, requires few digital I/O pins, and enables us to distribute the LED strips freely in the room.

When selecting a target on a controller, for most controllers it is necessary to know the position of other targets nearby. For example, for the *Sun Controller*, the user must know what other targets are in the same direction as the target, to know which of the three buttons in one direction needs to be pressed. Therefore, it must be easy to see targets that are not active at the time, to make orientation as easy as possible. To make targets more recognizable, we decided to put a paper with a yellow oval behind each target. This should help to find the targets easily, as the yellow colour sticks out in the room. As result, the targets look like in Figure 3.9

Paper with a yellow oval is placed behind the targets to increase their visibility.

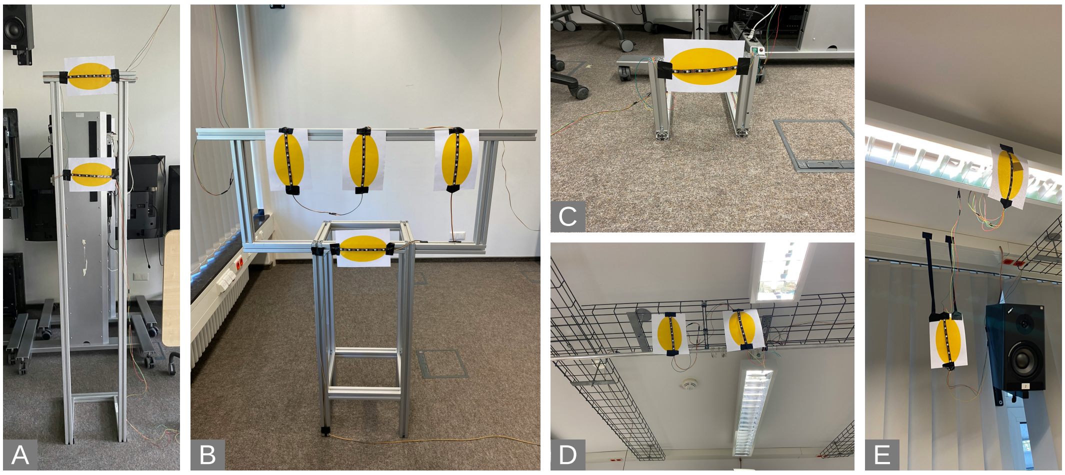


Figure 3.10: Aluminum extrusion constructions were used to distribute targets across the room. Other targets were taped to the ceiling. Construction A represents a floor lamp with two light bulbs, B represents a lamp with three bulbs above a table that has a controllable device placed on top. Construction C represents an LED strip near the ground, D a ceiling lamp with two light bulbs, and E represent blinds in the back with a ceiling light in front.

We used aluminum extrusion profiles to distribute the targets across the room.

As the targets need to be distributed across the room at different heights, we needed constructions in different sizes we could attach the targets on. We decided to use aluminum extrusion profiles, as these allowed us to freely create structures of variable form. The targets were distributed by taping them on these aluminum extrusion constructions, on tables, or on a metal mesh that is mounted under the ceiling, as seen in Figure 3.10.

3.2.3 Room Design

To design the room for the study, we decided to construct an exemplary living room while considering the requirements defined in Section 3.2.1. For planning the room we used the 3D room planner planoplan[1]. We included patterns that can regularly be found in living rooms, like a TV with a soundbar in front of it and a light source on the ground which can be found in the top right of Figure 3.11. While keeping a somewhat realistic setting in mind, we also

[1]`https://planoplan.com/en/` (Accessed: June 2, 2022)

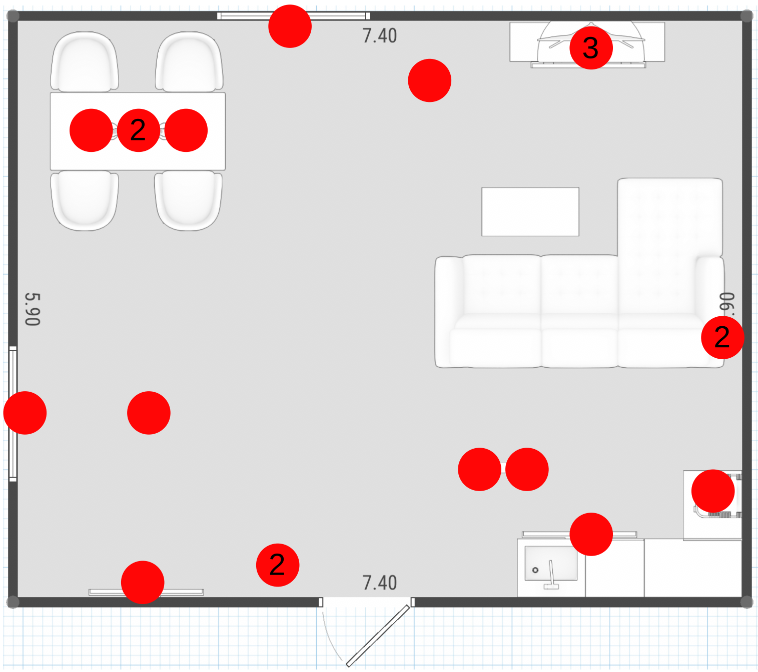


Figure 3.11: Plan of the exemplary living room. Targets are marked with red dots. When multiple targets are placed on top of each other, the number of targets is given in the circle. In total, 20 targets were distributed in the room.

included target patterns that test problematic cases on the proposed controllers. For example, a window with blinds is placed behind a ceiling light (cf. Figure 3.10 E), so that the target further away from the user is placed lower than the closer target, especially testing the intuitiveness of the *Sun Controller* in such cases

Chapter 4

User Study

We conducted a user study to investigate the effect of different mapping techniques on performance and user satisfaction. We want to explore which properties of a spatial mapping help understanding it quickly and intuitively. The possible properties of a mapping are for example the abstraction level or if it is absolute or relative. Furthermore, we want to investigate if external parameters, like the user's position in the room or the visibility of the controller, influence which mapping techniques are favourable. In the user study, participants were asked to select targets inside an exemplary living room on different controllers.

4.1 Independent Variables

4.1.1 Controllers

We investigated five controllers using different mapping techniques.

To investigate the effect of different mappings and their properties, five different controllers were used in the study. The controllers use different spatial mapping techniques to map targets in the room to user input. They were constructed and designed as described in Section 3.1. The following controllers were used in the user study:

- The *Pillar Map* is a miniature model of the room which uses an absolute spatial mapping where devices are represented as buttons on pillars.
- The *Zelda Map* is a top-down view of the room split into three levels. Each level corresponds to a height level and devices are represented as red dots on the controller which can be pressed. The spatial mapping is absolute.
- *Cluster Map* follows a similar approach as it also provides a top-down view of the room, but does not split the room into different levels. Instead, devices that would be on the same spot on the map, as their position only differs in their height, are placed below each other and surrounded by a box. The mapping is again absolute.
- The *Sun Controller* is the first relative mapping. The controller is separated into twelve directions with three buttons in each direction that are used to select a device.
- Lastly, the *Direction Swiping Controller* is a controller that works by swiping in the direction of the device that should be selected.

4.1.2 Visibility

Controllers were tested with vision and eyes-free.

A cardboard box was placed over the controller to force eyes-free interaction.

In the real world, the controller is not always visible to the user, due to dim lighting or because it is placed outside the field of view of the user. Therefore, we want to investigate eyes-free interaction with each controller. Each controller will be tested in both scenarios, when having vision on it and when participants cannot see the controller. For tries with eyes-free interaction, a cardboard box is placed over the controller to prevent the participants from looking at it. We decided on a cardboard box, as it allowed the controller and participant to stay in the same position to avoid effects that could be caused by other changes of the setup, like sitting not directly in front of the controller. In future, tries where the user could not see the controller, as a cardboard box was placed above it, will be referred to as *eyes-free*. Tries

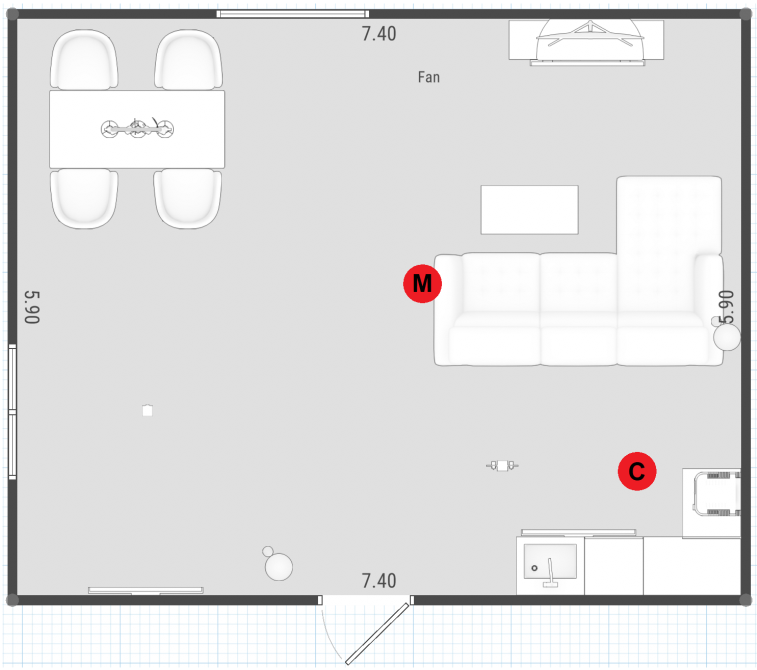


Figure 4.1: User positions in the room during the user study. Participants were placed in the middle of the room (M) or in the corner (C).

without the cardboard, where participants were able to see the controller will be referred to as *with vision*.

