Control your home with augmented reality

Daniel Tovesson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2020

MASTER'S THESIS

Control your home with augmented reality

Daniel Tovesson

dic15dto@student.lu.se

May 4, 2020

Lund University

Control your home with augmented reality

Published by
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)
Division: Ergonomics and Aerosol Technology
Supervisors:
Günter Alce, gunter.alce@design.lth.se
Pär Sikö, par.siko@jayway.com
Examiner:
Mattias Wallergård, mattias.wallergard@design.lth.se

Abstract

As connected devices, or Internet of Things (IoT), are becoming increasingly more common in our homes we need a way to control them remotely. Remote control can be achieved with native applications created by the providers of these IoT devices which can result in you needing several applications to control your IoT devices. Furthermore, issues can arise when naming your IoT devices, e.g. if they are located close to each other it might be hard to separate them by name. Can a relatively new technology such as augmented reality (AR) be used to solve these problems? With AR you can add virtual objects to a real-world environment. By integrating virtual objects with the real-world environment it is easy to understand the context of where the virtual objects are located. AR technology is developing at a high speed and with recent advances in Apple's ARKit, it is getting more accessible by moving into our phones.

This thesis aims to explore how useful it would be to use AR on a mobile phone as a user interface for controlling IoT devices. A prototype where you can control connected light bulbs, a connected speaker, and a motion detector alarm was developed and evaluated in a user study with 20 test participants. The results show potential using AR to control your IoT devices but also indicate that the technology might not be mature in its current state.

Keywords: Augmented reality, Internet of Things, Smart homes, ARKit, Interaction design, User-Centered Design

Sammanfattning

Uppkopplade enheter, eller Internet of Things (IoT), blir alltmer vanliga i våra hem vilket resulterar i ett behov att fjärrstyra dem. Man kan åstadkomma detta via applikationer som distribueras av tillverkarna av dessa IoT enheter, vilket kan resultera i att du behöver flertalet applikationer för att kontrollera dina IoT enheter. Problem kan också uppstå när man ska namnge sina IoT enheter, det kan till exempel vara svårt att separera dom med namn om de är placerade nära varandra. Kan en relativt ny teknologi som förstärkt verklighet (AR), augmented reality på engelska, användas för att lösa dessa problem? Med AR har man möjligheten att lägga till virtuella element i en verklig miljö vilket gör att det är lätt att förstå i kontexten vart de olika virtuella elementen är placerade. AR teknologi utvecklas i en hög takt och med de framsteg som Apple gjort med deras ARKit blir teknologin alltmer tillgänglig genom att flytta in i våra telefoner.

Målet med denna uppsats är att utvärdera hur användbart det är att använda AR på en mobiltelefon för att kontrollera IoT enheter. En prototyp där man kan kontrollera uppkopplade lampor, en uppkopplad högtalare och ett rörelsealarm har utvecklats och utvärderats i en användarstudie med 20 testpersoner. Resultatet visar potential i användandet av AR för att kontrollera IoT enheter men visar också att teknologin kanske inte är mogen nu i sin nuvarande form.

Nyckelord: Förstärkt verklighet, Sakernas internet, Smarta hem, ARKit, Interaktionsdesign, Användarcentrerad design

Acknowledgments

This master thesis was done in collaboration with Jayway AB Malmö and the Department of Design Sciences at Lunds University. I want to thank my supervisor Günter Alce for the great support and continuous feedback during the project.

I also want to thank Jayway and all employees that have provided me with valuable feedback during the project. A special thanks to Pär Sikö for supporting me and giving me the tools I needed for this project.

Lastly, I want to thank everyone who participated in the user study, it provided me with invaluable data for this project.

Contents

Glossary

AR Augmented reality. 1, 5–8, 11–14, 16, 19, 22, 24–35, 39, 41, 42, 44, 45, 48, 55

CV Computer vision. 6

Hi-fi High-fidelity. 14, 24

IoT Internet of Things. 1, 5–8, 11, 13–15, 23–26, 31–35, 39, 41, 42, 48

IQR Interquartile range. 27, 28, 33

Lo-fi Low-fidelity. 14

Mid-fi Medium-fidelity. 14, 18, 32

NASA-TLX NASA Task Load Index. 3, 9, 10, 14, 25–29, 33, 35, 40, 44, 49, 53

SUS System Usability Scale. 3, 9, 14, 25, 26, 28, 32, 35, 40, 44, 49, 52

UCD User-Centered Design. 9

VE Virtual environments. 8

WIM World in Miniature. 6

Chapter 1
Introduction

Today we interact with numerous devices, usually at a distance. To interact with these devices remote controls are often used. This has lately become a bigger problem with the increasing number of devices in our homes. The buttons on a remote control were initially duplicated from the devices so that you had two ways to control them [9]. These days most contemporary remotes have become the primary face to interact with your devices, this has lead to scalability issues. As the number of remote controls has increased it is getting more difficult to interact with your devices. Hence, the universal remote was invented. Although, the universal remote introduced new problems: it was often limited to entertainment systems, had difficult setup issues, and poorly adaptable interfaces [9]. As our devices get smarter and connected to the internet (IoT) an additional way of controlling your devices has emerged - applications. However, with multiple IoT device manufacturers, it can result in you needing multiple applications to control your IoT devices. With these problems in mind, could the relatively new technology augmented reality (AR) be used to control your IoT devices?

According to Azuma [6]: "Augmented Reality enhances a user's perception of and interaction with the real world. The virtual objects display information that the user cannot directly detect with his own senses". With the increasing number of IoT devices in our homes, it is therefore interesting to explore if the benefits of seeing everything in its real-world environment can be used to enhance the user's experience when interacting with IoT devices.

1.1 Purpose and goal

The purpose of this thesis was to explore how useful it would be to use AR as a user interface for controlling IoT devices. For this thesis the following research questions were asked:

1. How far has AR technology come and can it be applied to IoT devices?

2. How intuitive is it to use AR in a mobile application compared to the native applications when controlling your IoT devices?

3. How well does AR technology work as a user interface compared to traditional ways of controlling IoT devices?

1.2 Limitations and scope

Due to this project being a master thesis with a limited time frame of 20 weeks it was necessary to set some limitations. These limitations were set to focus on answering the research questions instead of building a complete production system. The limitations were the following:

- Integrate only a few IoT devices, in this case, Philips Hue light bulbs, a Sonos Play:1 speaker, and a motion detector.
- Focus on making it work in one room, in this case, a conference room at the Jayway office in Malmö.
- Manually add the devices in the AR application. The IoT devices used were not aware of their position in the room, so this could not be done automatically.
- Use ARKit only, a framework developed by Apple that you can use to create AR applications, which only work on iOS devices. ARKit has been around for years and is a well-developed tool for developing AR applications. Because of the time limit and previous experience with iOS, it was decided that most progress would be made using ARKit.

1.3 Related work

Previous research has explored interaction techniques between connected devices in smart homes. One paper describes UbiCompass, a novel IoT interaction concept [1]. The user interacts with a smartwatch face prototype to control their smart home. Five different connected devices could be controlled using a simple interaction. Icons are placed around the watch face, where each icon represents a connected device. To select a device there is an arrow placed by the 12 on the watch face. When you rotate the watch the icons rotate, and once an icon is aligned under the 12 on the watch face it can be selected by pressing a button, Control, on the watch face.

Another idea is Tag-It, which is based on computer vision (CV). It uses two wearable technologies, a head-worn wearable computer (Google Glass), and a chest-worn depth sensor (Tango). Google Glass generates and displays virtual information to the user while Tango provides robust indoor position tracking for Google Glass [1].

All in all, a user of an IoT system needs to be able to perform four basic tasks: (1) discover devices, (2) select a particular device, (3) view the device's status, and (4) control the device [9].

Alce et al. [2] explore AR as a user interface for IoT. Controlling the device can be quite cumbersome, as they do not visualize the real world, e.g. Apple HomeKit and Samsung SmartThings. Research has therefore been done exploring the benefits of using AR to interact with different connected devices. Three interaction models have been compared; floating icons, World in Miniature (WIM), and floating menu, which was developed using the Microsoft HoloLens, the Unity game engine, and Microsoft Visual Studio.

Floating icons are based on the idea to have icons places approximately where the device is located. The icons are round, flat and always facing the user. The icon changes color once the user's head is in the same direction as the icon, informing the user that actions can be performed on that icon. Thereafter the user can simply use the HoloLens tap gesture to activate the device's main function. For instance, for a lamp, clicking on its corresponding icon would turn it on or off, and the icon changes to gray if the device is turned off.

Inspiration was taken from these papers when developing the prototype. The selecting feature in the UbiCompass is similar to the one in the AR application, the difference is that devices are selected automatically once aligned with the bullseye. Floating icons were also used in a simple form. Instead of having icons showing the type of device a gray sphere was used. Once it was selected it was turned green, so you could not see if e.g. a light was on by looking at the sphere. Finally, the four basic tasks users of an IoT system need to be able to perform were taken into consideration. In the AR application users can (1) discover devices by looking in the AR view, (2) select a particular device by aligning the bullseye on the gray sphere, (3) view the device's status when it is selected, and (4) control the device when it is selected.

Chapter 2
Theoretical background

Two different techniques have been used during the development of this project: AR and IoT. Using ARKit these techniques were integrated into a mobile application. During the development process, interaction design has been practiced. The theory behind AR, interaction design, IoT, and ARKit will be explained further in this chapter.

2.1 Augmented reality

Azuma [6] describes AR in the following way:

> Augmented reality (AR) is a variation of virtual environments (VE), or virtual reality as it is more commonly called. VE technologies completely immerse a user in a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it.

He then continues with:

> AR can be thought of as the "middle ground" between VE (completely synthetic) and telepresence (completely real).

Furthermore, Azuma et al. [5] describe an AR system with three characteristics: (1) combines real and virtual, (2) interactive in real-time, and (3) registered in 3-D.

AR can, therefore, be described as a way of adding virtual objects in a real-world environment. The real-world environment can either be a video feed, e.g. in a mobile phone, or seen through optical lenses, e.g. AR glasses.

2.2 Interaction design

2.2.1 User-Centered Design

User-Centered Design (UCD) is an iterative design process [8]. That means the focus is on the users and their needs in each step of the design process. With UCD, you can use a variety of different methods and tools to help develop an understanding of user needs. Norman [10] points out that design should: "Make sure that (1) the user can figure out what to do, and (2) the user can tell what is going on." The iterative design process of a UCD can be divided into four parts: analysis, design, evaluation, and implementation. This is illustrated in figure 2.1.

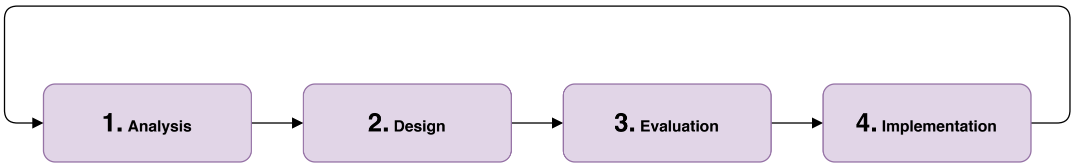


Figure 2.1: An iterative design process

Brainstorming

The goal with brainstorming is to produce as many ideas as possible from a group of people [4]. An important part of brainstorming is that no ideas should be criticized and crazy ideas should be encouraged. By doing that people start thinking outside the box and a lot of ideas can be generated. The first step in brainstorming is to define a problem. After that, the participants are given several minutes to generate ideas. The ideas are then read aloud and after that discussed in the group. They can then be rated and evaluated, but keep in mind to not criticize any of the ideas.

