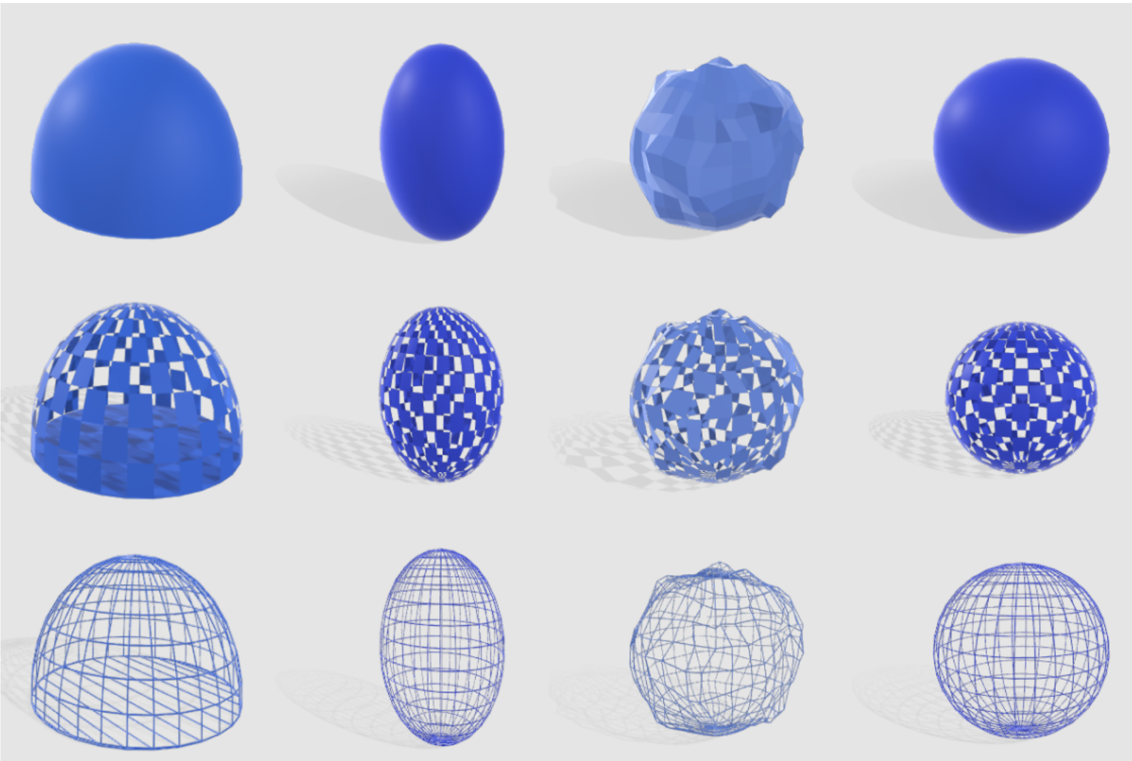
An Augmented Reality Android Application For Interaction with Sound Zones



VGIS
Faculty of IT and Design
Aalborg University, Denmark
June 2, 2021

10th Semester - Masters Thesis
Vision, Graphics and Interactive Systems
Fredrik Bajers Vej 7
9220 Aalborg

Title:

An Augmented Reality Android Application For Interaction with Sound Zones

Project:

Master's Thesis

Theme:

Augmented Reality
and Interactive Systems

Project period:

Spring Semester 2021

Project group:

Group 1043

Participants:

Andriy Bogdanov
Márton Havasi
Oliver Hansen

Supervisors:

Lars Bo Larsen

Page count: 125 pages
Hand in date: June 2, 2021

Abstract:

This project aims to provide a mean of interaction with sound zones through Augmented Reality (AR). To realise this task an Android mobile application has been developed. The underlying problem associated with the sound zone interaction has been established to be the hidden affordance of the sound zones. To address this problem it has been suggested to utilise AR to create the sound zone visualisers with an option for user interaction. A research into the interaction methods with AR has been conducted, where touch-screen gestures have been established as a tool to achieve the interaction goal. This report provides a thorough description of the Design process which elaborates on the design choices through development. The user interaction options have been outlined and a set of requirements has been established. Several candidates for the sound zone visualisations have been created via the GV Design Sprint method, where the exact features and properties of the proposed solution have been described. The design sprint resulted in 3 visualisation concepts and 4 different shapes the sound zones could take on. The Implementation part of this report has thoroughly described the architecture of the system along with the usage of ARCore and Sceneform to implement AR interaction. The usability of the application and the designed visualisations has been tested, where test participant feedback has been implemented in the second iteration of implementation. Based on the results it can be concluded that the developed application allows for interaction with sound zones through AR.

This Master's thesis project in Vision, Graphics and Interactive Systems, IT & Electronics - Faculty of IT and Design, is written and edited by group 1043. The group consists of Oliver Hansen, Andriy Bogdanov and Márton Havasi. The report is made in the time period from February 2021 to June 2021.

The references are cited with the use of The American Institute of Physics (AIP) referencing method, which means that sources will be cited as "[n]", where n is the order of appearance. In the Bibliography, they will be listed in the same manner. Illustrations will be labelled under the chapters they belong to and not the different sections they are included in. As an example, the first figure in Chapter 1 will be labelled "Figure 1.1", followed by 1.2, 1.3, etc. Tables and Equations follow the same principles.

The source code for this project can be found at the following Github repository: `https://github.com/oeth16/sceneform-android-sdk-master`. The repository contains the Android Studio project used to develop the solution as well as the complied Android Application Package(APK) which can be downloaded to install the application.

The authors would like to specially thank Thomas Gorisse for maintaining a open source fork of Sceneform Android SDK which can be found here:`https://github.com/ThomasGorisse/sceneform-android-sdk`

Table of Contents

1 Introduction

When people share a space it can often be desired to listen to a sound source in private. An example of this would be a medium sized office where one of the employees wants to listen to music, but the others in the office would like to work without listening to music. Currently the solution to this problem is to use headset or earphones[1] such that the employee can listen to music in private without disturbing the others in the office. The same can be applied to other shared spaces such as hospital wards or living rooms, where one of the residents would like to listen to music without disturbing the other residents. For example in a domestic environment such as the living room could be the following: two people are sitting in a living room, they are attending two different online lectures and do not want to disturb each other. The solution to this is that they are both using a headset when listening to the lecture.

Now, the two people sharing the living room are no longer attending a lecture, instead they are engaged in recreational activities. One person is listening to their favourite album, while the other person is watching the news on the TV. In this scenario the person listening to music is wearing headphones, and this allows both of the people to listen to their desired media in private. However if the person watching the news would like to engage in conversation with the other person about something which was just featured on the news, this would be difficult as the other person is wearing headphones and might be difficult to reach. This either requires yelling, waving or some other means of getting the person's attention so they can remove the headphones before conversation can be engaged. This is the general issue with using headphones and earplugs to remove the unwanted sounds in a room [2]. Those solutions solve a specific problem but limit the interaction between the people in the room [2].

To solve this problem sound zones could be used. The concept of sound zones is that a specific sound source can be played in a specified area allowing those inside the area listen to the media [3]. In Figure 1.1 a living room is seen with three different sound zones, each zone able to play different media content allowing the people inside the zones to listen in private without disturbing anyone. People inside or outside a zone can easily interact and talk to people in the zones as they would still be able to hear the people outside of the zone, unlike when wearing headphones or earplugs.

Sound zones can be used in many different settings such as offices[5], hospitals[2] and domestic environments[6] where the later will be the primary focus of this report.

For a user the biggest challenge when interacting with sound zones is that sound inside a sound zone does not conform to the preconceived notion of how sound works when emitted from a regular speaker. The regular speaker emits sound around itself, thus filling the entire space with sound, while the sound zones are perceived as separate regions of space where the sound can be perceived. Because of this difference, the perception of sound zones is a unique experience, and it cannot be associated with the regular sound speaker experience. [2]

Along with this, sound zones also have spatial characteristics such as size and position which the

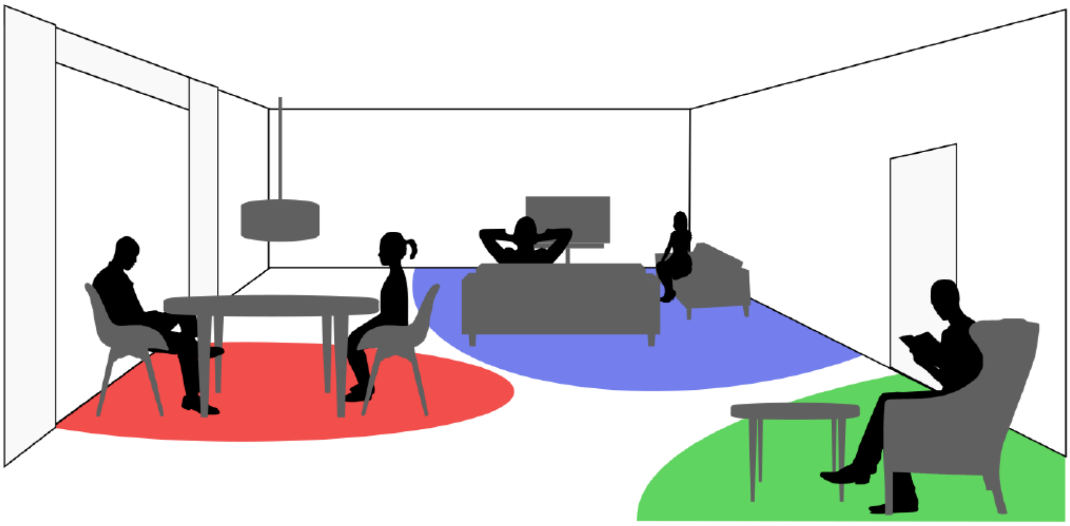


Figure 1.1: *Sketch illustrating the concept of several sound zones in a domestic environment. [4]*

user must be made aware of to facilitate intuitive interaction with the zones [2]. One possible means of facilitating this is through the use of Augmented Reality(AR)[7] to visualise the sound zones.

1.1 Initial Problem Statement

Sharing a space while wanting to listen to a sound source in private can cause people to isolate themselves from interaction with the people they share the space with. A proposed solution to this problem is the concept of sound zones. This solution faces several issues which must be dealt with before it can be considered a suitable solution to the problem. These issues relate to the characteristics of sound zones and how they do differ from the user preconceived notion of sound behaviour. To overcome these issues visualisation of the sound zones using AR is proposed. An initial problem statement has been formulated to help direct the problem analysis which will focus on the sound zone technology and how to interact with sound zones using AR visualisation. This leads to the following initial problem formulation and a set of resulting research question which are to be investigated.

How can Sound Zones be visualised in a domestic environment using Augmented Reality?

Research Questions

- What are the factors which have an impact on the setup of the sound zones?
- How much information has to be conveyed to the user when visualising a sound zone?
- Should the visualisation of a sound zone change depending on the specific use case?
- Is the usage of Augmented Reality to visualise sound zones better suited for specific use cases?
- What are the interaction possibilities while using AR technology? Does it differ from conventional interaction principles and guidelines?

Part I

Problem Analysis

2 | Sound Zones

In this project the concept of the sound zones refers to the effect created when the sound field is reproduced through space to a desired location using sound control strategies to localise the playback signal. The result of this event is an area in space, where the playback signal is reproduced. As seen in Figure 1.1, the sound zone resembles a region in space with a distinct playback signal broadcast to it. [4]

2.1 Sound zone setup

It is generally impossible to create a sound zone by using a single speaker. This is due to the nature of sound propagation, which means that the desired playback signal in the sound zone will not match the original playback signal broadcast from the single loudspeaker. In this project it is assumed that the sound zones are created via the method called beamforming. Beamforming is a method of signal processing where the signal, in this case sound, is transmitted in a specific direction. This is done with a set of loudspeakers, where the spatial separation between the loudspeakers can be used to filter out the interfering signals of no interest, such as noise, and boost the desired signal through superposition. [8]

In the loudspeaker array, each of the speakers transmits the signal with it's own phase and amplitude so that the constructive and destructive interference can be achieved. A sketch of the beamforming loudspeaker array system can be seen in Figure 2.1. [8]

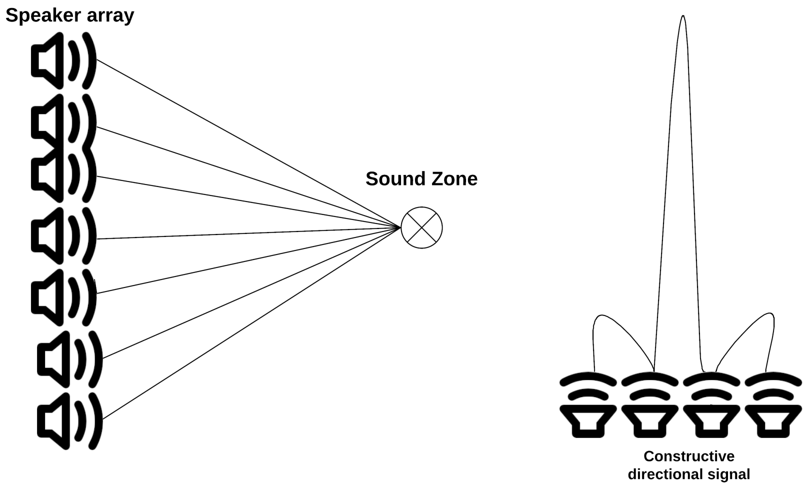


Figure 2.1: *An example of beamforming system.*

As can be seen in the Figure 2.1, the constructive signal interference results in the increased strength of signals emitted from the speaker array pointing to a uniform direction, while the destructive interference results in dampened signal pattern. The focus point for the speakers is referred as a control point.[8]

2.1.1 Room acoustics

Another important factor used in creation of the sound zones is the room acoustics. The sound propagation in the environment is affected by the layout and the boundaries of the given environment. The room layout and the reflective properties of the surfaces of the environment need to be taken into consideration while creating sound zones. The strategies which are used to minimise the effect of the environment on the sound zones are called the room compensation strategies. These strategies are generally revolved around a sound field at either a specific point in space, a partitioned region of the room, or a global sound field control of the entire environment. [9]

The acoustic contrast control is a method of creating sound zones where the goal is to create a sound zone with a contrast to the soundscape in the environment. The method revolves around maximisation of the ratio between the sound zone and the rest of the control zone. [9]

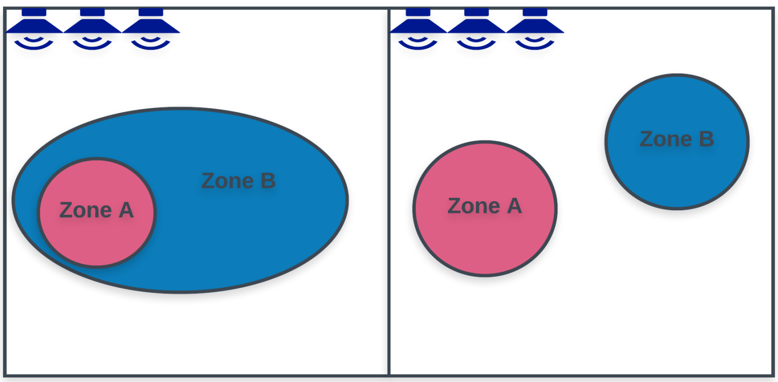


Figure 2.2: *Acoustic contrast control*

In the Figure 2.2 the acoustic contrast control is illustrated. In the left part of the picture the control points are located inside the zones A and B, but the zone B is expanded to limit the overall sound pressure in the room with respect to zone A. Alternatively, in the right side of the picture, the proportion of sound pressures between the discrete sound zones is maximised, thus allowing for formation of multiple sound zones in the environment. [9]

2.1.2 Sound zone control strategies

The difficulties in controlling the sound zones arise from the size of the audible frequency range being 20 Hz to 20 kHz, hence the soundscape in the environment changes while frequency is changed. In the work done by Møller[4] the methods of sound zone control have been investigated where the approach to the control of the sound zones can be seen in Figure 2.3.

At low frequency ranges the working approach is to target the control point at the local level, where a specific point in the sound zone space is selected. At the medium frequency ranges the control revolves around using the beamforming. At the high frequency ranges the inherent directivity of the speakers can be utilised. [4]

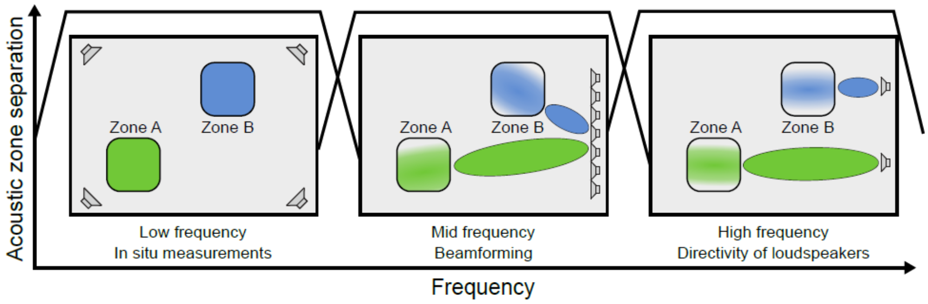


Figure 2.3: *Sound zone control strategies at different frequencies [4]*

2.2 Soundscapes in the domestic environment

The focus of this project is the sound zone interaction within the domestic environment. Inside the household the possible soundscapes can vary largely due to the large diversity of people preferences, therefore it is impossible to list all the potential sound zone types there could be. However, it is possible to group the sound zone setups according to the user needs. The study conducted by Lundgaard et al[6] has determined how sound zone technology can be related to the domestic environments. The authors of the paper establish four types of situations where the sound zone technology can be grouped to. These types are such as follows:

- Social - connected
- Private - connected
- Private - separated
- Social - separated

Social - connected
The social - connected category is suitable for scenarios when there are several people in an environment, each is involved with their own activity, however there is a desire to relate and communicate with each other in non-intrusive manner. The expression for communication is a core factor, yet it needs to be conveyed in a manner which is not disruptive to the individual sound zones soundscape.[6]

Private - connected
Similarly to the first category, the private - connected category shares the similar type of scenario where several people share the common environment, however the desire to experience personal soundscape is more apparent compared to the social - connected type. Participants who described the private - connected category express the desire to relate to the other people in the room, but at the levels which do not disrupt a personal sound zone experience. [6]

Private - separated
Private - separated category resembles do-not-disturb scenarios, where people sharing the environment have a need to experience their soundscape privately in order to either not disturb the other people or not to compromise their own experience for the sake of other people. Without the sound zone technology a compromise might be illustrated with the usage of headphones to experience own soundscape. It is acceptable, yet it delimits the potential of an experience compared to using the speakers, as in a case of watching a movie. [6]

Social - separated

The social - separated category is suitable for the scenarios where the people sharing the environment would like to adjust the volume or the contents of the soundscape in the environment. This can be illustrated with preference for the loudness of the broadcast music in the sound zone: some people prefer to play it louder compared to others. This category is desirable in the scenarios when people are trying to determine the satisfactory soundscape environment where their own needs are fulfilled, while doing the same social activity. [6]

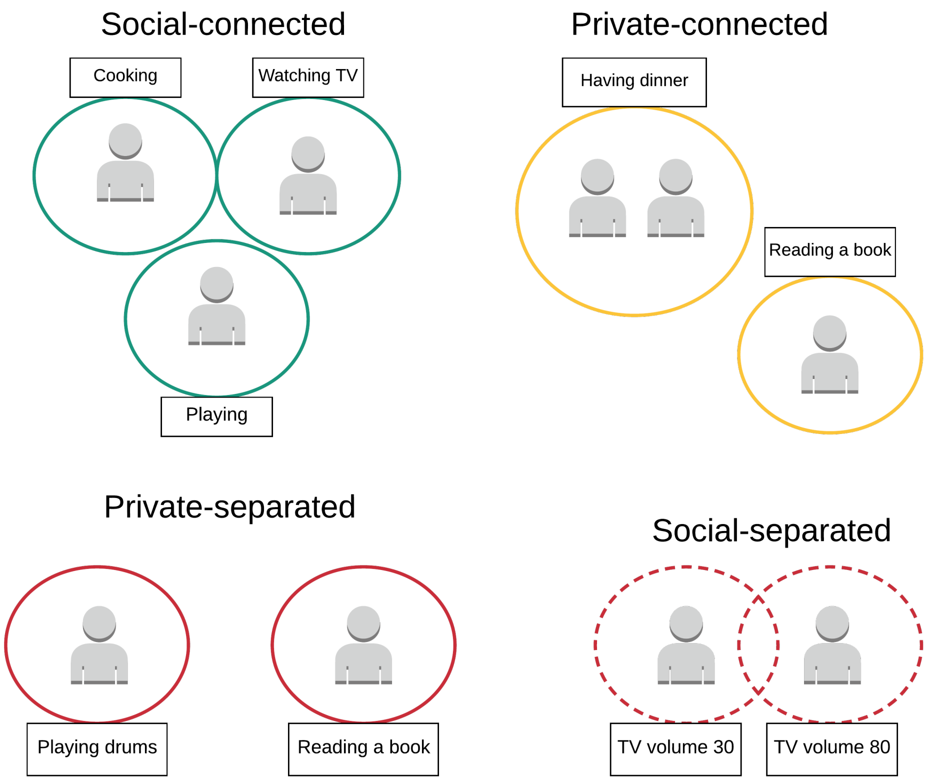


Figure 2.4: *Four categories of domestic soundscapes adapted from Lundgaard et al. [6]*

In the Figure 2.4 the four categories are illustrated with the example of activities which are suitable for these scenarios. These four quadrants provide a possibility of more general approach in establishing the relevant soundscape scenarios for the domestic households, as they encapsulate the essential motivations of the people interacting with the sound zones. Despite being different in settings and needs, these four categories share the same characteristic - the need to perceive the desire of people inside and outside the sound zones. Therefore there is a requirement to accommodate the communication need of the people who are sharing the soundscape. [6].

Apart from the described scenarios, there is an additional type of a soundscape, which is the transitional or dynamic sound zone scenario. In the cases when a person needs to relocate away from the sound zone space, they might want not to disrupt the audio feed from the sound zone, instead take the sound zone experience with them. The transitional sound zone can be categorised within the established four types shown in Figure 2.4.

2.3 Summary

This project aims to realise this requirement through the means of using augmented reality to visualise the sound zones. The framework introduced by Lundgaard et al.[6] will be adapted in this project where the described use cases will be used as a template for grouping the sound zones. In order to understand the user needs when interacting with sound zones, research into affordance of sound zones and interaction methods regarding augmented reality will be conducted in the following chapters.

3 Affordance of Sound Zones

Interacting with sound zones can be a complicated task due to their behaviour which differs greatly from the preconceived notion of how sound works as described in Chapter 1. In this chapter the underlying issue causing interaction with sound zones to be difficult will be investigated and it will be how visualisation of the sound zones in AR can solve this issue. Following this the affordances of sound zones will be explored and analysed with regard to general use cases described in Section 2.2. By establishing the affordances for a set of overall use cases it can be determined how to visualise the sound zones such that they reflect the underlying affordances.

3.1 Interacting with sound zones

In a paper by *Wensveen et al.*[10] a framework was proposed for coupling the action and function of users through feedback and feedforward. This framework can be used to analyse the underlying issues related to interaction with sound zones. Feedback is described as the information provided by a system to the user whenever the users performs an action. Feedforward is then seen as the opposite of this, it is the information given to the user before they are performing an action, i.e the feedforward guides the user to perform the action and feedback informs the user if the action was successful or not. For both feedback and feedforward the information provided to the user is put into three categories, these being functional, augmented and inherent feedback/feedforward. How the different types of information is presented to the user can be seen in Figure 3.1. [10]

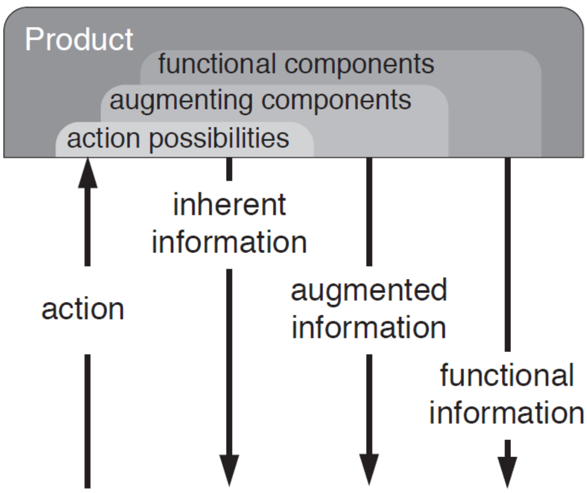


Figure 3.1: *The different types of information that a device can provide to the user when they perform an action [10].*

Functional feedforward is supposed to guide the user about the general purpose of the product, whereas functional feedback informs the user if the action they just performed was successful or not. If not direct coupling between the actions performed by an user and the actual functions of the product, then augmented feedback is needed. This is the case for most modern electronics where the user interacts with a screen. Here augmented feedback could be a button altering its visual state showing that the user successfully clicked the button. Augmented feedforward on the other hand is used to inform the user about what they should do from another source than the actual product. An example being usage of text labels or pictograms to indicate the function of a button on the screen. Inherent feedback is the information given to the user simply by performing the action, such as pressing a physical button and being able to hear/feel that the button was pressed. Inherent feedforward describes the possible actions that a user can perform on a given object based on the perceptual information available, e.g. being able to rotate open a door by using the handle. This is essentially an interpretation of the objects affordance which will be described more in depth in the following section. [10]

As it stands the sound zones do not provide any information to the user apart from the sound that would propagates from a zone. This sound does not provide the user any information through, neither through feedback nor feedforward. Therefore the issues related to interaction with sound zones can be contributed to the lack of information provided by the sound zones. By introducing either augmented feedback or feedforward as a means of providing information, the interaction issues might be solved. The next section will take a closer look at affordances and how these might be used to determine what information should be shown to the user.

3.2 Affordances

The term affordance was first introduced by psychologist James Gibson, he used this term to refer to all possible actions that a user could take with a given item based on the user's physical capability. Affordances can be described as a set of properties which show the possible actions a user can perform with a given object. An example of this is that a chair affords sitting for a person if that person has legs, or a tree affords climbing for a monkey, but not for an elephant. [11]

In the book: *The Design of Everyday Things*[11], Donald Norman applies affordances to the design of everyday objects. He defined affordances as the perceivable action possibilities of the user. Instead of considering all the actions the user is capable of i.e. the actions which the user considers possible at the given moment. Norman distinguishes between what he calls *real affordance* and what he coined *Perceived affordance*, where real affordance applies to the physical characteristics of an object as described by Gibson. Perceived affordance as defined by Norman is then about the characteristics of a devices appearance and what actions users consider possible based on said appearance. [11]

In 1991 Bill Gaver introduces three types of affordances: Perceptible, Hidden, and False. His reasoning for introducing these three was to distinguish affordances from the perceptual information available about the affordances. Gaver proposes the idea that affordances are independent of perception, the affordance of an object exists whether the user perceives it or not, based on this he defined his three types of affordances. *Perceptible Affordance* is when perceptual information is available for an existing affordance, such that the user is prompted to perform the correct action. However when no perceptual information is available for an affordance, then it is called an *Hidden Affordance* and the user must infer the affordance based on past experience or other evidence. If there is information which suggests to the user that an affordance exists, but no affordances exists then this is what is referred to as *False Affordance*. If no affordances exist and there is no perceptual information indication that a affordance exists, then this is what Gaver calls *Correct Rejection*. In Figure 3.2 the framework for affordances defined by Gaver can be seen with the x-axis defining if an affordance is present, and the y-axis defining if perceptual affordance is available. [12]

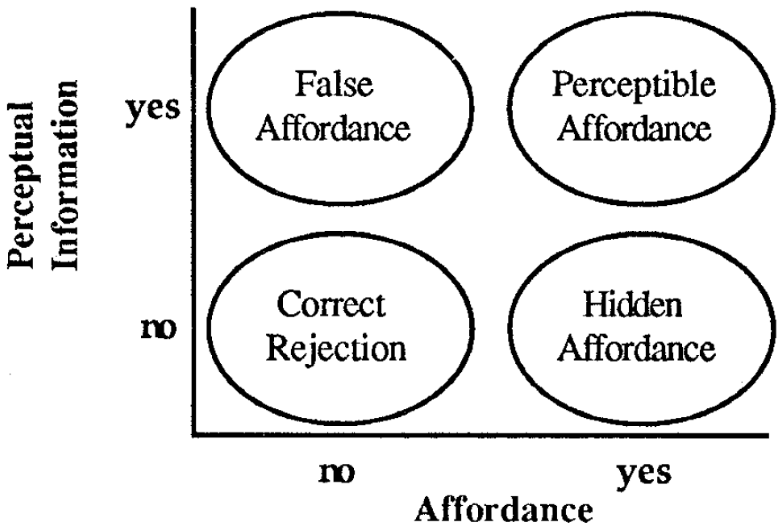


Figure 3.2: *The framework proposed by Bill Gaver for applying the notion of affordances in design [12].*

Using this framework for affordances proposed by Gaver on sound zone technology highlights the underlying issue which was also described earlier in Section 3.1. This issue is that the only information which is provided to the user is the sound coming from the zone. On its own this information does not help the user interact with the zone, as it can be hard for the user to determine the exact location of the zone based on the sound emitted by the zone. In fact if the user had to localise the zone based on the sound this could even become quite frustrating for the user. The issue which was just described is what Gaver refers to as a *Hidden Affordance*, sound zone does have an affordance but it is currently hidden as the information provided to the user does not make the affordance perceivable. As determine in Section 3.1 the sound zone must be visualised in a way such that this hidden affordance becomes perceivable for the user.

