NTNU
Norwegian University of
Science and Technology

Room-Scale Virtual Reality for Interprofessional Collaborative Healthcare Training

Developing a Multi-Interface Application for a Smart Virtual University Hospital

Henrik Trandem

Master of Science in Computer Science

Submission date:	June 2017
Supervisor:	Frank Lindseth, IDI
Co-supervisor:	Ekaterina Prasolova-Førland, IPL Aslak Steinsbekk, ISM

Norwegian University of Science and Technology
Department of Computer Science

Henrik Trandem

Room-Scale Virtual Reality for Interprofessional Collaborative Healthcare Training

Developing a Multi-Interface Application for a Smart Virtual University Hospital

Master's thesis in Computer Science

Trondheim, June 2017

Supervisors: Frank Lindseth, Ekaterina Prasolova-Førland, and Aslak Steinsbekk

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

This master's thesis investigates how room-scale virtual reality can be combined with traditional interfaces to facilitate collaborative interprofessional healthcare training. Students from medical- and nursing professions rarely get to practice interprofessional training and to deal with this issue; collaborative virtual environments are used to simulate patient care, procedural training, and collaborative training. Many of the existing solutions are outdated, however, and room-scale virtual reality could be just what the users need to improve the existing solutions. However, room-scale solutions are not very common today, and as such, other interfaces also need to be supported. A key research point has therefore been how different devices and interfaces need to adapt for them to be included into a smart university hospital. As the widely-used architectures utilized today do not support room-scale VR, a prototype application with a new architecture has been developed from the ground up, using the Unity game engine. The requirements of the application have been based on a role-playing case from an existing solution. The prototype and the existing solution were tested by the same group of people, and the results showed that room-scale virtual reality increased realism and immersion for the users. Additionally, supporting several interfaces in the same environment proved to be a challenging task. Finally, the thesis concludes that a multi-interface application used for interprofessional collaborative healthcare training can be developed in a few months, given that not all interaction methods provide the same functionality. It also seems evident that room-scale virtual reality makes procedural tasks much more lifelike.

Keywords: Room-scale, virtual reality, collaborative training, healthcare training, development, prototyping, multi-interface, adaptation, smart virtual university hospital

Acknowledgements

Firstly, I would like to thank my family - Heidi, Øivind, Camilla, Olivia, and Dennis - who have supported me in all thinkable and unthinkable ways over the years. The same goes for all my friends – you are awesome.

I would also like to thank all my teachers and lecturers at Tanum, Rykkinn, Valler, and NTNU, especially Jostein, Richard, Robert, and Gry, who have kept me motivated over the years. Alf-Inge and EiT-Egil at NTNU also deserve extra thanks.

My supervisors Frank, Aslak, and Ekaterina, deserve a great thank for guiding me through this project. Aslak has truly helped me understand the requirements and needs of the end users, and Ekaterina has provided me with invaluable feedback during my work.

All those who have assisted me during development and testing of the prototype have been a huge asset, and special thanks to Antti, Knut Erik, Torkjel, Kristian, and Carina for helping me thoroughly test the final prototype.

Thanks to the "Epic VR" group chat, which has always kept me updated on the latest in virtual reality.

Finally, a special thanks to Carina who has taken such good care of me, Cosmos, and Midna.

Table of contents

List of figures

List of tables

1 Introduction

“Virtual reality was once the dream of science fiction. But the internet was also once a dream, and so were computers and smartphones. The future is coming and we have a chance to build it together.” – Mark Zuckerberg on acquiring Oculus VR in 2014 [1].

Ever since the release of the first Oculus Rift in 2013, the interest for virtual reality (VR) has skyrocketed. Subsequently, numerous new actors have emerged, and today VR is used for virtually anything. With the introduction of room-scale virtual reality, immersion within a virtual world has truly risen to a whole new level as the technology can track a user’s movement and mirror it virtually. It is still up for debate what the main usages of VR will be in the near future, but it has undoubtedly had and will continue having a great impact on education.

1.1 Project Background

Healthcare education is a challenging task. It usually revolves around one single notable element – the patient. The fact that the patient is the common denominator within medical- and nursing education results in a series of challenges. The following subsection explains some of the identified challenges concerning healthcare education.

1.1.1 Challenges in Healthcare Education

One identified challenge is that students need to be able to interact with a patient somehow. As the number of students attending medical studies increases and as the length of a patient’s stay wants to be kept at a minimum, the availability of time for interacting with patients in real-life situations is low [2]. Secondly, following a patient over time is difficult, as when one moves to a different unit, one usually interacts with a different patient [2]. Thirdly, team-based training is becoming increasingly prominent in most disciplines. Like in most fields today, interaction and collaboration are done in interprofessional teams, and in health care, this practice is known as inter-professional education (IPE). IPE is widely considered a central competency by the practice field [3], but in most universities, it is not utilized due to the challenges mentioned above. These challenges often result in medical students and nursing students not being able to practice together during their studies, and when they have finally finished their education, they have no experience in working with each other or other professionals [2].

1.1.2 Answering These Challenges

The challenges mentioned above could be approached in many ways, and the most obvious approach is through technology. Unsurprisingly, technology is already employed in medical education around the world and includes approaches like computer-assisted learning, mobile devices, games, wearables, and, most importantly, simulations [4].

Simulations usually happen through medical simulators (mainly used for procedural training) or virtual environments. For interprofessional training, a Collaborative Virtual Environment (CVE) can be utilized. Snowdown et al. define a CVE as a computer-based, distributed, virtual space where people can meet and interact with others, with agents or with virtual artifacts [5]. Thus, by employing a CVE in a medical setting, one could simulate patient interaction and IPE in a virtual world and ultimately solve the problems discussed in the previous section.

An example of such a solution is a CVE concept developed as a collaboration between the Norwegian University of Science and Technology (NTNU) and St. Olavs University Hospital in Trondheim called VirSam [6]. VirSam is a direct answer to the challenges mentioned above and allows 4th year medical students and 3rd year nursing students to practice cross-professional collaboration to prepare for clinical practice in a virtual world [6].

1.2 Project Motivation

1.2.1 Possibilities of Room-Scale Virtual Reality

Up until recently, most virtual reality experiences have been seated ones. The term "room-scale" does not actually have a technical definition. It was a term introduced by actors within the VR field to describe how the seated VR experiences could *scale* to an entire room, meaning that users could stand up and use their body to move around in a virtual world [7]. Even though the term's original meaning refers to the scaling of rooms, it is most frequently used to describe a *mapping between reality and virtual reality,* which will be the way it is utilized in this thesis. Since room-scale technology has only been available to consumers for a couple of years and since the price has not dropped substantially, it is often too expensive for many consumers. The three most widely used room-scale platforms (HTC Vive, Oculus Rift + Touch, PlayStation VR) only track the users' head and hands. Tracking only three points may not seem impressive, but since the rotation in addition to the position of the devices is tracked accurately, this has turned out to be adequate in providing realistic and immersive experiences [9].

Room-scale allows for more natural interaction with the environment, as well as with other users. By using the body, a user can manipulate virtual objects in a realistic and precise way, opening possibilities for simulating procedural activities in virtual worlds. Even though room-scale VR has only been available to the general public for about a year, several examples of room-scale VR facilitating surgical- and procedural training have already emerged [10][11].

These solutions can replicate advanced hardware and medical machinery for less than 1% of the actual cost [12]. Most of these solutions are offline, however, and do not include any way in which the user can collaborate with other professionals.

1.2.2 Limitations of Existing Collaborative Virtual Environments

Over the years, numerous examples of collaborative healthcare activities set in virtual worlds have emerged, most of which are set in a virtual world called Second Life [13] ([14][15][16][17]). Prasolova-Førland et al. (2016) have already developed and tested several solutions for medical education and collaboration in Second Life [2], one of which turned into VirSam mentioned in the previous section. VirSam's participants state that it provides a rich and immersive learning experience, good practice for communication and is fun. However, they also say that it suffers from many technical issues and that it is not user-friendly [2]. Another limitation with VirSam in Second Life is that the only way interaction with the environment can happen is by pointing and clicking with a mouse, which is not very natural. Looking back at the room-scale VR solutions mentioned earlier, these provide natural interaction with the environment but feature no collaborative mechanisms.

1.2.3 Smart Education and Adaptation of Interfaces

The subtitle of the thesis implies that a *multi-interface* application should be developed and that the resulting application should be used in a *smart virtual university hospital.* The term smart virtual university hospital derives from a broader term called *smart university,* which *"involves a comprehensive modernization of all education processes"* (Tikhomirov & Dneprovskaya, 2015). A smart virtual university hospital, therefore, wishes to modernize the education process related to medical teachings in a virtual world. The question is then how this modernization can be realized. Uskov et al. (2016) list several *smartness levels* that can be found in a smart university, one of which being *adaptation* [20]. Adaptation essentially means to adapt or customize content to support every user. This includes both content and technological platforms. Prasolova-Førland et al. (2016) list several important *features* needed in a smart virtual university hospital and maps them to the different smartness levels discussed by Uskov et al. [2]. The feature *personalized interface and interaction* explains that different *interfaces* like VR interfaces can increase the realism of interacting with the environment, other characters, and equipment and that the identified smartness level is adaptation [2]. This feature brings up the second part of the thesis subtitle, in that the application should support different interfaces. An interface in computer science can mean several things [18]. In the context of this thesis, the users are vital, and as such an interface will mean something a user can utilize to interact with the elements of the application. This could mean anything ranging from a virtual reality head-mounted display (HMD) to an input controller, to the user-interface (UI) that a user sees.

1.2.4 Getting the Best of Both Worlds

Seeing that CVEs are an integral part of medical simulation and noticing the possibilities that room-scale VR can provide, an opportunity to modernize healthcare education arises. This is the main motivation behind the project.

1.3 Research Goal

The goal of this thesis is to investigate *how room-scale virtual reality can be combined with traditional interfaces to facilitate collaborative interprofessional healthcare training.*

Since the mentioned VirSam's purpose is to let nursing- and medical students practice interprofessional education, it is an excellent starting point for exploring the thesis goal. However, because VirSam is played in Second Life, it is limited by the architecture and functionality that Second Life provides, and as Second Life does not support room-scale VR, a *new software platform* is needed [2].

Because of this, the thesis goal will be researched through *prototyping* of a new *application.* The prototype application will be created using a *game engine*, and *one of the two cases* developed for VirSam will be the basis for said prototype. How the case works and is played out is explained in detail in section 3.1 in the requirements specification.

1.4 Research Questions

To sum up the research goal and structure the exploration process, four research questions (RQs) are introduced below.

RQ1 How can room-scale virtual reality improve upon interprofessional collaborative training by enhancing procedural elements?

In healthcare, it is important to communicate in a clear way both with the patient, relatives of the patient, and other professionals while performing certain tasks [21]. Medical- and nursing students have little time working together when out in practice, and to simulate a realistic environment, procedural- and collaborative training need to be practiced simultaneously. Room-scale VR allows for much more realistic interaction with objects and other users compared to mouse and keyboard. Most consumer room-scale VR hardware today have built-in microphones and audio [22] and multiplayer capabilities [23], meaning that there, in theory, is nothing in the way of setting up a collaborative training environment that also contains procedural elements.

RQ2 What are the requirements of a multi-interface virtual reality application used for interprofessional training, and how can such an application be created making use of a modern game engine?

As explained earlier, there is a need for a new software platform as the current platform (Second Life) in which VirSam takes place has several problems (discussed further in section 2.1), and does not support room-scale VR. For one developer to be able to create a cross-platform multiplayer VR application in a few months, a game engine is needed to speed up the development process. The game engine needs to promote rapid prototyping, and more specifically rapid application development (RAD) [24]. RAD means different ideas can be tested in a short amount of time, which is essential when there are so many different interfaces (and user interfaces) that could and should be supported in the application [24]. Also, with only about 4 months available for development, the *feasibility* of adding certain software elements must be discovered quickly and continuously [24].

Both the requirements of the game engine and the requirements of the prototype application will be detailed. To survey the requirements of the application, a framework called the *user-place-artifact* framework (introduced in 3.3 and discussed further in Chapter 6), will be used.

RQ3 How does the solution facilitate *smart adaptation*, and what are the challenges concerning virtual simulation from different types of devices?

With the Internet, Internet of Things (IoT), Ubiquitous Computing, VR, AR, and technological advances we have seen over the past decade, the possibilities of educational activities promoted by technology seem endless. As new interfaces emerge every day, the task of supporting them all becomes increasingly challenging. This challenge is the core of the discussion in RQ3.

RQ2 states that the resulting prototype should be *multi-interface*. The challenge then arises as to how one can *adapt* the application to support different interfaces *smartly*, as discussed in section 1.2.3. For the scope of this project, three different playstyles will be implemented: mouse and keyboard, Xbox-controller with HMD, and room-scale with Oculus Touch and HTC Vive. These playstyles all offer different ways of moving a virtual character, and different ways of interacting with objects. Object interaction is an especially difficult topic concerning adaptation and is typically something that needs to change depending on what interface is used.

Due to the limited time available, mobile and mobile VR solutions will not be implemented but will be part of the literary research and discussion. To further understand the challenges of how interfaces, devices, and virtual places need to adapt in a VR application, section 2.8 presents some of the most widely used VR interfaces.

Note that this thesis uses a term called "desktop" quite prevalently. This term is used to describe the interaction method "mouse and keyboard" used in combination with a regular 2D computer screen (i.e. no VR is used), to increase readability. The term includes laptops, desktop computers, and workstations alike.

RQ4 In what way does the presented solution compare in terms of user experience to existing solutions?

Second Life was released in 2003, and much like the Mercedes-Benz W210 which ceased production that same year, the product has become rusty over the past 15 years. With problems concerning stability and usability, the users of VirSam spend too much time fiddling with sound, technical issues, as well as learning how to use the software, resulting in minimal time devoted to playing through the two cases of VirSam. Second Life is rich in functionality, which certainly can be a good thing, but for a user who will most likely only use the application once or twice, it is simply too much. In fact, the users need to watch a 13-minute long instructional video on how to log in and control the avatar [25]. Furthermore, Second Life has little to no way of moving or manipulating objects, which is essential if one is to achieve an adequate grade of manipulating procedural elements. All these factors illustrate a need for a new architecture to realize RQ1. Thus, the architecture of the application is also explained, as a proper architecture is necessary to make the application both user-friendly and developer-friendly.

1.5 Thesis Structure

Chapter 1 – Introduction

The first chapter introduces the project background, as well as the motivation behind the project. It also details the research questions that will be looked further into throughout this thesis.

Chapter 2 – Theory and Related Work

The second chapter details the theory and related work. Theory related to the research questions introduced in section 1.4 are critical areas of research and include subjects like Virtual University Hospital, adaptation of interfaces, VR in educational entertainment, collaborative medical training, procedural training, VR devices, and game engines.

Chapter 3 – Requirements Specification

The second chapter describes the geriatrics case that will be employed in the implementation of the new prototype. It also gives an overview of how testing was conducted throughout the course of the project, and finally, lists the requirements of the final application prototype.

Chapter 4 – Architecture

The fourth chapter details the architecture of the application. Firstly, it lists the most important stakeholders, followed by some factors that have driven the architecture in the direction it has taken. Finally, it explains some of the tactics used to facilitate the architectural drivers.

Chapter 5 – Prototyping

The fifth chapter shows how the chosen game engine, including frameworks, assets, and toolkits, were used to realize the requirements of the application.

Chapter 6 – Results

The sixth chapter explains how the requirements of Chapter 3 are realized in the final prototype. It also details the results of testing the final prototype. For the final user test, the case presented in Section 3.1 is played by a group of people, both in the new prototype and in Second Life. Finally, the results of the two tests are shown.

Chapter 7 – Discussion

The seventh chapter discusses the research questions based on test findings, theory, and related work.

Chapter 8 – Conclusion

The eighth chapter sums up the results and discussion and concludes the thesis. Future work and how others can build upon the research of this thesis is also explained.

2 Theory and Related Work

As explained in the introduction, this project seeks to research how room-scale virtual reality can facilitate interprofessional healthcare education as well as the development of a multi-interface application for a smart virtual university hospital. To understand the challenges and possibilities of these tasks, one must know what a virtual university hospital (VUH) is and what kind of educational activities it supports (section 2.1). Section 2.2 discusses how adaptation is important in a smart virtual university hospital. To understand why one would want to introduce room-scale VR in healthcare education, section 2.3 explains why entertainment and VR can be powerful learning tools. Further, to address some of the usability issues of Second Life, section 2.4 presents a concept known as *game flow*. Sections 2.5 and 2.6 detail procedural- and collaborative training respectively, as these activities, should be practiced simultaneously in a healthcare setting. Section 2.7 presents a framework for identifying requirements of the mentioned prototype, and section 2.8 discusses different VR interfaces that can be used in a smart VUH. Finally, sections 2.9, 2.10, and 2.11 discuss the requirements of the game engine needed to realize the prototype application and also which game engine was selected for this project.

2.1 Virtual University Hospital

Prasolova-Førland et al. (2016) have for several years worked on a holistic system in the form of a Virtual University Hospital (VUH) for tackling the challenges mentioned in 1.1.1 [2]. During this work, they have surveyed the different medical *educational activities* that can take place in a VUH. Table 1 lists the most important identified activities in a VUH.

Table 1: Important Educational Activities in a Virtual University Hospital.

Activity	**Content, facilities, and technological solutions**
Patient simulation	Operating room, patient ward, emergency area, virtual humans, interactive hospital equipment, VR interface
Procedure training	Various hospital departments (operating room, patient ward, emergency area), information using videos and posters, interactive hospital equipment, VR interface
Lectures, e.g. anatomy	Classrooms, lecture halls, 3D interactive models of organs and the human body, posters, videos, VR interface

Role-plays (team training, patient communication)	Operating room, patient ward, emergency area, reception/outpatient clinics, interactive hospital equipment, VR interface

The VUH is part of a virtual campus set inside the world of Second Life [26]. Some different projects have been developed or researched for the VUH over the past couple of years [27][28][29] and facilitate several of the educational activities listed in Table 1. VirSam mentioned in the introduction is the latest addition to the VUH, and as stated earlier, VirSam’s primary objective is to provide fourth year medical and third year nursing students with the possibility of interprofessional communication training in a smart virtual environment. This means that *role-play* (i.e. team- and communication training) is the main identified *activity* of VirSam. Furthermore, VirSam also contains elements from the *patient simulation* and *procedure training* activities as the two cases are set in a hospital environment, featuring various hospital equipment. This becomes clearer in the requirements specification (Chapter 3).

2.2 Smart Adaptation

For the software system to facilitate *smart adaptation*, it needs to support different devices and adapt and customize each different device based on what the requirements of the application are. As a device is limited by its hardware, only so much can be done on the device or interface regarding adaptation. This means that the virtual environment, including *artifacts* and *places,* may need to change to support the conglomerate of interfaces in existence.

Section 1.2.3 mentions how *adaptation* is key to supporting multiple interfaces and interaction methods in a smart VUH. This is the most essential identified *feature* for the scope of this project, but Prasolova-Førland et al. (2016) also discuss other areas of a smart virtual university hospital where adaptation is necessary [2]. The additional features they list relevant to this project are *“scenario building,” “personalized learning path,”* “*individual feedback,” “virtual agents,” “general hardware support,”* and finally *“integration: portal, scheduling, and run-time systems.* Not all these will a part of this project scope, however, meaning that some are more relevant for the future of the application.

2.3 Virtual Reality and Entertainment for Education

Educational entertainment (edutainment) is the notion of using entertainment to facilitate learning. Gee, J., P. (2003) argues that schools, workplaces, and families can use games and game technologies to enhance learning [30]. The idea is that when learning through games, one becomes engaged, motivated and immersed, resulting in a better learning outcome.

For VR in particular, a case study on the impact of VR on academic performance was conducted by Beijing Bluefocus E-Commerce Co., Ltd. In 2016 [31] In this study, a group of examinees learns about celestial physics for thirty minutes. Half the group uses traditional learning methods, and the other half uses VR physics teaching, both groups guided by the same teacher to avoid deviation in results caused by the professional difference among teachers. The two halves then take the same test, and the scores were 27.4% higher for the examinees using VR. The pass rate also skyrocketed from 40% to 90%. The study also conducted a retention test, where the two halves would do the same test two weeks later. Now the scores were 32.4% higher. The higher scores could indicate that VR improves knowledge retention, allowing the students to keep the information in long-term memory easier.

2.4 Game Flow

When role-playing in a CVE, the software typically contains all or most elements found in a video game as CVEs allow users to interact and collaborate inside virtual worlds. *Flow* is a somewhat abstract term that describes how gratifying a task can be to a person. Sweetser, P., & Wyeth, P. (2005) looked at psychologist Csikszentmihalyi's flow model [32], and how it could be adapted to video game development [33]. Csikszentmihalyi (1990) states that flow is an experience so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous [34]. Sweetser, P., & Wyeth, P. (2005) proposed the game flow model based on the flow model, which are eight points that need to be considered when wanting to promote flow in a video game (shown in Table 2).

Table 2: Flow table. The right column explains what type of flow is needed for facilitating important video game concepts.

Games Literature	**Flow**
The Game	A task that can be completed
Concentration	Ability to concentrate on the task
Challenge Player Skills	Perceived skills should match challenges, and both must exceed a certain threshold
Control	Allowed to exercise a sense of control over actions
Clear Goals	The task has clear goals
Feedback	The task provides immediate feedback

Immersion	Deep but effortless involvement, reduced concern for self and sense of time
Social Interaction	n/a (this breaks flow, but is still important)

If one were to analyze any arbitrary modern computer game, one would most certainly find all these eight categories realized to a certain extent. With the introduction of room-scale VR, this thesis seeks to improve upon some of the flow elements found in VirSam and Second Life. Room-scale VR increases *immersion* immensely and could potentially allow the user to feel a greater sense of *control* of the avatar. The geriatrics case in VirSam consist of four different scenes, but this information is unavailable inside Second Life and must either be printed or found in an internet browser outside Second Life. This problem means *clear goals* are hard to find, which again causes a negative ripple effect for both *concentration* and immersion. RQ1 wants to enhance procedural elements in a collaborative environment, which challenges the user's concentration and professional *skills*.

Social Interaction is probably the primary flow element in a CVE. Sweetser, P., & Wyeth, P. (2005) states that social interaction breaks flow, but that it is still a vital part of game flow. In the case of a CVE with role-playing though, it does the exact opposite of breaking flow. In a video game used for entertainment, the player might use social interaction to talk about the car he or she bought last week, which obviously breaks flow if one is playing as an archer in a game set in medieval times. However, in VirSam's case, social interaction is used to convey information about the avatar's role and actions to other role-players through movement, animations, and voice communication, as discussed by Kleven N. (2014) [35].

Looking at the game flow model, as well as the functionality and needs of VirSam, it is apparent that if this model is taken into considering during development, it can provide some useful guidelines to what elements need to be present to ensure a good user experience, which is the core of RQ4.

2.5 Procedural Training, Surgery Simulators, and VR

Even though the word "procedure" typically is connected with complex operations performed by surgeons, the word *procedural* is defined within medicine as:

> *Relating to or comprising memory or knowledge concerned with how to manipulate symbols, concepts, and rules to accomplish a task or solve a problem [36].*

More simply put, “procedural” means doing things to achieve a goal, which is the way the term “procedural” is used throughout this thesis. Hence, “procedural” can also include simpler tasks such as providing pre- and post-operation care, measuring blood pressure, and helping a patient stand up or handing over a glass of water.

Surgery simulators were first introduced in the 1980s. In the late 90s simulators good enough to be able to simulate a realistic situation emerged. An example of such a simulator is the da Vinci Surgical System [37]. This system allows surgeons or students to practice operations with a physical device. However, complex physical simulators like these are often very expensive. As stated briefly in the introduction, VR allows for learning in areas where practical experiments can be costly or dangerous – especially medical experiments involving expensive machinery or equipment [12]. Room-scale VR can enable medical students to perform complex procedures repeatedly without having to purchase any form of physical ragdoll or equipment for the experiment. It can also potentially allow students and teachers to perform procedures together from different geographical locations. With the introduction of VR, most, if not all the functionality of a surgery simulator could be implemented with minimal cost and higher modifiability.

Ever since the release of HTC Vive in 2016, which embodies the user in a virtual world in a very realistic and accurate way, several VR medical simulators have been showcased. Osso VR [10] is an excellent example of how VR can realize surgical skill training. Virtual Medical Coaching (VMC) [11] in New Zealand recently showcased the “VMC Suite’s X-Ray Trainer, ” which provides medical imaging education in VR and features comprehensive equipment and lifelike patients for simulating taking X-Rays [12] [38]. VMC state that the hardware used costs less than 1% of real-world x-ray machine counterparts [12] – a point that truly illustrates the potential of VR.

2.6 Collaborative Training

Some of the most common competencies in medical team training include cooperation, leadership, coordination, awareness, communication, situational awareness, leadership, and situation monitoring (Healey, AN., Undre, S., & Vincent, CA., 2004) [39]. Several of these competencies can be identified in Baker et al. (2005), which among other things discuss classroom-based training programs [40].

Mark J. W. Lee (2009) states that well-designed learning interventions using virtual world environments can manifest the key ingredients or elements of collaborative learning [41]. 3D

CVEs are not without problems, however, and in some cases, the features of a 3D environment or world may work to the educator's disadvantage and function as a distraction, in the end resulting in discouraging students from attending to the key conceptual tasks in a collaborative learning activity [41]. These distractions illustrate a need to guide students to where the action takes place, and not include too many features or vast areas with lots of functionality.

From 2.3, it is evident that VR can have a positive learning effect if done in the right way. Prasolova-Førland et al. (2016) state that:

> *The use of VR technologies goes beyond procedural skills. Possibilities for synchronous communication and interaction allow using these technologies, at the moment mostly with the desktop interface, for various collaborative learning approaches, as well as facilitate situated learning and project-based learning approaches. In addition, virtual learning environments can also support interprofessional and distributed medical teams working together on complex cases [2].*

This statement shows that VR could be used for collaborative- and communication training in addition to procedural training mentioned in Section 2.5. However, minimal research exists on the topic of combining these activities. Meetings with stakeholders have revealed that that body- and hand movement, as well as procedural training, is a natural part of a nurse's or doctor's everyday activities and that tasks like these should be practiced alongside collaborative training. RQ1 was introduced to discover how the activities mentioned above can be combined.

2.7 Identifying Requirements of a Collaborative Virtual Environment (CVE)

Having a way of characterizing a CVE in terms of its users, environment, and artifacts, is important, as it allows for easier understanding the requirements of the software. Prasolova-Førland (2004) suggests a framework for characterizing a CVE [42] that does not revolve around the technology used, but rather around the activity done at a certain time. The framework is inspired by the Activity Centered Model (Gifford & Enyedy, 1999) [43] and Activity theory (Engeström, 1987; Vygotskij, 1978) [44][45]. With this framework, the activity of a user is in focus. As an activity takes *place* in a virtual environment with other *users*, and with interactive *artifacts*, the framework will be referred to as *user-place-artifact*. This thesis will use the *user-place-artifact* framework as a means of identifying and characterizing the requirements of the final prototype, helping realizing RQ2.

Note that an artifact discussed in this thesis is used the way it is defined within the Activity Theory and Human-Computer Interaction (Kaptelinin) [46] and that it does not refer to digital- or visual artifacts regarding unintended alteration of data in a digital process.

2.8 Virtual Reality Hardware

To further understand RQ3, different viable VR interfaces should be studied. This section discusses different VR hardware that potentially could be used in an SVUH.

2.8.1 Oculus Rift

Oculus Rift is now in its third iteration (CV1) and was the HMD that kicked the VR train into gear with the release of the DK1 in 2013. The success of the Rift resulted in Facebook ultimately acquiring Oculus VR, and Oculus is today arguably the biggest driving force of VR. The Rift has High Definition (HD) resolution and a refresh rate of 90Hz. In fall 2016, Oculus released Touch, which are tracked hand controllers that enable room-scale VR for the user. The Rift also ships with an Xbox One controller and a small remote called Oculus Remote, meaning that Oculus Rift supports different input methods. With three interaction interfaces already identified, and from just one manufacturer, the challenge to support different interfaces from several manufacturers in one application can already be recognized (RQ3).

2.8.2 HTC Vive

HTC Vive is the main competitor of the Oculus Rift. It has a lot of the same functionality and very similar specifications. Unlike the Rift, it comes with room scale VR out of the box, featuring two wall-mounted cameras and two controllers in addition to the HMD. The advantage with wall-mounted cameras as opposed to the desktop sensors user by Rift is that the cameras rarely lose tracking of the controllers. Like the Rift, HTC Vive is expensive, in addition to needing a powerful gaming computer, which means that even though they are both available to consumers, they have not reached their full market potential yet. HTC Vive was the first consumer product to feature well-working room-scale virtual reality, and room-scale VR seems to be slowly taking over for seated experiences as nausea is reduced and immersion is increased.

2.8.3 Sony PlayStation VR

Sony PlayStation VR is the newest addition to the wired VR HMD family. Although PlayStation is primarily a gaming console meant for entertainment, there is no reason why it could not be used for educational purposes as well. In less than a year, PlayStation VR has sold more than one million units [47], meaning it has sold much faster than Oculus Rift and HTC

Vive. The main reason for this is that more people own a PlayStation 4 than a powerful gaming computer, and also that it sells at a lower price than its competitors.

2.8.4 Google Cardboard

One of the simplest and cheapest HMD solutions out on the market today is the Google Cardboard. It is simply a cardboard box with a pair of powerful lenses, where the user straps its phone on the backside of the lenses, straps the box to its head, and the phone splits the screen output into two smaller frames. What separates this from the wired HMDs is that the phone does all the processing, meaning the solution is wireless, so one can move around while using them. The Google Cardboard costs no more than 100 NOK (about 10 EUR), meaning the investment primarily lies in the smartphone. Because virtually everyone has a smartphone these days, this is an excellent way to experience VR for a low price. Gamepads can be connected to the phone via Bluetooth if one wants to interact with the environment in an application.

2.8.5 Samsung Gear VR

Samsung Gear VR is a cooperation between Oculus and Samsung. In 2013, Oculus started a cooperation with Samsung, wanting to make a portable and cheaper version of the Rift. In 2015, the Samsung Gear VR was released, which is essentially a fancier Google Cardboard. Unlike the Cardboard, the Gear VR has a plastic casing with a built-in touch panel on the side, which enables user interaction. For simple applications, the touch panel might be adequate, but for bigger applications, a Bluetooth gamepad can be used.

The Samsung Gear VR can only fit new Samsung phones into it. This limitation means that one is confident that the hardware of the phone is adequate to run new applications, but it also means one may need to buy a new phone if one does not already own a Samsung phone. It costs nearly ten times more than Google Cardboard, but the build quality is also much higher, and many people feel that it is worth the higher price.

2.8.6 Google Daydream View

Google Daydream View is Google’s answer to the Samsung Gear VR. Unlike Gear VR, it aims to support most Android phones, though only newer phones are supported. Being made of fabric instead of plastic, the Daydream View it is also a bit lighter than the Gear VR.

2.8.7 Additional Interfaces

In addition to the HMDs and controllers mentioned in the previous sections several “alternative” interfaces exist. To capture body movement, motion capturing devices like Leap

Motion and Kinect can be used. For increased immersion and feedback, haptic feedback devices like Araig Haptic and Teslasuit give the user full-body haptic feedback. Furthermore, as the room-scale solutions normally are wired, they are limited by the length of the wire and tracking distance of their respective sensors. To not limit users regarding space, one can use a treadmill like the Virtuix Omni to simulate leg movement, when in reality the user is standing still. VR Backpacks like the MSI VR One can also remove the limitations of the wired HMD cables by allowing the users to carry a computer on their back.

All in all, there are numerous different VR interfaces, some more accessible to consumers than others. Supporting all of them is an impossible task, but making use of a game engine that supports the most widely used ones is essential if one is to start realizing some of the features part of a smart virtual university hospital mentioned in section 2.2. This is discussed further in the following section.

2.9 Game Engine - Software

Creating a networked 3D-application takes a lot of time and effort, and when adding VR to the stack, it will certainly be a task too grand for one person to create everything from scratch. This is where game engines can be helpful. Game engines are software frameworks purposely made for creating video games. The most advanced game engines include tools for handling rendering of graphics, collision detection, physics, sound, scripting, AI, and networking. They also abstract away the low-level parts of the application, making it easier for a developer to start creating end-user content right away.

It is evident that an immersive VR application featuring multiplayer support, voice chat, modern graphics, and several different input and output devices will have a high number of requirements. As the game engine chosen for this project will be such an integral part of the development, it is of particular importance to make sure the game engine supports as many requirements of the application as possible

The rest of this section discusses the high-level functional requirements of the game engine needed to realize a working application. Since a pre-defined case (mentioned in section 3.1) was going to be implemented in the prototype, the requirements of the game engine were quickly discovered. Table 3 shows an overview of the game engine requirements (GE) for the application. Each category is further detailed below the table. The requirements having high priority will be valued higher when selecting an appropriate game engine for the application. Keep in mind that these requirements are initial high-level requirements for the game engine and that they do not make use of the *user-place-artifact* framework discussed in section 2.7. The *user-place-artifact* framework is used for the requirements of the final prototype and can be found in section 3.3.

Table 3: Game Engine Requirements

Requirement	ID	Description	Priority
Cross-platform	GE1	The engine should be able to deploy its application to different target systems without the user having to spend much time altering the code.	High
3D Graphics	GE2	The engine should provide tools for handling graphics and provide a relatively realistic look.	High
Animated 3D Characters	GE3	The engine should support 3D characters that can be controlled by a player.	High
Object Interaction	GE4	The engine should provide support for touching or clicking on objects to interact with them.	High
Virtual Reality	GE5	The engine should support Virtual Reality and some of the most widely used VR solutions like HTC Vive, Oculus Rift, Gear VR, and Cardboard. The engine should also support room scale VR, as it is essential for the project.	High
Networking	GE6	The engine should provide stable networking functionality and a minimum of six simultaneous users.	High
Voice Communication	GE7	The engine should provide voice communication functionality.	High
Rapid Prototyping	GE8	The engine should facilitate rapid prototyping so that trying	High

		out different approaches can be done quickly.	
Ease of Use	GE9	The engine should be easy to learn and use.	Medium
Modifiability	GE10	Making changes to existing code and deploying the application should be quick and easy to do.	Medium
Well-documented	GE11	The game engine should be well-documented so that when a developer encounters an issue, it is easy to figure out and fix.	Medium

2.9.1 GE1: Cross-platform

Cross-platform can mean two different things: to support different physical devices like desktop computers, phones, wired VR solutions and wireless VR solutions or to support different operating systems. In this thesis, cross-platform will include both meanings. For desktop, Windows has been the most dominant operating system for VR, and it was not until early June 2017 that Apple announced VR support for the HTC Vive [48]. Oculus is yet to announce MacOS support.

A problem when creating a room-scale VR application today is that not many people have access to this kind of equipment, as it is expensive, and requires much space to set up. Because of this, even though we have seen the enormous potential of VR in many different disciplines, the future of VR is not yet determined. In an interview, the CEO of Valve stated that *"We think VR is going great. It's going in a way that's consistent with our expectations"*, but he also said that *"We're also pretty comfortable with the idea that it will turn out to be a complete failure."* [49]. There are many different actors on the market with numerous VR solutions, and knowing which one to pick is never easy. This emphasizes the point that the future of VR is not yet determined, and with that in mind, it is important to create an easily modifiable application that can be used on different target platforms. Regular desktop PCs or laptops will most likely still be around for decades, and with the future of VR still undetermined, supporting regular mouse and keyboard interaction is important to reach a significant target audience.

Today, it is the wired VR solutions like HTC Vive and Oculus Rift that are considered the most immersive and advanced, but mobile phones have become increasingly powerful over the past couple of years, and are usually only a few years behind desktop computers when it comes to processing power. What is interesting is that even though the mobile solutions are less advanced than the wired ones, about 93 percent of HMDs in the US in 2016 were smartphone based [50].

2.9.2 GE2: 3D-graphics

The game engine needs to support relatively advanced 3D-graphics to improve upon VirSam graphically. 3D graphics is a must for virtual reality, as the user needs to be able to look around the 3D environment to get the profound sense of immersion that VR can provide. It is also essential that the graphics can be optimized for different VR solutions, as one needs a high refresh rate to avoid VR sickness. With realistic graphics and VR support, it is much easier to feel immersed and for a nurse or a doctor to relate to the patients and people in the application.

2.9.3 GE3: Animated 3D Characters

When playing as a room-scale character, one usually only wears an HMD and one controller in each hand. With only three tracking points, the system has no way of knowing where the rest of the user’s body is located. A result of this is that it is hard to animate a room-scale player’s avatar precisely. Room-scale avatars are therefore usually depicted as a floating head with two hands, sometimes with the upper torso showing as well [51].

When playing on a regular desktop PC, whole-body animated avatars are the norm, as the animations can be predefined. Conversations with stakeholders involved in the creation of VirSam have revealed that player animations are central in a medical setting, as providing emotional support to patients and relatives is an important everyday task performed by nurses. With animations, it is much easier for the user to relate to other avatars.

2.9.4 GE4: Object Interaction

An essential part of a medical training application is for the user to be able to interact with objects. This includes interacting with other players or Non-Player Controlled characters (NPCs), as well as picking up and using objects to perform certain tasks. If using room-scale VR technology, interaction is most likely done by physically picking them up using the VR controller. If one was to participate in the application without using VR, picking up objects could be done by clicking on them with the mouse. Object interaction is essential for realizing any kinds of procedural training (RQ1), making it an important GE for realizing the prototype application.

2.9.5 GE5: Virtual Reality

Virtual reality has been mentioned in the previous sub-sections and is not surprisingly an important part of this project. The game engine needs to support the most widely used VR solutions, both mobile and wired solutions. Mobile solutions include products such as the Google Cardboard, Samsung Gear VR, and Google Daydream View. The biggest wired solutions out today are Oculus Rift and HTC Vive. It is important that setting up VR for the project is a quick and easy task so that the creation of the end product can commence as quickly as possible.

2.9.6 GE6: Networking

For any developer having little experience concerning low-level networking implementations, it is vital that the game engine can provide some tool for making networking as facile as possible. Ideally, the networking solution should provide server hosting and matchmaking so that the developer does not have to write low-level networking code from scratch. This also makes server maintenance easier in the future.

2.9.7 GE7: Voice Communication

Looking at RQ1, voice communication is a must in a networked collaborative application. This should either be provided with the network functionality or added without having to alter the existing network code too extensively.

2.9.8 GE8: Rapid Prototyping

Another important factor is how the engine allows for rapid prototyping, as a big part of this project will be to test different approaches and interaction methods, getting feedback, and iterating upon those solutions based on the feedback provided.

2.9.9 GE9: Ease of Use

With a scope of only a few months, the developer has limited time learning new software or frameworks. If the game engine is easy to use or the developer is familiar with the software, more can be accomplished in less time.

2.9.10 GE10: Modifiability

The application made with the engine must also be easy to modify and maintain. Coming up with a solution that supports all state-of-the-art VR solutions with different interaction methods, as well as modern graphics and networking capabilities within the short time frame of this project is close to impossible. This means that the ability to modify existing software, as well as adding new software in the future is even more important than the state of the prototype when this thesis is concluded.

2.9.11 GE11: Well-documented

This requirement is connected to GE9: Ease of Use. If the game engine is well-documented, it is easier and quicker to discover solutions to problems. One can also often find solutions to problems in external fora like StackOverflow [52]. These fora can often be helpful as if the software has many users, chances are someone else has encountered the same problem.

2.10 Unity vs. Unreal

The game engine choice boiled down to two alternatives: Unity 3D, and Unreal Engine. These game engines have been around for many years, and both have nice graphics, support multiple platforms, and support VR. The table below shows a rough comparison of these game engines and how well they support each identified GE.