Questionnaires

Questionnaires can be used to gather information about a test participant's experience within the area as well as their feelings after the test has been finished. Some examples are:

- **Personal information:** gathers information about the participant. It can be gender, age and previous experience with the system that is to be tested.
- **System Usability Scale (SUS):** a tool for measuring the usability of a system [16]. SUS has become an industry standard, and can therefore effectively be used to differentiate between usable and unusable systems. SUS consists of 10 questions where the participant gets to choose one of the five responses that range from *Strongly disagree* to *Strongly agree*. An example of a SUS can be seen in appendix C.3. To interpret and compare the results the participant's scores for each question are converted to a new number, added together and then multiplied by 2.5 to convert the original scored of 0-40 to 0-100. Based on research, a SUS score above 68 would be considered above average and anything below 68 is below average [16].
- **NASA Task Load Index (NASA-TLX):** a subjective workload assessment tool [14]. Based on a weighted average of ratings on six sub-scales:

- Mental demand
- Physical demand
- Temporal demand
- Performance
- Effort
- Frustration

Each sub-scale are converted to scores between 0-100 by multiplying the participant's scores by 5. This is a simplified version of NASA-TLX referred to as "Raw TLX" which has been shown to have a high correlation with NASA-TLX [7]. An example of a NASA-TLX can be seen in appendix C.4.

A modified version of NASA-TLX called NASA-RTLX (Raw TLX) was used to minimize the time to answer questionnaires during the test session. Furthermore, a high correlation between NASA-TLX and NASA-RTLX scores has been shown [7].

Interviews

Interviews can be used to gather additional information about the user's experience with the product [4]. You can use something called a semi-structured interview where you have some questions as support but the structure of the interview should be like a normal conversation. That is a good way of getting the test participant to open up and express their feelings about the product.

Scenarios

Used to put the test participant into a certain scenario [4] in a story. This makes it easier for the test participant to understand and visualize the tasks that should be done. The scenarios can be used before a test session to give the test participant a better understanding of the different tasks the test participant will perform during the test session.

Prototyping

A prototype is a draft version of a product [15]. It can be anything from paper drawings (low-fidelity) to an application that allows the user to explore pieces that are fully functioning. A prototype can, therefore, be used to explore ideas and show the intention behind a feature or the overall design concept to users before investing time and money into development.

2.2.2 Conceptual model

A conceptual model is something we form in our minds to mentally simulate a device's operation [10]. A conceptual model can be realized by the device's visible structure, especially from signifiers, constraints, and mappings.

Signifiers

As mentioned by Norman [10], "Signifiers signal things, in particular what actions are possible and how they should be done." A signifier can be an object indicating that it can be interacted with, e.g. a doorknob handle. Having good signifiers is key to making something intuitive and give the user's a good experience when interacting with a product, without the need for additional instructions.

Constraints

Take a scissor as an example, you don't need any instructions to use it. That is because the holes where you insert your fingers are signifiers [10]. The size of the wholes serves as constraints, limiting the possible number of fingers to insert. Constraints can, therefore, be explained as an indicator for a user how to interact with a given device, it limits the user to a possible set of operations.

Mapping

Mapping indicates the relationship between two things [10]. Take the scissor as an example again, the mapping between the holes and fingers are suggested and constrained by the holes.

2.2.3 Feedback

With feedback, you can give the user information that an action has been performed [10]. An example of a device without feedback is a pencil that leaves no mark when drawing. Good feedback gives the user assurance than a task has been performed and the user can continue doing the next task, without being stuck on the first task.

2.3 Internet of Things

In an article posted by Margaret Rose, Internet of Things (IoT) is defined as follows [13]:

> The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

An IoT "thing" can be a heart monitor implant inside a person, a bio-chip transponder inside a farm animal, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an IP address and can transfer data over a network.

2.4 ARKit

ARKit is Apple's AR development platform for iOS mobile devices [12]. It has been around since 2017 and is currently at version 3. It uses the iOS device's camera, accelerometers, gyroscope and context awareness, to map the environment when the device is moved. It has since its release been

 2. Theoretical background

adopted by many developers and has increased in popularity. In 2018 over 13 million AR apps built using Apple's ARKit had been downloaded since its release [11].

Chapter 3
Development process

In this chapter, the development process is described. In section 3.1 the methods used for the development and testing of the AR application are described. Section 3.2 describes the different iterations made during the development process of the AR application.

3.1 Method

An AR application was created in an iterative development process that can be seen in figure 3.1. With the AR application you could control different IoT devices. These devices are the following:

1. Three Philips Hue light bulbs
2. Sonos Play:1 speaker
3. Motion detector

The last device, the motion detector, is not connected to the internet. The device just shows static data that has been made up.

In the initial phase of the project, a brainstorming session was held with five participants. The participants were presented with two different cases:

1. You are on the sofa and want to check what lights are on in the house, you also want to control the connected devices. You bring out your phone, which is context-aware, and the AR camera view is already open. What do you want to see?
2. You are on vacation and can not remember if you left a light on. You bring out your phone and the AR view is active. What do you want to see?

The different cases were presented separately and after each case had been presented the participants got 3 minutes to generate ideas on Post-it notes. When the idea generation phase was done each participant got to present their ideas and receive feedback for 3 minutes.

After the brainstorming session, it was decided to create an AR application where users can control different IoT devices by pointing at them. When pointing at them the IoT device should be selected and the user can interact with it either by using toggles on the screen or voice commands.

It is hard to visualize AR in a low-fidelity (Lo-fi) prototype so it was decided to iteratively develop a medium-fidelity (Mid-fi) prototype. Once all features were in place the high-fidelity (Hi-fi) prototype was finished and ready for the user study.

The AR application was compared to the native applications created by the manufacturers to control the IoT devices. To compare these a total of 20 test participants were chosen. Half of them started interacting with the AR application while the other half started with the native applications. To interact with the application(s) a number of scenarios were read to the user that was connected to different tasks they were supposed to perform. The scenarios and tasks can be seen in the test plan in appendix A, section A.4, and A.6. After that was done they filled out a SUS and NASA-TLX. When that was done they tested the application(s) they did not test in the first test and filled out a SUS and NASA-TLX for that test as well. This data was then compared to see if there was a difference in how the test participant experienced the different systems. All tests were also recorded with sound so that the task completion time could be measured. This data was then compared between the applications to see if the AR application is a more effective user interface when interacting with your IoT devices.

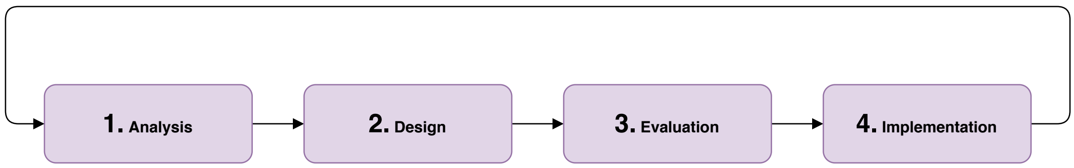


Figure 3.1: The development process

3.2 Implementation

The design and development phase was done iteratively. Each iteration consisted of three parts, specifying requirements, implementation, and evaluation. The evaluation was performed after each implementation step was finished, to confirm that it works as intended. The product was evaluated on both Jayway employees and by myself.

3.2.1 Control light bulbs

As none of the IoT devices used were aware of their position in the room virtual objects had to be manually placed. In order to save time when testing the AR application, it was decided to implement functionality where you can save, load, and remove world maps. The world maps have information about reference points as well as the virtual objects that have been placed out. With this functionality in place, you can save maps for different rooms for different purposes as well as loading maps after restarting the application.

Finally, IoT devices were connected to the virtual elements. Philips Hue Color light bulbs were decided to be the first IoT devices to implement because Philips Hue has an open API that can be used for controlling the light bulbs.

Requirements

- Add a debug state where you can add virtual objects and make sure that a normal user can not access it.
- Click on the screen to place an object on the surface where the screen was tapped.
- Click on a virtual object to remove it.
- Save, load, and remove maps with virtual objects.
- Turn on/off the light bulb.
- Change the brightness of the light bulb.
- Change the color of the light bulb.

Implementation

To make the debug state not accessible by normal users a gesture that a user would not normally do when interacting with the app had to be figured out. First, three consecutive taps on the screen were tried out, but this was soon realized as something the user could accidentally do while interacting with the application. Finally, a four-finger swipe was decided as the gesture to activate the debug state. To place a virtual object on a surface a feature in ARKit called World Tracking was used. Using World Tracking you could use ray-casting to figure out the position of the surface where the screen was tapped and place a virtual object there. A virtual object placed on a surface can be seen in figure 3.2.

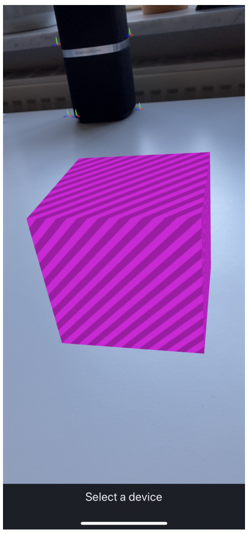


Figure 3.2: A virtual object placed on a surface

Removing a virtual object is as simple as adding one. Two different modes were added to achieve this: add and remove which can be seen in figure 3.3. If you are in the mode add and tap on the screen a virtual object is added, and if you are in the mode remove and tap on a virtual object it is removed from the AR view.

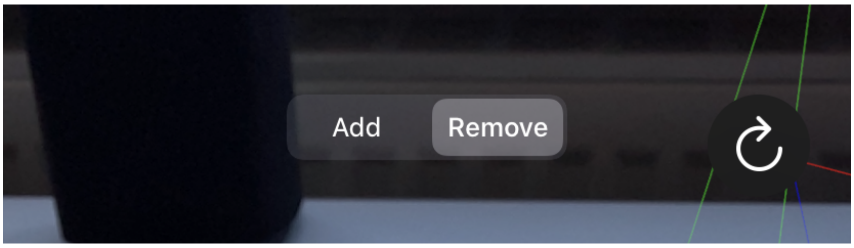


Figure 3.3: The toggle for the different modes

Buttons for saving, loading, and removing maps were added in the debug state which can be seen in figure 3.4. When the save button was pressed you could choose to either create a new map or replace it with an existing map which can be seen in figure 3.5a. The world map was then saved to the phones' local storage. When loading/removing a map you got to choose which map you wanted to load/remove which can be seen in figure 3.5b and 3.5c.

