

# Prototyping Tool for Large Scale Interactive Experiences

Venkata Satya Anurag Gargeya Unnava and Yiyang Liu

DEPARTMENT OF DESIGN SCIENCES  
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY  
2023

MASTER THESIS



# Prototyping Tool for Large Scale Interactive Experiences

A Study on the design, development and usability of a  
prototyping tool for large scale interactive experiences.

Venkata Satya Anurag Gargeya Unnava  
ve5660un-s@student.lu.se

Yiyang Liu  
yi7463li-s@student.lu.se

**SONY**<sup>®</sup>



**LUND**  
UNIVERSITY

# Prototyping Tool for Large Scale Interactive Experiences.

A Study on the design, development and usability of a prototyping tool for large scale interactive experiences.

Copyright © 2023 Venkata Satya Anurag Gargeya Unnava and Yiyang Liu

*Published by*

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

Subject: Virtual Reality and Augmented Reality (MAMM15)

Division: Department of Design Sciences

Supervisor: Günter Alce

Co-supervisor: Johan Hansson, Moe Takamura

Examiner: Mattias Wallergård

# Abstract

In the age where entertainment is the driving force behind the functioning of large industries, companies like SONY have figured out a way to market their entertainment content including movies, music, and games using large-scale interactive marketing and promotional content. Building and testing such content before it is released on large and complex devices is a continuously challenging task and there have been efforts both in academia and industry made to ameliorate this content production workflow.

The thesis aimed to address the common problem associated with large interactive content and propose a novel solution with virtual reality. This thesis thoroughly discussed a use case of a visualizer that lets the designers, developers, and stakeholder experience such large-scale content as a simulation both on a desktop computer and in virtual reality. We had an in-depth discussion about the design and development cycle of a visualizer and talked about the methods and technology used to build it. The thesis also described in detail the evaluation of the first iteration of the product. Although there were some limitations to the study, we concluded that virtual prototyping is indeed a good way of tackling design problems for large interactive experiences.

# Popular Science Summary

Computer simulations are digital reconstructions of the real world with a specific focus on the required parameters. The simulated environment provides a play area for developers and testers to perform testing and check if the product behaves in the expected way. Among many forms of simulation, visualized representation is the most looked upon one. Visualizations are digital methods that present and communicate information visually. Our vision has the largest "bandwidth" for information, therefore, is the most useful sense in product development and testing. The activity of using simple yet effective representations and simulations to assist the development and testing of a product is referred to as prototyping. Prototyping can take many forms from comic drawings on paper to almost finished products; the form that we are interested in is prototyping using computer simulations.

Virtual Reality or VR, is a technology that aims to provide increasingly more "real" senses in the digital space and ultimately to replace our own interface with the real world, could serve as a technology that provides increasingly better simulations. Therefore, VR is a promising technology for digital prototyping methods. We believed that the current state of VR technology is ready for it. The use case we chose to dive in, is the simulation of large-scale interactive content which played a vital role as an attention grabber in the growing entertainment market. These contents are designed to be attractive and interactive, but often very limited because of the lack of testing. We aimed to change this situation by adapting VR to our work.

The challenging part is the interactions surrounding VR. As a novel technology, we had to carefully examine the interactions between us and VR. For example, are the user behaviors required in the VR experience natural and intuitive? We wanted to investigate if prototyping in VR is an effective way.

# Acknowledgements

This Master thesis was only made possible with the help and efforts of more than a few people whom we would like to extend our deepest gratitude.

Beginning with Johan Hansson, Moe Takamura, and the team at SONY Nordic, we thank you for hosting our thesis and making us feel welcome in the team. Thank you for helping us in every stage of the thesis, from design to development to user testing, and for all the resources, design inputs, and helpful feedback that enabled us in creating this small piece of software. We hope that it will help the team test the Interactive Content Solution experiences and expedite the development process.

Following this, we would like to express our gratitude to our supervisor Dr. Günter Alce, whose support and positive attitude were a big reason why we could complete this enormous undertaking. We would like to thank him for his input through out the thesis, especially for the guidance through the user testing which helped us come up with our finest work.

Additionally, we would like to thank Johan Herrstrom from the SONY team for helping us out with the level design and setup.

We would also like to thank everyone who participated in the user test sessions and provided us with valuable feedback. We appreciate your feedback very much and it would help us refine the product further and make it more usable.

Lastly, we would like to thank our families and friends who were great support during this project and kept us motivated throughout.

*Lund,  
Anurag Unnava, Yiyang Liu*

# Acronyms and Abbreviations

- **VR**-Virtual Reality
- **AR**-Augmented Reality
- **XR**-Extended Reality
- **MR**-Mixed Reality
- **Hi-fi**-High Fidelity Prototype
- **Lo-fi**-Fidelity Prototype
- **RQ**-Research Question
- **SUS**-System Usability Scale
- **UCD**-User Centered Design
- **UI**-User Interface
- **UX**-User Experience
- **TLX**-Task Load Index
- **LID**-Large Interactive Display
- **MVP**-Minimum Viable Product
- **ICS**-Interactive Content Solution

# Contents

<b>Abstract</b>	<b>I</b>
<b>Acknowledgements</b>	<b>III</b>
<b>Acronyms and Abbreviations</b>	<b>IV</b>
<b>Table of Contents</b>	<b>VII</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 SONY . . . . .	1
1.2.1 Interactive Content Solution (ICS) . . . . .	2
1.3 Purpose and Goals . . . . .	3
1.3.1 Goals . . . . .	3
1.3.2 Research Questions . . . . .	3
1.4 Global Goals . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Prototyping With Virtual Reality . . . . .	5
2.2 Large Interactive Displays . . . . .	5
<b>3 Method and Theory</b>	<b>7</b>
3.1 Virtual Reality . . . . .	7
3.2 Interaction Design and Activities . . . . .	7
3.2.1 Identifying Needs and Establishing Requirements . . . . .	7
3.2.2 Developing Alternative Strategies . . . . .	8
3.2.3 Building interactive versions of the designs . . . . .	8
3.2.4 Evaluating . . . . .	8
3.2.5 Interface metaphor . . . . .	8
3.3 Design Guidelines . . . . .	9
3.3.1 User Centered Design . . . . .	9
3.3.2 Norman's Seven Fundamental Design Principles . . . . .	9
3.4 User Study . . . . .	10
3.4.1 Who are the users? . . . . .	10
3.4.2 Data-gathering . . . . .	10
3.5 Usability and User Experience . . . . .	11
3.6 Brainstorming . . . . .	12
3.7 Conceptual Design . . . . .	12
3.7.1 Personas . . . . .	12
3.7.2 Storyboard . . . . .	13



3.7.3	User Flow . . . . .	13
3.8	Prototyping . . . . .	13
3.8.1	Low Fidelity Prototypes . . . . .	14
3.8.2	High Fidelity Prototypes . . . . .	14
3.9	Evaluation . . . . .	14
3.9.1	Usability testing . . . . .	14
3.9.2	Questionnaire . . . . .	15
<b>4</b>	<b>Design phase</b>	<b>17</b>
4.1	Identifying Needs . . . . .	17
4.1.1	Literature study . . . . .	17
4.1.2	Interviews with user expert . . . . .	18
4.1.3	User personas . . . . .	19
4.2	Establish Requirements . . . . .	22
4.3	Conceptual Design . . . . .	24
4.3.1	Brainstorming session . . . . .	25
4.3.2	Storyboard . . . . .	25
4.3.3	Feedback from the first iteration . . . . .	25
4.4	Low Fidelity Prototypes . . . . .	27
4.4.1	Important design choices . . . . .	27
4.4.2	First Prototype . . . . .	28
4.4.3	Second Prototype . . . . .	30
<b>5</b>	<b>Development Phase</b>	<b>31</b>
5.1	Introduction . . . . .	31
5.2	Resources and SDKs . . . . .	31
5.2.1	Unity3D and Visual Studio . . . . .	31
5.2.2	Oculus XR Plugin and Oculus Air Link . . . . .	32
5.3	Architecture . . . . .	32
5.4	Controls and Interaction . . . . .	33
5.4.1	Mouse and Keyboard . . . . .	33
5.4.2	Virtual Reality . . . . .	33
<b>6</b>	<b>Testing Phase</b>	<b>36</b>
6.1	Introduction . . . . .	36
6.2	Why Test? . . . . .	36
6.3	What to test for? . . . . .	37
6.4	Test plan . . . . .	37
6.4.1	Participants . . . . .	37
6.4.2	Resources . . . . .	38
6.4.3	Tasks during testing . . . . .	38
6.4.4	Registration and Pre-screening Questions . . . . .	38
6.5	Test sessions . . . . .	38
6.5.1	Moderator and observer role . . . . .	39
6.5.2	Briefing . . . . .	40
6.5.3	Testing . . . . .	40
6.5.4	Debriefing . . . . .	41
6.6	Results . . . . .	41
6.6.1	Design and Development Phase . . . . .	41

6.6.2	User Testing . . . . .	41
6.6.3	Method of Analysis . . . . .	42
6.6.4	Evaluation . . . . .	42
<b>7</b>	<b>Discussion</b>	<b>46</b>
7.1	Reflection on Research Findings . . . . .	46
7.1.1	Hints and Instructions . . . . .	46
7.1.2	Free-fly Camera . . . . .	46
7.2	Limitations and Delimitations . . . . .	47
7.2.1	Explorative User Study At Earlier Stages . . . . .	47
7.2.2	Frame Drops Due to Connection . . . . .	47
7.2.3	Limited Participant In Testing . . . . .	47
7.3	Future Directions . . . . .	47
7.3.1	More Intuitive Interaction . . . . .	48
7.3.2	Custom Screens . . . . .	48
7.3.3	Customizable Environment . . . . .	48
7.3.4	Enclosed Environment . . . . .	48
<b>8</b>	<b>Conclusion</b>	<b>49</b>
	<b>Bibliography</b>	<b>50</b>
<b>A</b>	<b>Appendix A</b>	<b>53</b>
A.1	Tables, Scenario and NASA-TLX . . . . .	53

# 1 Introduction

*This chapter introduced the thesis topic and gave a glimpse of the Interactive Content Solution ecosystem in which the product that is part of this thesis was built.*

## 1.1 Background

A simulation is the imitation of the operation of a real-world process or system over time[1]. Simulators are devices or software or sometimes both that allow the user to experience these simulations. Visualization is any technique for creating images, diagrams, or animations to communicate a message[2]. Visualizers are tools that enable the user to form an understanding of the data from the said simulation. Simulators and visualizers, both desktop and Virtual Reality (VR) based are great enablers for developers, stakeholders, and customers to be able to test, understand and enjoy the products which otherwise would be very time and resource-consuming to build in the real world. This allows the user to interact with the product in the simulator well before it is brought to life in the real world which for a developer is helpful as it allows him to perform rapid testing. For a designer, it helps enhance usability and helps catch design flaws well in advance. For the stakeholders, it gives them an understanding of the progress of the product and allows them to relay course correction instructions to the development team. In some cases, the simulator could also enable collaboration thus enhancing communication and decision making for all the parties involved in the product's life cycle.

VR simulations and prototyping has found a large audience in numerous fields including product design and development, training and simulations, real estate, architecture, education, marketing, healthcare, research prototyping, etc, and has been greatly helping out stakeholders and developers alike[3]. This encouraged us to propose the inclusion of VR in the visualizer that was built during this thesis. The foundation of this project is a standalone interactive content editor about which we discuss below.

## 1.2 SONY

SONY[4] is an Entertainment and technology company headquartered in Japan with an office operating out of Lund, Sweden. The team here is responsible for building large-scale interactive content using game engines usually for promotion and marketing of the entertainment content (movies, games, etc.) produced by SONY.

One of many products built by this team is the Interactive Content Solution which is discussed below.

## 1.2.1 Interactive Content Solution (ICS)

Interactive Content Solution or ICS is a suite of applications developed by SONY that allow the user to design, create, manage, and run large-scale interactive content. This suite includes an editor application to design and create the content, a cloud application to manage and schedule the content, and an agent application to run the content.

### ICS Editor

The ICS Editor is a content designer and builder application developed using Unity3D [5]. It allows the user to design and build touch and motion interactive content. It houses all the tools required to design and create content including image editing tools, video and audio managers that handle playing the respective contents, and the touch and motion managers that enable interaction. Using ICS editor, the user can create an application through a graphical user interface[6] to help potential clients of the solution make and edit large-scale interactive experiences easily. The experiences so built can be packaged and exported as executable files which can be used to run the content on any Windows computer.

The interaction is enabled by a suite of LIDAR sensors for touch interaction and camera setups for skeleton tracking that enabled motion interaction. To display the content, as this content is usually large (sometimes 20x10 feet), projectors and large LED screens are used. The intended end user of this editor is the building management team, invited artists, or volunteers. While it is fairly easy to use in terms of providing an easy editor to create visual effects and interactions, one particular issue to be noted is: since a significant number of users do not have extensive experience working with 3D geometry, it is very likely that they lose the sense of scale in their final work. Hence, we developed a visualization tool both for desktop and VR that can help solve this issue. This will even be beneficial for experienced users to see their work in one to one scale and spot potential design flaws.