4.1.3 User Position

The study was conducted on two positions in the room.

As the relative spatial mappings change depending on the user's position in the room, but the absolute mappings do not, we want to investigate the effect of the user's position in a room. We decided to use two different positions. The first position is in the *middle* of the room, marked with "M" in Figure 4.1. This means, that targets are distributed relatively equally around the user. Additionally, the position of a target can be described as the same relative to the room as relative to the user. So if a light is in the back right of the room, it is also to the back right of the user. The other position used during the study was on the back right *corner* of the room and is marked with "C" in Figure 4.1. In that position, targets are heavily distributed to the front and left of the user. Also, devices that are in the back right of the room might now be to the left of the user.

4.1.4 Target Position

We distinguish between targets in the front and in the back of the participant.

The next interesting factor is the effect of the target's position on its ease of discovery on the controller. To compare different user positions, the targets' position relative to the user is the decisive factor. In total, 20 targets will be distributed in the room. The position of each target will not be changed during the study, we only investigate the difference between different targets. Especially, if a target is placed to the front or behind the user is relevant. As turning around changes the perspective of the user this may result in a higher difficulty of finding the target on the controller. Therefore we will distinguish between targets placed in *front* of participants and targets placed to the *back* of participants when evaluating the results.

4.1.5 Controller Experience

Lastly, we wanted to explore how experience with the controller changes the performance. Some controllers could be easy to use at first, but not increase in their performance even after multiple tries. Others may be hard to use at first, but easier, the more experience the participant has. Therefore, controllers will be tested twice with vision, once *without experience* and once *with experience*, to investigate the learning effect on the controller.

4.2 Experimental Design

Combining all factors, a total of 60 (5 controllers x 2 target positions x 3 tries (2 visible + 1 eyes-free) x 2 user positions) conditions were tested. During a first pilot study, testing only in one user position took about 80 minutes.

Testing the second user position would have taken roughly the same amount of time, which we identified as taking too long. We, therefore, decided to make the user position a between-subjects factor to reduce the duration of the study. All other factors were tested in a within-subjects design. A second pilot study confirmed the duration for testing one user position to be between 60 and 90 minutes. As result, each participant tested 30 conditions, acquiring a total of 300 (5 controllers x 20 targets, 3 tries) data points per participant.

The user position was tested as a between-subjects factor, the other factors as within-subjects.

Additionally, participants were asked to fill out a questionnaire for each controller used. Depending on the controller, the questionnaire contained 13 to 15 questions with a five-point Likert scale. The questionnaire for the *Cluster Map* contained one additional question, while the questionnaire for the *Direction Swiping Controller* left out a question on how easy it was to feel buttons, as the controller does not have buttons. This results in 70 additional data points per participant. Furthermore, the participants could provide free text answers to three additional questions per controller. The questionnaire can be found in Appendix A "User Study Questionnaire".

Participants filled a questionnaire with 13 to 15 questions per controller.

To reduce order effects, the order in which participants tested the controllers was counterbalanced using a balanced Latin square (cf. Appendix B "Latin Square"). The order in which targets were to be selected during an experiment was randomized. When distributing the participants onto the different room positions, we tried to roughly balance the average age, gender, and handedness over both conditions.

4.3 Participants

16 persons (6 male, 9 female, 1 other), between 23 and 59 years old (M = 29.25, SD = 10.16), participated in the user study. 15 participants were right-handed and one participant was left-handed. The user study was conducted in a room at the university to which each participant had to come to.

Figure 4.2: Room of the user study from the perspective of a participant sitting in the corner of the room.

For each user position in the room, eight participants conducted the user study. For the central position, the participants (3 male, 4 female, 1 other) were between 23 and 59 years old (M = 29.13, SD = 11.45), for the the position in the corner, participants (3 male, 5 female, 0 other) were aged between 23 and 54 (M = 29.38, SD = 9.71).

Participants were asked to either bring a COVID-19 test certificate or take a corona self-test at the chair. The conductor did the same, to reduce the likelihood of a COVID-19 infection.

4.4 Apparatus

During the user study, the controllers defined before were used for participants to make selections. One additional button was placed in front of the controller. This button was used for setting a starting point for time measurement and will be referred to as *the homing button*.

WS2812B LED strips were used as targets in the room. To recognize and handle user input on the controllers, an Arduino Mega 2560 was used. The same Arduino also controlled the LED strips and measured the accuracy and targeting time of participants. To persist the data and send commands to start and stop the experiment, an Apple MacBook Air M1 (Late 2020) was connected to the Arduino.

The room was set up according to the plan made in Section 3.2.3. A place for the conductor was positioned in the front right of the room, as can be seen in Figure 4.2. There, the MacBook was placed, which was used to control the Arduino and the experiment. This place will be referred to as the *conductor desk* in the future.

At one of the user positions as defined in Section 4.1.3, a desk was placed which will be called the *experiment desk* from now on. On the experiment desk, the homing button and one of the controllers were placed. During the experiment, the users were sitting centrally behind this desk and operating the controller. Next to the experiment desk, a dimmable ring light was placed. It was positioned so that it faced away from the desk to not blind the participant when turned on.

An additional desk was positioned to the left of the room. This desk was used by participants to fill out forms and questionnaires. This allowed the experiment conductor to prepare the next part of the study and exchange controllers on the experiment desk while keeping distance from the participant. Following this, we will refer to this desk as the *document desk*.

4.5 Task

Participants had to select marked targets in the room.

In the study, participants were asked to select marked targets in the room using their controller as accurately and fast as possible. A run consisted of targeting all 20 targets in the room exactly once. When the run started, the first target was marked immediately. Only one target in the room was marked at a time and participants had one try to hit the target. A target was marked by lighting up the corresponding LED strip in blue.

Participants had to look for the target, press the homing button, and then select the target on the controller.

The participants searched for the target in the room. Once they found it, they should press the homing button placed in front of them. Only after that, they should look at the controller and make the input to select the target. They were specifically entreated to not think about what input is required to select the target before pressing the homing button. All these actions should be done, only using the participants' strong hands. Especially for the eyes-free interaction, this is important, as it assures, that participants need to find the correct user input from the same starting position, increasing the internal validity. After the participant made a selection on the controller, the target that was actually selected by the participant, no matter if the selection was correct or wrong, was lid up in green. Immediately after they made the selection, the next target was marked automatically. After all 20 targets have been targeted exactly once, all targets lid up in green to signal the end of the run.

4.6 Study Procedure

Before the participant arrived, the room was prepared by the conductor. The experiment desk was placed at the position assigned to the participant and the controller that should be tested first according to the balanced Latin square was put on it. Additionally, the blinds of the room were closed, to reduce external lighting.

At the beginning of the study, participants were welcomed and asked to disinfect their hands. Subsequently, they were offered snacks on the document desk and asked to fill out a traceability form, due to the still ongoing COVID-19 pandemic. After that, the conductor explained the purpose of the study, what data is gathered, and how this data is processed and anonymized. Following, the general study procedure and setup were explained. Participants were offered to take breaks after every run if they wanted to. This included, that they will be using five different controllers to select targets in the room and that targets are represented by LED strips. The participants were then asked to sign an informed consent form.

Before starting the experiment, participants were asked to fill out traceability and informed consent forms.

Afterwards, the participants were led to the experiment desk and asked to take a seat behind the desk. The constructor requested them to take a look around and make an overview of the targets distributed in the room. If the participants missed targets, they were made aware of them. Thereafter, the exact experiment procedure was explained, as described in the previous section. When no questions regarding the task were left open, the participants were introduced to the first controller.

Each controller was first only explained on a schematic level. This meant, that the mapping technique was explained, but no concrete examples using the study room were given. The participants then were given the opportunity to ask questions regarding the mapping. Questions that asked which button belongs to which target were not answered. This was done to ensure that participants had to follow their intuition in the first run. When all questions regarding the mapping were answered and the participants felt ready, the conductor went to the *conductor desk* and the first run was started.

Before the first run, the controllers were only explained on a schematic level.

During the experiment, the ceiling lighting of the room was turned off. Instead, the ring light was turned on. This darkened the room, making it easier to find the luminous target. Before the first try on the first controller, participants were asked for their preferred light intensity, so that they could comfortably see the controller.

Lighting in the room was reduced to make the targets easier to find.

After the first run, participants could now ask specific ques-

Participants could practice with the controller after the first run.