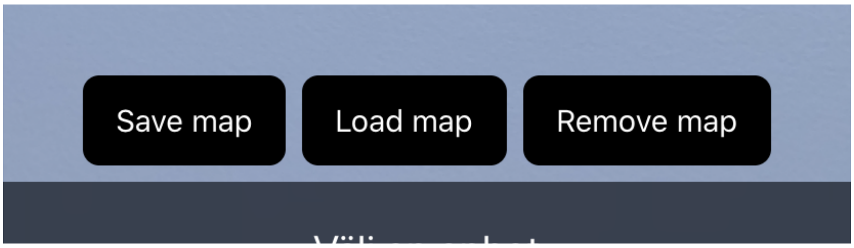


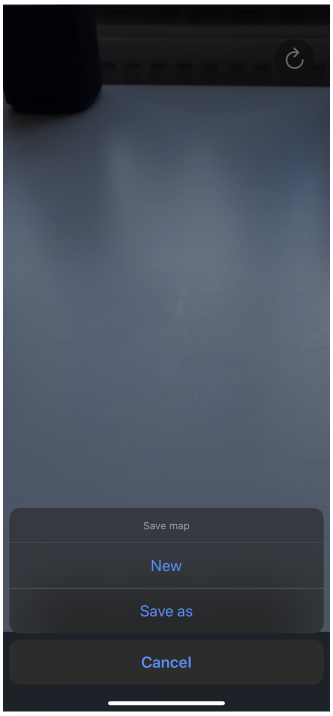
Figure 3.4: Buttons for saving, loading and removing a world map

To satisfy the requirements of controlling individual light bulbs you needed to figure out a way to connect light bulbs to certain virtual objects. This was done by presenting the user with a table of available devices when the user touches the screen which can be seen in figure 3.6. In the table you can see which light bulb is which, but if you are unsure you can click the cell and it will flash for a second. For each light bulb, you have the option to press a connect button, this button connects the light bulb to the virtual object you are creating.

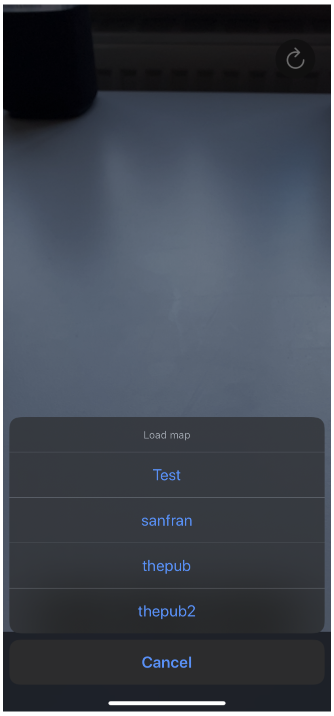
The next issue to solve was the selection of virtual objects. This was done by creating a bullseye in the middle of the screen which, once aligned with the virtual object, selected that object. To give the user good feedback for when the virtual object is selected it was made green and the phone vibrated.

When the virtual object was selected its name was shown, a switch for turning the light bulb on and off, a slider for changing the brightness of the light bulb, and a button for opening a color picker menu which can be seen in figure 3.7a. The button for opening the color picker menu had the same color as the light bulb. The color picker menu gave the user the option to set the light bulb to any color, and it was dismissed by swiping it down or pressing the button "Stäng", meaning "Close" in English, which can be seen in figure 3.7b.

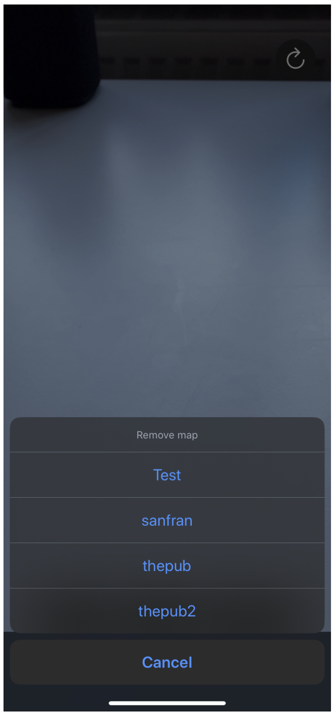
Lastly, voice commands for the different actions you could perform on the light bulbs were integrated. For this Apple's Speech framework was used [3]. To turn on/off a light bulb you would



(a) Saving a world map



(b) Loading a world map



(c) Removing a world map

Figure 3.5: Menus for saving, loading and removing a world map

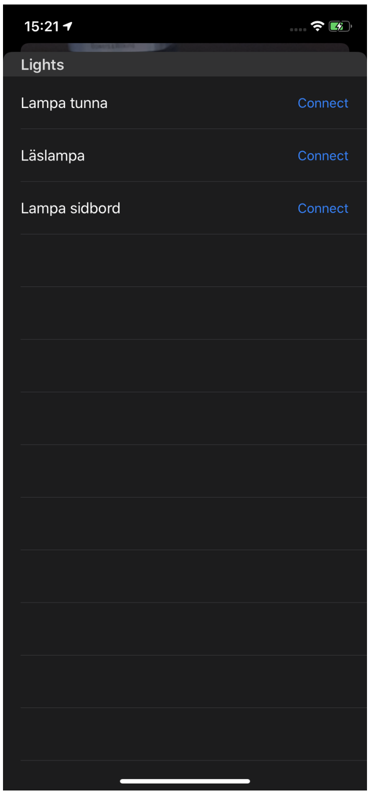
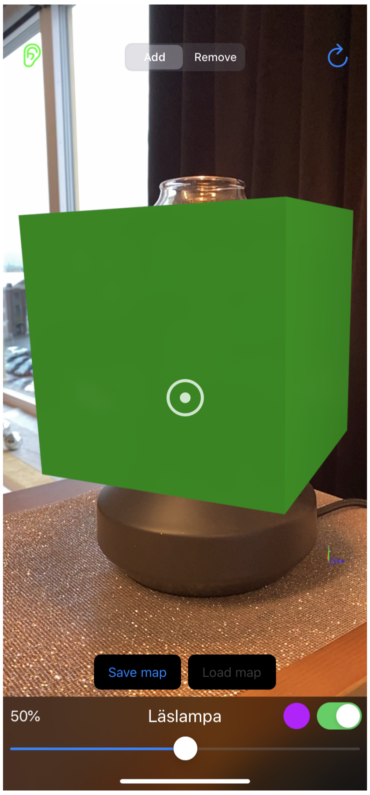


Figure 3.6: List of available lights

speak the Swedish words *tänd/släck*, which means turn on and turn off respectively. To change the brightness of the light bulb you would just ask it to set itself to a certain percentage and it would. Changing the light bulbs color with your voice was challenging, it was considered that you should be able to tell it to change itself to e.g. red or green, but then you would decide which kind of red and green that would be. In the end, a simpler solution was chosen, if you speak the Swedish word *färg*, which means color, the color picker menu would open for you. Then you had to manually select the color.

Apple's Speech framework uses a lot of processing power so in order to optimize performance the application would only listen to you when you had a virtual object selected. In the debug state an ear symbol was added, which can be seen in the top left corner of figure 3.7a, showing if the application was listening or not to confirm that the feature was working correctly when selecting and deselecting virtual objects.



(a) Light bulb selected

(b) Color picker view

Figure 3.7: Controlling light bulb

Evaluation

To evaluate this iteration a Mid-fi prototype test was conducted using the current implementation of the application. Each test subject was asked to perform four tasks on one light bulb:

1. Turn on the light.
2. Change the color of the light.

3. Set the brightness to 100%.

4. Explore the different voice commands, you can turn on/off the light bulb, change the brightness, and open the color picker menu.

For these test sessions, three test participants were chosen. The test participants were all working at the Jayway office as developers. The observations made during the test sessions can be seen in table 3.1

Table 3.1: Observations from the test session

# Task	#	Test participant observation
1	1	No issues
	2	Tried to change the slider first, eventually figured out that the switch should be used
	3	Had some issues keeping the light selected
2	1	Opened the color picker menu but did accidentally dismiss it because the participant deselected the light
	2	Had issues finding the button to open the color picker, it had the color white so it looked similar to the switch
	3	No issues
3	1	Had problems keeping the light selected while moving the slider, eventually succeeded
	2	Same as above
	3	Same as above
4	1	Managed to execute all voice commands but had to repeat some of them
	2	Did not work at all, and initially tried to speak English with the application
	3	Only managed to turn off the light, nothing else. Started each sentence with "Hey Siri"

To sum it up it was clear that test participants had problems keeping the light bulb selected. It was suggested by some of the test participants that a light bulb could be kept selected until manually dismissed or another virtual object is selected. Some test participants also noted that the virtual objects obscured the light bulb so that you could not see the light bulb through the AR view. Which language to use when speaking to the application and how to toggle the color picker menu also has to be made clearer. Finally, the voice commands did not work as intended, they did not work without issue for any of the test participants.

3.2.2 Improving light bulb controls

Conducting high-fidelity testing the results showed flaws in the implementation. This iteration focuses on fixing those flaws.

Requirements

- Improve the selection of virtual objects.
- Do not obscure the light bulbs with the virtual objects.
- Make it clearer how to toggle the color picket menu.
- Improve voice commands.

Implementation

To improve the selection of virtual objects it was decided to keep the virtual objects selected until manually dismissed or a new one was selected. To stop the virtual objects from obscuring the light bulbs the virtual objects were placed 10 cm above where the user pressed, now instead hovering over the light bulbs. To address the issue with the color picker menu the view for controlling the light bulb was redesigned. Now the whole view was clickable and the same color as the light bulb, much like the native Philips Hue application which can be seen in figure 3.8a. After that, the issue with the voice commands was investigated. It was realized that the voice commands are sent via a network request and can, therefore, start buffering if it has to analyze a lot of speech. There is functionality for doing all the processing on the device but that does not work in Swedish at the time of the implementation. It was then decided to not listen when the virtual object is selected, but instead, listen when the user presses a microphone button that is visible when a virtual object is selected. The user can then use voice commands on the light to do a task, and then have to press the button again to perform a new task. All these changes can be seen in figure 3.8b.

Evaluation

The selection of virtual objects now worked much better. The color picker menu was also much easier to open now. The speech worked better but could still buffer up if the user pressed the microphone button and then started speaking random sentences before trying to control the light.

3.2.3 Connection issues

While developing a new issue was raised. Sometimes the requests to the light bulb failed. That was discovered to be because the phone was on the wrong Wi-Fi network.