### ICS Visualizer

The ICS Visualizer is the solution proposed to the above-mentioned problem of not being able to visualize content built in the ICS editor at scale for users with and without working knowledge of 3D geometry alike. It creates a game-like play area for the user to load the content and run it to experience it in real-time. It allows the user to interact with the content using a mouse, keyboard, and a 3D human character(avatar). The user can control the avatar, make it walk around, and interact with the content using mouse and keyboard inputs. The application is also connected to a VR headset that enables the user to interact with the content in first person and see it work at one to one scale.

## 1.3 Purpose and Goals

The purpose of this thesis was to define, design, and develop a desktop and VR simulator for large-scale experiences built using the ICS Editor. The aim also extended to testing the application built on the grounds of usability, consistency, and functionality as well as gathering feedback from regular users, developers, and designers, to refine the application in future iterations.

This purpose boiled down to the following goals which enabled us to divide the thesis development process into sections and create user stories, features, and tasks. This way we could keep track of what phase were we in and also allowed us to divide work among ourselves.

### 1.3.1 Goals

- Define and design a workflow for the simulator.
- Design the UI, and create wireframes, and sprites that would be used to create the actual application.
- Develop the simulator using Unity3D.
  - Build the functioning UI.
  - Build the desktop simulator.
  - Build the VR extension for the simulator.
  - Conduct manual testing simultaneously.
- Design the user tests
  - Create a scenario.
  - Create the post-test questionnaire.
  - Create consent forms.
  - Handle miscellaneous logistics.
- Conduct the user testing.
- Record and report the findings and conclusions.

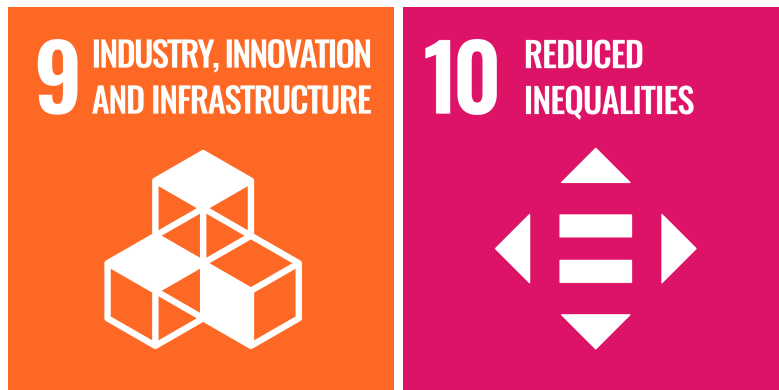
### 1.3.2 Research Questions

By conducting this thesis, we answer the following research questions.

- How to best apply VR in the development of large-scale interactive content?
- What impact does VR have on the design and development of such interactive content?

## 1.4 Global Goals

This thesis aligned with Industry, Innovation, and Infrastructure as well as the Reduced Inequalities goals in the United Nations Sustainable Development Goals [7]. More specifically, we helped to promote more inclusive industrialization and foster innovations through empowerment, and to reduce the gap between people with sophisticated training and people who did not have access to that kind of training. Our goal for this project was to give less technologically sophisticated users a way of previewing and thereby having some understanding of the often larger-scale interactive content that they are developing themselves.



**Figure 1.1:** United Nations Sustainable Development Goals nine and ten [8]

## 2 Literature Review

*This part presents our literature research and the search method we applied. The literature research was done early in the exploration phase where we were presented with an interactive hallway and Interactive Content Solution(ICS) used to develop it.*

### 2.1 Prototyping With Virtual Reality

The first topic of interest was prototyping with VR, a research topic inspired by past research by our supervisor Gunter Alce. We then continued the search in Lund University's library resources(*LUBsearch*[9]). In searching for information of interest, we used keywords *Prototype* and *Virtual Reality* together in the search. In addition to *LUBsearch*, *Google Scholar*[10] was also used during the search with the same keywords. From the literature, we could see extensive research has been conducted on utilizing virtual reality as a prototyping technique. It was safe to say that prototyping with VR is a much more sophisticated technique than before.

Zhang et al.[11] compared several different prototyping methods including using VR in a product design setting. The study went on to conclude that in a more static setting, VR had a limited superiority over computer prototyping but cannot replace paper prototyping because it inevitably lacks richness and complexity of reality according to Bishop and Rohrmann[12]. However, this conclusion applied more to a more static product with limited interaction. For more interactive systems such as smart homes and IoT devices, the capabilities of VR were more deeply appreciated. In his study on prototyping to simulate AR interaction, Alce et al.[13] argued that handling the extensive functionalities of an interactive device was more stable and cost-effective than relying on a well-trained human agent otherwise known as the wizard in The Wizard of Oz method. More importantly, VR had predominantly been used as a visualization tool for design proposals rather than actual interactable prototypes. Alce et al.[14] thought that the latest generation of commercial VR has changed this situation for that it allowed users to work with virtual environments in an easier manner. In their work on prototyping AR and IoT devices with VR [13], we have learned that it was really important to remember the discrepancy between real and virtual and that we were testing to see the strengths and weaknesses of the virtual interaction that will certainly have some imperfect transfer to reality.

### 2.2 Large Interactive Displays

The second topic of interest we explored was large interactive displays(LID). We used keywords *Large interactive display* on *Google Scholar*[10] along with *ubiquit-*

*ous computing.* From the literature, we learned that LIDs can be of great social value: promoting community awareness, sharing community history, welcoming visitors and newcomers, etc[15]. However, despite their near-ubiquitous presence in our everyday environments, the consensus amongst pervasive display researchers was that deployments are largely undervalued by viewers[16]. The phenomenon is described as *display blindness* and *interaction blindness* in design research. Researchers have tried numerous ways of overcoming this problem, for example, Du et al.[17] proposed to use of video and animation would reduce this phenomenon. Meanwhile, Parker et al.[18] suggested that focusing on people, location, and community is essential to the design of a successful LID that provides value to its user. Most research focused on the design recommendations for LIDs, but little has been discussed about the actual development of such experiences. We believed that with better prototyping techniques, such as VR prototyping, designers of LIDs would be able to better employ the recommendations above by seeing and feeling the experience designed.



# 3 Method and Theory

*This chapter presented the essential theories used in the thesis, including theories about interaction design, universal design, and other design guidelines.*

## 3.1 Virtual Reality

Wallergard et al. [19] defined virtual reality (VR) as "a technology that strives to credibly imitate and replace real sensory stimuli while allowing interactions to be performed as realistically as possible." Based on this definition, we could say that regardless of the different paradigms of VR, from a head-mounted display (HMD) to a cave VR system, VR provides an enclosed interaction loop between humans and computers. In the scope of this thesis, we would use VR to simulate the environment of other interactive experiences.

## 3.2 Interaction Design and Activities

Interaction design concerns how products support the way of communication and interaction in people's everyday as well as working lives [20]. The main goal of the design is to convey to users when, where, why, and how they should interact with certain products [21]. There were four main activities that constitute the design process of interaction design: *establish requirements, designing alternatives, prototyping, and evaluating* [20]. The activities are intended to inform one another [20] and should make up an iterative process with the user in its primal focus and a clear usability goal to aim for.

### 3.2.1 Identifying Needs and Establishing Requirements

Establishing requirements is a fundamental part of the entire design process [20]. This activity specifies the purpose and intended functionalities of the product as well as user goals to be achieved by the end product. In order to establish the requirements, one must identify and analyze user needs. User data can be gathered using both qualitative and quantitative methods; these data are either empirical or analytical, formative or summative data. The data is then analyzed to inform user needs, and furthermore, our requirements. Don Norman suggests we first ask "*What do I want to accomplish?*" in each stage of the design process [21]. This clearly illustrates the importance of identifying needs and establishing requirements. Sharp, Rogers, and Preece [20] also stressed the importance of an agreed-upon and documented goal at the beginning of a good design.

### **3.2.2 Developing Alternative Strategies**

Developing alternative strategies is core to designing. There can be two sub-activities in this activity: conceptual design and physical design[20]. This activity produces both conceptual models that describe what the product does, behaves like, or looks like and physical materials that will be used in future work such as icons. It is important that we considered all the alternatives in every stage of the design.

### **3.2.3 Building interactive versions of the designs**

Building interactive versions in each stage of the design serves as a form of communication and evaluation. It can be roughly separated into lo-fi prototyping and hi-fi prototyping. The former usually relies on an illustrator performing and explaining whereas the latter is often a stand-alone device or software. It is inevitable during the design process to involve users; building such interactive versions is the most sensible way for users to evaluate such designs[20].

### **3.2.4 Evaluating**

Evaluating was the last stage of any design process [20]. The evaluation determines if the goals and requirements established in the beginning are met and fulfilled. The evaluation is to be done repeatedly during the design process. The form of evaluation includes user testing and usability testing.

### **3.2.5 Interface metaphor**

An interface metaphor is a way to conceptualize the design. Using interface metaphors means that the conceptual model is developed to be similar to a physical entity but also has its own properties and behaviors[20]. Interface metaphors were based on conceptual models that combine familiar knowledge with new concepts[20], popular examples include mailboxes, folders, etc. The pros of metaphors are obvious: they provide users with a familiar orienting device and help them learn the interface quickly[20]. However, there were oppositions to the method: oftentimes, designers who try to make the product behave exactly like the physical entity to be compared miss the point of using metaphors while others use metaphors only as a literal model[20]. Another major criticism is that metaphors are not only limiting to designers, but they are also limiting to users, as they can obstruct users from understanding the real functionality beyond the functionalities of the compared entity.

## 3.3 Design Guidelines

### 3.3.1 User Centered Design

User-Centered Design (UCD) is a school of design thinking that stresses the importance of user involvement in the design process. UCD is a ubiquitous process in ID, suggesting that user needs should always be the driving force behind product development; the key to successful product development is the continuous involvement of users. There were three guidelines proposed by Gould and Lewis in their article *Designing for Usability* [22]:

- *Early focus on users and tasks*: It was important to ask in direct contact with the users and potential users as opposed to speculating with second-hand materials about them [22]. Interviews and field studies should be carried out by the design team. Understanding what users currently have was crucial to the design team, which is eager to provide a "better" solution. To avoid the typical hammer-and-nail situation, Gould and Lewis suggest we could keep a group of intended users close by to increase the saliency and importance of usability issues [22].
- *Empirical measurement*: User tests with actual behavioral measurements of usability as early as possible can be a great help to design. Concrete evidence gained in user tests is always more reliable than the speculation and experience of designers. The test should be user-oriented rather than system-oriented, as it is the user's response to such a system that concerns the designers. This will be discussed in more detail in Section 3.10.
- *Iterative design*: The design ought to be iterative in order to consider the results from testing at each stage. Based on feedback from each stage of development, we can therefore improve the product.

### 3.3.2 Norman's Seven Fundamental Design Principles

Don Norman in his "The Design of Everyday Things" [21] proposed seven fundamental principles in design:

- *Discoverability* makes it possible for users to determine the current state of a system and their possible actions.
- *Feedback* requires the system to provide full and continuous information about the results of the user's action and current state. The user could be easily confused or panicked when they do not receive ample and immediate feedback.
- *Conceptual model* is the information that has been projected by the design into the user's mind, leading to understanding and a feeling of control.
- *Affordances* suggests to the user that this system could be used to achieve desired actions. Much like "Chekhov's gun", every part of a system should have a clear indication of what it is used for.

- *Signifiers* ensure that discoverability and feedback are well communicated and intelligible.
- *Mappings* describes the relationship between controls and their actions. For example, a good envelope icon tells users that this tool could be used and understood similarly to physical mail.
- *Constraints* limits people to stay on the intended functionalities by conveying the physical, logical, semantic, and cultural constraints.

## 3.4 User Study

A good user study is crucial to the development of any product as it collects information about the users and their needs. The main goal of conducting a user study can be viewed as *identifying needs* in the first part of interaction design (ID), *establishing requirements*. The user data gathered through various means will lay the foundation for future design. Different information-gathering techniques can be used in user studies. A rough paradigm of a user study provided by Magnusson et al.[23] suggests we "set an objective, decide on the methods, design the tasks, determine the setting, decide what to record, determine the roles, determine which users to involve, and prepare."

### 3.4.1 Who are the users?

We first have to determine our target user group in the user study, it is vital to the whole UCD process[20]. The users can be classified into the following categories:

- *Primary users* are using the system frequently.
- *Secondary users* use the system occasionally or are indirectly using the system.
- *Tertiary users* are affected by the system somehow, like installing, introducing, or purchasing, etc...

### 3.4.2 Data-gathering

Data gathering is key in a user study. Depending on the type of data collected, the collected data could be classified into quantitative and qualitative data[24]. There are different means to collect data from users, which typically include interviews, questionnaires, and field observations.

#### Qualitative versus Quantitative

Quantitative data refers to data that can be represented numerically, but qualitative data cannot. Quantitative data gives a broad yet efficient overview of the user group

that provides a good foundation for understanding users while granting credibility to our decision-making process. However, quantitative data lacks the ability to explain the motives behind the phenomena exposed by it. Only qualitative data can provide complex and more thorough answers to questions like *what, why, and how*. Qualitative data can help the designers understand the context and constraints of a product and identify behavioral patterns as well as potential users [25].