Table 4: Unity vs. Unreal

Game Engine Requirement	Unity 5	Unreal Engine 4
GE1: Cross-platform	High	High
GE2: 3D graphics	Medium	High
GE3: Animated 3D Characters	High	High
GE4: Object Interaction	High	High
GE5: Virtual Reality	High	High
GE6: Networking	High	Medium
GE7: Voice Communication	High	Medium
GE8: Rapid Prototyping	High	High
GE9: Ease of Use	High	Medium

GE10: Modifiability	High	High
GE11: Well-documented	High	Medium

The consensus is that Unreal Engine 4 is superior regarding graphics and that Unity 5 takes less time to learn and is better documented. What is not shown from Table 4 is that Unity's asset store is much bigger than Unreal's Marketplace. In the Unity asset store, a free networking solution that also includes voice communication was found that suited the project very well.

2.11 Chosen game engine – Unity

After extensive research and an informal chat explaining the requirements of the project to Alf Inge Wang – a Ph.D. in game development at NTNU [53], it was apparent that both game engines suited the requirements well, and that the choice mainly boiled down to personal preference. Having worked on several VR projects in Unity over the past year, Unity was chosen.

3 Requirements Specification

This chapter introduces the requirements specification of the final prototype. Firstly, the case that will be implemented is described. Secondly, the requirements specification of the final prototype is listed. Each requirement and how the final environment was set up is detailed in Chapter 6.

3.1 Case

This section introduces the case that is used in the application. Two cases are used today in VirSam – a gynecology case, and a geriatrics case. Both these cases have been created and realized as a cooperation between medical personnel and computer science personnel over a period of several years. The learning outcomes of the cases have emerged from the curriculum of nursing- and medical students and are presented in Prasolova Førland et al. (2016) [2]. As the cases have evolved over a long time and have proven successful, the cases themselves should not be tampered with too much. However, some minor alterations have been made to the geriatrics case deployed in the prototype.

The reasoning behind modifying the case is that this thesis wants to address some of the issues with VirSam and because room-scale VR might introduce challenges to the students that they normally would not encounter in real-life situations. The small changes should not affect the learning objectives though, as they mainly consist of some role text modifications to increase the users' understanding of the possibilities concerning room-scale.

The following subsections detail the geriatrics case mentioned above, which is the case used in the implementation of the prototype. The information provided here is a translation of the information the students get before playing the geriatrics case in VirSam [54] and can also be found in Prasolova-Førland et al. (2016) [2].

3.1.1 Case Description

The geriatric case represents long-term patient trajectory from admission to home care.

Practical information

The role-play may have from 3 to 6 participants. With 6 participants, the roles are the patient, a doctor at the geriatric department, a nurse at the ward, a relative of the patient, a municipal employee, and a game master. With 5 participants, the role of the relative is omitted. With 4 participants, the municipality employee is also the game master. With 3 participants, there is no municipal employee, and the patient is also the game master. The game master acts as an information provider who has information about test results, which may be disclosed to other participants upon request.

Clinical situation

The scenario takes place at the inpatient ward at the geriatric department. An elderly patient, 84 years, is hospitalized after a fall at home. The patient spent several hours at the hospital A&E (accident & emergency) department prior to transferring to the geriatric ward in the evening. According to information from the A&E department, a home care employee found the patient on the floor of his home. It is unclear how long the patient has been lying on the floor, but it was probably for several hours. The patient is described as confused and anxious and cannot account for the events himself. The patient has severe pain in the back and is rather thin. It is reported that, according to the home care services, the patient has recently complained about dizziness and feeling shaky. He usually uses a walker. The spouse of the patient died one year ago. He has two children and four grandchildren. He lives in an apartment on the 3rd floor with a lift. The patient is usually fairly self-reliant but uses home care services for certain tasks such as taking on/off compression stockings. He also gets his medicines in a so-called multi-dose system where the medicines are prepacked. He also has a remote control safety alarm. His past diseases include hypertension, diabetes, and osteoporosis with previous compression fractures. The drugs he takes include Metoprolol depot, Albyl E, Furix, Metformin, Calcichew-D, Alendronate, Sobril, Imovane, and Codeine as needed.

Role-play

There are 4 scenes to be played. Those who are not participants in the specific scenes are residing in the background or the waiting room in the virtual hospital. Their primary function is to observe. Details about each role can be found in Norwegian in 0.

Scene 1: The first evaluation

The scene takes place inside the patient’s room in the geriatric department shortly after the arrival of the patient at the ward. Participants include the patient, doctor, nurse, and relative of the patient (Fig.3). The main purpose of this scene is to make an initial assessment of the patient's situation. The doctor and nurse need to clarify who does what and inform the patient and the relative. The scene is concluded with the doctor and the nurse summing up what needs to be done to clarify the situation (e.g., which tests to be done) before leaving the patient room and proceeding to the meeting room.

Scene 2: Interdisciplinary planning meeting

The scene takes place inside the meeting room. Such meetings normally are held at a fixed meeting time on the first business day after the patient's admission to the geriatric ward. Participants are the doctor and the nurse (who are often joined by a physiotherapist and an occupational therapist). The purpose is to present the patient (hospitalization cause, findings so far), clarify the treatment objectives and actions, and start planning the discharge of the patient. A designated game master provides information about the test results and findings from examinations that were requested in scene 1. The scene concludes with the doctor and nurse summing up a course of action before the doctor leaves the room.

Scene 3: Phone call to the municipality

The scene takes place inside the meeting room. The time is a given number of days prior to the planned discharge date (i.e. several weeks after scene 2). Participants are the nurse and a municipal employee. The goal of the call is to start the discharging process and clarify how the municipality might contribute with personalized home care or a temporary admission of the patient to a rehabilitation or nursing home. Ending the telephone call concludes the scene.

Scene 4: Discharge Meeting

The scene takes place inside the meeting room a given number of days after scene 3. Participants are the patient, the doctor, the nurse at the ward, the relative(s) of the patient, and a municipal employee. The purpose is to plan the patient's discharge as well as any necessary further action, such as home care. Toward the end of the meeting, the participants sum up the further action plan for the patient.

3.1.2 Post-Scenario Actions

When the scenario is finished, the participants discuss experiences like what went well and what went badly. The participants should also find out what changes they can do to solve the case more efficiently. When this has been discussed, the case is played once more. The second play through is then discussed in the same way, and the participants should share what went better this time and why. Finally, everyone reads through a comment sheet with some theory which can clarify the learning outcome or if any actions were unclear during the play through.

Each participant is also required to go through a questionnaire where they answer what they think of the scenario and the technical solution. This part is mainly for the creators of the project to get feedback from the users.

3.1.3 Problems with VirSam in Second Life

As mentioned earlier, in addition to Second Life not supporting room-scale VR, it suffers from many other issues. Since VirSam is a mini-game inside this virtual world, it is limited by the functionality it provides. Even though Second Life is relatively rich in functionality, it is not very customizable, and as it was released in 2003, it is outdated in many areas. Prasolova-Førland et al. (2016) list the biggest challenges the users of VirSam experienced when role-playing, and mention that Second Life suffers from difficulties with sound, camera movement, trouble reading case information (as it does not exist inside the game, and is usually printed out on paper), and that it generally has too much functionality unneeded for the cases [2].

A VR HMD can also be used in Second Life, and the users of VirSam have stated that the scenarios feel much more immersive and lifelike when using this. However, VR is not very well implemented, and simulator sickness [55] frequently occurs, making the cases very demanding to get through without feeling ill [2]. Furthermore, a VR interface makes reading the case information even more challenging, as the HMD needs to be removed each time a user wants to know about the case.

To address some of the issues identified in VirSam, *usability* has been the most important part of the prototype architecture. This is discussed in detail in Chapter 4 and is the core of RQ4.

3.2 Testing

This section details how the testing for the prototype was realized. It explains the testing methods used through the course of this thesis, and finally the three phases of testing. The first phase was an initial user test at an expo, the second phase was continuous testing during development, and the third phase was a pilot test on an early prototype similar to the final prototype.

3.2.1 Testing Methods

Both qualitative and quantitative research methods were used over the course of this project, as was the case for the evaluation of the final prototype. Quantitative data was gathered through surveys, as well as an examination of survey results from the development of VirSam. The qualitative data was collected through conversations and casual interviews with participants, observations during testing and role-play, and a focus group during testing of the final prototype. Because the surveys gathered no personally identifiable information and as no audio was recorded during the qualitative research the results have not been reported to the Norwegian Centre for Research Data (NSD).

3.2.2 Common Domain Language

Callele et al. (2005) discuss how requirements engineering in the video game industry can be a difficult task, especially because video games often involve multidisciplinary teams [56]. Teams put together of people from different disciplines are especially demanding on understanding requirements, as they do not perceive problems and possibilities the same way. A proposed solution to dealing with this issue is to agree upon a "common domain language" – a language that everyone understands.

For this project, two different disciplines are interconnected: computer science, and medical science. During a workshop with some of the stakeholders from the medical discipline, it was immediately discovered that things that a computer scientist finds trivial are in fact not trivial for the medical staff, and vice versa. Callele et al. (2005) say that "*Once all stakeholders fully commit to the domain language, then a set of requirements that captures the stakeholders wants and needs can be generated.*" This is precisely what was done for both parties during the workshop. The medical staff understood the technical requirements of the system better, and the computer scientists understood the medical requirements of the system better. This made the requirements for the final prototype much clearer.

3.2.3 Phase 1 – Initial Testing

The first testing involving other users was done during the Technoport 2017 Expo [57] in Trondheim on March 8. For this expo, an offline scene was created to lower the risk of complications. The most important part of the initial testing was to explore different interaction methods and to find out which buttons were the most intuitive to use for various tasks rather than testing collaborative functionality.

The expo was a fitting place to get some feedback on basic things like movement, controller usage, and object interaction in room-scale VR. The stand consisted of a desktop computer with an Oculus Rift with Touch controllers and a big TV connected to the computer. On the TV, attendees not using the Rift could see what the person wearing the Rift was seeing and doing. During the day, more than 50 people visited the stand. Through a virtual survey, observations, and casual interviews, feedback was collected on what worked well and what could be improved. The most important lesson learned during this phase was that having too much functionality on the controllers is not a good idea if the user is given little time learning the application.

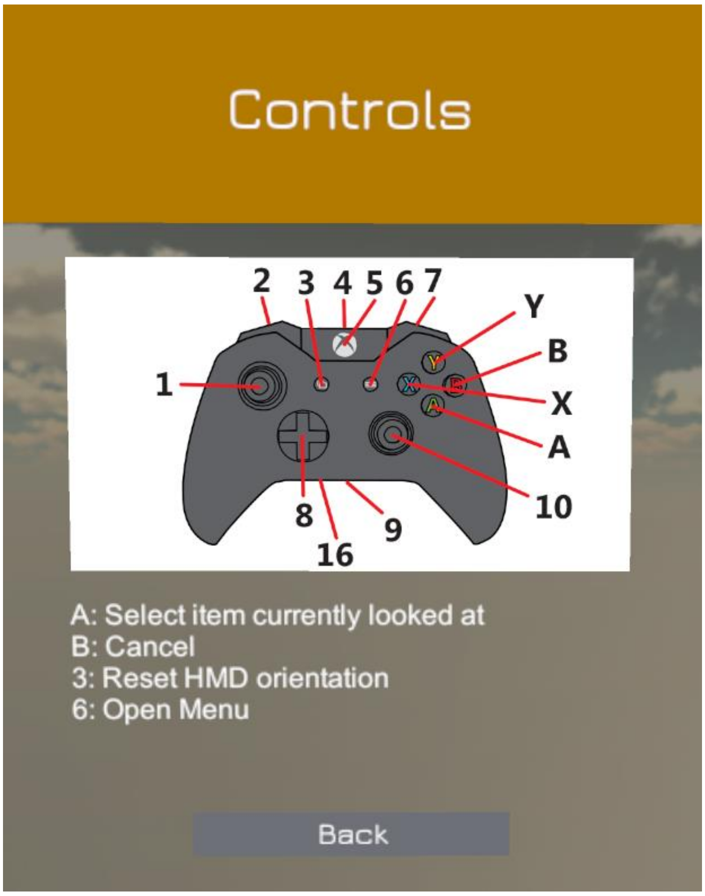
3.2.4 Phase 2 – Continuous Testing During Development

As VR is so fascinating to most people, finding test users and getting feedback is easy. Throughout the course of the project, testing was done alongside the development in an iterative process, as is important when utilizing rapid application development (mentioned in RQ2). When new features had been implemented, a meeting was typically scheduled with friends, co-students or supervisors to get feedback. The feedback was then used to develop the prototype further.

As mentioned earlier, an interaction method that was researched was controlling an animated avatar with an Xbox-controller while wearing an HMD (i.e. no room-scale is used). When VirSam in Second Life is played using VR, a similar approach to this is utilized, the difference being that in Second Life, mouse and keyboard are used instead of a gamepad. This interaction method was not polished enough to make it into the final prototype, however. Thus, some if the lessons learning regarding gamepad engagement are presented below.

Seeing the Input Device

A problem when using either of the interaction methods mentioned above while wearing an HMD is that the user cannot see the input device. This means the User Interface (UI) inside the HMD needs to guide the user in some way so that the HMD does not need to be removed each time the user wants to click a button. The reasoning behind introducing a gamepad is that it has fewer buttons for the user to keep track of. During testing, however, it was clear that an Xbox-controller could be even more confusing than mouse and keyboard for the end user, as they are more familiar with mouse and keyboard. Even though a help menu was displayed at startup (shown in Figure 1), the users kept forgetting which button did what, and when one forgets which button opens the menu, it is problematic to get back to the help page again.



Controls

A: Select item currently looked at
B: Cancel
3: Reset HMD orientation
6: Open Menu

Figure 1: Help Menu of Xbox-controller Interaction.

Centering of Head

When playing using a gamepad while wearing an HMD instead of using room-scale, one typically sits down. When the game starts, the in-game camera should then be placed inside the avatar’s head, and the player should see the game world from the avatar’s perspective. This way, when the player rotates the head, the avatar head will follow. The question is - what happens when the player *moves* the head in a direction instead of rotating it in place? Then the camera would be moved outside the avatar head, which breaks immersion as the player suddenly can see the avatar from a third-person perspective. One would think that the game engine could just force the camera to stay centered in the middle of the avatar’s head, but this causes severe simulator sickness, resulting in nausea after a few seconds for most players. A solution for coping with this issue is to have a reset button, which the user can manually press from time to time for centering the camera again. This functionality was not entirely polished though, which meant the interaction method could not be a part of the prototype at this point.

Simulator Sickness

To begin with, much of the same discomfort experienced in Second Life was encountered in the prototype when using a gamepad paired with an HMD. Simulator sickness is a conflict between the visual and bodily senses [55]. The biggest issue was that avatars used a running animation when moving. This caused a very high simulated acceleration that made users feel ill instantly. The movement- and acceleration speeds were then lowered, and the running animation was changed to a walking animation, resulting in a much more enjoyable experience.

Another small detail that helped to improve the simulator sickness issues was to move the camera outside the animations. For immersion, a subtle standing animation is used when an avatar is idling, and when this animation is playing, the avatar's head moves slightly. In the beginning, the camera was attached to the head, which resulted in the camera moving constantly. When the camera was moved outside this animation, the VR experience became more pleasant.

3.2.5 Phase 3 – Pilot Test

A pilot test was conducted at St. Olav's University Hospital during a networking conference (MedSimNorge) held by the Norwegian medical simulator center (MSS) [58]. During this conference, nurses from across Norway gather to discuss medical simulation. For the pilot test, three groups of six people tested the application, which provided valuable feedback on how the application would cope in the hands of nurses. Even though the application was not finished, it was good enough for the case to be played through without any game-breaking bugs. What was interesting with this test was that the average age of the test users was around 50 years old, meaning that the users had little to no experience with VR or video games. The main stakeholders, nursing- and medical students, are necessarily more familiar with this kind of technology, but the test gave an interesting view on how important it is to make things simple and intuitive. The users were supposed to answer a survey when the scenario was over, but due to limited time, no surveys were answered. Hence, the observations from the pilot test are purely qualitative in the form of casual interviews, conversation, and observations. Some key points that were discovered in the pilot test are mentioned below.

User Interface

When clicking a chair to sit down, the avatar was not immediately placed in the chair, but instead a "sit down"-button appeared on the screen. The buttons and user interface, in general, were too small for many of them to see. As a result, the sit-down button was removed, and instead, clicking a chair places the avatar directly into a chair. The UI size was also increased quite substantially.

For teleportation in VR, a straight line was used. It was discovered that the straight line made it hard to aim down at the floor as the controller needed to be tilted at an angle uncomfortable to the wrist. This line was later changed to a Bézier curve pointer, meaning one can aim straight forward or up and be able to hit the ground.

Movement

An interesting observation regarding movement was that the users, who had little technical knowledge, generally struggled more with the mouse and keyboard movement than when using room-scale. Moving around in room-scale is so similar to moving around in real-life that this way of playing the scenario was more intuitive for most users than playing.

3.3 Requirements

The requirements have evolved as a result of studying the case explained in 3.1, conversations with experts in the field, past experiences with VirSam, findings in Chapter 0, feedback from user testing, as well as experience from previous projects. The requirements are categorized into three different categories based on the *user-place-artifact* framework explained in section 2.7.

Like discussed in 2.7, when using the *user-place-artifact* framework, one tries to look at how a scenario can be realized in terms of *activities*, not in terms of the technology the user has at its disposal. However, since mouse- and keyboard interaction and room-scale interaction are so fundamentally different, how each requirement is realized will vary based on which interaction method is used. Challenges connected to using this framework when supporting different interfaces is discussed in 7.2.

The following three subsections give and overview of the final user requirements, place requirements, and artifact requirements, respectively. How these requirements were realized in the final prototype is further detailed in section 6.1 in the Results chapter.

3.3.1 *User* Requirements

The *user* requirements explain how the users of the application are represented in the scenarios, as well as how they participate. These include things like avatar appearance and animations, communication, interaction between users, navigation and camera movement and sense of presence. Table 5 shows the user requirements (U) of the application.

Table 5: User Requirements

Requirement	ID	Description
Presence	U1	The user should be represented as a virtual character and feel as if inside a virtual world. Other players should see the user's virtual character and be able to react to its actions, animations, body language and movement.
Avatar	U2	The user should be represented visually as an avatar. The avatar functions as an embodiment of the user.
Identity	U3	The user's avatar should have an identity. This means having a name and a backstory relevant to the case.
Role	U4	The user should have a role and know what to do in each scene of the case. The role should reflect how the user communicates and collaborates in a scene.
Navigation	U5	The user should be able to navigate around in the virtual world in different ways depending on the embodiment chosen.
Change View	U6	The user should be able to alter the view of the environment by manipulating the viewpoint.
Voice Communication	U7	The user should be able to communicate with other users through voice communication. The user should be able to see which user is speaking.

3.3.2 *Place* Requirements

The *place* requirements explain how the virtual space should be designed to serve the goals or realize the scenarios of the application. These include, but is not limited to what rooms are needed for a scenario, appearance of the environment, and how each room in the environment is connected to one another. Table 6 gives an overview of the place requirements (P) needed to realize the case.

Table 6: Place Requirements

Requirement	ID	Description
Meeting Room	P1	The meeting room should contain the following interactive artifacts: • Set of chairs • Phone The meeting room should facilitate the user’s ability to play through scene 2, scene 3, and scene 4 of the case.

Patient Room	P2	The patient room should contain the following interactive *artifacts*: • Set of chairs • Patient bed • Sphygmomanometer • Tablet • Cup The patient room should facilitate the user’s ability to play through scene 1 one of the case.
Hallway	P3	The hallway should mainly be a place for spawning and learning the application. It should contain the following interactive artifacts: • Set of chairs • Couch • Phone The hallway should facilitate the user’s ability to play through scene 3 of the case.
Environment	P4	The environment should look like a hospital environment. Each room should have static props relevant to the function of the room. Each room should be near the others.

3.3.3 *Artifacts* Requirements

The *artifacts* requirements explain what virtual objects are needed in the scene to serve the goals or realize the scenarios of the application. Thus, an artifact is a tool mediating the user's activities. The artifacts requirements explain what functionality the objects should have, as well as how an object should interact with another. The artifacts requirements (A) of the case are shown in Table 7.

Table 7: Artifacts Requirements

Requirement	**ID**	**Description**
Collider	A1	All objects should have colliders, meaning that if an object or a player touches another, the system will react accordingly
Interactive	A2	All objects relevant to the scenario should be interactive, meaning the user can interact with it in some way. It can also mean that different objects can interact with each other.
Chair	A3	Each chair should facilitate the user's ability to sit down in it.
Bed	A4	The bed should facilitate the user's ability to lie down in it.
Sphygmomanometer	A5	The sphygmomanometer should facilitate the user's ability to read the patient's blood values. The sphygmomanometer should have the ability to be placed around the patient's arm. The sphygmomanometer should transmit the measured blood values to the tablet.
Tablet	A6	The tablet should facilitate the user's ability to read the values measured by the sphygmomanometer. The tablet should receive the values transmitted by the sphygmomanometer.
Cup	A7	The cup should facilitate the user's ability to simulate giving a cup of liquid to another user.
Phone	A8	The phone should facilitate the user's ability to simulate having a phone call with another user.

4 Architecture

Creating an application from scratch puts a high demand on the architecture behind the application. To address some of the issues encountered when playing VirSam in Second Life, it is important to analyze how the stakeholders are using the application and more importantly how they want to utilize the application. It is also vital to explore the most important drivers for the architecture, as well as how the application should evolve in the future. This chapter will not dive into technical details, but rather explain some architectural aspects that have helped form the requirements of the prototype, providing valuable insights for RQ2. Also since usability issues have resulted in some negative aspects concerning the user experience of VirSam, usability is important for RQ4.

4.1 Stakeholders

4.1.1 End User

The end user is the primary stakeholder. It is important to have the medical- and nurse students in mind throughout the project as it is they who ultimately will make use of the application.

4.1.2 Supervisors

The people involved in this project have several years of experience in their respective fields, some in medical studies and some in computer science, and their advice should be heeded along the way. Aslak Steinsbekk [59] has been involved in the medical aspects of VirSam from day one, meaning he knows better than anyone the needs and wants of the end users. Ekaterina Prasolova-Førland [60] has been working on Virtual Collaborative Environments and Virtual Reality for nearly 20 years and has been involved in the technical development of VirSam in Second Life. Both Aslak and Ekaterina know the strengths and weaknesses of the current solution very well.

4.1.3 Developers

Because modifiability is so central to this project, future developers are essential stakeholders. It is therefore important that adding functionality to the application is easy if the project is to evolve and survive in the long run.

4.1.4 Unity Technologies and Assets

The application must conform to the policies and agreements of both Unity Technologies and any third-party assets or software used in the application.

4.2 Architectural Drivers

When developing an application, one necessarily cannot simply jot down a series of requirements and immediately start working on them. One also needs to envision *how* these requirements should be realized to satisfy the users' needs. Bass et al. (2013) talk about different Quality Attributes (QAs) that need to be taken into consideration during software engineering [61]. These attributes can provide guidelines to what one should focus on when developing to end up with the desired results. The two most important QAs for this project are *usability* and *modifiability*.

4.2.1 Usability

Usability describes how easy it is for a user to accomplish a desired task and the kind of user support the system provides. The areas that usability comprises include learning system features, using a system efficiently, minimizing the impact of errors, adapting the system to user needs, and increasing confidence and satisfaction (Bass et al., 2013). RQ4 asks about the user experience of the presented solution. User experience is closely related to usability, but where usability asks how intuitively and easily a user can perform a task, user experience asks how a person feels when using a product.

Section 3.1.3 explains how both usability and user experience are realized poorly in Second Life. The game crashes and is unstable; the sound quality is low; and the game has too much functionality, making it time-consuming learning the necessary functionality - the list goes on. Furthermore, when looking at Second Life's ability to realize research question RQ1 regarding combining procedural- and collaborative training, it is evident that Second Life does not support realistic procedural training at all, and that many things that medical- and nursing students want to practice, cannot be realized. The main goal of VirSam is to enable as many students as possible to play through a virtual case in a short time frame. Moreover, a typical end user has little knowledge of computer games or virtual reality, putting high demands on the usability of the system.

4.2.2 Modifiability

Modifiability describes how easy it is to add new software or modify existing software in the system when the system is already deployed. Modifiability is a key QA for the prototype as only one case is implemented during the scope of this project. For the application to evolve in the future, it is vital that adding new functionality can be done quickly and easily. Also, RQ3 puts especially high demands on modifiability, as the application should adapt to different devices and interfaces.

4.3 Architectural Tactics

In addition to presenting how QAs drive the architecture of a software program, Bass et al. also discuss some *Architectural Tactics* (ATs) that can be used to support QAs. This section describes some of the ATs used to facilitate both usability and modifiability in this project.

4.3.1 Architectural Tactics Facilitating Usability

Regarding usability, the philosophy behind the new application has been to keep things simple, quick to learn, and user-friendly. These aspects were realized by keeping functionality to a minimum and having access to a menu from all different interaction methods, providing information about how the case should be played or how the character is controlled.

Achieving a high degree of usability is essential when making an application. It is of particular importance when the domain is unknown to the user, which a room-scale VR application necessarily is to many medical- and nursing students. In 2017, room-scale VR is something that most people have not had the opportunity to test. It is then especially important to give *feedback* to the user, make the app easy to *learn* and let the user take advantage of the system without spending too much *time* getting to know it.

Below are some of the ATs used to facilitate usability in the prototype. Bass et al. (2013) refer to the ATs used for usability QAs as *support user initiative* and *support system initiative,* which are somewhat abstract terms. To make the terms less abstract, the sub-qualities *feedback, learnability,* and *efficiency,* which are closely related to the support user initiative AT, are discussed instead. Interestingly, these qualities are also closely related to some of the game flow elements discussed in 2.4 needed to keep a user entertained, emphasizing that the architecture of a software program can have a direct impact on user experience.

Feedback

Giving feedback is one of the most important tactics for increasing usability, both when a user does something correctly and incorrectly. It is nearly impossible to create a system that is 100% foolproof, so when the user does something that is not intended, it is important to give feedback on what they are doing wrong. For room-scale movement, this is solved by coloring the teleportation curve red when the user is trying to teleport somewhere forbidden, and the curve is colored green in the opposite scenario. This is shown in Figure 2.

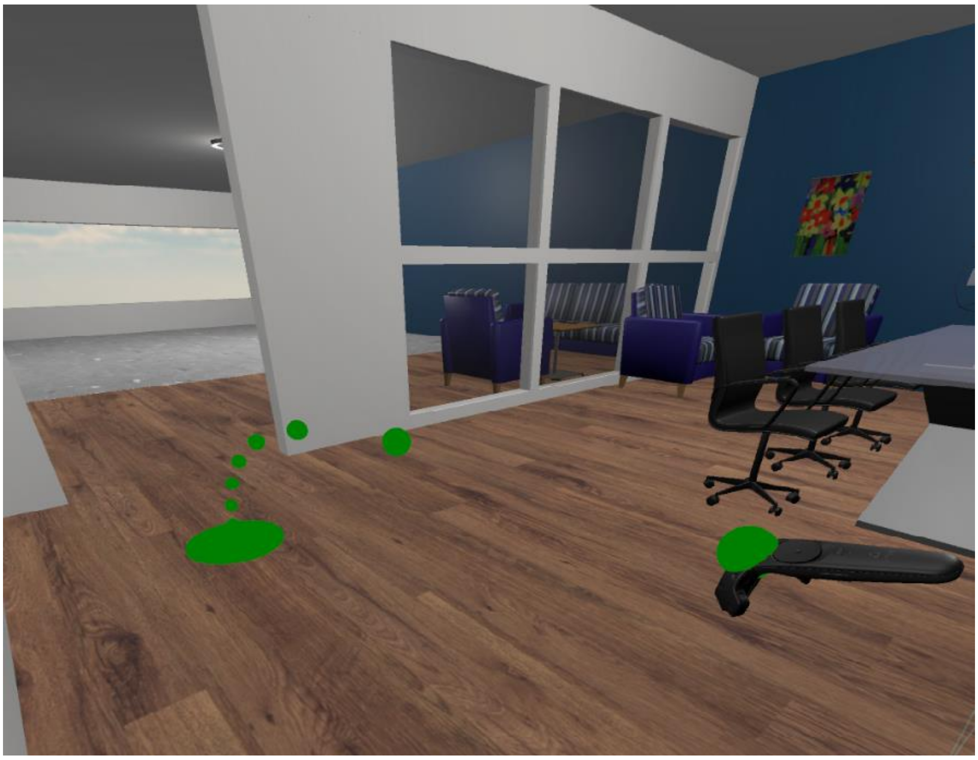


Figure 2: Room-scale teleportation curve. The green color indicates that the targeted location is valid.

Room-scale menu navigation is done by touching the touchpad on the Vive and by using the joystick on the Touch. When a menu element is highlighted, the pad or joystick can then be pressed down to click the highlighted element. The menu gives the user feedback in several ways. Icons illustrate what a menu element means and when an element is highlighted, a yellow info text displays the functionality of the menu element textually. The user's thumb placement is represented as a small dot on the Vive controller, letting users know the location of their thumbs. The controller also rumbles when an element is highlighted or clicked. All this can be seen in Figure 3.

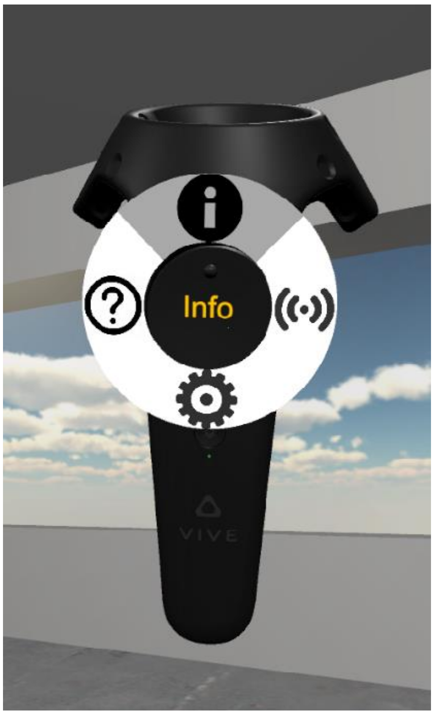


Figure 3: The radial menu of the right controller. Yellow text displays which menu element is highlighted. The small dot above the yellow text is a virtual representation of where the player's thumb is located.

Other things done to facilitate feedback include highlighting of objects, icons indicating when a player is speaking, 3D sound, and a "ding" sound for when a player joins or leaves. These aspects are further detailed in the prototyping chapter (Chapter 5) and the results chapter (Chapter 6).

Learnability

Learnability has to do with how intuitive the application is to use and how quickly a user learns how to use the different functions. Realism and embodiment is a natural part of a VR simulator, and if a room-scale system simulates realism, users spend less time learning the application, like discovered in the pilot test (subsection 3.2.5).

As mentioned earlier, test phases 1 and 2 (detailed in section 3.2) revealed that having too much functionality on the controllers could be confusing and led to low learnability. Thus, a principle used in the implementation has been to keep functionality inside the *artifacts* instead of handing functionality to the *users*. For example, the only thing a user can do to the sphygmomanometer (A5) is to pick it up and drop it. It is the sphygmomanometer itself that handles everything from providing highlighting when near a patient's arm, snapping to the patient's arm, changing ownership of the object (i.e. networking ownership), or transferring blood values to the tablet (A6). This way, users can spend less time on learning the application. Putting functionality in the artifacts also promotes modularity which can increase modifiability (discussed in subsection 4.3.2).

Complex applications often include a tutorial to speed up the learning process. Both Steam (for HTC Vive) and Oculus (for Oculus Touch) have made good tutorials on how to use their respective interfaces, where they explain the different buttons and functionality. The tutorials only take a couple of minutes to go through and are probably something all users of the system should experience to get an idea of how they can interact with the virtual world.

A similar approach could be used in the medical application itself, but as the geriatrics case is quite simple when it comes to functionality, this has not been in focus this time around. If the functionality repertoire were to increase in the future, an application-specific tutorial explicitly guiding the user through features when opening the app could be an essential part of the learning process.

Instead of a complete tutorial, the user is presented with tooltip information explaining what each button does (shown in Figure 4). As the tooltips are shown by default, the user has to enter the menu and turn them off, forcing the user to learn basic functionality before playing the case.

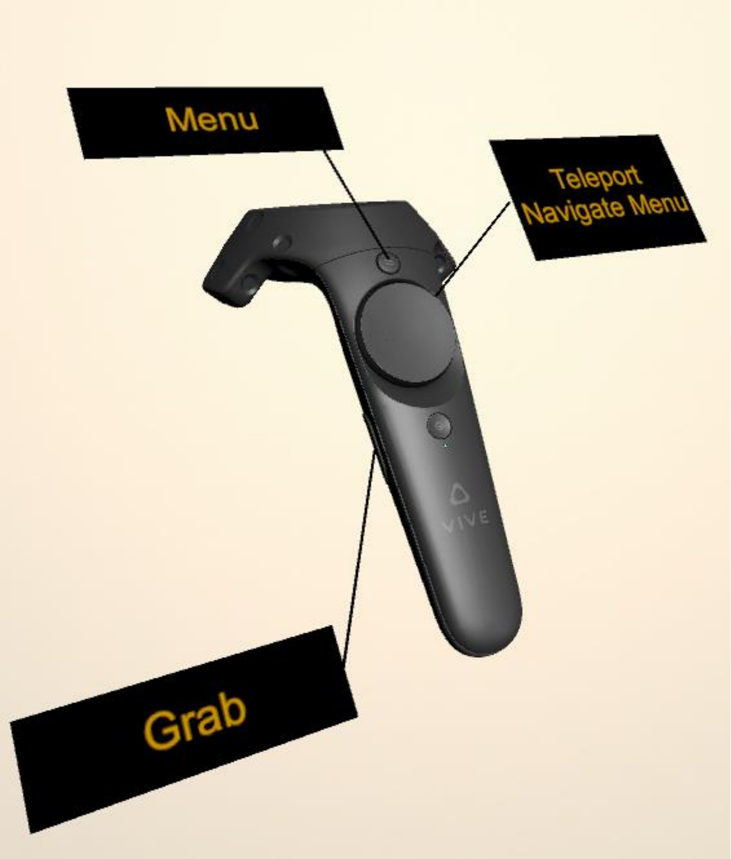


Figure 4 Room-scale controller tooltips. Each tooltip's line points directly to their respective button, guiding the user in the desired direction.

Efficiency

Efficiency describes how efficiently a user can execute a given task. For this case, *speed* is not the main purpose, as it is more important for the students to do a given task *correctly* - both regarding procedural activities and in terms of communication. Nevertheless, as the students playing VirSam in Second Life spend too much time on tasks irrelevant to the role-play, speeding up or avoiding tasks unrelated to the case is important.

One identified problem with Second Life is that the UI is untidy and confusing, featuring too many UI elements (Figure 5). Creating menus for a hybrid app is not necessarily a simple task. Each interaction method requires a different kind of menu, as a UI method suited for one device might be unsuited for another. Thus, separate menus were created for the various interaction methods.

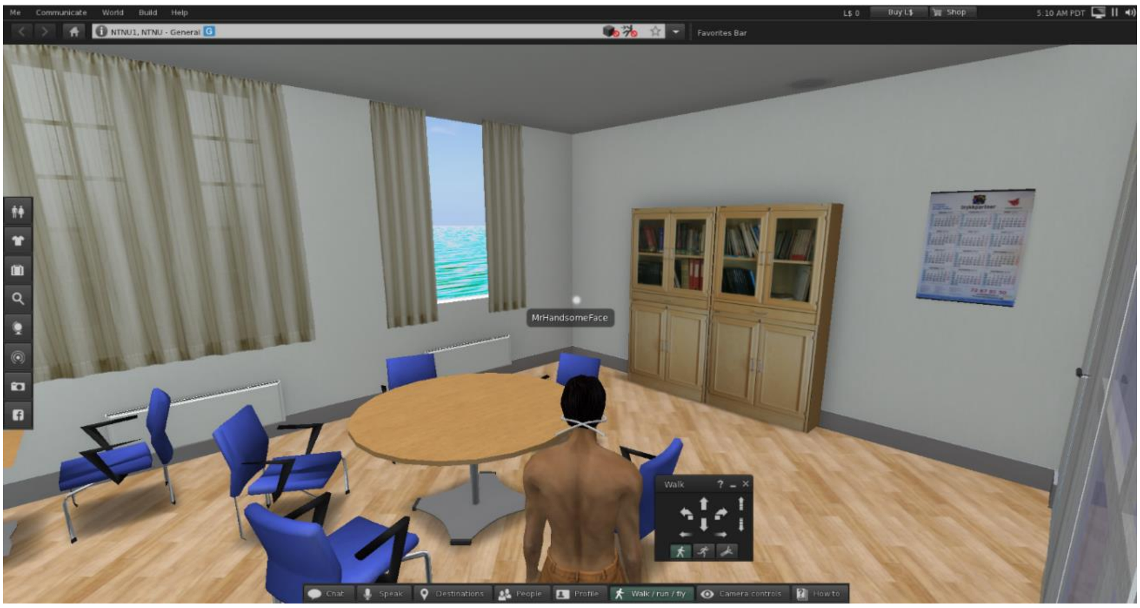


Figure 5: Second Life UI. The UI introduces a conglomerate of functionality, most of which is unneeded for a VirSam user. The UI contains a top bar, a side bar, and a bottom bar, with several more UI boxes popping up when pressing a menu element.

For mouse and keyboard, a standard 2D overlay was used. When in-game, the only UI element presented to the player is a menu button in the top left corner, making the UI much less overwhelming. This can be seen in Figure 6.

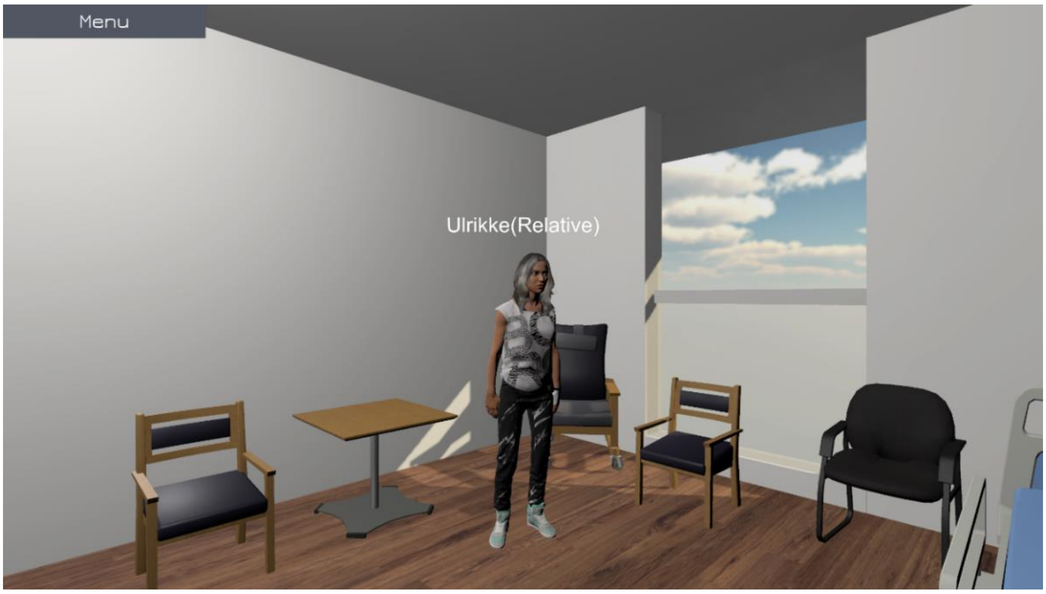


Figure 6: Prototype UI. There is only one UI element visible. This way, there are fewer distractions, which makes it easier for the user to focus on the role-play.