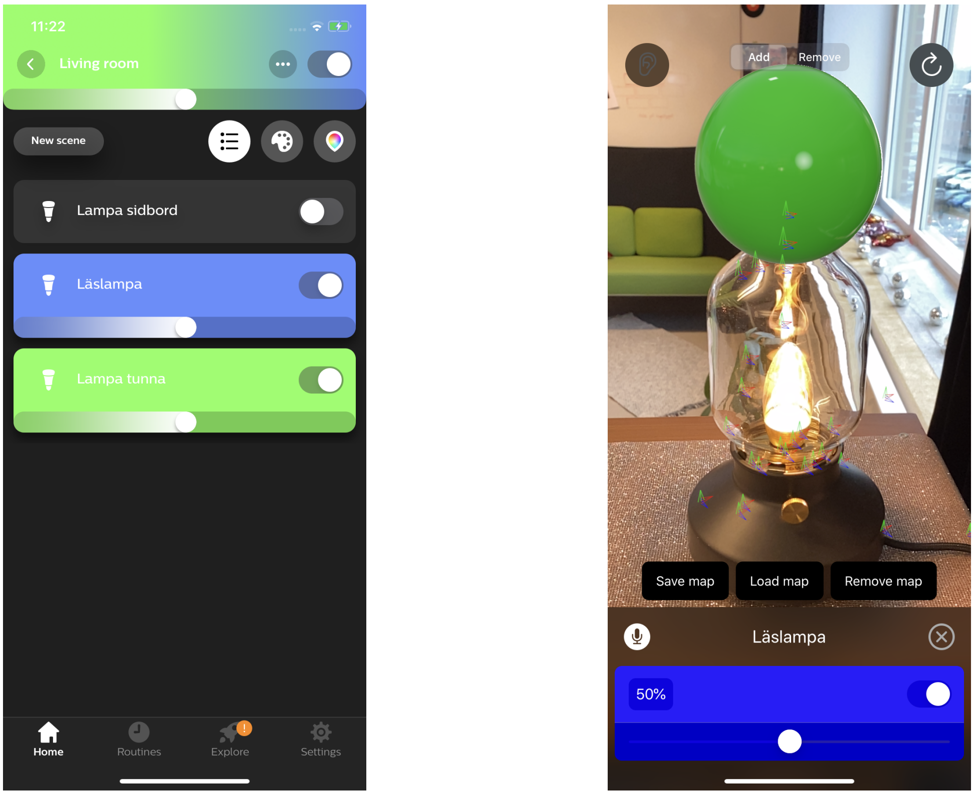
Requirements

- Prevent the user from being on the wrong network.

Implementation

When the user is on the wrong Wi-Fi the application is now disabled and a red screen is shown, telling the user to switch to the correct Wi-Fi, which can be seen in figure 3.9.

 3. Development process



(a) Philips Hue application **(b)** Updated light bulb selection

Figure 3.8: Improved controls

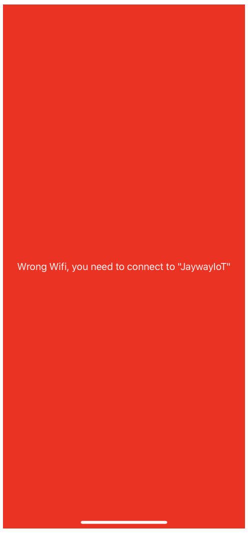


Figure 3.9: Warning that the user is on the wrong Wi-Fi

Evaluation

This was an effective way of solving the issue, the user can not use the application if the user is on the wrong Wi-Fi. This eliminates possible issues during testing. Although, this could be solved more cleanly. If more time had been put on developing this feature you could disable all buttons in the view and a small alert could ask the user to change to the correct Wi-Fi, instead of blocking the whole view. However, this feature was not essential to the functionality of the application and was therefore not prioritized.

3.2.4 Controlling speaker

With some time left for developing the application further, it was decided that a speaker would be interesting to implement, more specifically a Sonos Play:1.

Requirements

- Play music.
- Pause the music.
- Skip to the previous track.
- Skip to the next track.
- Change the volume of the music.
- Show album art.
- See the current track that is playing.
- See the album's name.
- See the artist's name.

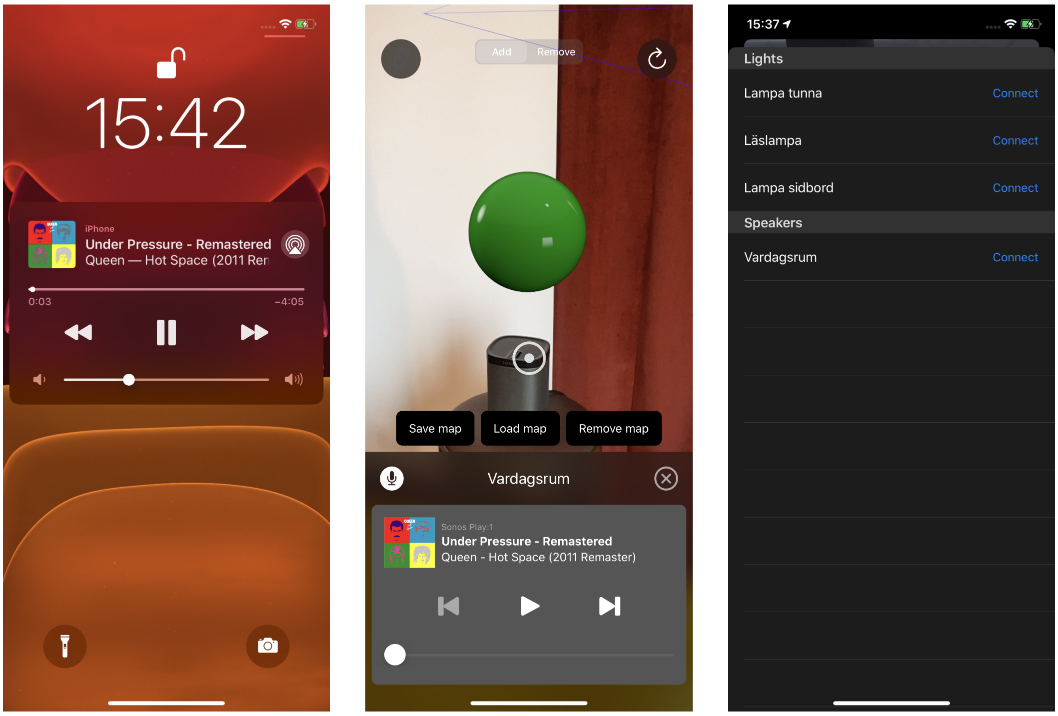
Implementation

The Sonos Play:1 speaker was controlled by sending HTTP requests to a node server found on GitHub. This was the easiest way of getting something to work but it required the node server to be running when interacting with the Sonos Play:1 speaker.

Inspiration was taken from Apple's media control menu that is present on the lock screen when playing e.g. music which can be seen in figure 3.10a. The controls that were added were a back button, next button, play/pause button and a slider for the volume. Above it, all the title of the track currently playing was added and under that the album name and artist name. To the left of these titles, the album art is shown.

Speech recognition was implemented here as well, to play the music you say *spela*, to pause music you say *pausa*, to go to the previous track you say *föregående*, to skip to next track you say *nästa*, and to change the volume you tell the speaker to set itself to a certain percentage. The completed control menu for speakers can be seen in figure 3.10b.

The process for adding a speaker to the AR view is the same as for light bulbs which can be seen in figure 3.10c.



(a) Apple's media control menu **(b)** Speaker selected **(c)** List of available speakers

Figure 3.10: Controlling the speaker

Evaluation

The feature worked well. It was easy to switch between the different devices and the speaker was responsive to the different actions you could perform with it. However, the speaker does not work if there is no queue active, which has to be started from a separate application. But for testing purposes this is sufficient enough as changing between playlists and searching for songs are not supported.

3.2.5 Motion detector status

With no more IoT devices left to integrate and no time to integrate new ones it was decided to implement a motion detector that just showed static data.

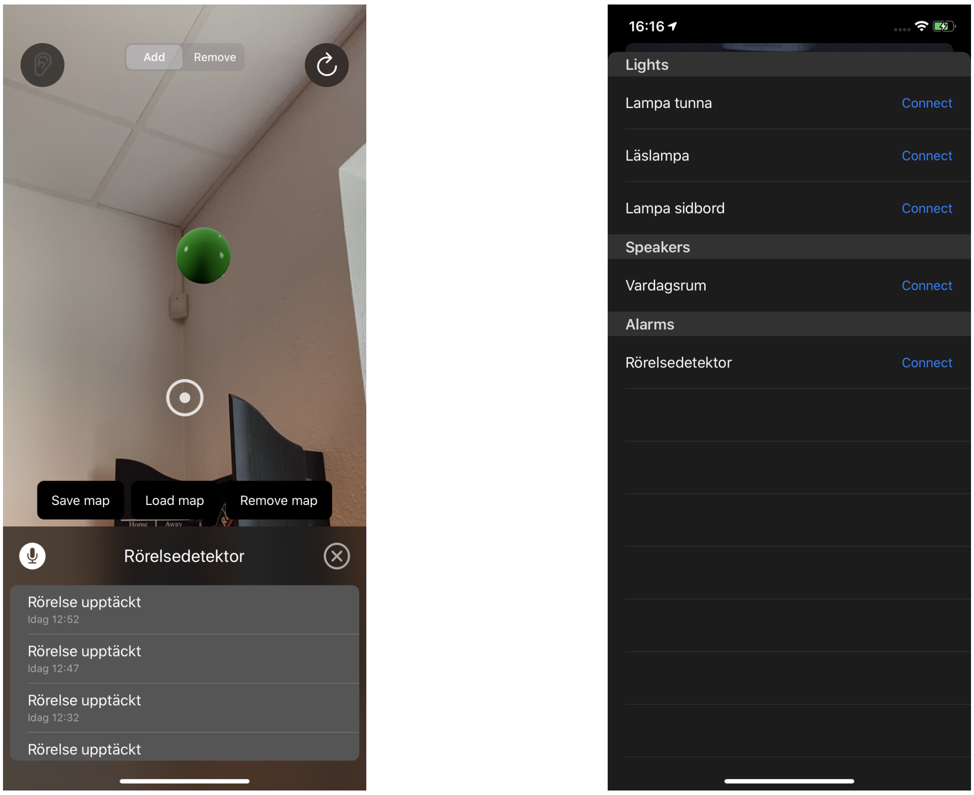
Requirements

- Show registered motion
- Show when the motion was registered

Implementation

When selecting a virtual object that is a smoke detector a list of detected motions is shown. Each list item has a title as well as the time for when the motion was detected which can be seen in figure 3.11a.

The process for adding a motion detector to the AR view is the same as for light bulbs and speakers which can be seen in figure 3.11b.



(a) Motion detector selected

(b) List of available motion detectors

Figure 3.11: Control motion detector

Evaluation

With the current implementation, it was very easy to add this. This means that the application could easily be expanded with new IoT devices showing static data. The Hi-fi prototype was now finished and ready for the user study.

3.2.6 Remove speech

After some testing and discussion, it was decided to remove the speech functionality as it worked quite poorly and would not add any value to the research.

Requirements

- Remove the possibility for users to control the application with their voice.

Implementation

The microphone button was hidden from the menu when selecting either a light bulb or a speaker.

Chapter 4
Evaluation

The purpose of the user study was to evaluate AR as a user interface for controlling IoT devices. The IoT devices were controlled with both their native application and the AR application that was created for the purpose of this user study so that they could later be compared against each other. All test participants were provided with the same phone, as well as the same setup of IoT devices so that everyone had the same prerequisites. The goal was to evaluate if the AR application proves to be a more efficient user interface for controlling IoT devices in the room the test participant was present in.