### Interviews, Questionnaires, and field observations

Interviews are a research method that mainly collects qualitative data and is typically used in earlier phases of the design process to help establish requirements. Depending on the different purposes of an interview, we select interviewees that have different relationships with the product to be developed. For example, we could have *user interviews* that focus on actual or potential users in the target group of a product in order to understand the current problems to be solved and the motivations of them using the product. We could also have a *subject matter expert” (SME) interview* that aims to extract information from people within the target domain. It provides very valuable information and fresh yet pointy perspectives on the design [25]. An interview could be *structured, semi-structured, or unstructured*. It is recommended to think through the outcomes before carrying out an interview, as it typically generates a large amount of data [23]. Questionnaires primarily collect quantitative data. One major advantage of this method is that it is inexpensive to carry out and analyze [23]. It is essential to make sure questions are understandable, in order, and unbiased when developing a questionnaire. Running a pilot test is always a good choice when using questionnaires. Field observation could be a tricky yet productive way of gathering quantitative and qualitative data by observing the users in actual scenarios with as little interference as possible. The drawback of such a method is also obvious: it is difficult to carry out and analyze; it also requires extensive skills and knowledge from the observer. It is important to triangulate the data that we found with different methods to keep the designer’s own presumptions as far away as possible from the actual data gathered.

## 3.5 Usability and User Experience

Usability and user experience (UX) are two common terms associated with goals in ID. According to the definition by ISO 9241-11, usability is ”the extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” [26], Jeff Rubin [27] defines usability from six key aspects:

- *Usefulness* is the degree to which a product enables users to achieve their goals.
- *Efficiency* is how quickly (and easily) the user can accomplish a goal.
- *Effectiveness* is the extent to which the product behaves as expected and achieves what the users intended.

- *Learnability* is how many skills must be learned and how well can the users remember such skills.
- *Satisfaction* is the user's perceptions, feelings, and opinions of the product.
- *Accessibility* is enabling the product to be usable by people who have disabilities.

According to the definition by ISO 9241-210, the term UX is instead defined as "a person's perceptions and responses resulting from the use and/or anticipated use of a product, system, or service" [6]. The term UX can, in other words, be described as how well a product is being perceived by users when taking the emotional effects into account. UX is all about the feeling and satisfaction that a user gets when interacting with a product." [28] Usability and UX are often used interchangeably in some contexts, but we stress the difference between the two notions: while usability focuses on meeting a broad spectrum of requirements when designing a product, UX instead focuses on the emotional and social influence of such product that could be dependent on experiences, attitudes, personality, and knowledge [24].

## 3.6 Brainstorming

Brainstorming is an individual or group method for generating ideas, increasing creative efficacy, or finding solutions to problems. The basic setup of a brainstorming session is "a clearly defined question", "a 'criticism free' time for generating ideas", and "discussing, critiquing, proposing ideas" [29]. The basic rules are:

- Aim for sheer quantity
- Defer judgment about the quality of ideas
- Encourage new and wild ideas

## 3.7 Conceptual Design

Conceptual design is the very first phase of *designing alternatives*. It gives an abstract description of how the product would be used and perceived by the user with a conceptual framework [20]. The design is still on an abstract level at this point, so only the major lines and dots are visualized with sketches. However, as simple as a conceptual design may get, the full desired functionalities should be present. In the following part, we will present several methods that we used in combination to communicate our conceptual design.

### 3.7.1 Personas

A persona is a detailed description of a typical user of the product under development [20]. It represents a category or summary of real users that the team encountered in

data collection or accumulated from previous research. Each persona has a distinct set of skills, attitudes, tasks, environment, and goals defined precisely by researchers, giving the imagined scenario a sound base of realism. For designers, personas provide a specific focus instead of a broad user group. It is therefore very common to produce several personas from a user study. The main advantage of doing so is that personas can accommodate a variety of users by designing for specific types of individual users with specific needs [25]. Additionally, personas can help designers build a concrete use case and communicate with the team and other people involved. Moreover, a persona urges designers to think thoroughly about an entire scene rather than specific functionalities.

### **3.7.2 Storyboard**

A storyboard is a method used to visualize and identify key activities that are performed when the user interacts with the product while performing a specific task [20]. Such interaction is depicted using simple sketches and words. Storyboarding has the advantage of providing context for the interaction. Designers can easily reach a mutual understanding of the context and environment in which the use of the product takes place.

### **3.7.3 User Flow**

A user flow is a diagram that shows the complete path a user takes when using the product. It is a very useful tool when it comes to activities that are more complex; designers can easily map each step taken by the users to their needs. It can also help minimize frustration during the course of use.

## **3.8 Prototyping**

*Prototyping*, or the making of prototypes, is an essential activity of interaction design[20]. The prototype is a simplified yet illustrative version of the end product to be designed[22]. It could serve various purposes, depending on the current stage of the design process. Different stages require a different level of detail. Early stages do not require much detail while emphasizing quick and agile development; later stages require more and more details to be as realistic and independent from human factors as possible. The prototypes are good tools for ideation and explorative studies. Using such a method, one is able to quickly solidify an idea and communicate it with other designers, clients, and stakeholders. The prototypes are also used for user testing and usability testing. By showing the users a concrete product, it is much easier to define and ask questions, compare it to other products, and illustrate potential weaknesses in the product. The feedback from the testing would be more valuable and reliable for the designers when making decisions and improving their designs. Moreover, prototyping is a relatively cheap way to go, as it requires much less time and resource commitment. Prototypes can take many forms; based on the level of detail and re-

source commitment, we roughly divide them into two categories: low-fidelity (low-fi) prototypes and high-fidelity (hi-fi) prototypes.

### 3.8.1 Low Fidelity Prototypes

A lo-fi prototype is a comparably simpler prototype that is often used in earlier stages of the design phase. It is a way for the designers to organize concrete thoughts from highly conceptual discussions. It is often made of paper, cardboard, or other quick and cost-effective means. As the method is rather primitive, this method almost inevitably involves human enactment, or "*Wizard of Oz*" method[30]. It would be easier for both users and designers themselves to reflect upon, modify, or even rebuild when not much time and resources have been dedicated to the development. It also performs very well when exploring different design alternatives.

### 3.8.2 High Fidelity Prototypes

A hi-fi prototype is a comparably more complex prototype that is often used in the late stages of the design phase[22]. It is usually intended to demonstrate a solution and conduct user tests. Hence, the form of such a prototype is most likely a stand-alone with little to no external help during use, often made with software and sometimes hardware as well. Compared to lo-fi prototypes, hi-fi prototypes are much more similar to the end product. A successful hi-fi prototype must demonstrate the main flow, comprehensibility, and concept of the goal product. Testing on this prototype would give a good reflection regarding the usability and UX of the final product as well as reveal fatal errors before the actual product is developed.

## 3.9 Evaluation

Evaluation of a product is typically carried out in the later stages of product development, usually at the end. Exactly how to evaluate a product depends heavily on its aim and constraints, as well as the resources that could be allocated. In this section, we will present several methods for evaluation.

### 3.9.1 Usability testing

One of the most common practices is usability testing. Usability testing stresses scientific methodology [27] that is defined by the following:

- *The subjects in the experiment must be carefully selected*
- *The experiment must be pre-designed*
- *The experiment should be carried out in a controlled environment*



- *The experiment should be moderated and documented*

However, it also has some differences from an actual scientific study. For example, it states a research question instead of a hypothesis and demands that it be rarely randomly sampled or have enough statistical significance. The test aims to recreate a reasonably similar use-case scenario. The core aim of a usability test is to see if the product "works as intended", and find possible improvements. There are several techniques that could be adopted in the testing.

## **Think Aloud**

*Think aloud* is a widely adopted technique in usability testing. By encouraging participants to speak out about their actions, intentions, and reactions while using the product the designers can collect data about the immediate thoughts and reactions of users. The actions and intentions of users give the designers a direct comparison against what they have thought of and planned out in their original design. The immediate reaction of users is especially valuable because people tend to filter out their confused and frustrated moments rather quickly, giving much more positive feedback, especially in test environments.

## **Observation**

Observing the participants is also a common practice in usability testing. It is crucial to plan out what to observe when planning the test so that the team can generate meaningful data, both qualitative and quantitative.

## **Debriefing**

*Debriefing* is crucial to a usability test. A debriefing session gives the test moderator a chance to discuss what happened during the test session with the participant. [24]The main focus of a debriefing session is for the test subjects to understand the reasoning behind each choice and action by users and their attitudes toward the product.

### **3.9.2 Questionnaire**

Questionnaires with rating scales associated with answers are more subjective; they provide quantitative data about the product that has been designed. The answers provide a direct and quantifiable answer to the question, "Is the product usable? We will present two kinds of questionnaires we used during testing.

## **System Usability Scale**

The System Usability Scale(SUS)[31] consists of ten statements about the usefulness and feeling of using the product. The ten statements would be answered by participants regarding how much they agree with them. Compiling the results, we could get a SUS score. Five of the ten questions are positive statements, and the other five are negative.

## **NASA Task Load Index**

The NASA task load index(NASA TLX)[32] is an assessment of workload on six scales, each containing seven points. The assessed scales are mental demand, physical demand, temporal demand, performance, effort, and frustration. It provides a detailed statistical answer focusing on the workload of a system.

# 4 Design phase

*In this chapter, we presented a detailed account of our design process until the development of our final prototype. This process consisted of several activities: identifying needs through a user study and interviews, establishing the requirements of our product, conveying our conceptual design, and building lo-fi prototypes. All the design decisions during this process were presented and motivated*

## 4.1 Identifying Needs

Our design process started with identifying needs. The aim of this thesis was to test possible improvements that VR can bring into the working process of developing experiences with large interactive systems. Our research questions for the user research were simple:

- What are the existing challenges in the course of design and development for large interactive experiences?
- What is the user profile of people designing and developing large interactive experiences?

We started our research in two different directions. Literature on the design and development of large interactive displays is one of our starting points. More thorough discussions were already presented in *Literature review* section, so in this section, we only discussed the takeaways from literature and their impact on our design thinking. The other direction was a user study. Since there already existed a target user group for ICS, and considering the limitations on both time and resource constraints, we decided that instead of performing a full-scale user study where we try to get direct contact with primary users who are going to be customers of our solution, we would rely on existing vision and knowledge of target users within the team. Hence, we set up interviews with experts and stakeholders of the entire solution in the team to extract information about users. But verbal descriptions and vague paintings of users were not enough to guide our design, therefore, personas were generated.

### 4.1.1 Literature study

Going into the literature, we aimed to answer research question one, "*What are the existing challenges in the course of design and development for large interactive experiences?*" From the literature research, we learned that these systems face major challenges of screen blindness and interaction blindness[17]. We believed that interaction blindness, in particular, can be dealt with through better prototyping methods. There are four major reasoning for our presumption:

- **Testing Interaction Visibility:** Through prototyping, designers can simulate and test different interactive elements, their placements, and visual cues on a large interactive display. By observing user interactions with the prototypes, they can identify potential areas of interaction blindness and make adjustments to improve visibility. For example, designers can experiment with the size, color, contrast, or animation of interactive elements to make them more attention-grabbing and noticeable.
- **Iterative Design and Feedback:** Prototyping allows for iterative design and feedback loops, enabling designers to quickly make changes and improvements based on user feedback. By involving users in the prototyping process, designers can gather valuable insights into the visibility and understandability of interactive elements. Users' feedback can help identify areas where interaction blindness may occur and guide design decisions to enhance visibility.
- **Realistic Simulation:** Prototyping methods that offer realistic simulations of the final interactive experience on large displays can provide a more accurate representation of how users will engage with the interface. By creating high-fidelity prototypes that replicate the display's size, resolution, and interaction capabilities, designers can better assess the visibility and effectiveness of interactive elements. This enables them to address any potential interaction blindness issues early in the design process.
- **User Testing and Observations:** More realistic prototypes are able to facilitate user testing and observations in a controlled environment. Designers can observe users interacting with the prototypes on large interactive displays, studying their behavior, and capturing insights on any instances of interaction blindness. By analyzing user interactions, designers can identify patterns, recurring issues, and areas where users tend to overlook or misunderstand interactive elements, leading to design refinements.

### 4.1.2 Interviews with user expert

As our project fits in a bigger solution, ICS, we decided to interview our user expert, the UI designer in our team. The interview is structured as an open interview because we already have a very specific research question to find out: *"What is the user profile of people designing and developing large interactive experiences?"* Firstly, we learned that ICS is meant to be a low-code, simplified environment to assist in the development of experiences presented in LIDs. So the user characteristics are shared with those of a typical low-code environment:

- **Technical Proficiency:** A user of a low-code environment needs to have some level of technical proficiency to be able to understand basic technology and software development concepts. The user also needs to be familiar with and enthusiastic about interactive technologies to some extent to understand common notions.
- **Problem-Solving and Iterative Thinking:** Low-code developers possess problem-solving skills and are comfortable with an iterative development pro-

cess. They should be able to identify issues or challenges during development, iteratively test and refine the interactive experiences, and find creative solutions to improve the overall user experience.

- **Collaboration and Communication:** Collaboration skills are important for low-code developers working on interactive experiences for large displays, as they often need to collaborate with designers, content creators, and other stakeholders involved in the project. Effective communication and the ability to understand and translate requirements into interactive elements are crucial for successful collaboration.

Secondly, this platform is targeted at designers and developers of experiences among whom artistic creativity is also valued. They have design sensibilities and a creative mindset. They need to understand visual aesthetics, user experience principles, and interactive design patterns to create engaging and visually appealing experiences that resonate with the target audience. Finally, the users tend to be adaptive and learn continuously. They are willing to learn new technologies and skills, which in turn will require a learnable system.