Figure 4.3: A look on the experiment desk from the participants' perspective for the eyes-free setup. The largest controller, the *Zelda Map* is placed below the cardboard box.

tions about the controller. They were also offered to practice with the controller as long as they wanted to. When they felt ready and had no more questions, we began with the second run.

A third run was done eyes-free.

For the third run, a cardboard box was placed on top of the controller. Participants were additionally asked to not try to look under the box in case it was possible. A view from the user's perspective on the eyes-free setup can be seen in Figure 4.3. They then again had time to practice with the controller, this time interacting eyes-free. After the participants signalled that they were ready, the third run was started.

Lastly, participants were asked to return to the document desk, where they were handed a questionnaire about the controller they just used. The conductor offered to answer questions if there were any uncertainties or difficulties in understanding the questionnaire. In the meantime, the experiment desk was prepared for the next controller to reduce waiting times.

This procedure was repeated for all five controllers. After the last controller, the participants were handed an additional questionnaire which included summarizing questions and asked for demographic information. Subsequently, the conductor again offered snacks to the participants.

4.7 Measurements

Targeting time and accuracy were measured by the Arduino.

As already mentioned, the measurements were done using an Arduino Mega 2560. During the experiments, we measured targeting times and accuracy. The targeting time was measured from the moment the homing button was pressed, until the moment a selection was made on the controller. This means that the time used to find a target was not included in the targeting time. For the *Sun Controller* and the *Direction Swiping Controller*, there were cases in which the input needed to be confirmed by a timeout of one second. In such cases, the timeout was subtracted from the targeting time.

The targeting time was measured between pressing the home button and making a selection on the controller.

4.8 Results

In the following section, *Cluster Map* will be referred to as *Cluster*, *Pillar Map* as *Pillars*, *Zelda Map* as *Zelda*, *Sun Controller* as *Sun*, and *Direction Swiping Controller* as *Swipe*.

4.8.1 Targeting Time

Controller, *visibility*, and *controller* $\times$ *visibility* have a significant effect on the targeting time.

Firstly, we analyzed the *targeting time* on the factors *user position*, *controller*, and *visibility*. The data is log-normal distributed, so we used a mixed-design ANOVA on the log-transformed *targeting time*. We found significant main effects of *visibility* ($F_{1,14} = 184.05, p < 0.001$) and *controller* ($F_{4,56} = 25.79, p < 0.001$) on the *targeting time*. We also found a significant interaction between *visibility* and *controller* ($F_{4,56} = 62.31, p < 0.001$).

After that, we analyzed the *targeting time* on the factors *user position*, *controller*, and *target position*. With mixed-design ANOVA on the log-transformed *targeting time*, we found an additional main effect of *target position* ($F_{1,15} = 22.56, p < 0.001$) and could confirm the main effect of the *controller*.

Lastly, we analyzed the effects of the factors *experience* and *controller* on the *targeting time*. A factorial repeated measures ANOVA, indicated an additional statistically significant effect on the *targeting time* of *experience* ($F_{1,14} = 104.46, p < 0.001$). There was no interaction between *experience* and *controller*.

Visibility

Interaction *with vision* is significantly faster than *eyes-free*.

The post-hoc test, a paired-samples t-test, indicated that *with vision* vs. *eyes-free* showed a statistically significant difference ($p < 0.001$) with *eyes-free* interaction (mean $(M) = 2.84s$, geometric confidence interval $(gCI) = [2.48s, 2.88s]$) being slower than interaction *with vision* ($M = 1.75s$, $gCI = [1.44s, 1.75s]$).

Controllers

The post-hoc test, paired-samples t-tests, corrected with Holm's sequential Bonferroni procedure, indicated that *Pillars* ($(M) = 1.69s$, $(gCI) = [1.44s, 1.87s]$) are significantly quicker than *Sun* ($M = 1.95s$, $gCI = [1.62s, 2.18s]$)($p < 0.05$), *Swipe* ($M = 2.68s$, $gCI = [2.32s, 2.95s]$)($p < 0.001$), and *Zelda* ($M = 2.53s$, $gCI = [2.01s, 2.84s]$)($p < 0.001$). They also indicated, that *Cluster* ($(M) = 1.71s$, $(gCI) = [1.37s, 1.93s]$) is significantly quicker than *Swipe* ($p < 0.001$) and *Zelda* ($p < 0.001$), and *Sun* is significantly quicker than *Swipe* ($p < 0.001$) and *Zelda* ($p < 0.001$). The mean targeting times and geometric confidence intervals are depicted in Figure 4.4.

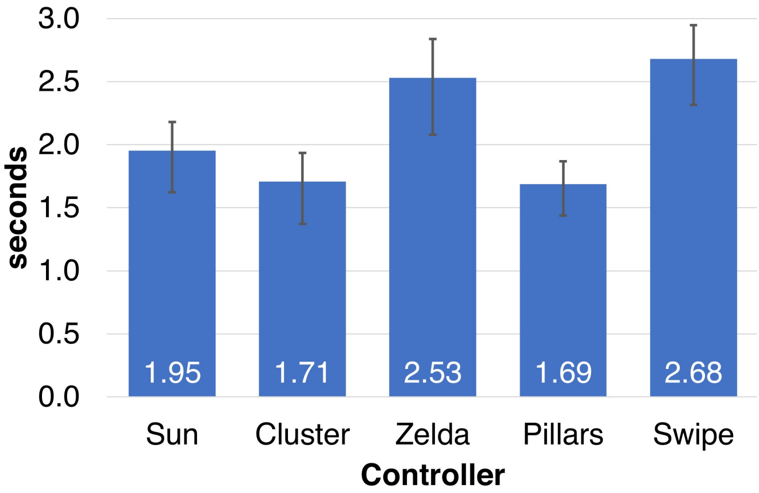


Figure 4.4: Mean targeting times for each controller. The bars show the mean targeting time, the lines show the geometric confidence interval.

Interaction Visibility × Controller

To further analyze the interaction, we fixed one factor and checked if the other factor has a significant effect. Using a one-way ANOVA we found significant main effects of *controller* when fixing the visibility to *with vision* ($F_{4,75} = 29.87, p < 0.001$), and when fixing it to *eyes-free* ($F_{4,75} = 4.61, p < 0.01$).

We fixed one factor of the interaction to check the effect of the other factor.

When fixing the visibility to *with vision*, both *Cluster* and *Pillars* are significantly faster than *Sun*, *Swipe*, and *Zelda* ($p < 0.001$ each). Furthermore, *Sun* is significantly quicker than *Swipe* ($p < 0.001$) and *Zelda* ($p < 0.05$) and *Zelda* is significantly quicker than *Swipe* ($p < 0.001$). The only comparison without significant difference is *Cluster* and *Pillars*.

With vision, *Cluster* and *Pillars* are the fastest controllers.

Eyes-free, Zelda is significantly slower than the other controllers. When the visibility is fixed to *eyes-free*, *Zelda* is significantly slower than *Cluster* ($p < 0.01$), *Pillars* ($p < 0.01$), *Sun* ($p < 0.001$), and *Swipe* ($p < 0.001$). All other comparisons do not have significant differences.

We then fixed the *controller*. Using a one-way ANOVA, we

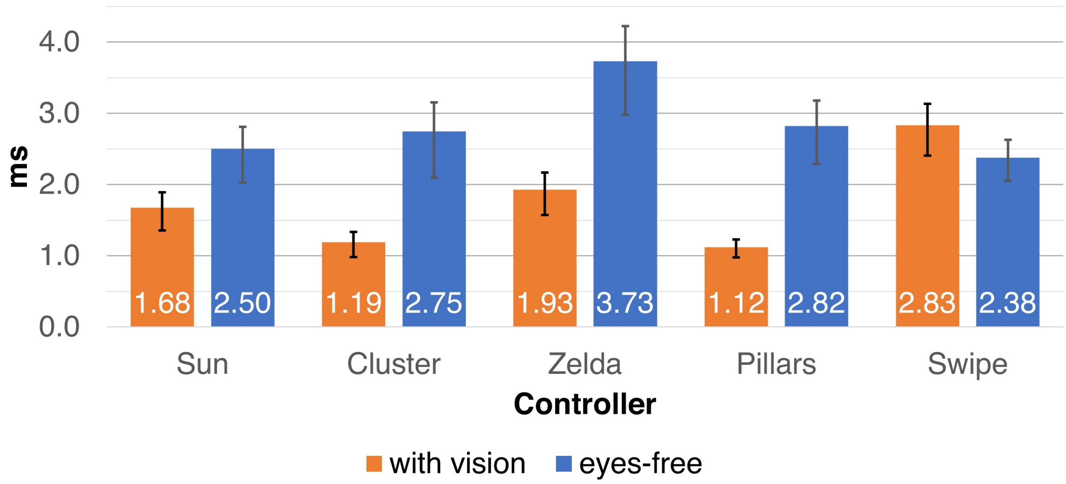


Figure 4.5: Mean targeting times for each controller depending on the visibility. The bars show the mean, the lines show the geometric confidence interval.