Half of the test participants started testing the AR application, while the other half started with the native applications. By doing that the transfer of learning effect could be avoided. Between these groups, the demographics were divided as evenly as possible.

In section 4.1 the outcome of the user study is evaluated with a focus on the participant information, setup, and procedure. The results of the SUS, NASA-TLX, observations, and interviews are evaluated in section 4.2.

4.1 User study

4.1.1 Participant information

A total of 20 test participants signed up and 30% (6) were women. The average age of the test participants was 26.1 and they had a total of six different occupations. More detail about the different ages and occupations can be found in appendix D in figure D.1 and D.2. All test participants answered that they knew what AR was and that they had used it before. All test participants except one knew what IoT was while five test participants had not used it before.

4.1.2 Setup

All test participants were provided with the same setup of IoT devices and the same phone, which was crucial so that everyone had the same prerequisites. The placement of the different IoT devices was good. They were spread out all over the room and it required the test participant to move around when interacting with the AR application which resulted in interesting feedback evaluated in section 4.2.2. However, there could have been more IoT devices in the room, that would probably provide interesting feedback regarding naming issues in the native applications.

4.1.3 Procedure

Using an orientation script to provide all test participants with the same information before the test was proven to be efficient, none of the test participants had any questions before the test. Having half of the test participants starting with the AR application and half with the native applications was also proven to be good. There was a lot of confusion regarding how you change the color of a light bulb because there is no evident signifier for that. If all test participants had started with the same application(s) the scores would have been unfair as the application(s) tested first would have had worse scores because of the transfer of learning effect when performing the second test. Finally, the interview provided a lot of interesting thoughts and feedback from the test participants that were not expressed during the test. The whole procedure can be seen in figure 4.1.

4.2 Results

4.2.1 SUS

The average SUS score was slightly higher for the AR application compared to the native applications which can be seen in table 4.1. The minimum SUS score was also higher for the AR application while the maximum SUS score was the same for both. The SUS score for each test participant can be seen in figure 4.2. Nine test participants considered the AR application to be better, ten test participants considered the native application the be better, and one test participant considered them equally usable. The AR application was considered to be below average (68) by one test participant while two test participants considered the native applications to be below average.

Table 4.1: Average, minimum and maximum SUS score

	AR	**Native**
Average	85.75	82.375
Min	60	50
Max	97.5	97.5

4.2.2 NASA-TLX

The average NASA-TLX score was slightly lower for the native applications which can be seen in table 4.2. The minimum NASA-TLX score was also lower for the native application while the max-

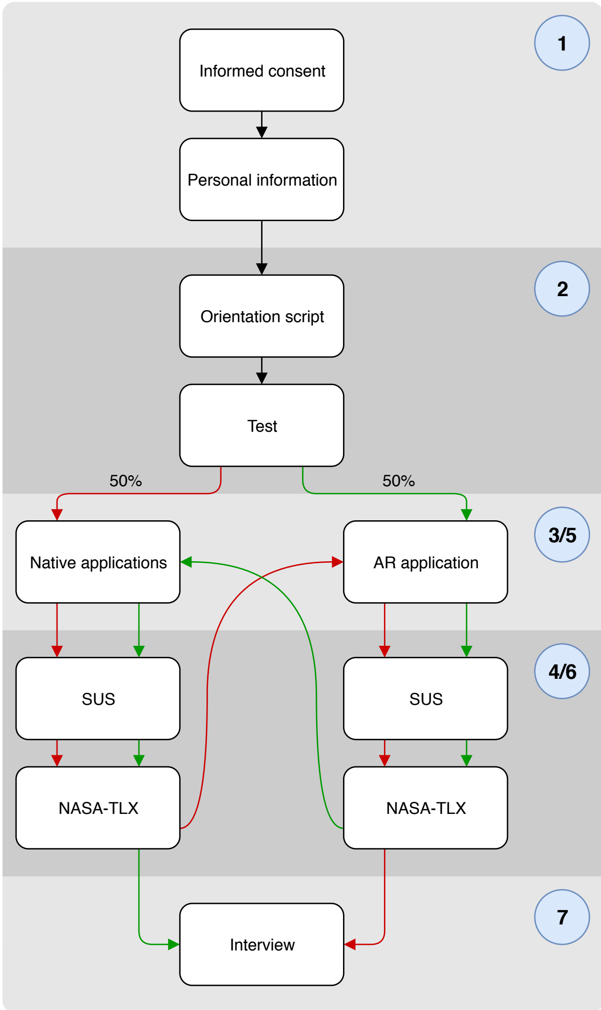


Figure 4.1: Graph over the procedure

imum NASA-TLX score was higher. The biggest difference between the systems was the physical demand where the AR application scored three times as high compared to the native applications. The median, interquartile range (IQR), minimum and maximum NASA-TLX score for each individual question can be seen in table 4.3 and 4.4 where the cells marked green indicate that the system outperformed the other in that question. The NASA-TLX scores for each test participant can be seen in figure 4.3. Six test participants gave a lower NASA-TLX score on the AR application, 13 test participants gave a lower NASA-TLX score on the native applications and one test participant had equal NASA-TLX scores.

Table 4.2: Average, minimum and maximum NASA-TLX score

	AR	Native
Average	100.75	93.25
Min	40	30
Max	190	230

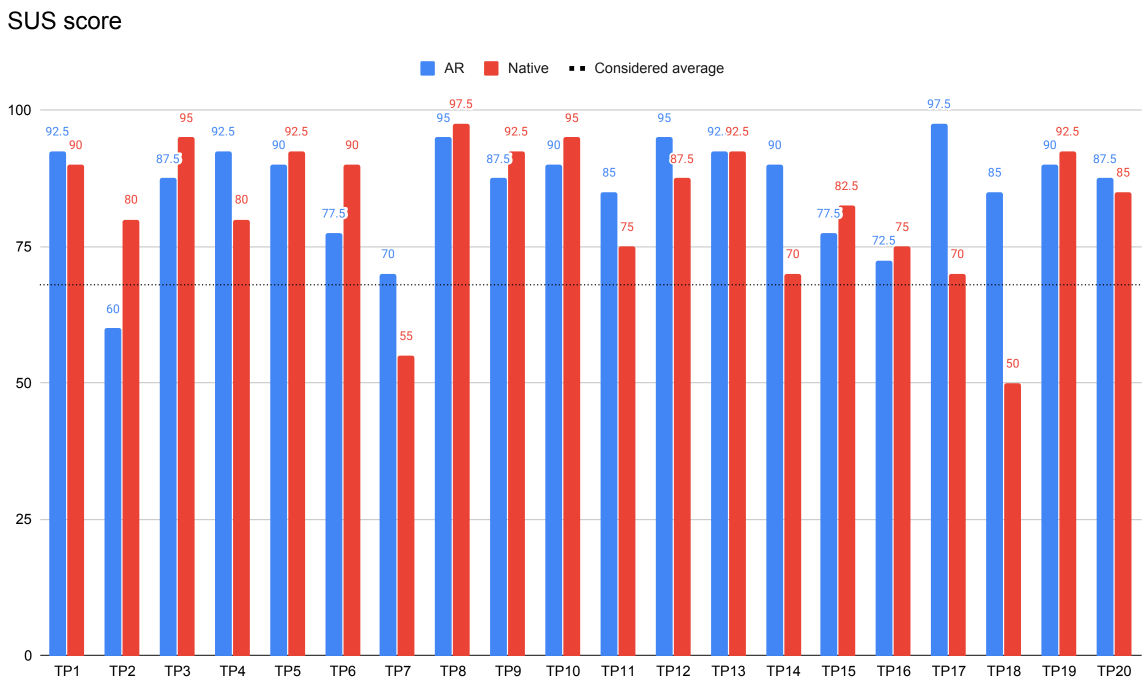


Figure 4.2: SUS score

Table 4.3: AR application median, IQR, minimum and maximum NASA-TLX scores. The cells highlighted in green show where the AR application were considered better

	Mental demand	Physical demand	Temporal demand	Performance	Effort	Frustration
Median	15	15	15	10	15	12.5
IQR	11.25	16.25	12.5	6.25	10	20
Min	5	5	5	5	5	5
Max	30	50	45	50	30	50

Table 4.4: Native applications median, IQR, minimum and maximum NASA-TLX scores. The cells highlighted in green show where the native applications were considered better

	Mental demand	Physical demand	Temporal demand	Performance	Effort	Frustration
Median	15	5	12.5	10	10	10
IQR	11.25	5	10	10	18.75	15
Min	5	5	5	5	5	5
Max	75	25	40	40	70	75

4.2.3 Observations

All test participants completed the tasks successfully, and the test leader only needed to assist one time each for the AR application, native applications, and task 9 that was specific for the AR application. The number of errors, however, differed between the AR application and native applications. Only one test participant performed no errors in the AR application while five test participants performed no errors in the native applications. More details about the number of errors using the AR application, native applications, and performing task 9 can be seen in appendix

 4. Evaluation

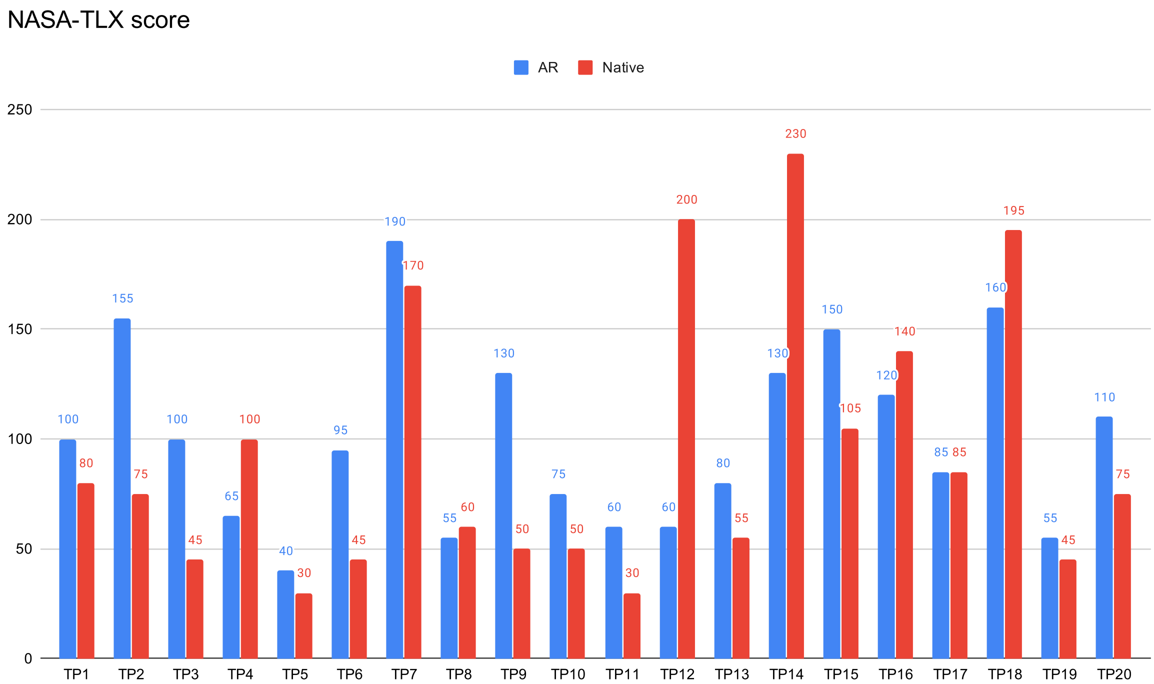


Figure 4.3: NASA-TLX score

E in figure E.1, E.2, and E.3. The majority of the test participants using the AR application had errors on the first task because they could not find the sphere above the light and either tried to click on "Välj en enhet" or press on the light, after that initial confusion the interaction was much smoother and less error-prone which can be seen in figure 4.4 and 4.5.