## Low code environment

A low-code environment is a software development platform that allows users to build applications with minimal coding or programming knowledge[33]. It provides a visual interface and pre-built components that enable users to create applications by dragging and dropping elements, configuring settings, and using declarative programming techniques. The aim of a low-code environment is to simplify and accelerate the application development process, making it accessible to a wider range of users, including non-technical professionals. Bock et al. [33] summarized several characteristics of low code platforms.

- Low-code platforms integrate various classical development components in one environment.
- Reuse is addressed at a generic architectural level, not at a domain-specific level.
- Productivity gains mainly ensue from reducing the efforts of routine tasks.
- No new technology.
- Risk of lock-in effects.

### 4.1.3 User personas

In order to make more specific design decisions, we needed to precisely define the characteristics of the intended users [23]. We created these personas based on the profile of our intended users of a low-code environment for developing large-scale interactive experiences.

## Persona 1

### Creative Designer - Clara

Background: Clara is a creative designer with a passion for interactive experiences. She has a background in graphic design and has recently started working on projects that involve large interactive displays like interactive hallways. Clara is a self-taught programmer and has a basic understanding of coding concepts, but her expertise lies in design principles and user experience.

Demographics:

- Age: 30
- Gender: Female
- Education: Bachelor's degree in Graphic Design
- Experience: 5 years of experience in graphic design, 1 year in interactive experience design

Goals and Motivations:

- Create visually stunning and engaging interactive experiences for large displays
- Simplify the development process by utilizing a low-code environment
- Enhance user experience through intuitive and creative interactions
- Collaborate with developers and content creators to bring ideas to life

Pain Points:

- Limited coding knowledge restricts her ability to create complex interactions
- Balancing design aesthetics with technical limitations of the low-code environment
- Ensuring the scalability and performance of interactive experiences on large displays

## Persona 2

### Content Developer - Alex

Background:

Alex is a content developer who specializes in creating interactive content for large interactive displays. They have a background in multimedia production and are skilled

in content creation tools such as video editing, animation, and interactive storytelling. Alex has a basic understanding of coding but prefers to use low-code environments to streamline the development process.

#### Demographics:

- Age: 27
- Gender: Male
- Education: Bachelor's degree in Multimedia Production
- Experience: 3 years of experience in content development, 1 year in interactive experience creation

#### Goals and Motivations:

- Develop interactive content that engages and educates audiences in interactive hallways
- Utilize their creative skills to create immersive and captivating experiences
- Collaborate with designers and developers to bring their interactive content ideas to life
- Use a low-code environment to quickly prototype and iterate on interactive concepts

#### Pain Points:

- Limited coding expertise restricts their ability to create complex interactions
- Balancing creative storytelling with the technical constraints of the low-code environment
- Ensuring the interactive content is accessible and inclusive to a diverse audience
- Keeping up with emerging technologies and trends in interactive display experiences

These personas represented different user types involved in developing interactive experiences for large displays within a low-code environment. Clara, the creative designer, focuses on visual aesthetics and user experience, while Alex, the content developer, emphasizes creative storytelling and collaboration. Both personas rely on the simplicity and efficiency of a low-code environment to bring their ideas to life while working within their respective areas of expertise.

## 4.2 Establish Requirements

Based on the user profile we generated in the above section, we had a clear picture of our user base: enthusiastic users of a low-code environment that assists their creative work. Based on this user profile, we decided to prioritize the following usability goals:

- *Learnability*: Since our users are technically proficient and enthusiastic, it is important for them to learn how to perform the basic functionalities quickly.
- *Efficiency*: One of the goals of a low-code environment is that it accelerates the design and development process with a simple platform with minimal functionalities oriented towards solving generic problems.
- *Accessibility*: Our customers could vary in age, gender, background, and even body conditions. So accessibility is very much looked upon.

Based on these usability goals, we have also highlighted some essential points in our consideration.

### Usability through UI

Low-code environments were designed to make application development accessible to users with varying technical backgrounds. By employing a simple UI, the platform reduces complexity and makes it easier for users to navigate and understand the available features and functionality. A straightforward UI reduces the learning curve and enables users to quickly grasp the platform's capabilities, facilitating a more efficient and enjoyable development experience.

Our user interface must be clear, with the minimum number of buttons and frames possible. This would effectively minimize miss clicks and give the user a clear and straightforward message on what to do. Alternate pages will also be minimized, we want the user to follow a one-way workflow until finishing the application use.

A simple UI:

- promotes an intuitive workflow by organizing features and options in a logical and usable manner. It allows users to easily locate the tools they need and perform actions without having to dig through complex menus or interfaces. This improves productivity and helps users stay focused on the task at hand, allowing them to concentrate on designing and developing their applications rather than struggling with the tool itself.
- with which users can quickly become proficient in using the low-code environment, reducing the need for extensive training and ongoing support. The intuitive nature of the UI allows users to rely on their own exploration and experimentation, minimizing their reliance on external assistance or documentation. This saves time and resources for both the users and the support teams.



- helps minimize the potential for errors or mistakes during the development process. Clear and straightforward interfaces make it less likely for users to accidentally select incorrect options or make unintended changes. By reducing the chances of errors, a simple UI promotes accuracy, saves time spent on debugging, and enhances the overall quality of the developed applications.

Inclusive design is a key consideration in UI development, and a simple UI contributes to better accessibility for a wider range of users. A streamlined and well-organized interface is easier to navigate for users with disabilities or impairments. It ensures that the low-code environment can be used by individuals with varying levels of technical expertise or physical abilities, fostering inclusivity in the development process.

### **Intuitive Design metaphors**

Design metaphors allow users to receive visual and behavioral hints that are mapped to prior knowledge and experiences on how to use the product[20]. A correct and effective mapping ensures the learnability of the product. By incorporating familiar visual cues and interactions, users can easily associate metaphorical representations with their existing mental models. This familiarity reduces the learning curve and allows users to quickly grasp how the system or interface works.

Metaphors also act as cognitive anchors, providing users with reference points and mental hooks to remember and recall the system's features and actions. When a design element or interaction aligns with a well-understood metaphor, users can easily associate and remember its purpose, making it easier to recall and use the system in subsequent interactions.

Taking into account the oppositions of using metaphors, we first Design metaphors that align with users' existing mental models allow for the transfer of skills from one context to another. If users are already familiar with a metaphor used in a particular system or interface, they can apply their knowledge and skills from that context to quickly learn and adapt to the new environment. This transfer of skills speeds up the learning process and increases user confidence.

When design metaphors are consistently applied throughout the system or interface, they contribute to coherence and predictability. Users can anticipate how different elements and interactions will behave based on their understanding of the metaphor, enhancing learnability and reducing the need for explicit instruction or documentation.

### **Efficient workflow**

An efficient workflow enables low-code developers to work more productively [20]. By eliminating bottlenecks, simplifying complex processes, and optimizing the user experience, developers can focus on creating and refining applications rather than

getting caught up in tedious or time-consuming tasks. This improved productivity allows developers to accomplish more in less time and with fewer resources.

An efficient workflow facilitates collaboration among team members involved in low-code development projects. By providing clear guidelines, standardizing processes, and enabling seamless integration with version control systems or collaboration tools, a well-designed workflow ensures smooth coordination and communication between team members. This promotes effective collaboration, reduces misunderstandings, and enhances overall project efficiency.

### **Ability to Assist communication**

The customer base of ICS is mostly business-oriented, which means our primary users mostly work in teams. This indicates that our product often involves multiple stakeholders, including designers, developers, and business users. Assisting communications within the prototyping tool promotes collaboration and facilitates the exchange of ideas, feedback, and suggestions. It enables stakeholders to provide real-time input, comment on prototypes, and participate in discussions, leading to more effective and informed decision-making.

Communication within the low-code prototyping tool also serves as documentation of discussions and decisions made during the prototyping phase. It provides a record of feedback, requirements, and design choices, making it easier to trace the evolution of the prototype and understand the reasoning behind certain design decisions. This documentation ensures that important information is captured and can be referenced later, supporting continuity and knowledge transfer within the project.

### **Conveying Accurate Scale and Contextual Information**

The key point of using VR prototyping on LIDs lies in the scale and contextual information that ordinary computer prototypes often fail to convey. Therefore, it would have been crucial for us to convey this information accurately as a substitute prototyping method. This information could help designers resolve the *interaction blindness* phenomenon as [18] suggests.

## **4.3 Conceptual Design**

As we established the core needs of users, we could move on to **conceptual design**. Our starting point was a brainstorming session. This section illustrated how we generated ideas, how they were communicated and received by stakeholders, and how the feedback further shaped the mental model.

### 4.3.1 Brainstorming session

The starting point of our conceptual design was a brainstorming session around several questions:

- Where do the functionalities of our product start and where should it end?
- What metaphor works best in the demonstration of prototypes to the users in VR?
- How do we allow for communication while using the product?

During the session, we each wrote down 3 thoughts on each question, and then compared notes to discuss our thoughts. This process would be repeated when any question was not answered properly.

For the first question, we had a pretty clear and unanimous answer. Our product should consist of 2 parts:

- **File input and setup:** Our product was part of a solution that contains an editor to facilitate the development of an interactive experience on LIDs. Such an editor would output a unified description file that we could use to generate interaction. Inputting files and generating assets according to the file was the starting point of the workflow. Since we wanted to incorporate free configuration functionality, we would also have to set up the spatial configuration of different experiences stored in different files
- **Inspection:** Once the file input and setup have been taken care of, the user should enter a preview mode where they can feel the generated prototype from a closer perspective with a free angle of observation.

### 4.3.2 Storyboard

In order to effectively communicate our mental model for the product, we decided to use the storyboard technique. We based the scenario on our persona Carla, the creative designer. She had just developed two interactive experiences with the editor in the solution package, but she had never seen it at full scale together. She wishes to inspect her work in VR. Figure 4.1 shows the intended workflow of Carla.

### 4.3.3 Feedback from the first iteration

As we discussed our initial design with our supervisors, we got valuable feedback on the need for a desktop version. There were several factors to consider:

- Current VR platforms are not equipped with a mature file input and output system, so migrating files would be interruptive for the workflow.

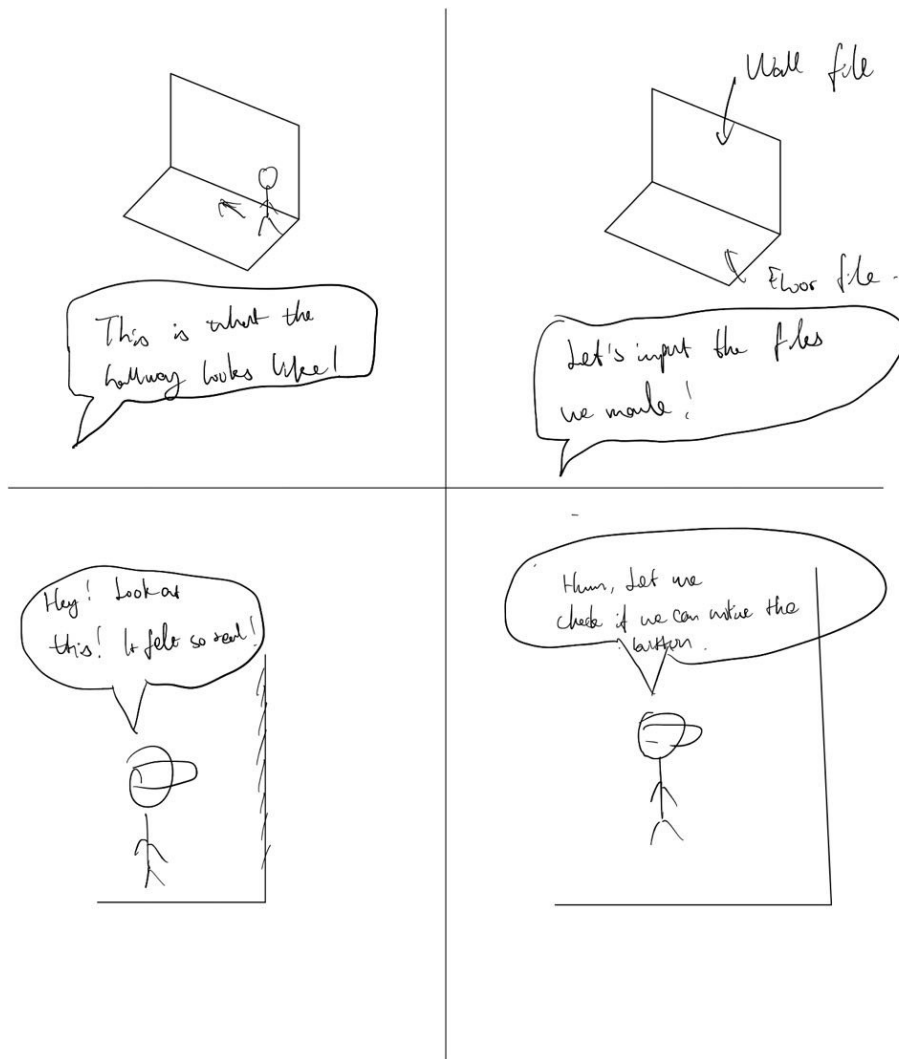


Figure 4.1: Storyboard

- Putting on a VR headset too often is tiring and hazardous.
- VR is blocking the communication between users.