When the menu button shown in Figure 6 is pressed, the main menu is shown (Figure 7). The menu only has a few elements, and it is clearly communicated what each button does. This centralization of information and settings contrasts that of Second Life, where functionality is scattered across the different menu buttons.

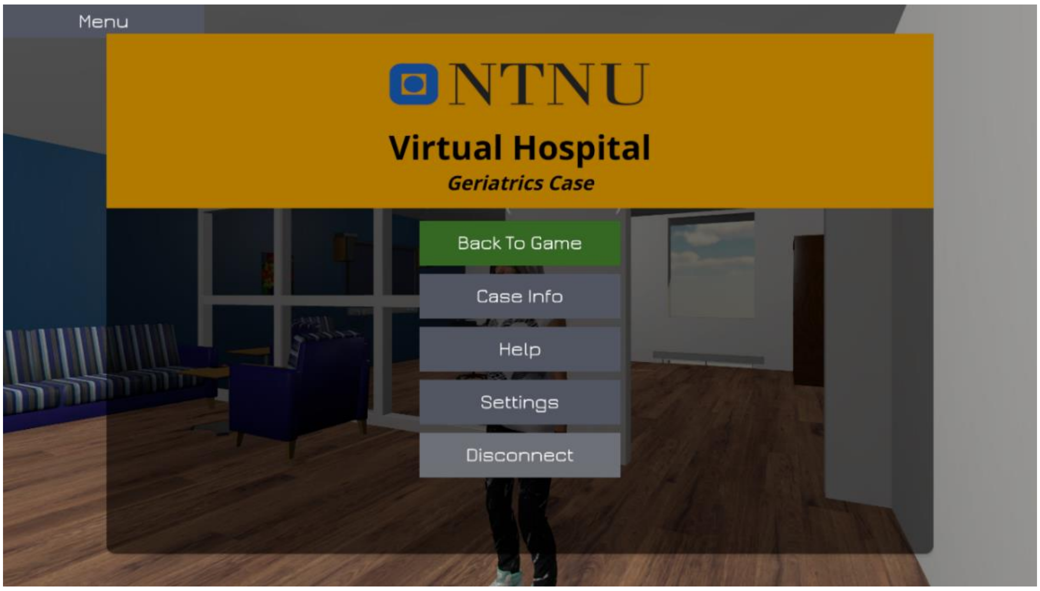


Figure 7: Prototype main menu. It is clearly communicated what each button does.

Another identified issue concerning efficiency is that Second Life users wearing HMDs cannot access the case information without removing the HMD, meaning the users have to take the HMD on and off to play the VirSam cases properly. To solve this, the case information was put inside the application, allowing users to have access to the case information on demand. This is vital, as spending significant time finding out what to do in a scene can break immersion and communication flow.

Proximity to the relevant environment is also an issue in Second Life, as users spawn far away from the hospital environment and need to navigate to it before they can start playing the case. In the prototype, the users, therefore, spawn inside the hospital environment automatically, increasing efficiency.

4.3.2 Architectural Tactics Facilitating Modifiability

To improve the longevity of the application, adding extra features or modifying existing software must be quick and easy for any future developers. Below are some of the ATs used to facilitate modifiability in the application.

Reduced Size of Modules

Object-oriented programming is a programming paradigm where one works with objects containing data. This is the most commonly used programming paradigm today, as it is easy to relate to an object for a human since we are all used to the three-dimensional world around us. Unity takes object-oriented programming a step further than regular programming, as one can visualize the object one is working with graphically. Say one wants to make a ball in Unity – then one simply creates a sphere, drags it out into a scene, and then the ball immediately has a graphical and a data representation. Now all the developer has to do is attach a script to the sphere, write the code for what the ball should do, and the ball behaves according to the script. In other words – the size of modules is usually no larger than the object itself. This helps enforce the single responsibility principle [62], as it makes it easy to write code where each piece of code performing a function can be considered a module of its own.

Reduced Coupling

Low coupling between two modules (i.e. the modules mentioned above) means that the interdependency between said modules is low. This means that modification of one module will not affect the other. Unity naturally allows for reduced coupling with the scene functionality that it provides. Scenes are almost entirely separated from each other, so making a change in one will not affect another one. With that being said, it is still easy to copy something from one scene to another one. This means the introduction of scenes does not lower the usability of the system. Keep in mind that a scene in Unity is not the same as a scene in the case. All four scenes of the case are played inside one Unity scene.

Deferral of Binding

Deferral of binding is the time between when a developer makes a modification to the code until it is run in production. The project is hosted on GitHub [62], which means several developers can work on it asynchronously. The branching system that Git [64] provides makes it easy to have one code base with several branches. For this project, the master branch is intended to be the stable state of the application that the end user utilizes. Development is done on a branch called “devel” or a sub-branch of devel. If one then wants to make changes to the final product, one “checks out” the devel branch. When on the devel branch, changes can be done and tested in a safe environment, and when the code has been thoroughly tested the code can be merged into the master branch and shipped to the end users.

The Git workflow is not necessarily needed when only one developer is working on a project in a prototyping phase, but if several developers are going to start working on the project in the future and the project is shipped out to the customer, it is a good thing that the infrastructure is already in place.

5 Prototyping

Creating a game from scratch in a few months is undoubtedly a challenging task for one developer, and when some of the requirements also include networking, voice communication, and VR support, the challenge becomes a big one.

Two sets of requirements are discussed in this thesis. The *user-place-artifact* requirements revolve around activities performed by the end user, and the game engine requirements (GEs) discuss what is required of the game engine to facilitate the *user-place-artifact* requirements of the application. In other words – the GEs are important to the developer.

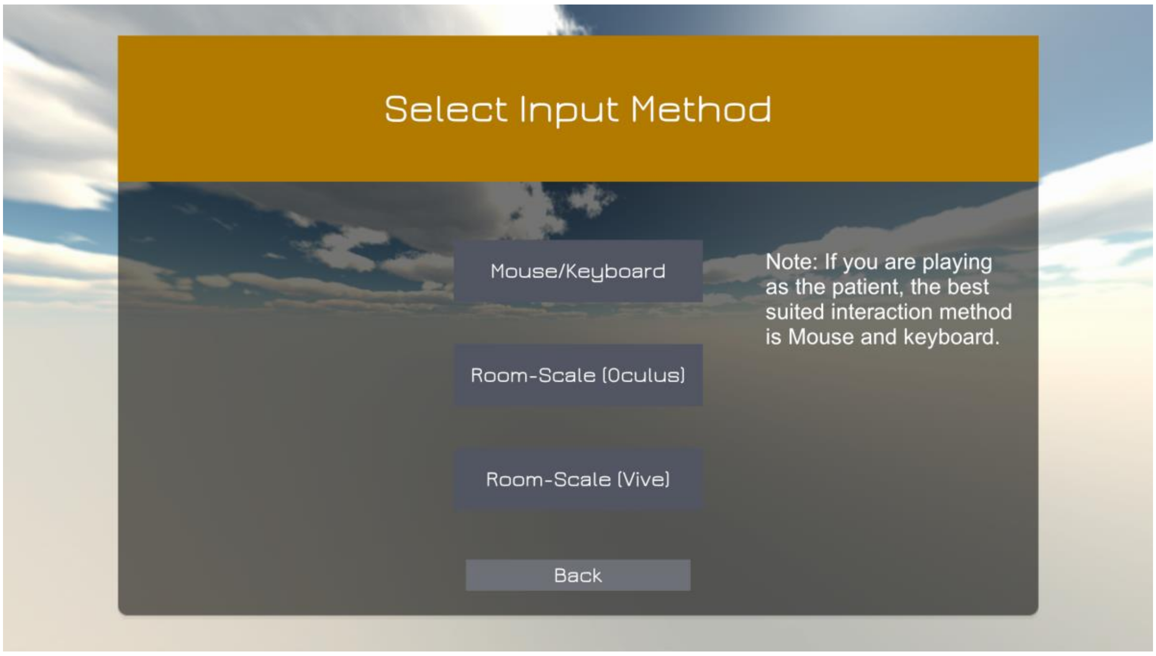
This chapter covers how Unity, including frameworks, assets, and toolkits, was used to realize the GEs listed in section 2.9 and furthermore helps answer RQ2.

GE1 Cross-platform

Even though a mobile solution is a highly relevant research field during the investigation of an SVUH, it was not included in the development process as it takes more time to customize than a solution supporting only “regular” computers. RQ3 asks how the presented solution facilitates smart adaptation of different interfaces and devices, and as Xbox-controller movement did not make it into the final prototype, GE1 is therefore realized through mouse- and keyboard interaction (desktop) and room-scale interaction (both HTC Vive and Oculus Touch).

It took quite some time deciding how the menu system should work and *adapt* to the different interaction methods. The core of the problem was that in the beginning, the application had no way of changing between VR and non-VR, meaning the user would be stuck with the initial interaction he or she chose. If the user then wanted to switch to a different interaction method, the entire application would have to be restarted. This is clearly not *smart adaptation* in any way, form, or shape. Thankfully, it was discovered that switching between VR and non-VR could be done using coroutines [65]. This made it possible to enable and disable VR at runtime, resulting in users having the ability to join and leave an ongoing role-playing session using different interfaces as they please. The case described in section 3.1.1 states that some players may need to switch roles when a new scene starts, and as some roles may be better suited for certain interaction methods, this form of adaptation is essential.

A consequence of the problems concerning changing interaction method, was that the main menu (i.e. the menu shown *before* the user selects an interaction method) is displayed on desktop for all players even though it could be displayed in the HMD straight away for machines that have an HMD connected at startup. The player selects the desired interaction method, and when “play” is pressed, the chosen interaction method is properly initialized, and the player is taken online (see Figure 8). Thus, it is not before a user goes online and inside the playable part of the application that the menu adapts to the chosen interaction method.



Note: If you are playing as the patient, the best suited interaction method is Mouse and keyboard.

Figure 8: Selecting input method in the main menu. The HMD does not turn on until the play button on the following screen is pressed.

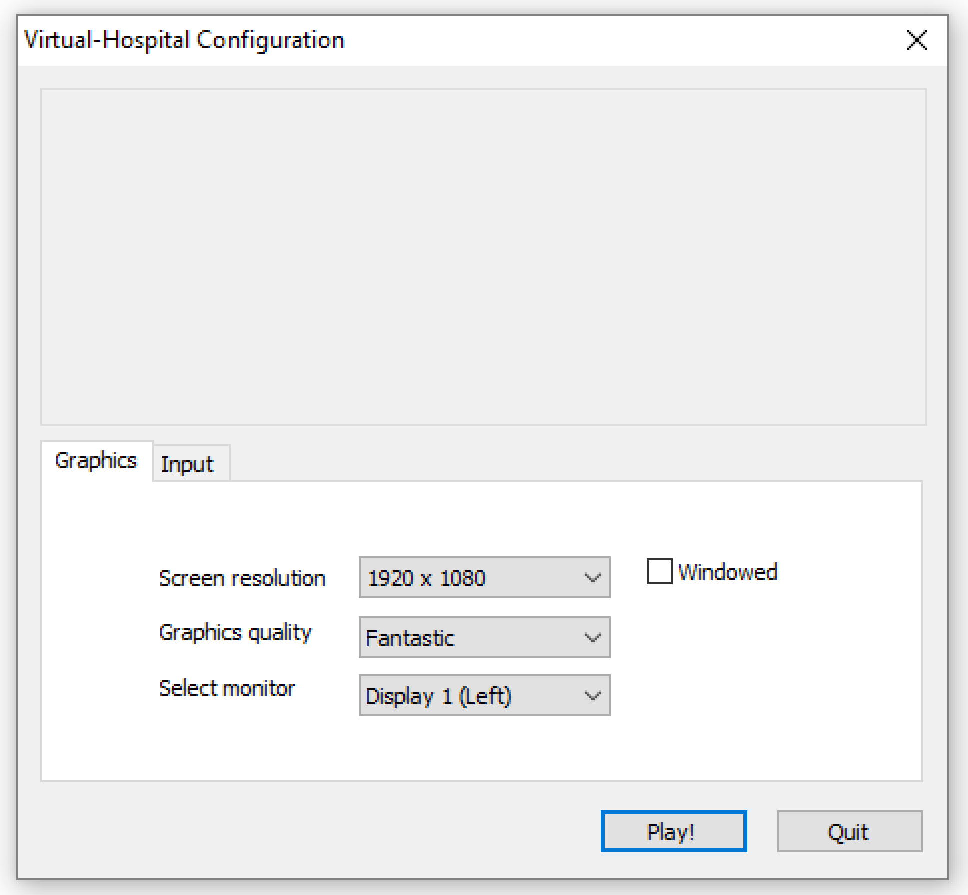
The fact that all users need to interact with the main menu using mouse and keyboard is not a problem for this prototype, however, as the VR solutions presented are connected to a computer with a regular 2D-screen and most likely also a mouse and a keyboard. The question is whether it is more intuitive that the main menu also adapts. Another concern is how a phone (or any device that is not a regular PC) user’s menu would look like. This is part of the discussion in Chapter 7.

GE2 3D Graphics

Adaptable Graphics

To realize an application that facilitates smart adaptability, one needs a way of adapting the graphics quality for it to perform with an adequate framerate on different platforms. Unity has the option to start the game using different graphics settings and to set the graphics quality to "fastest" means the application will run on several-year-old computers that have no dedicated Graphics Processing Unit (GPU). Figure 9 shows the dialog shown on application startup.

Graphics are also easily adapted to mobile solutions as Unity has a built-in shader optimized for smartphones.



Screen resolution 1920 x 1080 □ Windowed

Graphics quality Fantastic

Select monitor Display 1 (Left)

Figure 9: Prototype graphics configuration. If the user is not satisfied with the frame rate, the graphics quality can be lowered.

Props

The great thing with Unity is that it has a large asset store. Some assets are free; others are expensive. To create a realistic hospital environment, two asset packs were acquired from the asset store [66]. These packs contained several useful props and avatars that ended up in the environment of the prototype.

Lighting

Lighting is a big part of creating realistic-looking scenes by producing ambiance and shadows. There are two basic ways in which this can be done: baked lighting and real-time lighting. With baked lighting, lighting data is rendered before the application runs and stored in files. This method can increase performance significantly, but is a time-consuming process in Unity and proved difficult to do correctly. Because of this, the lighting in the scene renders real-time. To make up for the decrease in performance of using real-time lighting, only six light sources exist in the environment. If the environment were to increase in size in the future, light baking would need further research.

GE3 Animated 3D Characters

This subsection illustrates some of the things that make Unity so *modifiable* and how new functionality can be added easily.

Characters

Kleven (2014) mentions that VirSam's test users listed avatar clothing as an important factor regarding information retrieval about avatars [35]. With this in mind, 3D characters relevant to the case were retrieved from various sources.

What makes Unity a powerful tool when creating characters is that it has an Avatar editor. In the Avatar editor, most 3D model formats can be imported, and if the model is placed in a T-shape, Unity will automatically detect limb positions and create a skeleton for the character (see Figure 10). The skeleton can then be used to animate the character.

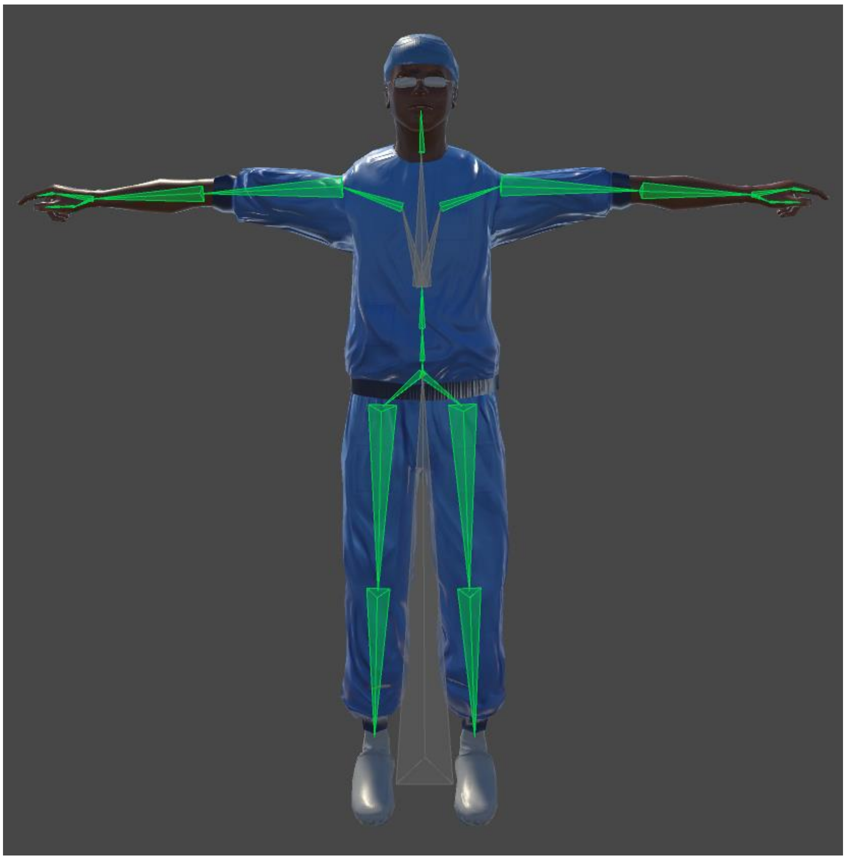


Figure 10: Unity's Avatar Editor. Unity has created a skeleton (the green arrows) based on the 3D model.

When the skeleton is created, the Avatar is ready for implementation. A Character Controller [68] and Animator [69] can be added to the avatar, and the character is ready for use.

Animations

An animation controller controls the Unity Animator. It can contain numerous different animations, and the developer can add new animations and define transitions between animations by linking them using arrows (Figure 11). Creating animations is a whole field of game development, so for this project, some predefined animations discovered in an asset called Taichi Character Pack [70] were used. This pack contains 98 animations, but with the narrow project scope in mind, only three of these (walk, sit, and lie back) were employed in the final prototype to reduce complexity.

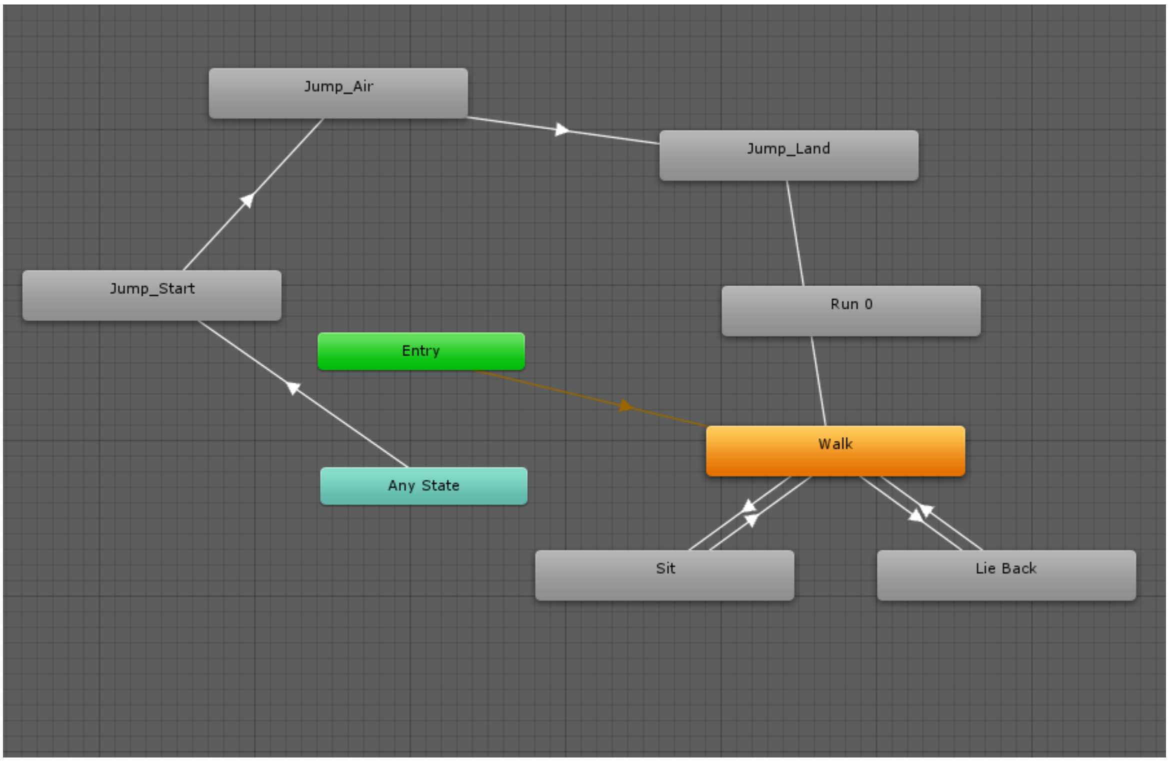


Figure 11: The animation controller used by each animated character in the application. The arrows indicate a transition between animation states. The jump animation was not mapped to a button, and therefore not accessible to the users.

GE4 Object Interaction

As RQ3 suggests, object interaction is usually done in different ways on different devices. From VirSam, it is evident that too much functionality available to the user can cause problems if the user has no experience with the domain or application beforehand. The philosophy behind the application has therefore been to keep things simple.

Desktop

When playing using mouse and keyboard, hovering over an object with the mouse highlights it in a bright orange color, suggesting that the object can be interacted with. The player can then interact with the highlighted object by clicking the left mouse button. Looking back at tactics for increasing usability in the Architecture chapter, this solution should, in theory, be much more user-friendly and efficient than Second Life. In Second Life, the user must first right click, then find the "Sit Here" option in a menu. This is illustrated in Figure 12 and Figure 13.

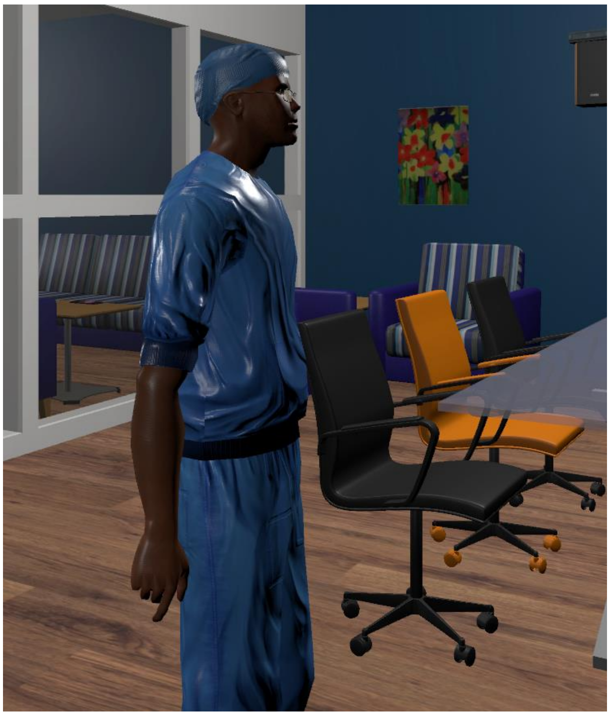


Figure 12: Prototype object interaction. Interactive objects glow orange for desktop users when the mouse hovers over them. If clicked, a predefined action is executed.

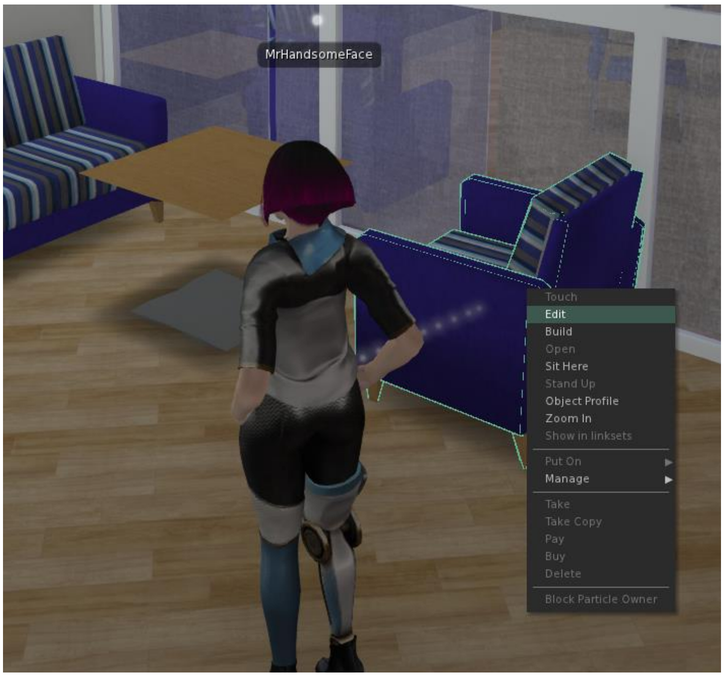


Figure 13: Second Life object interaction. Objects first need to be right-clicked, then the correct interaction must be found in a menu. Many menu choices are not even available to the user.

HMD and Xbox-controller

When playing using HMD and Xbox controller, the same principle as with mouse and keyboard is used. The user can highlight an interactive object and execute an action by pressing the A-button on the controller. As implementing a mouse in VR is complicated, a gaze pointer was created for this interaction method. A gaze pointer simply means that the player needs to look at an object to highlight it. Figure 14 shows how a user might look down to highlight an object.

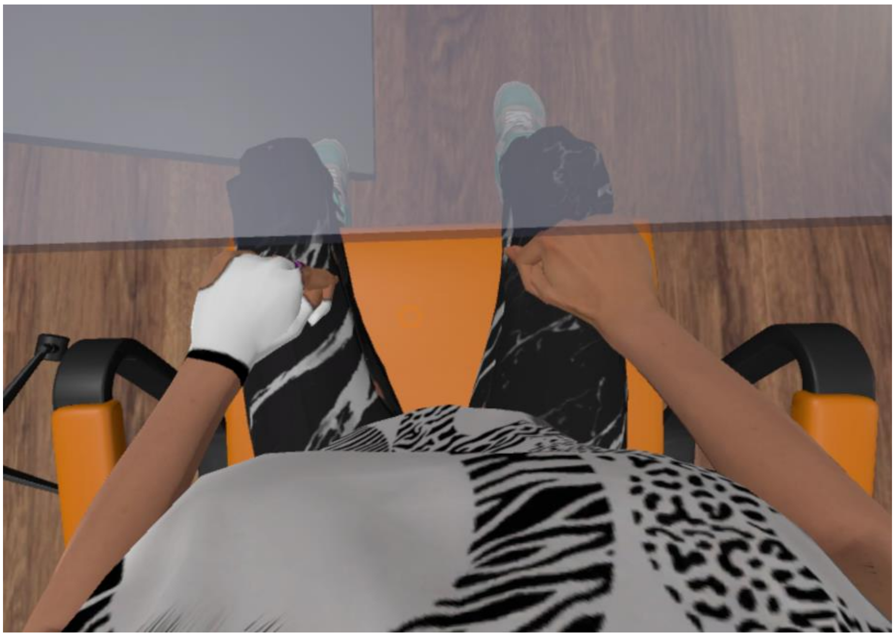


Figure 14: An avatar sitting in a chair in the meeting room. An orange circle in the middle of the screen indicates where the user is looking. This figure also illustrates how the first-person point of view looks like when using an HMD.

Room Scale

Virtual Reality ToolKit (VRTK) is a free toolkit that can be found in the Unity asset store [71]. It contains scripts, models, and examples for much of the basic functionality needed in a room-scale application. Not having to code basic functionality like button input on the controllers or handling teleportation saves time, which is vital for a project this size, and with limited time to spend. VRTK is used in this application for realizing object interaction, teleportation, and for menus when playing in room-scale.

GE5 Virtual Reality

To be able to create and build a VR application, a Software Development Kit (SDK) supporting VR is needed. For this project, the most critical VR devices will be HTC Vive and Oculus Rift. Valve, the creator of the digital distribution platform Steam and also co-creator of the HTC Vive, has also made the SteamVR SDK which, thankfully, supports not only HTC Vive but also Oculus Rift. SteamVR can be downloaded as an asset pack in Unity [72]. The developer can then enable this SDK in the project settings by selecting “OpenVR” (the API that allows the developer to take advantage of SteamVR) as Unity’s target virtual reality SDK. The toolkit VRTK mentioned in the previous section also makes use of SteamVR.

GE6 Networking

For networking, Photon Unity Networking (PUN) [73] was used. PUN is a free asset only available for Unity that simplifies the process of creating a networked application. It abstracts away the low-level code in the form of scripts that can be attached to the avatar game objects. The positional data for each player is then sent from the client controlling the player to every other client.

Photon has several hosting options, and Photon Cloud was selected for this project. With Photon Cloud, one does not have to worry about hosting and maintaining a server. If the application has no more than 20 concurrent users, it can be used free of charge. A one-time payment can be made to support 100 concurrent users [74].

GE7 Voice Communication

One of the reasons why PUN was selected is that it also supports voice chat. The Photon Voice asset pack includes excellent example scenes and scenarios that show how to implement the scripts correctly. Because of this, implementing voice was a simple process.

PUN makes use of Unity’s built-in audio sources which support 3D sound. With 3D sound, a user can hear where the sound is coming from, which makes the audio more lifelike.

GE8 Rapid Prototyping

Considering that the scope of the project only stretches out over a period of five months, it is important to be able to get started on the project quickly, without having to do much low-level programming. Unity is a toolkit that promotes rapid prototyping, which according to Greenberg (2007) is the most important step for triggering the creativity in a developer [75]. Like stated in a discussion about Unity:

> *Unity's modular system and usability allows for quickly developing a prototype of an idea. It has features like drag & drop editing, shaders, animation and other systems already in place to allow diving right into developing a game [76].*

Even though this is “just” a quote from a forum, it backs up Greenberg’s conclusion (Greenberg, 2007, pp. 155-156) that toolkits make it easier for researchers to create new breakthroughs through rapid prototyping of many new ideas.

GE9 Ease of Use

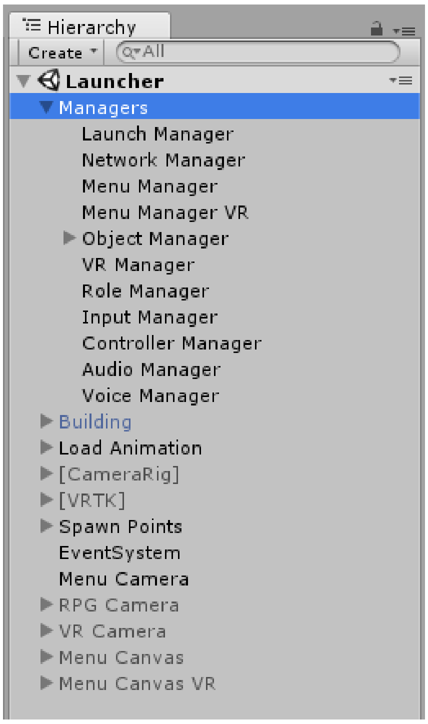
Unity has a highly customizable GUI where the user can move and resize each window after personal preference. This makes it easy for the developer to set up the development environment to their liking. Scripts are coded in C#, and can simply be double-clicked to open Microsoft Visual Studio [77], which functions as the Integrated Development Environment (IDE) for coding. Both the Visual Studio IDE and the C# coding language are considered effective programming tools, and Visual Studio provides feedback regarding errors and efficiency in the form of autocompleting code. With more than a year’s practice in Unity, Visual Studio, and C# alike, the development process went fluently.

GE10 Modifiability

Unity uses a concept known as prefabs to save game object presets. By making use of prefabs, one can store the changes done to a game object for easy reuse. A good example of prefabs is the player character discussed in GE3. After all relevant scripts are attached to the model, the character can be stored as a prefab. This prefab can then be reused for the creation of a new character, meaning the developer does not need to reattach all the same scripts.

The toolkit VRTK discussed in GE4 promotes a high degree of modifiability. It was originally created for HTC Vive, but with its high cohesion and low coupling, reuse of the different modules was easy, and the toolkit turned out to work just as well for the Oculus Touch (though with some minor modifications.)

To reduce the size modules, a set of “managers” were introduced (shown in Figure 15). Each manager has a specific purpose, meaning that modifying one manager does not affect another one. This modularization of classes also helped with development efficiency as it was easy knowing which class needed modification to introduce new features or modifying existing ones.



```
Launcher
  Managers
    Launch Manager
    Network Manager
    Menu Manager
    Menu Manager VR
    Object Manager
    VR Manager
    Role Manager
    Input Manager
    Controller Manager
    Audio Manager
    Voice Manager
  Building
  Load Animation
  [CameraRig]
  [VRTK]
  Spawn Points
  EventSystem
  Menu Camera
  RPG Camera
  VR Camera
  Menu Canvas
  Menu Canvas VR
```

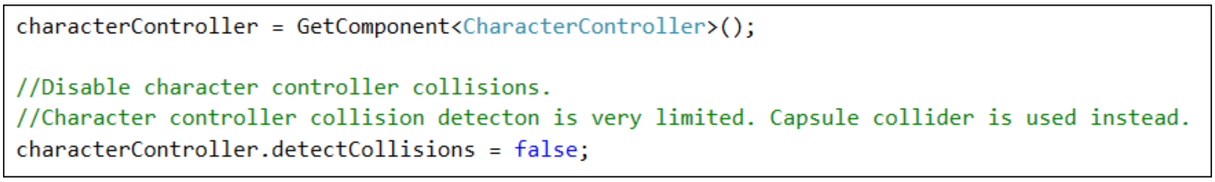
Figure 15: The object hierarchy of the offline scene used in the prototype. The first element contains the managers, and as each manager has a specific purpose, making changes to one does not affect another.

GE11 Well-documented

One of the strongest reasons for using Unity is that there are so many ways in which the developer can receive help. The official documentation [78] is user-friendly, as it normally provides a description of how the functionality works, pictures, and code examples. Furthermore, as mentioned earlier, many Unity asset packs include sample scenes, which allows the developer to see exactly how certain functionality could be implemented.

Since Unity is so vastly used, there are also many external guides and tutorials on numerous subjects. A notable guide used at the beginning of this project was a YouTube-video of a live stream where a developer creates a multiplayer application in Unity for HTC Vive and Oculus Rift, in which PUN (discussed in GE6) just so happened to be used as well [79].

The notion of well-documented code has been utilized in this project as well. As future developers are essential stakeholders, comments are used extensively to clarify decisions that may seem unusual or where elaboration may be needed (shown in Figure 16).



```
characterController = GetComponent<CharacterController>();

//Disable character controller collisions.
//Character controller collision detecton is very limited. Capsule collider is used instead.
characterController.detectCollisions = false;
```

Figure 16: Comments in the PlayerManager class. The comments explain why a decision that may seem odd has been made.

6 Results

This chapter details how the requirements listed in section 3.3 are realized in the final prototype (first part of RQ2). It also explains how the testing of the prototype was conducted and details the test results. The test results are relevant for RQ3 and RQ4 as how the requirements are realized ultimately affects in what way interfaces need to adapt, as well as user experience.

6.1 Final Prototype

6.1.1 User Requirements

U1: Presence

When in the main menu, the user chooses interaction method, role in the case, and presses play. When the play button is pressed, the player is immediately taken online and spawned into the hallway. When the connection is set up, the chosen interaction method is also initialized. If the user has chosen room-scale, the HMD activates. Otherwise, the avatar is displayed in the third person when playing on desktop. At this point, the player is fully present in the scene and is free to start exploring the environment.

Presence is an area in which virtual reality excels at as it allows users to feel part of an entirely simulated environment [9]. A high field of view, precise movement tracking, and low latency are some of the aspects that the virtual reality team at Valve mention for establishing presence [80][81].

U2: Avatar

Each player is depicted as a virtual avatar. Full-body avatars represent users playing on desktop. The look and gender of the avatar changes depending on which role the player chooses when entering the game to reflect the avatar identity. The “relative” avatar used in the case can be seen in Figure 17.

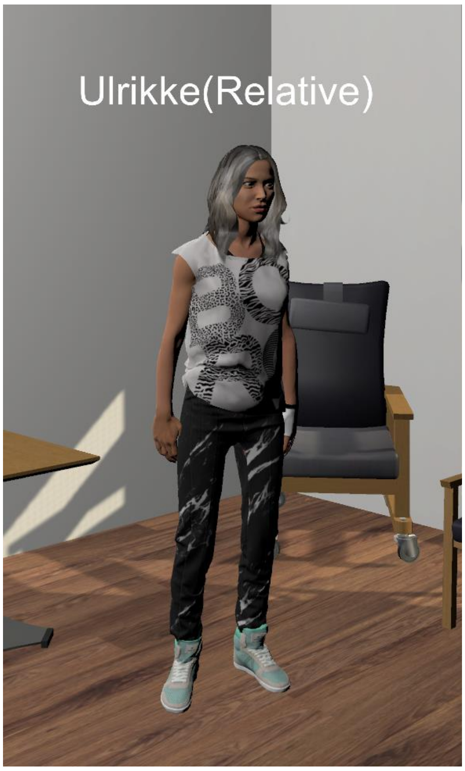


Figure 17: The "relative" avatar controlled by a user on desktop (mouse and keyboard interaction). The name and role of the avatar are visible above the avatar for all players to see.

Because of the challenges regarding animating room-scale avatars discussed in 2.9.3, room-scale players are only depicted as a floating head with two controllers (Figure 18). To be able to separate one room-scale avatar from another, the color of the head changes depending on the chosen role.

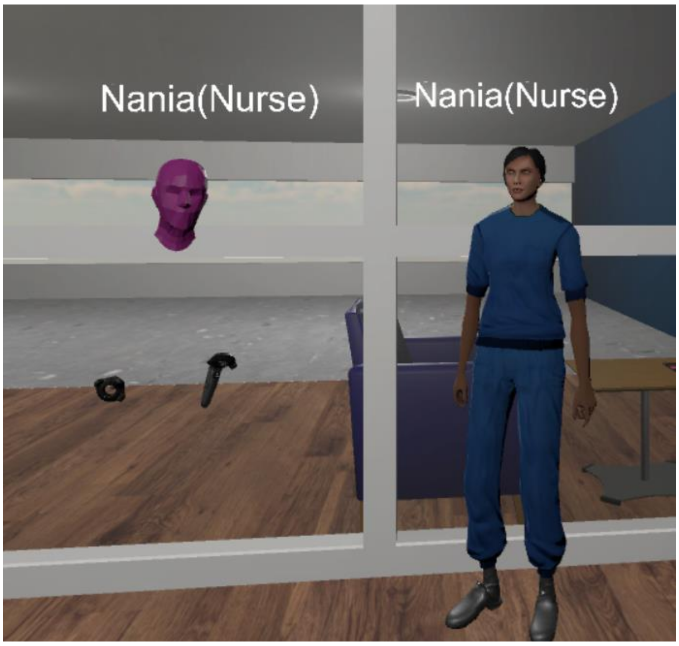


Figure 18: The same role player on different interaction methods. The floating room-scale VR head changes color depending on which role the user has selected to easier separate one room-scale avatar from another.