Eyes-free interaction is slower for all controllers but *Swipe*.

found significant main effects of *visibility* for *Sun* ($F_{1,30} = 13.31, p < 0.001$), *Cluster* ($F_{1,30} = 45.44, p < 0.001$), *Zelda* ($F_{1,30} = 34.64, p < 0.001$), and *Pillars* ($F_{1,30} = 92.02, p < 0.001$), but not for *Swipe*.

Paired-samples t-tests indicated, that for all controllers but *Swipe*, interaction *with vision* is significantly faster ($p < 0.001$) than *eyes-free* interaction. The means and geometric confidence intervals can be seen in Figure 4.5.

Target Position

Targets in the front are selected faster than targets in the back.

A paired-samples t-test indicated that targets in the *front* compared to targets in the *back* show a statistically significant difference ($p < 0.001$) with targets in the *front* ($M = 2.00s$, $gCI = [1.77s, 2.05s]$) being targeted faster than targets in the *back* ($M = 2.33s$, $gCI = [1.98s, 2.35s]$).

Experience

According to a paired-samples t-test, the runs *with experience* ($M = 1.50s$, $gCI = [1.23s, 1.50s]$) were significantly faster ($p < 0.001$) than the runs *without experience* ($M = 2.00s$, $gCI = [1.63s, 1.99s]$).

4.8.2 Accuracy

At first, we analyzed the *accuracy* of the factor *controller* and *user position*. The data is normally distributed. A mixed-design ANOVA indicated a significant main effect of the *controller* ($F_{4,56} = 26.44, p < 0.001$) on the *accuracy*. No main effect or interaction with the *user position* was indicated. The main effect of the factor *controller* was also indicated by all following analyses, but will no longer be mentioned, as it was analyzed separately. Only interactions with the factor *controller* will be mentioned.

The accuracy data is no longer normally distributed when split for *visibility* or *target position*. Therefore we did a non-parametric analysis of variance based on the Aligned Rank Transform (ART) of the factors *visibility* and *controller*. This showed a statistically significant effect of *visibility* ($F_{1,135} = 7.24, p < 0.01$) and of the interaction *visibility* $\times$ *controller* ($F_{4,135} = 3.69, p < 0.01$).

Afterwards, we did an analysis of the factors *target position* and *controller*, again using ART. It indicated a significant effect of *target position* ($F_{1,135} = 61.66, p < 0.001$). No more effects were indicated.

Subsequently, we analyzed the effect of the factors *experience* and *controller* on the *accuracy*, using ART. A significant effect of the *experience* was indicated ($F_{1,135} = 45.21, p < 0.001$). Additionally, we found a statistically significant interaction of *experience* and *controller* ($F_{4,135} = 3.74, p < 0.01$).

Lastly, we investigated further effects of the *user position*. We could not find a way to do a nonparametric analysis of between- and within-subjects factors at once. Therefore we decided to fix the *user position* and analyze the effects of the factors *visibility*, *controller*, and *target position* based on the *user position*. Which factors and interactions have a

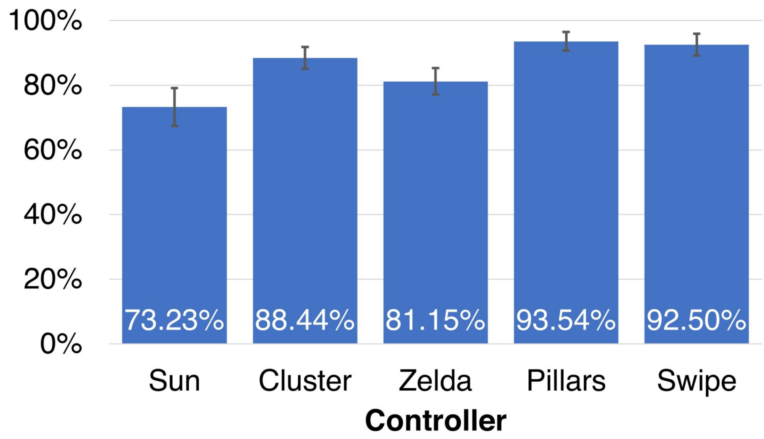


Figure 4.6: Mean accuracy for each controller. The bars show the mean, the lines show the confidence interval.

significant effect can be seen in Table 4.1. The effects of the factors were analyzed using ART.

Factor	Middle	Corner
Visibility	$F_{1,63} = 10.45, p < 0.01$	n.s.
Visibility : Controller	n.s.	$F_{4,63} = 3.28, p < 0.05$
Target Position	$F_{1,63} = 23.45, p < 0.001$	$F_{1,63} = 45.63, p < 0.001$
Target Position : Controller	n.s.	n.s.

Table 4.1: Investigated main effects when fixing the *user position* to either the middle or the corner. The p-value and F-value are given when a significant main effect was indicated, otherwise, no significant difference was indicated (n.s.).

Controller

Sun and *Zelda* had worse accuracy than the other controllers.

Post-hoc tests using paired-samples t-tests, corrected with Holm's sequential Bonferroni procedure, indicated that the *accuracy* using *Pillars* ($M = 93.54\%$, confidence interval $(CI) = [90.68\%, 96.40\%]$) is significantly higher than when using *Cluster* ($M = 88.44\%$, $CI = [85.08\%, 91.81\%]$)($p < 0.05$), *Sun* ($M = 73.23\%$, $CI = [67.33\%, 79.13\%]$)($p < 0.001$), and *Zelda* ($M = 81.15\%$, $CI = [77.05\%, 85.24\%]$)($p < 0.001$). They also indicated, that the *accuracy* using *Cluster* is significantly higher than when using *Sun* ($p < 0.001$) and *Zelda* ($p < 0.05$), and *accuracy* with *Swipe* ($M = 92.50\%$,

$CI = [89.15\%, 95.85\%]$) is significantly higher than with *Sun* ($p < 0.001$) and *Zelda* ($p < 0.01$). For the other comparisons, no significant difference was found. The mean targeting times and confidence intervals are depicted in Figure 4.6.

Visibility

For the *visibility*, a post-hoc Wilcoxon signed-rank test indicated that interaction *with vision* ($(M) = 87.13\%$, standard deviation $(SD) = 10.19\%$) is significantly more accurate ($p < 0.05$) than *eyes-free* interaction ($M = 83.06\%$, $SD = 14.85\%$).

Accuracy *with vision* was significantly higher than without.

Interaction Visibility × Controller

Similar to the procedure in Section 4.8.1, we will fix one of the factors and investigate the effect of the other factor. According to Wilcoxon signed-rank tests, the *visibility* has a statistically significant effect on the accuracy when fixing the controller to *Cluster* ($p < 0.01$), *Sun* ($p < 0.01$), and *Swipe* ($p < 0.01$). For *Cluster* and *Sun*, interaction *with vision* has higher accuracy, for *Swipe*, *eyes-free* interaction is more accurate.

Cluster and *Sun* had significantly worse accuracy when used *eyes-free*.

When fixing the *visibility*, Friedman tests indicated, that for both, interaction *with vision* and *eyes-free* interaction, the factor *controller* had a significant effect ($p < 0.001$ for both).

For interaction *with vision*, ten post-hoc Wilcoxon signed-rank tests, corrected with Holm's sequential Bonferroni procedure, indicated, that the accuracy with *Sun* is significantly lower than with *Cluster* ($p < 0.01$), *Pillars* ($p < 0.01$), *Swipe* ($p < 0.05$), and *Zelda* ($p < 0.05$). Furthermore, accuracy with *Zelda* is significantly lower than with *Cluster* ($p < 0.05$) and *Pillars* ($p < 0.05$).

With vision, *Sun* has significantly worse accuracy than the other controllers.

For *eyes-free* interaction, another ten post-hoc Wilcoxon signed-rank tests, corrected with Holm's sequential Bonferroni procedure showed a significant difference in accu-

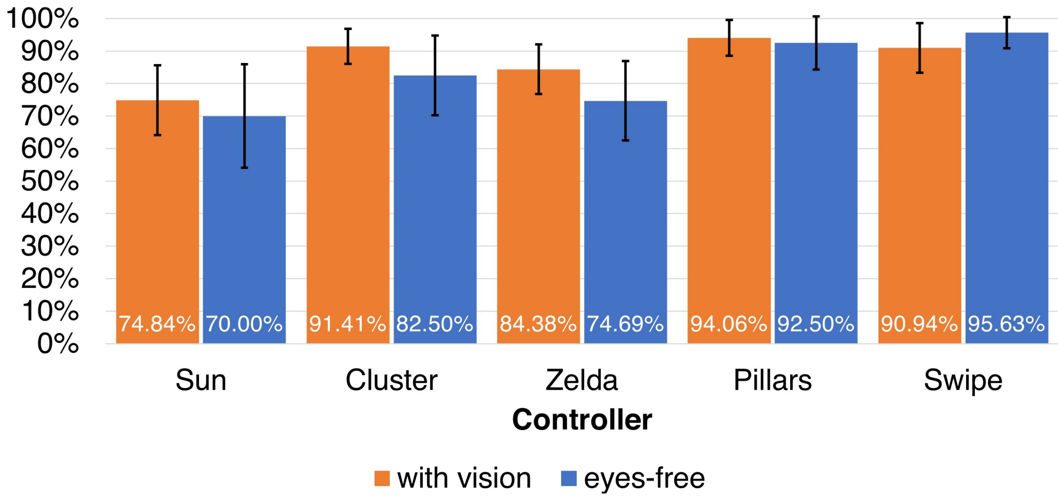


Figure 4.7: Mean accuracy for each controller dependent on the visibility. Bars show the mean, the lines show the standard deviation.

racy between *Pillars* and *Cluster* ($p < 0.05$), *Sun* ($p < 0.01$), and *Zelda* ($p < 0.05$), with *Pillars'* accuracy being higher than the others. They also indicated that the accuracy with *Swipe* is significantly higher than with *Cluster*, *Sun*, and *Zelda* ($p < 0.01$ each). Lastly, the accuracy with *Cluster* is significantly higher than with *Sun* ($p < 0.05$). The mean accuracy and their standard deviation can be seen in Figure 4.7.