Based on the feedback, we revised the workflow from the first iteration and added the desktop option. Now the revised workflow became:

- **File input and setup**
- **Desktop inspection:** Once the file input and setup have been taken care of, the user should enter a preview mode where they can feel the generated prototype from a closer perspective with a free angle of observation.
- **VR inspection:** There will be an optional VR inspection functionality to activate the VR interaction.

## 4.4 Low Fidelity Prototypes

After agreeing upon the mental model, we moved on to lo-fi prototyping. The major challenge here was to decide on the form of interaction, as it not only depended on the functionalities we aimed to provide but also on the time/resource constraints and prior knowledge or experience possessed by the user.

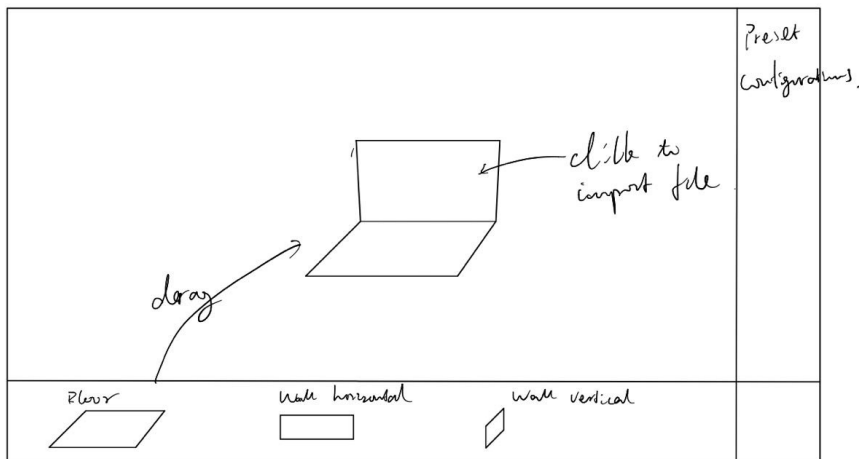
### 4.4.1 Important design choices

Along the way in designing our lo-fi prototypes, we faced several design choices. The way we resolved them can be summarized into three steps:

- Go through our usability goals, list the pros and cons for each goal, analyze the effect both choices had on each goal, and weigh their importance in our consideration by their respective level of importance to our overall usability as presented above.
- Picture ourselves as our personas and see if either solution satisfied their needs or avoided their weaknesses.
- Check technological viability.

### Hand Tracking

The functionality of hand tracking was proposed by us at a very early stage of our work as a trial of ideation. Naturally, during the stage of conceptual design, we tried to validate the idea. Considering our three prioritized usability goals, hand tracking performed very well on learnability as it requires little to no learning compared to using



**Figure 4.2:** Storyboard

hand controls; it had a moderate negative impact on efficiency because the current version of hand tracking responded slower than hand controls and could potentially confuse the user; as for accessibility, it was more natural and intuitive for inexperienced users, the other impacts were unclear. We then examined the solution with hand tracking using our personas; Alex and Clara both fit pretty well with the solution. However, the solution failed the technological viability test as we were not able to input and output files inside the VR headset. We hence turned to Oculus Airlink which ruled out the choice for hand tracking. This pivot will be discussed more in depth in the next chapter.

## Desktop Demonstrations

There was also a debate on whether we should have a desktop part to demonstrate the models of LID generated from the input files. It had a negative impact on learnability as it required the users to also learn desktop interactions; on the other hand, by having a desktop demonstration, accessibility was gradually improved. Both our personas could solve problems using the desktop version only, which would further improve the workflow. We have adopted it in our prototypes.

### 4.4.2 First Prototype

#### User Interface

In this version, we designed a free interface where interactive screens can be built into the desired configuration. There will also be pre-made and saved configurations. During the design of the UI interface, perceived affordances and mapping proposed by Norman[21] played an important role. See figure 4.2.

## **Desktop Preview**

Our metaphor for desktop preview was the third-person perspective in gaming. This meant that the camera would follow mouse movement and we controlled the virtual character that represented people walking by the interactive display using the arrow keys or the "WASD" keys on the keyboard. The reasons behind this were:

- Our primary users have good knowledge and enthusiasm for technology, which overlaps with people who have experience in video games, where this controlling technique is more common. This guarantees that most of our users will have prior knowledge that helps them grasp the controls quickly and easily.
- This combination of controls is also proven to be intuitive and quickly learnable by the gaming industry.
- The third-person perspective provides a rather free and immersive angle while giving the user an affordance to pay attention to what happens beside the character as it moves, making it easier to notice the interactive elements in the experience to be demonstrated.

## **VR Preview**

The user would be placed in a scale-accurate model of the experience previously designed and use hand tracking to "touch" the interactive objects and activate interaction.

## **Feedback**

The UI could be simpler and more constrained. There are not a lot of configurations, so we could provide a list of them, this saves a lot of development time and also lowers the error rate of our future product by limiting the choice of users.

The third-person perspective is good for a single user to interact, but it does not facilitate communication as well.

## **Technological Validation**

We tried to implement the first prototype in VR as a validation of the technology, but we then found out that hand-tracking cannot be performed when the headset is hooked up to the computer with Air-Link. This would mean that we have to rely on controllers to keep track of the hand.

### 4.4.3 Second Prototype

Based on the first prototype and the feedback we got, we developed the second prototype. We changed the import UI into a selection of different configurations and files to import; this would lower the error rate as well as development resources. We also added a free camera, which the user can control when temporarily severing the controls from the third-person perspective camera.



# 5 Development Phase

*This chapter presents an overview of how the development was conducted by discussing the methods and all the technologies used.*

## 5.1 Introduction

To build the ICS Visualizer, Unity3D (more information about it in the following sections) was chosen as the IDE for a few reasons. Apart from the fact that the base project, the ICS Editor, was built in Unity, it also provided support for all the features we intended to include in the visualizer. It also had great documentation, an active community, and a support team to help with any issues we faced. The SDKs we needed to make it work with VR were also tried and tested, which encouraged us to go ahead with Unity.

## 5.2 Resources and SDKs

Following are the major resources and SDKs used to build the ICS Visualizer.

- Unity3D
- Visual Studio 2019
- GIT
- Oculus XR Plugin (for Oculus Air Link)
- XR Plugin Management
- Computer with a high-performance GPU
- Oculus Quest 2
- 5GHz WiFi Router

### 5.2.1 Unity3D and Visual Studio

Unity3D [5] is a popular 3D game engine developed by Unity Technologies [34]. Its ease of use and the ability to create high-fidelity 3D experiences has made it the weapon of choice for developers who are especially trying to create immersive virtual reality experiences. Its powers also extend to support for a large selection of SDKs and

plug-ins that allow the user to build a complete desktop application with the required game mechanics. Unity3D is cross-platform compatible, has a large user base, and has continual updates and support, which makes it future-proof as well. Visual Studio [35] is an IDE (integrated development environment) that allows the developer to structure, write, compile, and debug their code in a single place [2]. It also supports version-control systems like GIT [36]. Unity3D and Visual Studio work in symphony to create interactive experiences where the visual and UI parts of the experience are constructed in Unity3D and the functional/back-end part is written in Visual Studio using C-Sharp as the programming language.

### 5.2.2 Oculus XR Plugin and Oculus Air Link

The Oculus XR plugin [37] is a software development tool developed by Oculus as a Unity plugin [38]. This SDK allows us to create native applications for all Oculus devices. It allows the developer to utilize the various features offered by the device, including hand tracking, see-through video, inputs from the controllers, and Oculus Air Link [39]. While designing the application, Oculus Air Link was especially interesting to us as it allowed us to stream the VR application from a desktop computer directly to the VR headset [39]. Following are the reasons why this feature held significant value to us:

- The headset uses WiFi 5 and above to stream the content, making it a tether-free experience.
- Since the content was only displayed in the VR headset and all the rendering and processing happened on the desktop computer, the application's quality was not restricted to the headset's GPU capabilities.
- This also improved the workflow of the application itself, as the developer did not have to install a separate application on the Oculus device to make this work. With a simple button click on the UI, the user could just put on the headset and experience the content. This also bypassed the file synchronization problem between the desktop visualizer and the headset. All the required files and assets lived on the desktop computer and were streamed to the headset in real-time.

## 5.3 Architecture

The project followed a singleton architecture [40] that was established in the base ICS Editor project. Briefly, a singleton architecture lets you create only one instance of the class in the project at runtime. This translated to creating a manager script (singleton) that kept track of multiple scripts that governed different features of the application and reported back to the manager-script.

## 5.4 Controls and Interaction

The controls and interaction were kept minimal and simple as far as possible, as this application would have users from various technical expertise levels, including those with no 3D software or gaming experience. It was made sure that all the navigation and features were put up as UI buttons in the application that could be interacted with just the mouse pointer.

### 5.4.1 Mouse and Keyboard

Inspiration was taken from general game controls. We used a simple third-person controller in which the player used the W-A-S-D or arrow keys to move and used the mouse to pan the camera. This input method allowed us to implement all the features we initially planned. Interaction with the content was mostly done by clicking using the mouse pointer for any touch content and by making the player character walk through the trigger area for any motion-triggered content.

#### Challenges

Since the visualizer allowed us to test different types of content, it was a challenge to include all the respective interactions into limited buttons and combinations without complicating it. For instance, one of the features we implemented was a free fly camera that allowed the user to set up the third-person camera independent of the player character. This was a feature requested by the stakeholders that allowed them to make screen recordings that could be shared to convey the experience. We made it happen by attaching the camera movement to the W-A-S-D keys and mouse by holding the right mouse button in that particular mode. This allowed us to move the player character using the same W-A-S-D keys and mouse when the right mouse key was released. This was one of the many ways we tackled the challenges when it came to keyboard and mouse interactions.

### 5.4.2 Virtual Reality

Interactions in VR are implemented with a few things in mind.

#### Locomotion

The inclusion of VR in the visualizer was to give the user a sense of one-to-one scale of the actual experience. This meant the user must be able to experience and interact with the content in the most natural way possible. For this, we added support for room-scale VR where the user could walk around naturally in the experience to move around. This brought forward another concern about the play area not being large enough in some cases. Another way to look at this concern is if the tester/user did not

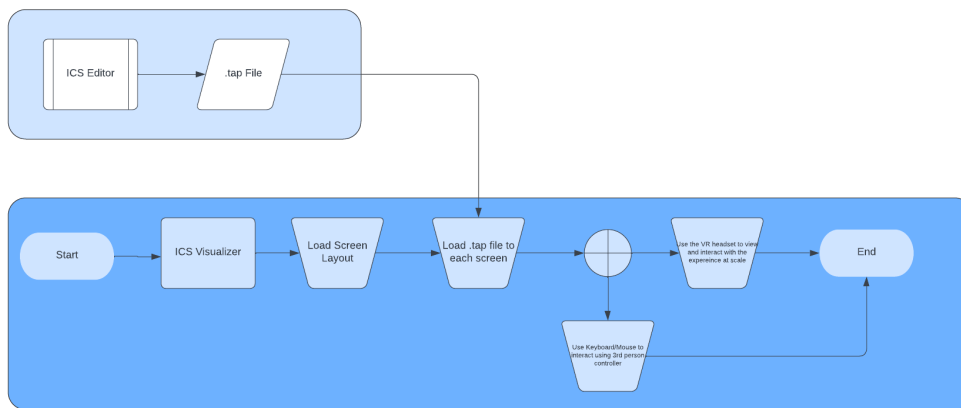
want to get up from their desk. This led to the inclusion of smooth locomotion using VR controllers. From [19], we learn that users feel nauseated due to cyber-sickness when using smooth locomotion. This happens due to the human brain receiving mixed signals from the eyes and vestibular system where one says the body is moving (the eyes seeing the environment move) and the other says the body is not moving (the vestibular system saying the body is stationary). To solve this problem, locomotion through teleportation was included in the visualizer. Initially, there was some reluctance with respect to adding teleportation due to three major reasons.

- The experience was primarily a walking experience and teleportation took the realism of the experience away.
- There was a learning curve to using the teleport mechanism well. Since a large part of our user base did not have gaming or 3D software experience, it would be cumbersome for them to learn this quickly thereby making the experience itself unusable.
- In teleportation, the movement would be too quick for the user to even notice. This meant that the user would miss out on any interaction that would happen during that time.