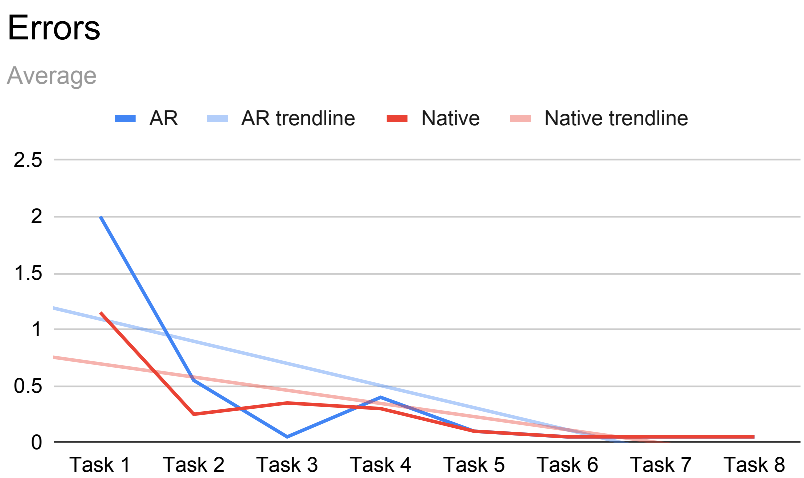


Figure 4.4: Average number of errors for each task

The average number of errors and time performing all tasks in the AR application was also higher compared to the native applications which can be seen in table 4.5 and 4.6. In table 4.7 the average, the minimum and maximum number of errors, assists and time in seconds for task 9 can

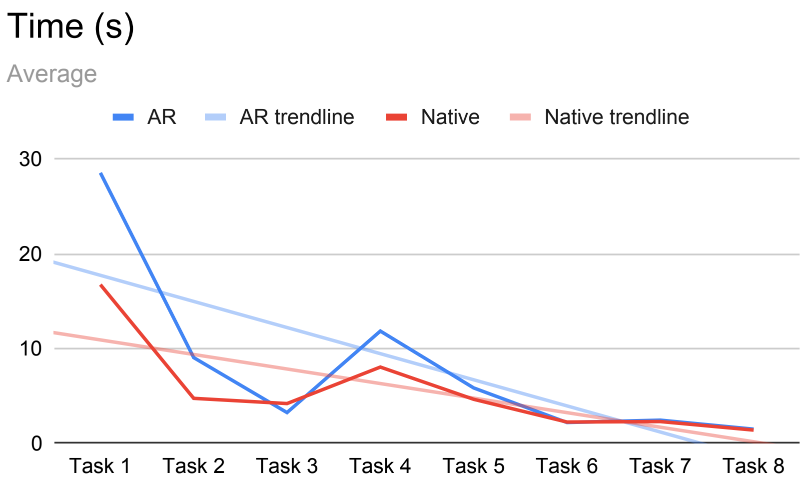


Figure 4.5: Average time in seconds for each task

be seen.

Table 4.5: Average, minimum and maximum errors, assists, and time for the AR application

	Average	Min	Max
Errors	3.25	0	9
Assists	0.1	0	1
Time (s)	64.7	23	123

Table 4.6: Average, minimum and maximum errors, assists, and time for the native application

	Average	Min	Max
Errors	2.15	0	9
Assists	0.1	0	1
Time (s)	46.8	14	106

Table 4.7: Average, minimum and maximum errors, assists, and time for task 9

	Average	Min	Max
Errors	0.3	0	2
Assists	0.05	0	1
Time (s)	12.55	5	29

Observing the test participants it was noticed that 11 test participants had issues figuring out how to change the color of the lights. They clicked around and tried using the brightness slider as they thought it was mapped to colors until they found out how to toggle the color picker menu.

When performing the second test, independent of which system they tested first, they had no issues changing the color of the light. Three test participants thought they had to aim the bullseye at the sphere to control the light and two test participants used the feature to control multiple lights simultaneously in the Philips Hue application which was not supported in the AR application.

4.2.4 Interviews

During the interviews the test participants were asked if they found it useful controlling IoT devices with AR, only one answered no. Test participants were also asked which of the two systems they preferred, the AR application or native applications. Four test participants said they would prefer the AR application, nine said they would prefer the native applications, five test participants said it would depend on the number of IoT devices they had to control (more devices would benefit the AR application more), and two test participants said they wanted a combination of both systems.

19 test participants mentioned that they liked the AR application because everything was shown in its real context. It was therefore easy to see which devices could be interacted with and separate them from each other.

11 test participants said that the AR application would have a bigger benefit over the native applications if there were a lot of devices in the room, making it easier to separate them in the AR application. Although, they raised concerns about controlling different rooms and if the devices were located close or behind each other.

Four test participants said that the AR application would be better if it was hard to separate the different devices by name.

Four test participants said that the AR application would be better if it was outside your home, e.g. in an office.

Five test participants said they liked the feature where you point to interact with an IoT device in the AR application, it felt intuitive and fun and made more sense than scrolling through lists.

Three test participants said that they were more used to traditional light switches and native applications when controlling lights. But they thought future generations would be more used to AR technology which would make it even more useful for that target group.

Three test participants said they did not think mobile AR was the best, the application would be more useful with AR glasses or maybe a designated remote control with a screen designated for AR.

Three test participants suggested that voice would be a good complement to control the IoT devices.

One test participant suggested that adding more information about the IoT devices in the AR view would give more value to the application, e.g. showing the status of each IoT device.

Chapter 5

Discussion

The following chapter discusses the development process and the results of the user study. In section 5.1 the good and bad of the development process are discussed. Section 5.2 discuss and analyze the results of the user study in depth.

5.1 Development process

The implementation of the AR application worked well. The Mid-fi prototype testing conducted during the implementation phase was proven to be crucial. It highlighted numerous issues with the current implementation that had not been noticed otherwise. It could have been beneficial to conduct one more test a couple of weeks before the user study. That would probably have highlighted issues with the sphere above each IoT device not being distinguishable from the background and the issue could have been fixed before the user study. Working iteratively with the AR application was an efficient development process. The AR application was developed with small iterations, one feature was added at a time. It was then easy to evaluate if it worked as intended before continuing on with the next feature.

5.2 User study

5.2.1 SUS

Although the average SUS score of the AR application was higher the difference was not big enough to be considered significant. The reason for the AR application getting a higher average score could be because four test participants scored the native applications to be significantly less useful than the AR application, which was only the case for one test participant for the AR application, as seen in figure 4.2. These differences could be because of the frustration experienced during the test because the applications were used for the first time.

5.2.2 NASA-TLX

Neither do the NASA-TLX scores differ enough to consider one system significantly better than the other. The biggest difference was the physical demand required to interact with the different systems. For the AR application, the median and IQR was three times as high compared to the native applications. This was something that was mentioned by the test participants during the interviews and it is clearly reflected in their answers in the NASA-TLX form. The reason for the big difference was probably because of the motion detector that was placed in the right corner behind the sofa. To execute the task the test participant had to move almost 180 degrees to see the motion detector and select it. This will always be an issue with the current implementation and would be something you would have to get used to, but having a combination of lists and AR could ease the physical demand. Then you could just use the list to interact with the motion detector, without the need for high physical demand.

5.2.3 Observations

The learning curve for the AR application was definitely higher, but the efficiency also increased more in the AR application as the test participants used it more. Almost twice as many errors were made in the AR application and it took almost twice the time to perform the first task compared to the native applications. The majority of the test participants did not see the sphere above the light that they needed to select to control it. If a short introduction had been made or if the sphere color would be the same as the background but inverted this could have been avoided and the two systems would probably have similar averages for the first task. The text "Välj en enhet" could also be replaced with something more explanatory like "Markera en enhet". As seen in figure E.1 and E.2 the rest of the average of tasks 2-8 did not differ as much as in task 1. There is a peak at task 4 and that was probably because the task itself was hard to understand and had to be repeated a couple of times. Changing the color of the light was shown to be an issue for more than half of the test participants (11). This was because of the absence of clear signifiers for changing the color of the light. The AR application intentionally copied the design of the Philips Hue application to get a fair comparison, so the way you change a light color could definitely be made clearer. It could be done by adding a color palette button of some kind, giving the user a clear signifier that the user should press there to change the light's color. A few test participants (3) did not understand that you do not need to aim the bullseye at the sphere to control the device, which increased the physical demand for performing the tasks. This is probably something the users will learn as they use the application multiple times. Lastly, two test participants used the feature of controlling multiple lights at once. This was not integrated into the AR application because of the time limit, but it could definitely be added.

5.2.4 Interviews

A clear majority thought it was useful to control IoT devices in AR, just one said that it was not useful, this shows that the feature could be useful. Four test participants also said they preferred the AR application over the native application while nine said the opposite. Five test participants said that it depended on the number of devices in the room and two wanted a combination. With the increasing number of IoT devices in our homes, it could be assumed that there would be enough devices in the room for the five test participants to prefer the AR application. This means that 11 test participants would want to use AR to control their IoT devices. All test participants except one also noted that it was nice to see everything in its real context, which was the main feature of the AR application. The raised concern about controlling different rooms is a valid one, it is hard to visualize how that would be done in AR without using lists. Devices located close and/or behind each other is also an issue that needs to be solved, however, that is an issue in the native applications, as well as they, would probably be hard to separate by name. There is also great potential in using the AR application outside your home. In a native setting, you will not have any idea of where everything is located and it would probably be easy to choose the wrong device to control. With an AR application you would see everything in its real context and it would be very easy to see which device is which. Using other tools for the AR application, e.g. AR glasses could also be beneficial. If they are developed to be more mobile and look similar to normal glasses it has the benefit of always being on and accessible. Then the user can simply look at e.g. a light bulb and tell it to change its brightness to 100%. That would probably be more efficient and quicker than the current implementation as well as the native applications, you just have to solve the issue regarding interaction with the devices once they are selected. Voice could definitely be used but you might want some gesture as a compliment as well, and it is hard to imagine what gestures future AR glasses can offer.