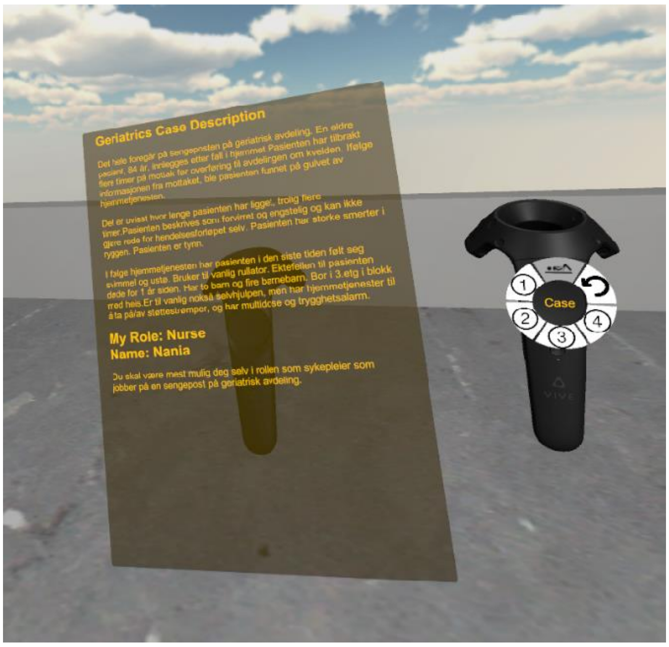
U3: Identity

Each player and their corresponding avatar have a back-story and a name depending on the role they play in the case. In this particular scenario, the identity of the patient and the relative (presumably the patient's daughter) are particularly vital, as it is ultimately their stories that the nurse, doctor, and municipality employee will react to. If the patient for instance states that he can take care of himself and his daughter feels that this is not the case, the other players need to take her advice into consideration.

It is also important to the case that the avatar's appearance reflects the identity. Even though it is not explicitly stated in the case how old the patient's two children are, they would probably be about 25 years younger than he is, meaning around 59 years old. Figure 17 shown earlier displays that the relative of the patient has gray hair, which can be fitting for a 59-year old.

U4: Role

The avatar appearance, identity, and role are all closely related. The role of the character defines what the player should say and do, and defines communication and collaboration patterns. Some tasks, like measuring blood pressure, might be more natural to do for the nurse than for a different role. To keep the role-playing as immersive as possible, it is therefore important that all player perform tasks that are natural to their given role, as this facilitates a more dynamic learning environment. The case description, as well as each player's role in a scene, can be found in the menu. Figure 19 shows how this looks like for a room-scale player.



Geriatrics Case Description

Det hele foregår på sengeposten på geriatrisk avdeling. En eldre pasient, 84 år, innlegges etter fall i hjemmet Pasienten har tilbrakt flere timer på mottak før overføring til avdelingen om kvelden. Ifølge informasjonen fra mottaket, ble pasienten funnet på gulvet av hjemmetjenesten.

Det er uvisst hvor lenge pasienten har ligget, trolig flere timer.Pasienten beskrives som forvirret og engstelig og kan ikke gjøre rede for hendelsesforløpet selv. Pasienten har sterke smerter i ryggen. Pasienten er tynn.

I følge hjemmetjenesten har pasienten i den siste tiden følt seg svimmel og ustø. Bruker til vanlig rullator. Ektefellen til pasienten døde for 1 år siden. Har to barn og fire barnebarn. Bor i 3.etg i blokk med heis.Er til vanlig nokså selvhjulpen, men har hjemmetjenester til å ta på/av støttestrømper, og har multidose og trygghetsalarm.

My Role: Nurse
Name: Nania

Du skal være mest mulig deg selv i rollen som sykepleier som jobber på en sengepost på geriatrisk avdeling.

Figure 19: The case description being read by a room-scale player. The right controller features the menu, and the left controller displays text based on what is selected in the menu. If any of the numbers in the menu are pressed, the description for that scene shown on the screen.

U5: Navigation

For desktop, the player controls the avatar by using either the WASD keys or the arrow keys. A, D, left arrow, and right arrow rotate the avatar, and W, S, up arrow, and down arrow move the avatar in the direction it is facing.

For room-scale, navigation is done in two different ways. The first one is the most intuitive for most people, where the user moves their body in physical space. To illustrate the physical boundaries, a *chaperone* (meaning helper in this context) grid is visible to the user of the HMD or a controller comes close to it (Figure 20). If the user wants to exceed the physical boundaries, teleportation is used.

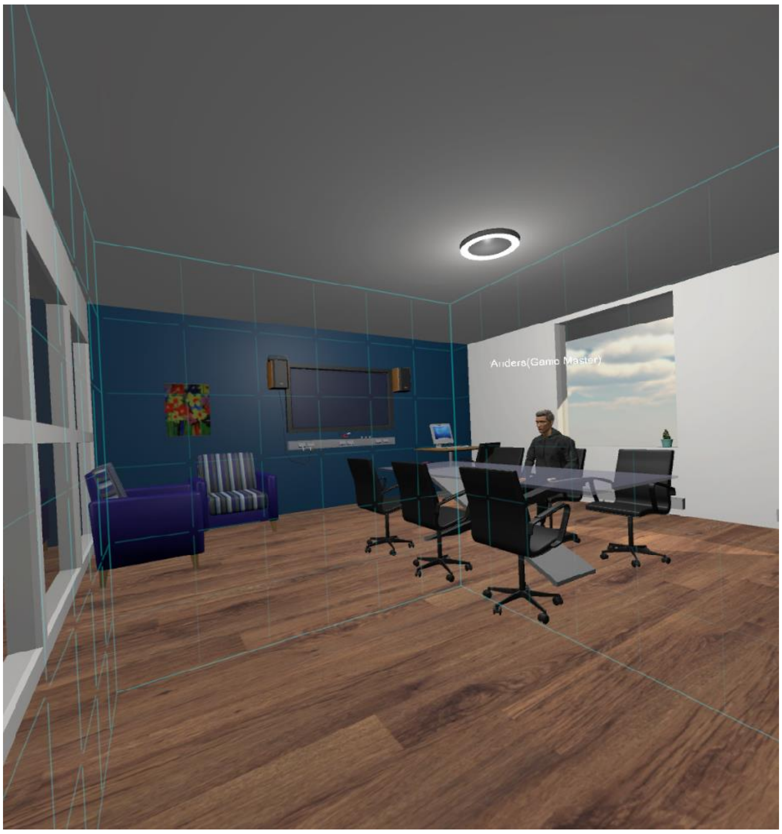


Figure 20: The chaperone grid seen when using an HMD. It only shows up when the HMD or a controller gets close to the boundaries.

Teleportation is a movement type that is very different to what people are used to in real-life but becomes intuitive to use for most people after only a couple of minutes using it. When holding in the touchpad (Vive) or Joystick (Touch), a dotted curve is displayed. When the button is released, the entire play area, including the chaperone grid and the player avatar is immediately taken to the location where the player is aiming. Section 3.1.3 discusses how simulator sickness is a big concern when playing in Second Life. The two interaction methods used by room-scale players (bodily movement and teleportation) impose minimal nausea on the user. The reason is that there is no conflict between bodily senses and perceived changes in the virtual world. Why teleportation works so well is that it forces no acceleration on the user as there is no continuous movement from A to B. The user simply pops up at B instantaneously.

U6: Change View

The “change view” requirement is realized differently for desktop and room-scale players. When playing in room-scale, the view experience mimics what humans are used to in real-life, where turning the head changes the viewpoint. When playing on desktop, the player is presented as an avatar in the center of the screen, and the camera angle can be altered by pressing the left mouse button and sliding the mouse in a horizontal direction. On desktop, the scroll wheel alters the camera zoom for the user to be able to zoom in on details in the scene, or in the opposite case get an overview of what is going on in a room.

U7: Voice Communication

To make voice communication as user-friendly as possible, the default microphone of the operating system immediately starts recording and transmitting audio data when two or more players join the application simultaneously. The audio software has a threshold value, ensuring that a player will not transmit noise when not speaking, however, if the microphone level is set too high, the player will always transfer noise to the other players. To cope with this, the microphone can be muted in the menu if the player is not part of a scene and just wants to observe the role-play. A loudspeaker icon is displayed over a speaking player to indicate to the other players who is speaking (Figure 21). Each audio source also uses 3D spatial sound, meaning a player can locate the direction the sound is coming from. This effect is kept to a minimum though, as it was discovered during testing of VirSam that if this effect is too prevalent, it can be more annoying than immersive. The audio volume can also be lowered if it is set too high.

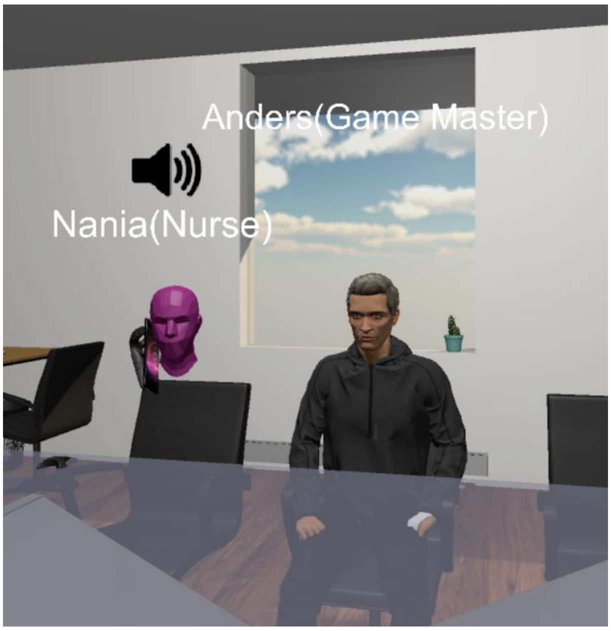


Figure 21: Two players sitting in the meeting room. The loudspeaker icon indicates that the nurse is speaking.

6.1.2 *Place* Requirements

P1: Meeting Room

The meeting room is used to play through scenes 2, 3, and 4. It contains props one normally finds in a meeting room, like a table with chairs and a TV-monitor. The interactive artifacts in the room are all the chairs and a phone lying on the table. The phone can be picked up by room-scale players and is used in scene 3 where the nurse and the municipality employee discuss the situation of the patient over the phone. Figure 22 shows an overview shot of the meeting room.

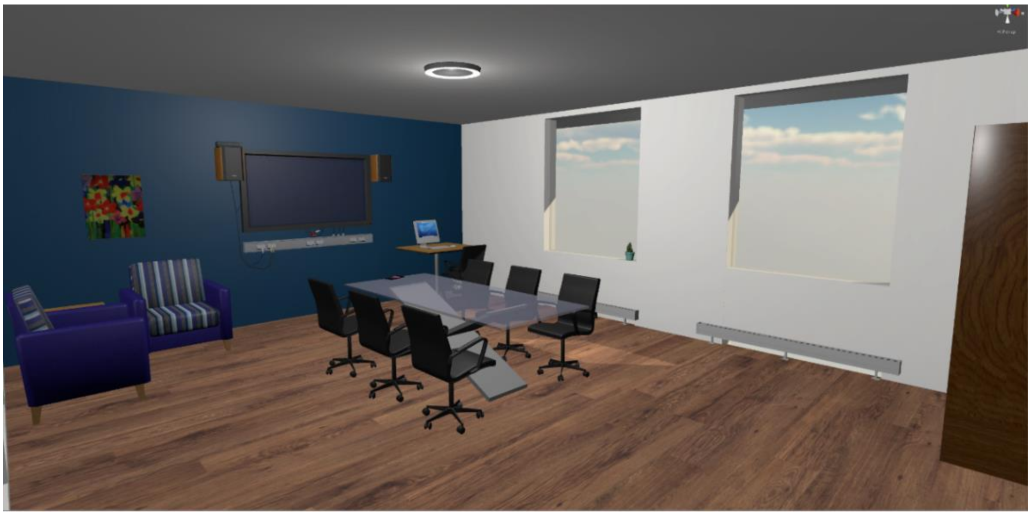


Figure 22: The meeting room. Props that one normally would find in a meeting room create a realistic scene.

P2: Patient Room

In the case deployed in the application, only one scene is played in the patient room – scene 1. During scene 1, the patient is lying in bed, and the doctor and nurse are supposed to do a first assessment of the situation alongside the daughter of the patient. The room contains a patient bed, a set of chairs and medical props to create a more realistic medical environment. Nursing students have requested the option to sit close by the patient, as they would do in a real-life scenario. This can be realized for room-scale players by moving the black chair closest to the patient bed, as this can be picked up and moved around (see Figure 23).

Like mentioned in 3.1, some modifications were done to the case used in VirSam to illustrate the potential of adding room-scale VR to a collaborative setting (RQ1). Thus, a sphygmomanometer and a tablet are placed on the table behind the bed, and a cup is located on top of the sink. The sphygmomanometer, tablet, and cup are discussed further in the artifact requirements section.

Figure 23 and Figure 24 show the patient room of the prototype and a real patient room, respectively. Even though the two rooms look similar, the real room contains a lot more details. This was also pointed out by several of the test users.

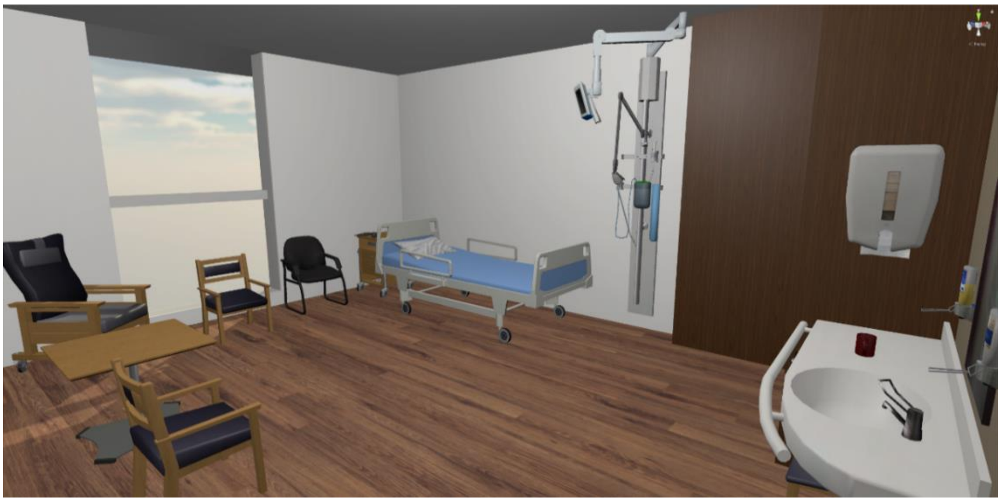


Figure 23: Prototype patient room. Props relevant to a hospital environment like a sink, soap dispensers, and a bed are present in the scene. Room-scale players can move the chair closest to the bed.

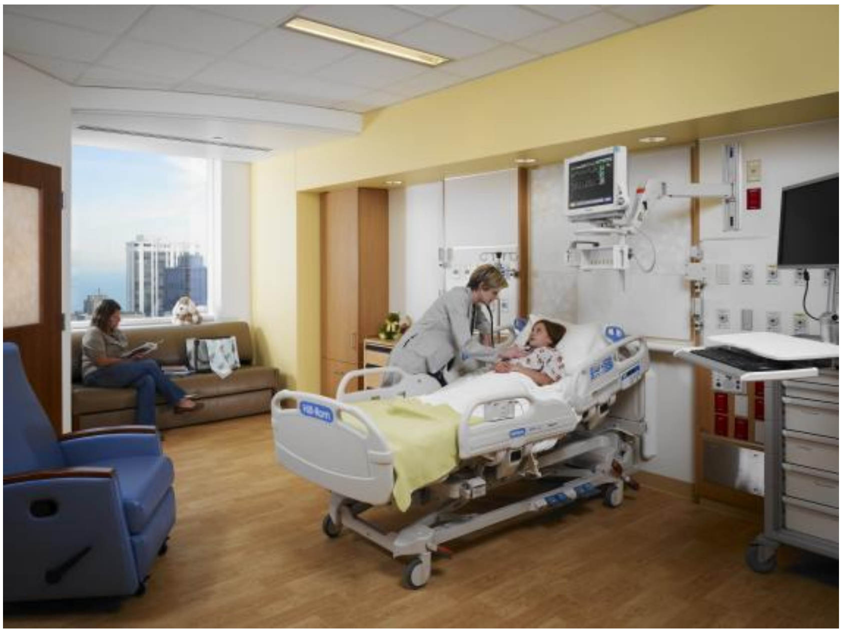


Figure 24: Real patient room. Comparing the real room to the one created for the prototype, one can see that it has a lot more details. Photo:[82].

P3: Hallway

The requirement specification states that the hallway should mainly be a place for spawning and learning the application (3.3.2). To begin with, the players spawned on a balcony outside the building, but after a few iterations with test users and stakeholders, it was clear that a recreational area like a balcony had no purpose, as the users wanted to get straight into the role playing. Because of this, the hallway was enlarged, and all players spawn in the hallway at a distance from where the role-play takes place so that newly joined players do not interrupt any existing role-play. The hallway connects the patient room and the meeting room and contains a phone for the phone call needed in scene 3. With one of the phones situated outside the meeting room, the municipality employee making a call to the geriatrics apartment can simulate the phone call coming from another geographical location. Figure 25 shows what the hallway looks like.

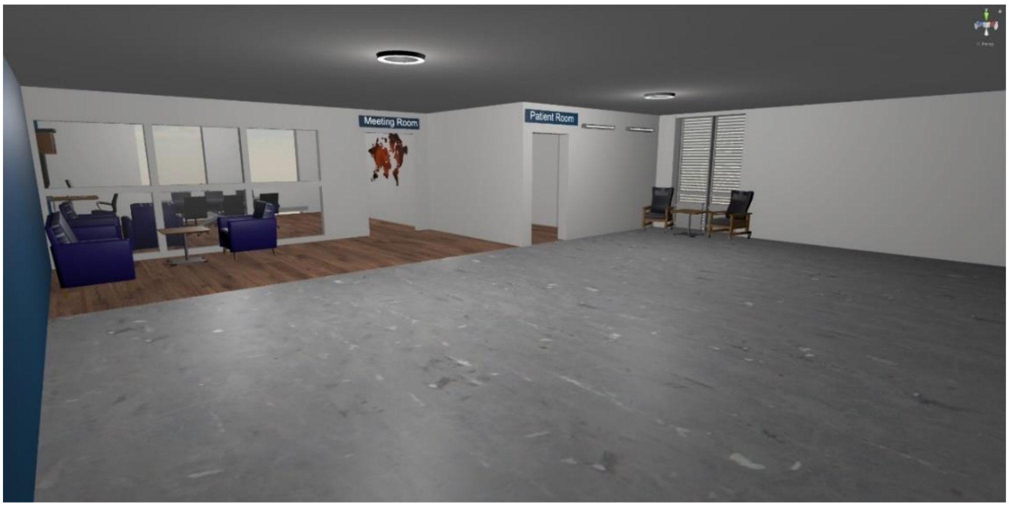


Figure 25: Hallway. The hallway has enough space for all players to spawn safely. It also connects the patient room and meeting room.

P4: Environment

As the requirements specification states, the environment should look like a hospital environment. Figure 25 shown above shows how the hallway is an open landscape with white walls and linoleum flooring, which is typical for hospital environments (see Figure 26). This helps to set the scene of the role-play, giving the players a feeling that they are inside a hospital environment.

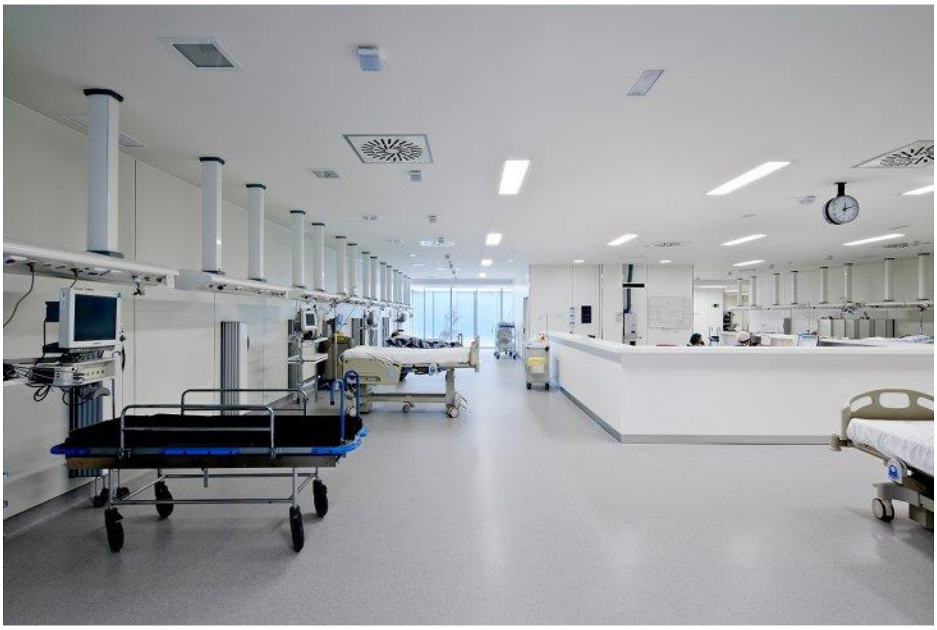


Figure 26: The hospital environment of Hospital Sant Joan de Reus. Photo: [83]

Furthermore, each room has some extra props in addition to the artifacts mentioned in the next section. The props create a complete environment and help towards realism and immersion. A dilemma frequently encountered when creating props for a room-scale application is which props should be grabbable and interactive, and which should be static. Like mentioned in 2.4, immersion and control are both important elements of game flow. Through testing, it has been revealed several times that if an object that one normally could pick up in real-life can in fact not be picked up in the virtual world, the user feels a sense of disappointment. The challenge then arises as to what props should be included and whether all of them should be interactive. Consequently, to not cause distractions for the role-players, only artifacts relevant to the roleplay in the scenes are interactive and grabbable for room-scale players, and small props that a user could potentially try to pick up are kept to a minimum. Big props like tables, chairs, windows a TV, and a shelf are instead included and are static and non-interactive.

In addition to props, several lamps are present in the environment, including a directional light acting as the sun. The light sources create shadows on the floor and walls, which helps improve the graphical realism in the scenario. A skybox (which is a large photo that encapsulates the environment) is used to create an environment outside the hospital.

6.1.3 *Artifacts* Requirements

A1: Collider

For an artifact to function properly in a 3D environment, it needs a collider. A collider is used to determine when two objects intersect. It is then up to the system to determine what to do with the object. If for instance a phone is dropped on a table, the phone should fall towards the table and stop when the two colliders intersect (see Figure 27). Most static objects in the scenes use *mesh colliders.* A mesh defines a 3D object in terms of vertices, edges, and faces, and a mesh collider uses the mesh to check for collisions. Mesh colliders are the most accurate, as they are as detailed as the mesh itself, but a drawback is that collisions can be computationally heavy to calculate if the mesh has many vertices. Thus, only static objects use mesh colliders, whereas interactive artifacts and players use box-, sphere-, or capsule colliders.

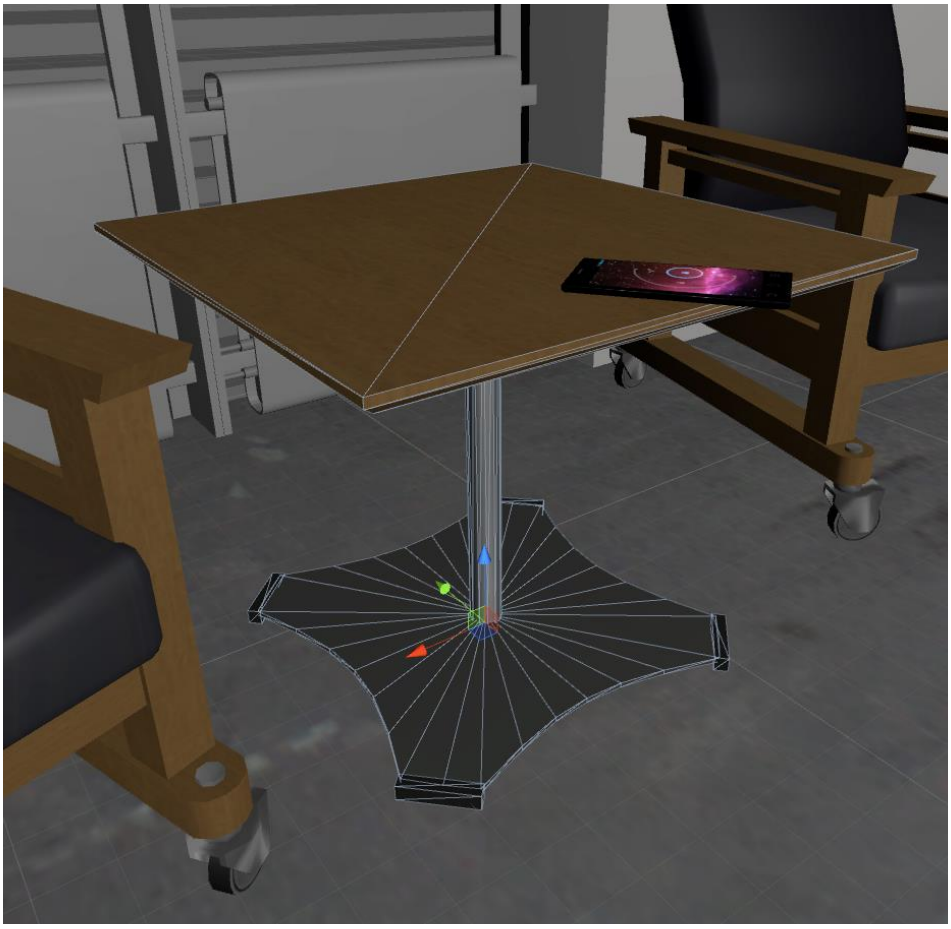


Figure 27: A table and its mesh collider. The blue lines wrapped around the table outline the mesh of the table. The phone stays on the table because both the table and the phone have colliders.

A2: Interactive

Interactive objects are objects that either the user can interact with or objects that can interact with other objects. All the artifacts should be interactive in one way or another. Section 4.2.1 talks about how feedback is valuable to the user for increasing usability. Consequently, all artifacts provide feedback in some way. Desktop users receive feedback when hovering over a chair or a bed in that the object glows bright orange, indicating that it is interactive. The same principle applies to room-scale objects, where an object glows when it is touched by a user (see Figure 28 and Figure 29).

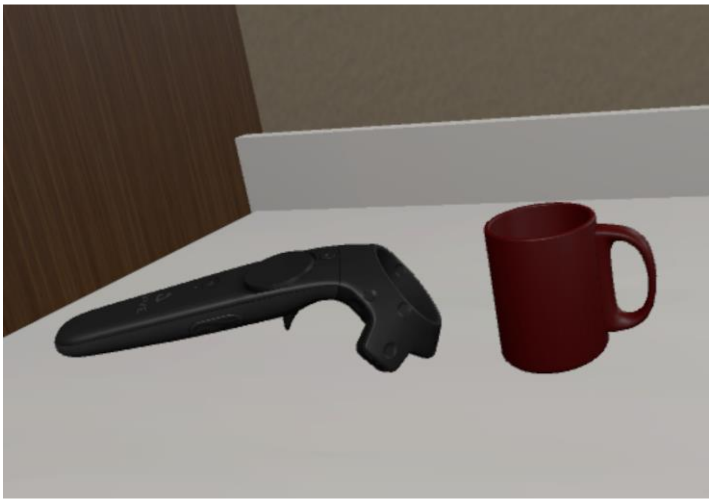


Figure 28: A player controller near a cup. The cup's color is burgundy.

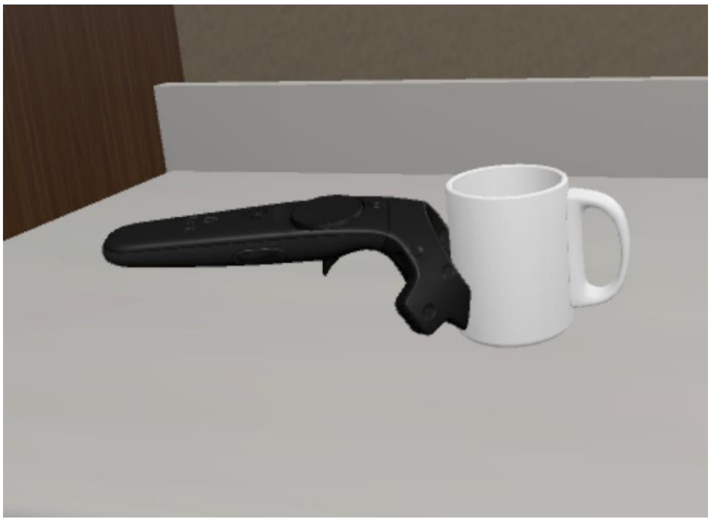


Figure 29: A player controller touching a cup. The cup's color changes to let the user know that it is interactive.

A problem that arises when multiple interaction methods are introduced is how each artifact can provide the same functionality for different interaction methods and devices. RQ3 discusses smart adaptability and advantages and disadvantages of different types of devices in role-playing. In fact, object interaction is one of the biggest challenges connected to RQ3. How is a desktop player supposed to pick up an object like a phone? Room-scale players simply teleport over to the phone and pick it up, but for a desktop player controlling an avatar, more factors need to be considered. The avatar would ideally need to navigate to the phone, then play an animation where the avatar stretches out its arm, bends over, grabs the phone, and finally holds it in its hand. As animations like these are time-consuming to implement, desktop players do not have the option to move or alter artifacts in the in the final prototype other than sitting or lying down. This issue is detailed further in the discussion chapter (Chapter 7).

A3: Chair

Chairs are simple when it comes to functionality and mediate the user's ability to sit. Once a chair is clicked, the avatar is teleported into the chair and the avatar animation changes to a sitting one. To begin with, all chairs could be picked up by room-scale players, but this became a problem when a desktop player wanted to sit in a chair that had been moved by a room-scale player. It follows from the section about interactive artifacts that desktop players have no way of moving a chair back to its original position if a room-scale player were to move it. Thus, all chairs except one are static and cannot be moved.

It follows from conversations with stakeholders that nurses want to be able to sit close to the patient when they are conversing. To realize this, one interactive chair is located close to the patient bed and can be moved next to the patient to increase immersion for nurses playing in room-scale.

A4: Bed

The bed is similar to chairs in that it cannot be moved. When a player clicks the bed, the player avatar animation is changed to lying, and the avatar is placed in the bed. A drop-down menu in the bottom right corner controls the different positions and animations the avatar can use. The patient role description states that the patient should be uneasy and moving. The drop-down menu then makes it easy for the patient to change positions in bed. For the prototype, only two lying states (lying side, and lying back) are implemented, but more positions could easily be added (as discussed concerning character animation in subsection 2.9.3).

In the main menu, before the game starts, the player is urged to play using mouse and keyboard if the patient role is selected, or to play using room-scale of the nurse or doctor role is selected. The reasoning behind this is that it is much easier for a desktop player to get the avatar to lie down in bed. If a room-scale player wanted to lie down, the player would have to physically lie down, meaning the physical space needs to have a mattress or a bed available. This matter is further discussed in Chapter 7.

A5: Sphygmomanometer

The sphygmomanometer is situated on a small table behind the bed. It can be picked up by room-scale players and placed around the patient’s upper right arm. Looking at RQ1, the sphygmomanometer truly shows the potential of combing procedural- and collaborative training. The nurse must move around the room to perform a task while conversing with both the patient and his relative. Figure 30 shows a sphygmomanometer being carried by a player.

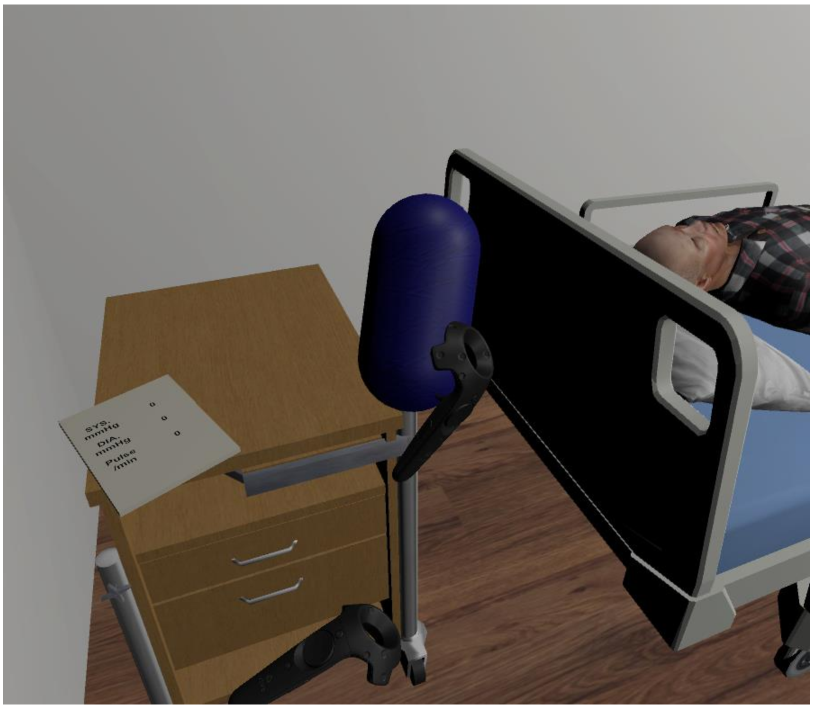


Figure 30: A player using HTC Vive holding the sphygmomanometer. The tablet is located on the table, and the patient is lying in bed, getting ready for the blood pressure sampling.

If the sphygmomanometer gets close to the patient's arm, a yellow placeholder is shown, indicating to the user that the sphygmomanometer will snap into place if dropped (Figure 31). When the device is attached, all players can read the measured blood values on the tablet and utilize the information to act accordingly, thus creating a more dynamic role-play. Figure 32 shows the sphygmomanometer attached to the patient's upper arm. The measured blood values are indicated on the tablet and can be seen by all players.

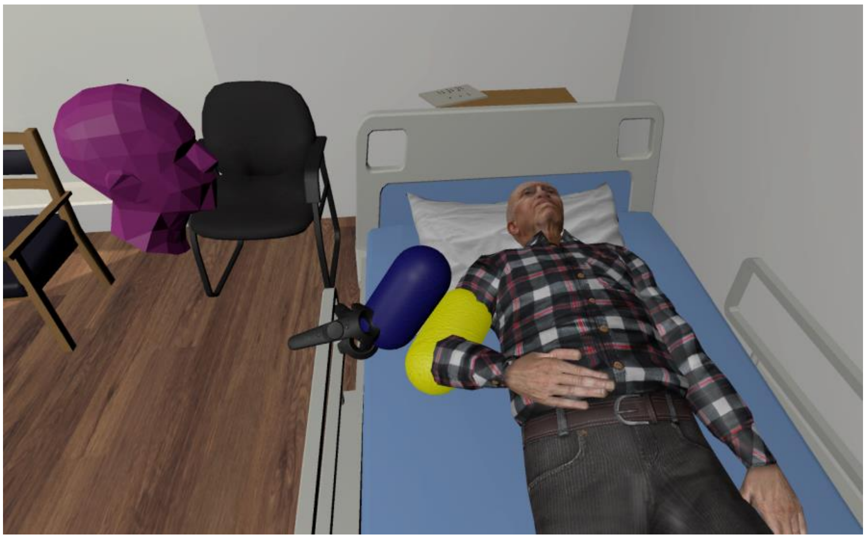


Figure 31: The sphygmomanometer about to be attached to the patient. When the nurse drops the object, it will snap into place and transmit the measured values to the tablet in the background.

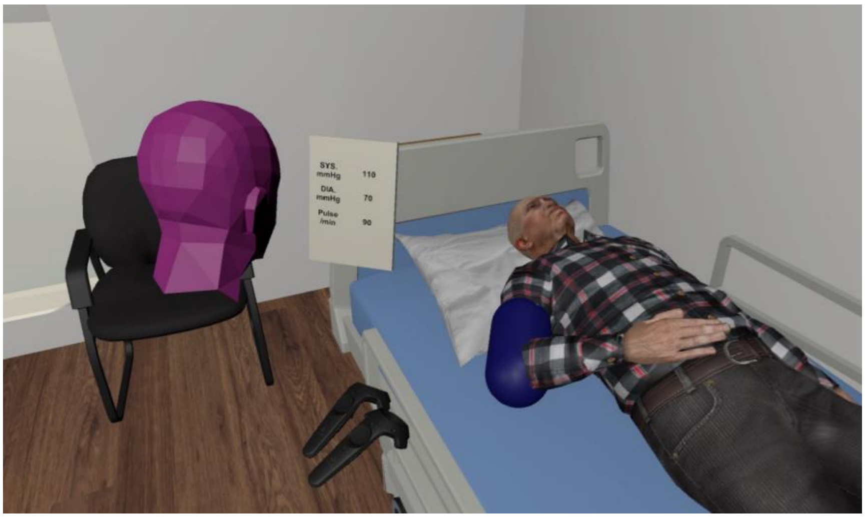


Figure 32: The sphygmomanometer has snapped into place and is transferring its measured values to the tablet. The tablet can be moved around and will not drop to the ground when a player stops carrying it.

Even though the shape of the sphygmomanometer is quite primitive in the prototype, the actions performed are very similar to what would be done in a real-life situation. Many things could be done to increase realism even further though, some of which are discussed in Chapter 7.

A6: Tablet

The tablet allows the players to read the values measured by the sphygmomanometer. A small sci-fi element has been added to the tablet in that it does not fall to the ground like other objects. This way, the user can store information in the environment in ways that are not possible in real life.

A7: Cup

Even though a cup is a primitive artifact, it illustrates that procedural training can be more than intubation and giving anesthesia. A nurse's job is also to provide pre- and post-operation care, and this could include something as simple as to give the patient a glass of water. The patient's role description states that he is thirsty, and if the nurse can physically hand him a cup of water the scenario immediately feels more lifelike to all participants. A cup being held under a tap is shown in Figure 33.

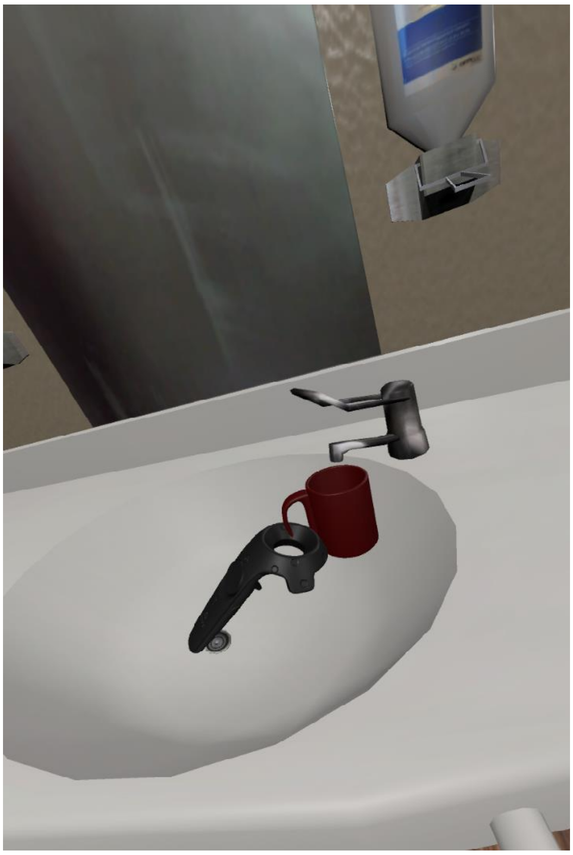


Figure 33: A cup being held under a tap. As the tap has no functionality, the user has to simulate filling the cup.

A8: Phone

The purpose of the phone is similar to that of the cup. Scene 3 of the case is a phone call between two people, and the phone creates an extra element of realism when a player picks up a phone and places it next to their ear. This is illustrated in Figure 34.