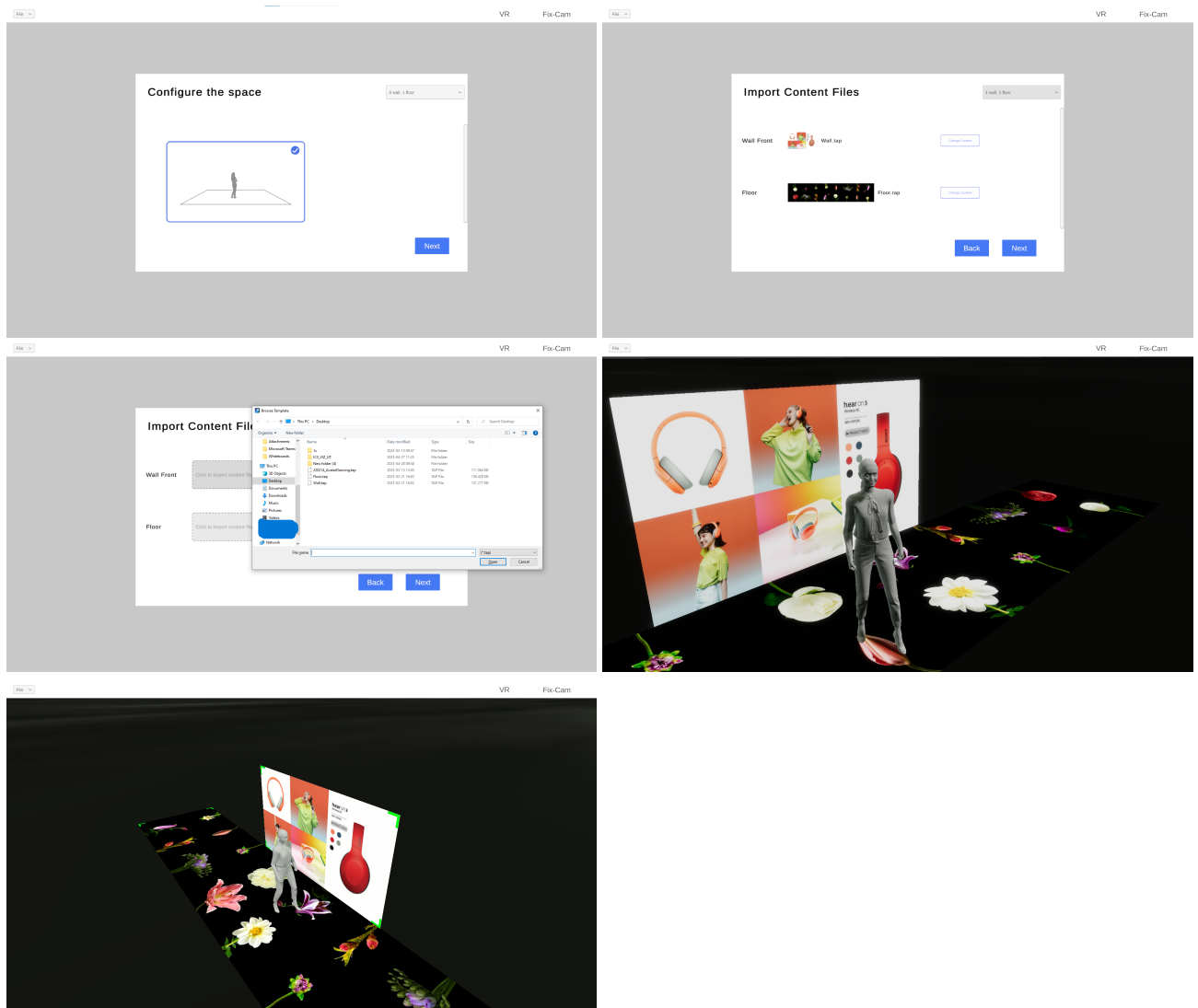
Despite all the above considerations, teleportation was included to accommodate user comfort. The teleport mechanism worked by letting the user choose where to move using a pointer and blacking out the screen for a microsecond before quickly moving the player to desired position. This gave the player an illusion of disappearing at the source and reappearing at the destination thus bypassing the motion sickness problem.

## **Interaction with content**

Interaction with content for VR was kept to natural interaction as most of the content had touch-based interaction. Apart from this, a consideration for content that used the motion interaction was also made and it was concluded that it would be easier to implement that using natural interaction where we tracked the user's hand position using the controller positions.



**Figure 5.1:** Flow of the ICS Visualizer applicaiton



**Figure 5.2:** Screenshots from the ICS Visualizer - in the order of the application flow from left to right

# 6 Testing Phase

*This chapter talks about the user testing that was conducted as part of the usability evaluation of the visualizer in detail.*

## 6.1 Introduction

In this chapter, we presented our usability testing, from how it was designed and planned to how it was carried out. We conducted the assessment test, which is typically done on a working or nearly-done prototype towards the end of the development cycle[27]. This chapter aims to answer the following questions:

- How did we justify conducting a test?
- What did we expect to get out of the test?
- How was the test planned?
- How did we conduct the test?
- What did we get from the test?

## 6.2 Why Test?

The reason for conducting such a test was to ensure that our product met our usability goals. Usability testing could inform design by gathering data from which to identify and rectify usability deficiencies[27].

At that stage, usability tests allowed us to observe how users interacted with the prototype, uncovering any usability issues, pain points, or areas of confusion. By directly observing user behavior and gathering feedback, we could identify specific problems and make informed improvements to enhance usability. Usability tests also provide an opportunity to validate design decisions and ensure that the prototype meets user needs and expectations. By observing how users engage with the prototype, we could assess whether the design aligned with their mental models, expectations, and workflows, confirming that the product was on the right track. Moreover, usability tests generated valuable insights and data that informed the iteration process. By analyzing user behavior, feedback, and performance metrics, we could prioritize design changes and make informed decisions on what improvements to implement in subsequent iterations, ensuring continuous enhancement of the prototype.

## 6.3 What to test for?

By this stage, our conceptual model was sound, therefore, the primary focus of our test was to assess how effectively the model had been implemented. As Rubin et al. stated, an assessment test should focus on flesh and bones rather than skeleton structure, which we would have focused on in explorative tests. In our test, we were specifically looking at four key usability features: usefulness, effectiveness, efficiency, and learnability. The reason behind this was that our users were looking for a quick and easy way to visualize and assess their interactive prototypes with minimal error and frustration. We downplayed the importance of satisfaction because this product was not meant to be used extensively, so as long as our product made sure that our users achieved what they set out to do, there would be satisfaction with the product to some extent. We partially overlooked accessibility in our testing for two reasons: the resource constraint, so that we did not focus on people with disabilities from the beginning, and the fact that we are not able to recruit volunteers from outside of the company.[27]

- Would the user be able to effectively learn and navigate themselves in the system when first exposed to it?
- How well did the users perform under our assigned task?
- How much effort was put into the task?
- What was the difference between VR and desktop versions in terms of usability?

## 6.4 Test plan

In this section, we presented how we planned the usability test based on the test plan we developed. There were several reasons to plan the test ahead and develop a test plan: [27]

- Test plan served as a blueprint for the test.
- Test plan was the communication vehicle between team members.
- Test plan helped us prepare resources ahead of time.
- Test plan was the focal point during testing.

### 6.4.1 Participants

We tested the application with ten subjects split into two groups. The subjects were recruited by Sony upon request. We aimed to have an equal distribution of gender and age among the subjects. The subjects should preferably have less background in 3D technology. Their experience in VR should be distributed to novice and experienced

users. During recruiting, they were asked to answer a small questionnaire to collect information about their age, gender, and prior experience with VR and related subjects to our thesis, such as video games, from which we drew the metaphor for the desktop application.

## **6.4.2 Resources**

In order to better prepare for the testing and also serve as internal communications with relevant parties, we listed our required resources for the testing. This includes a meeting room with enough space for participants to safely operate VR, a camera to film the screen and participant at the same time, a recorder, a computer, a WiFi router, a VR headset, several experience files as test material, and consent forms.

## **6.4.3 Tasks during testing**

In order to minimize human error and control the testing conditions, we developed the task list based on our workflow table 6.1. This was meant for members of the group rather than the participants to prepare for the test. As for the participants, we provided a working scenario intended to guide them through the experience. The tasks ought to be done by the user with minimal external help from us after reading the scenario we provided. Only when the situation is outside of the scenario and the scope of our testing would the moderator intervene. In the scenario, we provided a context in which the participants were placed in the role of interactive solution developers using our system to develop an interactive experience for clients abroad, and a mission of examining the separate scenes developed by other digital artists in the team to make sure that they were usable according to the design. Then the scenario guided the participants through separate steps, from importing files to activating the VR experience.

## **6.4.4 Registration and Pre-screening Questions**

We used an online scheduler to schedule our test sessions. They should take no more than 30 minutes, so there was no need for breaks in between two sessions. When the participants registered, they were required to provide their age, gender, prior experience level with video games, and prior experience with VR.

## **6.5 Test sessions**

This section presents how we conducted our test. The session is divided into three parts: briefing, testing, and debriefing. We took on the roles of moderator and observer. Throughout the test session, the screen and VR content are filmed.



**Table 6.1:** Task List for User testing

Task	Description	Expected Time	Criteria
Choose Environment	Choose the pre-set environment given by the moderator	10 seconds	The user chose the setting and went on to the next page
Import files	Import several files generated by moderator into the test program	30 seconds	The user imported the files and went on to next page
Explore the pre-viewed file	The user will enter a desktop version where they control a character to roam around the pre-viewed environment	3 minutes	Successfully triggered the intended interactions
Enter VR(not for desktop users)	Click to enter VR preview	1 minute	Entered VR Preview
Explore in VR	Control their virtual avatar to move around in the same space	5 minutes	Successfully triggered the intended interactions

### 6.5.1 Moderator and observer role

As we developed a scenario to guide the user, we did not feel the need to constantly interfere. Therefore, we did not have a very strict role division but rather an agile and adjustable division of work. However, it was important to have a clear idea of the division between the two. To put it in simple words, the moderator's responsibilities included:

- greets and have the participant sign the consent form;
- roughly introduces our project as well as other related information to the participant;
- answer any questions that the participant encountered during the test session that does not have a direct conflict with our goal of testing;
- debrief the participant.

The observer's responsibilities included:

- Time each task;

- Take notes during the test session;
- Manage the recorder and camera.

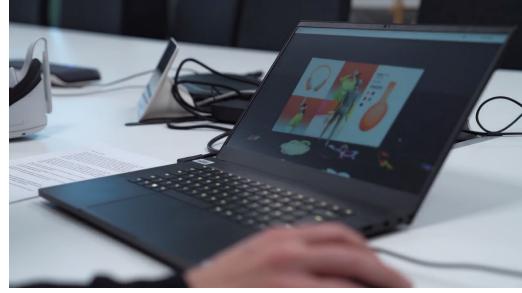
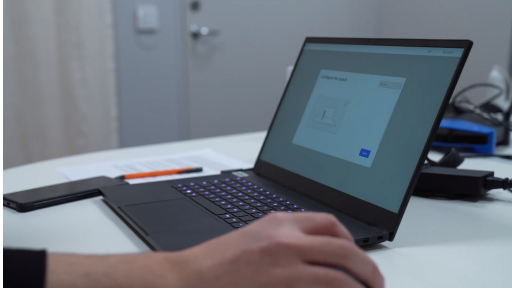
### 6.5.2 Briefing

Since our scenario provided some context to the user about a specific use case, we did not have to over-explain the test material to the user. We focused the briefing session on what we had done and what we wanted to test. During the short briefing, we told the participant that we had developed a visualizer for the ICS Editor program and that they were going to give it a try under the use case we provided. We also reminded the participant that we wanted to focus on the test of our prototype rather than the experience presented in our application, so any questions raised during testing that involved the experience that we used as test material inside could be discussed, but questions that would interfere with the testing might not be answered.

### 6.5.3 Testing

The participants were guided by the scenario we provided, and they had to follow it. We encouraged the participants to understand the scenario and try to execute the tasks as best they could so that we could get an organic test result. Of course, help was given when the participants felt stuck or uncomfortable. The setup for the user testing was as follows: A computer with a mouse and keyboard running the ICS Visualizer was set up in a well-lit room. The computer and an Oculus Quest 2 headset were connected to the same 5 GHz WiFi network. The room, apart from the desk, had ample space for the player to move around when they tested the visualizer in VR. A SONY digital camera was pointed at the computer screen to record the testing. Although screen recording software was initially considered, keeping in mind the computer's capabilities and possible performance drops that could occur because of running the screen recording software, the idea was scrapped, and instead, an external camera was used to record the screen. The participants were asked to think aloud [41] while performing the tasks from the scenario, and a voice recorder was placed in their proximity to record the same.

The scenario gave the user a background on what they would be testing by referring to one of the installations that they just walked by in the SONY office. Following that, it asked the participants to create a one-wall and one-floor screen setup, just like in the SONY office installation, and asked them to load specific content onto the screen setup, one of them being the wall content and the other being the floor content. Then they were asked to test the loaded content's functionality by moving around the experience using a keyboard and mouse. The participants were then asked to switch to a free-fly camera to set up the camera at a different angle and make the player walk around with the camera watching the player character from a fixed third-person point-of-view. Finally, the participants were asked to switch to VR and test the content in VR by using their hand controller for interaction and moving around by either walking or by using the controller's thumbstick to move virtually.



**Figure 6.1:** User testing session

### 6.5.4 Debriefing

The test session was wrapped up with a debriefing session, which included the following things: Once the participants completed all the tasks in the scenario, we indicated clearly that they were completed so that there was no ambiguity about their successful completion. Then the participant was asked to fill out a NASA-TLX form. After that, the participants were asked to answer a SUS questionnaire verbally, which we made notes of. We ended the debriefing with an open interview where we, while making notes, asked the participants what they thought about the application, its consistency, and any other general feedback.

## 6.6 Results

In this section, we present the outcome of the design and development phases, along with how we analyzed the results of the user testing and what we got after the analysis.

### 6.6.1 Design and Development Phase

The design and development phases resulted in an application that allowed the user to run and test the content built using the ICS Editor. It included features like avatar interaction with the content, simulated motion interaction with the avatar walking in the environment, simulated touch interaction using a mouse pointer, a free-fly camera to change point-of-view, and, of course, being able to experience the content in VR.