Target Position

The accuracy is significantly higher for targets in the *front*.

For the *target position*, a post-hoc Wilcoxon signed-rank test indicated that the accuracy for targets in the *front* ($M = 90.36\%$, $SD = 9.45\%$) is significantly higher ($p < 0.001$) than for targets in the back *back* ($M = 78.60\%$, $SD = 16.54\%$).

Experience

A post-hoc Wilcoxon signed-rank test indicated, that *with experience* ($M = 91.75\%$, $SD = 9.00\%$), the accuracy is higher than *without experience* ($M = 82.50\%$, $SD = 13.59\%$).

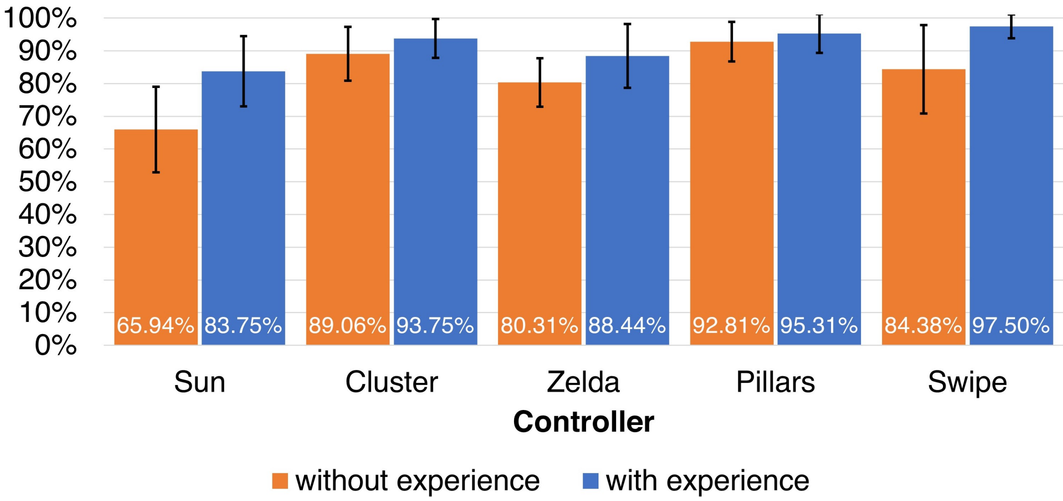


Figure 4.8: Mean accuracy for each controller dependent on the experience. Bars show the mean, the lines show the standard deviation.

Interaction Experience × Controller

To investigate the interaction, we will again fix one of the factors to test the effect of the other factor. Wilcoxon signed-rank tests indicated, that the *experience* has a statistically significant effect on the accuracy when fixing the controller to *Sun* ($p < 0.001$), *Zelda* ($p < 0.01$), *Pillars* ($p < 0.05$), and *Swipe* ($p < 0.001$). For all four controllers, the accuracy was significantly higher *with experience* than *without*.

For *Sun*, *Zelda*, *Pillars*, and *Swipe*, accuracy was significantly higher *with experience*.

When fixing the *experience*, Friedman tests indicated, that for both, interaction *without experience* and *with experience*, the *controller* had a significant effect ($p < 0.001$ for both).

Without experience, the accuracy with *Sun* is significantly lower than with *Cluster* ($p < 0.01$), *Pillars* ($p < 0.01$), *Swipe* ($p < 0.05$), and *Zelda* ($p < 0.05$), as indicated by ten post-hoc Wilcoxon signed-rank tests, corrected with Holm's sequential Bonferroni procedure. Additionally, accuracy with *Zelda* is significantly lower than with *Cluster* ($p < 0.05$) and *Pillars* ($p < 0.05$).

Sun has significantly worse accuracy *without experience* than the other controllers.

With experience, ten post-hoc Wilcoxon signed-rank tests showed, that the only significant differences are between *Cluster* and *Sun* ($p < 0.05$) and *Swipe* and *Sun* ($p < 0.05$).

Both times, *Sun* has the worse accuracy. The mean accuracy for each controller *with* and *without experience* is depicted in Figure 4.8.

User Position Middle: Visibility

In the *middle* of the room accuracy *with vision* is higher than without.

When the *user position* is fixed to the *middle*, a post-hoc Wilcoxon signed-rank test indicated, that the accuracy when interacting *with vision* ($M = 88.31\%$, $SD = 9.41\%$) is significantly higher ($p < 0.01$) than when interacting *eyes-free* ($M = 82.88\%$, $SD = 15.02\%$).

User Position Corner: Visibility × Controller

For the interaction between visibility and controller, we again fixed one of the factors, additionally to the user position that is fixed to the *corner*.

While fixing the visibility, Friedman tests indicated, that the controller has a significant effect for interaction *with vision* ($p < 0.001$) and *eyes-free* ($p < 0.001$) interaction. The post-hoc Wilcoxon signed-rank tests on the other side do not indicate any significant differences between any controllers.

When the controller is fixed, the visibility only has a significant effect on the accuracy for the controllers *Zelda* ($p < 0.05$) and *Swipe* ($p < 0.05$), as shown by Wilcoxon signed-rank tests. For *Zelda*, interaction *with vision* is more accurate, for *Swipe*, the accuracy is higher during *eyes-free* interaction.

User Position Middle: Target Position

When fixing the user position to the *middle*, a post-hoc Wilcoxon signed-rank test indicated, that the accuracy when selecting targets in the *front* ($M = 91.48\%$, $SD = 9.38\%$) is significantly higher ($p < 0.001$) than when selecting targets in the *back* ($M = 81.79\%$, $SD = 13.89\%$).

User Position Corner: Target Position

For the *user position middle*, the accuracy when selecting targets in the *front* ($M = 89.24\%$, $SD = 9.51\%$) is also significantly higher ($p < 0.001$) than when selecting targets in the *back* ($M = 75.42\%$, $SD = 18.44\%$), as indicated by a post-hoc Wilcoxon signed-rank test.

4.8.3 Questionnaire

To analyze the questionnaires, we used Friedman tests and post-hoc Wilcoxon signed-rank tests, corrected with Holm's sequential Bonferroni procedure. We analyzed the first 14 questions (Q1 - Q14) of the questionnaire and the effect the controller had on participants' answers. Q14 was not asked for *Swipe*. For all questions but Q6, the Friedman tests indicated a significant effect of the controller. For the questions Q10 and Q14, the post-hoc tests did not indicate significant differences between any controllers. The results of the post-hoc tests can be seen in Table 4.2.

For all questions but Q6, Q10, and Q14 there were significant differences between controllers.

	Cluster	Pillars	Sun	Swipe	Zelda
Cluster	-	Q4	Q2, Q3, Q5, Q7 - Q9		Q1 - Q5, Q7, Q8
Pillars		-	Q2, Q5, Q11 - Q13		Q1 - Q3, Q5, Q8, Q9, Q12, Q13
Sun		Q4	-		Q4
Swipe	Q13	Q4	Q1 - Q3, Q5, Q9, Q11 - Q13	-	Q1 - Q5, Q7, Q9, Q11 - Q13
Zelda					-

Table 4.2: Significant differences between the answers to the questionnaire questions depending on the controller. In each row, the questions are given for which the controller was ranked significantly better than the controller in the column. For example, *Cluster* was rated significantly better than *Pillars* for Q4.