Rex Hartson[13] defined four additional affordances in 2003 *Physical Affordance*, *Cognitive Affordance*, *Sensory Affordance*, and *Functional Affordance* the definition of these types of affordances can be seen in Table 3.1. These were proposed as a response to what Hartson calls general misuse and misunderstanding of the term affordance.

Physical affordance is when a design feature allows the user to perform a physical action in the interface. This can be designing buttons of a certain size to ensure that users can click on them accurately. Essentially physical affordance is about clearly showing the user what something will do based on the design features e.g. a button with "Add to cart" on it clearly indicating that this button adds the selected goods to the cart, or the well known pause/play buttons in media players where pictograms are used instead of text to clearly indicate to the user what the button does. Cognitive affordance is design features that help the user understand something about the interface or notice key elements. An example being clearly labelled text indicating the function of a button. Sensory affordance considers design features which helps user sense something, this is particularly used to aid user in sensing cognitive and physical affordances e.g. auditory or tactile feedback to bring the users attention to a cognitive or physical affordance. Functional affordance is more related to system as a whole and the overall functionality and usefulness of it

Type	Description	Example
Cognitive	Design feature that helps users in knowing something	A button label that helps users know what will happen if they click on it
Physical	Design feature that helps users in doing a physical action in the interface	A button that is large enough so that users can click on it accurately
Sensory	Design feature that helps users sense something	A label font size large enough to read easily
Functional	Design feature that helps users accomplish work	The internal system ability to sort a series of numbers (invoked by users clicking on the Sort button)

Table 3.1: *Definition of the four different types of affordances defined by Harston [13].*

and in general should help the user achieve their underlying goal. [13]

Going forward the four types of affordances defined by Hartson will be used to identify and describe the underlying affordances of the sound zone use cases described in Chapter 2. The reasoning for choosing Hartson's work is that it incorporates the works of both Norman and Gaver into the definitions and provides a clear set of definitions for the four types defined.

3.3 Use case specific affordance

In this section the four use cases described in Section 2.2 will be examined and analysed in order to determine what the underlying affordance of the sound zone(s) is for each use case. The analysis will be done using Hartson's four affordance types and simply describing the nature of the sound zone(s) in each case. The results from said analysis can be seen in Table 3.2.

	Description of affordance	Affordance type
Social - connected	Conversation with people inside and outside of zone	Cognitive
Private - connected	Privacy User is reachable for others	Cognitive & Sensory
Social - separated	Sharing of sound source Conversation with others	Cognitive
Private - separated	Privacy User does not wish to be disturbed	Cognitive

Table 3.2: *The primary affordance and type of affordance which has been found for the four examined use cases.*

3.3.1 Social - connected

In this particular scenario there is an desire to listen to a personal sound source, and still be able to interact with others. One example is listening to a podcast while being able to talk to ones roommates, but not forcing them to also listen to the same podcast while talking to them. The sound zone in this scenario should allow a personal sound source but also facilitate interaction between the user of the sound zone and people outside it allow them to be social and connect to the people they share a space with. The affordance of a sound zone in this scenario is then being able to have a two way conversation with people inside and outside of the sound zone. This affordance can be implemented through visualisation as a cognitive affordance by visualising the zone in a way that clearly illustrates that you are open to having a conversation and not isolated despite being inside a sound zone.

3.3.2 Private - connected

The user case which is described as private - connected is mostly a scenario where two or more people occupy the same space, but one of them desires privacy e.g. wanting to focus on working. The other people in the space might however want to get the attention of the one who desires privacy, meaning the person cannot isolate themselves completely. A sound zone in this scenario must afford privacy to the user, but at the same time allow people outside the zone to reach the person inside the zone. The desire for privacy could be incorporated through visualisation as a cognitive affordance, while the option for reaching out to the user inside the zone could be implemented as a sensory affordance, e.g. a pinging sound to notify the user inside the zone that someone wants to talk to him.

3.3.3 Social - separated

This case is similar to the first one as the desire here is to be social, but what makes this case stand out is that there is only one sound source. An example of this is two people watching a movie and they have their own sound zone which both have the TV audio output as the sound source. This allows one of them to listen to the movie audio in English, while the other can listen to it at a louder audio level as well as in danish. Here the desire is mostly to share a sound source but also allow conversation between the users sharing the source. Therefore this sort of sound zone must afford sharing of a sound source as well as interaction between the people sharing the source. This could be visualised in the same manner as social - connected zone using cognitive affordance to indicate the affordances, it should however also be visualised that the zone are sharing the same source through a cognitive affordance.

3.3.4 Private - separated

The scenario described in this use case mainly focuses on the desire of complete silence, or complete control over the sounds inside a zone. As the current sound zone technology cannot prevent sounds from entering a zone, the desire of complete silence is not possible, and therefore this scenario will mostly focus on the aspect of privacy and control of the sounds inside the zone. The desire is therefore seen as complete privacy and the sound zone is seen as a "do not disturb" zone where the user can control the sound source and focus. The sound zone must therefore afford the user privacy and signal that the user inside does not want to be disturbed. This can be implemented as a cognitive affordance by visualising the zone in way where it is clear that the zone provides privacy and at the same time signals people outside the zone that the person does not want to be disturbed.

3.4 Summary

In this chapter the underlying interaction issues with sound zones were explored. It was discovered that sound zones provide the user with a limited amount of information, resulting in them being hard to interact with. Sound zones only provide information in the form of sound propagation. By analysing the affordance of the sound zones using the framework proposed by Bill Gaver[12] it was found that sound zones have what is called a hidden affordance. Meaning that an affordance exists for the sound zone, but the information provided to the user does not make the affordance perceptible. By using the affordance types proposed by Rex Hartson[13] the sound zone use cases were examined and analysed, resulting in a set of proposals to make the hidden affordance perceivable. It was found that the affordance of each zone type was primarily cognitive and sensory, based on this it was concluded that the affordance could be make perceivable by visualising the different sound zone types according to the affordance of that particular sound zone type. In the next chapter various means of interaction using AR will be explored along with an overview of the various platforms AR technology can be implemented on.

4 Interaction Methods with Augmented Reality

This chapter investigates different state of the art interaction methods using augmented reality. The goal of this chapter is to see what are the trends in AR interaction and to explore if there is any modality for implementation that could complement touch gestures.

4.1 Mobile applications using AR

This section will go over various mobile applications that utilise AR technology in some manner. The main focus will be to investigate the means of interaction between the user and the application and how the applications use AR. The goal of the section is to investigate popular AR applications to get a general idea of how interaction with an AR application can achieved.

4.1.1 Google applications and IKEA Place

There are several smart phone applications available on the market that incorporated augmented reality in some way. One of the leading developers of AR solutions is Google who implemented AR in several of its applications such as Lens, Maps and Translate as well. One of the commonly used functions of Google Lens is the visualisation of objects that are not available for the users in their environment, such as different animals. The users can search in Google's search bar for different animals and then visualise them. This happens by user tapping the screen. While the user taps the screen, they selects the surface where they wants the AR object to be spawned. After this, the application opens the camera of the device and renders the AR animal onto the selected surface. The size then can be adjusted by performing a pinching gesture with the fingers, where the size of the animal is expressed in a percentage value compared to the average size of the given animal. In Figure 4.1 below the visualisation of a hedgehog in a domestic environment can be seen, with the size of it increased to 125% compared to an average size hedgehog. [14]

Other than using AR for entertainment Google used AR for more practical uses. In Google Maps they created live view which allows the user to render navigational directions onto their screen. While using Maps to navigate the user can simply bring up the camera on the phone and look around. Arrows showing the directions and street names show up over the ground, using AR. An example of live view in Google Maps can be seen in Figure 4.3.[14]

Google Translate also uses AR by translating text to the desired language. Instead of having to type in a foreign language text, the user simply has to scan it using the phone's camera. Google Translate recognises the text and spawns the translated text over the original one using

AR. In Figure 4.2 a sign saying *exit* is translated from French to English. All of Google's AR applications use ARCOre, which is Google's own SDK for AR applications. [15; 14]

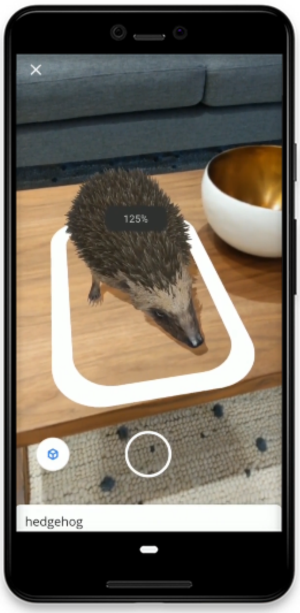


Figure 4.1: *Hedgehog visualised with Google Lens.[14]*

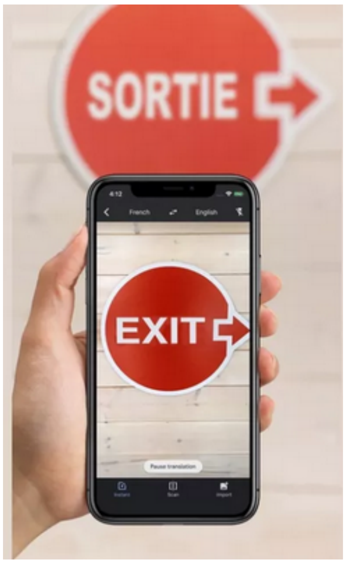


Figure 4.2: *Google Translate using AR.[14]*

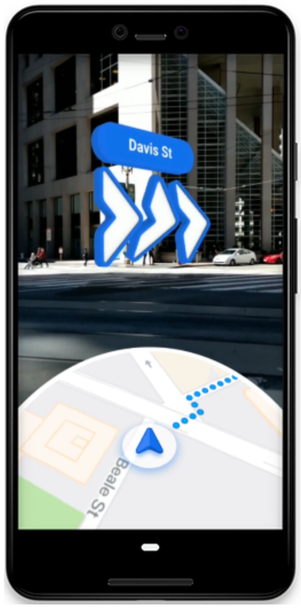


Figure 4.3: *Live view in Google Maps.[14]*

Another widely used app that implemented AR is the IKEA Place app, seen in Figure 4.4 that allows people to place furniture in their homes via AR, before buying them. The furniture that appears in the app is rendered with realistic lighting and actual size of the product in question, giving the users a realistic picture of what they can expect.[16]

Figure 4.4: *Placing a virtual armchair in the room with IKEA Place.[16]*

4.1.2 Mobile AR interaction with hand gestures

One of the commonly researched interaction methods with AR comes in the form of using hand gestures. Gestures are captured by a camera, then interpreted by a processing unit which then are translated into commands for interacting with virtual objects.

Chun et al.[17] introduces another method for AR interaction with hand gestures, designed for mobile phones. They define three main operations that can be performed on virtual objects: Translation, scaling and adjusting value. Translation is the movements of virtual objects in space which can be achieved by moving the hand across the screen. Scaling can increase or decrease the size of the object and can be done by performing a pinching movement the same way as one would zoom in or out on a touch screen. Adjusting Value entails the transparency of the object in this case and can be adjusted by moving the hand away or closer to the screen. To maintain a certain transparency the user has to quickly remove their hand from the image. The system at work can be seen in Figure 4.5 below.

Figure 4.5: *Interaction with AR objects using a smartphone. [17]*

The system works by detecting the hand using background subtraction which means that in this case a depth measurement is not required opposed to the previously presented work. The authors also performed user studies in which they showed that majority of test personnel had no trouble interacting with the application, but some of them had difficulties of controlling object in AR. [17]

4.2 Desktop applications using AR

This section presents applications that use non-mobile, desktop interfaces for interaction with augmented reality.

4.2.1 User interaction in augmented reality (Patent)

In 2012 a patent publication was submitted by *Hilliges et al.*[18] and Microsoft Corporation, describing several scenarios for interaction with augmented reality. This section presents two possibilities for interaction in augmented reality.

One of the scenarios presents a direct user interaction method where there is one virtual object in the scene, in this case the cylinder. The objects can be controlled by using hand gestures which is being captured by a camera. All the computational processes are performed by an external computing device. This setup can be seen in Figure 4.6. In this scenario the camera is equipped with a depth sensor. It captures the position and orientation of the user's hand and allow interaction with the virtual object using predefined gestures. The virtual environment and the interactions can be observed by the user on the display device, which is in this case a screen. The different stages of interaction in the system as it is defined in the patent[18]:

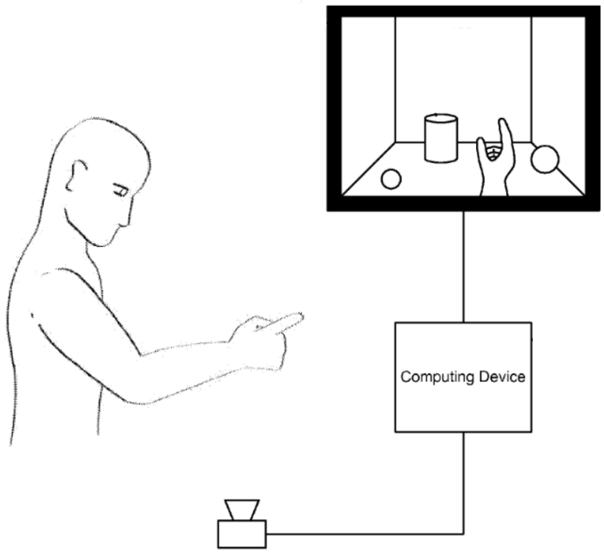


Figure 4.6: *First scenario. [18]*

The different stages of interaction in the system as it is defined in the patent [18] for the previously mentioned scenario in Figure 4.6 are:

1. Generate and display 3D augmented reality environment.
2. Receive images of user's hand from depth camera.
3. Track movement and pose of hand in 6 degree of freedom (DOF).
4. Monitor pose of hand and detect gestures.
5. Trigger associated interaction between hand and virtual object responsive to detecting a given gesture.[18]

In the next example, seen in Figure 4.7 controls for interaction are rendered onto the user's non-dominant hand. Different controls can be associated with the fingers on the non-dominant hand which in this case are (from right to left):[18]

- Copy
- Paste
- Send
- Save
- New

These controls can be activated by touching the fingers on the non-dominant hand by a finger on the other hand thus also providing haptic feedback to the user. In this scenario the users holds the virtual object in their non-dominant hand palm and the aforementioned controls can be performed on this virtual object. An alternative method for activating the controls instead of touching the fingers with the opposite hand is to bend the fingers to which the controls are assigned. [18]

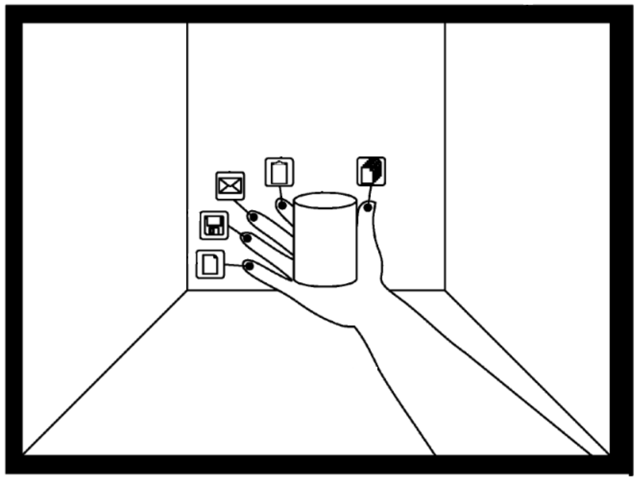


Figure 4.7: *Second scenario. [18]*

4.2.2 Ubii

One of the works in the field of hand gesture interaction with AR is a system called Ubii[19] (Ubiquitous interface and interaction) developed by *Lin et al.*[19] In their work they present an integrated interface system for hand gesture interaction where the user wears smart glasses which are responsible for displaying the interface using augmented reality and capturing the hand gestures performed by the user.

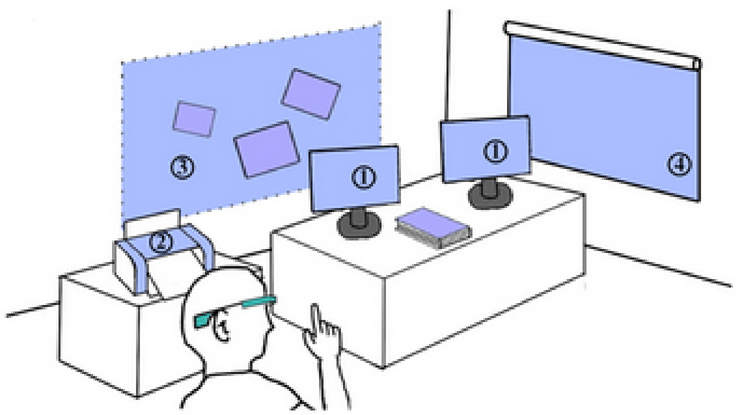


Figure 4.8: *Office setup for Ubii. [19]*

The Ubii system consists of a set of smart devices connected together with which the user can interact through smart glasses and gestures. In the Figure 4.8 an office setup can be seen with classic office equipment in the scene and the user wearing the smart glasses. Ubii allows users to perform operations that are physically out of their reach such as printing, projecting and file transfer. This solution also applies computation offloading to perform computationally heavy operations on a separate processing unit instead of on the smart glass itself. The authors show that by using Ubii these common office operations can be shortened down in time for more efficient work pace. [19]

An example of how the Ubii system uses gestures for office work, can be seen in Figure 4.9. In this case the user would like to copy a file from one of the computers and transfer to the other computer at the other side of the table. The user has to simply move their hand over the file and perform a pinching gesture with the thumb and pointing finger to pick up the file. The gesture for selecting the file is recorded by the smart glasses. After this the user simply moves the hand over the other computer and releases the file which results in the file being transferred to the other computer.

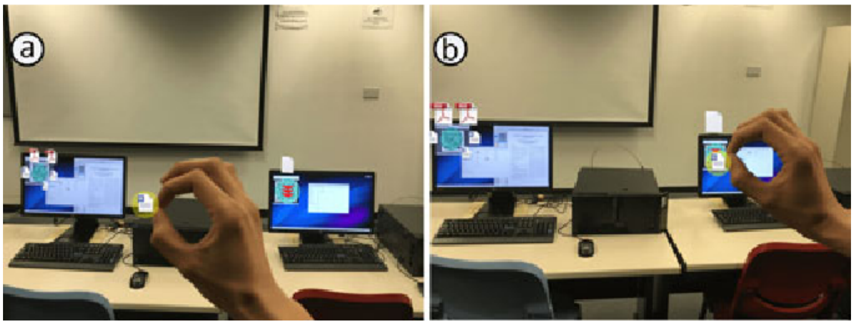


Figure 4.9: *Office setup for Ubii. [19]*

4.2.3 Hands in Space

Another work in the field of AR interaction with hand gestures is the Hands in Space system, developed by *Billinghurst et al.*[20]. In their solution the user sits at a table where there is a Kinect camera mounted above the table, looking straight down. The depth measurements of the Kinect camera are used so when the user's hand is placed above the tabletop it allows the system to render objects into the hand of the person. One use for Hand in Space system is in the PhobiAR app that helps treating people with fear of spiders. The system can be seen in Figure 4.10 below.

Figure 4.10: *PhobiAR helps patients with fear for spiders by letting them interact with virtual spiders. [20]*

With the PhobiAR app the users can overcome their fear of spiders by interacting with virtual ones. They can pick them up and let them crawl on their hand by performing certain gestures.

Another example where the system is used is the movement of virtual objects by hand. Objects can be picked up by a simple pinching gesture of the fingers. The gesture is recognised by using skin detection, finding and tracking of the contours of the fingertips of the thumb and index finger. In this case the system is made up of two modalities. The Kinect camera is responsible for the depth tracking, while the fingertip operations are performed by a smartphone. Implementing both on the smartphone is not possible due to the hardware limitations of the smartphone, not being equipped with a sensor that is able to measure depth. The system setup can be seen in Figure 4.11 below.

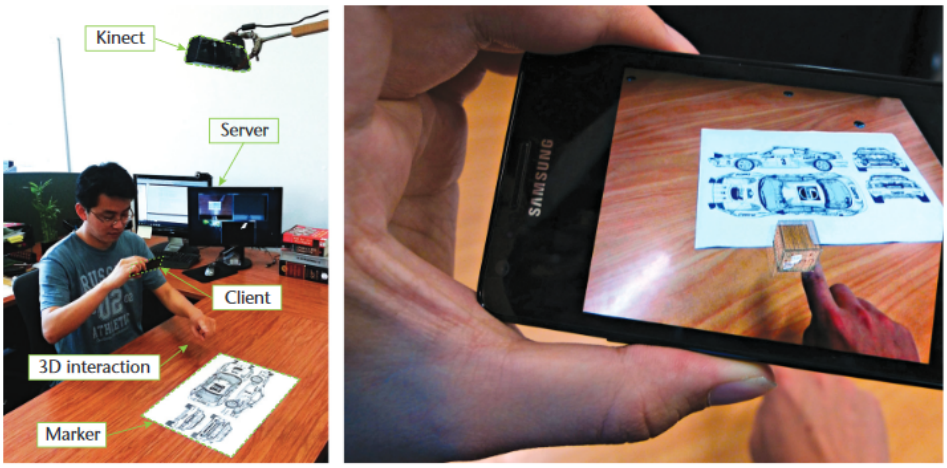


Figure 4.11: *System setup for the Hands in Space system. [20]*

4.3 Overview

	Mobile	Desktop	Wearable	Interaction Method
Google AR Applications and IKEA Place [14] [16]	Yes	No	No	- Touchscreen gestures - Moving camera around
Microsoft Patent Publication [18]	No	Yes	No	- Hand gestures, recorded by camera
Ubii [19]	No	Yes	Yes	- Hand gestures, recorded by wearable (smart glasses)
Hands in Space [20]	Yes	Yes	No	- Hand gestures, recorded by Kinect camera
Real-time Hand Interaction [17]	Yes	No	No	- Hand gestures, recorded by camera - Visualisation through smartphone

Table 4.1: *Overview of the different AR interaction systems, presented previously in this chapter.*

The Table 4.1 aims to compare the different AR interaction solutions that were presented in this chapter. Since the goal is to develop an smart phone application for sound zone interaction the main focus is on the Google[14], IKEA[16] and the Real-time hand interaction [17] applications due to the fact that these solutions use a smartphone. The problem with the Real-time hand interaction app is that it uses a lot of additional hardware such as an external depth camera and a desktop computer as well which could be problematic for users to set up on their own. Because of this the focus will be on the smart phone applications that doesn't use any additional hardware for augmented reality interaction, these being the Google and IKEA applications. Without exception these apps use touch gestures through the device's touchscreen, thus it is important to get an understanding of conventional gestures and unwritten rules regarding touch gestures in the field of application design.

4.4 Touch gestures

Rather than focusing on developing a particular set of gestures which can be used to interact with the application, the well known conventions for 2D gesture interaction on a smartphone as defined by Google's Material Design[21] will be used instead. It defines a set of graphical design guidelines and conventions regarding application interaction design which enables a smooth interaction between the users and the device. Using well known conventions helps convey information to the users as most are familiar with the commands/gestures. As stated in Section 3.1 there is a lack of information shown to the user regarding the sound zones, requiring introduction of either augmented feedforward/feedback. By using these well known conventions it makes the usage of additional augmented feedforward less necessary. As sound zones have a hidden affordance, the only way for user to infer the affordance is through past experience. This is possible when interaction is done according to these well known conventions. This allows users to interact with our application based on their previous experience with other applications using the same conventions. Most people are already familiar with or even expect to be able to control an application using the 2D touch gestures described by Material Design. Therefore using these conventions ensures that the users pre-established notion of interaction with other applications can be translated to interaction with our application.[21]

The touch gestures can be divided into three main groups:

- Navigational
- Action
- Transform

For all three categories it is important that the specific gestures provide a realistic response meaning there exist a set of established conventions which are important to keep in mind when implementing these gestures. These include for example that a swiping gesture used for navigating between tabs shouldn't trigger any animations, but the movements of the tabs should match the swiping velocity of the user, which makes users feel like they are more in control of the gesture controls.

4.4.1 Navigational gestures

Navigational gestures complement other input methods such as on-screen buttons to help the user navigate around in the application. It should be said that this is mostly the same gestures, but the function of the gesture depends on the context. An example of this is the swipe and scroll/pan gestures as these all require the user to slide their finger across the screen, but based on the context the effect of these change. The navigational gestures include the following [21]:

- **Tap:** Similar to buttons, the user can reach a destination within the app by tapping on an element
- **Scroll and pan:** By sliding the surfaces in different directions the user can reach different parts of the available content. Examples for this can be moving around on a map, or scrolling through a web page.
- **Drag:** Dragging in navigational gestures can be used to bring non-visible content into view and hide it when they are visible.
- **Swipe:** Swiping is used for navigating horizontally between different windows or tabs. An example for this could be changing between different images in the photo gallery of the device.
- **Pinch:** By using two fingers and pinching the users can open up different windows and then close them by scaling these windows.

4.4.2 Action gestures

Action gestures work similarly to pressing an on-screen button, meaning they trigger some kind of an action which allows users to interact with certain elements on the screen. Action gestures include[21]:

- **Tap and long press:** Tapping and long pressing in the context of action gestures allow the user to trigger an action and access functionalities that are not visible by default. For example by long tapping the user can select a message after which the option to delete that message becomes visible.
- **Swipe:** By swiping over an element additional functions can be made visible similar to tap and long press. Following the previous example, a message can be marked as important by swiping over it horizontally.

4.4.3 Transform gestures

Transform gestures allow users to modify an on-screen object's certain attributes such as size, position and alignment. Examples for transform gestures are[21]:

- **Double tap and pinching:** Double tapping on pinching over content such as images allows users to zoom in and out of it or scale the given object.
- **Pick up and move:** By combining a long tap and dragging objects on screen can be selected and relocated.

- **Compound gestures:** Compound gestures entail gestures that combine several gestures allowing the user transition between different gestures seamlessly. Example of this can be a map application where the user is able to use a pinching gesture to zoom in and out of the map, but at the same is able to rotate the map by moving the two fingers accordingly.

4.5 Summary and conclusion

This chapter investigated several state of the art applications that use different methods for interaction with augmented reality. The goal is to develop a smart phone application which enables users to interact with sound zones through augmented reality. The problem with majority of the solutions presented earlier is that they require a lot of additional hardware to smart phones which can be problematic for users to easily set up in their domestic environments. It has been decided that the focus will be on mobile applications that only require a smart phone and nothing else, and it was found that all the presented applications which fall into this category use touch gestures for interaction. After this several touch gestures have been investigated in app design which will be used and serve as a baseline during the design phase of this project.

5 | Problem Analysis Conclusion

The research conducted in the Problem Analysis has provided an insight into the sound zone interaction issues. The analysis first looked into the fundamentals of the sound zone technology and the various use cases related to the sound zones in the domestic environment. The challenges revolved around sound zone setup have been investigated, where the factors which influence the control of the sound zones have been determined. The framework for categorising the sound zones into the four categories has been adopted from Lundgaard et al. [2], as this framework will be used to create the scenarios for the sound zones in this project. Additionally, the fifth scenario with the transitional sound zone has been introduced While analysing the nature of the sound zones, it has been found out that there exist an essential need to convey the user needs while interacting with sound zones.

For this reason the underlying issue which hinders interaction with sound zones was analysed, and it was found that the issue is caused by there being a lack of information provided to the user. Using the *Interaction Frogger Framework*[10] it was determined that currently there is no feedback and only limited feedforward provided to the user in the form of sound propagation from the zone. Therefore there is a room for introduction of the augmented feedforward through AR. This would be done by visualising the sound zone, easily allowing the user to locate the zone and see its position. In order to determine how to best visualise the sound zones the term affordance was researched through the framework established by Bill Gaver [12]. It was found that sound zones have what is called a hidden affordance, and that this affordance must be made perceivable to the user. This further solidifies the need to visualise sound zones. An analysis was done using affordance types proposed by Rex Hartson[13] to determine the affordance and type of affordance of sound zones in different use cases. The affordance and type will be used to visualise the sound zones in a way which conveys the functionality of the sound zone in a given use case. It was determined that the sound zone would be visualised differently depending on the use case as the intended usage of the sound zone differs greatly between the use case.