### 6.6.2 User Testing

We performed the usability test on 10 participants, among whom 7 are male, 3 female, 6 under 35, 4 above, 3 people with a low level of prior experience, and 7 with a high level of experience in VR and/or gaming. We collected quantitative data from questionnaires as well as qualitative data from the semi-structured interview during the debriefing and observation notes.

### 6.6.3 Method of Analysis

We have employed both qualitative and quantitative analysis on the results. In general, quantitative analysis is performed first in order to discover trends and distributions in the results. Then the hypothesis derived from statistical analysis would be triangulated with qualitative data.

#### SUS Questionnaire and Nasa-TLX Questionnaire

Based on the relevance to our usability goal set previously, we rate the answers based on the positivity or negativity (relative to our usability goal) of the participant's answer. Positive answers will be rated with positive scores and negative ones with negative scores. Then we can compute the mean value both for overall scores and for specific questions. We also compute the variance for each answer to see if there exists a split opinion on specific issues. We chose to use variance instead of standard deviation because our data did not have a visibly big difference and trends might have been missed using standard deviation. We would then cross-reference these data with their age, gender, and prior experience level we collected in the registration.

#### Interviews and Observation Notes

Qualitative data were organized and transcribed. The analysis will mainly focus on quantitative data, however, qualitative data could provide a valuable supplement to the tendencies spotted by statistical analysis. Moreover, these data opened a discussion into more in-depth reasons behind the participants' answers.

### 6.6.4 Evaluation

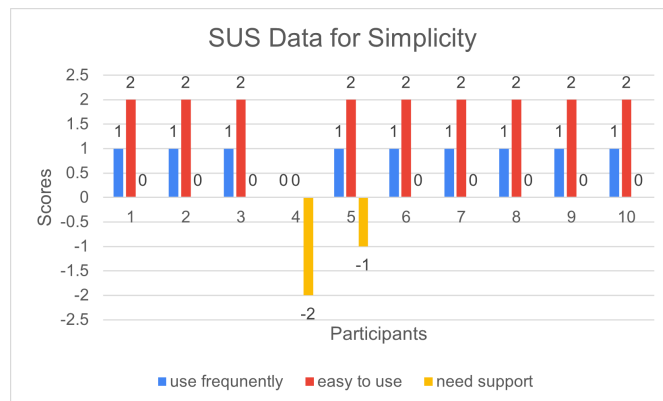
We analyzed our results with the six usability aspects by [27] with a heavier weight on the aspects that we prioritized in the design phase. We formulated the following variables whose added weight would tell us the extent to which the application met them.

- Complexity
- Ease of use
- Need for support
- Quick learnability
- Cumbersome
- Need to learn a lot before using the application

We weighed these variables that we deemed more important in our overall scoring with scores between -2 and 2 and other variables between 1 and -1. These, along with a few other minor variables like confidence, consistency, and proper integration, gave us a scale between -9 and 7 that could be used to measure the applications, with -9 being absolutely unusable and 7 being a delight to use. The user's tests revealed that the application scored an average score of 4.8 with a variance of 5, which can be supported by the data collected from the test sessions. For example, we noticed that age weighed more than gender in deciding the score, thus making the application more inclusive. The findings with respect to the parameters were discussed in detail below.

## Simplicity

Variables affecting simplicity were "complexity", "ease of use", and "need for support". All these variables saw positive responses, apart from a few outliers. On the variable "ease of use", the application had an average score of 1.8 out of 2.0 with a variance of 0.4, and on the variable "need for support", the application scored -0.3 out of -2.0 (with -2 being a lot of support and 0 being no support needed) with a variance of 0.45. One of the outliers was a male participant who was over the age of 35. Gender did not play a role in this, and this could be observed in the data. The reason could be that people over the age of thirty-five possibly did not have as much exposure to computers and technology, which is why they answered negatively for the variable "need for support". This assumption was also confirmed when we cross-checked if the participant had experience in games and VR, which he did not.

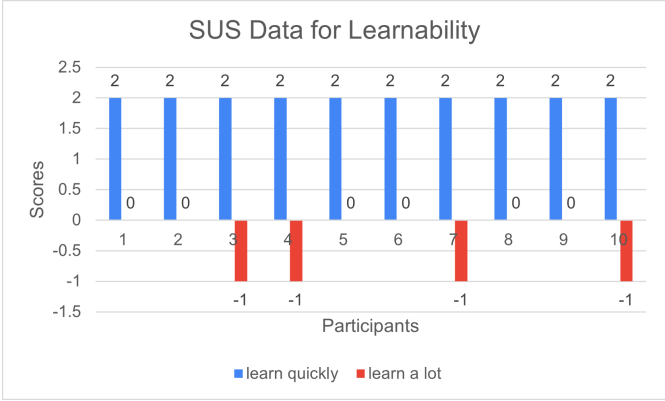


**Figure 6.2:** Data considered for evaluating the simplicity of the application.

## Learnability

"Quick learnability" and "Need to learn a lot before using the application" were the variables that impacted the learnability parameter. These variables were also met with positive answers, with an average score of 2.0 out of 2.0 with 0 variances for "quick to learn" and -0.4 out of -2.0 (where -2 is needed to learn a lot and 0 is no extra learning needed) with a variance of 0.2 for "need to learn a lot before using the application". All the participants felt that they could learn to use the application quite quickly and

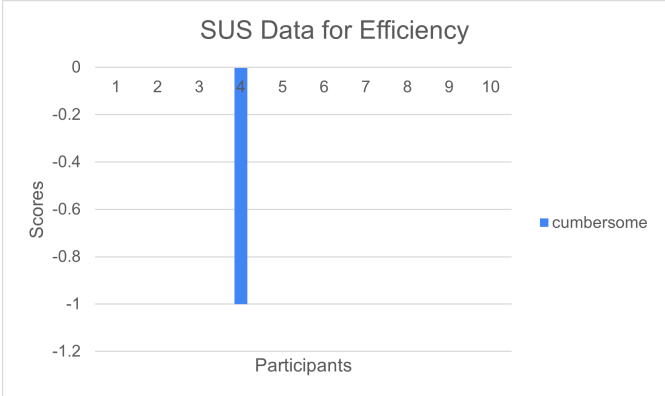
said that there was no significant learning curve. Most of the participants also felt that they did not need to learn a lot of things prior to using the application. There were four participants who felt that they needed to learn a few things—not a lot—before they used the application. Two of them were over the age of thirty-five, one male and one female participant, and the other two were under thirty-five with equal gender distribution. This again pointed towards the application being gender inclusive.



**Figure 6.3:** Data considered for evaluating the learnability of the application.

**Efficiency**

The variable "cumbersome" weighed in when it came to measuring efficiency. Efficiency is the amount of outcome per unit of work [42]. Cumbersome software is difficult to navigate, and the user faces resistance to completing the task at hand. The visualizer had an average score of -0.1 out of -2.0 (where -2 was extremely cumbersome and 0 was not cumbersome at all) with a variance of 0.1. We had one participant who said the experience was a little cumbersome, and we assume that it was because the participant was over the age of 35 and lacked experience with computer games.



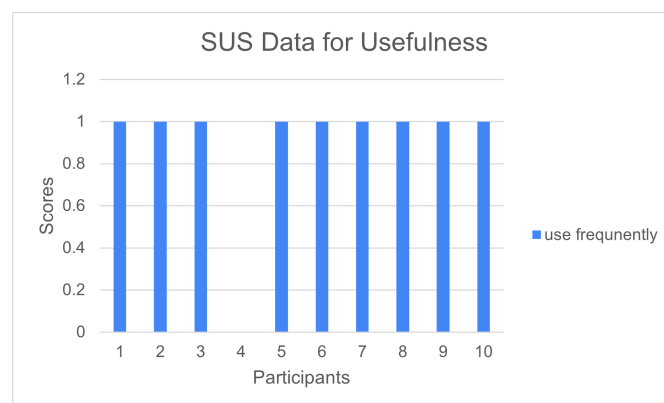
**Figure 6.4:** Data considered for evaluating the efficiency of the application.

## Accessibility

Accessibility was a multi-variate equation as we targeted a large demographic with multiple age groups and genders. Our test results show that the application was gender inclusive as both positives and negatives of the application were pointed out by both genders equally. We found out that the application is not for all age groups inclusive as participants over the age of thirty-five had slight difficulty using it. While this is one place where the application fails, the age group also falls into the upper limit of the ages that the application was designed to include. Participants aged between twenty-five and thirty-five did not report difficulties in using the application.

## Usefulness

The SUS questionnaire started with the question "I would use this application frequently (agree/disagree)" and we had an almost unanimous response that they would use the application. One outlier was a participant who reported that he would not use it. From this, we could conclude that the application is quite useful.



**Figure 6.5:** Data considered for evaluating the usefulness of the application.

## Satisfaction

Satisfaction with using any application was a result of being able to perform the tasks given and reach the end. We had mixed responses for this parameter from the participants. The variance of performance rating was high(3.57). This indicates although the participants were able to perform all the tasks they were given, they did not agree on having high performance because some participants completed it with external help.

# 7 Discussion

*In this chapter, we discussed an in-depth analysis and interpretation of the research findings as well as limitations and future directions*

## 7.1 Reflection on Research Findings

Based on our findings from the usability research, it is safe to say that we have fulfilled 3 of our usability goals: simplicity, learnability, and efficiency; inclusiveness based on gender is fulfilled, but age and relevant prior experience are affecting the use of our product. Apart from these goals, usefulness also received unanimously positive opinions from participants. However, there are pain points revealed in the usability test:

- *Satisfaction*: User satisfaction is called into question when a split view emerged on the self-ranked performance. We presumed that this is partly because the application itself did not have a definite end such that it might not have given the user a sense of completeness and satisfaction. Some participants also rated performance as low because they were able to finish the tasks required with help from the moderator rather than relying on the hints inside the application.
- *Controls*: There have been a lot of complaints by test participants that the controls are less than intuitive, with both mouse-keyboard interaction and VR. We presumed that this is a major reason why participants who are older and less experienced with VR tend to report a more negative score on satisfaction and performance.

### 7.1.1 Hints and Instructions

Several participants suggested that they could use more hints on the screen to remind them how to control the virtual character and move the camera, as well as hints on how to activate the VR option.

### 7.1.2 Free-fly Camera

This functionality confused six out of ten participants, some of whom questioned the reason for having such functionality. The original idea behind the functionality is to assist communications among users with a recording tool; a third-person view is better than a first-person one if the use involves an observer. However, the complex interaction led to clear frustration from test participants. Moreover, the lack of sharing



prompts or functionality prevented the users from generating the intended conceptual model for the camera.

## **7.2 Limitations and Delimitations**

There have been some limitations due to time, resources, and numerous other constraints. This part will mention the limitations and possible delimitations.

### **7.2.1 Explorative User Study At Earlier Stages**

We did not conduct mass surveys or other forms of explorative study in the early stages both because of a lack of time and access to the target users. Our user profile is based on the already established user profile of the ICS editor. We focused instead on the needs and requirements arising from literature and interview.

### **7.2.2 Frame Drops Due to Connection**

Oculus Air Link was chosen to have a tether-free experience which came with a cost. Since we had access only to a WiFi 5 router, and the air link worked well with WiFi 6, we experienced frame drops in the experience which could cause nausea among some users. Part of the reason was the WiFi 5 router; the experience itself was a high-quality one and since the computer used for testing was constantly running, it was not operating at its optimal CPU temperatures which also was a reason for the frame drops.

### **7.2.3 Limited Participant In Testing**

Since we could only recruit participants within Sony, it is hard to find more participants who may be able to produce more statistically significant data. We had to supplement our findings with qualitative generated during the test session like notes and interviews. We also cannot diversify the backgrounds of our participants due to the limited number of responded participants.

## **7.3 Future Directions**

This section presented some of the future directions that would have added great value to the product.

### **7.3.1 More Intuitive Interaction**

The mouse and keyboard interaction is very limiting and targeted users of our solution may not have a relevant background in 3D software. The interaction with the camera in the desktop version could be made more intuitive to the users. The interaction with VR could also be made more intuitive by applying hand-tracking rather than hand controls. This would open up the application to be able to prototype more kinds of interaction. It would also intuitively lead the user to touch and/or wave rather than pressing buttons on the hand control.

### **7.3.2 Custom Screens**

The current version of the application has a preset collection of screen setups like, one wall one floor, two walls one floor, etc. It lacks the ability to allow the user to create custom screen layouts. That is something that could be considered for the future scope. The feature could also be expanded to support importing a 3D model of the room (.fbx, .obj, etc) in which the screens were supposed to be set up and allow the user to dynamically load content onto them.

### **7.3.3 Customizable Environment**

We learned from the literature that it is important for designers to be aware of the environment of LIDs to be able to produce better experiences[18]. A customizable environment would allow designers to check how their experience fits in and draws attention in the surrounding environment.

### **7.3.4 Enclosed Environment**

The enclosed environment for the visualizer is one topic on which the view is divided. Some say that an enclosed environment would make the user feel claustrophobic and thereby break the experience while others say that experiencing the visualizer in an enclosed environment instead of a void of infinity would give them a sense of scale and direction.

## 8 Conclusion

With this thesis, we set out to answer two research questions, and all through the process, we kept coming back to the said questions and update our research methodology. Hence, all the phases, including design, development, and testing had the same goal: to answer the following questions

- How best to apply VR in the development of large-scale interactive content?
- What impact does VR have on the design and development of such interactive content?