4.8.4 Ranking

The rankings of the controllers can be seen in Figure 4.9. In general, the controllers *Zelda* and *Sun* were ranked worse

Swipe was the most preferred controller

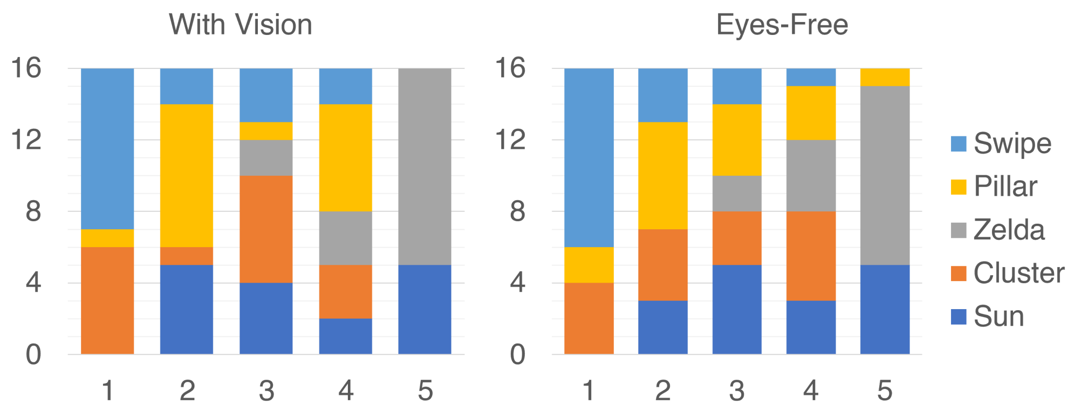


Figure 4.9: Ranking of the controllers from 1 = best to 5 = worst. The ranking was split in interaction with vision and eyes-free interaction.

than the other controllers, both were never ranked as the best controller, while *Zelda* was also not ranked as the second best controller. Eleven out of 16 participants ranked *Zelda* as the worst controller with vision, ten ranked it as the worst when interacting *eyes-free*. All but two participants ranked *Cluster* better than *Zelda* for both, interaction with vision and eyes-free interaction. *Cluster* was never ranked as the worst controller. For *eyes-free* interaction, it was almost equally ranked between ranks one and four, *with vision* it got ranked first and third most often. The controller that was ranked first most often is *Swipe*. It also never got ranked as the worst controller.

4.9 Discussion

Lower abstraction levels help improve the targeting times of a controller.

Lower abstraction levels seem to decrease the targeting time of the controllers. This effect is the strongest when having vision to the controller, which is indicated by *Pillars* and *Cluster* being the fastest controllers when interacting with vision, being significantly faster than all other controllers. Although *Zelda* has a similar abstraction level to *Cluster*, it had worse targeting times. This can be explained by the problems of participants in understanding the mapping and how the controller worked, as indicated by the questionnaire results. Multiple participants said, that they found the mapping of *Zelda* complex and had to think much about where the target is located on the controller. They also had difficulties reaching every part of the controller, especially the buttons furthest away from the user.

For eyes-free interaction, the abstraction level does not seem to have a strong effect on the targeting time. Besides *Zelda* being significantly slower than the other controllers, there were no other significant differences. The additional time with *Zelda* can be explained with the same reasoning as for the interaction with vision.

Participants had difficulties understanding the mappings of *Zelda* and *Sun*.

For accuracy, no clear conclusion related to the detail level can be made. In general, *Pillars*, *Swipe*, and *Cluster* performed significantly better than the other two controllers. For both, *Zelda* and *Sun*, participants had significantly more difficulties in understanding which input was mapped to which target than with the other controllers, as indicated by Q5 of the questionnaire.

Cluster is easy to understand from the start.

Without experience, *Sun* had significantly worse accuracy than all other controllers, indicating, that the mapping is especially hard to understand in the beginning. While the effect of experience does not seem to differ between controllers for the targeting time, it does for the accuracy. All controllers but *Cluster* have significantly better accuracy with experience on the controller. As the accuracy of *Cluster* is already at 89.05% for the first try, the mapping is easy to understand from the start.

Swipe does not rely on visual information, increasing the accuracy interacting *eyes-free*.

Without vision, *Swipe* and *Pillars* are the most accurate. The advantage of *Swipe* is, that it does not rely on visual or tactile information on the controller. A participant said after the study, that they did not look at the controller even when they were able to. This can also explain the significantly increased accuracy of *Swipe* when used eyes-free compared to with vision. Most of the mistakes with *Swipe* were caused by problems with the input technique rather than problems with the mapping. As the run without vision was done last, participants were familiar with the controller and the factor of losing vision to the controller does not seem to have an effect on the accuracy. Overall, *Swipe* was the least affected by the loss of vision, as there is no significant effect of visibility on the targeting time. *Pillars* probably have a high accuracy *eyes-free*, as the three-dimensional map helps orientating easily, which was also discovered by Holloway et al. [2018].

There was no indication of relative or absolute mappings performing better.

If the mapping is absolute or relative does not seem to have an effect on the mapping's performance. Regarding the accuracy, *Swipe* is among the better-performing controllers and *Sun* is among the worst-performing. For the targeting time, it is vice versa. While some participants sitting in the corner were not immediately sure if the mappings were relative to them or the centre of the room, there is no indication of this problem in the measured data.

The personal preferences of the participants are in favour of *Swipe*. The main reason given was that they had fun using it and could easily imagine how it would be integrated into their home. The latter was also the main problem participants had with *Pillars*. While some participants said it could be a cool gadget, most others said, they would not want to have such a large controller in their living room.

We recommend *Swipe* and *Pillars* for *eyes-free* interaction.

We would recommend choosing the mapping technique based on the scenario it is planned for. In scenarios where the controller is often used eyes-free, *Swipe* and *Pillars* are probably the easiest to use. They both achieve high accuracy and low targeting times without vision, while *Pillars* additionally offer significantly better targeting times with vision. The advantage of *Swipe* is, that it can be integrated into the environment seamlessly, as it does not require visual elements. Although the results do not indicate effects of the user position, attention should be paid when positioning the controller in a corner, as targeting times could suffer from increasing targets per direction. For scenarios in which the controller is mostly visible, we suggest *Cluster* and *Pillars*. The high detail levels allow finding targets quickly on both controllers.

For interaction *with vision* we recommend *Cluster* and *Pillars*.

4.10 Limitations

During the experiments, it sometimes happened, that the user's selection was wrongly classified as target missed. This happened especially for the *Sun Controller* when multiple targets were mapped to a single button, as the button recognition logic is volatile to voltage fluctuations in these cases. When this happened, we reacted depending on the run that was currently performed. The second and eyes-free run was restarted and repeated, as the participants already had the opportunity to practice with the controller. If the error happened in the first run, the run was continued. Afterwards, we corrected the data manually, which was possible for all occasions.

After running the experiment with participant six, we found, that during the first run for the *Zelda Map*, the measurement logs have a gap of about 15 seconds in which no data was saved. This error only occurred once and we could not find the source of the problem. As result, we are missing information for two targets in this run that could not be replicated. These two measurements are therefore not considered for the calculation of the accuracy and targeting time.

When evaluating the questionnaire, we noticed that the questionnaire of participant eight did not include answers for Q12 for the *Sun Controller* and Q9 for the *Pillar Map*. As consequence, we removed all answers to Q9 and Q12 by participant eight for the evaluation.

Chapter 5

Summary and Future Work

5.1 Summary and Contributions

In this thesis, we evaluated mapping techniques to map devices in a real-world environment to a controller. The purpose of the study was to investigate which properties of a mapping make it easy to understand and make the interaction with it fast and accurate.

For this, we constructed five different controllers that make use of different mapping techniques. We then conducted a user study with 16 participants to compare the controllers on their targeting time and accuracy. The study was held in an exemplary living room, that was specifically constructed for the study. The living room contains common patterns and edge cases that should test the controllers in diverse scenarios. Participants of the study used the controllers to select targets distributed in the room. We measured the targeting times and the accuracy of participants and analyzed the results to figure out which mapping techniques excel in which scenarios. Therefore, we tested the controllers with and without vision, on targets in different positions in the room, and on different user positions in the room.

The results showed, that lower abstraction levels help improve targeting times when the controller is visible, as *Cluster* and *Pillars* were significantly faster than the other controllers for interaction with vision. For eyes-free interaction, there was no indication that the abstraction level has an effect on the targeting time. The accuracy was the highest for the controllers *Pillars*, *Swipe*, and *Cluster*. For the other controllers, participants had difficulties in understanding which input is mapped to which target. Whether a mapping is absolute or relative to the user's position did not indicate to have an effect on the targeting time and accuracy.

Concluding we can recommend mapping techniques with low abstraction, like *Cluster* and *Pillars* for scenarios where eyes-free interaction is rare. If eyes-free interaction is required, we propose to use mapping techniques like *Swipe*, which do not rely on visual and tactile information, or *Pillars*, which have tactile elements that are easy to recognize.

5.2 Future Work

In the future, we would like to do a follow-up study to investigate the effect of the orientation of a controller. In our user study, all controllers were orientated horizontally. For some scenarios, it might be desirable to place the controller vertically, for example when placing it on the side of a couch or seat. When doing so, directions on the controller do not longer correspond directly to the directions in the room. Therefore, different types of controllers might be desirable in that case.

On the other side, we want to investigate how the less abstract controllers can be designed to make them adjustable to changes in the room. While there are already methods to create dynamically changing textile interfaces, the specific design for the controllers still needs to be figured out.

The mappings used during our user study only rely on spatial information. In following studies, we want to investigate how to design other mapping techniques that also use other information. For example, tactile icons that represent the devices could be used. With these, controllers like *Cluster*, *Zelda*, or *Pillars* could be enhanced. As investigated by Holloway et al. [2018], three-dimensional icons help orientating on a tactile map and could therefore increase targeting time and accuracy for eyes-free interaction. We also want to explore mapping techniques that rely on other information, for example by classifying devices based on their device type.