The chapter about interaction with augmented reality looked into different state of the art solutions in the field of AR interaction. The research focused on two main types of products, one being mobile and the other is desktop applications. Since the goal of this project is developing a mobile phone application for sound zone interaction, the main focus was on interaction within mobile apps. It was found that most of these applications either use touch or hand gestures, however implementing hand gestures could require additional hardware such as depth cameras or more computational power, requiring a computer which complicates the setup. It was concluded that the system should comprise of only one device, in this case the smart phone so anyone can easily learn how to use the system due to the fact that it is following the known conventions outlined by Material Design[21]. After this different touch gestures in mobile applications where examined to learn about the conventional touch gestures used in applications nowadays. These will then serve as a guideline in the design phase of this project.

5.1 Final problem statement

Based on the findings of the problem analysis it has been established that the scope of this project is to create the visualisations of sound zones according to the domestic environment use cases. The interaction with sound zones will be realised through an AR application implemented on a smart phone with the focus being on using the touch gesture conventions described in Material Design for interacting with the sound zones. Going forward the different sound zone visualisations must be determined and the specific interaction options must also be established. The development platform for the application as well as the framework for AR implementation must also be determined before implementation can begin. This has lead to the following final problem statement.

How can a mobile AR application be developed to implement the use case specific visualisations of sound zones?

Part II

Design

6 Proposed Solution

This chapter will provide an overview of the ideal solution. This solution assumes that the proposed application is a part of a complex system which operates the sound zones in the domestic environment. The major assumption is that the application is capable of communicating with the control unit for the loudspeaker arrays and it serves as an installation tool for setting up the sound zone environment. Ideally, the solution we are working on is envisioned to be a control UI for the sound zone management.

6.1 Ideal solution system

Based on the sound zone use cases described in Section 2.2 several types of sound zones will be created, allowing the user to select the type of zone which best suits them for the given activity. The goal is that the user should be able to perceive the affordance of the sound zone through a visualisation of the zone in augmented reality. The user will primarily use the application to visualise sound zones during configuration and setup. By visualising the sound zones differently and making the visualisation match the use case, the user should be able to understand which zone type best applies to a given activity. The users will be able to setup and configure sound zones using the application and if needed the zones can be configured by the user later. During creation of a sound zone the user will be asked to select which type of sound zone they want to create. They will also be presented with the option of giving the zone a name.

The application should communicate with a sound zone generation system so that the sound zones created in the application is provided with the required configuration of the sound zone control points resulting in the actual creation of sound zones. This is needed for the virtual placement in the application and actual positions of the sound zones in the real environment to be aligned.

The ideal solution should allow the user to know their position relative to the environment. This is needed for easier and more accurate placement of the sound zones. Also, tracking is vital for the dynamic/transitional sound zone scenario, where the sound zone location needs to be linked to the user location for it to follow them. Tracking of the user position in relation to some known origin point is needed if dynamic sound zones are to be implemented. This will mainly be used if a sound zone is to follow the user around, and to gather information about how many users there are inside a given zone.

In the ideal solution scenario, the sound zones are assumed to be nearly perfect isolated regions of space, where the media playing inside the sound zone can only be heard if inside the sound zone. In reality, the sound zone is far from a perfectly isolated soundscape and instead is better described as an area where the maximum contrast to the surrounding sound has been achieved [4]. The outer noises and sound leakage from the other sound zones can be heard by a person inside the sound zone. It should also be noted that in reality people outside a sound zone will

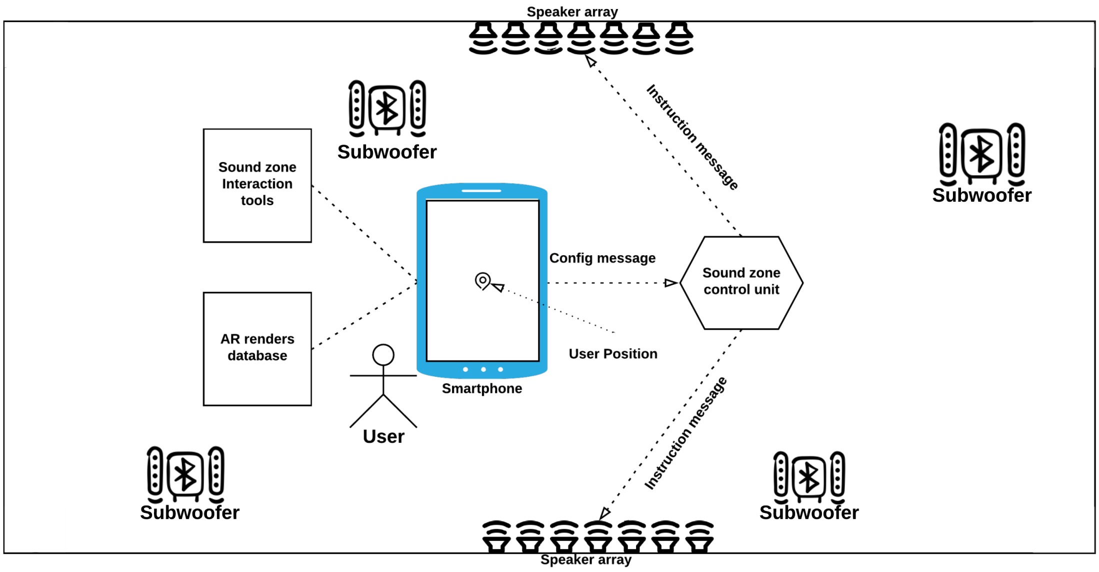


Figure 6.1: *Sketch of the ideal solution showing the contents of the application as well as the room setup required for the sound zone system.*

be able to hear the audio emitting from the zone, it will just be significantly lower compared to being inside the sound zone. This is due to the limitations of the current state of the acoustic technology available during the time of this project.

In the Figure 6.1 a sketch of such a scenario is illustrated. The sound zones are created either via speaker arrays and supplementary subwoofers placed at various locations in the room. The smartphone application acts as a master instruction the sound zone control unit to correct the speaker array according to the user preferences. The application allows for the sound zone interaction and chooses a suitable AR render from a database for a sound zone based on user preferences.

In the ideal solution scenario the sound zones would mimic the behaviour of a Wi-Fi connected speaker, so that the selected audio content can be streamed to it. This would be similar in function how users can selected a speaker device when playing music in their preferred music player, here each sound zone would be considered its own speaker device which the user could select.

6.1.1 User interaction with sound zones

As described above the users will be able to interact with the sound zones through the application, both during the setup stage and if they wish to configure them again later. The interactions possible for the user has been determined to be the following list:

- Create a sound zone.
- Delete a sound zone.
- Change the size of a sound zone.
- Move the sound zone around to change its position.
- Change properties of the sound zone such as the zone type and name.
- Configure a specific sound zone to follow the user.
- Swap the position and properties of two sound zones.
- Duplicate a given sound zone and its properties.

Creation and deletion of the sound zones are determined to be the fundamental operations allowing for basic interaction with sound zones. The remaining operations is seen as compound operations consisting of some combination of creation and deletion. In essence the remaining operations could be achieved by performing a number of creation/deletion operations. Introducing the scaling and movement operations allow the users to modify a sound zone without having to delete it and create a new zone. The ability to change the properties of the sound zone allows for user customisation in the form of changing between sound zone types and zone name which removes the need to re-create the sound zone from scratch, but instead adjust it to the user preferences. In this case these attributes utilise user ability to distinguish between the sound zones. Assuming that the final product has several versions of the sound zone visualisers the user can choose from, there would be a need to differentiate between the sound zones of the same type, hence possibly colour of the visualisers can be used.

Having a zone following the user is required for a dynamic sound zone scenario. Swapping the positions of the created sound zones allows for an easier customisation of the setup, so that the user would not need to go through all the creation steps from the beginning.

6.1.2 Automatic sound zone behaviour

After the user has created a sound zone, it needs to follow predefined behaviour so that the system can be stable. This concerns the scenarios where sound zones overlap or there is a possible zone collision like in the following zone case. The private - connected scenario from Figure 2.4 is an example of overlapping zones. Even with the assumption of the perfect sound zones, the private - connected scenario creates a region where two soundscapes are mixed. The overlap region would result in a chaotic noisy experience which would ruin the sound zone experience for the user. A simple solution to this issue would be a buffer radius to the zones maximum size where another sound zone cannot be placed. There is a need to notify the user about zone collisions and it can be implemented via instruction message telling about zone overlap and providing them instructions on how to resolve the issue.

The case of the dynamic sound zone introduces another possibility of zones colliding with each other. The sound zone buffer solution might not be applicable here, since that would require person to walk a specific trajectory to avoid existing sound zones in the environment. Without actively scanning of the environment for the sound zones, this would be a very challenging task. A possible solution to this problem is prioritisation of a particular sound zone over another. For example, the dynamic sound zone is active at all time and in case of collisions the stationary zones are muted to allow the following zone to pass through; or an opposite behaviour - the dynamic sound zone is muted when it enters an already existing sound zones.

Apart from the zone collision problem, the behaviour of the dynamic sound zone needs to be programmed to address the environmental boundaries. For example, if the user goes outside the limits where the sound zone can be created, the dynamic sound zone needs to remember the exit point and stay there until the user returns within the working boundary or even return to a given position if the user leaves the area.

The solution which has just been described is seen as the ideal solution, yet some of the functionalities are more important than others when it comes to creating a minimal viable product (MVP). The MVP needs to include the fundamental properties of user interaction and automatic sound zone behaviour while allowing for addition of the supplementary functionalities. Therefore the next section will go into which parts of the ideal solutions will be prioritised when it comes to the implementation.

6.2 Delimitation

This project will focus on creating a smartphone application for the sound zone visualisation and interaction. Integrating the application with the sound zone generation system is deemed as out of scope for this project. It is seen as possible to perform this integration but the focus of this project is on the interaction and visualisation aspects of the sound zones and application features focused on these aspects will therefore take priority during implementation.

The core principle of the solution is to focus on providing information that improves user interaction with sound zones as discussed in Chapter 3. Because of this, the work will be focused on creating an application that ensures that the hidden affordance of sound zones becomes perceivable to the user through AR visualisation.

Going forward several assumptions will be made about the sound zone behaviour such as sound propagation, sound leakage from the zone and the general shape of the zone. The actual shape of a sound zone is difficult to determine and visualise accurately therefore the shape visualised might not be accurate, but rather focus on making the hidden affordance of the sound zones perceivable. Visualising all the physical properties of a sound zone would result in sound being visualised in the entire room, essentially filling the screen with clutter distracting from the main sound zone. Therefore only the sound zone itself will be visualised and not the sound travelling from the speaker array or the sound propagating through the room. It is deemed that the physical representation of the sound zone will be simplified to contain only the essential features of the sound zones as based on the assumptions described above. These features are the sound zone location, size, and type.

Several limitations must be made regarding the features of the application to keep the development focus on interaction and visualisation. The features for the interaction as described in Section 6.1 can be split into several categories describing the priority of the features in regards to the project as can be seen in Table 6.1 .

The bottom layer consists of the features considered as needed for the application to work, which are creating and deleting sound zones. All other features can be described as compound features

Sound Zone Features	
Nice to have	Follow Swap Duplicate
Quality of life improvements	Scale Move Change properties
Fundamentals	Create Delete

Table 6.1: *The features of a sound zone put into different categories depending on the implementation priority.*

with functionality of the fundamental features, such as scaling and moving which would equal deleting a zone and creating a new with in the new position with the desired scale. The top level features such as zone following the user and swapping the zones will be implemented if there will be remaining time as those are not crucial to the functionality of the application.

The tracking features described in Section 6.1 will not be included. If time allows for implementation of dynamic sound zones then the user position will be obtained by using pre-defined locations instead.

The Wi-Fi based speaker device system will also not be implemented as this does not specifically add anything to the sound zone interaction and visualisation part of the application, but is rather seen as a feature that is needed if the application was to be launched on the market. The feature is seen as important in the grand scheme of things, but does not add anything to the focus areas of this project.

6.3 Requirements

The following set of the requirements serve as project goals with regard to implementation of the application. Rather than a strict set of requirements to be implemented, these should serve as a goals for the design and the implementation of the application. These goals are listed in the rank of priority and will serve as a guide while implementing the solution:

6.3.1 Sound zone interaction features

These requirements detail the specific interaction features which should be implemented for the sound zones. These requirements are ranked in a decreasing priority according to the features described in Table 6.1.

1. A user must be able to create a sound zone at a given location.
2. It must be possible for a user to delete a given sound zone.
3. The user must be able to change the zone position of a sound zone by moving it around.
4. It must be possible to change the size of a given sound zone.
5. An user should be able to change the type of any existing sound zones along with any zone specific properties defined for the given zone.

6. It should be possible to swap the location and configuration of two zones.
7. The user should be able to duplicate an existing sound zone.
8. It should be possible for a sound zone to follow the user around if this is desired.

Apart from these requirements the solution needs to fulfil basic functionalities of a smartphone application, such as functional UI and robustness in operation.

6.3.2 Application requirements

The requirements below describe the general features which must be considered when designing and implementing the application.

- The application must utilise the design conventions described by Goggle's Material Design [21].
- The application must be able to save the current pose and visualisation of any placed sound zones.
- The application must be able to load the pose and visualisation of any previously saved sound zones.
- The visualisation of sound zones should be implemented in augmented reality.
- It should be possible to create more than one sound zone.
- Each sound zone type should be visualised according to the particular use cases matching the type.
- The user should be notified if they try to place a sound zone in an area which already contains another sound zone.
- If a user leaves the room while a sound zone is following them, they should be notified that the sound zone has stopped following them.
- The user needs to be notified when the scaling of the sound zone is overlapping with another zone.
- When a sound zone is following the user it should be able to handle situations where it would overlap with existing zone.

Having established the requirements needed to design the application it is possible to begin the design process, in the next chapter the final design of the application will be described along with the methods and techniques used in the design process.

7 Design Concept

This chapter aims to develop a design concept for the proposed solution. To accomplish this goal, the methods from the GV Design Sprint [22] have been used. These methods were applied to create a design concept for the visualisation of sound zones and the graphical user interface. The chapter is split into two parts, the sound zone design and the GUI design.

7.1 Design sprint

For both designs a sprint was conducted, both of which were based on the principles of the GV Design Sprint[22]. The idea behind a design sprint is to intensively output a lot of ideas over a short period of time, from which then a best candidate for the solution can be derived. For both design sprints the following steps were followed:

1. 20 minutes discussion about the problem at hand: This step ensures that everyone is on the same page regarding what the design sprint will be about and have the same understanding about the question at hand.
2. Crazy8: In Crazy8 each member of the group generates eight ideas in the form of sketches during a 5 minute sketching session.
3. Discussing the sketches: Everyone takes their turn and explains the sketches to the other members of the group.
4. Voting: After this the group members are voting on the 3 sketches and ideas they liked the most.
5. Discussion: The voting is followed by a discussion where the pros and cons of each concept was discussed.
6. Combination: During the discussion the concepts with the most votes were combined to create a new concept with key features of each concept.

7.1.1 Sound zone design sprint

The sound zone design sprint was seeking answers to two main questions which were identified during the first discussion:

- How should the sound zone look like when visualised with AR?
- How should the sound zone convey information to the users?

With these in mind every member of the group created eight sketches in a five minute sketching session after which these sketches have been discussed. There were differences in the approaches to the sketching as some group members focused on the specific design details of the sound zones while other's focus was on the information conveyed to the user. Because of this the different concepts were combined to create a final concept that incorporate the desirable details of each concept into one. The different approaches from the sketches can be seen in the Figures 7.1, 7.2 and 7.3.

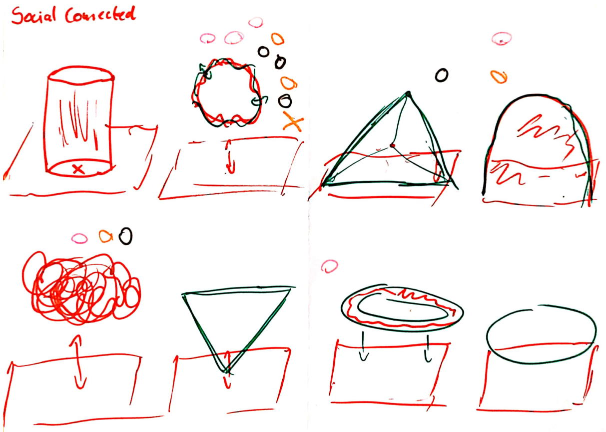


Figure 7.1: *Sketch 1 of the first design sprint, showing possible shapes the sound zones could take on.*

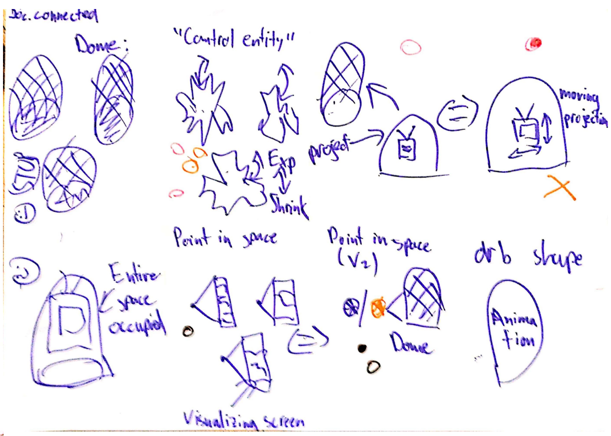


Figure 7.2: *Sketch 2 of the first design sprint, showing possible shapes the sound zones could take on.*

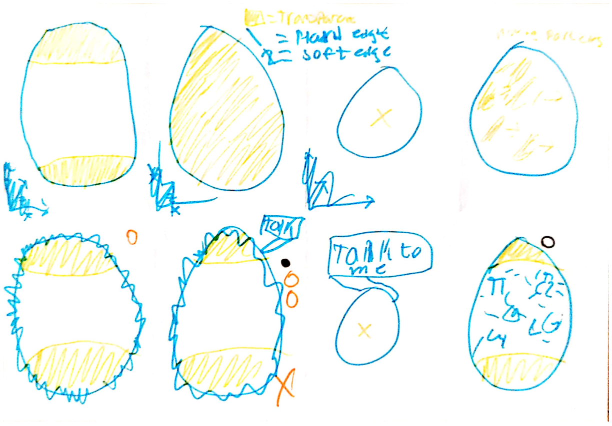


Figure 7.3: *Sketch 3 of the first design sprint with the focus being on information about the zone conveyed to the user.*

In Figures 7.1 and 7.2 the focus was on the different shapes the sound zones could possibly take on and not the information it conveys to the user. The shapes that appeared in both sketches are an orb and a dome shape.

The orb is thought to best illustrate the abstract shape of the sound zone, as the circumference of the sound zone can be matched to the circumference of the orb. The idea was that the orb shape can be distorted to show the lack of clear definition of where the zone starts and ends. The dome shape has been chosen for the similar reason to the orb shape, as it is would be easy to observe the area occupied by the sound zone when looking at the dome.

Thus these were decided to be included in one of the visualisation designs that will be implemented and tested by users. The other three designs, the egg shape and a shape-shifting orb were decided by voting to be included in the designs for implementation.

The sketches from Figure 7.3 use an egg like shape for the sound zones in all solutions. However the emphasis in these solutions was on the information the zone conveys to other users, outside of the sound zone. The idea is that when a sound zone is created the user has the option to select the type of the zone in regards to the desired level of privacy. The visualisation of these zones would work by filling out the sound zone with a colour, meaning the more full a zone is, the more private it is. In this case a fully coloured zone would mean that the person does not want to be disturbed, while an empty one, where only the edges of the sound zone are visible is willing to interact with others. This can be mapped to the use cases described in Section 2.2 where a private zone would resemble the private - separated use case. The public zone would then resemble the social - connected use case and a mixed zone which can be described as in-between the public and private zone.

The abstract entity candidate has been selected as it might best convey the content of the sound zone. For example in the scenario when music is being played in the sound zone, the abstract entity shape would resonate accordingly to the rhythm of the music, making it it clear to the user that the sound zone affords listening to music based on the movement of the zone.

During the discussion of the sketches it was decided that it would be beneficial to combine these approaches in a way that the different sound zone visualisation design proposals would give an answer to the two questions that were formulated at the beginning of the design sprint. Figure 7.1 illustrates different visualisation options of a sound zone. Four of these shapes have been selected based on voting by each of the group members and can be seen in Figure 7.4. All of these sound zone shapes will be implemented in the application where the users can select which shape they prefer.

The decision on the shape of the sound zones was combined with the sketches in Figure 7.3, where the focus was on the information the zones convey to the users around it. The goal of this sprint was to generate different shapes for visualising sound zones. However the sprint also resulted in a more conceptual idea on how the sound zone would convey information to users. Based on this two more concepts were generated resulting in three unique concepts on how sound zones can convey information.

7.2 Final visualisation designs

Based on the design sprint that was detailed earlier in this chapter three visualisation concepts and four different shapes have been created. These will be evaluated by the test participants at the prototype user test stage of the project.

7.2.1 Shape designs

Four shapes were chosen as the final shapes for the visualisation of sound zones and can be seen in Figure 7.4. These shapes provide customisation options for the user so that they can choose the shape they prefer.

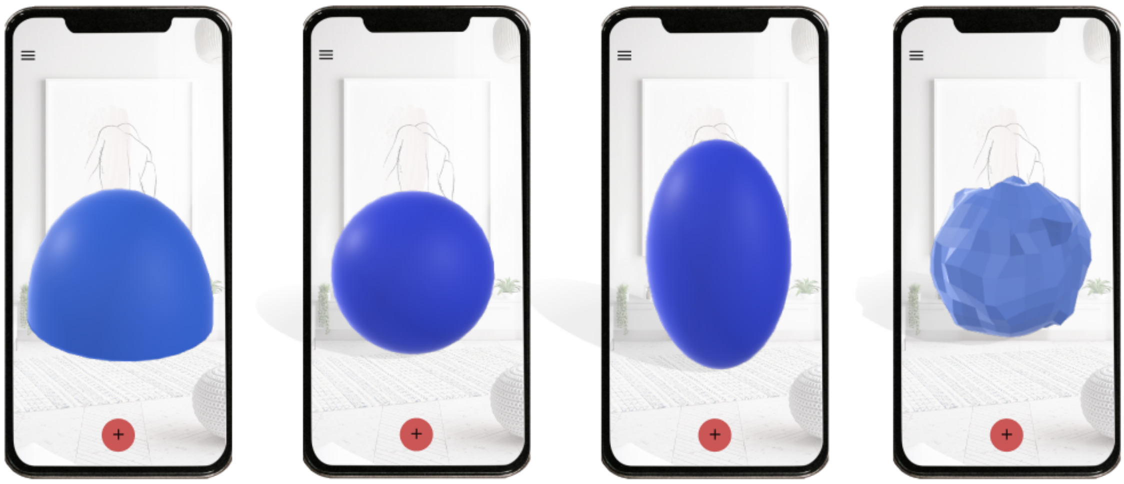


Figure 7.4: *The different shapes that will be implemented for sound zone visualisation (from left to right): dome, orb, egg shape, abstract entity*

7.2.2 Visualisation concepts

The goal of the concept design sprint was to decide on how to visualise sound zones in AR, the output of this design sprint was four shapes and three visualisation concepts as seen in Figure 7.5. The general idea is that each concept contains three types: Private, mixed and public. Each type is then to be used depending on the social context that best fits the type.

Partially covered/uncovered
This concept is aiming to tell other users the type of the sound zone by partially or fully covering the sound zone. The idea with this concept that a fully covered sound zone would be private, a partially covered zone would be mixed and a not covered (wireframe) sound zone would be of public type.

Transparency of sound zone
The second concept conveys users about the type of the sound zone by adjusting the transparency levels. A private zone would not be transparent at all, resembling a solid object that should indicate a desire for privacy, while a public zone would be the most transparent, be made of a glass/gas material conveying people that it is possible to walk through and into the sound zone. A mixed type sound zone would have a transparency level that is in-between the other two types.

Icon on sound zone
The icon concept conveys the type of the zone to other users in a more explicit way: An icon is rendered onto the surface of the sound zone. The purpose of the icon is to convey the type of the sound zone in a form of a well-known visualisation which can be related to the public or private state. These icons are known to mostly everyone and were picked from the Material Design icon library [21].

Four shape design and three visualisation concepts have been developed during the design sprint. These will be part of the application at the initial implementation stage and will be evaluated by test participants during user test via a survey. Based on the results of the user test the most popular shape and concept will be used in the final implementation of the application.

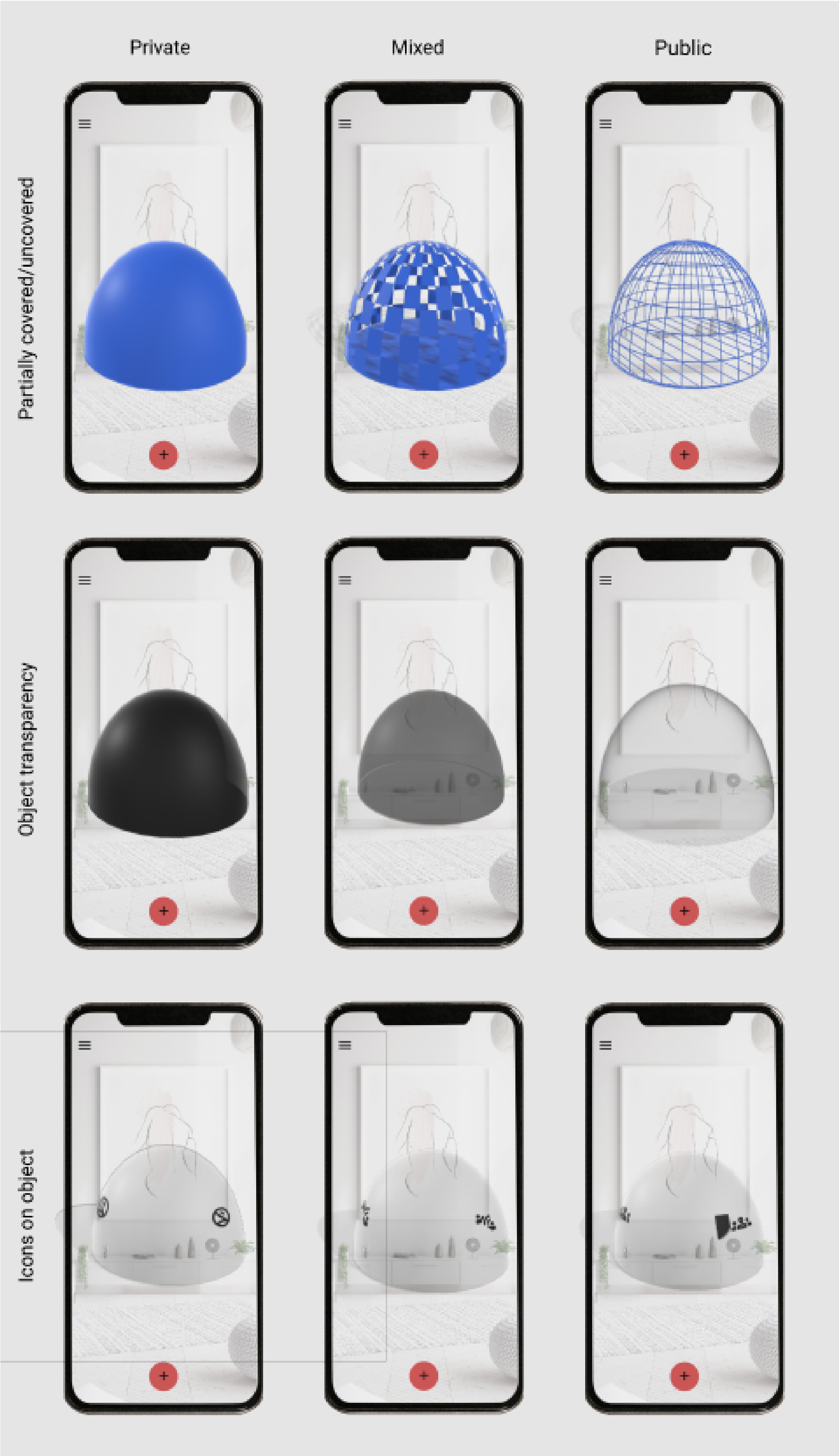


Figure 7.5: *The different concepts that have been developed during the design sprint and will be evaluated by test participants at the user test stage.*

7.2.3 UI design sprint

The second design sprint was responsible for the specific design of the application which entails the different functionalities and the UI design. The first discussion reached in agreement that the design should be simple and only contain the necessary functionalities, which were also defined as:

- Create and remove sound zones.
- Move sound zones.
- Save and edit existing setups.
- Select the type of sound zones at the creation step.