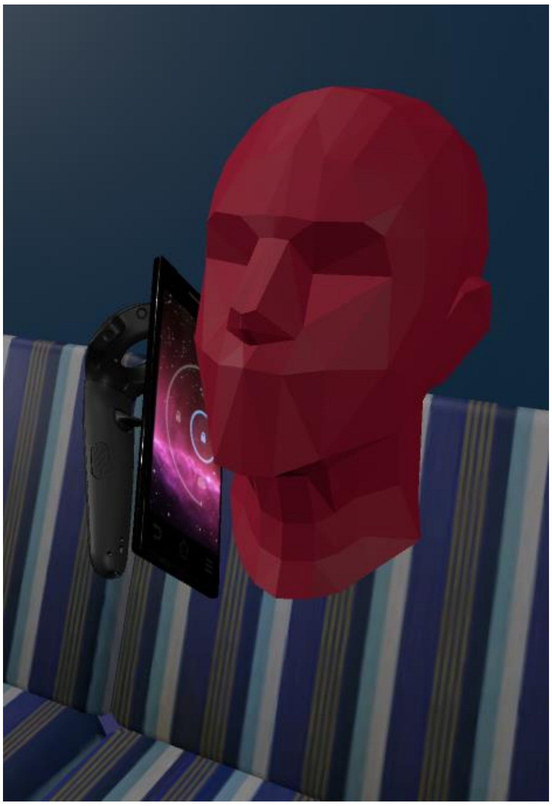


Figure 34: A phone being held by a room-scale avatar. The two phones in the environment are located in different rooms as the phone call happens over a bigger geographical distance.

6.2 Testing Approach

The testing approach done for the final application was set up so that information about each research question could be gathered. The test group consisted of six people (including the researcher), and the five participants all knew the researcher beforehand. A more obvious approach would be to let nursing- and medical students do this testing, but the choice of testers can be justified in some ways. The first reason is that RQ1 is relatively clearly understood from the early phases, including meetings and testing with stakeholders and the pilot test conducted at St. Olavs. Another reason is that the communication during the focus group conducted in each section flows more freely, making it easier to get feedback for RQ2, RQ3, and RQ4. Since the users were not healthcare professionals, and as RQ1 necessarily needs relevant stakeholders to conduct the testing, the findings related to RQ1 during the user testing are not relevant regarding how room-scale can improve interprofessional collaborative training. They are, however, relevant concerning what is added to the scenario regarding realism. Therefore, surveying the requirements, problems connected to supporting different interfaces, and user experience can all be explored by users that do not know the domain that well.

The group was split up into pairs of two, each participant with different computer- and VR knowledge. Each pair of individuals was located at different geographical locations and had either an Oculus Rift with Touch controllers or an HTC Vive at its disposal.

Looking back at the testing methods described in 3.2.1, three key methods were used during the final testing: two surveys, two focus groups, and observation of role-playing. The testing was conducted in two consecutive sessions. In the first session, the prototype was tested by playing through the geriatrics case - both in room-scale VR and on desktop. In the second session, the same case was played on desktop, this time making use of Second Life. After each session, the participants answered a survey and participated in a discussion facilitating the research questions. The researcher was present inside the virtual world during role-playing for observational purposes, but could not intervene if the players were having trouble with the technology.

6.2.1 Session One – Room-scale vs. desktop

The group would first play through the geriatrics case explained in 3.1 using the application. Looking back at the research questions, RQ3 wanted to explore how the solution facilitates smart adaptability as well as advantages and disadvantages of participating in edutainment role-play from different types of devices. This was the main research area of session 1.

To see the advantages and disadvantages clearly, the case was played twice by each user in the same role - once in room-scale and once on desktop. Because the same role was used twice, each user could see and feel the difference between the two interaction methods and how they affected the role play. The most significant results of session one are detailed in 6.3.

6.2.2 Session Two – Second Life vs. prototype

In the second session, the group would play through the same case using Second Life and answer a survey comparing the prototype to Second Life. This session is the main answer to RQ4 as it provides a direct comparison between the two software solutions. In comparing the two solutions, one can also check how well the requirements discussed in section 3.3 have been met (i.e. RQ2).

6.3 Test Results

The test results presented in the following subsections are a concatenation between the survey and notes from the focus group gathered after each session. The survey contains both a series of quantifiable Likert scale answers ranging from 1 to 5, as well as some qualitative questions that seek to get some more general feedback or suggestions. Appendix B and 0 show the complete surveys from session one and two respectively. Figure 25 shows four of the six test users sitting around the table in the meeting room.

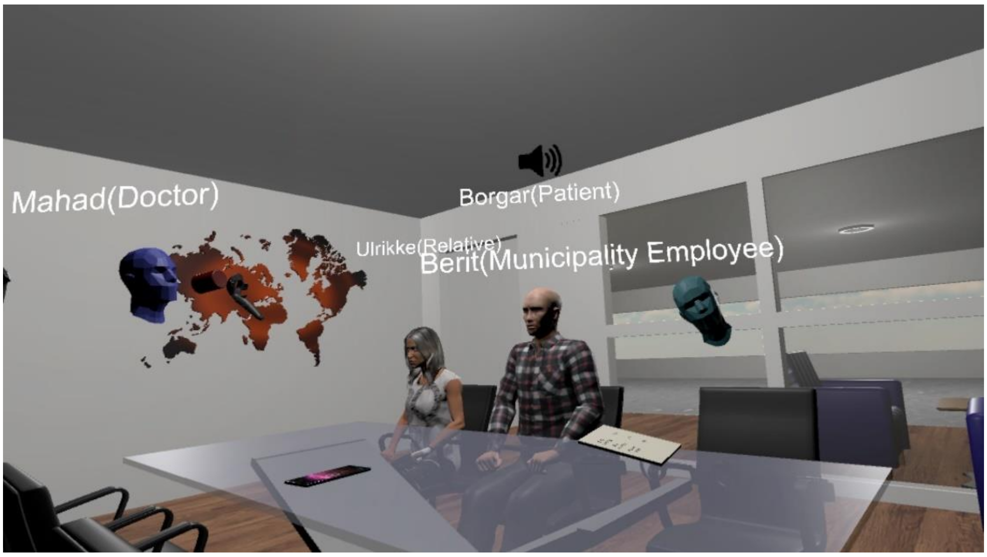


Figure 35: Four players around the table in the meeting room. At this point, two users utilized HTC Vive, one user utilized Oculus Touch, and two players played on desktop.

6.3.1 Session 1 Results – Room-scale Vs. Desktop

Immersion

The players rated room-scale sense of immersion to 4,8 and desktop sense of immersion to 3,5. These results are unsurprising, as many people who try room-scale VR for the first time immediately exclaim: "*Wow, it feels like I'm actually there*." At the end of the survey, the users were asked whether they preferred playing using desktop and in room-scale and also why the preferred that interaction method. The users unanimously answered that they preferred room-scale, and all of them stated the increased immersion as one of the reasons why they preferred it.

Avatars

The most interesting results concerning avatars was that most users wanted the room-scale avatars to have hands. Room-scale hands were researched at the beginning of the project, and several hand models exist free of charge. A problem, however, was that for the hands to seem lifelike, they would need different animations. As explained earlier, animations can be hard and time-consuming to make. Therefore, hands were not implemented.

The test users were also asked how realistic the full-body avatars controlled by desktop players were. The average mark was 3,0, which is an ok score. Nonetheless, the users stated the avatars looked realistic graphically, but that more animations, mouth animations, and facial expressions could take the avatars to the next level.

Case, Identity, and Role

As discussed in the Introduction, one of the biggest issues with VirSam is reading the case information, as it is situated outside the application. This information is stored in the menu both in room-scale and on desktop, and the users were asked how easy it was to navigate to and read this information in the prototype. On desktop, the score was 4,0, and in room-scale, the score was 3,6. During the focus group, it was discovered that the lower score was not necessarily a result of unintuitive VR menus, but rather that the information was a bit hard to read. As Figure 19 shows, the screen displaying the case information has some transparency, and the users wanted the background to be a tad less transparent. That being said, the users who had trouble reading the text were all near-sighted and were not wearing glasses or lenses. VR lenses are powerful and make objects appear as if they are far away even though the screen is just a couple of cm away from the user's eyes [84]. As a result, near-sighted users must use glasses or lenses. One user tried rereading the text after putting on glasses, and then the text was easy to read. This illustrates a need to inform users that vision correction is beneficial.

Navigation and Camera

No players struggled to navigate to the desired location either on desktop or in room-scale, though desktop scored a bit higher. However, a want for rotating the room-scale avatar without moving the body was expressed by three users. It is debated whether this is functionality that should exist in a room-scale application, and this is discussed further in Chapter 7.

The players also answered that it was easier getting a view of the important things in the scenes by moving the camera in room-scale than on desktop. The camera used by desktop players can only be controlled horizontally and cannot zoom all the way in, which some users stated made it difficult to observe details in scenes.

Rooms and artifacts

When all users got to role-play the same role using the two different interaction methods, some advantages and disadvantages of both were clearer. The users stated that the artifacts needed to play through the case were present. However, the issue discussed concerning A2: Interactive in section 6.1.3 arose quickly. This issue expresses a challenge for realizing smart adaptivity (RQ3) in that giving all different interaction methods the same possibilities might not be possible, maybe not even desirable. As desktop players have no way of interacting with for instance the sphygmomanometer in this prototype, a nurse playing on desktop is not able to perform the *activity* of measuring the patient's blood pressure. Furthermore, sitting down is simple when playing on desktop, as the user just clicks a chair, and is immediately teleported into it. In room-scale, however, the user controls the avatar with their body, which means that a user trying to sit down will fall to the ground. One user teleported to a chair and kept standing while others teleported to a virtual chair and then sat down in a physical chair. This resulted in the avatars having different heights.

Some other wants the users expressed were doors, a working sink, an environment outside the window, and the ability to greet desktop avatars properly. Some of these topics are detailed further in Chapter 7.

6.3.2 Session 2 Results – Prototype vs. Second Life

Immersion

In the survey, the players were asked to rank how immersed they felt in each of the three gameplay styles (prototype room-scale, prototype desktop, and Second Life non-VR). All players stated that room-scale was way superior to the others concerning immersion and sense of presence.

Furthermore, four out of five players ranked prototype desktop higher than VirSam regarding immersion which is interesting when most users stated that Second Life was better regarding props, environment, clothing, and animations.

Avatars, Graphics, and Realism

The users concluded that the avatars in Second Life were more relevant to the case in terms of clothing, but that they looked graphically more realistic in the prototype. During development of the prototype, a search for models fitting the case well was conducted. Even though fitting models were discovered quickly, most of these cost between 50 EUR and 500 EUR, which are stupendous amounts of money to spend on just one model for a prototype. As a result, some of the models used were free, while others were included in the two hospital environment asset packs mentioned earlier. This shows that an adequate end product can be achieved when on a budget, but that to create a state of the art solution, the project needs a bigger scope or more funding.

Regarding animations, all users felt the avatar mouth animations when a player was talking in Second Life made the character much more lifelike. The same was discovered in the early stages of the project but was deprioritized due to other aspects needing more work. Additional and better facial- and body animations are undoubtedly aspects needing more work in the future.

Most users also stated that the environment looked more realistic when it comes to props in Second Life. The VirSam environment in Second Life has more props, and the feeling one can get when playing the prototype is that it feels a bit empty. A simple solution is simply to add more props, but then again it is important to keep the issues explained in 6.1.3 about interactive artifacts in mind - that a user can feel disappointed if a prop is not interactive. Also, too many props or interactive items not relevant to the scenario can be a source of distraction.

Several users also stated that the framerate in Second Life was very low compared to the framerate of the prototype and that Second Life felt “choppy” and slow. The test users utilized computers varying in performance - from high-end gaming machines to a five-year-old midrange laptop. Even the gaming machines had low framerate, which is an indication that the Second Life game engine is probably not very well optimized for newer graphics cards. This is unsurprising though, seeing that Second Life is nearly 15 years old.

Case, Identity, and Role

When playing Second Life, the users got to experience the issues concerning reading case information and role descriptions. It was earlier stated that the users of VirSam typically meet up at the same geographical location even though it is an online application. In this case, the case information is printed out on paper for the users, but to realize a Virtual University Hospital with smart, adaptable places and different devices, printing the case information is obviously not a viable solution. All users stated that having the case information inside the application solved this issue, though two users expressed that they wanted the information even more accessible. For desktop, this could mean adding a button called “case” to the in-game UI. In room-scale, a nice suggestion was that the user should have the option to keep the scene information visible even when the menu is closed.

Navigation and Camera

Concerning navigation, all users but one ranked the prototype easier to use than VirSam. The last user liked how one in Second Life can click on the floor, and then the game automatically navigates the avatar for the user to that location.

The users rated the overall camera experience quite similarly in Second Life and the prototype, but three users wanted to have first-person view in the prototype. The first-person camera in Second Life works quite well, and all the user has to do to activate it is to zoom all the way in. The mouse then controls both the avatar head and the camera. This makes it possible for the player to look at other avatars when conversing, increasing realism through more realistic social interaction. One user also expressed a want to tilt the camera vertically as well. This is something that was researched in the early stages of the project, but proved difficult to do and was ultimately scrapped.

Usability and Sound

The Architecture chapter talks about the usability issues in VirSam, something that has been a key point during the creation of the application and its architecture. The users were asked which application was more user-friendly, and everyone answered that the prototype was better in most ways. The menus were clearer and simpler, the case information was much easier to find, and the time it took after the players joined to the role-playing began was much shorter. Second Life was too big for the users, and they stated that having to walk a long distance just to get to where the case should be played was annoying and confusing. One player states that:

> *VirSam was "too" big, and you had to pass areas unrelated to the case to get to areas of interest. I didn't really know exactly what was going on when I first entered VirSam, but that might change if I play it again.*

This answer illustrates the point that Second Life might work for users that have time exploring the interface and functions thoroughly, but for a user that is going to use it once or twice, it is too big.

The users also said that the sound quality was better and clearer in the prototype than in Second Life. As voice communication is such an integral part of the role-play, it is important that the sound quality is top-shelf. In the prototype, the microphone is transmitting by default and can be turned off in the menu, whereas in Second Life it is the other way around. When transmitting by default, the application is simplified even further.

Overall Experience

All users preferred the prototype to Second Life. The consensus was that the room-scale experience took the case to a new level and that the desktop experience in the prototype and Second Life was about the same overall. The desktop experience felt simpler and smoother in the prototype, but the environment and animations were more finished in Second Life.

The users also provided feedback on what needed the most improving, both in room-scale and on desktop. This is a subject for future work.

7 Discussion

This chapter discusses the research questions based on findings in Chapter 0 as well as findings discovered during the development and testing of the prototype.

To reiterate, the goal of the thesis is to investigate how room-scale virtual reality can be combined with traditional interfaces to facilitate collaborative interprofessional healthcare training. The first research question has a lot in common with the thesis goal and seeks to survey how room-scale virtual reality can improve upon interprofessional collaborative training by enhancing procedural elements (RQ1). Since a new software platform was needed to explore room-scale virtual reality, a prototype application supporting multiple interfaces has been created using a game engine, and the requirements of the application and the game engine have been detailed (RQ2). For the application to be used in a future smart virtual university hospital, it needs to facilitate smart adaptation, which imposes some challenges concerning how virtual simulation can be achieved from different interfaces. This is the core of (RQ3). Finally, a user test surveying how the prototype application compares regarding user experience to the existing solution (VirSam played in Second Life) has been conducted (RQ4).

7.1 Introducing room-scale in Interprofessional Collaborative Training

7.1.1 Improving Procedural Elements

As stated in RQ1, medical- and nursing students should ideally practice communicating with patients, each other, and other professionals, meanwhile performing procedural tasks as this ultimately will be a part of their everyday activities when their studies are concluded. RQ1 asked how room-scale virtual reality can improve upon interprofessional collaborative training by enhancing procedural elements. As section 6.2 discusses, the findings concerning RQ1 has mainly been researched through prototyping and early testing, and that the results from the final testing are not conclusive in terms of how improving procedural elements can enhance healthcare education. The test does, however, give an indication of how room-scale virtual reality can improve upon procedural elements by allowing natural body movement.

The sphygmomanometer shows the potential of how room-scale virtual reality can improve procedural elements. During Session 1 of the final testing, all users stated that they preferred room-scale to desktop. The main reasons for this were that the experience was much more immersive and that interacting with the objects was intuitive and lifelike. Prasolova-Førland et al. (2016) state that:

> *Increasing the realism of interaction with the environment, other characters and equipment, e.g. through motion capturing devices and virtual patients/agents. Facilities for communication and collaborative work, workspace awareness, also in VR.*

are some of the needed in smart education in a smart virtual university hospital [2]. From examining the requirements of the final prototype, it is evident that both interacting with other avatars and interacting with equipment is realized. Requirement A5 shows how users can pick up and attach the sphygmomanometer to the patient. The fact that players can communicate (U7) and move around (U5) in an environment set in a medical setting (P4) while using their hands and body to manipulate procedural elements such as the sphygmomanometer (A5), tablet (A6), cup (A7), and phone (A8), indicates that room-scale VR can facilitate both procedural and collaborative training simultaneously. The artifacts can even be handed over to other room-scale users, again demonstrating interaction between characters.

The activity of measuring blood pressure by performing procedural actions is an immense improvement over just telling the other users what you are doing – which is the way it is done today [2]. However, there is still much room for improvement. To begin with, the patient's sleeve cannot be lifted by either the nurse or the patient, and measuring blood pressure through a shirt is not the correct way of doing it. The look of the sphygmomanometer is also very primitive, so creating a cloth-like sphygmomanometer that needs to be attached to the patient's arm instead of just snapping to it would be more realistic. Unity has a physics-based solution for simulating fabrics called Cloth [85] that could be used, but this is significantly more complex than using a simple shape like the capsule that is used today. Additionally, since the application supports sound, the sphygmomanometer could potentially simulate Korotkoff sounds [86] that nurses or doctors listen to when measuring blood pressure instead of just displaying the values. This would add another layer of realism as the user would have to perceive the aural environment in addition to the visual.

7.1.2 Room-Scale Avatars Not Having a Body

Another thing most users mentioned in session one of the final prototype testing was that the room-scale avatars did not have a body. One user thought there was something wrong with the application until another user told her that it was meant to be like that. The introduction mentions that the patient is the most important element in medical education, and if the patient were to play using room-scale, it would have no body in the prototype.

The first step in giving room-scale avatars more realistic bodies is probably giving them hands. This is something that most of the test users felt was missing, and as Oculus has depicted controllers as hands for a long time, and as Steam recently showcased a new controller for HTC Vive depicted as hands, this is likely a natural direction to go in. As previously stated, this was dropped due to the complexity of animations, but even stiff and static hands might actually make the avatar feel more like a real person and should be considered in the future. Conversation with stakeholders has also revealed that hand movement and gesticulation is such an integral part of a nurse's or doctor's everyday activities, that an animated hand would probably feel more natural to them.

A question then arises as to what the controller should look like. The introduction states that room-scale VR is a mapping between reality and virtual reality. When the controllers are shown in VR, it may be easy to find the correct button as the physical and the virtual representations are the same, but if the controller is depicted as a hand, this mapping is no longer the same.

Section 2.8 listed some different virtual reality hardware, and as stated about HTC Vive, this was the solution that started the interest concerning room-scale VR. Vive controllers are held much like sticks (Figure 36), and through testing, many users have shown that the grip button on the sides, which is intended to use for picking up objects, is hard to press in and not intuitive, to begin with. One reason to this might be that the user is already holding the controllers and it is not intuitive to grip and hold an object when you are already holding an object. The Oculus Touch controller was released by HTC's competitor roughly half a year after the HTC Vive, as Oculus also saw the potential in room-scale VR. This controller addresses some of the issues with the stick design of the Vive as the user's hand is more intertwined with the controller. Because the user's hands naturally rest inside the Touch controller, and as each button has capacitive sensors that know whether a finger is touching it or not, this controller is better suited for depicting the user's hands. This is illustrated in Figure 37.

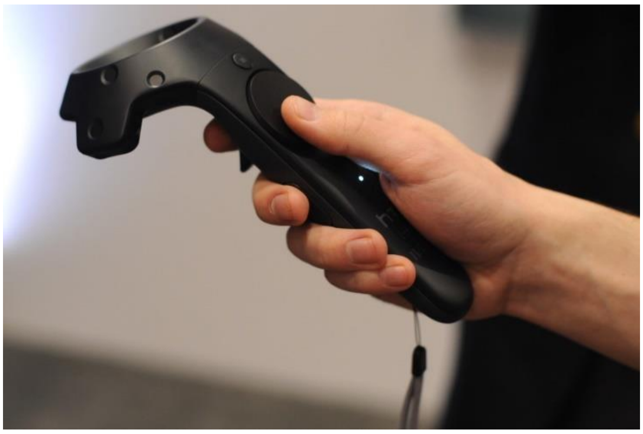


Figure 36: A user holding the HTC Vive controller. The area where the user naturally rests the palm has few buttons, and it is, therefore, difficult to portray the controller as a hand in the virtual world. Photo: [89]

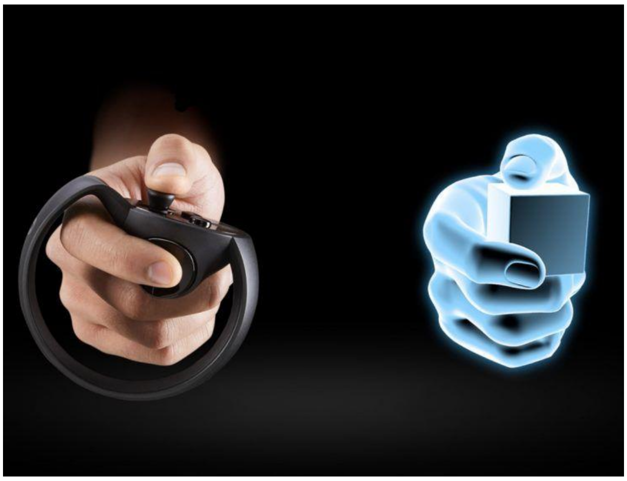


Figure 37: Oculus Touch being held by a user. This controller is easier to map to virtual, animated hands as the user's hands naturally rest inside the controller and all buttons have capacitive sensors. Photo: [88]

Many would say that the Touch controller is superior to the Vive controller for smaller and more precise tasks, but one still needs to hold the controller, meaning that the issue explained above concerning gripping still exists. In June 2017, HTC revealed several details on a new pair of controllers called Knuckles [87]. These controllers take room-scale hand experience one step further as they are not dropped when the user lets go of the controller (Figure 38). Both the grip and the top touchpad has sensors that know when they are being touched, which allows for accurate finger tracking. This may be an even more intuitive and lifelike controller than the Touch controllers and illustrates that producers are constantly pushing more products out to consumers that can improve upon realism, but which also means more interfaces need to be supported by developers. It is therefore up to developers to come up with a general solution that fits most new devices. This is the reason why the OpenVR API mentioned in the prototyping chapter is used – in theory, it supports all virtual reality hardware instead of only supporting their own product (HTC Vive).

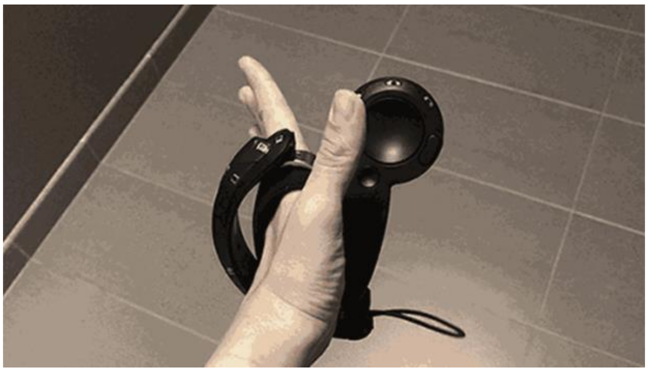


Figure 38: The upcoming Knuckles controller from HTC. The user can release the grip of the controller, and the controllers stay attached. Photo: [87]

7.2 Prototyping and Requirements

7.2.1 *User-Place-Artifact* Framework and Multiple Interfaces

Since the case to be implemented was known early in the process, the *user-place-artifact* framework detailed in 2.7 was conducive to development. As the framework revolves around the activities that users can perform instead of technological details, the requirements specification, therefore, functioned more like a “goal” wanting to be reached, instead of providing technical details, which was well-suited for an iterative development process with a lot of prototyping and testing [24] . This means fewer resources were spent defining platform-specific requirements and more time was spent developing the activities needed to realize the case.

However, the fact that the *user-place-artifact* framework form requirements based on activities can also be a challenge. As mentioned in section 6.1.3, some requirements (particularly the interactivity of artifacts) are realized differently for different interaction methods, and some requirements are realized for one interaction method and not for the other. The framework was well-suited for surveying the overall functionality needed to facilitate the activities of the case, but as some interfaces are so different, the requirements specification may, therefore, fail to capture the essence of the activity. Requirement U4 states that activities should be performed based on what role the user has in the scenario, and requirement A2 states that all artifacts should be interactive. A result of all requirements not being fully realized for all interaction methods is therefore that if a nurse or doctor play on desktop, they will have to explain their actions orally (like in VirSam) instead of actually performing them. A2 in section 6.1.3 explains how objects could be manipulated on desktop using animations, but an even simpler approach is to click the object, and then the object is automatically teleported into the player's hand. A predefined area could then be highlighted (much like the sphygmomanometer highlighter shown in A5) to indicate where the object can go, and if this area is pressed, the object is teleported to the predefined area. The main reason why this was not implemented is that the scope of the project only allowed time to implement certain things, and as room-scale VR being integrated into existing environments is the main research area of this thesis, this was prioritized instead.

7.2.2 Possible Additions or Improvements

One thing several users missed in the prototype was the ability to zoom the camera all the way in and view the environment in first-person. When this is done, the camera can rotate the avatar head, which makes looking at other avatars more immersive – both for the user looking around and for the user being looked at.

7.3 Smart Adaptability and Different Devices

7.3.1 Should All Interfaces Support the Same Activities?

A question arises from section 7.2.1 whether it is a problem that some activities are realized differently on different devices. As stated earlier, the purpose of VirSam is to allow 4th year medical students and 3rd year nursing students to practice cross-professional collaboration to prepare for clinical practice in a virtual world. Section 2.1 mentions that *role-plays* is the main identified activity of VirSam, but also states that *patient simulation* and *procedure training* are important in a virtual university hospital. To know what should be the next step of the prototype, one would have to analyze which of these activities, including the content, facilities, and technological solutions are the most important for facilitating the goal of VirSam.

Section 2.2 briefly mentions what features of a smart virtual university are affected by adaptation. The people behind VirSam are in the process of realizing the feature *general hardware support* at St. Olavs University Hospital, as they are installing several VUH labs with Oculus Rift and Touch controllers. This could mean that there are always users role-playing in the scenario using room-scale interaction. Whether to focus on supporting many interfaces and interaction methods or to improve upon the room-scale experience by giving each user a more realistic body in the future, is not clear. This is a subject for further discussion.

7.3.2 Dynamic Scenarios

Section 2.2 lists *scenario building* as a feature relevant to this project. The scenario building feature talks about how the scenario could change for each playthrough – both through randomization of values and by letting the users control the scenarios more freely. For this project, some changes that could make the scenario more dynamic could be that the user decides the name of the avatar and that the user can choose the gender of the avatar. This would make it easier for a user to "create" an avatar identity that is more closely related to their own, making role-playing more fluent. Also, values shown by medical equipment such as the sphygmomanometer do not have to be predefined. The number generating code for the blood pressure values was randomized within certain limits to begin with, but was later changed to fit the description of the case. If instead all values used in the case were generated for each play-through, the cases could feel much more dynamic and lifelike.

7.4 User Experience

The introduction chapter and section 3.1.3 detail many of the issues related to role-playing the cases of VirSam in Second Life. Providing a better user experience than the one provided by Second Life has been important throughout the course of this project, as reflected by the research conducted in the Architecture chapter. The test results from Session 2 gives a direct comparison between Second Life and the prototype, and all users state that they prefer the prototype to Second Life. The main reason for this was the increased immersion that the room-scale experience provided, but the desktop experience in the prototype was valued about the same as Second Life.

7.4.1 Usability Improvements

Chapter 4 lists a number of architectural tactics for increasing usability. Early identification of usability problems in Second Life was vital to understanding how things could be done differently in the prototype. By giving users feedback and increasing learnability and efficiency by keeping the user interface simple and intuitive, the usability in the prototype is way higher than in Second Life. Most users stated the sound quality was better and clearer in the prototype, which is essential in a role-playing setting. The case information is also readily available inside the application for all three interaction methods (even though some users stated that it should be even more accessible), which is a significant improvement over Second Life which has no way of displaying the case information. As mentioned in section 2.4, if certain factors are considered during development, the flow of the application will increase, ultimately increasing user satisfaction. Since the case information is now available inside the game, goals are much clearer. Some of these elements are closely related to the architectural tactics used for increasing usability, and as the results from Session 2 shows, the users appreciated the simplicity and efficiency of the prototype, even though the environment was more complete in Second Life.

7.4.2 Virtual Reality and Nausea

Prasolova-Førland et al. (2016) state concerning simulator sickness that:

> *This is a practical problem that must be solved to be able to take full advantage of the higher level of immersion. The 'cyber sickness' is likely to be remedied by using the new consumer versions of the VR goggles, rather than the development kits we used, which includes better tracking solutions. [2]*

During the development and testing of the Xbox-movement paired with an HMD discussed in 3.2.4, it was discovered that inducing simulator sickness for users is very easy even on the consumer version of the Vive and Rift. Oculus lists a number of factors contributing to simulator sickness [55], and when some of these were addressed (like decreasing movement speed and acceleration), the problems decreased dramatically. Even though the specifications of the hardware undoubtedly is a contributing factor regarding simulator sickness, this could indicate that optimizing the software solution is even more vital. Through the course of the project it had also become apparent that room-scale VR imposes little to no simulator sickness on users, which illustrates that when there is no conflict between visual and bodily senses, the problems relating to nausea disappear.

8 Conclusion

This thesis has explored *how room-scale virtual reality can be combined with traditional interfaces to facilitate collaborative interprofessional healthcare training*. The exploration has been conducted by exploring areas where room-scale VR and collaboration is used today, and most importantly through development and testing of a prototype application.

8.1 Using Room-Scale to Improve Procedural Elements in Virtual Worlds

Through background research, prototyping and testing, this thesis has illustrated how room-scale virtual reality can improve upon interprofessional collaborative training by enhancing procedural elements. Several artifacts like the sphygmomanometer and tablet can be picked up by users and moved to different locations, mimicking the actions that healthcare personnel would do in a real-life situation. This clearly contrasts the existing solution played in Second Life where students must explain their actions orally to the other users. The most widely used room-scale solutions also have headsets and microphones integrated, meaning users can practice collaboration and communication while performing these tasks. Some challenges related to online room-scale avatars is how they should look. Today's widespread room-scale solutions typically only track the head and hands, meaning that the system does not know where the user's body is. Because of this, room-scale avatars either show a floating head and arms, which means they are usually less lifelike than full-body avatars. The case employed in the prototype illustrates that the patient needs to be full-body for the sphygmomanometer to attach to his arm. It seems evident that room-scale provides a better experience overall, which is also stated by the test users, but it might be challenging to support interprofessional collaborative training if all players are using room-scale.

8.2 Developing an Application for Several Interfaces in a few Months

Because the application should support several interfaces and since only a few months can be spent on development, the game engine, paired with rapid prototyping and a lot of testing, made sure that most of the requirements were met. Even though Unity supports many different interfaces and target platforms, it still takes time to customize the user interface to all platforms. Initially, mobile solutions were also going to be a part of the project, but this was scrapped quickly, as challenges related to the other interaction methods (especially Xbox-controller + HMD) proved time-consuming to cope with. These challenges could indicate that the project scope was too big, but since all users preferred the prototype to Second Life, this also shows that supporting numerous interfaces is possible if a game engine facilitating rapid application development is used. Unity with its modularity and "prefab" mechanisms has shown that one does not need to create an entirely new application if one wants to support several interfaces and that many elements can be reused, shortening development time.

8.3 Smart Adaptation of Interfaces is Challenging

As new interfaces emerge every day, it is a big challenge supporting them all, and knowing which interfaces are best suited for different activities is not always easy. It is therefore important to develop in a modifiable way so that when a newer and better product arrives, it can quickly be integrated into the system. Steam has probably also realized this, and that is why they want to support as many VR interfaces as possible by making the SteamVR SDK available to everyone. If the SteamVR SDK was locked to the HTC Vive, creating an application that supports both HTC Vive and Oculus Rift would not be possible in the short timeframe of this project.

As stated in the discussion, some requirements are realized differently for various interaction methods, and a question for further development is whether all interaction methods should facilitate all requirements equally, or whether it is better to limit some interaction methods to some activities and roles of the case.

8.4 Early Identification of Stakeholder Needs is Key to Providing a Good User Experience

Because several meetings with stakeholders were conducted in the beginning, it quickly became apparent that the existing platform suffered from significant usability issues. By surveying these issues through testing and related work, some architectural tactics could be employed in the creation of the architecture to facilitate both end users and future developers. Through testing, it is evident that the prototype facilitates usability much better than existing solutions. The modifiability tactics employed should also make development for future developers less tedious and more efficient, but this remains to be seen.

8.5 Future work

From the things discussed in this chapter, as well as the previous chapter, some directions for future work spring to mind. To structure the directions in which the prototype can evolve, Table 8 lists future directions that the application can involve in, some upsides and implications of said direction and which key identified smart virtual university hospital features are identified for each direction. The project could evolve in one of two major directions. It can either support all kinds of devices better or focus solely on room-scale and animations. An assessment would need to be done on what is more important; supporting various interfaces, meaning the application can reach a bigger audience, or improving upon the room-scale experience and making more realistic avatars that can be interacted with.

Table 8: Relevant future work for the prototype application

Future Direction	**Upsides**	**Implications**	**Key identified SVUH features**
All interaction methods realize all requirements	Anyone can join from anywhere with the interfaces they have available. Mobile solutions (including VR) would increase the availability and user base.	More resources spent on development and testing. Quality assuring of scenarios that only have one interaction method present. Some users may get a better experience than others.	Personalized interface and interaction. Mobile, ubiquitous and augmented.
Limiting certain interaction methods to certain roles	Minimal changes to current prototype are needed. Ensuring all users get the same experience. More time improving the existing solution.	Must be done at a centralized geographical location so that the users can rotate through interaction methods and get to role-play a role relevant to their study.	General hardware support.
Limit to room-scale only	More immersive, fun, and realistic. Much better procedural training.	Must be done at a centralized geographical location as most people do not have room-scale equipment. More resources spent on making room-scale avatars realistic and interactive (i.e. body, hands, clothing, facial expressions.) Find a good way of lying down in room-scale or implementing virtual agents that can alleviate	General hardware support. Virtual agents.

		tasks difficult to perform in room-scale (e.g. lying down).	
Improving the geriatrics case	Removing the unwanted "game master" role and moving his role information into the environment or to other relevant avatar roles. More interactive artifacts can facilitate more activities.	Must create a system for storing information in the environment or on each user.	Scenario Building. Personalized learning path. Individual Feedback.

9 References

[1] *Mark Zuckerberg*. (2017). *Facebook.com*. Retrieved 20 June 2017, from https://www.facebook.com/zuck/posts/10101319050523971

[2] Prasolova-Førland, E., Steinsbekk, A., Fominykh, M., & Lindseth, F. (2016). Practicing Interprofessional Team Communication and Collaboration in a Smart Virtual University Hospital. Trondheim, Norway.

[3] Nester, J. (2016). The Importance of Interprofessional Practice and Education in the Era of Accountable Care. NCMJ vol. 77, no. 2.

[4] Guze, P. (2015) Using Technology to Meet the Challenges of Medical Education. Trans Am Clin Climatol Assoc. 2015; 126: 260–270.

[5] Snowdown, D., Churchill, E.F. & Munro, A.J. (2001). Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction. In Churchill, E.F., Snowdown, D., & Munro, A.J. (eds.). Collaborative Virtual Environments, Chapter 1, pp. 3-17. Springer-Verlag London Ltd.

[6] *VirSam*. (2017). *Virsam.no*. Retrieved 15 June 2017, from http://virsam.no/

[7] What is room-scale VR (and what can you do with it)? - SAPPHIRE Nation - Community blog by SAPPHIRE Technology. (2016). SAPPHIRE Nation - Community blog by SAPPHIRE Technology. Retrieved 26 June 2017, from https://sapphirenation.net/room-scale-vr/

[8] (2017). *Roadtovr.com*. Retrieved 26 June 2017, from http://www.roadtovr.com/htc-explains-vive-price-cost-lowered-cut-discount/

[9] *How Virtual Reality Immersion Will Shape Our Future*. (2017). *Appreal-vr.com*. Retrieved 26 June 2017, from https://appreal-vr.com/blog/virtual-reality-immersion-what-is-it-and-how-it-works/

[10] *Home*. (2017). *Osso VR*. Retrieved 16 June 2017, from http://ossovr.com/

[11] Radiography simulation training | Medical Simulation Center New Zealand. (2017). Virtual Medical Coaching. Retrieved 16 June 2017, from http://www.virtualmedicalcoaching.com/

[12] *Using VR to Create the Schools of the Future*. (2017). *VIVE Blog*. Retrieved 16 June 2017, from https://blog.vive.com/us/2017/03/13/schools-of-the-future-using-vive-to-provide-complete-medical-imaging-education-in-vr/

[13] Official Site | Second Life - Virtual Worlds, Virtual Reality, VR, Avatars, Free 3D Chat. (2017). Secondlife.com. Retrieved 26 June 2017, from http://secondlife.com/

[14] Clare Linden A resident in Second Life. The future of healthcare: 3 questions for dave taylor of imperial college london. Blog, 2009.

[15] Nicholas A. Spooner and Momhamed Khadra. Piloting innovative pedagogy for medical education.

[16] Boulos MN, Hetherington L, Wheeler S. Second life: an overview of the potential of 3-D virtual worlds in medical and health education. Health Info Libr J. 2007;24(4):233–45.

[17] Beard L, Wilson K, Morra D, Keelan J. A survey of health-related activities on second life. J Med Internet Res. 2009;11(2):e17

[18] *Interface*. (2017). *TheFreeDictionary.com*. Retrieved 21 June 2017, from http://www.thefreedictionary.com/interface

[19] Tikhomirov V., Dneprovskaya N. (2015) Development of strategy for smart University. Paper presented at the Open Education Global international conference, Banff, Canada,

[20] Uskov V.L., Bakken J.P., Pandey A., Singh U., Yalamanchili M., Penumatsa A.: Smart University Taxonomy: Features, Components, Systems. In: Uskov LV, Howlett JR, Jain CL (eds.) Smart Education and e-Learning 2016. pp. 3–14. Springer International Publishing, Cham (2016)

[21] O'Daniel, M., & Rosenstein, A. Patient Safety and Quality: An Evidence-Based Handbook for Nurses. Rockville (MD): Agency for Healthcare Research and Quality (US); 2008 Apr.

[22] (2017). *Digitaltrends.com*. Retrieved 21 June 2017, from https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/

[23] *Here Are the 9 Best Multiplayer Vive Games You Can Play Right Now*. (2016). *UploadVR*. Retrieved 21 June 2017, from https://uploadvr.com/9-best-multiplayer-vive-games-play-right-now/

[24] Powell-Morse, A. (2016). *What Is Rapid Application Development (RAD) and How Do You Use It?*. *Airbrake Blog*. Retrieved 26 June 2017, from https://airbrake.io/blog/sdlc/rapid-application-development

[25] *VirSam*. (2017). *Virsam.no - Instructions*. Retrieved 15 June 2017, from http://virsam.no/instruksjoner.html

[26] Prasolova-Førland, E., Fominykh, M. & Wyeld, T. (2010). Virtual Campus of NTNU as a place for 3D Educational Visualizations. In Z. Abas, I. Jung & J. Luca (Eds.), *Proceedings of Global Learn 2010* (pp. 3593-3600). Association for the Advancement of Computing in Education (AACE).