Chapter 6
Conclusions

The goal was to evaluate if AR technology has come far enough to be applied to IoT devices, how intuitive it is to control IoT devices in AR compared to the native applications, and if AR technology is a user interface that could be preferred over traditional ways of controlling IoT devices. The results from the SUS score show that the AR application was considered more useful, but not by a lot. The NASA-TLX, however, showed that the workload of performing the tasks was considered higher using the AR application, but not by a lot neither. The higher workload in the AR application was mainly because of the high physical effort required to perform the tasks, compared to the native applications. So AR technology has come far enough to be applied to IoT devices and it is intuitive to use. The scores were similar but the AR application was compared to native applications that have been developed over several years while the AR application was developed during a few weeks. With that in mind, AR technology could be a user interface preferred over traditional ways of controlling IoT devices, but the current implementation is not there yet.

To sum it up, three key things can be taken from this thesis: (1) it must be clear what the user is supposed to interact with, (2) the physical effort was higher than expected, and (3) mobile phone is probably not the best use case.

References

[1] Günter Alce, Andreas Espinoza, Ted Hartzell, Staffan Olsson, Dennis Samuelsson, and Mattias Wallergård. UbiCompass: An IoT Interaction Concept. *Advances in Human-Computer Interaction*, 2018, 2018.

[2] Günter Alce, Maximilian Roszko, Henrik Edlund, Sandra Olsson, Johan Svedberg, and Mattias Wallergård. [POSTER] AR as a User Interface for The Internet of Things—Comparing Three Interaction Models. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 81–86. IEEE, 2017.

[3] Apple. Speech. `https://developer.apple.com/documentation/speech`. Accessed in 2020-01-22.

[4] Mattias Arvola. *Interaktionsdesign och UX: om att skapa en god användarupplevelse*. Studentlitteratur AB, 2014.

[5] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[6] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[7] Alex Cao, Keshav K Chintamani, Abhilash K Pandya, and R Darin Ellis. NASA TLX: Software for assessing subjective mental workload. *Behavior research methods*, 41(1):113–117, 2009.

[8] Interaction Design Foundation. User Centered Design. `https://www.interaction-design.org/literature/topics/user-centered-design`. Accessed in 2020-01-10.

[9] David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. Proxemic-aware controls: Designing remote controls for ubiquitous computing ecologies. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 187–198, 2015.

[10] Donald A. Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, Inc., USA, 2013.

[11] Sarah Perez. ARKit-only apps top 13 million installs, nearly half from games. https://techcrunch.com/2018/03/28/arkit-only-apps-top-13-million-installs-nearly-half-are-games/. Accessed in 2020-01-21.

[12] Margaret Rouse. ARKit. https://whatis.techtarget.com/definition/ARKit. Accessed in 2020-01-21.

[13] Margaret Rouse. Internet of things (IoT). https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT. Accessed in 2020-01-21.

[14] Phil So. NASA TLX: TASK LOAD INDEX. https://humansystems.arc.nasa.gov/groups/tlx/. Accessed in 2020-01-21.

[15] usability.gov. Prototyping. https://www.usability.gov/how-to-and-tools/methods/prototyping.html. Accessed in 2020-01-21.

[16] usability.gov. System Usability Scale (SUS). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. Accessed in 2020-01-20.

Appendices

Appendix A
Test plan

A.1 Purpose

The purpose of this test is to evaluate AR as a user interface for controlling IoT devices. Tests will be performed at the Malmö Jayway office in the conference room The Pub. Test participants will perform a set of tasks where they control and check the status of different IoT devices. The devices included in this test are three Philips Hue Color light bulbs, a Sonos Play:1 speaker, and a motion detector. The IoT devices will be controlled with both their native application and the AR application that was created for the purpose of these tests. All test participants will be provided with the same phone, as well as the same setup of light bulbs, speakers and motion detectors. The goal is to evaluate if the AR application proves to be a quicker user interface for controlling IoT devices in the room the test participant is present in.

Half of the test participants will start testing the AR application, while the other half will start with the native applications. Between these groups, the demographics will be divided as evenly as possible.

A.2 Test participants

The target group for this test is anyone with the need of controlling IoT devices in their home. Therefore the test participants can be widely chosen. Altogether, 20 test participants were selected. The test participants will primarily be chosen from the following groups:

- Employees at Jayway both with and without prior knowledge and experience of AR technology.
- Friends both with and without prior knowledge and experience of AR technology.
- Voluntary testers with and without prior knowledge and experience of AR technology.

A.3 Test data

Objective, demographic, and subjective data will be gathered:

- Objective data
  - Task completion, true or false.
  - Errors, the number of errors the test participant did.
  - Assists, how many times the test leader had to assist the test participant.
  - Elapsed time, measured in seconds.
  - Analysis of behavior and execution during the test.
- Demographic data
  - Personal information form
- Subjective data
  - SUS
  - NASA-TLX

A.4 Tasks

The tasks that will be performed during the tests by the test participant can be seen in table A.1.

Table A.1: Test participant tasks

#	Task	#	Subtasks	Finished when	Time cap
1	Turn on the light	1.1 1.2 1.3	Locate the light Aim the bullseye at the light Press the light switch	The light is turned on	2 min
2	Change the lights color to blue	2.1 2.2	Open the color picker menu Select the color blue	The light is blue	2 min
3	Change the brightness to around 50%	3.1 3.2	Dismiss the color picker Drag the brightness slider to around 50%	The light is on around 50% brightness	2 min
4	Change the green light to blue	4.1 4.2 4.3 4.4	Locate the green light Aim the bullseye at the green light Open the color picker menu Select the color blue	The light is blue	2 min
5	Start playing music on the speaker	5.1 5.2 5.3	Locate the speaker Aim the bullseye at the speaker Press the play button	The speaker is playing music	2 min
6	Play the next track	6.1	Press the next track button	The speaker is playing the next track	1 min
7	Set the speaker volume to half	7.1	Drag the volume slider to half the volume	The speaker volume is half	1 min
8	Pause the music	8.1	Press the pause button	The music is paused	1 min
9	Check if the motion detector registered motion yesterday	9.1 9.2 9.3	Locate the motion detector Scroll through the list Find the log event for yesterday	The test participant says yes	3 min

A.5 Orientation script

An orientation script is used to describe what will happen during the test session and is intended to put the test subjects at ease. The test leader explains to the test participant what it is they will be doing and that it is the product that is tested, not them. The orientation script was used to ensure that all test participants had the same information before starting the test cases. The orientation script can be seen below:

> Hi and welcome to this test session. I am your test leader during this test and will now read from a manuscript to ensure that all test participants get the same information before the test. During the test, which will last for around 30 minutes, you will interact with an AR application, a Philips Hue application, and a Sonos application. If you do not know what AR is, Pokémon Go is a famous example. It is a technique where you put virtual elements in the real world. With the applications, you will control different IoT devices, which are devices connected to the internet.
>
> The IoT devices that you will interact with are the lights located at the side table, by the chair, in the corner on the barrel, the speaker behind the chair, and the motion detector in the corner.
>
> During the test, you are encouraged to think aloud and express what you feel and experience during the test. I want to point out that it is the different systems that will be evaluated and not you. I will read a couple of scenarios that will be your starting point for interacting with the AR application. The test session will be timed, so perform the tasks as efficiently as possible, but do not feel stressed. Feel free to ask me to repeat the scenarios if there is something unclear. If you have any questions now or during the test, just ask me.

A.6 Task scenarios

To give the participants a context to the tasks they were about to perform, different scenarios were presented for each assignment:

1. You just got home and landed on the sofa. You realize that the light on your side table is shut off. Use the phone you have in front of you and turn the light on with the help of the open application.
2. The color yellow is not your favorite color, change the light bulb color to blue.
3. The light bulb is a bit bright for your taste, set the brightness to 50%.
4. You want all the lights in the room to match, turn the green light blue.
5. You are in the mood for some music, start playing music from the speaker.
6. You spent the whole day yesterday listening to Queen, skip to the next track.
7. Now we are talking! But you can not hear yourself thinking. Set the speaker volume to half.

8. Enough music, for now, you can pause the playback.
9. This setup of IoT devices took a long time to achieve and cost a lot of money. To make sure no one steals your invaluable devices you have set up a motion alarm. Check if someone was in the room yesterday.

A.7 Roles

The tests are held by myself and therefore I had the roles of both test leader and observer.

A.8 Equipment and testing environment

The test was conducted on an iPhone 11 running iOS 13.2.3. The iPhone had the AR application, Philips Hue application and Sonos application installed as the process of installing an application on an iPhone was not included in the test. The phone and IoT devices were all connected to the same WiFi beforehand so that they could interact with each other. The room the tests were performed in was a conference room at the Jayway office in Malmö called The Pub. In the conference room three Philips Hue Color light bulbs were setup as well as a Sonos Play:1 speaker. A motion detection alarm was also present in the room but it was not connected to the system, the app just showed static data about it. Before each test the environment was set to a default state:

1. The light on a side table beside the sofa was turned off with brightness 100% and the color yellow as seen in figure A.1.
2. The light on a barrel was turned on with brightness 50% and the color green as seen in figure A.2.
3. The reading light above an armchair was turned on with brightness 50% and the color blue as seen in figure A.3.
4. The speaker on a stool was paused and on the first track of a custom playlist with the volume set to 30% as seen in figure A.4.
5. The motion detector in a corner with static data showing events from today and yesterday as seen in figure A.5.

Before each test, the application that the test participant was supposed to interact with was opened by the test leader. The AR application was on its initial screen without any devices selected which can be seen in figure A.6, the Philips Hue application was on its Home screen which can be seen in figure A.7, and the Sonos application was on its My Sonos screen which can be seen in figure A.8.

All tests were recorded using the screen recording feature of the phone the tests were performed on as well as the microphone to pick up voice.

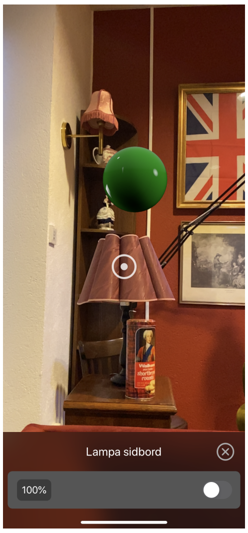


Figure A.1: Light on side-table

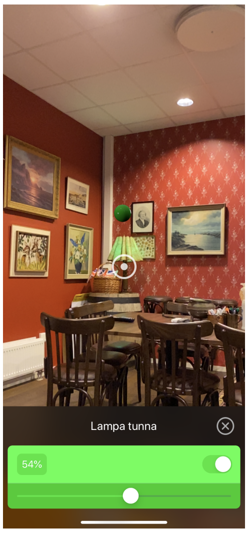


Figure A.2: Light on the barrel

A.9 Procedure

1. **Formalities:** The test leader was responsible for getting the test participant in the test room. All participants were given contact information to the test leader and were encouraged to contact the test leader when they arrived at the office. Once the test participant arrived at the test room they were asked to sit down on the sofa while the test leader sat on a chair beside the sofa. The test participant was then asked to fill in informed consent and a personal information form.

2. **Introduction:** The orientation script was read for the test participant to explain the background and purpose of the study.