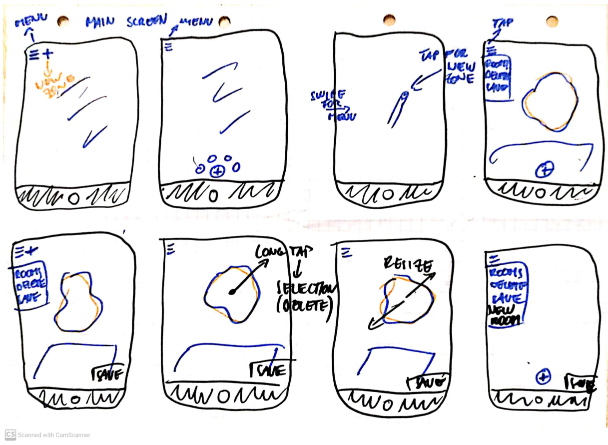


Figure 7.6: *Sketch 1 of the second design sprint showing the GUI layout and different functionalities.*

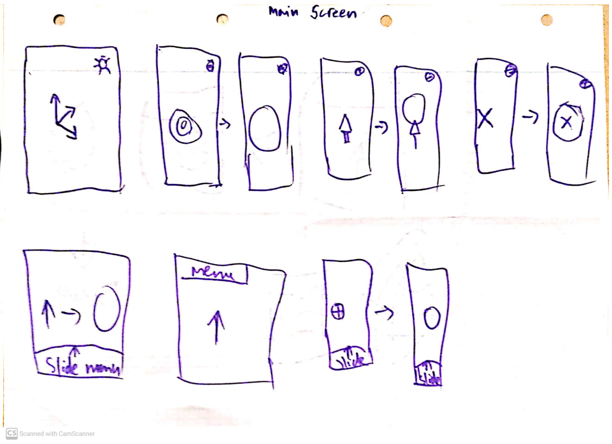


Figure 7.7: *Sketch 2 of the second design sprint showing different sound zone interaction functionalities of the application.*

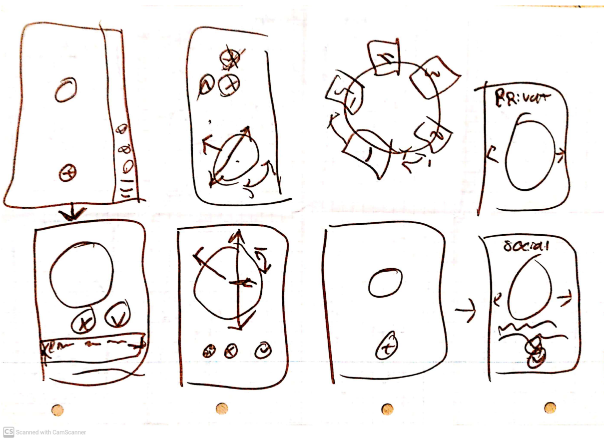


Figure 7.8: *Sketch 3 of the second design sprint showing the GUI layout and different functionalities.*

The sketches were created with the goal of answering the previously formulated questions on UI design. The first question that required an answer was how to create a sound zone. Both in Figure 7.6 and 7.8 the choice sketched was an 'add' button at the bottom of the screen, positioned similarly as most smart phone's camera shutter button. By placing a plus sign within the boundaries of the button it is clear for the user that by tapping this button something will be created.

The sketches in Figure 7.7 focused less on the specific layout of the UI elements on the screen, but more so about the different interaction possibilities with an already existing sound zone. Based on this it was decided that movement of the sound zones would be possible with a drag gesture and resizing of an existing sound zone can be achieved by a pinching gesture using two fingers. These concepts follow the conventions that were described in Material Design [21] thus they will be implemented.

For the saving and deletion of existing sound zones the chosen solution is a sidebar menu as it is seen in Figure 7.6. The menu can be brought up by tapping on the button in the top left corner of the screen and it would consist a list of already existing sound zones which can be deleted if necessary. Within this menu the user would have the ability to rename and change the sound zone type.

To get an answer for the last GUI design question the chosen solutions have been derived from the sketches in Figure 7.6 and 7.8. These sketches both show a solution where when the users tap the creation button they are given the options to select the type of the sound zone above the creation button. By switching between the different types the visualisation of the given zone should change accordingly and the user should have the ability to modify the type of the sound zone as it was described before.

7.3 Design proposal

After the design sprints a design proposal was created for both the sound zone designs and the graphical user interface of the application along with its functionalities which can be seen in Figure 7.9. The user interface elements used in the design sketches originate from Material Design [21]. The design decisions that are shown in this section are corresponding to the outcome of the design sprint that were explained earlier in this chapter.

The dome in Figure 7.9 is used as an example of possible visualiser to the sound zone. The different concept sketches show the following:

7.3.1 Figure 1

The first figure shows what the user sees when opening the app for the first time. In this view the camera is already turned on showing it's view and the app is ready for the sound zone creation. There are only two buttons available on the starting screen: In the top left corner a menu button) and a big plus button on the bottom which is when tapped, allows the user to create a sound zone.

7.3.2 Figure 2

After the creation button has been tapped the user is given the option to create a sound zone. This launches a menu, where the user can select the type, shape, and name the sound zone before placing it. In Figure 2 the selected type is private and the shape is a dome. Before placing the zone the user can see the a preview of the selected type and shape. It is also at this stage that the user is able to move the sound zone to a different location which can be achieved by a drag gesture. Resizing can be done by a pinching gesture.

Figure 7.9: *An example design based on the proposed solution*

7.3.3 Figure 3 and 4

In Figure 3 the contents of the hamburger menu can be seen. In this menu the list of the already existing sound zones can be seen which can each be deleted by tapping on the delete button next to them. In Figure 3 there are three sound zones: Living room couch zone, Kitchen zone, and Bedroom zone. The delete buttons are illustrated as the black circles with a cross next to the sound zone names.

By tapping on a specific sound zone the attributes of the zone can be modified (Fig. 4), which in this case are its name, type and shape. By changing the type or shape, the visualisation also changes. If the changes are finalised they can be confirmed by tapping on the "confirm" button.

7.3.4 Figure 5 and 6

Figure 5 and 6 show the other two types of sound zones that are possible to be created with the application.

While in Figure 2 a fully covered zone can be seen which is has been changed to the partially covered in Figure 5. In Figure 6 the type has been changed to public which is visualised using a wireframe.

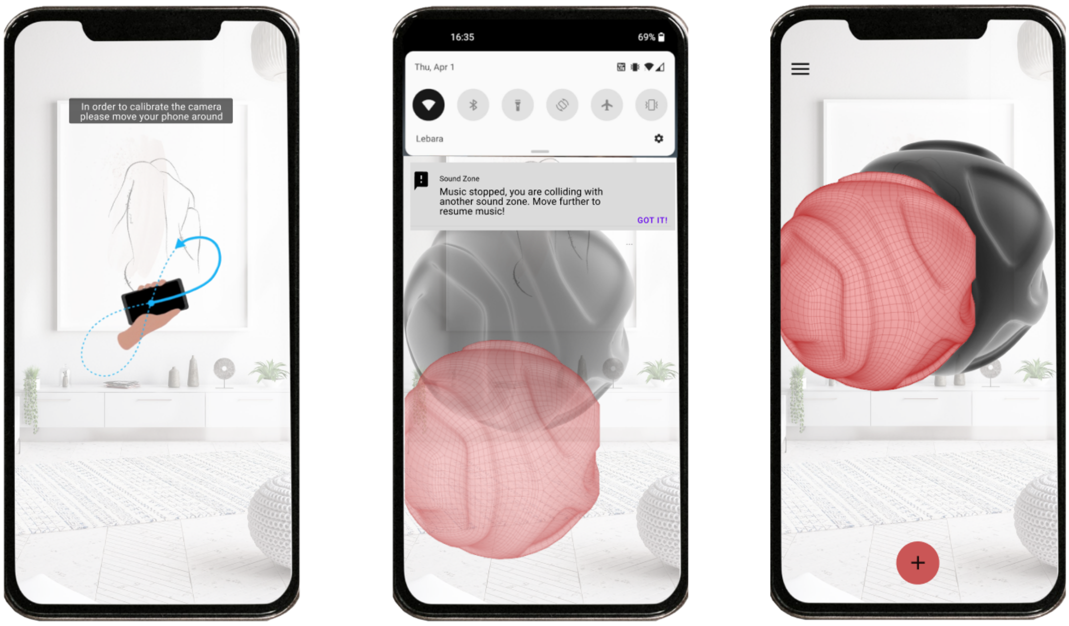


Figure 7.10: *User notifications*

In Figure 7.10 more of the application interactions can be seen. At the start of the application the user is prompted to wave the phone around in order to get the features of the surfaces in the room. This action is referred as the calibration process and it is illustrated on the left picture.

In the case of collision between sound zones the user is notified about the collision via a push notification as seen in the middle picture of Figure 7.10. The user is informed that a zone collision has occurred and they need to move the sound zone for it to function normally.

The picture to the right illustrates what happens if the user tries to create a sound zone in a place of pre-existing zone. The colour of the sound zone is changed to red in order to indicate that action is not possible to perform.

Figure 7.11: *Loading setup*

The Figure 7.11 illustrates what happens after the user loads an existing setup. The zones are labelled with own names which makes differentiation between the zones easier.

7.4 Summary

Four distinct shapes and three unique concepts were generated for visualising sound zones. A UI design was also created which outlines the specific interactions possible for navigating the application as well as the touch gestures used to interact with sound zones. Now that a design has been created it is possible to begin the implementation of the application.

Part III

Implementation

8 System Overview

The first part of the implementation procedure is to identify the essential components of the system which should be implemented. The system architecture which allows for the inclusion of said essential components will then be designed. Afterwards the tools needed to implement the system should be selected such that the tools best suited for the task are selected. First the system architecture used in this project will be described, along with a description of the components making up the entire system. Following this the development tools and their purpose will be explained.

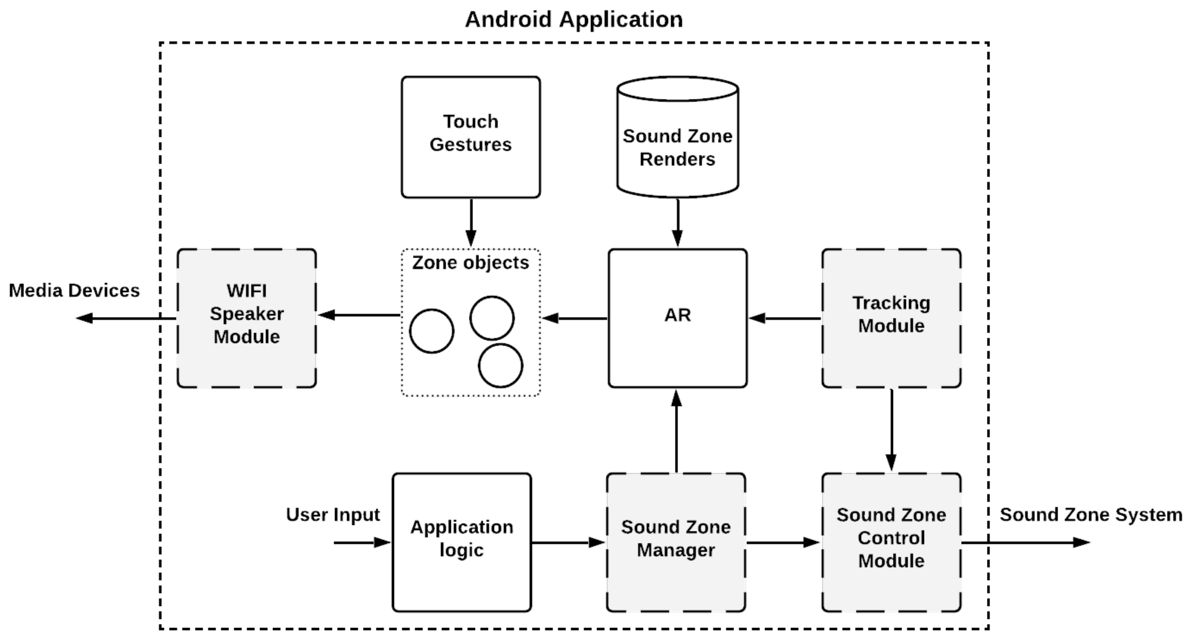


Figure 8.1: *System architecture showing the essential components and the communication flow between the components.*

8.1 System architecture

The most essential component needed to build the application was identified to be the AR module as the visualisation and interaction with AR objects is the core feature. The application must be able to show AR objects which the user can interact with. It was decided that the input and output of the gesture system and the AR system should be handled by a logic controller that ensures that the correct actions are applied to the AR objects based on the given gesture. Based on this the following system architecture was designed as seen in Figure 8.1.

The system architecture is based on a modular design pattern as this allows for a scalable solution where each functionality can be implemented as separate modules. Modules can then be added or removed if needed. This allows for a fast and iterative development cycle. The main reasoning for selecting a modular design pattern is the ease of integrating further components in the future. It is already known that the system would need to integrate four more components and these can be seen as the greyed out modules in Figure 8.1. One such module would be a control module handling the communication with the sound zone system by sending instruction messages to the system about where zones should be placed or if a zone position should be updated. The current application does not include this module, instead the zone creation is performed by the AR module.

Introducing the control module would also introduce a sound zone manager module that the application logic would communicate with. This module would pass the sound zone creation instruction to the sound zone generation system and the AR module.

Currently the application logic is wired directly to the AR module, which highlights that the only difference in the "future architecture" and the implemented architecture is that the application logic interfaces a different module, but the remaining logic flow would not be interrupted. Further additions such as the Wi-Fi speaker system would interface each individual sound zone object and allow these objects to act as Wi-Fi connectable speakers, accessible to any media devices that can stream content over Wi-Fi. However these additional features were not implemented as was described in Section 6.2 therefore only the implemented modules will be described.

8.1.1 Application logic

The application logic is responsible for running the smartphone application, processing any gesture input performed by the user as well as ensuring that communication between the different modules is sent back and forth and processed correctly. Most of the logic that is inside this module is related to navigation between different elements of the UI. The application handles interaction between the user and the UI. Based on the type of user input a task is executed, this means that the application logic module essentially is a state-machine where a state is assigned based on the user input. The general UI logic will be described further in Chapter 9.

8.1.2 Augmented reality module

The AR module is designed to handle the creation of 3D objects and the visualisation of the objects in AR by superimposing said 3D objects on the phones camera. When seen through the phones camera this creates the illusion that the 3D objects are placed in the real world. In this case the 3D objects are different renders of sound zones which will be described further in Chapter 10. The whole purpose of this module is to manage the life cycle of the sound zone AR objects. This module ensures that they are created and placed at the position selected by the user, stays in that position and also removes zone objects when the user wants to remove them. The tasks performed by this module can be summed up as below [23]:

- Calibrate the application by detecting sets or clusters of unique and recognisable features.
- Create 3D objects and render selected 3D models.
- Anchor AR objects to the detected clusters of features, ensuring that the objects do not drift over time or change position when moving the phone.

- Track the AR objects location and the users location.
- Handle interaction with the phone's camera.
- Superimpose the created model on the camera.

An example of these tasks could be a scenario where the user opens the application, here the AR Module will immediately prompt the user to wave the phone around such that it can detect as many features in the room as possible. Following this the user then select a location and places a sound zone at said location. Here the AR Module first creates a 3D object and assigns the correct sound zone render to the object. The object is then anchored to a cluster of feature points at the selected location. Following this the AR Module ensures that the sound zone model is rendered on the phone screen and superimposed on the camera. [23]

8.1.3 Sound zone objects

The *zone objects* are not considered an actual module, rather it should be seen as a representation of the sound zones inside the application. This software representation of a sound zone is linked to the 3D model which will be visualised using AR. This allows the program to store the relevant information about a zone such as its properties. These objects are programmed to detect touch-screen gestures performed on them such as dragging, pinching, tapping and rotation. The objects will be able to apply the correct transformation to the connected 3D model based on the recognised gesture. In the case where an outwards pinch gesture is detected, the zone object is programmed to increase the size of the connected 3D model. This results in the user being able to increase the size of sound zones by pinching them on the screen.

8.2 Application features

In Section 6.3 a list of desired features for the application was outlined. This section will clarify which of these features were implemented in the application, this will at the same time serve as a brief introduction of the functionalities of the implemented application.

The developed application allows creation of sound zones through a menu where the user selects the type, shape and name of the zone. The user then can tap the screen to place the zone. It is possible to place up to three sound zones at any time. After creation the user can through a menu access the properties of the zone where it is possible to the name, shape or type. Changing the type or shape of the zone will result in the sound zone visuals to be updated in real-time. If the user desires it they can delete a sound zone through the properties menu.

Users can interact with a placed sound zone through touch-screen gestures e.g. moving the zone by dragging it around or changing the size by pinching the zone. At creation all sound zones are anchored to the position where they are placed resulting in them not moving around but staying in place unless the user moves the zone. The sound zones utilise a technology known as persistent AR[24] to ensure that the created sound zone can be loaded between different application sessions.

8.3 Development tools

In this section the different tools used for development will be described along with some tools which were considered but was not picked in the end. The usage of each tool will be briefly explained along with a short description of the relevant terminology related to the tool. For the development of this application the following set of tools were used:

- Android Studio.
- ARCore.
- Sceneform.
- Blender

8.3.1 Android Studio

Android Studio[25] is an IDE dedicated to development of Android applications, for this project it was used to develop an Android smartphone application. The main reasoning for choosing Android Studio as the main development platform was the ease of integration with native Android features such as the camera and Wi-Fi connection. Android Studio allows one to easily detect the touch gestures a user performs on a touch screen, this specific feature is very important and makes Android Studio a very good candidate. Furthermore, Android Studio directly supports the use of the Material Design conventions which was the major point for using Android Studio over other development environments. Applications developed in Android Studio are focused around what is called *Activities*[26] sometimes also referred to as a View or Screen. The main activity is the Android equivalent of what is usually called the main function in other programming paradigms. An application can consists of several activities which allow for navigation between different screens within the application. The system structure of the developed Android application will be explained further in Chapter 9.

8.3.2 Unity

Another candidate which could be used as the main development platform was the Unity[27] game engine which allows for building cross-platform AR applications. The benefit of Unity as a development tool is that it allows one to simulate the whole environment inside the Unity 3D world. What makes Unity different from Android studio is that it is a game engine, meaning it offers more functionalities when it comes to interaction and simulation of the 3D objects created inside Unity. However the primary downside of using Unity is the limited integration with Android native features which was offered by Android studio. Unity does not provide a lot of support for developing comprehensive GUI's which would make it difficult to follow the Material Design conventions when developing the application.

8.3.3 ARCore

ARCore[15] is a Software Development Kit(SDK) developed by Google that allows the development of AR applications. ARCore allows the application to detect the environment using the phone's camera(Feature recognition) and also allows interaction with the environment by placing 3D objects which are rendered on top of the phones camera. Within ARCore is a wide array of different APIs which can be used for creating interactive AR experience, one of the key points for selecting ARCore as the AR SDK was the inclusion of *Cloud Anchors*, which allows to store the detected feature points and objects in the cloud. This enables what is known as persistent AR where the user can place AR objects in one session and load the same objects in another session. [15]

8.3.4 Sceneform

Sceneform[28] is a 3D framework built by Google which allows one to build ARCore applications without using low-level 3D rendering APIs such as OpenGL. Essentially, Sceneform adds a layer of abstraction on top of ARCore which introduces a "Sceneview" into the application. The idea with this is that all the AR actions happens inside a scene which contains a tree like structure where each object is placed somewhere along the tree with the scene being the root of the tree. Sceneform also provides an set of gesture recognition functionalities that can detect when specific gestures are performed on an AR object placed within the scene. [28]

8.3.5 ARFoundation

If the choice of development platform had been Unity, AR Foundation[29] would have been the selected AR API for this project. AR Foundation is high-level API developed by Unity which is built upon ARCore, meaning that AR Foundation can be seen as a Unity specific implementation of the ARCore SDK. [29]

8.3.6 Blender

Blender is a 3D modelling tool which allows for the creation and manipulation of 3D models. Blender was used to create 36 different models one of these can be seen in Figure 8.2 where the other models can be seen in Appendix A. The created models were used to visualise the sound zones, and were modelled according to the specifications and design concepts outlined in Chapter 7 and then rendered and exported to a database used by the Android application.

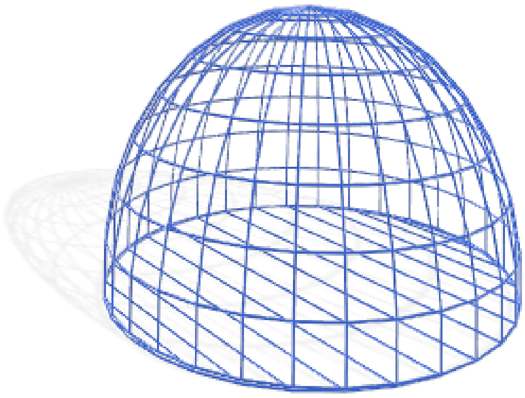


Figure 8.2: *A 3D model of a social type sound zone using a dome shape.*

8.4 Summary

The system architecture that comprises the application has been described along with the modules that make up the architecture. The capabilities for introduction of future modules was also discussed. A brief overview and description of each module was given where the essential functions of each module were outlined. Android Studio was chosen over Unity as the main development platform as Android Studio is better suited for making Android applications where user interaction is the focus point. ARCore was selected rather than ARFoundation because ARCore allows for better integration with Android Studio. One downside to selecting Android Studio over Unity is the lack of 3D world building and simulation which Unity provides. This is however solved by using the Sceneform API which allows importing and rendering of 3D models for ARCore applications in Android Studio. The next chapters will focus on the logic contained in each module and the underlying technologies will be described.

9 | Android Application

The focus of this chapter will be to provide an insight on how the Android application was developed. Specifically an overview of the project structure will be given along with reasoning for the selected structure e.g. why certain templates where used over others. Another focus of this chapter will be the UI design of the application and how the Material Design conventions were considered in the development process.

9.1 Structure and UI design

As briefly explained in Section 8.3 Android applications are focused around what is called activities, different templates for these activities are available inside Android Studio and each are suited for different usages. An example of this is a login activity which contains the needed UI elements to create a login page, another example is an activity containing a scroll-view which creates an interface where the user can scroll through the contents. The different activities contain predefined UI widgets, in some cases this is not ideal e.g. if no activity fits the exact design needs. In this project an empty activity was selected, as a clean slate was desired such that the UI components could be added manually according to the design outlined in Chapter 7.

Besides activities there is another UI element known as fragments which behaves similarly to activities, however fragments are supposed to be reusable UI elements which exists inside an activity. Common usages for fragments are dialog messages or the content of menus. Fragments allow for more modularity in the application, therefore most of the elements in the application has been implemented as fragments contained inside a single activity. The application has been developed using a single activity to contain the main UI elements. It was decided that no further activities would be needed as the main interact happens on the same screen at all times. Creation of sound zones and editing already created zones is implemented through fragments which make up the remaining UI elements in the application.

9.1.1 The main activity

The default screen which the user will be shown throughout their usage of the application can be seen in Figure 9.1.

Figure 9.1: *The default view in the application containing (1) Fragment used to view the camera and AR objects, (2) a Floating Action Button(FAB) for creating sound zones, (3) a hamburger menu, and (4) a load button.*

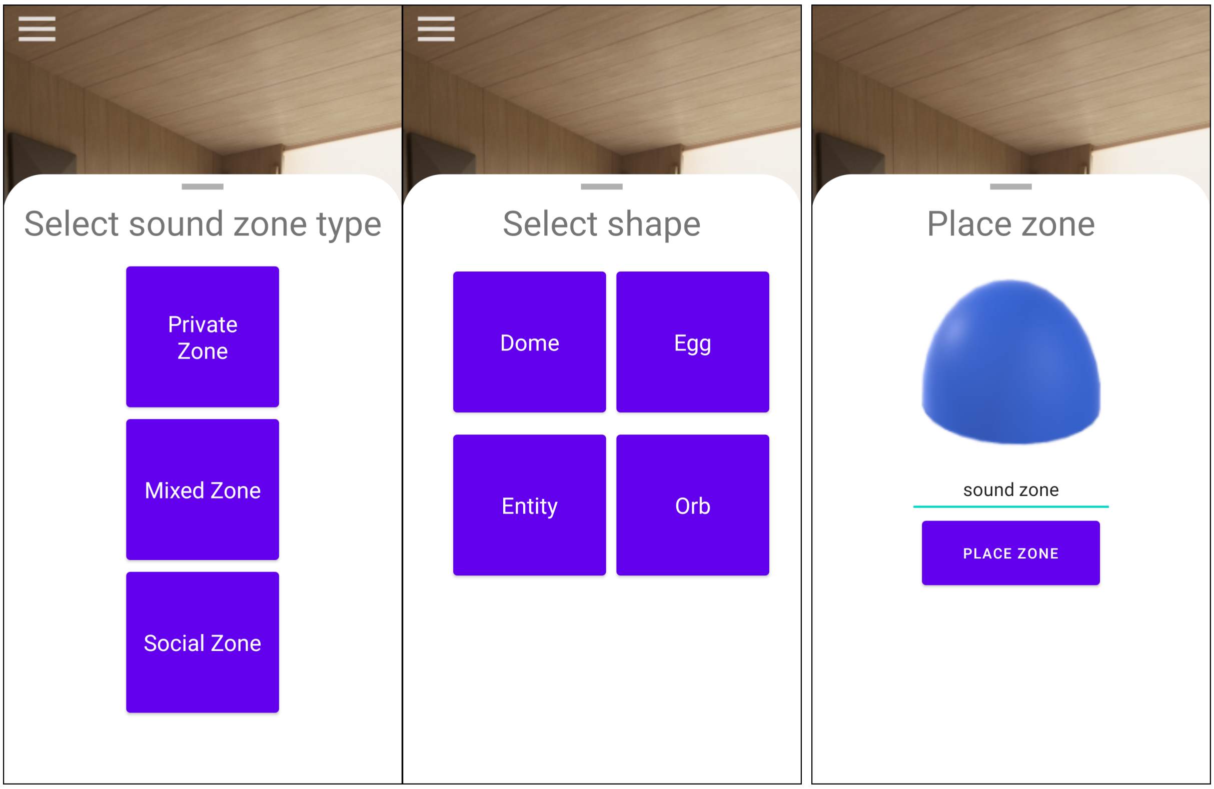
This screen consists the *MainActivity* which is the equivalent to what in other programming paradigms is known as the main, essentially the MainActivity is the entry point for the application. Contained inside the MainActivity is the following four UI elements:

- ARFragment
- Add Sound Zone button
- Hamburger Menu
- Load button

Taking up the entire space of the screen is the ARFragment which is a fragment which handles all the AR interaction and allows the phone's camera to be displayed in the view. Any AR objects which are created will be placed inside this fragment. The remaining three UI elements are also placed inside the MainActivity, they are arranged such that they appear on top of the AR Fragment. The UI elements were placed at the expected positions for their type of widgets as seen in Figure 9.1. It is normal convention that the hamburger menu can be found in the top left corner, and people expect the Floating Action Button (FAB) to be located at the bottom centre, since it is clearly visible and accessible at that location. A FAB is generally used to contain the main action inside an activity [30], in this case this would be adding sound zones, therefore the add sound zones button was created as a FAB. When the FAB is tapped it takes the user through the sound zone creation procedure as seen in Figure 9.2. Right next to the FAB is a button used to load any previously stored sound zones in the room. When tapping on the hamburger menu it opens a sidebar from the side where the user can access any created zones.

9.1.2 Creating a sound zone

When clicking the sound zone creation FAB a sheet will open from the bottom of the screen where the user will be presented with several options for creating sound zones. The first selection which can be seen in Figure 9.2a where the user must select which type of sound zone they would like to create. This can either be: Private, mixed or public. Once the sound zone type has been selected the user will then be asked to select one of four different shapes: Dome, egg, orb, or abstract entity as seen in Figure 9.2b. After having select both the type and shape of the sound zone the user will be shown a preview of the selected zone as seen in Figure 9.2c. The user then has the opportunity to provide the sound zone with a name of their choosing and can then tap on the "Place Zone" button to place the zone. The content of the sheet which is shown in Figure 9.2 is implemented as three different fragments which are inserted depending on the which step of the creation procedure the user is at.