[27] Kleven N.F., Prasolova-Førland E., Fominykh M., Hansen A., Rasmussen G., Sagberg L.M., Lindseth F.: Training nurses and educating the public using a virtual operating room with Oculus Rift. In: Thwaites H, Kenderdine S, Shaw J (eds.) International Conference on Virtual Systems & Multimedia (VSMM), Hong Kong, December 9–12, pp. 206–213. IEEE, New York, NY (2014)

[28] Kleven N.F., Prasolova-Førland E., Fominykh M., Hansen A., Rasmussen G., Sagberg L.M., Lindseth F.: Virtual Operating Room for Collaborative Training of Surgical Nurses. In: Baloian N, Burstein F, Ogata H, Santoro F, Zurita G (eds.) 20th International Conference on Collaboration and Technology (CRIWG), Santiago, Chile, September 7–10, Lecture Notes in Computer Science, pp. 223–238. Springer, Berlin, Heidelberg (2014)

[29] Mosand, A. Virtual humans in the Virtual Hospital. NTNU. Trondheim, Norway.

[30] Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. University of Wisconsin-Madison, Wisconsin.

[31] (2017). *Cdn.uploadvr.com*. A Case Study - The Impact of VR on Academic Performance. Retrieved 15 June 2017, from https://cdn.uploadvr.com/wp-content/uploads/2016/11/A-Case-Study-The-Impact-of-VR-on-Academic-Performance_20161125.pdf

[32] *Flow model theory by Mihaly Csikszentmihalyi | ToolsHero*. (2013). *ToolsHero*. Retrieved 15 June 2017, from https://www.toolshero.com/effectiveness/flow-model-csikszentmihalyi/

[33] Sweetser, P., & Wyeth, P. (2005). GameFlow: A Model for Evaluating Player Enjoyment in Games. The University of Queensland, St Lucio, Australia.

[34] Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Experience. Harper Perennial, New York.

[35] Kleven, N. (2014). Virtual University hospital as an arena for medical training and health education (p. 48). NTNU, Trondheim, Norway.

[36] *Definition of PROCEDURAL*. (2017). *Merriam-webster.com*. Retrieved 15 June 2017, from https://www.merriam-webster.com/dictionary/procedural

[37] *da Vinci Surgery - Minimally Invasive Robotic Surgery with the da Vinci Surgical System*. (2017). *Davincisurgery.com*. Retrieved 16 June 2017, from http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/

[38] *Virtual Reality - X-ray of the Mortice Ankle joint*. (2017). *YouTube*. Retrieved 16 June 2017, from https://www.youtube.com/watch?v=EaCpvggu8-o

[39] Healey AN, Undre S, Vincent CA. Developing observational measures of performance in surgical teams. Qual Saf Health Care. 2004;13(Suppl 1):i33–i4

[40] Baker D.P., Gustafson S., Beaubien J.M., Salas E., Barach P.: Medical Team Training Programs in Health Care. In: Henriksen K, Battles JB, Marks ES, Lewin DI (eds.) Advances in Patient Safety: From Research to Implementation (Volume 4: Programs, Tools, and Products). Advances in Patient Safety. Rockville (MD) (2005)

[41] Mark J. W. Lee (2009). How Can 3d Virtual Worlds Be Used To Support Collaborative Learning? An Analysis Of Cases From The Literature. Journal of e-Learning Je-LKS and Knowledge Society — Vol. 5, n. 1, February 2009 (p. 156).

[42] Prasolova-Førland, E. (2004). Supporting Social Awareness among University Students with Collaborative Virtual Environments (pp. 24-31). NTNU, Trondheim, Norway.

[43] Gifford, B.R.& Enyedy, N.D. (1999). Activity Centered Design: Towards a Theoretical Framework for CSCL. In Proceedings of CSCL1999, December 1215, Palo Alto, CA, USA, (pp. 189-196). Lawrence Erlbaum Associates

[44] Engeström, Y. (1987). Learning by Expanding: An Activity-Theoretical Approach to Developmental Research. Orienta-Konsultit, Helsinki.

[45] Vygotskij, L.S. (1978). Mind and Society. Harvard University Press, Cambridge, MA.

[46] *Activity Theory: The Encyclopedia of Human-Computer Interaction, 2nd Ed.*. (2017). *The Interaction Design Foundation*. Retrieved 11 June 2017, from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/activity-theory

[47] *Everything Sony Told Us About the Future of PlayStation*. (2017). *Time.com*. Retrieved 16 June 2017, from http://time.com/4804768/playstation-4-ps4-pro-psvr-sales/

[48] *MacOS finally supports VR*. (2017). *Engadget*. Retrieved 16 June 2017, from https://www.engadget.com/2017/06/05/apple-macos-vr-external-gpu-imac/

[49] *Valve CEO: 'We're comfortable with the idea that VR will turn out to be a complete failure'*. (2017). *StamfordAdvocate*. Retrieved 16 June 2017, from http://www.stamfordadvocate.com/technology/businessinsider/article/Valve-CEO-We-re-comfortable-with-the-idea-that-10935960.php

[50] Takahashi, D. (2016). *Only 6% of Americans will own a VR headset in 2016*. *VentureBeat*. Retrieved 16 June 2017, from https://venturebeat.com/2016/10/13/only-6-of-americans-will-own-a-vr-headset-in-2016/

[51] *Unity Features*. (2017). *Developer.oculus.com*. Retrieved 26 June 2017, from https://developer.oculus.com/documentation/avatarsdk/latest/concepts/avatars-sdk-unity/#avatars-sdk-unity-avatars-sdk-unity-custom-grip-poses

[52] *Stack Overflow*. (2017). *Stackoverflow.com*. Retrieved 16 June 2017, from https://stackoverflow.com/

[53] *Alf Inge Wang - NTNU*. (2017). *Ntnu.edu*. Retrieved 16 June 2017, from https://www.ntnu.edu/employees/alf.inge.wang

[54] *VirSam*. (2017). *Virsam.no – Geriatrics Case*. Retrieved 15 June 2017, from http://virsam.no/case_geriatri.html

[55] *Simulator Sickness*. (2017). *Developer.oculus.com*. Retrieved 26 June 2017, from https://developer.oculus.com/design/latest/concepts/bp_app_simulator_sickness/

[56] Callele, D., Neufeldt E., & Scheider, K. Requirements Engineering and the Creative Process in the Video Game Industry. IEEE (2005). University of Saskatchewan Saskatoon, Saskatchewan, Canada S7N 5C9.

[57] *Innovation Expo 2017 – Technoport*. (2017). *Conference.technoport.no*. Retrieved 19 June 2017, from http://conference.technoport.no/index.php/innovationexpo/

[58] *Hjem – Medisinsk SimulatorSenter*. (2017). *Medisinsk SimulatorSenter*. Retrieved 19 June 2017, from http://www.simulatorsenteret.no/

[59] *Aslak Steinsbekk - NTNU*. (2017). *Ntnu.no*. Retrieved 16 June 2017, from https://www.ntnu.no/ansatte/aslak.steinsbekk

[60] *Ekaterina Prasolova-Førland - NTNU*. (2017). *Ntnu.no*. Retrieved 16 June 2017, from https://www.ntnu.no/ansatte/ekaterip

[61] Bass, L., Clements, P., and Kazman, R. Software Architecture in Practice (Third Edition). NJ: Addison-Wesley Professional, 2013

[62] *Single Responsibility Principle | Object Oriented Design*. (2017). *Oodesign.com*. Retrieved 26 June 2017, from http://www.oodesign.com/single-responsibility-principle.html

[63] *Build software better, together*. (2017). *GitHub*. Retrieved 17 June 2017, from https://github.com/

[64] *Git*. (2017). *Git-scm.com*. Retrieved 17 June 2017, from https://git-scm.com/

[65] Technologies, U. (2017). *Unity - Manual: Coroutines*. *Docs.unity3d.com*. Retrieved 17 June 2017, from https://docs.unity3d.com/Manual/Coroutines.html

[66] *Asset Store*. (2017). *Assetstore.unity3d.com – Hospital Room* . Retrieved 7 June 2017, from https://www.assetstore.unity3d.com/en/#!/content/57399

[67] *Asset Store*. (2017). *Assetstore.unity3d.com – Operating Room*. Retrieved 17 June 2017, from https://www.assetstore.unity3d.com/en/#!/content/18295

[68] Technologies, U. (2017). *Unity - Manual: Character Controller*. *Docs.unity3d.com*. Retrieved 7 June 2017, from https://docs.unity3d.com/Manual/class-CharacterController.html

[69] Technologies, U. (2017). *Unity - Scripting API: Animator*. *Docs.unity3d.com*. Retrieved 7 June 2017, from https://docs.unity3d.com/ScriptReference/Animator.html

[70] *Asset Store*. (2017). *Assetstore.unity3d.com*. Retrieved 7 June 2017, from https://www.assetstore.unity3d.com/en/#!/content/15667

[71] *Asset Store*. (2017). *Assetstore.unity3d.com*. Retrieved 7 June 2017, from https://www.assetstore.unity3d.com/en/#!/content/64131

[72] *Asset Store*. (2017). *Assetstore.unity3d.com*. Retrieved 17 June 2017, from https://www.assetstore.unity3d.com/en/#!/content/32647

[73] Photon Unity 3D Networking Framework SDKs and Game Backend | Photon: Multiplayer Made Simple . (2017). Photonengine.com. Retrieved 7 June 2017, from https://www.photonengine.com/en/PUN

[74] *Photon PUN Pricing Plans | Photon: Multiplayer Made Simple* . (2017). *Photonengine.com*. Retrieved 7 June 2017, from https://www.photonengine.com/en/PUN/Pricing#plan-20

[75] Greenberg, S. Multimed Tools Appl (2007) 32: 139. doi:10.1007/s11042-006-0062-y

[76] *Slant - Unity - What are the best 3D game engines? Slant*. Retrieved 18 June 2017, from https://www.slant.co/topics/1495/viewpoints/1/~3d-game-engines~unity#4

[77] Webster, L. (2017). *Visual Studio IDE, Code Editor, Team Services, & Mobile Center*. *Visual Studio*. Retrieved 18 June 2017, from https://www.visualstudio.com/

[78] Technologies, U. (2017). *Unity - Manual: Unity User Manual (5.6)*. *Docs.unity3d.com*. Retrieved 18 June 2017, from https://docs.unity3d.com/Manual/index.html

[79] *Building a Multiplayer Metaverse with Rift + Vive*. (2017). *YouTube*. Retrieved 18 June 2017, from https://www.youtube.com/watch?v=6pvSC-RsWc8

[80] What VR Could, Should, and Almost Certainly Will Be within Two Years (Steam Dev Days 2014). (2017). YouTube. Retrieved 26 June 2017, from https://www.youtube.com/watch?v=G-2dQoeqVVo

[81] (2017). Media.steampowered.com. Retrieved 26 June 2017, from http://media.steampowered.com/apps/abrashblog/Abrash%20Dev%20Days%202014.pdf

[82] (2017). *S-media-cache-ak0.pinimg.com*. Retrieved 19 June 2017, from https://s-media-cache-ak0.pinimg.com/originals/97/38/bd/9738bd6d52d7725f9d0752f2f1516809.jpg

[83] (2017). *Assets.inhabitat.com*. Retrieved 19 June 2017, from http://assets.inhabitat.com/wp-content/blogs.dir/1/files/2011/09/Sant-Joan-Hospital-de-Reus-Mario-Corea-Arquitectura-10.jpg

[84] *How Lenses for Virtual Reality Headsets Work*. (2017). *YouTube*. Retrieved 14 June 2017, from https://www.youtube.com/watch?v=NCBEYaC876A

[85] Technologies, U. (2017). *Unity - Manual: Cloth*. *Docs.unity3d.com*. Retrieved 22 June 2017, from https://docs.unity3d.com/Manual/class-Cloth.html

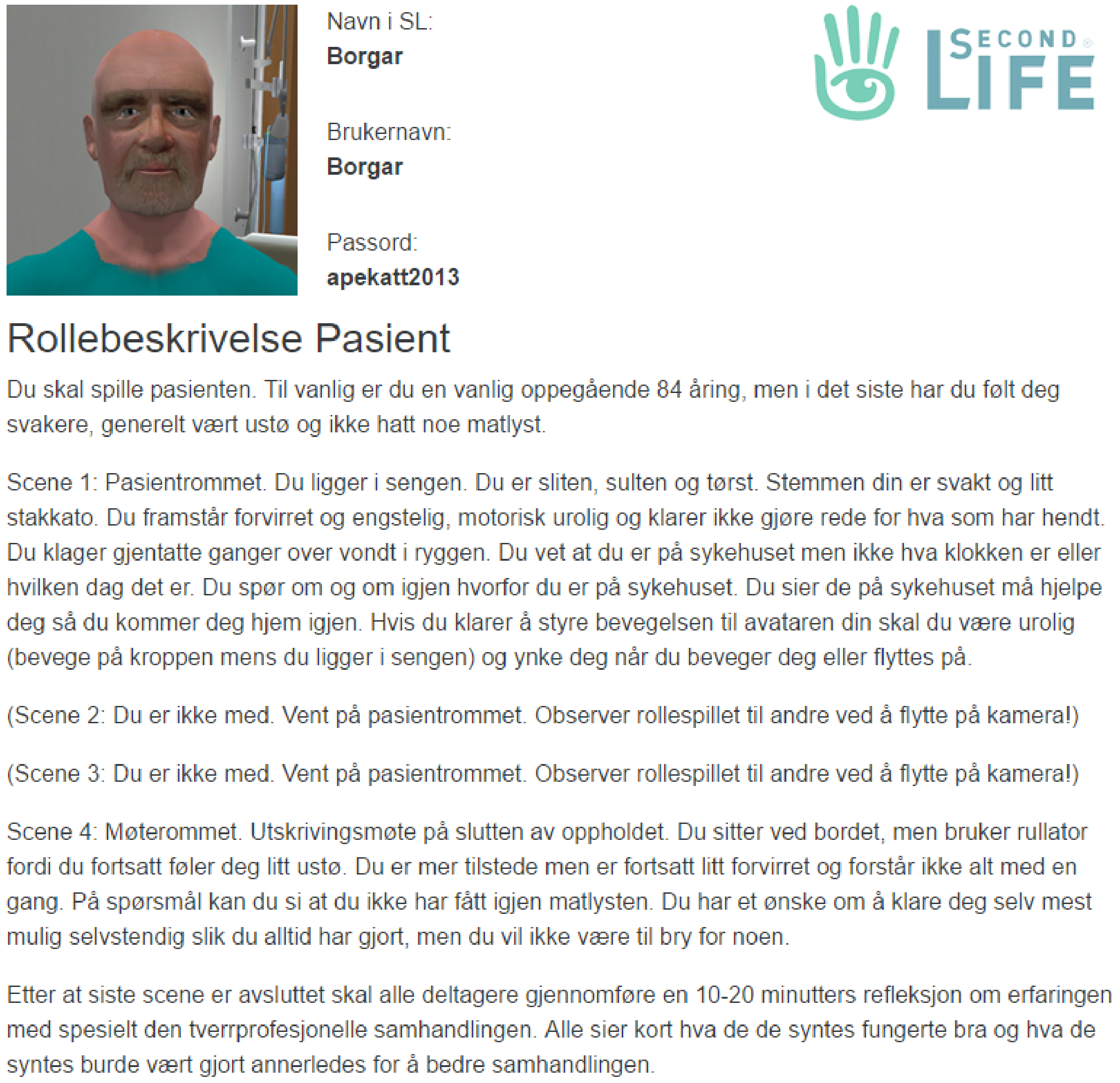
[86] American Diagnostic Corporation - Core Medical Device Manufacturer. Stethoscopes, Blood Pressure, Thermometry, and EENT. (2017). Adctoday.com. Retrieved 22 June 2017, from http://adctoday.com/blog/intro-blood-pressure

[87] (2017). *Roadtovr.com*. Retrieved 26 June 2017, from http://www.roadtovr.com/valve-reveals-detail-knuckles-motion-controllers/

[88] Reed, A., & Langley, H. (2017). *Virtual reality peripherals to check out*. *Wareable*. Retrieved 26 June 2017, from https://www.wareable.com/vr/vr-peripherals-controllers

[89] (2017). *A4.images.reviewed.com*. Retrieved 26 June 2017, from http://a4.images.reviewed.com/image/fetch/c_limit,w_856,h_570,q_89,f_auto/https://reviewed-production.s3.amazonaws.com/attachment/95645bba928346f9/HTC_Vive_Controller.jpg

Appendix A VirSam Geriatrics Role Descriptions



Navn i SL:
Borgar

Brukernavn:
Borgar

Passord:
apekatt2013

Rollebeskrivelse Pasient

Du skal spille pasienten. Til vanlig er du en vanlig oppegående 84 åring, men i det siste har du følt deg svakere, generelt vært ustø og ikke hatt noe matlyst.

Scene 1: Pasientrommet. Du ligger i sengen. Du er sliten, sulten og tørst. Stemmen din er svakt og litt stakkato. Du framstår forvirret og engstelig, motorisk urolig og klarer ikke gjøre rede for hva som har hendt. Du klager gjentatte ganger over vondt i ryggen. Du vet at du er på sykehuset men ikke hva klokken er eller hvilken dag det er. Du spør om og om igjen hvorfor du er på sykehuset. Du sier de på sykehuset må hjelpe deg så du kommer deg hjem igjen. Hvis du klarer å styre bevegelsen til avataren din skal du være urolig (bevege på kroppen mens du ligger i sengen) og ynke deg når du beveger deg eller flyttes på.

(Scene 2: Du er ikke med. Vent på pasientrommet. Observer rollespillet til andre ved å flytte på kamera!)

(Scene 3: Du er ikke med. Vent på pasientrommet. Observer rollespillet til andre ved å flytte på kamera!)

Scene 4: Møterommet. Utskrivingsmøte på slutten av oppholdet. Du sitter ved bordet, men bruker rullator fordi du fortsatt føler deg litt ustø. Du er mer tilstede men er fortsatt litt forvirret og forstår ikke alt med en gang. På spørsmål kan du si at du ikke har fått igjen matlysten. Du har et ønske om å klare deg selv mest mulig selvstendig slik du alltid har gjort, men du vil ikke være til bry for noen.

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 10-20 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen.

Navn i SL:
Kajandra

Brukernavn:
Kajandra

Passord:
apekatt2013

Navn i SL:
Thormod

Brukernavn:
Thormod

Passord:
apekatt2013

Rollebeskrivelse Lege

Du skal være mest mulig deg selv i rollen som lege som jobber på geriatrisk avdeling.

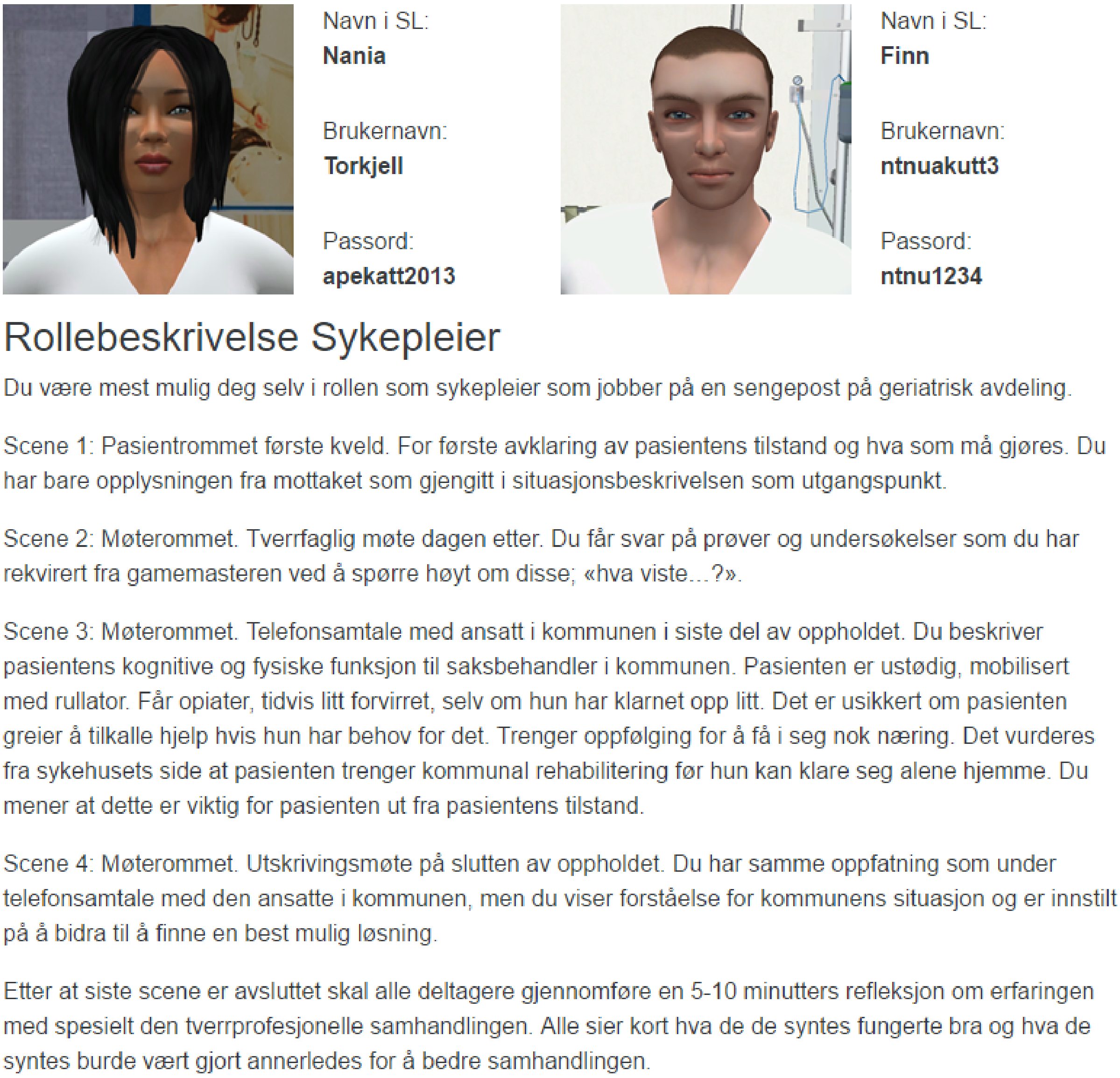
Scene 1: Pasientrommet første kveld. For første avklaring av pasientens tilstand og hva som må gjøres. Du har bare opplysningen fra mottaket som gjengitt i situasjonsbeskrivelsen som utgangspunkt.

Scene 2: Møterommet. Tverrfaglig møte dagen etter. Du får svar på prøver og undersøkelser som du har rekvirert fra gamemasteren ved å spørre høyt om disse; «hva viste…?».

(Scene 3: Du er ikke med. Du venter på venterommet. Observer rollespillet ved å styre kamera!)

Scene 4: Møterommet. Utskrivingsmøte på slutten av oppholdet. Pasienten får opiater og er tidvis litt forvirret selv om han har klarnet opp. Du er usikker på om pasienten greier å tilkalle hjelp hvis han har behov for det og du vurderer at pasienten trenger oppfølging for å få i seg nok næring.

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 10-20 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen.



Navn i SL:
Nania

Brukernavn:
Torkjell

Passord:
apekatt2013

Navn i SL:
Finn

Brukernavn:
ntnuakutt3

Passord:
ntnu1234

Rollebeskrivelse Sykepleier

Du være mest mulig deg selv i rollen som sykepleier som jobber på en sengepost på geriatrisk avdeling.

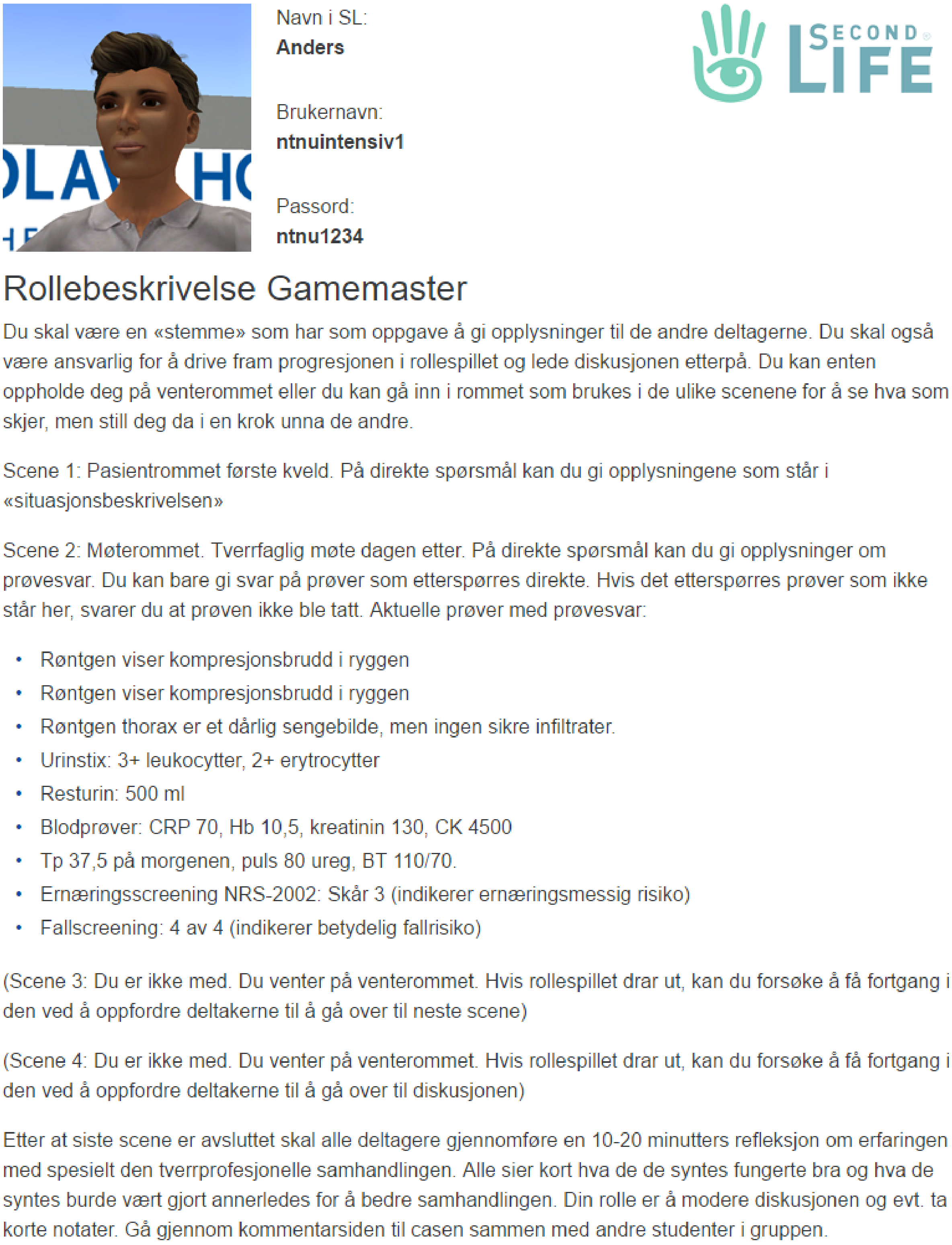
Scene 1: Pasientrommet første kveld. For første avklaring av pasientens tilstand og hva som må gjøres. Du har bare opplysningen fra mottaket som gjengitt i situasjonsbeskrivelsen som utgangspunkt.

Scene 2: Møterommet. Tverrfaglig møte dagen etter. Du får svar på prøver og undersøkelser som du har rekvirert fra gamemasteren ved å spørre høyt om disse; «hva viste…?».

Scene 3: Møterommet. Telefonsamtale med ansatt i kommunen i siste del av oppholdet. Du beskriver pasientens kognitive og fysiske funksjon til saksbehandler i kommunen. Pasienten er ustødig, mobilisert med rullator. Får opiater, tidvis litt forvirret, selv om hun har klarnet opp litt. Det er usikkert om pasienten greier å tilkalle hjelp hvis hun har behov for det. Trenger oppfølging for å få i seg nok næring. Det vurderes fra sykehusets side at pasienten trenger kommunal rehabilitering før hun kan klare seg alene hjemme. Du mener at dette er viktig for pasienten ut fra pasientens tilstand.

Scene 4: Møterommet. Utskrivingsmøte på slutten av oppholdet. Du har samme oppfatning som under telefonsamtale med den ansatte i kommunen, men du viser forståelse for kommunens situasjon og er innstilt på å bidra til å finne en best mulig løsning.

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 5-10 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen.



Navn i SL:
Anders

Brukernavn:
ntnuintensiv1

Passord:
ntnu1234

Rollebeskrivelse Gamemaster

Du skal være en «stemme» som har som oppgave å gi opplysninger til de andre deltagerne. Du skal også være ansvarlig for å drive fram progresjonen i rollespillet og lede diskusjonen etterpå. Du kan enten oppholde deg på venterommet eller du kan gå inn i rommet som brukes i de ulike scenene for å se hva som skjer, men still deg da i en krok unna de andre.

Scene 1: Pasientrommet første kveld. På direkte spørsmål kan du gi opplysningene som står i «situasjonsbeskrivelsen»

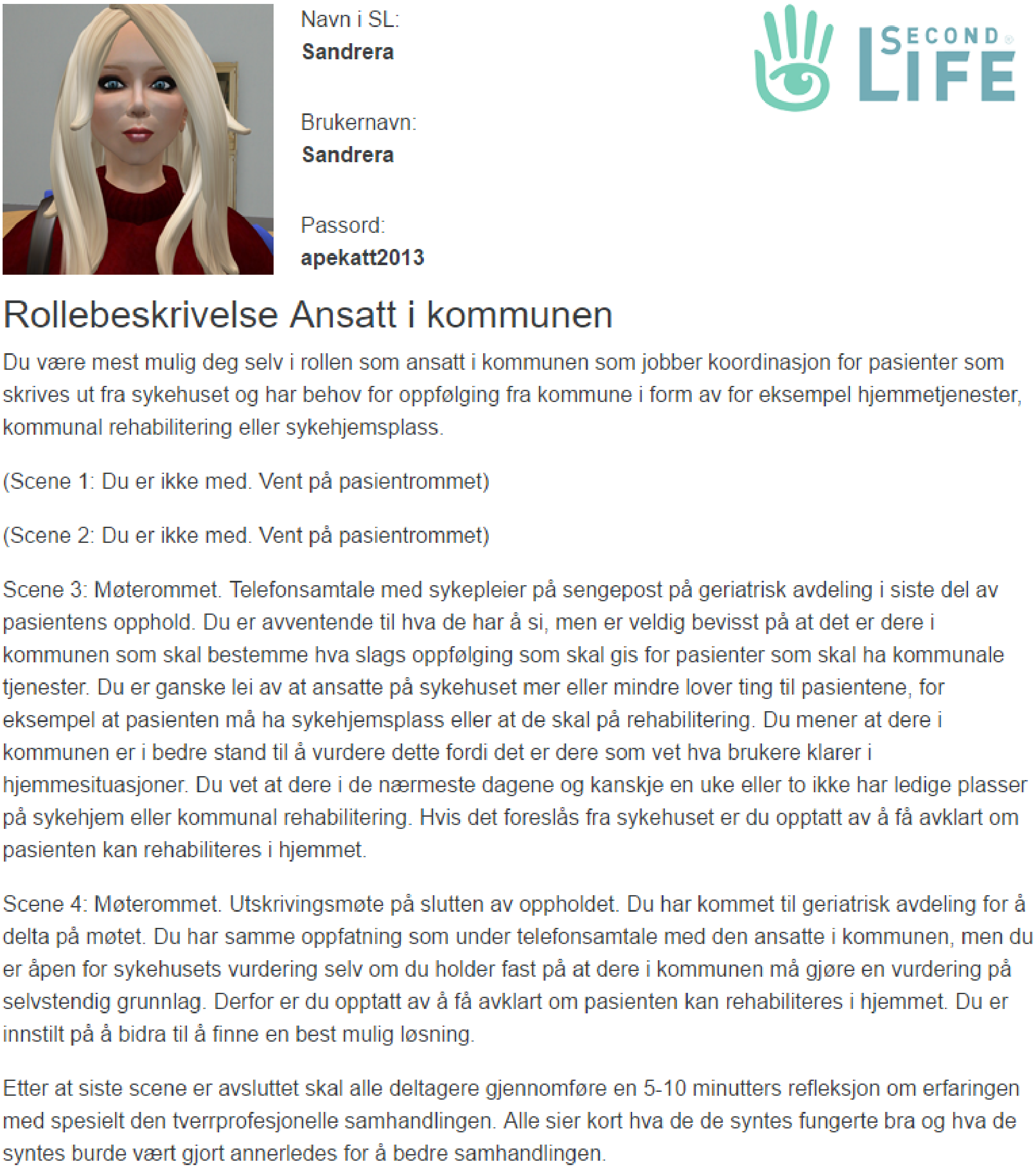
Scene 2: Møterommet. Tverrfaglig møte dagen etter. På direkte spørsmål kan du gi opplysninger om prøvesvar. Du kan bare gi svar på prøver som etterspørres direkte. Hvis det etterspørres prøver som ikke står her, svarer du at prøven ikke ble tatt. Aktuelle prøver med prøvesvar:

- Røntgen viser kompresjonsbrudd i ryggen
- Røntgen viser kompresjonsbrudd i ryggen
- Røntgen thorax er et dårlig sengebilde, men ingen sikre infiltrater.
- Urinstix: 3+ leukocytter, 2+ erytrocytter
- Resturin: 500 ml
- Blodprøver: CRP 70, Hb 10,5, kreatinin 130, CK 4500
- Tp 37,5 på morgenen, puls 80 ureg, BT 110/70.
- Ernæringsscreening NRS-2002: Skår 3 (indikerer ernæringsmessig risiko)
- Fallscreening: 4 av 4 (indikerer betydelig fallrisiko)

(Scene 3: Du er ikke med. Du venter på venterommet. Hvis rollespillet drar ut, kan du forsøke å få fortgang i den ved å oppfordre deltakerne til å gå over til neste scene)

(Scene 4: Du er ikke med. Du venter på venterommet. Hvis rollespillet drar ut, kan du forsøke å få fortgang i den ved å oppfordre deltakerne til å gå over til diskusjonen)

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 10-20 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen. Din rolle er å modere diskusjonen og evt. ta korte notater. Gå gjennom kommentarsiden til casen sammen med andre studenter i gruppen.



Navn i SL:
Sandrera

Brukernavn:
Sandrera

Passord:
apekatt2013

Rollebeskrivelse Ansatt i kommunen

Du være mest mulig deg selv i rollen som ansatt i kommunen som jobber koordinasjon for pasienter som skrives ut fra sykehuset og har behov for oppfølging fra kommune i form av for eksempel hjemmetjenester, kommunal rehabilitering eller sykehjemsplass.

(Scene 1: Du er ikke med. Vent på pasientrommet)

(Scene 2: Du er ikke med. Vent på pasientrommet)

Scene 3: Møterommet. Telefonsamtale med sykepleier på sengepost på geriatrisk avdeling i siste del av pasientens opphold. Du er avventende til hva de har å si, men er veldig bevisst på at det er dere i kommunen som skal bestemme hva slags oppfølging som skal gis for pasienter som skal ha kommunale tjenester. Du er ganske lei av at ansatte på sykehuset mer eller mindre lover ting til pasientene, for eksempel at pasienten må ha sykehjemsplass eller at de skal på rehabilitering. Du mener at dere i kommunen er i bedre stand til å vurdere dette fordi det er dere som vet hva brukere klarer i hjemmesituasjoner. Du vet at dere i de nærmeste dagene og kanskje en uke eller to ikke har ledige plasser på sykehjem eller kommunal rehabilitering. Hvis det foreslås fra sykehuset er du opptatt av å få avklart om pasienten kan rehabiliteres i hjemmet.

Scene 4: Møterommet. Utskrivingsmøte på slutten av oppholdet. Du har kommet til geriatrisk avdeling for å delta på møtet. Du har samme oppfatning som under telefonsamtale med den ansatte i kommunen, men du er åpen for sykehusets vurdering selv om du holder fast på at dere i kommunen må gjøre en vurdering på selvstendig grunnlag. Derfor er du opptatt av å få avklart om pasienten kan rehabiliteres i hjemmet. Du er innstilt på å bidra til å finne en best mulig løsning.

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 5-10 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen.

Navn i SL:
Rannja

Brukernavn:
Rannja

Passord:
apekatt2013

Rollebeskrivelse Pårørende til pasient

Du skal spille det godt voksne barnet til pasienten (pårørende). Du er opptatt av at helsepersonellet skal få gjort jobben sin slik at du bidrar med all informasjon du har for at de skal kunne gi best mulig behandling.

Scene 1: Pasientrommet. Du har akkurat kommet til sykehuset og vet ikke hva som har skjedd ut over at du ble oppringt en stund etter at din forelder ble innlagt i går. Du forteller om din forelders situasjon og bakgrunnen når din forelder ikke selv kan redegjøre for dette. Du forteller at din forelder til vanlig er en oppegående og selvstendig 84 åring. Du opplyser på spørsmål at din forelder i de siste ukene har klaget over å føle seg litt ute av form, ikke sovet så godt pga hyppigere vannlating om natten og hatt dårligere matlyst. Du hadde derfor tenkt å ordne så din foreldre fikk time hos fastlegen, men rakk ikke det før dette skjedde. Du dikter ved behov opp ting om din forelder som ikke er beskrevet i situasjonsbeskrivelsen, for eksempel om interesser og tidligere liv.

(Scene 2: Du er ikke med. Vent på pasientrommet. Observer rollespillet til andre!)

(Scene 3: Du er ikke med. Vent på pasientrommet. Observer rollespillet til andre!)

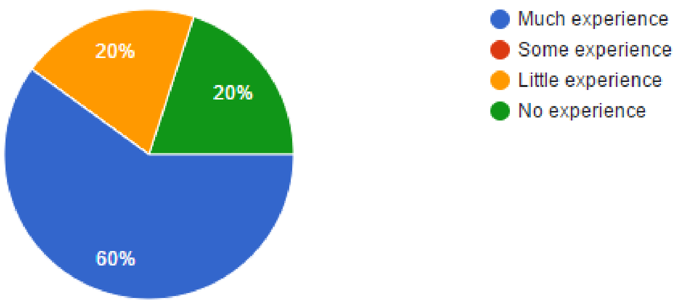
Scene 4: Møterommet. Utskrivingsmøte på slutten av oppholdet. Du er opptatt av at din forelder får god oppfølging etter utskriving, men er samtidig klar på at din forelder har ønske om å klare seg selv mest mulig selvstendig slik han/hun alltid har gjort. Du er derfor opptatt av at din forelder ikke blir for lenge på sykehuset fordi du har hørt at det å komme i vanlig aktivitet er veldig viktig. Du og din søsken bor selv så langt unna at dere ikke kan bistå i det daglige, men dere og andre i familien kan hjelpe til regelmessig i helger og ved avtale på noen hverdager.

Etter at siste scene er avsluttet skal alle deltagere gjennomføre en 10-20 minutters refleksjon om erfaringen med spesielt den tverrprofesjonelle samhandlingen. Alle sier kort hva de de syntes fungerte bra og hva de syntes burde vært gjort annerledes for å bedre samhandlingen.