Lastly, we want to investigate how a target selection controller can be integrated into a larger controller, that is also able to control the selected devices. For this, it should be explored how single input elements, like buttons and sliders, should be combined to create a complex controller.

Appendix A

User Study Questionnaire

ID:___

Sun Mapping

	Fully disagree				Fully agree
The way the controller worked was easy to understand.	○	○	○	○	○
The controller was easy to use from the start.	○	○	○	○	○
Overall, the controller was easy to use.	○	○	○	○	○
I could reach all parts of the controller easily.	○	○	○	○	○
The mapping of the button to the corresponding target light was easy to understand.	○	○	○	○	○
When the controller was visible:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
When using the controller eyes-free:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
I could recognize the buttons well.	○	○	○	○	○

What did you like about the controller?

Figure A.1: First page of the questionnaire for the *Sun Controller*.

ID:___

What did you not like about the controller?

What would you do to enhance the controller?

Additional comments

Figure A.2: Second page of the questionnaire for the *Sun Controller*.

ID:___

Zelda Mapping

	Fully disagree				Fully agree
The way the controller worked was easy to understand.	○	○	○	○	○
The controller was easy to use from the start.	○	○	○	○	○
Overall, the controller was easy to use.	○	○	○	○	○
I could reach all parts of the controller easily.	○	○	○	○	○
The mapping of the button to the corresponding target light was easy to understand.	○	○	○	○	○
When the controller was visible:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
When using the controller eyes-free:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
I could recognize the buttons well.	○	○	○	○	○

What did you like about the controller?

Figure A.3: First page of the questionnaire for the *Zelda Map*.

ID:___

What did you not like about the controller?

What would you do to enhance the controller?

Additional comments

Figure A.4: Second page of the questionnaire for the *Zelda Map*.

ID:___

Pillar Map

	Fully disagree				Fully agree
The way the controller worked was easy to understand.	◯	◯	◯	◯	◯
The controller was easy to use from the start.	◯	◯	◯	◯	◯
Overall, the controller was easy to use.	◯	◯	◯	◯	◯
I could reach all parts of the controller easily.	◯	◯	◯	◯	◯
The mapping of the button to the corresponding target light was easy to understand.	◯	◯	◯	◯	◯
When the controller was visible:					
All elements of the controller were clear to me.	◯	◯	◯	◯	◯
Orientating on the controller was easy.	◯	◯	◯	◯	◯
I could select the target light quickly.	◯	◯	◯	◯	◯
I could select the correct target light confidently.	◯	◯	◯	◯	◯
When using the controller eyes-free:					
All elements of the controller were clear to me.	◯	◯	◯	◯	◯
Orientating on the controller was easy.	◯	◯	◯	◯	◯
I could select the target light quickly.	◯	◯	◯	◯	◯
I could select the correct target light confidently.	◯	◯	◯	◯	◯
I could recognize the buttons well.	◯	◯	◯	◯	◯

What did you like about the controller?

Figure A.5: First page of the questionnaire for the *Pillar Map*.

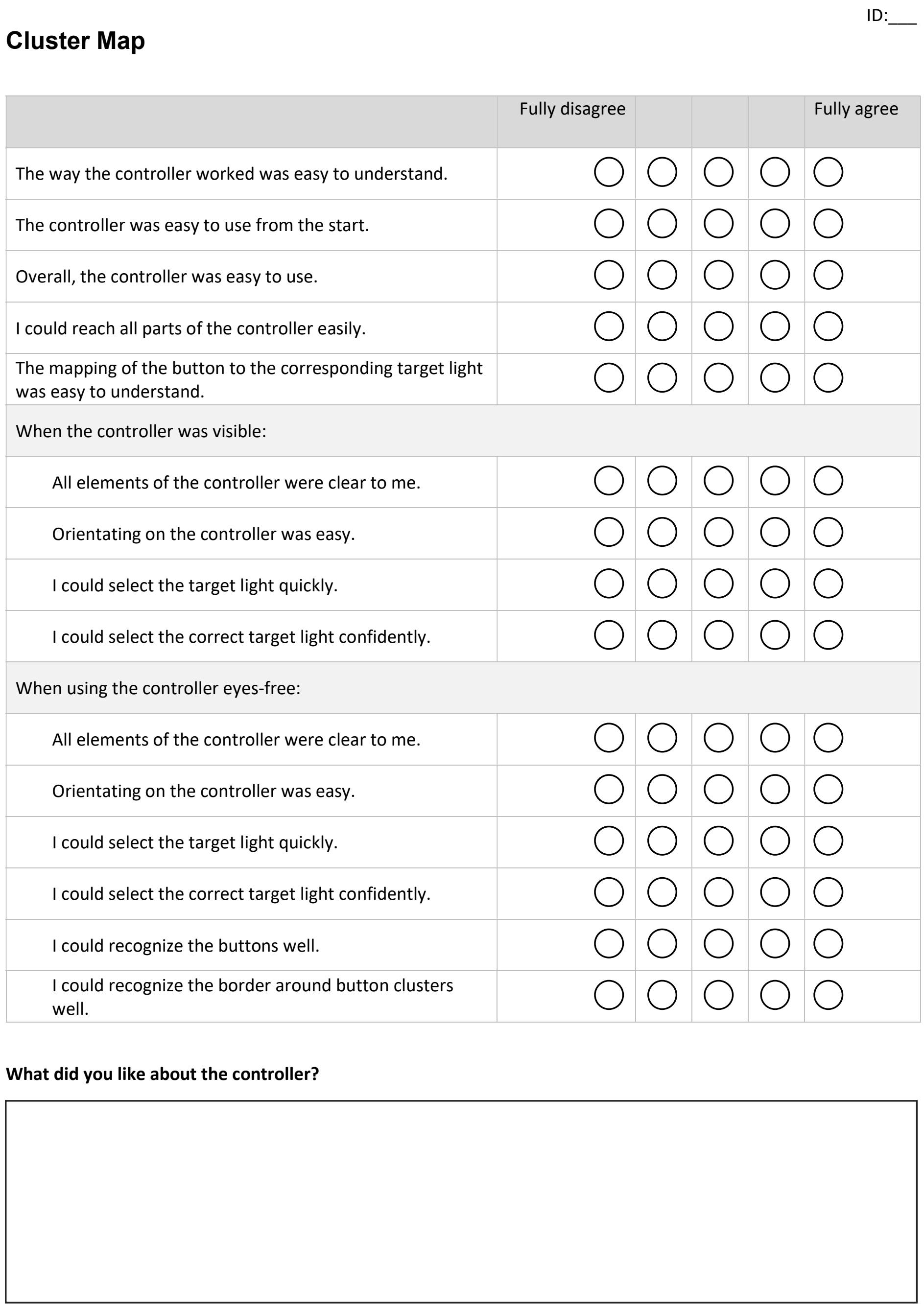
ID:___

What did you not like about the controller?

What would you do to enhance the controller?

Additional comments

Figure A.6: Second page of the questionnaire for the *Pillar Map*.



ID:___

Cluster Map

	Fully disagree				Fully agree
The way the controller worked was easy to understand.	○	○	○	○	○
The controller was easy to use from the start.	○	○	○	○	○
Overall, the controller was easy to use.	○	○	○	○	○
I could reach all parts of the controller easily.	○	○	○	○	○
The mapping of the button to the corresponding target light was easy to understand.	○	○	○	○	○
When the controller was visible:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
When using the controller eyes-free:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
I could recognize the buttons well.	○	○	○	○	○
I could recognize the border around button clusters well.	○	○	○	○	○

What did you like about the controller?

Figure A.7: First page of the questionnaire for the *Cluster Map*.

Figure A.8: Second page of the questionnaire for the *Cluster Map*.

ID:___

Direction Swiping

	Fully disagree				Fully agree
The way the controller worked was easy to understand.	○	○	○	○	○
The controller was easy to use from the start.	○	○	○	○	○
Overall, the controller was easy to use.	○	○	○	○	○
I could reach all parts of the controller easily.	○	○	○	○	○
The mapping of the button to the corresponding target light was easy to understand.	○	○	○	○	○
When the controller was visible:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○
When using the controller eyes-free:					
All elements of the controller were clear to me.	○	○	○	○	○
Orientating on the controller was easy.	○	○	○	○	○
I could select the target light quickly.	○	○	○	○	○
I could select the correct target light confidently.	○	○	○	○	○

What did you like about the controller?

Figure A.9: First page of the questionnaire for the *Direction Swiping Controller*.

ID:___

What did you not like about the controller?

What would you do to enhance the controller?

Additional comments

Figure A.10: Second page of the questionnaire for the *Direction Swiping Controller*.

ID:___

What did you not like about the controller?

What would you do to enhance the controller?

Additional comments

Figure A.11: Summary questionnaire asking for demographic information and a ranking of the controllers.