(a) *The user is asked to select between either the three types of sound zones: Private, mixed or public sound zone.*

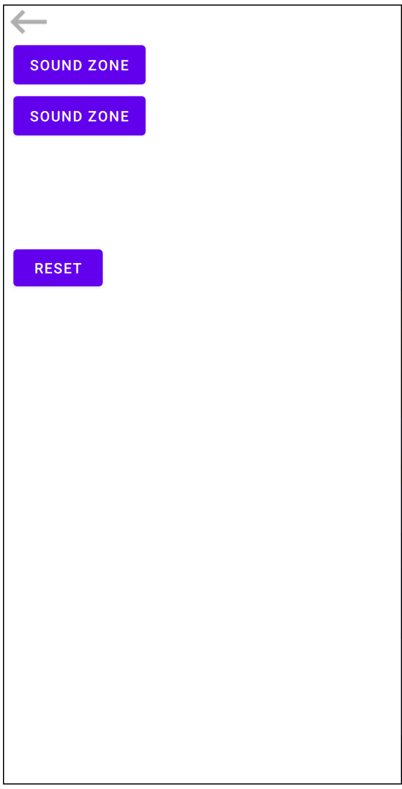
(b) *Selection between the four possibles sound zone shapes: Dome, egg, orb, and abstract entity.*

(c) *The user sees a preview of the select zone and can optionally provide a custom name and then place the zone.*

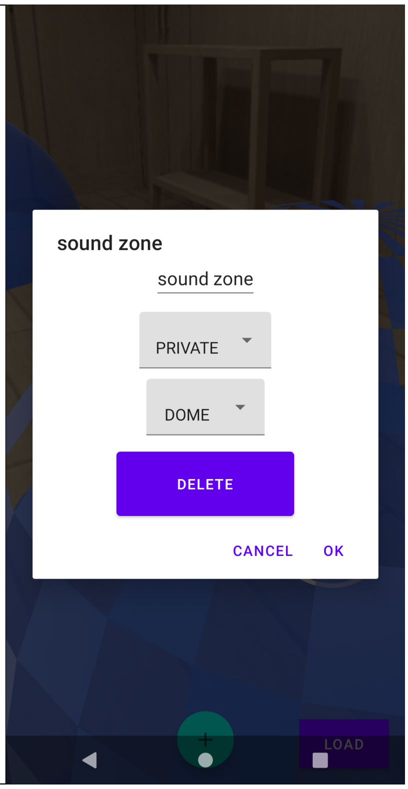
Figure 9.2: *The sound zone creation process where the user has to select which type and shape the zone should have before placing it.*

9.1.3 Viewing and editing properties of a sound zone

After creating a sound zone, it is added to the menu as seen in Figure 9.3a. Inside the menu there is also a "Reset" button. This button allows the user to remove all placed zones and also deletes any sound zones stored, essentially resetting the application. The user can see all the created sound zones in the menu and can tap on a given zone to view the current properties as can be seen in Figure 9.3b. The properties which can be seen are the current zone name, type and shape. In 9.3b the zone simply named "Sound zone" and is of the type "Private" and is using the "Dome" shape. Besides viewing and changing the properties, it is also possible to delete the selected zone. This is done by pressing the "Delete" button which will cause the sound zone associated with the menu to be removed. Pressing the delete button automatically takes the user to the main screen and closes the dialog. The "OK" and the "Cancel" buttons at the bottom of the view are implemented according to the material design conventions with not having a border. This ensures that these buttons do not distract from the contents of the dialog. [21].



(a) *After clicking the hamburger menu the user is shown a list of the created sound zones.*



(b) *The user is able to change the name, type, and shape of the selected sound zone.*

Figure 9.3: *The menu contains all created sound zones and allows the user to edit their properties.*

9.1.4 Overlapping sound zones

In several scenarios sound zone collisions and overlap is possible, in these cases the user should be notified about the overlapping zones. In the notification message as seen in Figure **??** the user is told that the zones are overlapping and instructed to resolve the issue by moving one of the conflicting zones.

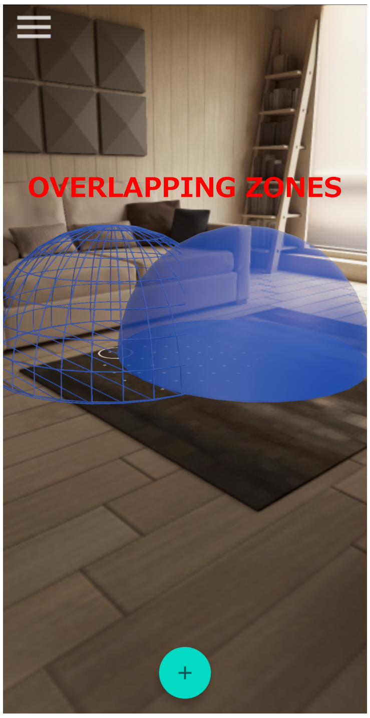


Figure 9.4: *Picture of the warning shown when sound zones are overlapping*

9.2 Summary

The UI elements of the developed application have been described along with the general UI structure. The procedure of creating sound zones has been described together with description of the UI elements at each of the stages of the sound zone creation. The chosen UI elements and the placement of them has been described where the reasoning behind has been chosen according to the conventions of Material Design[21].

10 | Augmented Reality Module

Managing the AR aspect of this application can be described as the most crucial task, but also the most computational heavy part. This module focuses on the AR experience and handles everything related to this, from the initialisation of the AR at startup, to management of AR objects throughout their life cycle. This chapter will be split into several sections each focusing on different aspects of the AR module. First the foundations of the AR module which is based on ARCore will be explained. Then the processes of creating and placing AR objects will be explained and how the Sceneform API was used to streamline this process. In the end of the chapter the usage of Cloud Anchors to obtain persistent AR will be described and the underlying process of using Cloud Anchors will also be explained.

10.1 ARCore

ARCore is developed by Google as a platform to create AR experiences [15]. ARCore provides a lot of the functionality on its own but it also provides access to several APIs which can provide further functionality to the AR application. ARCore enables the phone to sense the environment and interact with it. This is done by gathering information about the environment and using the gathered information for interaction. The essence of AR is that a virtual world is put on top of the real world. Usually the only physical object connecting these two worlds is the device creating the AR experience. These two worlds should each be seen as two Cartesian coordinate frames the *AR Frame* and the *World Frame*. These two frames exist independent of each other, but are aligned such that the origin is at the same location(The phone) as seen in Figure 10.1.

The geometry of the AR Frame is known, but the geometry of the World frame is unknown. Therefore placement of objects in the AR Frame could overlap with existing geometry inside the World Frame. ARCore allows the phone to gather information about the World Frame and uses this information to ensure that objects placed in the AR Frame follow the geometry of the World Frame.

10.1.1 Gathering information

As mentioned above ARCore allows the gathering of information such that AR objects can be placed in a manner that it follows the geometry inside the World Frame. An Example of object placement without information can result in the object appearing within a table or a wall where the same placement using the gathered information would be aware of the table and rather place the object on top of the table.

The entry point for the ARCore API in the application is the AR Session, which contains all the relevant AR processes and manages the entire session life cycle. The AR Session controls when the

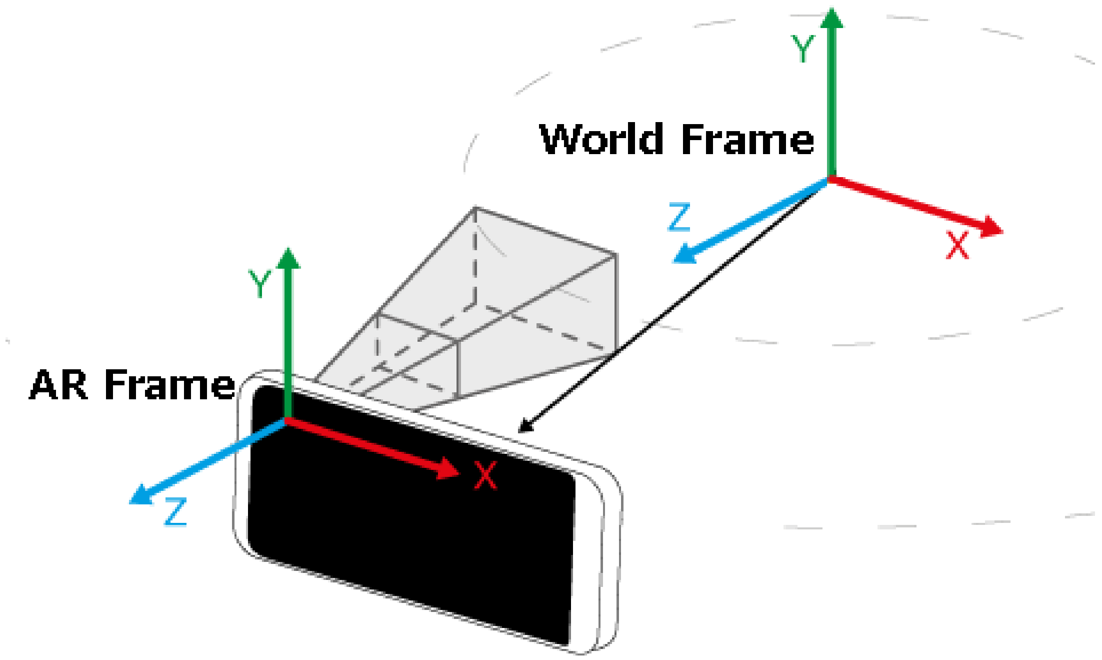


Figure 10.1: *The AR frame and the world frame in relation to each other, adapted from [31].*

application starts and stops the AR experience. Inside the AR session three essential processes run that all contribute to the understanding of the World Frame by gathering information in different ways [23]. The three processes are:

- Motion tracking
- Environmental understanding
- Light estimation

It should be noted that the light estimation methods were not used in this project, contrary to the motion tracking and environmental understanding concepts.

Motion Tracking
Motion tracking allows the phone to keep track of its position relative to the surrounding world, essentially ensuring that the AR Frame and the World Frame stay aligned as the phone is moved around. The tracking is performed using Simultaneous Localisation and Mapping(SLAM)[23], to keep track of where the phone is in relation to the world frame. ARCore uses the camera feed to detect visually distinct points in each image. These points are what is referred to as feature points. By tracking the location of the detected feature points it is possible to determine the change in orientation. This information is then combined with data from the phone's Inertial Measurement Unit(IMU). By combining the output of the SLAM algorithm with the IMU data the current pose of the phone can be determined relative to the World Frame. The location of the phone is not actively used in this project however having the AR Frame and the World Frame aligned at all times is crucial therefore the motion tracking was used. [23]

Environmental Understanding
Environmental understanding is used to detect the size and location of different objects and surfaces in the world and gathers information about these which can then be added to the World Frame. The more information is gathered the easier it is to create a good AR experience. The feature points described above not only powers the SLAM tracking but is also the basis for

understanding the room geometry of the World Frame. ARCore looks for clusters of feature points that lie along either horizontal or vertical surfaces, such as floor, tables, shelves, or walls. ARCore groups these surface into clusters and creates them as a plane. Planes are in the primary surfaces on which objects are placed inside ARCore. Rooms with a lack of features e.g. empty rooms with white or highly reflective walls provide less features than rooms with distinct textures and a lot of objects. In the developed application a "calibration procedure" is performed at startup where the user is instructed through an animation to move the phone around. This is to allow detection of feature points across the room such that planes can be detected. [23]

Light Estimation
ARCore is able to detect the current lighting conditions in the room and can provide this information such that the objects placed in the AR Frame can be lighted under the same conditions as those seen in the World Frame. [23]

10.1.2 Anchors

ARCore will continue to learn about the environment and gather information while the application is in use. Because of this some features of the world are better understood only after gathering a certain amount of features.

The longer ARCore is running, the more reliable information about the environment it gathers. This behaviour introduces a problem, when the original detection of the plane is updated with a newer, richer plane detection. Hence, if the user would select a spot to place an AR object, the object location would shift after the ARCore run time has detected a larger number of features for the same plane, as it turned out that the plane is bigger than it has been originally detected to be.

To ensure that objects placed on a plane stays in the same pose relative to the plane the concept of anchors is introduced. The anchors essentially store the relative coordinate position to the plane they are attached to. Anchors enable objects to stay in the same pose relative to the plane they are placed on, this also considers if the plane is updated due to new features being discovered. In that case the relationship between the object and the plane remains intact even if the plane is moved due to updated feature points.

10.2 Sceneform

To visualise and render 3D models the Sceneform[28] 3D Framework was used together with ARCore. What Sceneform contributes to the AR experience is realistic 3D rendering at run time and the introduction of a hierarchical scene structure where the AR objects are then placed within the scene structure. Sceneform was developed by Google but following the 1.17.1 release the project was archived and both development and support by Google was halted in 2020 as they wanted to integrate it into ARCore. The version of Sceneform used in this project is an open source fork of the 1.16 version of Sceneform[32].

When using Sceneform together with ARCore a few changes are made. The main change is the addition of scenes with a hierarchical graph structure which changes how AR objects are managed.

Sceneform introduces the concept of a SceneView which contains a scene where all AR objects are placed inside. As objects are placed on planes, these are also now placed inside the scene along with the detected feature points. Any objects inside a scene is organised into a hierarchical graph

```
AR Session
└── ArSceneView
    └── Scene
        └── Plane(Feature cluster)
            ├── Anchor
            └── CloudAnchor
                └── AnchorNode
                    └── TransformableNode
                        └── Renderable
                            └── 3D Model
```

Figure 10.2: *The hierarchical structure of a scene graph containing a single object with an 3D sound zone model attached.*

structure where each element has a parent and one/several children. Sceneform also introduces a construct called Nodes, these represent a transformation within the scene graph, these nodes can contain a Renderable, this is the Sceneform object that is rendered to the screen. By assigning a 3D model to a renderable the desired 3D model is displayed in the application.

Sceneform combines the functionalities of anchors from ARCore with those of the Node creating AnchorNodes which is a node that is automatically positioned in the world based on an ARCore anchor. This allows the node to remain relative in position to an anchor while also maintaining the relevant transformation relative to the other objects in the scene graph.

In Figure 10.2 an example of the scene graph can be seen. It is running inside an ARCore session with one AR object placed inside the scene.

Here it can be seen that the AR Session is the root of the tree, containing all the other processes. The ARSceneView is the top layer which renders the contents of an underlying scene.

The ARSceneView is integrated with ARCore such that it gets access to the frames which ARCore processes when doing motion tracking and feature detection. A scene is added with the ARSceneView being its parent object. It should be noted that after the application has started the tree does not change from the Scene and up, only the children of the scene change.

The AnchorNode would be added to the scene e.g. after the placement of an object. An TransformableNode will then be created and added to the AnchorNode as its child. The TransformableNode contains a renderable which Sceneform looks for when determining which objects to render. A 3D model is assigned to the Renderable, in this case it would be a 3D model of a sound zone. The TransformableNode allows for manipulation of the rendered model though touch-screen gestures.

After having placed an object it is possible to perform gestures on the model and the Sceneform transformation system can then detect theses gestures and apply appropriate actions to the Renderable based on the gesture. This can be increasing the size of the Renderable when detecting a outwards pinching gesture.

10.2.1 Placing a sound zone

The procedure followed when placing an sound zone in AR using ARCore and Sceneform consist of several small steps that all contribute to the final results, which is a sound zone anchored to a plane in the AR Frame. The flowchart describing the placement process can be seen in Figure 10.3. It should be noted that the procedure assumes that the application is started and that initial calibration has been performed before placing a sound zone. The procedure is what takes place after "Place Zone" button in Figure 9.2c has been tapped.

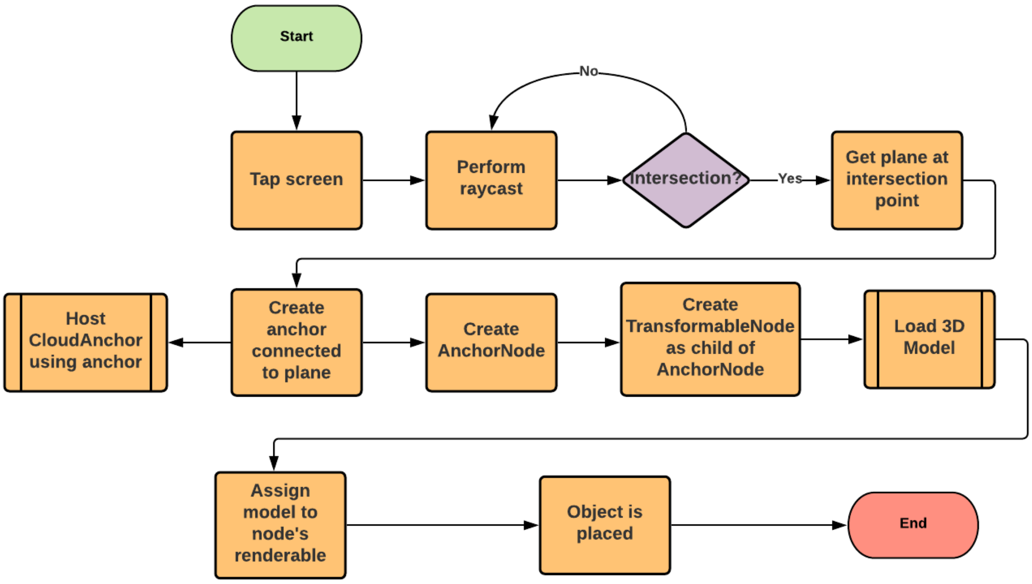


Figure 10.3: *The procedure describing the back-end operations performed when placing a sound zone*

First the user must tap the screen to select where the sound zone should be placed. Then a raycast from the camera is done: The first object that intersects with the ray is determined to be the tapped location. Then the any feature points at the tapped location is found and the plane which contains these features is found. An anchor is placed at the tapped location and is connected to the plane.

Then an AnchorNode is created based on the just created anchor. Then a TransformableNode is created and added as the child of the AnchorNode ensuring that the object will be anchored to the tapped location. A 3D model is then loaded based on the type and shape selected for the sound zone. This model is loaded and assigned to the Renderable of the TransformableNode created earlier. The next rendered frame after adding the model to the Renderable will then contain the 3D model at the tapped location.

10.3 Google Cloud Anchors

To ensure that the AR objects representing sound zones stay in place between application sessions persistent AR is needed. Persistent AR can be defined as AR objects that stay in place even if the application is closed and then opened again. Up until recently saving the position of AR objects and recreating them at the same location has been a hard to solve problem, but with the development of Cloud Anchors by Google it is possible to store pose of an object in the cloud. [24]

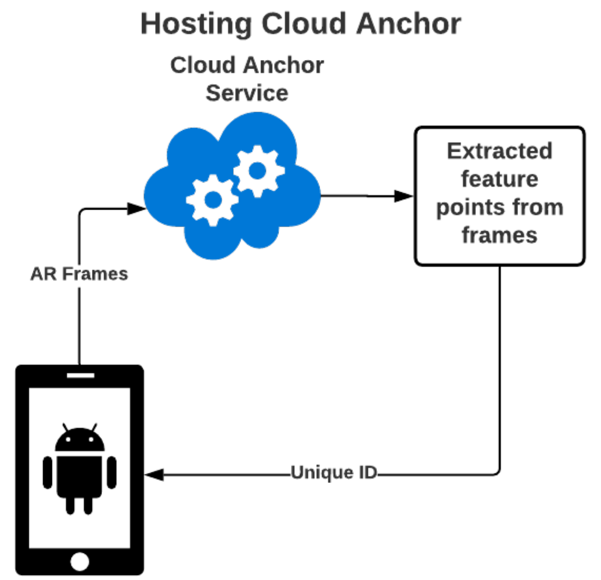
10.3.1 Hosting and resolving cloud anchors

Before the Cloud Anchors can be used an API key must be obtained, this key will be associated with a Google Developer account. To actually access Cloud Anchors the API key must be included in the Android Manifest file such that the application will be able to use the API key to validate the different requests to the cloud service. At a high level the procedure for creating and loading AR objects from the cloud can be described by the following four steps [24]:

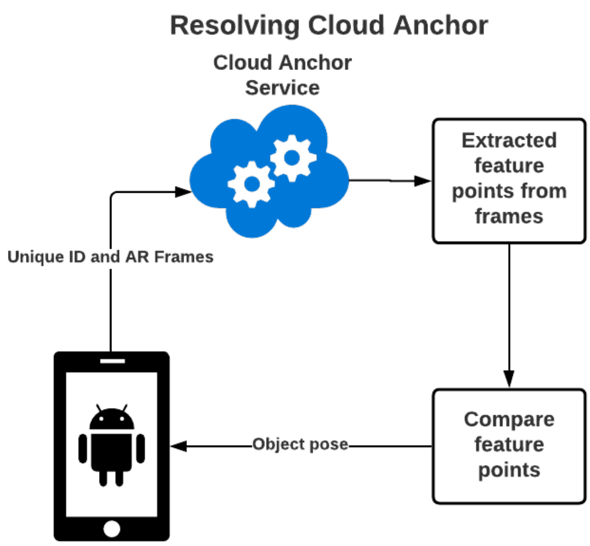
1. A user creates a local anchor(Places sound zone)
2. The hosting process starts, then ARCore uploads data about the anchor to cloud service and an Unique ID is returned.
3. The unique ID is stored using the name given to the sound zone at creation.
4. By pressing the load button ARCore will attempt to resolve any stored ID's and re-create the sound zone AR objects.

The process of storing and retrieving an AR object using Google Cloud Anchors can be seen in Figure 10.4.

When an AR object is to be stored in the cloud the application will stream video data from the phone to the ARCore Cloud Anchor Service which extracts feature points from the streamed data as seen in Figure 10.4a. These feature points are then associated with the pose of the object. A unique ID is generated which can be used to later load the object from the cloud. Passing the unique ID of a created sound zone AR object will initiate a resolving process where the pose connected to the ID will be retrieved as seen in Figure 10.4b. Data is streamed to the Cloud Anchor Service which looks for any feature points in the data. Any feature points which are extracted will then be compared to the stored features associated with the stored pose. Through this comparison it is possible for the Cloud Anchor Service to create a 3D feature map and determine the most likely pose of the stored object. It is then possible to use the retrieved pose to construct an AR object in the same location as the stored object. [24]



(a) *Hosting an anchor in the cloud using the Cloud Anchor Service.*



(b) *Resolving a Cloud Anchor based on the unique ID associated with a hosted object.*

Figure 10.4: *The procedure for hosting a Cloud Anchor and then resolving the hosted anchor at an later point.*

10.4 Summary

The different mechanics of ARCore has been explained and how ARCore is used to create the AR experience in the application. The underlying processes that power ARCore such as motion tracking, environmental understanding and light estimation has also been elaborated. Along with this, the usage of Sceneform together with ARCore was described, specifically how Sceneform organises the AR session into a hierarchical graph structure. Lastly the use of Cloud Anchors to obtain persistent AR was described along with the process of hosting and resolving cloud anchors.

Part IV

Tests

11 | Prototype Evaluation

This chapter will describe the test process of the prototype solution. The prototype solution has been built as outlined in Section 8.2. The prototype solution was aimed to be the minimum viable product(MVP), which would have the fundamental and quality of life features, as described in the sound zone features Table 6.1.

11.1 Prototype evaluation goals

In order to determine the usability of the developed application, goals were established to decide whether the MVP qualifies to fulfil its basic purpose. The usability goals can be summarised in the following list:

- **Application functionality**. Fundamental goal of the tests was to investigate whether the solution lives up to the requirements outlined in Section 6.3 and is indeed capable of providing an interface for sound zone interaction.
- **UX of the developed solution**. In order to investigate whether the developed solution is usable and provides a good user experience the user feedback on the UI design is required.
- **Application learnability & simplicity**. One of the goals for the tests was to determine how long time would it take a user to get familiar with the application as well as whether the user can operate the application with ease.

The application functionality revolves around the interaction features developed in the application. These include sound zone creation, sound zone property manipulation, sound zone scaling, moving sound zones and sound zone configuration saving/loading operations. Apart from the mentioned operations, it was desired to investigate the stability of the application and determine if there are any unexpected or erroneous behaviour within the application.

The goal to investigate the user experience was to determine the user preferences when it comes to the design of the application. These include the location and shape of the UI elements in the application, such as buttons properties: Name, size, location, visibility.

The application simplicity & learnability goals revolve around the navigation of the application. These include the ease of access to the menu, the gesture interaction with the AR objects, user ability to change the properties of the created sound zone according to their preferences.

Apart from these usability goals, we have decided to test whether the sound concepts and types can convey their intended usage based on the design. In order to test this, the concepts for the sound zone visualisers are presented to the user, where the user might point out the strengths and weaknesses of the concept design. It was also decided to test which of the four sound zone shapes would be preferred the most. This was done by showing the shapes to the user and allowing them to select which shapes they liked most based on their personal preferences.

The general goal for both the usability and concept tests is to get as much user feedback as possible so that the developed prototype can be improved in order to accommodate the user preferences.

11.2 Methods

These methods have been chosen for the test and post test evaluation.

- **System Usability Scale**. The System Usability Scale(SUS) is a standardised tool to measure the usability of the application. The SUS has been chosen as a method to evaluate the prototype usability as it allows for a measurable way of determining whether the prototype lives up to its expectations.
- **Interview**. A short interview is conducted prior the tests to gather background information as well as provide them information about the procedure. The user is asked few questions about himself and he is briefed about the purpose of the upcoming tests.
- **Task list**. The test participant is asked to perform a sequence of tasks related to the developed application. They are encouraged to speak out loud and explain their thought process while performing the tasks. Users can ask questions regarding the tasks, if such need arises. However, the test participants is not provided with assistance on how to solve the task, instead they are encouraged to figure out how the system works on their own.
- **Concept shape and ranking scale**. Test participant is provided with a scale for the sound zone concepts and shapes and asked to rank them according to their preferences.
- **The Think-Aloud Protocol** The test participant is asked to think out loud while performing the tasks[33]. Such that they explain every step they perform along the way.
- **Wizard of Oz**.[34] The sound zone audio experience is simulated through a Bluetooth neckband speaker[35], which is controlled by a "wizard"(test instructor), simulating the sound experience as it would be if the actual sound zone system would be online.

11.3 Setup and equipment

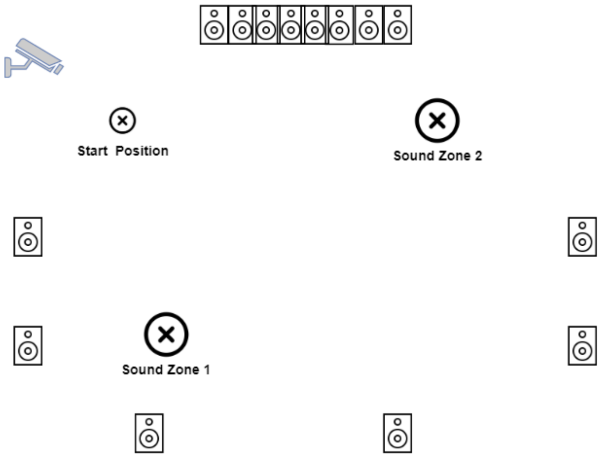
The following list of equipment is used to conduct the tests.

- **Sound zone related equipment**. These include the speaker array and a variety of subwoofers located throughout the room. Due to the restrictions the sound zone setup could not be used for the prototype evaluation. Instead, the sound zone acoustic experience has been simulated with a wearable neckband speaker[35]. The neckband as seen in Figure 11.1 provides a suitable substitute for the actual sound zone due to its property to direct the sound and it allows the user to be able to interact with others while listening to the sound at the same time.
- **Camera**. The room used for the test is equipped with a number of cameras positioned at various spots in the ceiling. One of the cameras has been used to record the test procedure.
- **Markers**. Three locations have been marked on the floor using duct-tape.
- **Android Smartphone** An Android smartphone with the developed application installed on it.

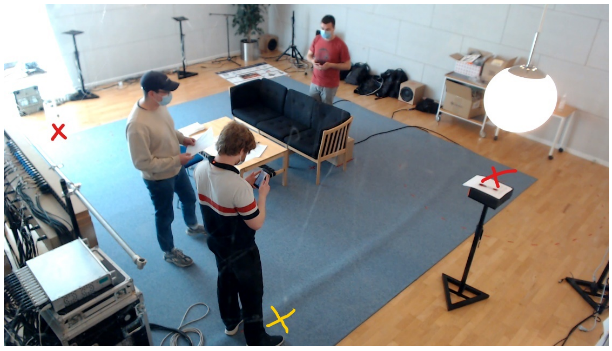
Figure 11.1: *JBL Soundgear Wearable Neckband [35]*

The simulation of the sound zone experience was achieved via mixing two audio tracks. A crossfade slider between the tracks was used to alternate between the tracks based on test subject proximity to a sound zone. Such that if a person approaches zone A one of the track which belongs to the sound zone A will become more apparent and the track which belongs to the zone B will become less audible. In the situation when the user is in between zones A and B, the two tracks are equally audible.