Appendix B Room-Scale vs. Desktop

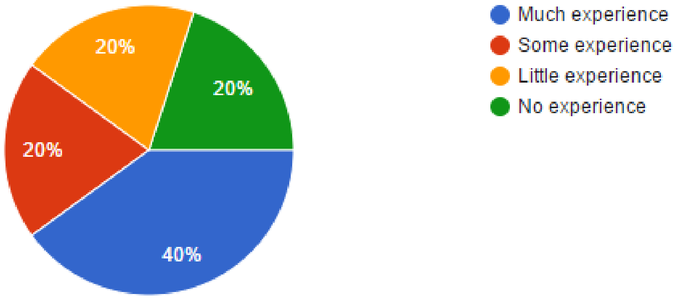
I have experience with virtual worlds (e.g. online games or simulators)

5 responses



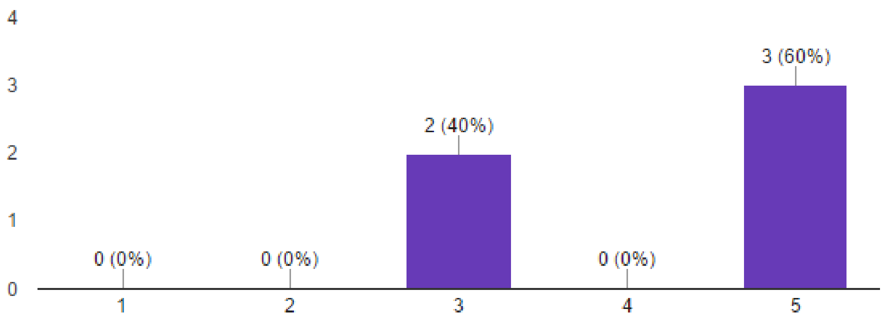
I have experience with virtual reality (e.g Oculus, Vive, Room-scale)

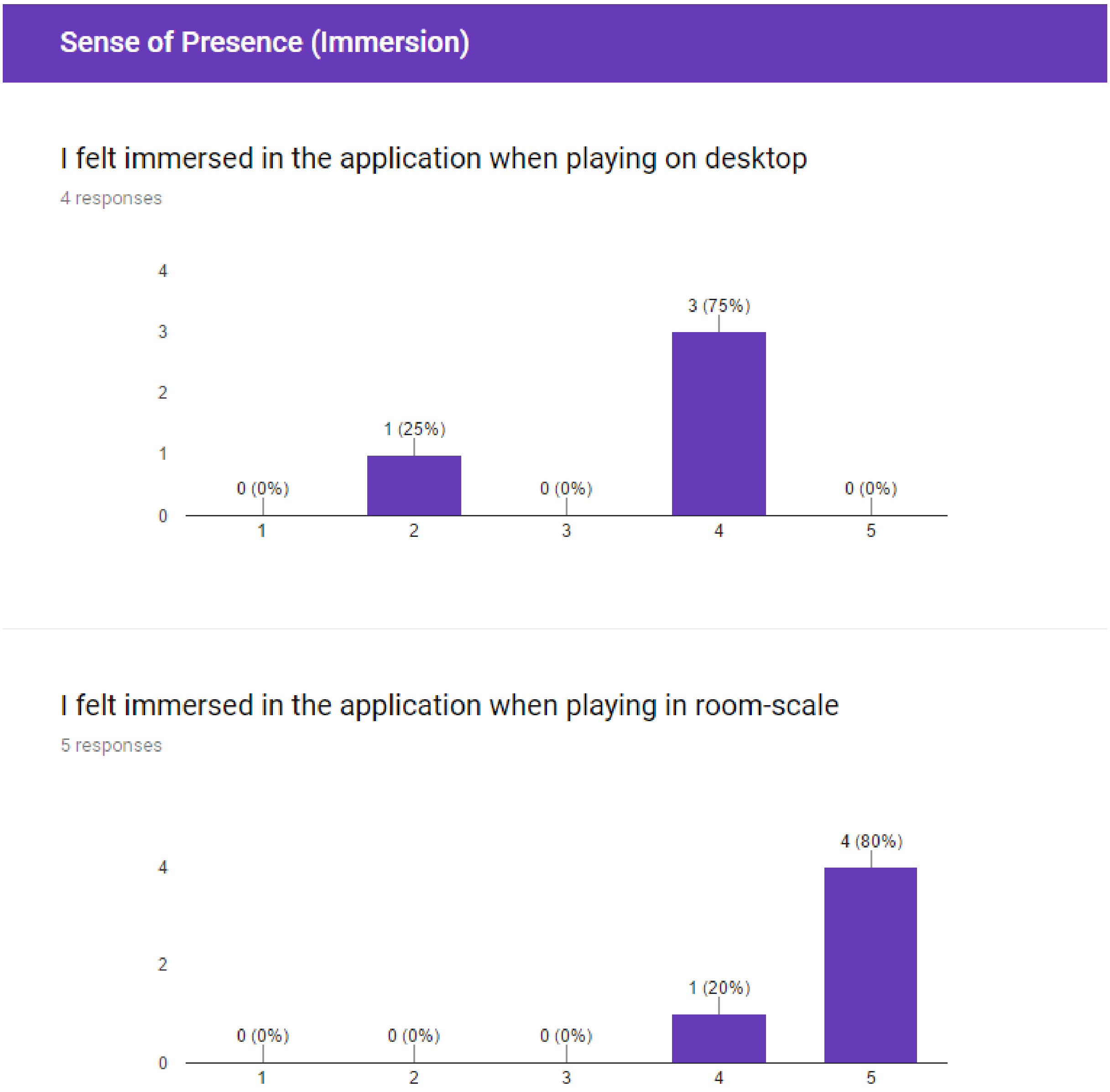
5 responses



How would you rate your expertise of using a PC?

5 responses





Sense of Presence (Immersion)

I felt immersed in the application when playing on desktop

4 responses

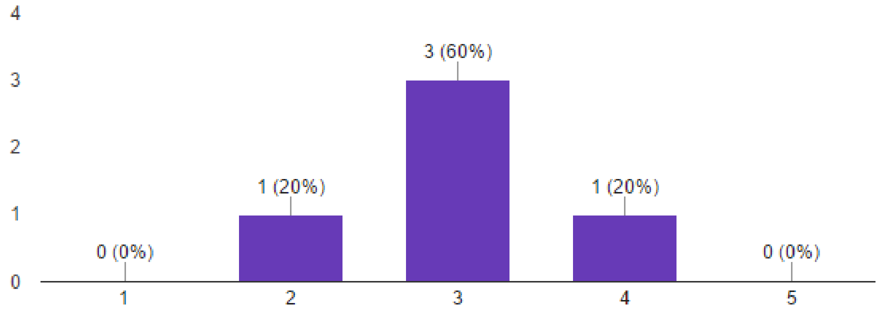
I felt immersed in the application when playing in room-scale

5 responses

Avatars

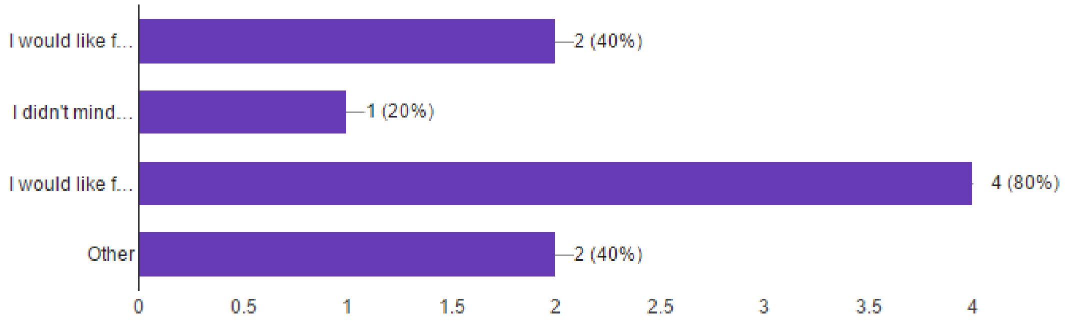
The animated full-body avatars looked realistic

5 responses



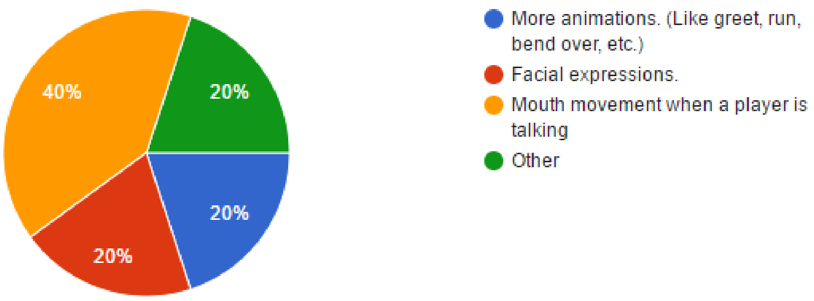
What did you think of the room-scale avatars?

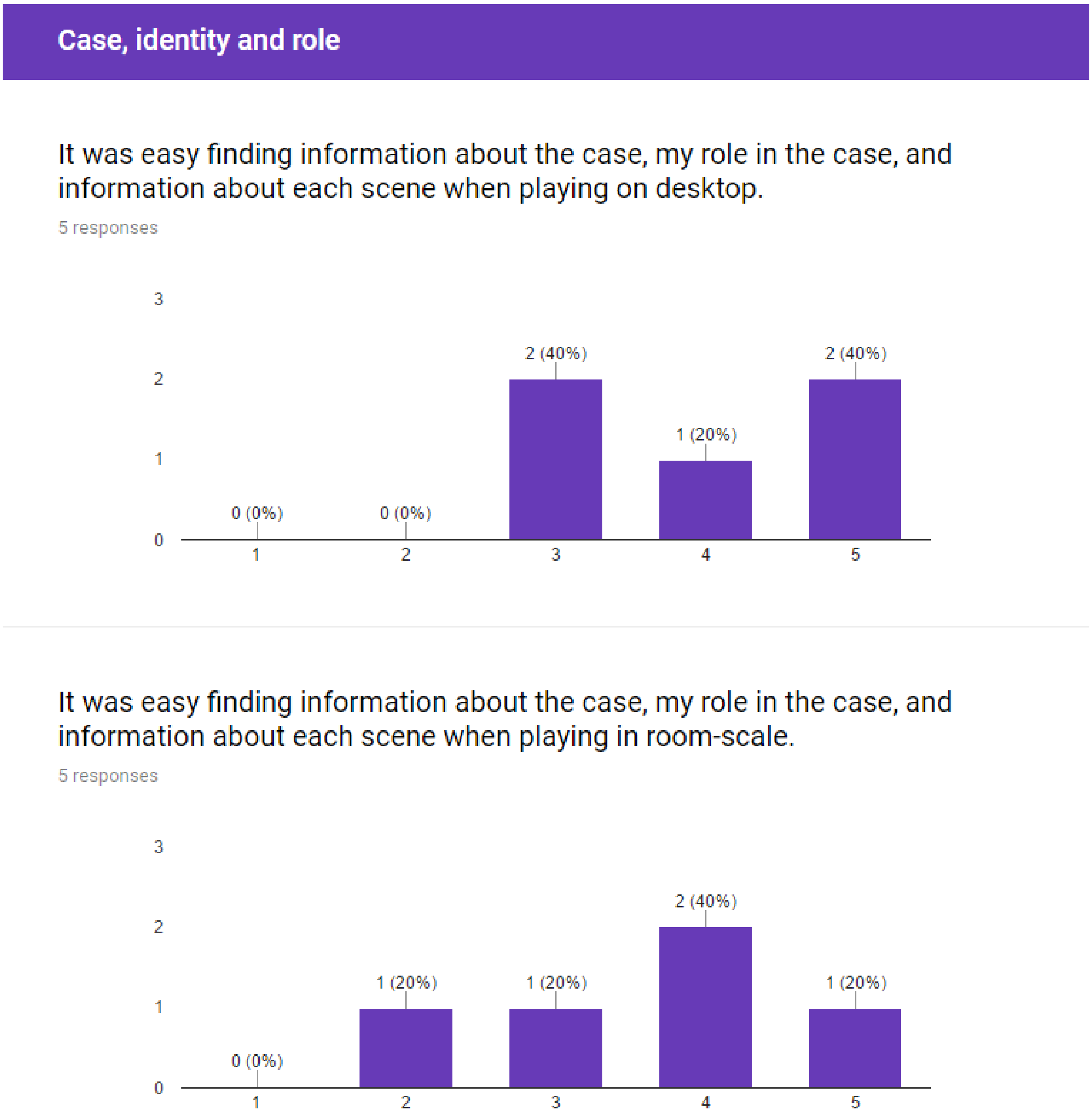
5 responses



I miss some of these animation features

5 responses





Case, identity and role

It was easy finding information about the case, my role in the case, and information about each scene when playing on desktop.

5 responses

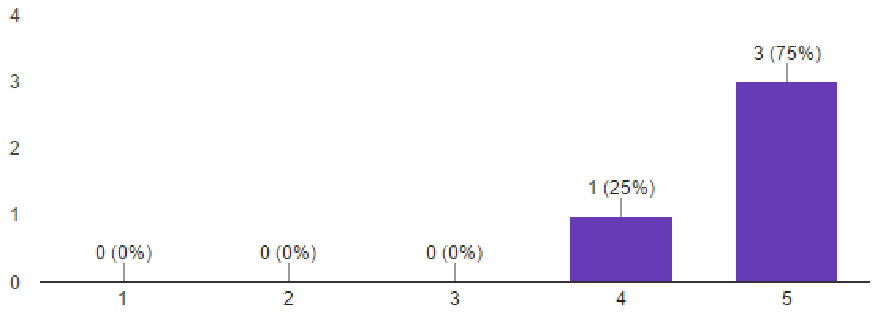
It was easy finding information about the case, my role in the case, and information about each scene when playing in room-scale.

5 responses

Navigation

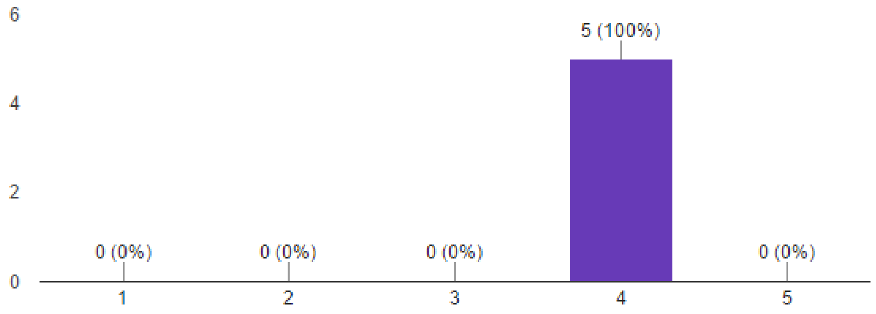
It was easy moving my avatar EXACTLY where I wanted it to go when playing on desktop

4 responses



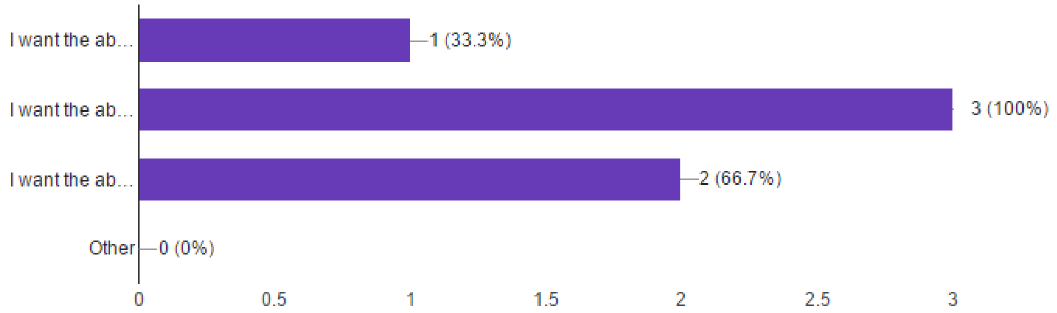
It was easy moving my avatar EXACTLY where I wanted it to go when playing in room-scale

5 responses



Additional navigation

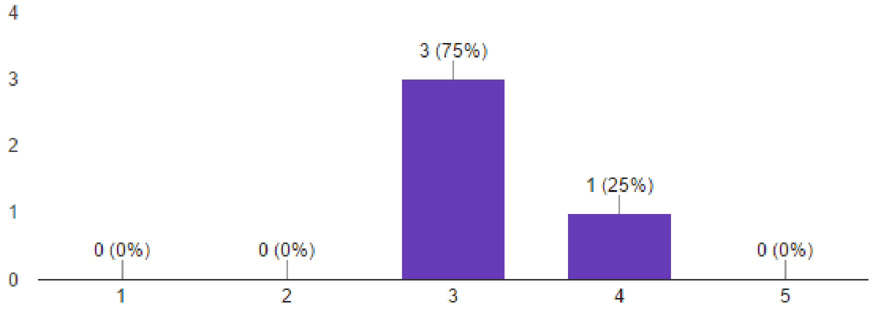
3 responses



Change view/Camera

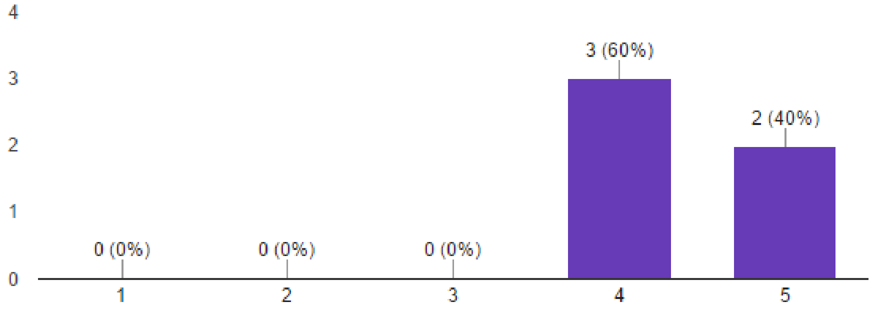
It was easy getting a view of the important things in the scene by moving the camera when playing on desktop

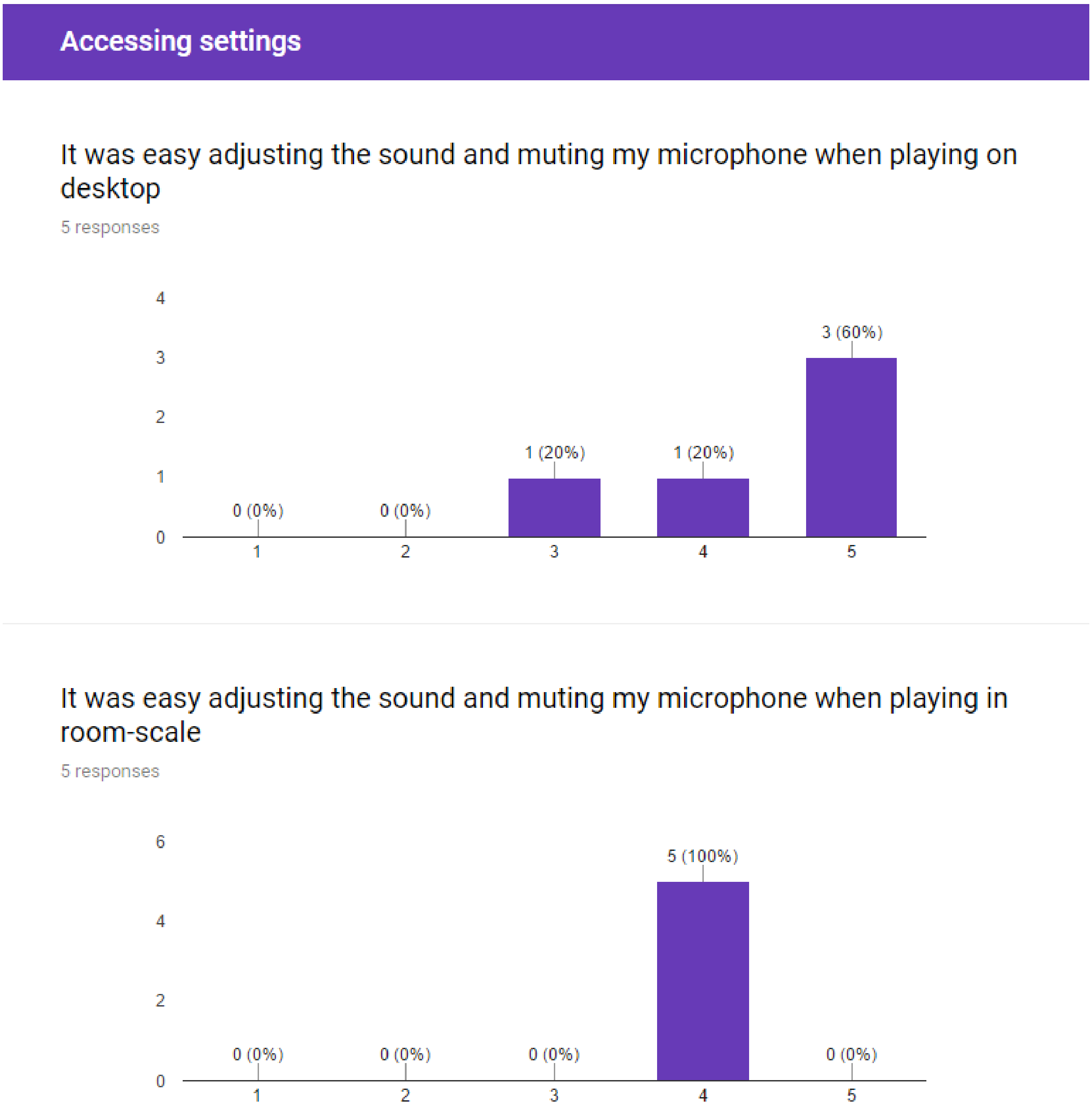
4 responses



It was easy getting a view of the important things in the scene by moving my head in room-scale

5 responses





Accessing settings

It was easy adjusting the sound and muting my microphone when playing on desktop

5 responses

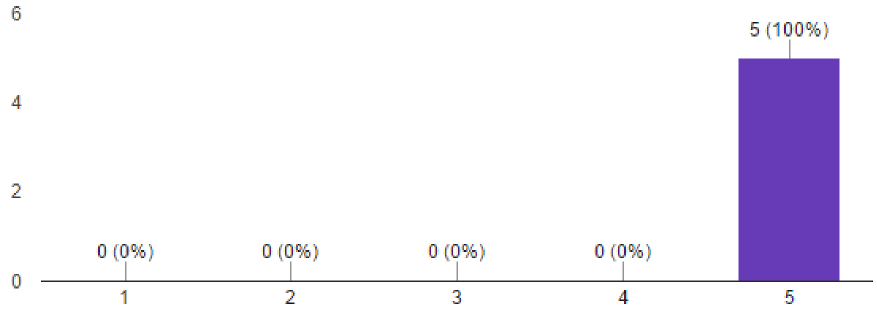
It was easy adjusting the sound and muting my microphone when playing in room-scale

5 responses

Rooms

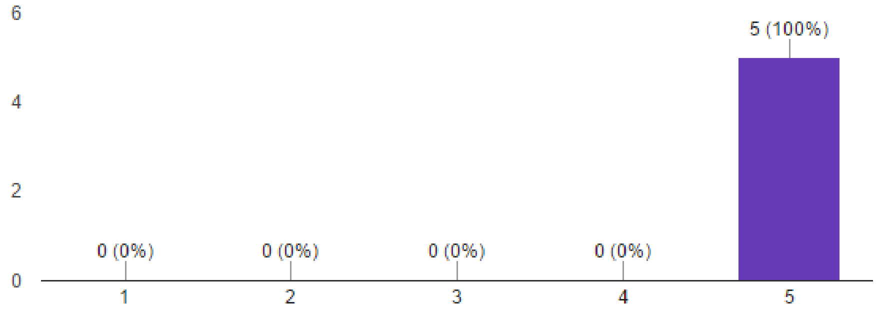
The meeting room contained the relevant objects for completing all scenes.

5 responses



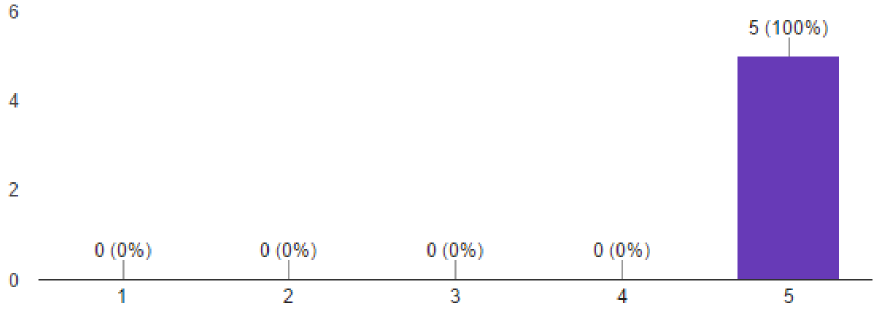
The patient room contained the relevant objects for completing all scenes.

5 responses



The hallway contained the relevant objects for completing all scenes.

5 responses



What was missing in terms of interactive objects?

2 responses

Doors. Working sink. Greeting the patient when he's laying in bed.

I missed the ability to pick up objects when playing on desktop.

What was missing in terms of props (for making the rooms/scenes even more realistic)?

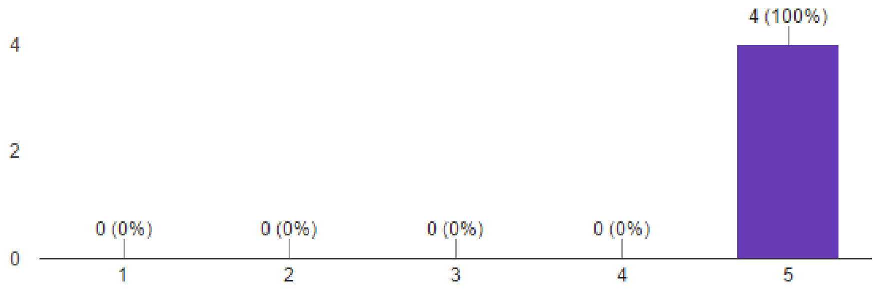
1 response

Utsikt fra vinduet

Interactive Objects

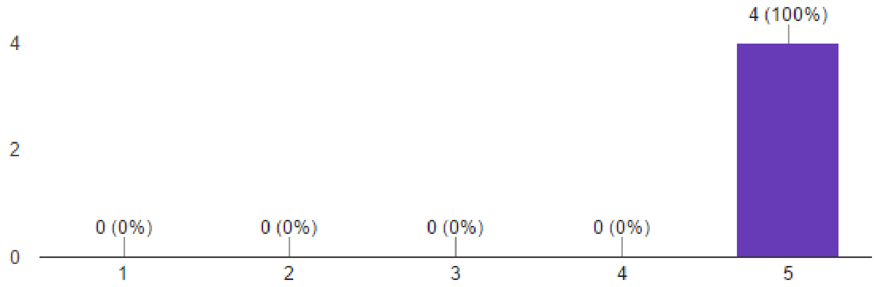
It was easy lying down in the bed while playing on desktop

4 responses



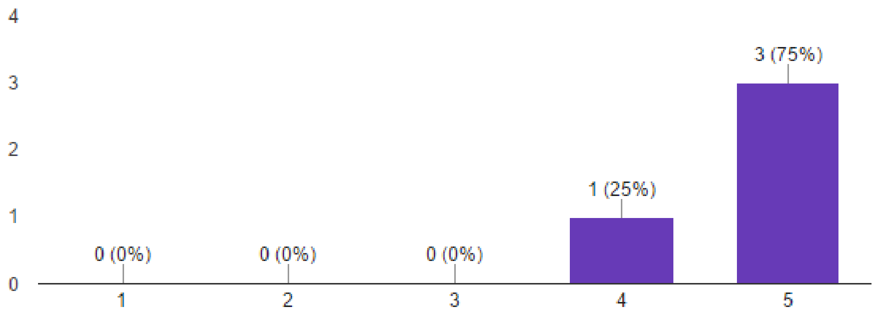
It was easy to sit down in chairs on desktop

4 responses



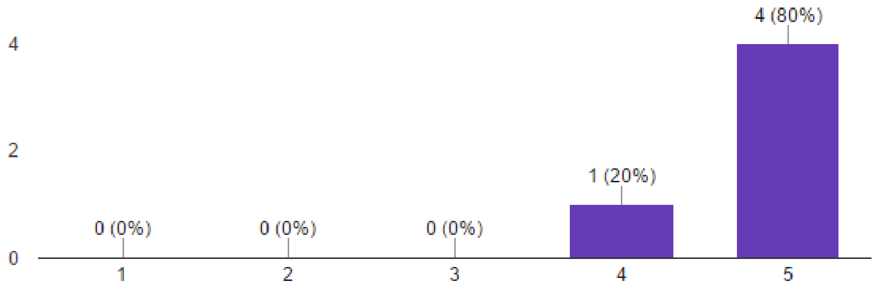
It was easy to get up from chairs on desktop

4 responses



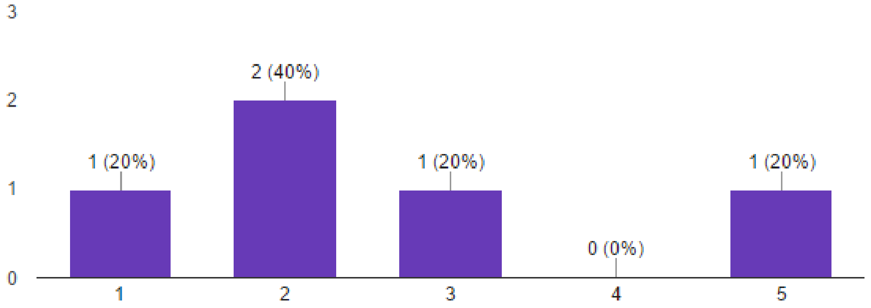
It was easy to see that an object was highlighted when the mouse was hovering over it.

5 responses



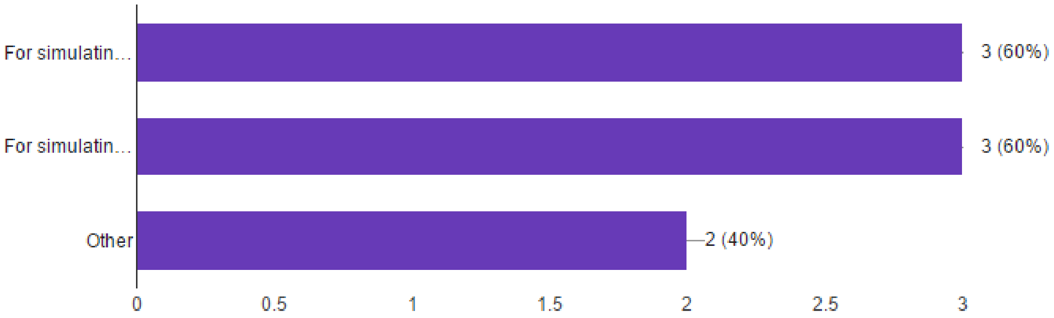
It was easy sitting down while playing in room-scale

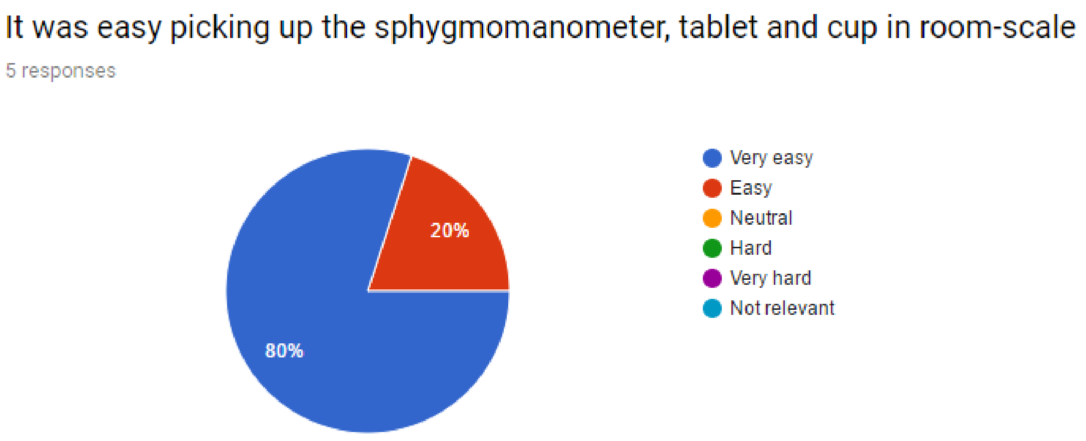
5 responses



Sitting down in room-scale:

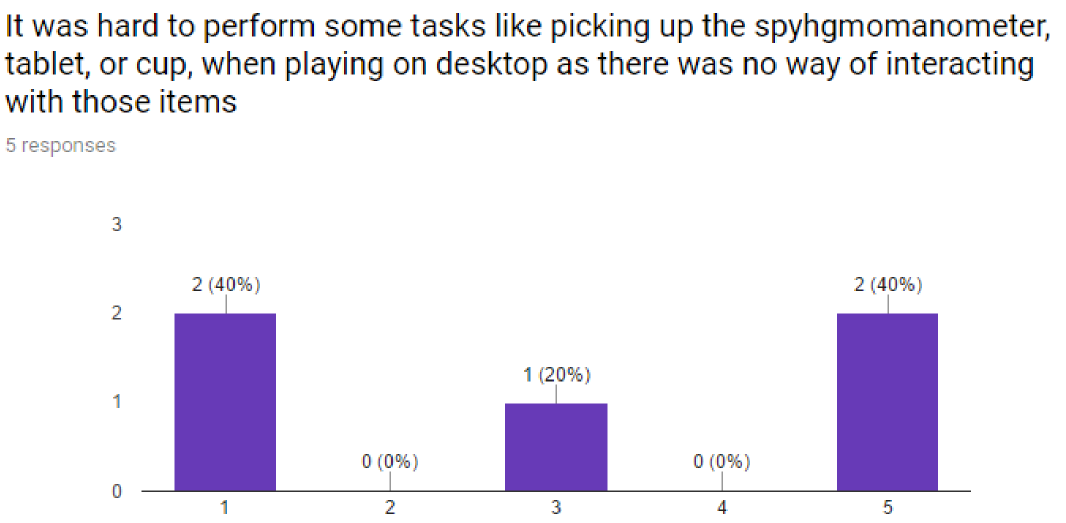
5 responses





It was easy picking up the sphygmomanometer, tablet and cup in room-scale

5 responses



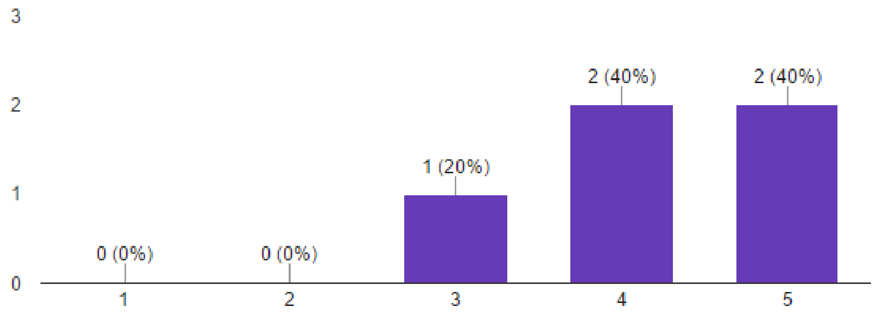
It was hard to perform some tasks like picking up the spyhgmomanometer, tablet, or cup, when playing on desktop as there was no way of interacting with those items

5 responses

Usability (user-friendliness)

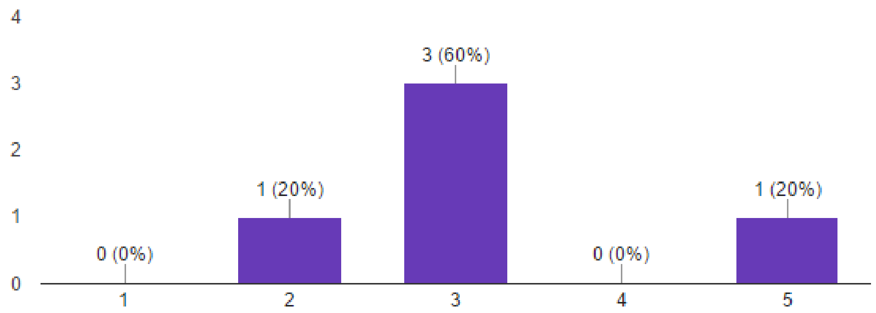
The desktop menus were easy to use, and reading information about the case and scenes was easy.

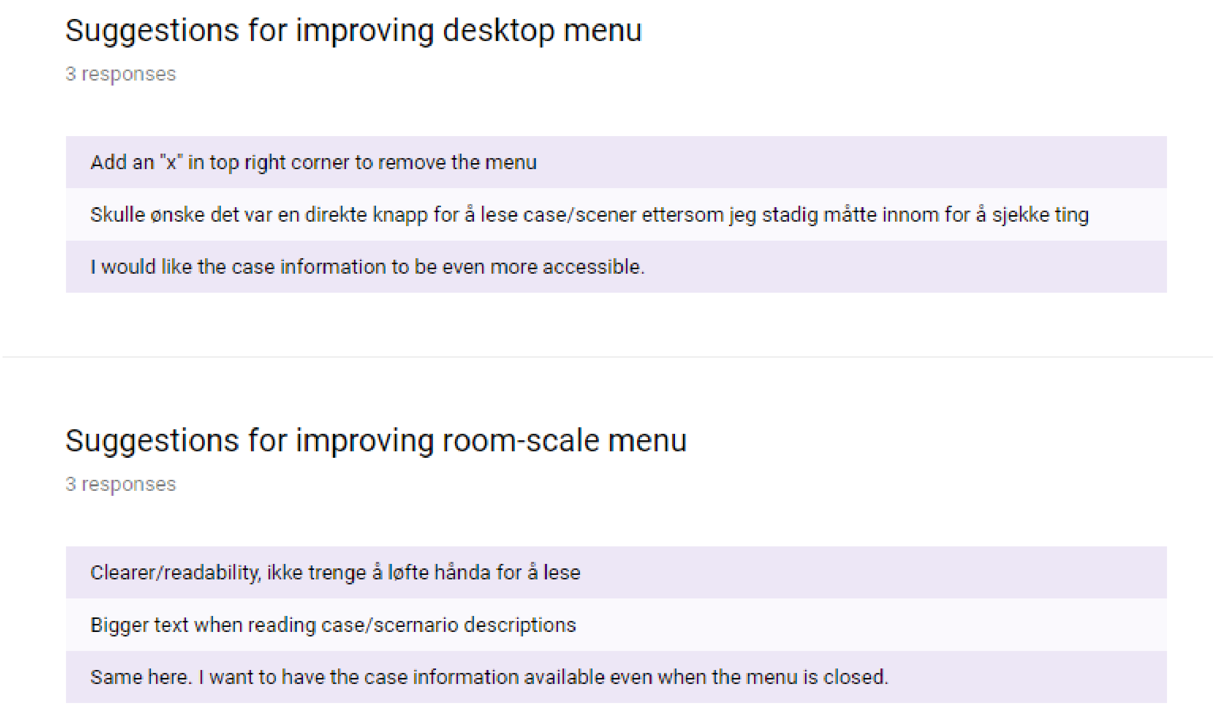
5 responses



The room-scale menus were easy to use, and reading information about the case and scenes was easy.

5 responses





Suggestions for improving desktop menu

3 responses

Add an "x" in top right corner to remove the menu

Skulle ønske det var en direkte knapp for å lese case/scener ettersom jeg stadig måtte innom for å sjekke ting

I would like the case information to be even more accessible.