Through our research, we concluded that virtual reality does a great job of communicating the scale of experiences. Since it also enabled natural interaction where the user could walk around and interact with the virtual environment naturally, we also hypothesized that the learning curve for experiences in VR would be quite small. This made us include VR as an extension to the desktop application, where the user could have control over the application from the comfort of his mouse and keyboard, and when he felt the need for natural interaction, he could switch to VR. These hypotheses were confirmed through our user testing sessions where the participants easily interacted with the content using VR. In fact, one of the participants went on to tell us that she found the VR interaction more natural and easy than the mouse and keyboard interaction, thus giving us the answer to our first question. VR can best help these large-scale simulations convey the scale of experiences and reduce the learning time required to master any interactions.

Moving on, we also found that VR enhances the eye for detail for designers and developers as all of the interaction was done from the first-person perspective. This allowed them to discover flaws that went unnoticed on a 2D screen. In addition to that, since the visualizer helped them visualize their creation, they understood the capabilities and limitations of the screen setups and the content, which boosted their creative freedom. It allowed them to create more refined and robust interactive content. VR also allowed the stakeholders to experience the content in the first person while it is being built and helped them make more informed decisions since they now understood the capabilities of the technology.

To conclude, VR is a young technology that is still trying to find its use cases and simulations and visualizations happen to be something VR can be really good at. Since the entertainment industry relies significantly on marketing the content produced through these large interactive experiences, it's about time they discovered visualizers in VR and harnessed their capabilities to streamline the design and development of such interactive content.

# Bibliography

- [1] Jerry Banks. *Discrete event system simulation*. Pearson Education India, 2005.
- [2] Wikipedia Contributors. 2023. URL: [https://en.wikipedia.org/wiki/Visualization\\_\(graphics\)](https://en.wikipedia.org/wiki/Visualization_(graphics)).
- [3] ScienceSoft. 2023. URL: <https://www.scnsoft.com/virtual-reality/prototyping>.
- [4] SONY. 2023. URL: <https://www.sony.com/en/>.
- [5] Unity Technologies. 2023. URL: <https://unity.com/>.
- [6] Wikipedia Contributors. 2023. URL: [https://en.wikipedia.org/wiki/Low-code\\_development\\_platform](https://en.wikipedia.org/wiki/Low-code_development_platform).
- [7] Department of Economic United Nations and Social Affairs Sustainable Development. *Transforming our world: the 2030 Agenda for Sustainable Development*. General Assembly. 2023. URL: <https://sdgs.un.org/2030agenda>.
- [8] Department of Economic United Nations and Social Affairs Sustainable Development. 2023. URL: <https://www.un.org/sustainabledevelopment/news/communications-material/>.
- [9] Lund University. 2023. URL: <https://www.lub.lu.se/en/find/lubsearch>.
- [10] Google. 2023. URL: <https://scholar.google.com/>.
- [11] Lufang Zhang, Xiayi Dai, Xingyu Xu, Gaoyan Shi and Zhichuan Tang. “User Experience Research on Product Design Evaluation based on Virtual Reality Prototyping”. In: *2020 IEEE International Conference on Progress in Informatics and Computing (PIC)*. 2020, pp. 214–218. DOI: 10.1109/PIC50277.2020.9350750.
- [12] I.D Bishop and B Rohrmann. “Subjective responses to simulated and real environments: a comparison”. In: *Landscape and Urban Planning* 65.4 (2003), pp. 261–277. ISSN: 0169-2046. DOI: [https://doi.org/10.1016/S0169-2046\(03\)00070-7](https://doi.org/10.1016/S0169-2046(03)00070-7). URL: <https://www.sciencedirect.com/science/article/pii/S0169204603000707>.
- [13] Günter Alce, Klas Hermodsson, Mattias Wallergård, Lars Thern and Tarik Hadzovic. “A Prototyping Method to Simulate Wearable Augmented Reality Interaction in a Virtual Environment - A Pilot Study”. In: (2015), pp. 18–28. DOI: 10.11159/vwhci.2015.003.
- [14] Günter Alce, Eva-Maria Ternblad and Mattias Wallergård. “Design and Evaluation of Three Interaction Models for Manipulating Internet of Things (IoT) Devices in Virtual Reality”. In: (2019). Ed. by David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler and Panayiotis Zaphiris, pp. 267–286.
- [15] Nick Taylor and Keith Cheverst. “Supporting Community Awareness with Interactive Displays”. In: *Computer* 45.5 (2012), pp. 26–32. DOI: 10.1109/MC.2012.113.

- [16] Nemanja Memarovic, Sarah Clinch and Florian Alt. “Understanding display blindness in future display deployments”. In: *Proceedings of the 4th international symposium on pervasive displays*. 2015, pp. 7–14.
- [17] Guiying Du, Lukas Lohoff, Jakub Krukar and Sergey Mukhametov. “Comparing two methods to overcome interaction blindness on public displays”. In: *Proceedings of the 5th ACM International Symposium on Pervasive Displays*. 2016, pp. 243–244.
- [18] Callum Parker, Martin Tomitsch, Nigel Davies, Nina Valkanova and Judy Kay. “Foundations for designing public interactive displays that provide value to users”. In: *Proceedings of the 2020 chi conference on human factors in computing systems*. 2020, pp. 1–12.
- [19] Mattias Wallergård, Günter Alce and Joakim Eriksson. *Virtual reality*. svenska. Studentlitteratur AB, May 2019. ISBN: 9789144122281.
- [20] Yvonne Rogers, Helen Sharp and Jennifer Preece. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2023.
- [21] Edward Tenner. “The design of everyday things by Donald Norman”. In: *Technology and Culture* 56.3 (2015), pp. 785–787.
- [22] John D Gould and Clayton Lewis. “Designing for usability: key principles and what designers think”. In: *Communications of the ACM* 28.3 (1985), pp. 300–311.
- [23] John D Gould and Clayton Lewis. “User Study Guidelines”. In: *Haptimap EU project* (2009).
- [24] JoAnn T Hackos. *Handbook of Usability Testing*. 1995.
- [25] Alan Cooper, Robert Reimann, David Cronin and Christopher Noessel. *About face: the essentials of interaction design*. John Wiley & Sons, 2014.
- [26] International Organization for Standardization. *Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, 2018*. URL: <https://www.iso.org/standard/77520.html>.
- [27] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing. 2. utg.* 2010.
- [28] International Organization for Standardization. *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems, 2019*. URL: <https://www.iso.org/standard/63500.html>.
- [29] Chauncey Wilson. *Brainstorming and beyond: a user-centered design method*. Newnes, 2013.
- [30] Bella Martin, Bruce Hanington and Bruce M Hanington. *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Pub, 2012.
- [31] Usability.gov. 2023. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
- [32] National Aeronautical and Space Administration. 2023. URL: <https://humansystems.arc.nasa.gov/groups/TLX/tlxpaperpencil.php>.
- [33] Alexander C Bock and Ulrich Frank. “Low-code platform”. In: *Business & Information Systems Engineering* 63 (2021), pp. 733–740.

- [34] Wikipedia Contributors. 2023. URL: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)).
- [35] Microsoft. 2023. URL: <https://visualstudio.microsoft.com/>.
- [36] Git. 2023. URL: <https://git-scm.com/>.
- [37] Unity Technologies. 2023. URL: <https://docs.unity3d.com/Packages/com.unity.xr.oculus@3.3/manual/index.html>.
- [38] Meta. 2023. URL: <https://developer.oculus.com/documentation/unity/unity-xr-plugin/>.
- [39] Meta. 2023. URL: <https://www.meta.com/en-gb/blog/quest/introducing-oculus-air-link-a-wireless-way-to-play-pc-vr-games-on-oculus-quest-2-plus-infinite-office-updates-support-for-120-hz-on-quest-2-and-more/>.
- [40] Wikipedia Contributors. 2023. URL: [https://en.wikipedia.org/wiki/Singleton\\_pattern](https://en.wikipedia.org/wiki/Singleton_pattern).
- [41] World Leaders in Research-Based User Experience. *Thinkingaloud*. 2023. URL: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>.
- [42] LinearB. 2023. URL: <https://linearb.io/blog/efficiency-in-software-engineering/>.

# Appendix A

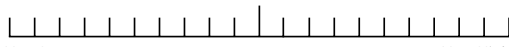
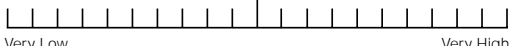
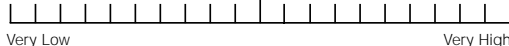
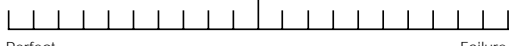
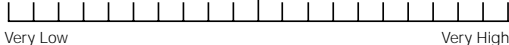
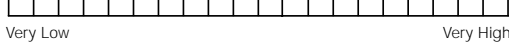
## Appendix A

### A.1 Tables, Scenario and NASA-TLX

**Figure 8.6**

**NASA Task Load Index**

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
Mental Demand	How mentally demanding was the task?	
Very Low		Very High
Physical Demand	How physically demanding was the task?	
Very Low		Very High
Temporal Demand	How hurried or rushed was the pace of the task?	
Very Low		Very High
Performance	How successful were you in accomplishing what you were asked to do?	
Perfect		Failure
Effort	How hard did you have to work to accomplish your level of performance?	
Very Low		Very High
Frustration	How insecure, discouraged, irritated, stressed, and annoyed were you?	
Very Low		Very High



System Usability Scale (SUS) Data													
Variables	Best case scores	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Variance	
use frequently	1	1	1	1	1	1	1	1	1	1	1	0.1	
too complex	-2	0	-1	0	0	0	0	0	0	0	0	0.1	
easy to use	2	2	2	2	2	2	2	2	2	2	2	0.4	
need support	-2	0	0	-2	-1	0	0	0	0	0	0	0.4555555556	
well integrated	1	1	0.5	1	0.5	0.5	0.5	0.5	0	1	0.5	0.1	
inconsistent	-1	0	0	0	0	0	0	0	0	0	-1	0.1	
learn quickly	2	2	2	2	2	2	2	2	2	2	2	0	
cumbersome	-2	0	0	-1	0	0	0	0	0	0	0	0.1	
confident	1	1	1	1	1	1	1	1	1	1	1	0.2333333333	
learn a lot	-2	0	0	-1	-1	0	0	-1	0	0	-1	0.2666666667	
<b>-9 to 7</b>		<b>7</b>	<b>6.5</b>	<b>4.5</b>	<b>-1</b>	<b>6.5</b>	<b>6.5</b>	<b>6.5</b>	<b>6</b>	<b>6</b>	<b>3.5</b>	<b>5</b>	
Age	<35	<35	>35	>35	<35	>35	<35	>35	<35	<35	>35		
Gender	m	m	m	m	m	m	f	m	f	f	f		
Exp in games (0-3)	3	2	3	0	1	3	0	0	2	0	0		
VR (0-3)	3	3	2	2	3	1	1	1	2	1	0		
NASA Task Load Index (TLX) Form Data													
Scales	Score Range	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Average	Variance
mental demand	2	3	5	6	6	6	4	5	4	4	4	3.8	1.066666667
physical demand	6	5	6	6	6	6	4	5	4	6	5	5.3	0.6777777778
temporal demand	6	4	6	6	4	6	4	5	4	5	5	4.9	0.7666666667
performance	0	3	0	0	5	1	4	4	0	3	3	2.3	3.566666667
effort	5	3	6	6	3	6	6	1	1	4	4	3.8	3.733333333
frustration	1	6	5	2	2	6	5	5	6	3	5	4.4	3.155555556
<b>0-36</b>		<b>20</b>	<b>24</b>	<b>28</b>	<b>23</b>	<b>30</b>	<b>26</b>	<b>25</b>	<b>19</b>	<b>25</b>	<b>25</b>	<b>24.5</b>	

55  
Table A.1 User testing data

Hello dear participant,

You have just seen an interactive content solution on your way up here. Isn't it cool? Now please imagine yourself working in a team that creates or distributes these interactive experiences. You and your teammates have just made a set of experiences under the requirement of your client who's organizing a big event in New York in which your work would be shown to the public. You want to make sure that the experience itself looks good, has an accurate and comfortable scale, the buttons are working in a reachable position, etc. However, the site has not even been built yet and setting up an exact same solution takes time and effort. You decided to use our visualizer to help preview the product that you are about to deliver.

The finished experience is stored in 2 files named "**wall.tap**" and "**floor.tap**" which run together to make a wall and floor setup, you ought to find and load them both into the program. (You should be able to find them on the **Desktop**.) One of them is a headphone commercial with buttons that could change the colour of the headphone displayed and a button to play a video; the other one tracks your position and displays cool content when you walk by.

Take a look around when you have loaded the experience. Maybe try out the buttons? (Could you find them, are they reachable?)

You would then want to show it to a colleague in a free-fly camera angle; try switching the camera mode to **free-fly camera** position.

You would then like to take a closer look at the experience as if you are the one actually interacting with it. Try activating **the VR option** and take a tour inside.

**VR Controls :**

1. **Walk freely inside using your left thumbstick or walk naturally in the experience (watch out for the red/blue boundary grid and do not pass it)**
2. **extend your hand and touch the content to interact with it.**

Perhaps you could test out something that your team missed during development! Tell us if you find anything that could be improved!

## A.2 User testing Scenario