The room setup can be seen in Figure 11.2. There are two marked locations on the floor where the test participant is tasked to place the sound zone. These two spots have been created according to the test design, so that the participants undergo the exact same procedure of placing and moving the zones from one location to another. In a 11.2b a test participant is located at the starting position (yellow cross mark) and asked to place a visualiser at sound zone 1 location (red cross mark)



(a) *Schematics of the room setup with location markers where the sound zones should be placed.*



(b) *Screenshot taken from the camera used to record the user test procedure.*

Figure 11.2: *The setup used for test can be seen here both as a schematic and as screenshot from the recorded footage.*

11.3.1 Supplementary test equipment

Apart from the equipment used to create the test environment, the following list of materials has been used:

- **Consent form**. A standard AAU consent form is given to the test participant to obtain the permission to collect data (video recording & notes).
- **Interview script**. A script has been created for the test instructor to follow such that the users were given the exact same instructions for the test procedure.
- **Sound zone use case sketches**. In order to obtain a better understanding of the sound zone types, the user is provided with sketches illustrating each of the scenarios for the sound zone types as well as explanation from the test instructor.
- **SUS Survey**. User is given a blank SUS score to fill out after performing all the test tasks. Again, he is encouraged to comment about his decisions ranking the prototype.
- **Concept sketches**. All the possible visualisers available in the current stage of the application are printed out and displayed on the board so that the user has a global overview over possible visualisers.
- **Print out ranking sheet for the concept scoring**. Test participant is provided with a score sheet where they is asked to rank the visualiser concepts and the shapes of the sound zone.

11.4 Test procedure

The interview serves as a starting point for the test procedure. At this stage user is briefed about the purpose of the tests and is explained what sound zones are. The interview questions are as follows:

- Are you from Denmark?
- If no, then how long have you been in Denmark and where are you originally from?
- How old are you?
- What is your professional background(study/ current occupation)
- Are you sharing your accommodation with other people (family/significant other/roommates)

11.4.1 Test participants

Ten test subjects has been recruited for the evaluation of the prototype. This number has been chosen based on the idea that the likelihood of finding unique prototype usability related issues diminishes after the number of test subjects exceeds five [36]. We have chosen to have an extra five participants in order to find potentially rare issues. From the conducted interviews it has been found out that the majority of the test participants are engineering students, which might affect the way test participant evaluate the prototype solution.

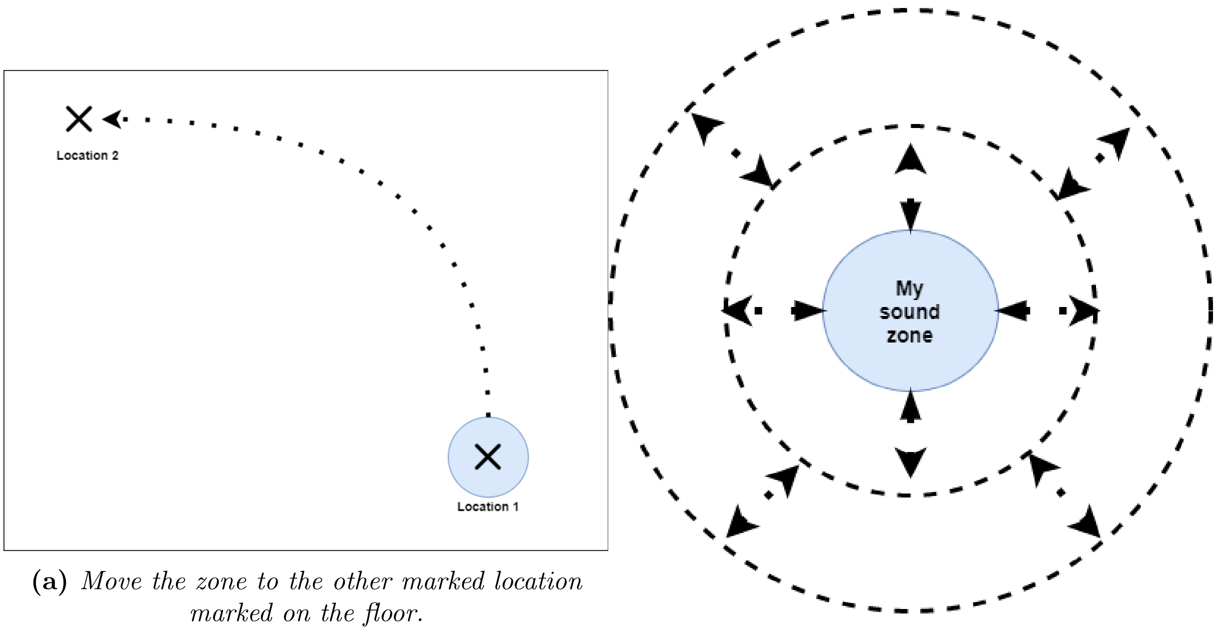
The test participants were primarily male AAU engineering students between the age of 20-32. Half the participants were danish residents with the rest being foreigners. 60% of the test participants were sharing their living accommodation with others while the rest lived alone.

11.4.2 Usability test

After the briefing the test participants were asked to perform the tasks that would ensure that they explore the features of the application. The participants' feedback was transcribed into the notes while the test was taking place. The task list is such as follows:

1. Create a private Sound Zone using the dome visualisation and name the Sound zone "My sound zone" and place it at the location marked on the floor.
2. Move the zone to the other marked location marked on the floor. Figure 11.3a
3. Change the size of the zone such that it matches the marked sizes on the floor. Figure 11.3b
4. Select "My Sound Zone" and change the type of the zone to from private to public.
5. Select "My Sound Zone" and change the shape of the zone from dome to orb.
6. Select "My Sound Zone" and change the name to "Sound zone 1".
7. Walk inside "Sound zone 1" and listen to the music for a few seconds, then walk back to the starting position.

8. Delete "Sound zone 1"
9. Create two sound zones, of arbitrary type and shape and name them "Sound zone 1" and "Sound zone 2" and place one at each marked location on the floor.
10. Walk to the position marked between the two zones and listen to the sounds for a few seconds.
11. Walk into "Sound zone 1" and listen to the sounds inside the zone for a few seconds, then do the same thing but for "Sound zone 2".
12. Walk back to the starting position.
13. Close the application and open it again to check whether the "Sound zone 1" and "Sound zone 2" still exists in the room.
14. Move "Sound zone 1"into the "Sound zone 2". Then act according to any messages shown on the screen.
15. Delete "Sound zone 2". Create a new sound zone with an arbitrary type, shape and name. Now try to place the zone at the position of "sound zone 1". Act according to any messages shown on the screen.



(a) *Move the zone to the other marked location marked on the floor.*

(b) *Change the size of the zone such that it matches the marked sizes on the floor.*

Figure 11.3: *Sound zone moving and resizing tasks*

After completing the tasks user is given the asked to rank the system according to SUS. The system usability scale is seen in the Figure 11.4

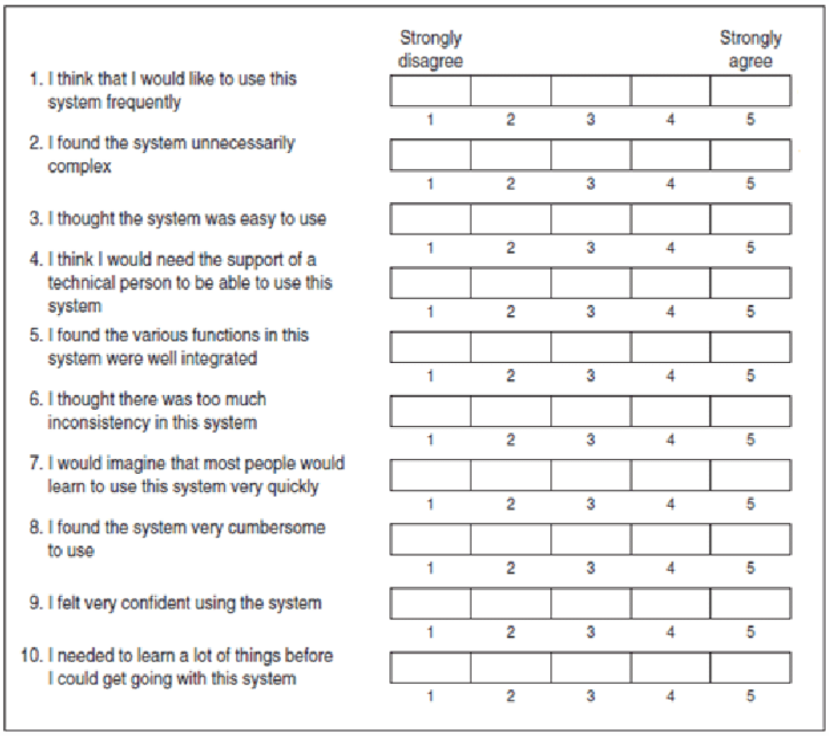


Figure 11.4: *System usability scale*

11.4.3 Concept test

The concept test consists of test subject evaluating the concepts of the sound zone concepts and shapes. These can be found in Appendix A. While reviewing the sketches, the user is provided with an option to see how the concepts are realised in the application.

The test participant is asked to rank the concepts according to the their preferences from 1 (most liked) to 3 (least liked) seen in Table 11.1

Partially covered	Transparent	Icon

Table 11.1: *Template used to rank the sound zone concepts from 1-3*

Similarly to the concepts ranking, the test participants were asked to rank the shapes as seen in Table 11.2.

Dome	Orb	Egg	Abstract Entity

Table 11.2: *Template used for ranking ranking the sound zone shapes from 1-4*

11.5 Summary

In this chapter the goals of the user tests have been defined along with prototype evaluation procedure. The methods and equipment needed to conduct the tests were also listed along with their usage. In the next chapter the results gathered from the tests will be shown.

12 | Test Results

This chapter presents the results of the user test. Firstly the outcome of the usability test will presented, followed by a discussion of the concept and shape test. The results will be shown using boxplots, stacked bar diagrams and affinity diagrams.

12.1 Usability test

After the participants have been informed about the purpose of the tests and answered the introductory questions the usability tests were performed. In this part of the test the users performed the set of tasks that are described in Subsection 11.4.2 using the Android application and wearing the neckband as it was described earlier. After completing the tasks the users were asked to fill out the SUS survey consisting of ten questions as seen in Figure 11.4. The answers of each participants to all of the questions were summarised and can be seen in Figure 12.1. In this table the participants can be seen in the rows, marked by *P1-P10* while the the different questions can be found in the columns marked by *Q1-Q10*. In the last column the SUS score for each of the participants have been calculated which then was averaged, resulting in an overall SUS score of **86.25**. It is important to note that the direction of the scale in every second question changes, meaning that a positive answer in the odd numbered questions equals a high score and in the even numbered questions a positive answer would be a low score. The SUS scores has been calculated by following the principles presented in the original paper on the SUS scale by John Brooke [37].

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	**SUS Score**
P1	4	1	5	1	4	2	5	1	5	1	**92.5**
P2	5	2	5	1	5	1	5	1	4	1	**95**
P3	4	2	5	1	4	2	4	2	4	2	**80**
P4	4	2	5	1	3	3	5	2	4	1	**80**
P5	4	1	5	1	4	3	3	1	5	1	**85**
P6	5	1	4	3	3	2	4	2	3	2	**72.5**
P7	5	2	4	1	5	1	4	3	4	1	**85**
P8	5	1	5	2	4	2	5	1	5	1	**92.5**
P9	5	1	5	1	5	1	5	1	4	1	**97.5**
P10	5	2	4	1	3	1	4	3	5	1	**82.5**
Final score											**86.25**

Figure 12.1: *Table, showing the 10 participants' answers on the different questions in the SUS questionnaire*

A boxplot was created from the final SUS scores which can be seen in Figure 12.2. The boxplot uses minimum, maximum and median values from the resulting scores which allows for visualisation of the distribution of the results along with possible outliers. From this boxplot it can be seen that the lowest score received was **72.5** while the highest was **97.5**. These two values are also happen to be the 10% and 90% percentiles of the data and are visualised in the form of whiskers that are connected horizontally to the box. From the 10 overall scores a median value of **85** was calculated which can be seen in the middle of the box. Two other values were calculated which are the first and third quartiles respectively. The first quartile represent the median value of the lower half of the dataset (after the values have been sorted in ascending order) while the third quartile value equals the median of the upper half of the dataset. The first quartile value was **80.625** while the third quartile value was **92.5**. This boxplot shows that the general feedback from the usability test was positive as there were no results lower than 72.5 and most of the scores moved around 85-90 with low variance which can be seen by the width of the box.

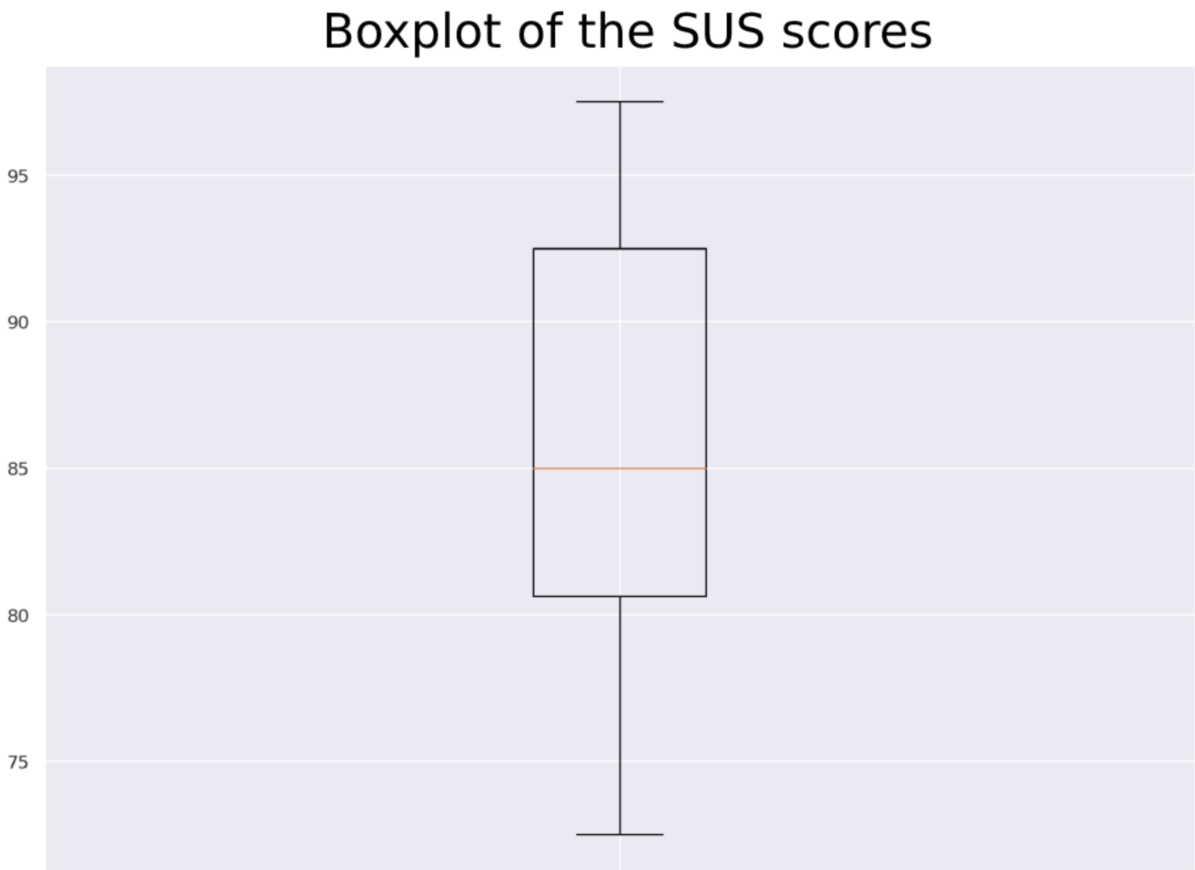


Figure 12.2: *A boxplot which shows the median values of all the SUS scores.*

After plotting the overall scores from each of the participants, boxplots were also created which show the scores from all of the participants on the specific questions in the SUS survey. These plots can be seen in Figure 12.3. It is important to note that the even numbered questions' scores were normalised such that they match the same scale and the odd numbered questions, resulting in a scale of 0 to 4. In these plots it is visible that none of the questions was scored lower than 2 which corresponds to the overall score boxplot above in Figure 12.2. After compiling the results it also became apparent that for the all the questions the third quartile median and the maximum value was the highest possible (4). The questions which received the lowest score were:

- **Question 5:** I found the various functions in this system were well integrated.
- **Question 6:** I thought there was too much inconsistency in the this system.
- **Question 9:** I felt very confident using the system.

Based on the verbal feedback from the test participants to the questions, the low score in Questions 5 and 6 can be attributed to the instability of the application regarding the crashing problem and the difficulties with loading an existing setup. The reason behind a low score for Question 9 could be attributed to all the test participants being first time users of the application, however most of them emphasised that they gained confidence over time.

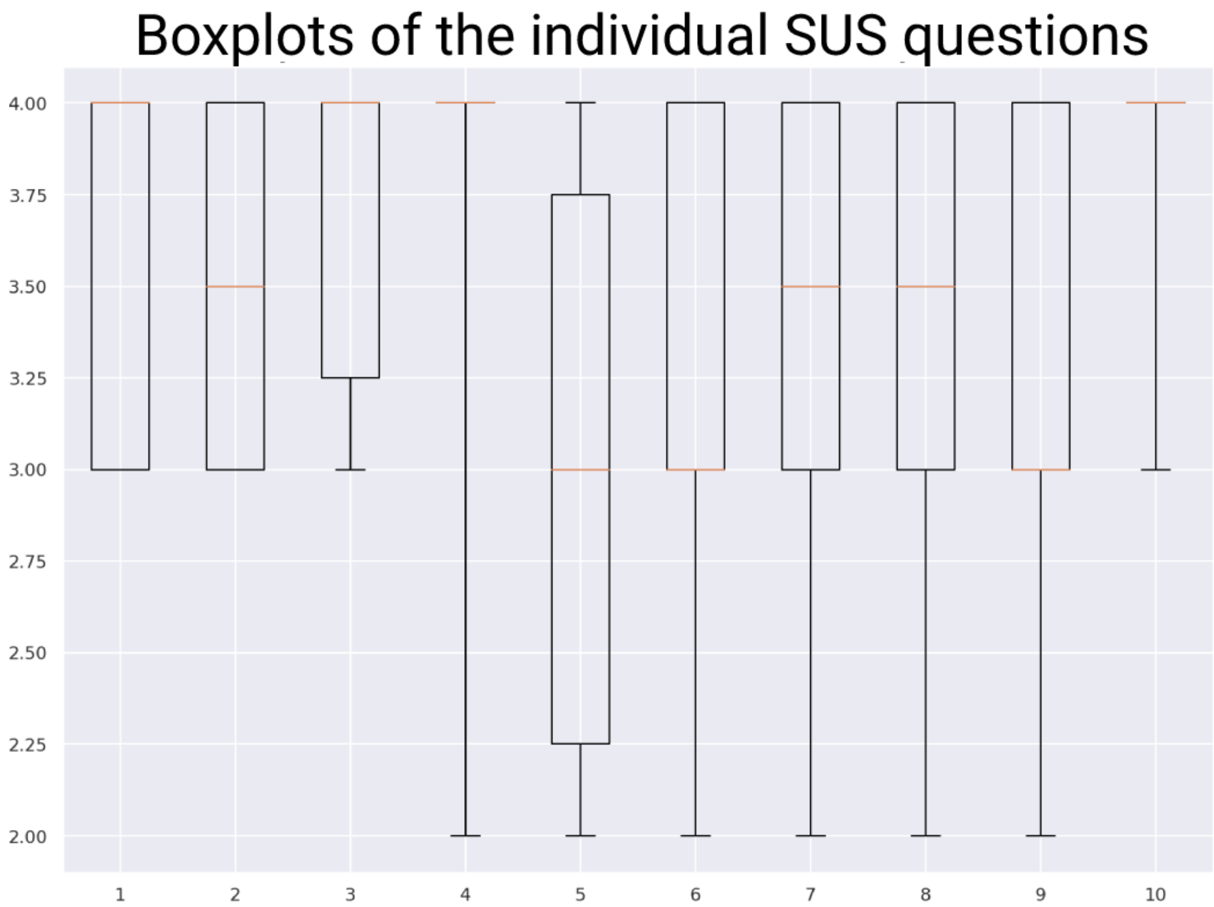


Figure 12.3: *Boxplots, summarising the results of SUS questionnaire questions.*

As the final part of the analysis for the usability test an affinity diagram was created which gathers all the feedback data that was recorded from the test participants. The diagram can be seen in Figure 12.4 below. The comments were sorted into three different groups and the comments that appeared more than once are marked with the number of recurrences.

The first group is a collection of suggestions that consist of different features that the users would like to have access to in the application and that are currently not implemented.

The second category consists of different errors and bugs that the test participants have experienced during the usability test of the application. Most of these comments revolved around the systematic instability of the application, specifically several crashes and errors regarding the loading of previous sound zone setups.

The third category comprises of general comments that are neither suggestions, nor pointing out errors specifically. These comments however are also showing an important problem regarding the movements of existing sound zones which could be implemented in a more robust way according to user feedback.

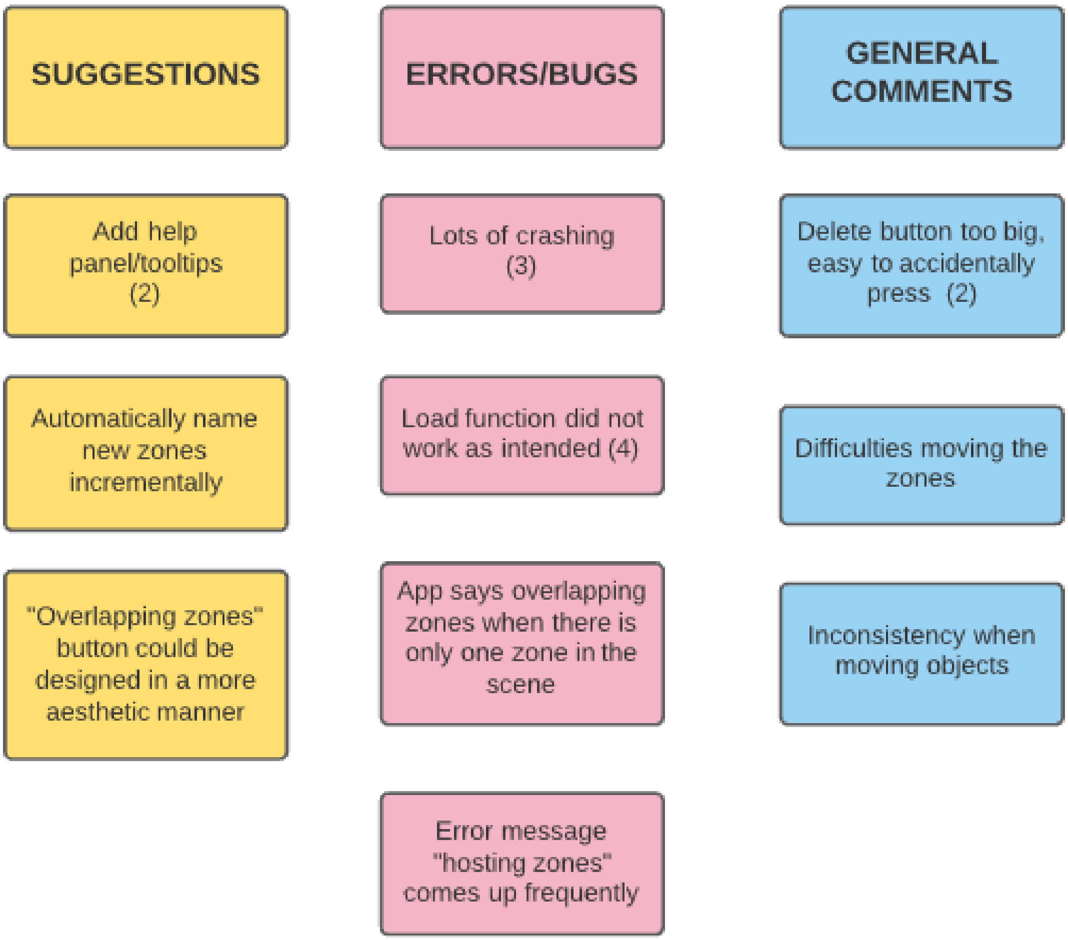


Figure 12.4: *Affinity diagram consisting the user comments on usability that were recorded during test.*

12.2 Visualisation test

In this section the results of the visualisation test were discussed. This part of the test was divided into two smaller parts: Firstly the test participants were asked about to rank different visualisation concepts according to their opinion. Secondly, they ranked and gave their opinion on the different visualisation shapes.

In Figure 12.5 the results of the concept rankings can be seen. The participants were asked to rank the different visualisation concept on a scale from 1-3, where the 1 is the most preferred concept and 3 is the least preferred. The participants have ranked the concepts based on their personal preference in regards to which concept they think best visualises the intended usage of the zone type. In the Figure 12.5 it can be seen that the partially covered concept was preferred by the most participants.

Some of the participants have chosen the transparent concept without hesitation as they have found that this concept is visually appealing and aesthetic than others. Most people who did not rank the transparency concept high, commented that it would probably have problems rendering the sound zones visible enough, depending on different conditions (e.g. lighting) and furthermore it could be problematic to differentiate between the different types since the transparency levels are not easily distinguishable. Regarding the icon concept, some of the participants had some comments regarding the icons themselves as in their opinion "different icons can have different meaning to different people". This feedback indicates that it is a challenging task to find an icon for the concept which would carry on the same perception universally to all the possible users.

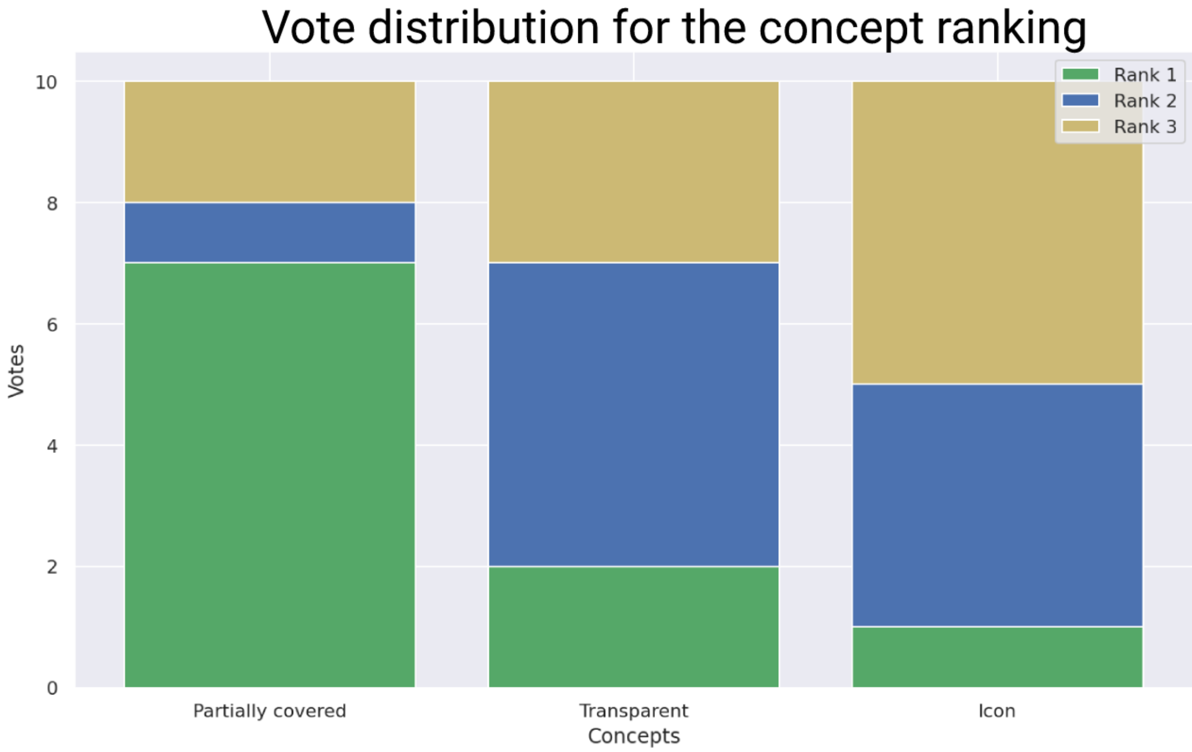


Figure 12.5: *The vote distribution of the three concepts' rankings as ranked by the test participants.*

In the second part of the visualisation tests the users were asked to rank the different shapes, these being the dome, egg, orb and the abstract entity. The test participants were asked: "Out of these available shapes, which one do you think best describes a sound zone?". The final score calculation follows the same logic as with the concept rankings before.