3. **Test 1:** The test leader read the task scenarios, waiting for each to finish before reading the next one. The test leader kept track of time limits, task completion, errors, elapsed time,

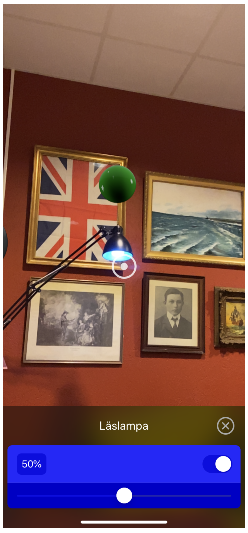


Figure A.3: Reading light

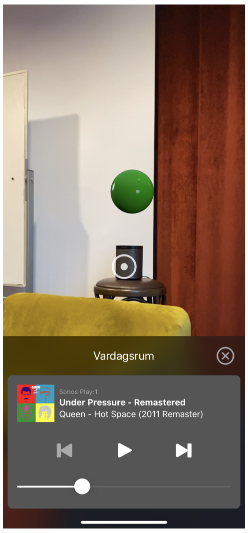


Figure A.4: Speaker on stool

and assisted the test participant if needed. Either the AR application or the Philips Hue and Sonos applications were tested.

4. **Questionnaires test 1:** The test participant was asked to fill in a SUS and a NASA-TLX.

5. **Test 2:** The application(s) that were not tested in the first test was now tested in the same fashion as above.

6. **Questionnaires test 2** An SUS and a NASA-TLX was filled in for test 2 as well.

7. **Interview:** A semi-structured interview took place. The test participant got to discuss the test freely.

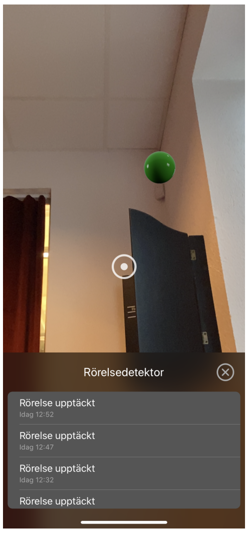


Figure A.5: Motion detector in the corner

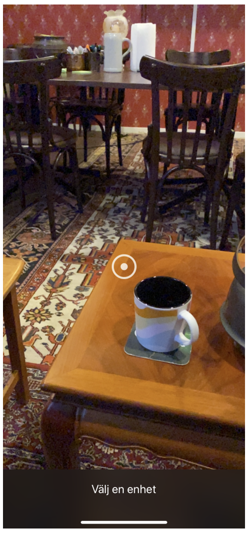


Figure A.6: AR application start screen

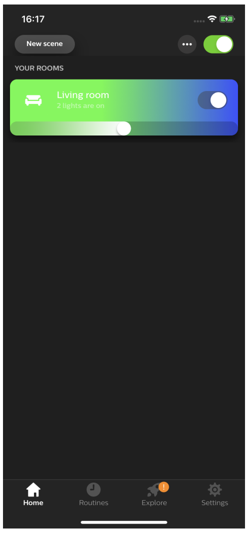


Figure A.7: Philips Hue application start screen

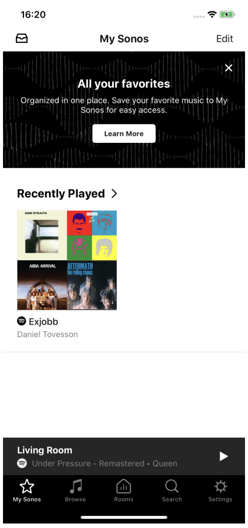


Figure A.8: Sonos application start screen

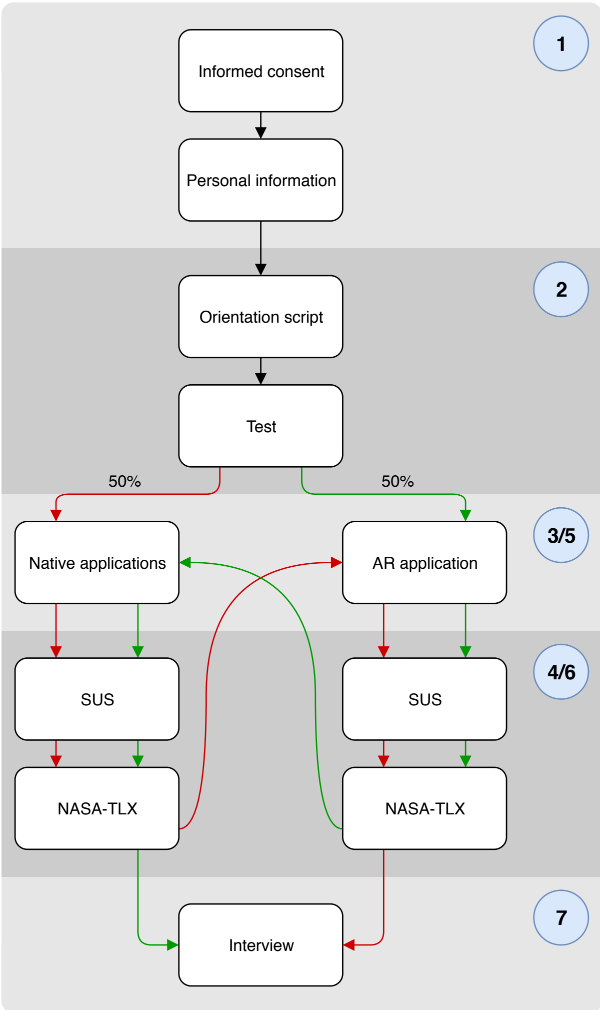


Figure A.9: Graph over the procedure

Appendix B
Interview questions

B.1 Questions

The semi-structured interview that was performed after each test can be found below.

1. What did you think about the overall interaction of the systems?
2. Did you find it useful to control the IoT devices in AR?
3. What features were most valuable for you and why?
4. Which of the two systems that you tested did you prefer?
5. Do you have any general thoughts about using AR for controlling IoT devices?

Appendix C
Forms

All test participants were asked to fill in two forms before the test, one for informed consent and one for personal information. The informed consent form can be seen in section C.1 and the personal information form can be seen in section C.2.

After each test, the test participants were asked to fill out two more forms, one for SUS and one for NASA-TLX. The SUS form can be seen in section C.3 and the NASA-TLX form can be seen in section C.4.

C.1 Informed consent

Participant ID:

Informed consent

Hi and welcome to this test session. During the test, which will last for around 30 minutes, you will interact with an Augmented reality (AR) application, a Philips Hue application, and a Sonos application. If you do not know what AR is, Pokémon Go is a famous example. It is a technique where you put virtual elements in the real world. With the applications, you will control different Internet of Things (IoT) devices, which are devices connected to the internet. The test session will be timed, so perform the tasks as efficiently as possible, but do not feel stressed.

Participating in this test session you agree to the following:

- The phone you are using during the test will have its screen recorded.
- The whole test session will be recorded with sound.
- Observations during the test session will be noted.
- In the report, you will be mentioned by a unique test ID, not your name.
- You can abort the test at any time without any specific reason.

By signing you agree to the above terms.

Name

Place and date

Signature

C.2 Personal information

Participant ID:

Personal information

Gender

☐ Male

☐ Female

☐ Other

☐ Prefer not to say

Age

Occupation

AR experience

Do you know what Augmented reality (AR) is?

☐ Yes

☐ No

Have you used AR on a mobile device before?

☐ Yes

☐ No

Do you know what Internet of Things (IoT) is?

☐ Yes

☐ No

Have you used IoT before?

☐ Yes

☐ No

C.3 SUS

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

Please provide any comments about the system:

C.4 NASA-TLX

Mental demand — How mentally demanding was the task?

☐ — You must choose one option

Very low … Very high

Physical demand — How physically demanding was the task?

☐ — You must choose one option

Very low … Very high

Temporal demand — How hurried or rushed was the pace of the task?

☐ — You must choose one option

Very low … Very high

Performance — How successful were you in accomplishing what you were asked to do?

☐ — You must choose one option

Perfect … Failure

Effort — How hard did you have to work to accomplish your level of performance?

☐ — You must choose one option

Very low … Very high

Frustration — How insecure, discouraged, irritated, stressed and annoyed were you?

☐ — You must choose one option

Very low … Very high

Appendix D

Participant information

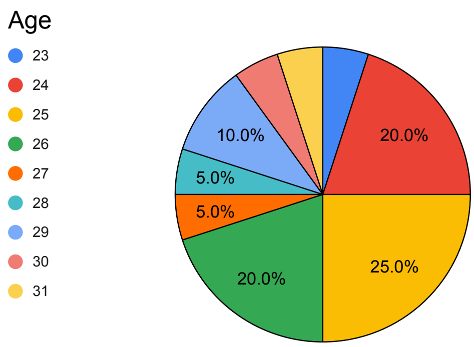


Figure D.1: Test participants age

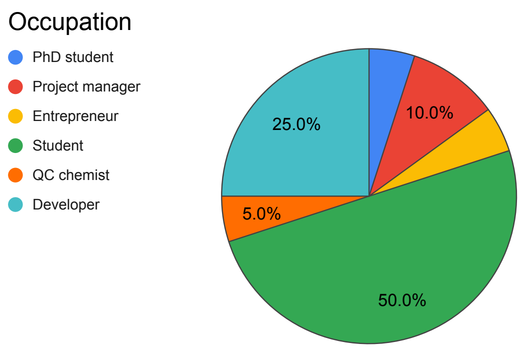


Figure D.2: Test participants occupation

Appendix E

Observations

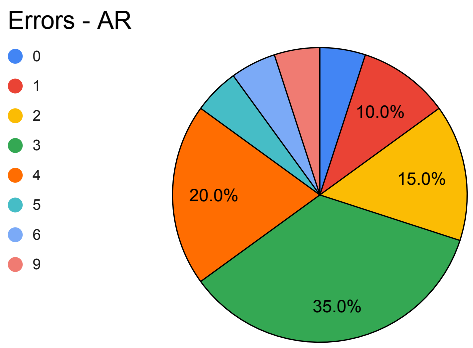


Figure E.1: Errors using the AR application

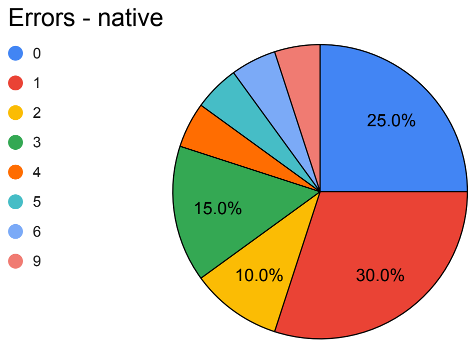


Figure E.2: Errors using the native applications

E. Observations

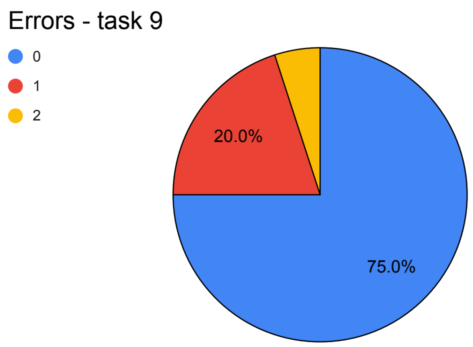


Figure E.3: Errors performing task 9