Appendix B

Latin Square

	1.	2.	3.	4.	5.
1	Sun	Cluster	Pillars	Zelda	Swipe
2	Pillars	Swipe	Sun	Zelda	Cluster
3	Zelda	Swipe	Cluster	Pillars	Sun
4	Cluster	Sun	Zelda	Pillars	Swipe
5	Pillars	Sun	Swipe	Cluster	Zelda
6	Swipe	Zelda	Pillars	Cluster	Sun
7	Cluster	Zelda	Sun	Swipe	Pillars
8	Sun	Pillars	Cluster	Swipe	Zelda
9	Swipe	Pillars	Zelda	Sun	Cluster
10	Zelda	Cluster	Swipe	Sun	Pillars

Figure B.1: The Latin square as it was followed to counterbalance order effects on the controllers.

Bibliography

Lea Albaugh, Scott Hudson, and Lining Yao. Digital fabrication of soft actuated objects by machine knitting. In Stephen Brewster, Geraldine Fitzpatrick, Anna Cox, and Vassilis Kostakos, editors, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, New York, NY, USA, 2019. ACM. ISBN 9781450359702. doi: 10.1145/3290605.3300414.

Fernando Alonso, José L. Fuertes, Ángel L. González, and Loïc Martínez. User-interface modelling for blind users. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, editors, *Computers Helping People with Special Needs*, volume 5105 of *Lecture Notes in Computer Science*, pages 789–796. Springer London, Limited, Guildford, 2008. ISBN 978-3-540-70539-0. doi: 10.1007/978-3-540-70540-6_117.

Jens Bornschein and Gerhard Weber. Digital drawing tools for blind users. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 21–28, [Place of publication not identified], 2017. ACM. ISBN 9781450352277. doi: 10.1145/3056540.3056542.

Laura Devendorf, Joanne Lo, Noura Howell, Jung Lin Lee, Nan-Wei Gong, M. Emre Karagozler, Shiho Fukuhara, Ivan Poupyrev, Eric Paulos, and Kimiko Ryokai. i don't want to wear a screen. In Jofish Kaye, editor, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 6028–6039, New York, 2016. Association for Computing Machinery. ISBN 9781450333627. doi: 10.1145/2858036.2858192.

Jun Gong, Yu Wu, Lei Yan, Teddy Seyed, and Xing-Dong

Yang. Tessutivo. In François Guimbretière, Michael Bernstein, and Katharina Reinecke, editors, *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 29–41, New York, NY, USA, 2019. ACM. ISBN 9781450368162. doi: 10.1145/3332165.3347897.

Nur Al-huda Hamdan, Jeffrey R. Blum, Florian Heller, Ravi Kanth Kosuru, and Jan Borchers. Grabbing at an angle. In Michael Beigl, Paul Lukowicz, Ulf Blanke, Kai Kunze, and Seungyon Claire Lee, editors, *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 1–7, New York, NY, USA, 2016. ACM. ISBN 9781450344609. doi: 10.1145/2971763.2971786.

Leona Holloway, Kim Marriott, and Matthew Butler. Accessible maps for the blind. In Regan Mandryk, Mark Hancock, Mark Perry, and Anna Cox, editors, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, New York, NY, USA, 2018. ACM. ISBN 9781450356206. doi: 10.1145/3173574.3173772.

Cedric Honnet, Hannah Perner-Wilson, Marc Teyssier, Bruno Fruchard, Jürgen Steimle, Ana C. Baptista, and Paul Strohmeier. Polysense: Augmenting textiles with electrical functionality using in-situ polymerization. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjørn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, New York, NY, USA, 2020. ACM. ISBN 9781450367080. doi: 10.1145/3313831.3376841.

Konstantin Klamka, Raimund Dachselt, and Jürgen Steimle. Rapid iron-on user interfaces: Hands-on fabrication of interactive textile prototypes. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjørn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, New York, NY, USA, 2020.

ACM. ISBN 9781450367080. doi: 10.1145/3313831.3376220.

Seonil Kwon, Hyuncheol Kim, Seungyeop Choi, Eun Gyo Jeong, Dohong Kim, Somin Lee, Ho Seung Lee, Young Cheol Seo, and Kyung Cheol Choi. Weavable and highly efficient organic light-emitting fibers for wearable electronics: A scalable, low-temperature process. *Nano letters*, 18(1):347–356, 2018. doi: 10.1021/acs.nanolett.7b04204.

Yiyue Luo, Kui Wu, Tomás Palacios, and Wojciech Matusik. Knitui: Fabricating interactive and sensing textiles with machine knitting. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Drucker, editors, *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, NY, USA, 2021. ACM. ISBN 9781450380966. doi: 10.1145/3411764.3445780.

Sara Mlakar and Michael Haller. Design investigation of embroidered interactive elements on non-wearable textile interfaces. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjørn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, New York, NY, USA, 2020. ACM. ISBN 9781450367080. doi: 10.1145/3313831.3376692.

Sara Mlakar, Mira Alida Haberfellner, Hans-Christian Jetter, and Michael Haller. Exploring affordances of surface gestures on textile user interfaces. In Wendy Ju, editor, *Designing Interactive Systems Conference 2021*, ACM Digital Library, pages 1159–1170, New York,NY,United States, 2021. Association for Computing Machinery. ISBN 9781450384766. doi: 10.1145/3461778.3462139.

Roberto A. Montano-Murillo, Cuong Nguyen, Rubaiat Habib Kazi, Sriram Subramanian, Stephen DiVerdi, and Diego Martinez-Plasencia. Slicing-volume: Hybrid 3d/2d multi-target selection technique for dense virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 53–62, [S.l.], 2020.

IEEE. ISBN 978-1-7281-5608-8. doi: 10.1109/VR46266.2020.00023.

Donald A. Norman. *The design of everyday things*. Basic Books, New York, revised and expanded ed. edition, 2013. ISBN 9780465050659. URL `http://swb.eblib.com/patron/FullRecord.aspx?p=1167019`.

Oliver Nowak, René Schäfer, Anke Brocker, Philipp Wacker, and Jan Borchers. Shaping textile sliders: An evaluation of form factors and tick marks for textile sliders. In Simone Barbosa, editor, *CHI Conference on Human Factors in Computing Systems*, ACM Digital Library, pages 1–14, New York,NY,United States, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517473.

Dominika Palivcová, Miroslav Macík, and Zdeněk Míkovec. Interactive tactile map as a tool for building spatial knowledge of visually impaired older adults. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjørn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–9, New York, NY, USA, 2020. ACM. ISBN 9781450368193. doi: 10.1145/3334480.3382912.

Richard Chulwoo Park, Hyunjae Lee, Hwan Kim, and Woohun Lee. The previewable switch. In Ron Wakkary, Steve Harrison, Carman Neustaedter, Shaowen Bardzell, and Eric Paulos, editors, *Proceedings of the 2014 conference on Designing interactive systems*, pages 191–194, New York, NY, USA, 2014. ACM. ISBN 9781450329026. doi: 10.1145/2598510.2600885.

Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoediauer, Martin Kaltenbrunner, Siegfried Bauer, and Michael Haller. Resi. In Patrick Baudisch, Albrecht Schmidt, and Andy Wilson, editors, *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 745–756, New

York, NY, USA, 2018. ACM. ISBN 9781450359481. doi: 10.1145/3242587.3242664.

E. R. Post and M. Orth. Smart fabric, or wearable clothing. In *Digest of Papers. First International Symposium on Wearable Computers*, pages 167–168. IEEE Comput. Soc, 1997. ISBN 0-8186-8192-6. doi: 10.1109/ISWC.1997.629937.

Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. Project jacquard. In Jofish Kaye, editor, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4216–4227, New York, 2016. Association for Computing Machinery. ISBN 9781450333627. doi: 10.1145/2858036.2858176.

Jung-Sim Roh. Textile touch sensors for wearable and ubiquitous interfaces. *Textile Research Journal*, 84(7):739–750, 2014. ISSN 0040-5175. doi: 10.1177/0040517513503733.

Jonathan Wonner, Jérôme Grosjean, Antonio Capobianco, and Dominique Bechmann. Starfish. In Mark Green, Wolfgang Stuerzlinger, Marc Erich Latoschik, and Bill Kapralos, editors, *VRST '12*, page 101, New York, N.Y., 2012. Association for Computing Machinery. ISBN 9781450314695. doi: 10.1145/2407336.2407356.

Tony Wu, Shiho Fukuhara, Nicholas Gillian, Kishore Sundara-Rajan, and Ivan Poupyrev. Zebrasense. In Shamsi Iqbal, Karon MacLean, Fanny Chevalier, and Stefanie Mueller, editors, *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 662–674, New York, NY, USA, 2020. ACM. ISBN 9781450375146. doi: 10.1145/3379337.3415886.

Mengjia Zhu, Amirhossein H. Memar, Aakar Gupta, Majed Samad, Priyanshu Agarwal, Yon Visell, Sean J. Keller, and Nicholas Colonnese. Pneusleeve: In-fabric multimodal actuation and sensing in a soft, compact, and expressive haptic sleeve. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjørn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Proceedings of the*

2020 CHI Conference on Human Factors in Computing Systems, pages 1–12, New York, NY, USA, 2020. ACM. ISBN 9781450367080. doi: 10.1145/3313831.3376333.

Index

Typeset June 2, 2022