Based on the answers received from the participants, almost all test participants preferred the dome shape as the number one visualisation shape. The reasoning behind their choices were usually based on the physical characteristics of the dome shape, specifically that it shows the footprint of how much area does the sound zone actually take in the room, thus it makes it easier to place the sound zone.

The second place in the rankings was taken by the abstract entity, however the test participants were told to think about this shape as object which has moving walls/sides (possibly to the rhythm of the music that is playing inside the sound zone) which was not yet implemented at the time of the user test.

The remaining two shapes were not as popular as the aforementioned ones and most test participants did not find them visually appealing in terms of sound zone visualisation nor did they find these shapes useful in representing a sound zone in terms of size and location.

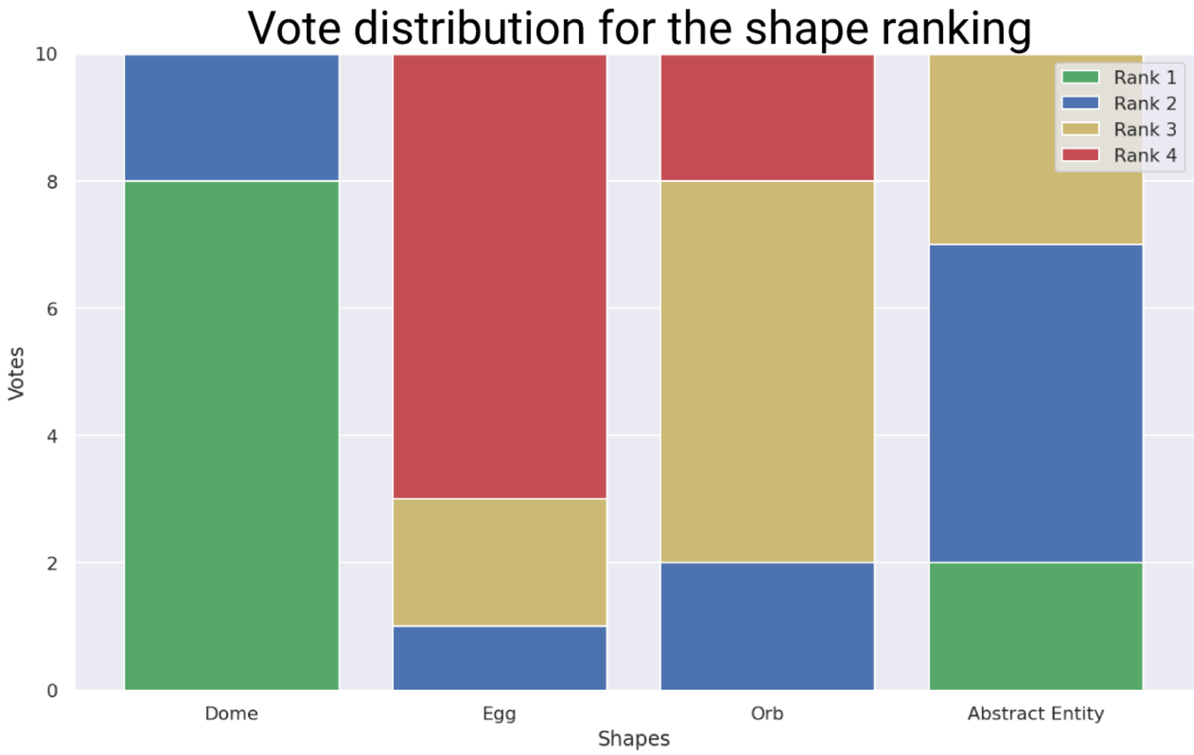


Figure 12.6: *The vote distribution of the four shapes' rankings as ranked by the test participants.*

Similar to the usability test an affinity diagram was created which collected the user feedback during the second part of the user test which can be seen in Figure 12.7. It is important to point out that since the questions already had predefined answers, the participants did not have a lot of freedom in their answers which is reflected in the structure of the diagram itself. Since the second part of the test consisted of two ranking surveys the affinity diagram is also built of two sides: concept specific comments, which correspond to comments made for the concept selection part of the test and shape specific comments which were said during the second part of the visualisation test. Within both of these categories the comments were split into two subcategories, namely suggestions and general comments.

In both categories suggestions mostly included the creation of new shapes or combination of already existing concepts/shapes. These include for example: the idea of an abstract dome (mixture of abstract entity and dome) or a new concept where the partially covered concept would be combined with the icon concept thus making the visualisation more explicit. During the analysis of the comments on the concept selection it was discovered that most test participants found visual aesthetics important which could be one explanation as to why the icon concept ranked last since it is arguably the least visually appealing in its current state. Another important thing to note is that some participants only ranked the abstract entity higher than some shapes when the assumption was made that the sides of the shape are moving/pulsating, a feature yet to be implemented.

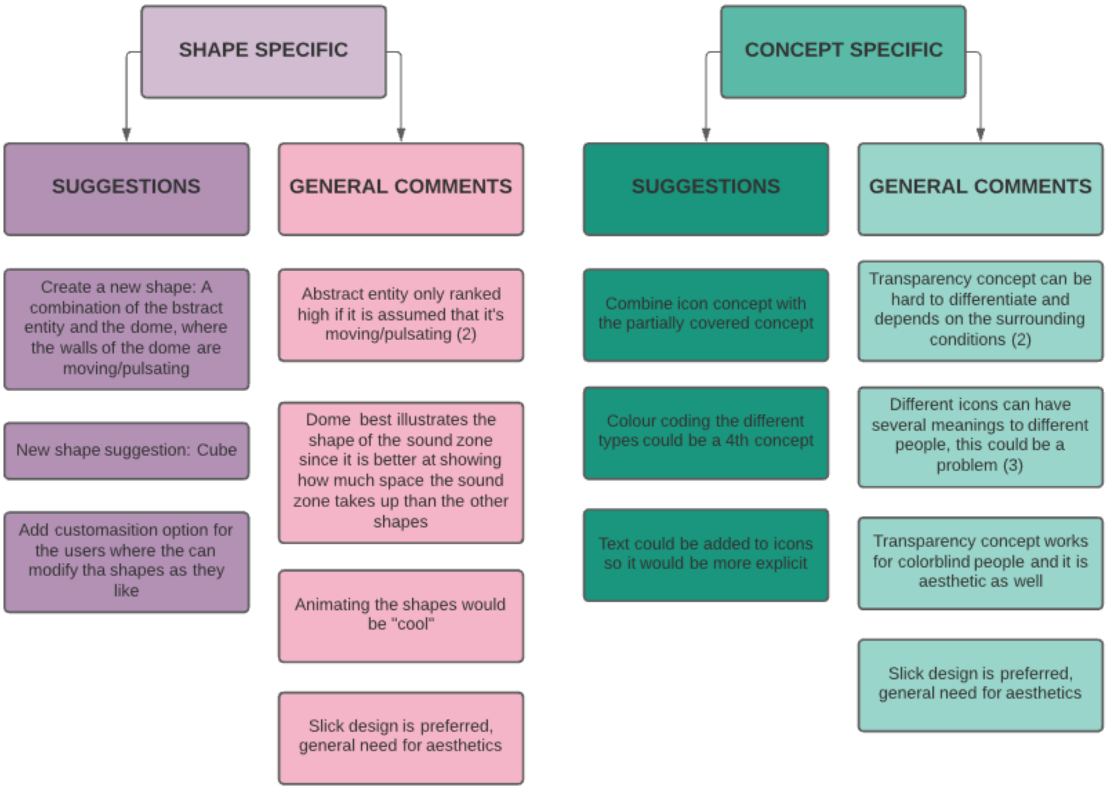


Figure 12.7: *Affinity diagram showing the comments the participants made during the second part of the user test.*

12.3 Summary

The test results will be discussed in the Discussion Chapter 14. Over the course of the user tests several suggestions and comments were extracted based on the feedback received from the participants. This feedback will serve as a basis for the following Improvements Chapter 13.

13 | Improvements

This chapter will identify the issues leading to the usability problems identified in the previous chapter. The focus will then be on designing and implementing improvements which could resolve the problems identified. The user feedback has been processed and it has been decided that the improvement of the prototype can be split into two categories: usability and stability improvements. The usability improvements are the subject of the most concern, since they revolve around the design of the application.

13.1 Usability improvements

The usability improvements revolve around changes in the application design. These changes should improve the ease of navigation of the application, improve the usability of implemented features, such as access to the menu, visibility issues and overall improved user experience.

The test participants' feedback regarding the usability issues has been made into a list where the fixes are prioritised. This ranking has been created based on the frequency of the problem appearing while we evaluated the prototype.

1. **Make the delete button smaller and less apparent**. The new delete button has to be smaller compared to the original version. The position of the button will change so that is located further away from zone concept and shape selection buttons such as seen in Figure 13.1b.

2. **Sound zone name numbering increases incrementally**. If the user didn't name the zone manually, the zone should have its own designated numerical number, so that the next nameless zone will have an incremental value. For example, the first zone is named "Sound zone 1" and next will "Sound zone 2". This fix is necessary to provide a possibility to distinguish between the not named zones as seen in Figure 13.1a.

3. **Make overlapping sound zones text into a toast message**. This is a visibility issue. The message which comes up in case of placing the zone on top of another zone needs to be more clearly visible to the user.

4. **Create bigger toast message when placing zone and make it stay longer on the screen**. The instruction message "Tap to place the zone" needs to be made move visible as well as the time how long it stays visible needs to be extended. During the prototype evaluation users had hard time noticing this instruction message due to its small size and short lifecycle.

5. **Add a "back" button to the zone creation menu**. The prototype version didn't have an option to go back to the previous state of the menu. Instead, user had to go through the entire zone creation process and only afterwards would they be able to change the properties of the zone (see Figure 13.3). Hence, a "back" button needs to be added to provide an option to come back to the previous menu screen.

6. **Add labels to the drop down menus**. Each of the drop down menus needs to have its own label for the user to be able to navigate easier through the application.

7. **Insert images instead of the text.** In the sound zone creation menu (Figure 13.3) it is desirable to have illustrations instead of the text while selecting either the zone type or the zone shape.

8. **Handle the error message.** During the prototype evaluation users have sometimes encountered an error code message which would not disappear. This message needs to be transcribed into a message which is easier to interpret and also not linger on the screen for more than a few seconds.

9. **Add help panel**. To help the users to understand what the sound zone types are a help menu needs to be introduced. This menu provides an explanation what is the purpose of the particular sound zone type.

10. **Guide & Assist feature**. Some of the test participants have suggested to implement a guide or a make a manual how to use the application. This improvement is ranked low due to the low occurrence of this issue while evaluating the prototype. This improvement can either be implemented as by creating an instruction walk through of the application at the first launch or the application could have a manual/help panel implemented.

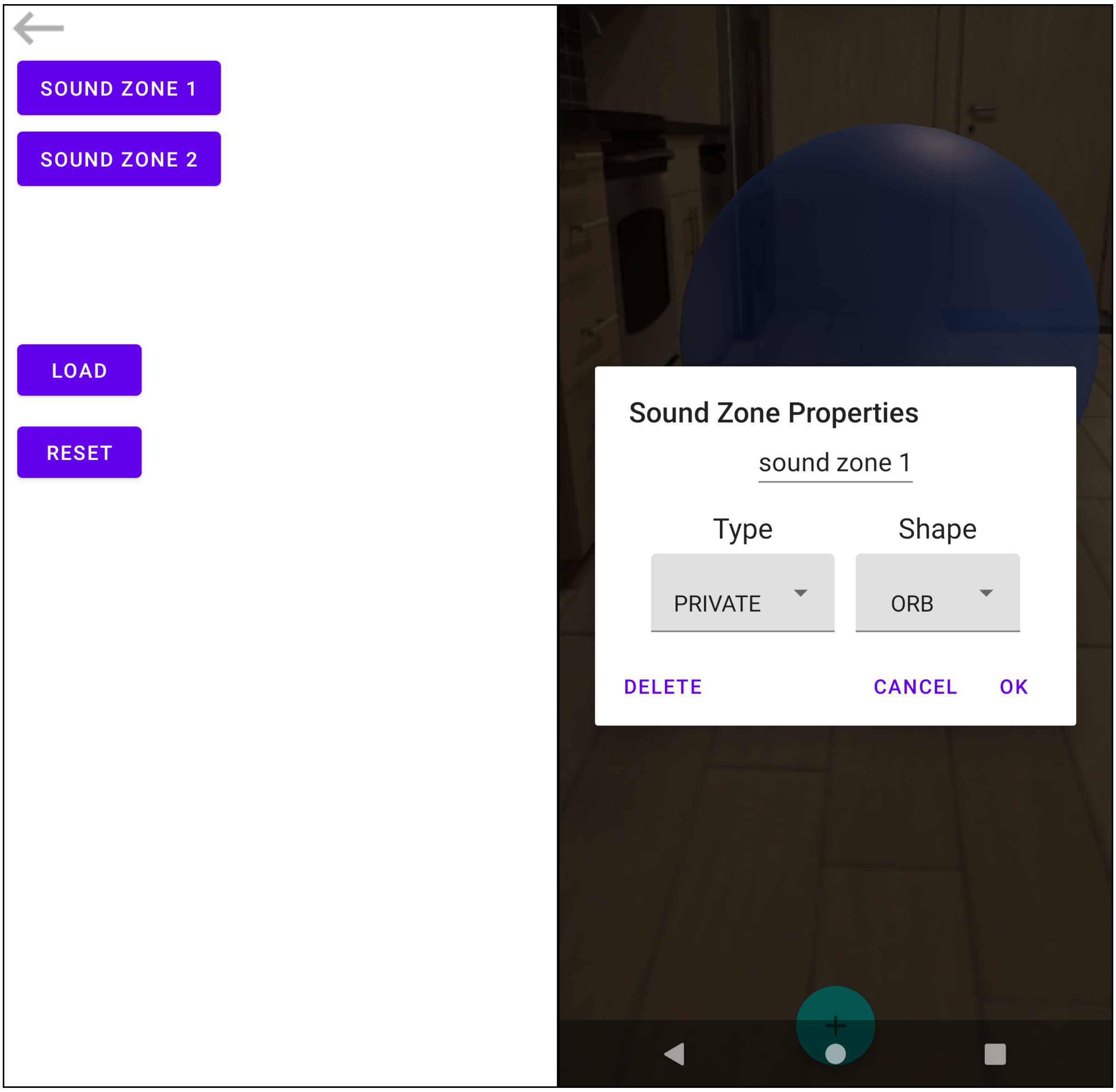
13.2 Implemented improvements

Based on the feedback, ten usability issues was identified, out of these issues the first seven issues were solved and implemented during the second implementation stage. The sound zone editor menu was changed to fit the user feedback as seen in Figure 13.1b where it can be seen that the delete button has not been moved down to match the "ok" and "cancel" buttons. The dropdown menus for selecting shape and type has been made slightly smaller and both have been assigned a label. The title of the menu has also been changed to "Sound zone editor" to better reflect the options possible in the menu.

The menu accessed when clicking the hamburger icon was also changed slightly as seen in Figure 13.1a. The "load" button was moved from the main screen into the menu, next to the "reset" button as these features are related to each other. Each sound zone now also automatically increments their name when they are created such that the first zone will be "Sound zone 1" and so forth. This is reflected in the buttons seen in the menu.

The menu shown after clicking the "Add sound zones" FAB was also adapted according to the feedback. Before all the buttons used for selecting type and shape had a label indicating their value. This has been changed such that the button includes both an image and a label indicating either type or shape as seen in Figure 13.3.

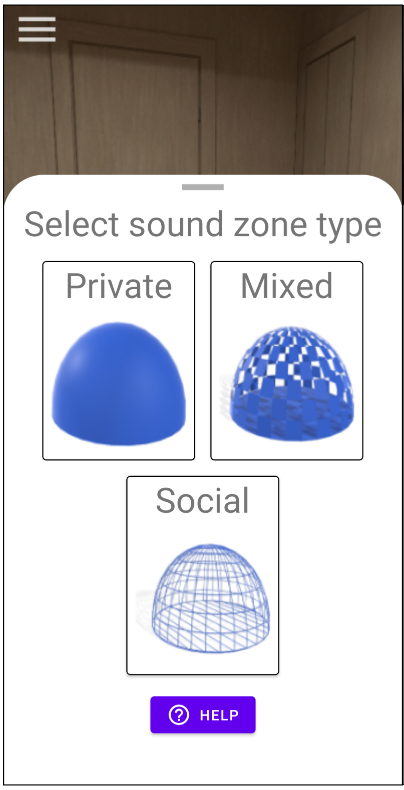
The warning message which is shown when a sound zone is overlapping with another zone has been changed from a static text shown on the screen as seen in Figure 9.4 to a toast message which is shown on the screen and then disappears when the overlapping stops as seen in Figure 13.2.



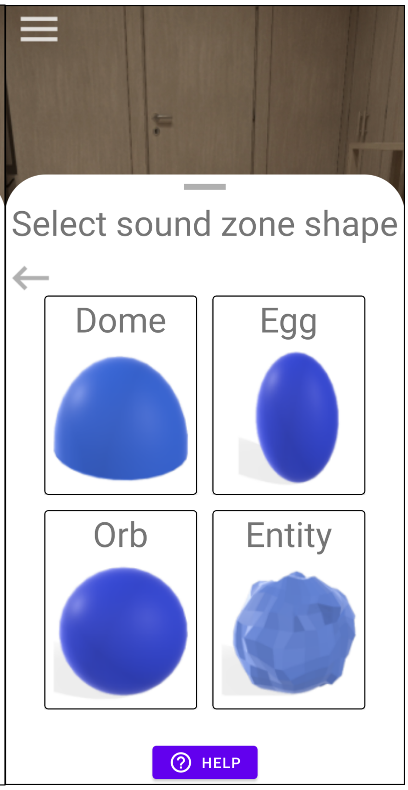
(a) *New look for the Delete button.* (b) *An incremental naming of the sound zones*

Figure 13.1: *The adapted version of the sound zone properties menu with re-arranged layout and less focus on the delete button.*

Figure 13.2: *The toast message shown to the user when sound zones are overlapping.*



(a) *The different sound zone types are now visualised rather than shown as text only.*



(b) *The shapes are changed into images instead of text and a back button was added.*

Figure 13.3: *Updated version of the sound zone creation menu.*

13.3 Stability improvements

The stability improvements are aiming to solve the crashes,different bugs and error messages encountered during the tests. The stability issues encountered can be seen in the list below:

- **Application robustness.** The prototype version crashes frequently, thus it needs to be found out what causes the crashes. Based on observations during the user test crashing of the application occurred when a user attempted to place three or more zones in the scene.
- **Inconsistent behaviour of the load function.** There are bugs associated with the load button, where sometimes it does not perform as intended by design. This error occurs when a user creates multiple sound zones, restarts the application, then tries to load the previously created zone setup using the load button. While encountering the bug, the zones do not appear at the same location as where they were originally placed.
- **Zone overlap message inconsistency.** Sometimes the "zone overlap" message appears when there is no such event taking place. This happens as "borders" of the sound zone models were not defined manually. Instead Sceneform approximates the space the object takes based on the 3D model. This causes the borders of most objects to be a square fit to the model rather than a circle. This sometimes cause the actual border used for overlap detection to be bigger than the model.
- **Random error messages showing up on the screen.** These messages are associated with ARCore and Google Cloud Anchors and should either be handled such that the errors are not shown to the user, or the correct instructions should be passed to the user so they can act accordingly.

The stability issues need to be addressed to achieve more consistent and reliable performance of the application. It was deemed that it would take to long to solve the stability issues during the second implementation stage. How to solve the stability issues will be described more in the discussion and future works chapters.

13.3.1 Summary

The feedback from the user test was examined and several issues with the application were determined. These issues were split into two categories; Usability issues and stability issues. It was decided that the usability issued would be solved and that the stability issues would then only be solved later due to time constraints. After identifying the problems causing usability issues, these problems were solved and implemented into the application. In the next chapter the general findings of the report and the test results will be discussed.

Part V

Evaluation

14 | Discussion

In this chapter the test results,application development and general approach to the project will be discussed. The overall evaluation of this project is based on the final problem statement as seen in Section 5.1 as well as the general requirement established in Section 6.3. The overall project goal was to develop an application which allowed for interaction with sound zones which would be visualised using AR. It was desired that the visualisations would convey the purpose of a specific type of sound zone. A set of user tests were conducted to both establish if the developed application allowed for easy interaction with sound zones, and if zones conveyed the intended purpose based on the visualisations associated with them.

14.1 Test execution

During the test users were introduced to the concept of sound zones at the beginning of the test procedure. The concept of sound zones can be described as quite abstract and hard to grasp as there is no direct equivalents to sound zone technology which the users might be able to relate to. Therefore it could often take some time before the users understood the concepts of sound zones and got and idea of how it worked. On top of being introduced to the sound zones user would also be introduced the different types of sound zones: private, mixed and public. The introductions were spaced out such that first the participants were introduced to sound zones, only introduced to the different types at the end of the test. Having to grasp both of these abstract concepts in a brief amount of time had some test persons struggle to grasp the concept of having several types of sound zones. It could be considered that the introduction of both concepts at once was too much. In the end all the tests persons acknowledged that they understood the concepts, but the introduction to both the general sound zones and specific types could have been structured in a more pedagogical manner. This could potentially lessen the confusion when introducing the users to these concepts.

When conducting the user tests it was not possible to use the actual sound zone system. It was assumed that it would be possible to access this equipment until a few days before the user tests were scheduled. We were unfortunately informed that B&O would not allow the usage of the equipment for our user test. As an alternative to the real sound zones a Bluetooth neckband was used as described in Section 11.3 and one of the group members controlled the audio throughout the test. In general the results from the user test can be inconclusive as we cannot claim that the test results would be similar if the test participants could experience the actual sound zone system. Despite this fact, we can argue that the desired test outcome was obtained, since we were aiming to investigate the usability of the developed Android application. At the current stage of development, the application can be treated as a standalone system which does not depend on the presence of the actual sound zones. Instead, the application is designed to provide an interaction tool with the sound zones in AR. The usage of the "wizard of oz" seemed successful as several users were not aware that the audio was controlled manually. The simulation could have been improved by building the simulation into the application and tracking the users position in relation to the sound zones. Then the volume of each zone would automatically be attenuated based on how far the users was from a given zone.

One persistent issue that was encountered was sporadic of the application which hindered the test execution. It should be noted that it only happened for some of the test participants. This might however have impacted the test results during those particular test runs.

14.2 Test results

For the user test a total of 10 participants were recruited, it should be noted that 8 out of the 10 participants had an engineering background. It can be said that this causes a major bias regarding the usability results. It can be assumed that this particular group of people can be considered above average when it comes to interacting with technology. However the two people without an engineering background both performed on par with the remaining participants as seen in Figure 12.3 and Figure 12.2 where no statistical outliers can be observed. However it can be assumed that if the same tests were conducted on a more varied group of participants the results might be different. In Figure 12.1 it can be seen that one participant ranked the application slightly worse than the others, this outlier was identified to be studying product design and psychology which particularly focuses on user interaction. The participant could have been more critical about the application due to their background and intimate knowledge with user interaction which might be the result for the slightly lower score.

For the evaluation 10 participants were recruited, and this is a adequate number for conducting the usability tests as the amount of new discovered errors in the application decreases rapidly after 7-8 test participants[38]. However for the visualisation evaluation it is uncertain if 10 participants is enough, as it the evaluation is based on personal preference. Despite the fact that we have tested the solution on 10 people, no new emerging trends have been observed. The test results confirm that there is a clear winner when it comes down to the sound zone concept as well as the sound zone shape. These are the partially covered concept and the dome shape. It would be preferred to have a large amount of participants to rule out biases due to personal opinions and instead be able to determine reoccurring trends in the rankings. The same can be said regarding the participants background, it would be desired that the participants represented a more general demographic, rather than a small subset of people(engineers).

In general the usability test can be said to have been a success with the overall SUS score being 86.25 out of 100 possible points. The result shows that the users found it easy to use the application or learned how to use it very quickly. There we some issues which caused some of the scores to be lower than others, such as the application being unstable and crashing.

This is seen in Figure 12.1 where the questions which were ranked lower compared to others were questions about the stability of the system: "I found the various functions in this system were well integrated" and "I thought there was too much inconsistency in this system".

Another low score was related to the statement: "I felt very confident using the system". We can argue that the score can be attributed to the stability issue as well, since the users were loosing confidence when the application was crashing. On top of this, the participants have tried to use a system which they had no prior knowledge or similar experience with, therefore the low score for the confidence in the system is such as expected by us.

Nevertheless, the majority of the test participants seemed enthusiastic about the concepts and the developed application.

The issues related to stability did however not seem to impact the overall SUS score much as the final score is still considered to be above the average SUS score of 65.9% for phone interfaces[39].

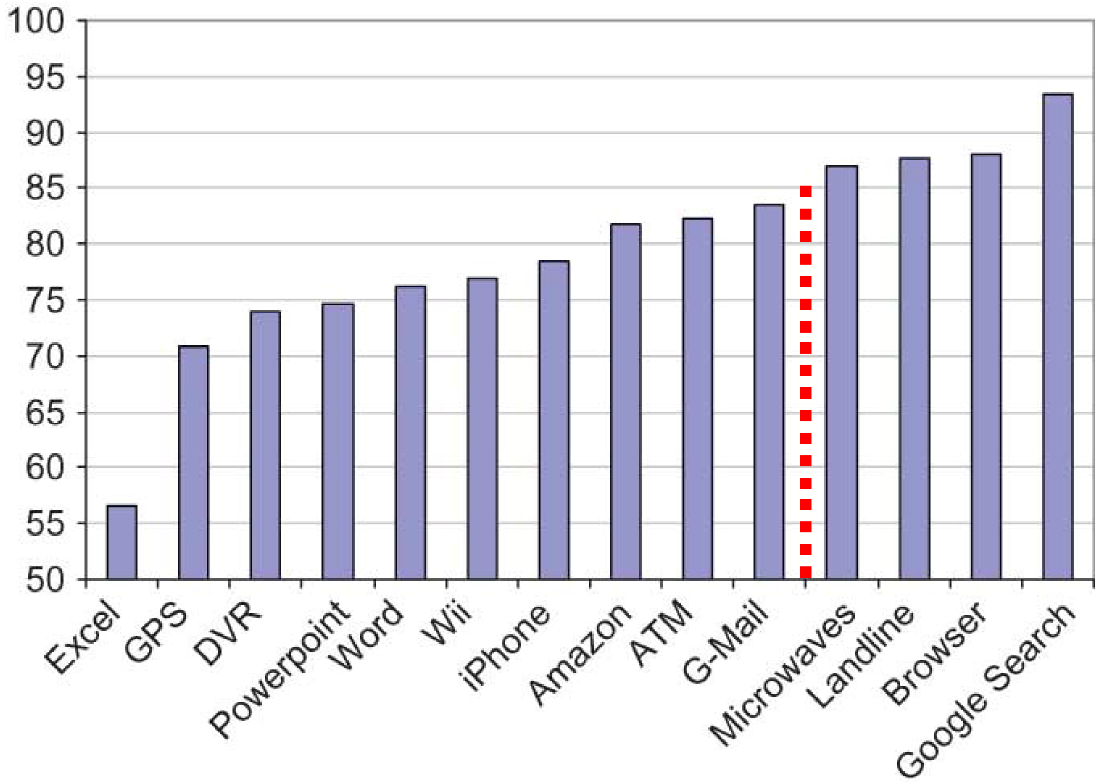


Figure 14.1: *Comparison of SUS score for several well known and widely used systems with our solution marked with a red dashed line.[40]*

In Figure 14.1 the SUS score of several well known systems can be seen as well as the score of our application which is marked by a red dashed line. The application plants itself among the top ranking systems which overall implies that the application was easy to use. Despite the high SUS score some issues related to SUS was encountered which in part can be contributed to the simplicity of the scale. The scale refers to "The system" which at times can be hard to define. Despite the interviewer explicitly stating that the system should be seen as the application on the phone it its current state and nothing else. Several users struggled with this and explained that they had a hard time not considering other aspects when ranking. It should however be noted that the overall usability of the application cannot be determined by a SUS score alone, it should be considered an indicator for future development.

The other part of the user tests focused on the visualisation of sound zones with the goal of evaluating the visualisation concepts and shapes designed for the sound zones. The participants were asked to rank the three different visualisation concepts as seen in Figure 12.5, namely the partially occluded, transparent and icon concepts. As with the visualisation concepts the participants were also asked to rank the difference sound zone shapes as seen in Figure 12.6.

It can clearly be seen that the participants favoured the partially occluded concept with minority preferring the transparent concept while only one participant ranked the icon concept as number one. The common reasoning for participants selecting the partially occluded concept was that the varying degree of openness in the model reflected the intended use of the sound zone well in their opinion. This could be point towards the users perceiving the hidden affordance of the sound zone based on the visualisation alone. The two participants who favoured the transparent concept noted that they found that particular concept more aesthetically pleasing than the others.

Only one participant ranked the icon concept as their number one saying that if people understand the icons then it would be the clearest way of showing the zone type. The other participants generally did not like the look of the icon concepts or did not believe that the icons could correctly convey the zone type. It should be noted that the icon concept was poorly executed compared to the two other concepts. The participants were informed of this, but it might be that their original impression left an impact while ranking the concepts. This can indicate that some users might prioritise the aesthetics more than the practicality of being able to clearly see the purpose of a zone type.

Figure 12.6 shows that a large majority of the participants preferred the dome shape over the other shapes. The main reason which most participants gave for ranking the dome highest was that it clearly showed them exactly how much floor space the sound zone took up. Compared to the three other shapes which were all floating in the air. This issue could however be solved by having the objects cast a shadow on the ground and it would then be possible to see the exact footprint of these objects. This could indicate that the participants are concerned about how much floor space a sound zone will occupy and that to them the sound zone visualisations affords a certain footprint based on the shape. By visualising the sound zones with AR the hidden affordance became perceivable. This is as augmented feedforward was introduced. The user is shown that a sound zone of a given shape affords a specific footprint. At the same time the user is also able to determine the exact location due to the introduction of augmented feedforward. By providing the user with the necessary information we facilitate interaction with the sound zones.

Only four shapes were created due to time limitations, however it could very well be that different shapes are better suited to represent a sound zone. One example of this could be a column which was actually requested by one of the test participants. The four shapes which were select all originated from the design sprint where the "crazy 8" method was used. This method forces the generation of several ideas in a short time-span. However it could be that a better approach, given enough time, would have been to include more shapes in the application and then have several stages of evaluation where the participants would evaluate which shapes best represent a sound zone.

14.3 Application development

This section will focus on issues encountered during the development phase and the prototype evaluation.

14.3.1 Application features

As described in Section 6.3 several features was designed which ideally would all have been implemented in the application. Out of the eight features five were implemented, the not implemented features were duplicating a sound zone, swapping the position of two sound zones. The last feature not implemented was having a sound zone being able to follow an user around when they move. It was decided that by using the available time on the primary features such as creating and moving sound zones. By doing this it was believed that a strong foundation could be built rather than incorporating all the features and having them work poorly.

14.3.2 Stability issues

During the prototype evaluation some stability issues were encountered when loading previously placed sound zones. The crashes were later identified to be caused by a bug encountered when users load previous zones while insufficient feature points have been gathered from the room. This causes the loaded zones to be placed at the "best match" which often will be near the users current position as this will be the area with the most feature points. If this is the case the loaded zones are placed directly on top of the user or floating in the air above them. If the users then try to place a new zone after the "failed loading" the application crashes, because then the maximum number of zones(3) exceeds the limit. It should be noted that the limit of three zones at a time was made purely to simplify the implementation stage. After a "failed" load the application thinks that no zones are placed, causing the application to follow improper logic when trying to place a new zone. This issue can be resolved in two ways: Solving the underlying issue with zones not being loaded, or handling the issue where the applications follows improper logic. In the case where the Cloud Anchors cannot be loaded correctly due to insufficient feature points instruction messages should be provided to guide the users on how to obtain the sufficient feature points such that the Cloud Anchors can be loaded correctly.

An issue encountered during an initial pilot test was the application using excessive processing power and overheating the phone causing the application to become unresponsive. This happened when two sound zone models were overlapping and the overlapping zone being coloured red to indicate the overlap as seen in Figure 7.10. Changing the colour of a renderable in Sceneform turned out to be rather complicated. Instead of turning the model red a new model was created with the default colour set to red. Every time two zones overlapped the model causing the overlap would be hidden and a model of the same shape but in red would be placed at the same position. A check was performed 30 times every second to see if any models overlapped to ensure real-time response if zones were moved into each other. However the frequency of this check and the processing required to swap the models caused performance issues. Lowering the frequency did not resolve the problem so rather than colouring the model red, a simple warning message was instead displayed on the screen if a zone was overlapping. The same can be said for the cases where the sound zones could not be saved correctly. Here it would also be beneficial to show instruction messages guiding the users on what to do if the zone cannot be saved.

14.4 General discussion

The selected development approach was to create a prototype solution based on a set of requirements. This prototype was then evaluated by conducting a series of user tests. This approach can be said to follow the waterfall development model where it might have been beneficial to utilise another development model instead. One such could have been agile development where the focus is on fast iterations and quickly producing new solutions based on the feedback gathered in the last iteration. This would consist of the same process of establishing requirements, designing, implementing and validation. Then this would be reviewed and based on the review the cycle would be repeated until the review process determines that no further iterations are needed. It can be said that some elements of agile development was incorporated as seen with the inclusion of the second development iteration. However the primary development approach would still be waterfall as the overall design did not change based on the review process.

During the design stage of the project, the GV Design Sprint[22] has been chosen as a method to generate concepts and the shapes for the sound zone visualisers. Though this is a viable method, because it stimulates the creative problem solving approach, the method has its limitations.

Our developer team consists of three people, which means that there is a limit to how much ideas could be developed according to the design sprint methodologies. By following the steps outlined in this method, we have yielded three concept types and four shapes for the sound zone visualisers. Instead of doing the design sprint, we could have tackled this goal from a different angle. For example, we could have created a larger number of concepts and shapes. What follows is the several stage screening process, where the created visualisations are presented to people in a form of surveys or questionnaires regarding which idea is best according to their preferences. Potentially, such an approach would result in final concepts to take form of an entirely different conceptualisation than the three concepts and four shapes.

Another theme which could have been investigated in this project is how the tests were conducted. Namely how the usability and visualisation tests were done together. A different approach could have been to segment this by conducting e.g. the usability test one day and the visualisation test later the same day or the day after. Alternatively the visualisation test could have been made into an online questionnaire and sent out to people. This could have given a larger amount of participants for the visualisation test and potentially introduced the participants to the sound zone concepts and different types of sound zones some time in advance before attending the usability test. Essentially giving them more time to grasp these abstract concepts which were introduced to them.

15 | Conclusion

This project was aiming to develop an application that allows users to interact with sound zones using augmented reality. The aim was to find an answer to the following problem statement:

How can a mobile AR application be developed the use case specific visualisations of sound zones?

A problem analysis was conducted which investigated the physical characteristics of sound zone technology. Chapter 2 found certain limitations of the current sound system such as minimal distance between sound zones and sound leakage both of which were taken into consideration when designing the application. Since this project focuses on soundscapes in the domestic environment it was also important to identify the different social scenarios that could occur in such an environment. A framework by Lundgaard et al.[6] which categorises the sound zones in the domestic environments was investigated. This framework was then used later in the design stage to define different sound zone types. During the problem analysis the underlying challenges with sound zone interaction were identified which concluded that sound zones are hard to interact with due to the lack of information conveyed to the users from the sound zones. The main challenge identified regarding sound zones was that they have a hidden affordance which is not perceivable by the user. This challenge can be solved by visualising the zones with the use of augmented reality. After deciding that the most efficient way to interact with sound zones would be through an augmented reality smartphone application it was important to investigate how users would be able to interact with AR itself. In order to find answers several state of the art augmented reality applications and systems were investigated and compared based on different criteria. It was found that using touch gestures would be a most viable solution in a smartphone application. Last but not least key touch gestures were identified that would serve an important part of the application design and implementation.

Based on the information from the problem analysis an ideal solution for the application was created which took into consideration the different use cases and all of the functionalities of the application. This description of the ideal solution was followed by a delimitation as certain functions could not be implemented due to several reasons such as time constraints and unavailable resources. It was concluded that the final scope of the development process would focus solely on an smart phone application which enables interaction with static sound zones through augmented reality. Furthermore the application would not be interfaced with the sound control system and it would also not be able to allow interaction with dynamic sound zones.

Based on the delimitation a set of prioritised requirements were derived which would serve as guidelines for the final application design. With these requirements in mind two design sprints were conducted by the authors. The first design sprint focused on creating concepts for the different shapes that a sound zone can take on when visualised with AR, resulting in four different shapes. The second sprint's goal was to output ideas for different visualisation concepts that convey the type of the given sound zone to other users around it.

The second design sprint resulted in three different concepts which along with the different shapes were surveyed at the user testing phase. During the implementation phase all of the outlined features from the design process were implemented successfully.

After a prototype was created the system was tested on 10 test participants as part of the user testing phase. The first part of the test focused on the usability aspects of the application. It can concluded that the application performed successfully based on the feedback received from the participants. The average SUS score for the developed application was 86.5%, which indicates that the solution can be considered a success in terms of usability. The second part of the testing was based on two surveys which asked the participants to rank the different visualisation concepts and the sound zone shapes in order of personal preference. The outcome of the user testing highlighted some of the key functionality drawbacks of the application and also resulted in a ranking for the visualisation concepts and the different shapes.

Based on the feedback from the user testing all the different comments and suggestion were sorted into categories and were also prioritised based on the number of times that specific comment appeared during the testing phase. Based on this a second iteration of implementation was conducted which focused on the implementation of the most frequent feedback into the application to improve its usability. It should be noted that the second iteration of implementation did not have a second iteration of user tests to validate the implementation.

In general it can be concluded that the previously set goals have been met with the exception of a sound zone following the user and the function of duplicating and swapping already existing sound zones. The main goal of visualising sound zones through augmented reality was successful and majority of users were able to grasp the idea why it is important to visualise sound zones. The application in its current state is designed in a way that integration of new modules and interfacing to the sound zone system is possible. The findings in this project indicate that the test participants think about sound zones in terms of their footprint, this information can be used in the future to design the sound zone visualisation according to how the users think about sound zones.

16 | Future Works

There are several possible improvements of the developed solution after meeting the minimum viable product criteria. Apart from the obvious system stability issues, these improvements can be split into two parts: the improvements related to the developed application and the improvements related to the implementation of the aspects of the ideal solution Chapter 6. With regard to the developed application there have been several delimitations considered while making the application as seen in Section 6.2. Several features of the application have not been implemented due to the time constraints associated with this project. These features are the dynamic sound zone scenario and the swap and duplicate sound zones. The dynamic sound zone scenario would require an implementation of a tracking system. In order for the sound zone to be able to follow a person, the person's position needs to be known. The tracking could be implemented by using the motion tracking functionality of ARCore [23]. This should ideally allow the application to know the position of the phone(user) in relation to any AR objects in a room.

Furthermore, the following zone automatic behaviour needs to be designed and programmed in a way such that the users' location is known to the system. This is needed to define the zone behaviour when the user leaves the room boundaries, as in what would happen with the following sound zone location & media broadcast when its owner is not in the room. Possibly, the following sound zone can exist only when the user is inside the room and it is destroyed and recreated based on user presence in the room. In order to notify the users about such sound zone behaviour, they can be notified with a warning message when leaving the boundaries of the room.

Alternatively this would not be a problem if the sound zone application supports a multi room setup where it is possible to add different rooms to the application. Then either the dynamic sound zone would stop following would be contained to one room, or potentially allowed to move it self to the new room. In the scenario where the user leaves is the house or enters a non registered room the zone should would then be left in the room last visited by the user. The sound zone swap and duplicate options can be implemented in order to provide an additional degree of interaction while creating the sound zones. The 3D models used for developing the prototype solution can be improved as well. One obvious addition is developing an application which can also be used on IOS devices. The sound zone renders can be enhanced such that they are reflective to the lighting conditions of the room which would result in a more seamless AR experience. The textures of the renders can be enhanced as well, either by providing a customisation option to the user to colour their sound zone as they like or creating a uniform more aesthetic texture pack for the renders. Furthermore, the sound zone renders can be animated in order to provide an extra degree of sound zone affordance. The animation can illustrate how the sound propagates inside the sound zone and it might provide a better comprehension about the nature of the sound zones to the user. There is a room for various customisation options to be implemented in the application. For example, arguably for some end users it would be desirable to be able to customise the appearance of the AR renders.

This could be accomplished via providing an option to design or create your own 3D model for the sound zone render.

The non-implemented modules of the system architecture as seen in Figure 8.1 need to be implemented to realise the vision of the ideal solution Chapter 6. These modules are the WI-FI speaker module, tracking module and the sound zone generation system. The inclusion of these modules would be required users to control sound zones through the application. As it is now the users only control the visualisations, and the actual sound zone control would have to be instructed to update according to the actions in the application.

The addition of the Wi-Fi speaker module can be implemented through a framework similar to Spotify[41], where the sound zones would work like the Wi-Fi speaker. The user would be able to find the sound zone as the playback device when playing music in e.g. Spotify. The sound zone would appear under the devices under the name that the user assigned to the specific sound zone.

The application should communicate with the sound zone generating system such that it can instruct this system about creation of sound zones. This would require specific instruction about the exact size of the zone and also where the sound zone should be generated in the room. The instructions sent to the sound zone generation system would then configure the soundbar such that the desired sound zone(s) are created.

One issue which could be encountered when placing zones is if zones are placed so they interfere with the creation of other zones. This could include creation of zone right in front of each other or placing a sound zone right in front of the sound bar. To prevent this the application could inform the user about suitable placement locations by changing the dots seen when placing objects to red if placement at that location is not possible. This would also apply if the user tries to place the zones in less than ideal locations such as in a corner. This is however already mostly dealt with by ARCore which already factors in the features of the room when placing the zone. Therefore if enough features are gathered ARCore will automatically "push" the zone a bit back such that it does not pass through the walls. In the cases where ARCore cannot handle the issue the system should use the same approach as mentioned above where it indicates that the picked location is not suitable for placement of a sound zone.

The proposed solution can be applied to other use cases outside the domestic household. Potentially, the system can be adapted to other areas such as work environment with offices, hospitals, or concert halls. The variables which would change are the established sound zone types (private, mixed and public) and 3D models as those have been developed with the domestic environment in mind. So that to adapt our solution to the new environment, such as offices, there would need to be done a similar investigation into the potential use cases for the sound zones in the given environment.

Stepping beyond the scope and goals for this project, there exist a variety of possible expansion side projects related to the sound zone interaction. for example, there is a possibility of creating a multi-modal interface for the system. One of such expansions could be a gesture based interaction with the sound zones. Though it might seem as a more complex way of interaction with sound zones, it could add an option for an alternative mode of control over the sound zones if the user would prefer it. This modality can be implemented via computer vision methods of gesture recognition, where the detected image would be a certain gesture performed by the user. Another potential modality could be voice control. The latter can be implemented similarly to the Google Home[42] smart speaker system where voice commands could then be used to interact with sound zones. These modalities can be combined together, where the user might point a location where he wants to place a sound zone and confirm the command with voice.

The true potential of the developed application would shine if the AR technologies would be more integrated into the everyday life of a regular person. This project has been developed with the purpose of setting up the sound zone, however its impact would appear more prominent if the application became ingrained in everyday usage, similarly to how there is an AR visualisation component in Google Maps[14]. One potential use case is that the sound zone application is integration with Google Home and the AR visualisation application is used for the setup. Then as explained above most tasks would be handled using voice commands and the application would serve as a advanced configuration to be used when it is beneficial to visualise the zones. One such scenario would be changing the current room setup where the position of each zone might change.

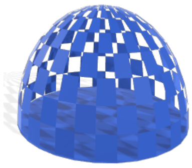
Part VI

Appendix

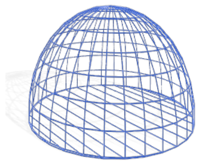
A Sound Zone 3D Models

A.1 Partially covered

(a) *Public dome*

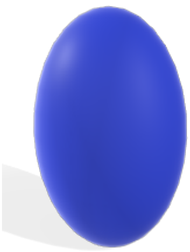


(b) *Mixed dome*

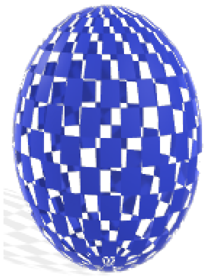


(c) *Private dome*

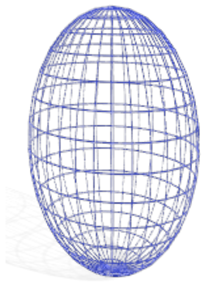
Figure A.1: *Public, mixed and private sound zones using the dome shape.*



(a) *Public egg*



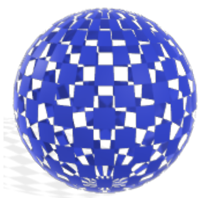
(b) *Mixed egg*



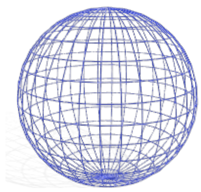
(c) *Private egg*

Figure A.2: *Public, mixed and private sound zones using the egg shape.*

(a) *Public orb*

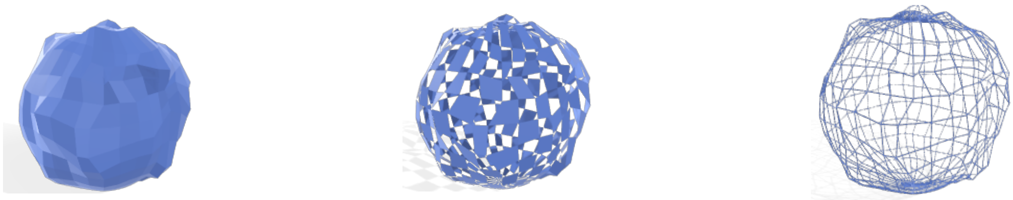


(b) *Mixed orb*



(c) *Private orb*

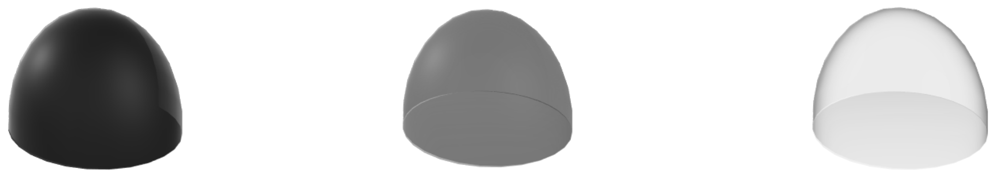
Figure A.3: *Public, mixed and private sound zones using the orb shape.*



(a) *Public abstract entity* **(b)** *Mixed abstract entity* **(c)** *Private abstract entity*

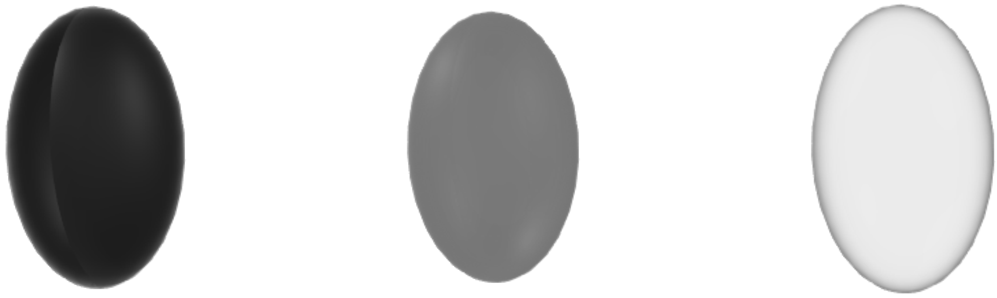
Figure A.4: *Public, mixed and private sound zones using the abstract entity shape.*

A.2 Transparent



(a) *Public dome* **(b)** *Mixed dome* **(c)** *Private dome*

Figure A.5: *Public, mixed and private sound zones using the dome shape.*

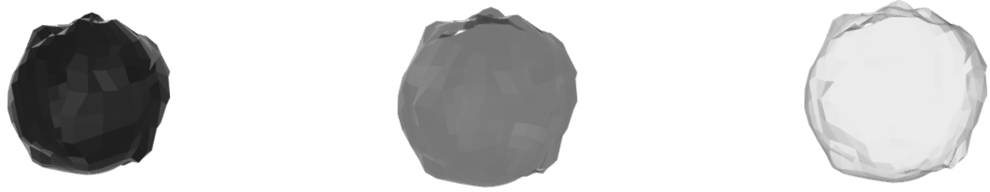


(a) *Public egg* **(b)** *Mixed egg* **(c)** *Private egg*

Figure A.6: *Public, mixed and private sound zones using the egg shape.*

(a) *Public orb* **(b)** *Mixed orb* **(c)** *Private orb*

Figure A.7: *Public, mixed and private sound zones using the orb shape.*



(a) *Public abstract entity* **(b)** *Mixed abstract entity* **(c)** *Private abstract entity*

Figure A.8: *Public, mixed and private sound zones using the abstract entity shape.*

A.3 Icons

(a) *Public dome* **(b)** *Mixed dome* **(c)** *Private dome*

Figure A.9: *Public, mixed and private sound zones using the dome shape.*

(a) *Public egg* **(b)** *Mixed egg* **(c)** *Private egg*

Figure A.10: *Public, mixed and private sound zones using the egg shape.*

(a) *Public orb* **(b)** *Mixed orb* **(c)** *Private orb*

Figure A.11: *Public, mixed and private sound zones using the orb shape.*

(a) *Public abstract entity* **(b)** *Mixed abstract entity* **(c)** *Private abstract entity*

Figure A.12: *Public, mixed and private sound zones using the abstract entity shape.*

Bibliography

[1] Michael Bull. *Investigating the Culture of Mobile Listening: From Walkman to iPod*, pages 131–149. Springer Netherlands, Dordrecht, 2006. ISBN 978-1-4020-4097-9. doi: 10.1007/1-4020-4097-0_7. URL `https://doi.org/10.1007/1-4020-4097-0_7`.

[2] Stine S. Lundgaard, Peter Nielsen, and Jesper Kjeldskov. Designing for domestic sound zone interaction. *Personal and Ubiquitous Computing*, 03 2020. doi: 10.1007/s00779-020-01387-2.

[3] Taewoong Lee, Jesper Nielsen, and Mads Christensen. Towards perceptually optimized sound zones: A proof-of-concept study. pages 136–140, 04 2019. doi: 10.1109/ICASSP.2019.8682902.

[4] Martin Bo Møller. *Sound Zone Control inside Spatially Confined Regions in Acoustic Enclosures*. PhD thesis, Aalborg Universitet, 2019. PhD supervisor: Prof. Jan Østergaard, Aalborg University Assistant PhD supervisor: Assoc. Prof. Efren Fernandez-Grande, Technical University of Denmark Industrial PhD Supervisor: Jesper Kjær Nielsen, Bang & Olufsen a/s Former PhD Supervisor: Assoc. Prof. Søren Krarup Olesen, Aalborg University Former Industrial PhD Supervisor: Jan Abildgaard Pedersen, formerly at Bang & Olufsen a/s.

[5] Terence Betlehem, Wen Zhang, Mark A. Poletti, and Thushara D. Abhayapala. Personal sound zones: Delivering interface-free audio to multiple listeners. *IEEE Signal Processing Magazine*, 32(2):81–91, 2015. doi: 10.1109/MSP.2014.2360707.

[6] Stine S. Lundgaard and Peter Axel Nielsen. Personalised soundscapes in homes. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, page 813–822, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450358507. URL `https://doi.org/10.1145/3322276.3322364`.

[7] Stine S. Lundgaard, Peter Axel Nielsen, and Jesper Kjeldskov. Interaction design for domestic sound zones. In *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, AM'19, page 248–251, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450372978. URL `https://doi.org/10.1145/3356590.3356630`.

[8] B. D. Van Veen and K. M. Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24, 1988. doi: 10.1109/53.665.

[9] Philip Coleman. *Loudspeaker Array Processing For Personal Sound Zone Reproduction.* PhD thesis, University of Surrey (United Kingdom), 2014. URL `https://epubs.surrey.ac.uk/854944/`.

[10] Stephan Wensveen, J. Djajadiningrat, and Kees Overbeeke. Interaction frogger: A design framework to couple action and function through feedback and feedforward. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, volume 177-184, pages 177–184, 01 2004. doi: 10.1145/1013115.1013140.

[11] Donald Norman. *The Design of Everyday Things*. Basic Books, September 2002.

[12] William W. Gaver. Technology affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, page 79–84, New York, NY, USA, 1991. Association for Computing Machinery. ISBN 0897913833. doi: 10.1145/108844.108856. URL `https://doi.org/10.1145/108844.108856`.

[13] Rex Hartson. Cognitive, physical, sensory and functional affordances in interaction design. behaviour & information technology. *Behaviour & IT*, 22:315–338, 09 2003. doi: 10.1080/01449290310001592587.

[14] Google. Ar/vr. URL `https://arvr.google.com/ar/`.

[15] Google Developers. Arcore overview, 2021. URL `https://developers.google.com/ar/discover`.

[16] IKEA. Ikea place ios application. URL `https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050`.

[17] Tobias Höllerer Wendy H. Chun. Real-time hand interaction for augmented reality on mobile phones. In *IUI '13: Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 307–314, 2013. doi: https://doi.org/10.1145/2449396.2449435.

[18] Hilliges et al. User interaction in augmented reality, 2012. URL `https://patentimages.storage.googleapis.com/43/27/d3/92e70d7eb77a70/US20120113223A1.pdf`.

[19] S. Lin, H. F. Cheng, W. Li, Z. Huang, P. Hui, and C. Peylo. Ubii: Physical world interaction through augmented reality. *IEEE Transactions on Mobile Computing*, 16(3): 872–885, 2017. doi: 10.1109/TMC.2016.2567378.

[20] Mark Billinghurst, Thammathip Piumsomboon, and Huidong Bai. Hands in space: Gesture interaction with augmented-reality interfaces. *Computer Graphics and Applications, IEEE*, 34:77–80, 05 2014. doi: 10.1109/MCG.2014.8.

[21] Google. Material design, 2014. URL `https://material.io/`.

[22] GV. The design sprint, 2019. URL `https://www.gv.com/sprint/`.

[23] Google Developers. Fundamental concepts, 2021. URL `https://developers.google.com/ar/discover/concepts`.

[24] Google Developers. Cloud anchors overview for android, 2021. URL `https://developers.google.com/ar/develop/java/cloud-anchors/overview-android`.

[25] Google Developers. Meet android studio, 2021. URL `https://developer.android.com/studio/intro`.

[26] Google Developers. Introduction to activities, 2019. URL `https://developer.android.com/guide/components/activities/intro-activities`.

[27] Unity Technologies. Unity platform, 2021. URL `https://unity.com/products/unity-platform`.

[28] Google Developers. Sceneform overview, 2020. URL https://developers.google.com/sceneform/develop.

[29] Unity Technologies. About ar foundation, 2021. URL https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html.

[30] Google. Buttons: floating action button, 2021. URL https://material.io/components/buttons-floating-action-button.

[31] PT Inc. Spatial frame of reference, 2021. URL https://library.vuforia.com/articles/Solution/spatial-frame-of-reference.html.

[32] Thomas Gorisse. Maintained sceneform sdk for android, 2021. URL https://github.com/thomasgorisse/sceneform-android-sdk.

[33] Grobe Susan Fonteyn Marsha, Kuipers Benjamin. A description of think aloud method and protocol analysis. *Qualitative Health Research - QUAL HEALTH RES*, 3:430–441, 11 1993. doi: 10.1177/104973239300300403.

[34] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems*, 1984. URL https://dl.acm.org/doi/10.1145/357417.357420.

[35] JBL. Jbl soundgear wearable ear-free wireless speaker, 2020. URL https://mm.jbl.com/bluetooth-speakers/JBL+SOUNDGEAR.html.

[36] Ph.D. Bob Bailey. Determining the correct number of usability test participants, 2006. URL https://www.usability.gov/get-involved/blog/2006/09/correct-number-of-test-participants.html.

[37] John Brooke. Sus - a quick and dirty usability scale, 1995. URL https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale.

[38] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 206–213, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915755. doi: 10.1145/169059.169166. URL https://doi.org/10.1145/169059.169166.

[39] Aaron Bangor, Phil Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Stud.*, 4:114–123, 04 2009.

[40] Philip T. Kortum and Aaron Bangor. Usability ratings for everyday products measured with the system usability scale. *International Journal of Human–Computer Interaction*, 29 (2):67–76, 2013. doi: 10.1080/10447318.2012.681221. URL https://doi.org/10.1080/10447318.2012.681221.

[41] Spotify AB. Spotify and sonos, 2020. URL https://support.sonos.com/s/article/1153?language=en_US#:~:text=You%20can%20control%20Sonos%20directly,your%20Sonos%20speaker%20or%20group.

[42] Google. Google home mini, 2021. URL https://store.google.com/us/product/google_home_mini_first_gen?hl=en-US.