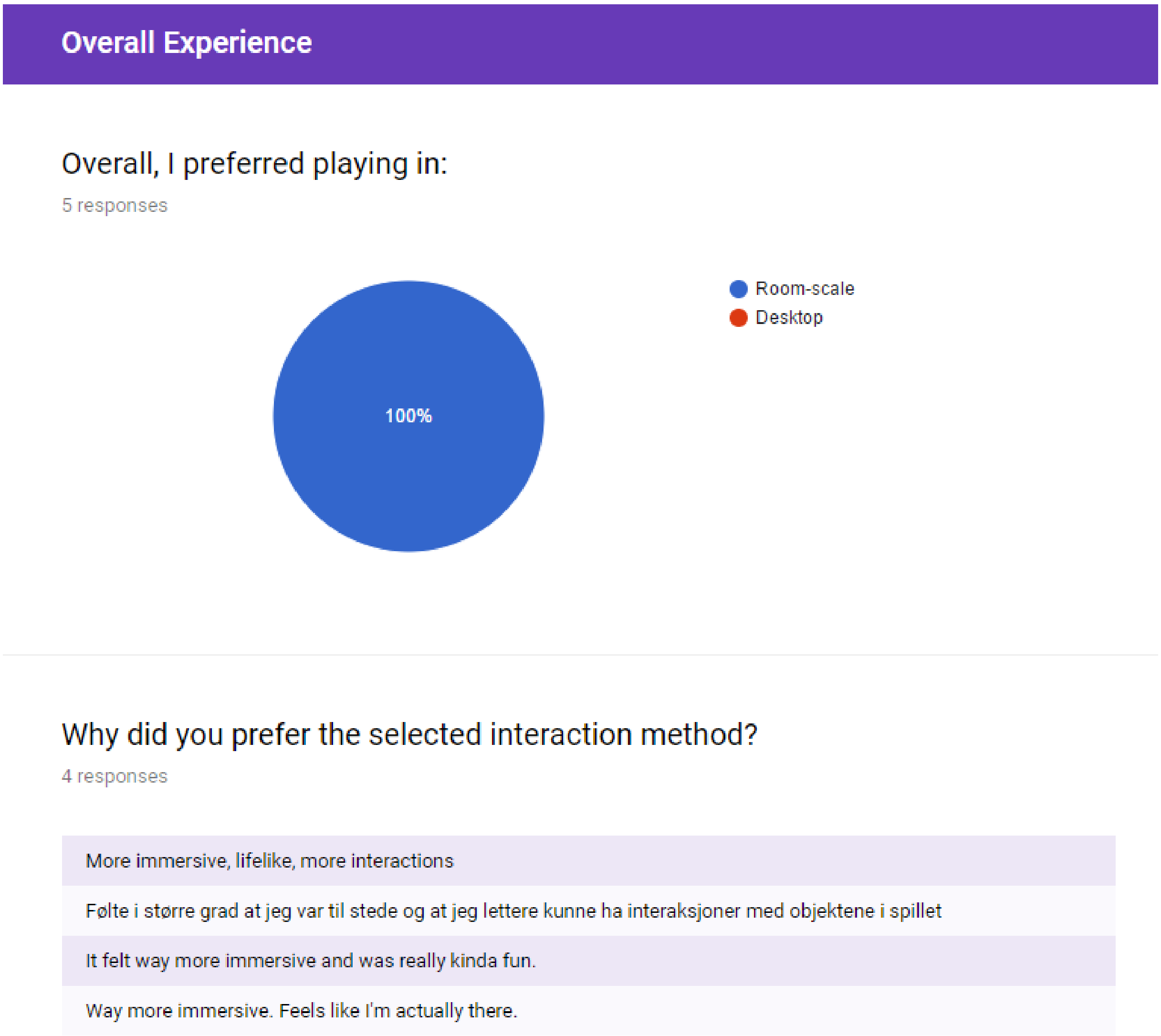
Suggestions for improving room-scale menu

3 responses

Clearer/readability, ikke trenge å løfte hånda for å lese

Bigger text when reading case/scernario descriptions

Same here. I want to have the case information available even when the menu is closed.



Overall Experience

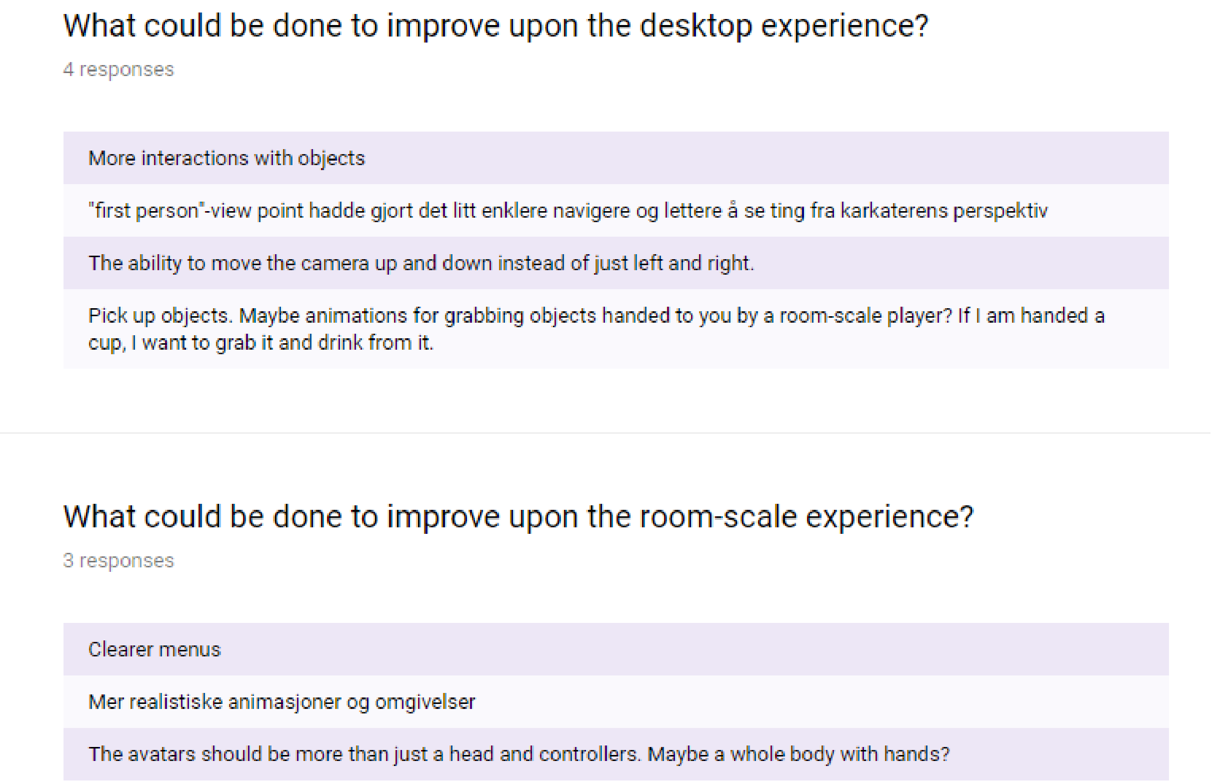
Overall, I preferred playing in:

5 responses

Why did you prefer the selected interaction method?

4 responses

More immersive, lifelike, more interactions
Følte i større grad at jeg var til stede og at jeg lettere kunne ha interaksjoner med objektene i spillet
It felt way more immersive and was really kinda fun.
Way more immersive. Feels like I'm actually there.



What could be done to improve upon the desktop experience?

4 responses

- More interactions with objects
- "first person"-view point hadde gjort det litt enklere navigere og lettere å se ting fra karkaterens perspektiv
- The ability to move the camera up and down instead of just left and right.
- Pick up objects. Maybe animations for grabbing objects handed to you by a room-scale player? If I am handed a cup, I want to grab it and drink from it.

What could be done to improve upon the room-scale experience?

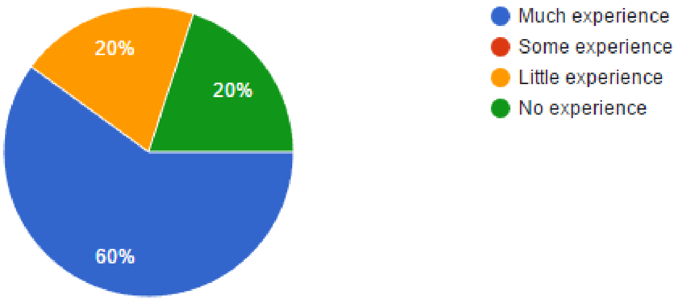
3 responses

- Clearer menus
- Mer realistiske animasjoner og omgivelser
- The avatars should be more than just a head and controllers. Maybe a whole body with hands?

Appendix C VirSam vs Prototype Survey Answers

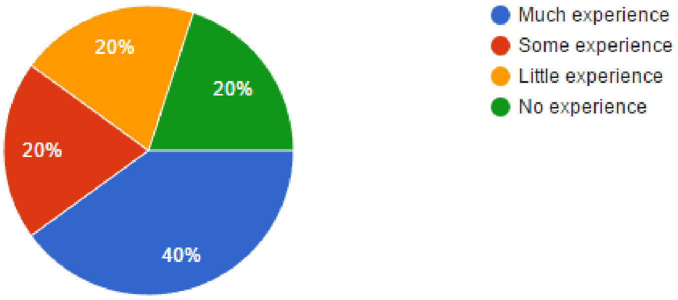
I have experience with virtual worlds (e.g. online games or simulators)

5 responses



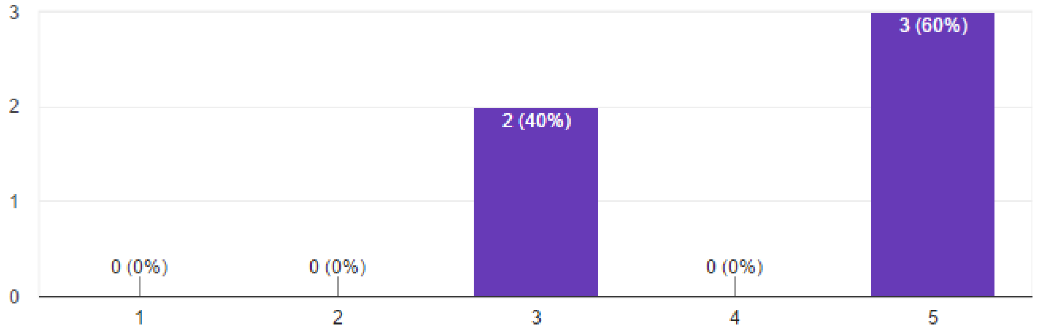
I have experience with virtual reality (e.g Oculus, Vive, Room-scale)

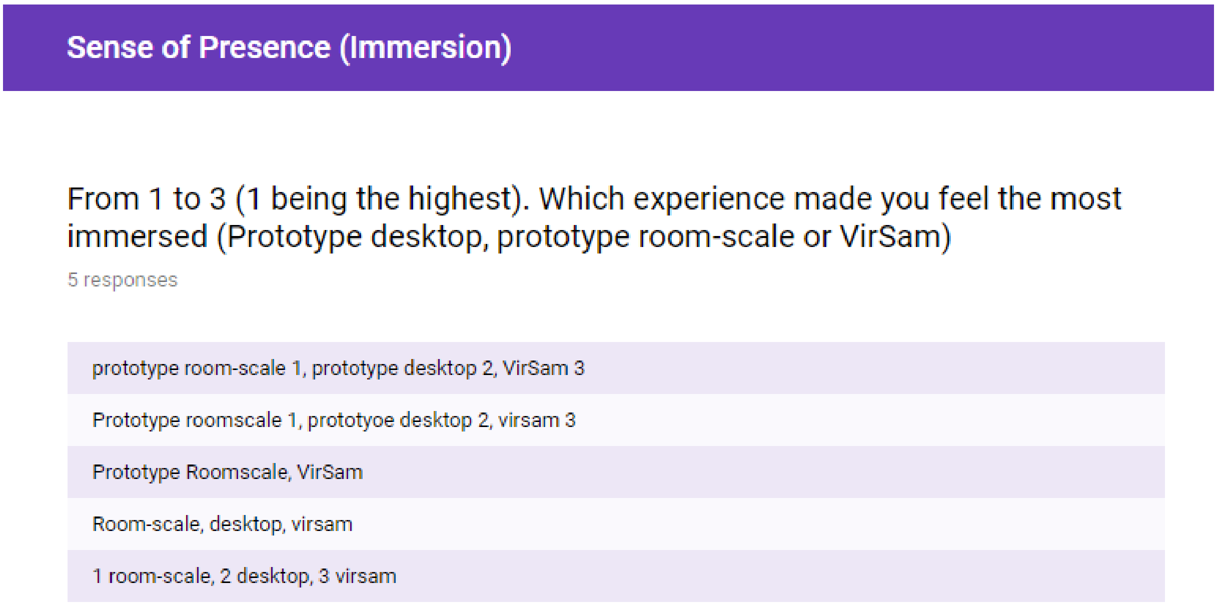
5 responses



How would you rate your expertise of using a PC?

5 responses





Sense of Presence (Immersion)

From 1 to 3 (1 being the highest). Which experience made you feel the most immersed (Prototype desktop, prototype room-scale or VirSam)

5 responses

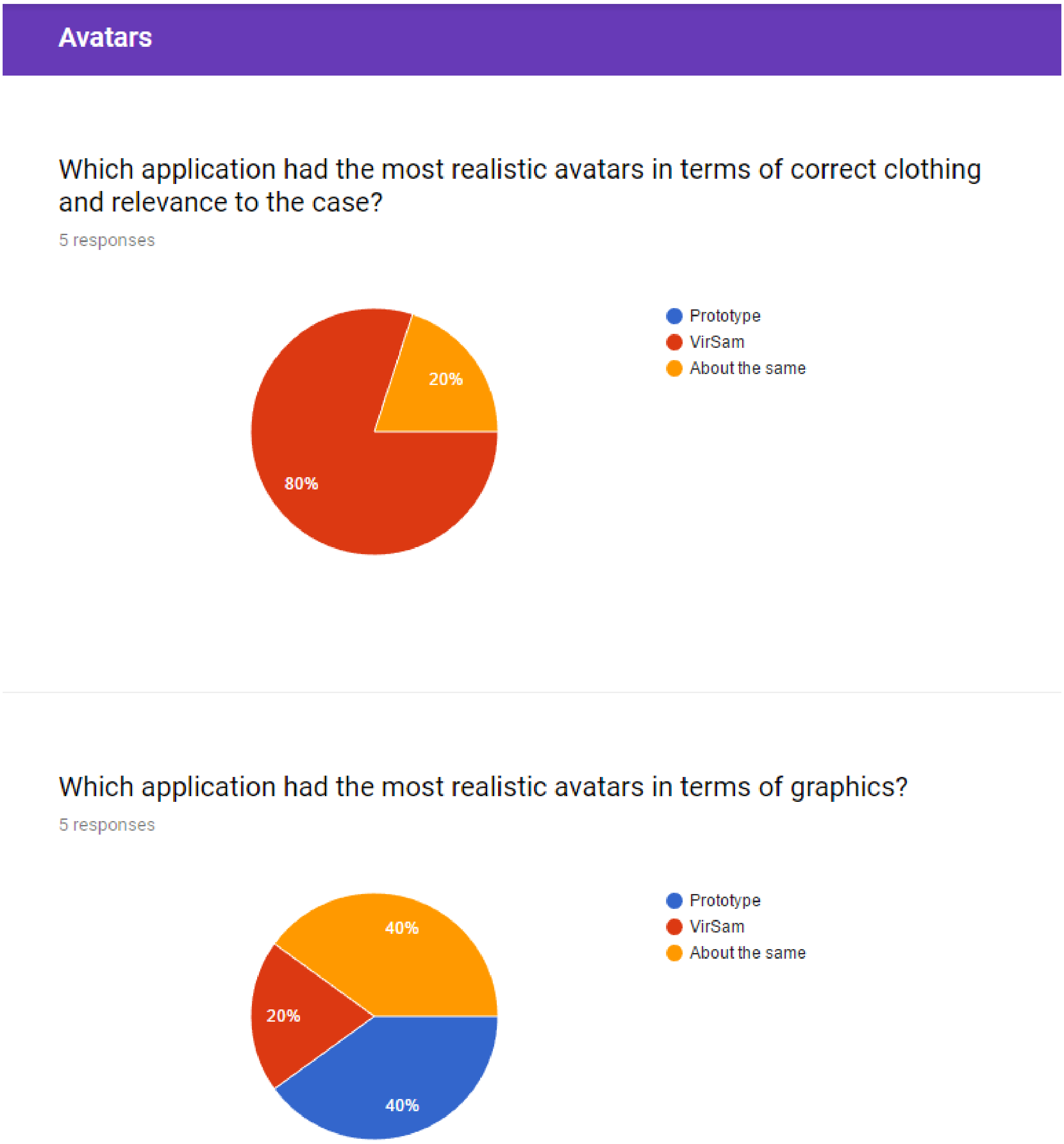
prototype room-scale 1, prototype desktop 2, VirSam 3

Prototype roomscale 1, protototyoe desktop 2, virsam 3

Prototype Roomscale, VirSam

Room-scale, desktop, virsam

1 room-scale, 2 desktop, 3 virsam



Avatars

Which application had the most realistic avatars in terms of correct clothing and relevance to the case?

5 responses

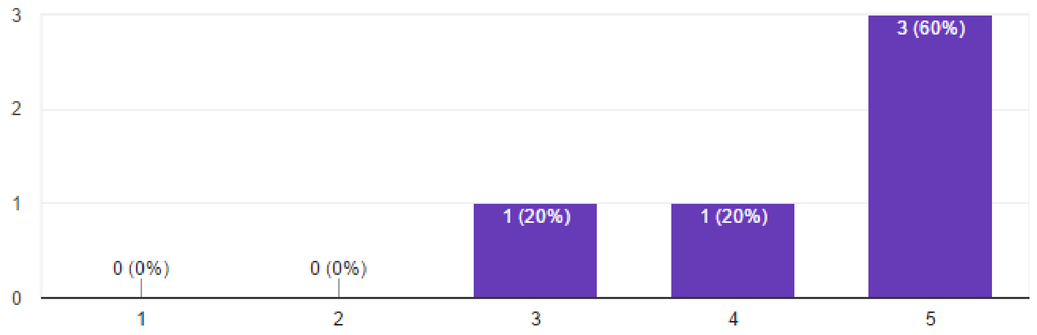
Which application had the most realistic avatars in terms of graphics?

5 responses

Case, identity and role

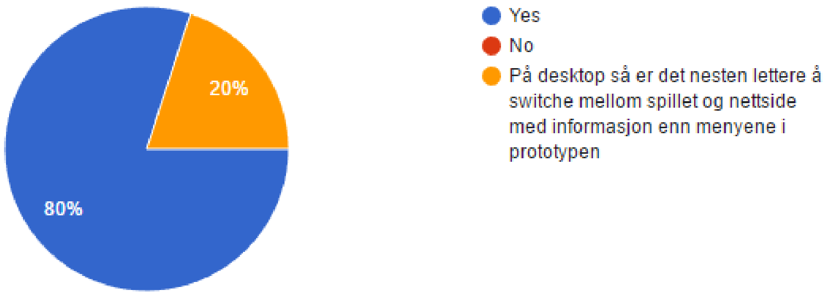
It was frustrating having to read about the case and my role outside the application in VirSam

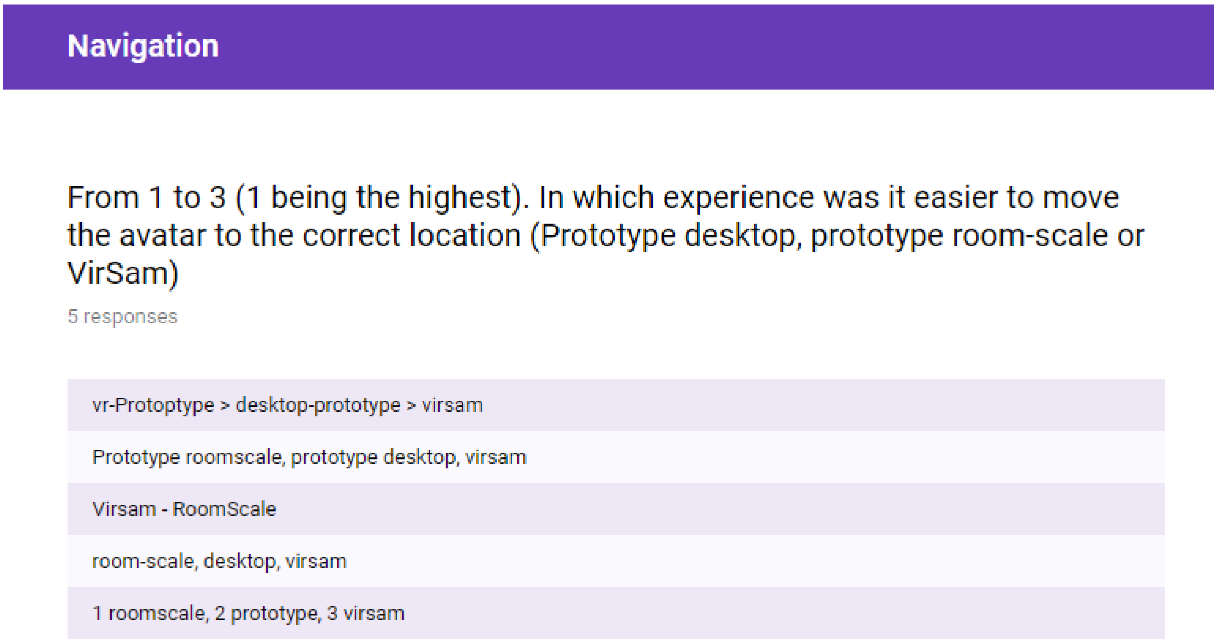
5 responses



Having the case and role information inside the application helped solve the above issue when using the Prototype

5 responses



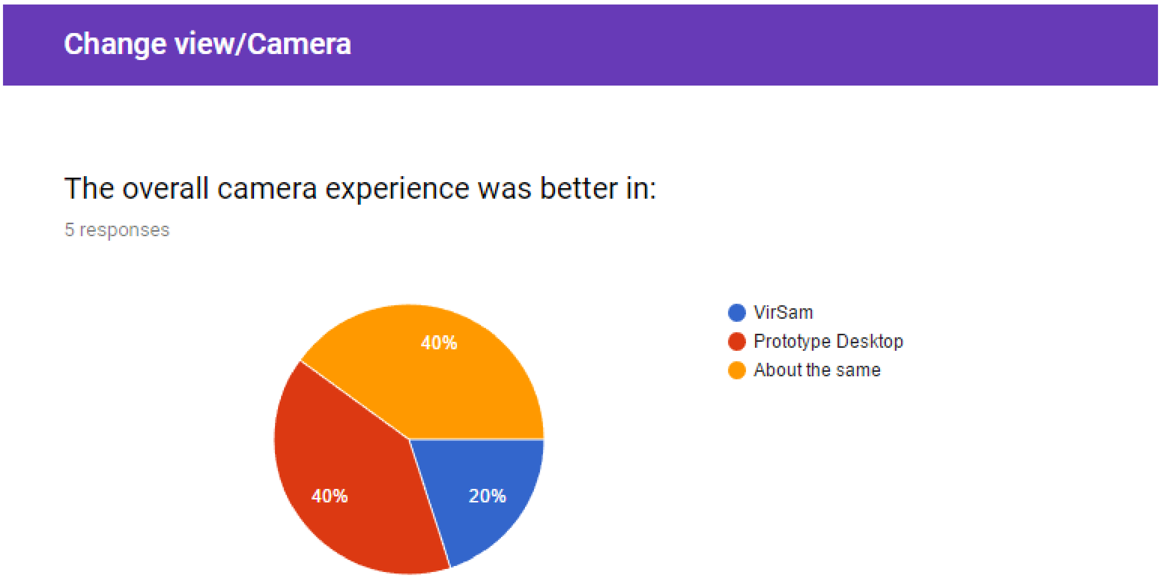


Navigation

From 1 to 3 (1 being the highest). In which experience was it easier to move the avatar to the correct location (Prototype desktop, prototype room-scale or VirSam)

5 responses

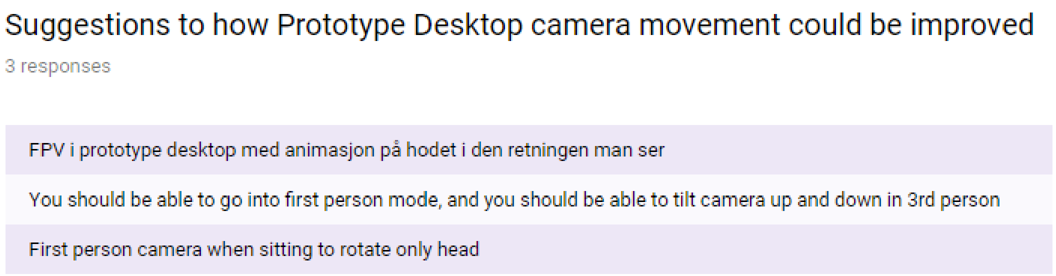
vr-Protoptype > desktop-prototype > virsam
Prototype roomscale, prototype desktop, virsam
Virsam - RoomScale
room-scale, desktop, virsam
1 roomscale, 2 prototype, 3 virsam



Change view/Camera

The overall camera experience was better in:

5 responses



Suggestions to how Prototype Desktop camera movement could be improved

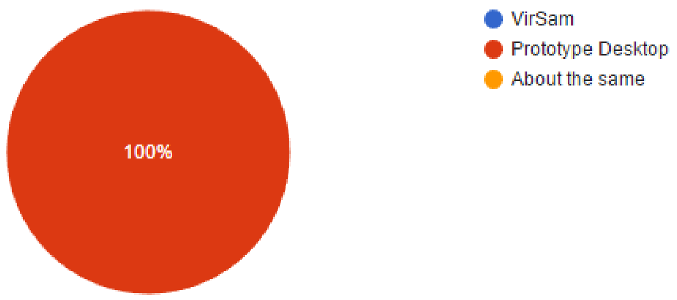
3 responses

FPV i prototype desktop med animasjon på hodet i den retningen man ser
You should be able to go into first person mode, and you should be able to tilt camera up and down in 3rd person
First person camera when sitting to rotate only head

Accessing settings

Finding relevant settings for the application was easier in:

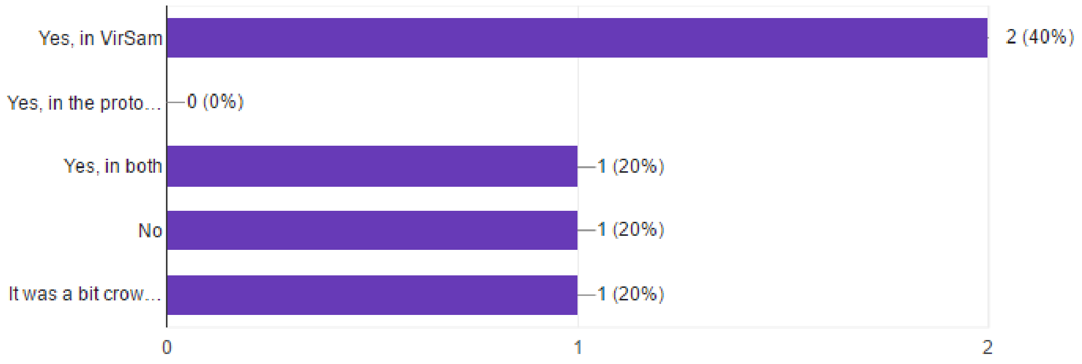
5 responses



Rooms

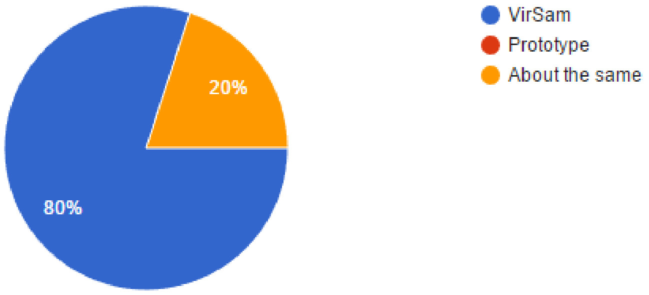
Did the rooms feel too small or crowded in any of the applications?

5 responses



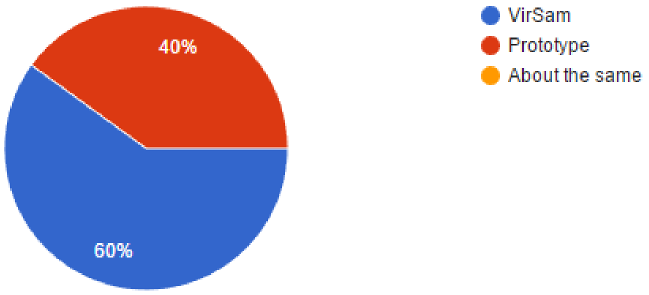
In which application did the rooms look more finished in terms of props?

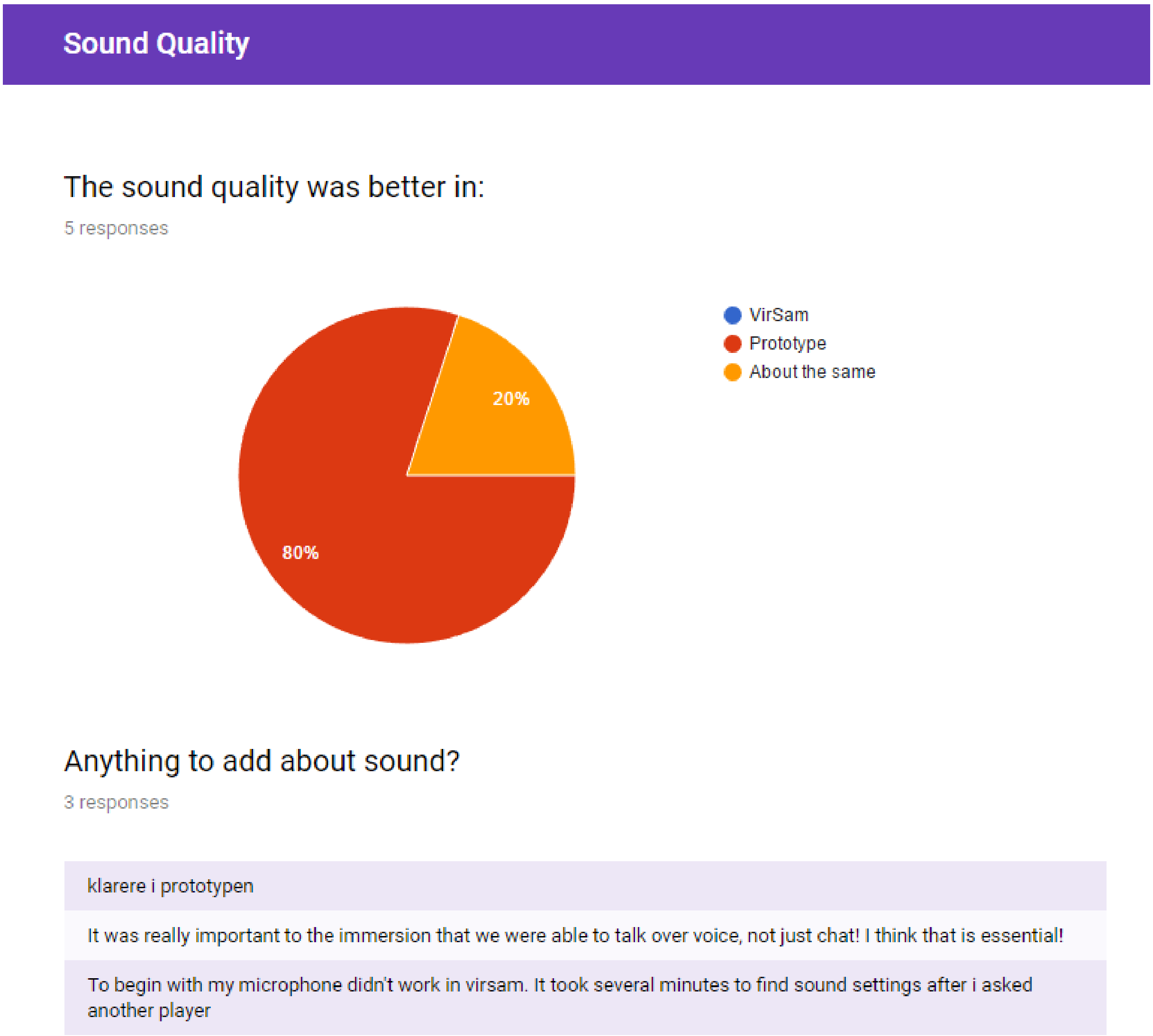
5 responses



In which application did the rooms look more finished in terms of realism/graphics?

5 responses





Sound Quality

The sound quality was better in:

5 responses

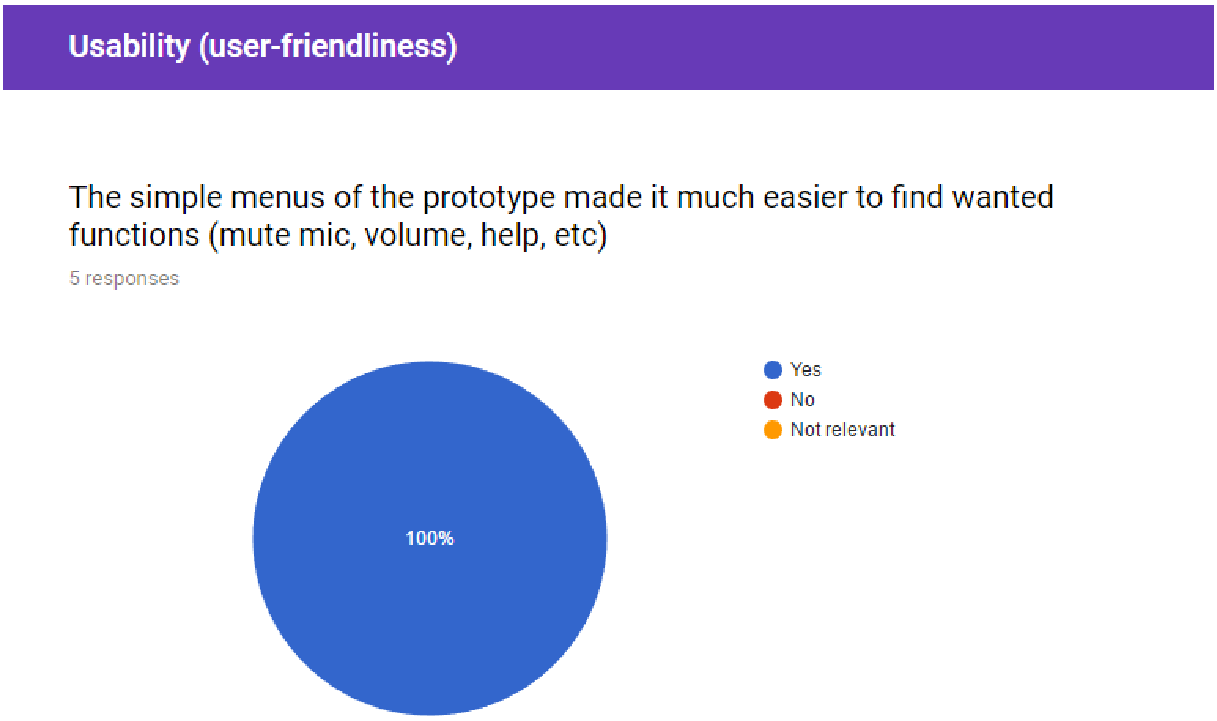
Anything to add about sound?

3 responses

klarere i prototypen

It was really important to the immersion that we were able to talk over voice, not just chat! I think that is essential!

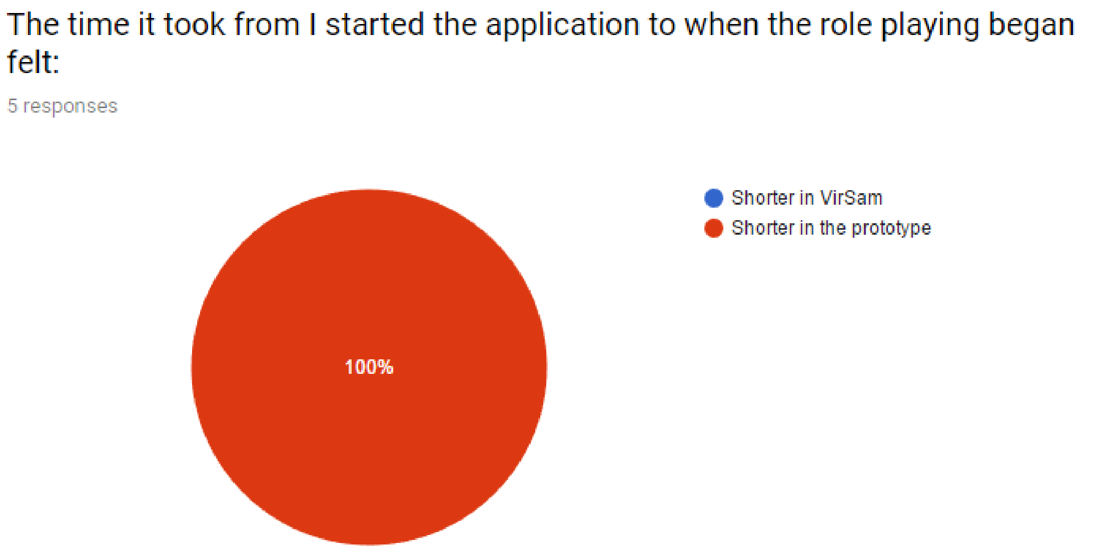
To begin with my microphone didn't work in virsam. It took several minutes to find sound settings after i asked another player



Usability (user-friendliness)

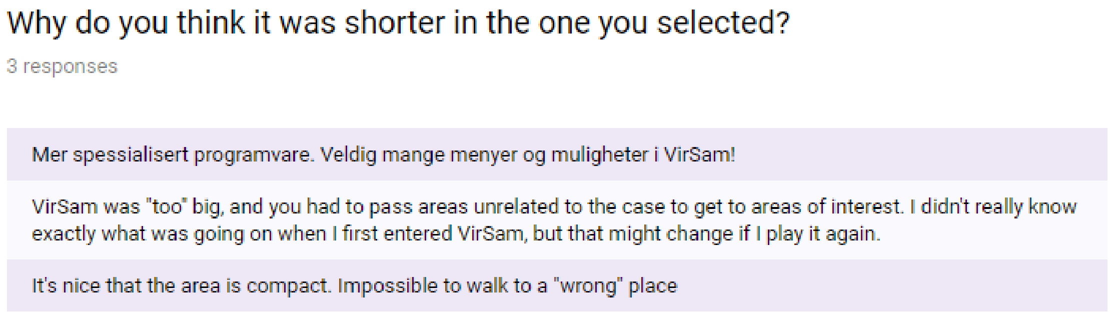
The simple menus of the prototype made it much easier to find wanted functions (mute mic, volume, help, etc)

5 responses



The time it took from I started the application to when the role playing began felt:

5 responses



Why do you think it was shorter in the one you selected?

3 responses

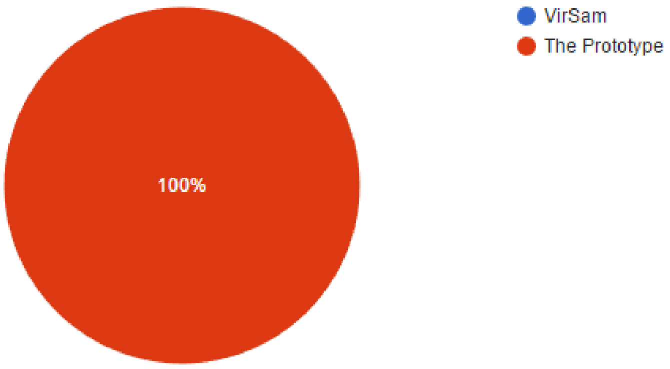
Mer spessialisert programvare. Veldig mange menyer og muligheter i VirSam!

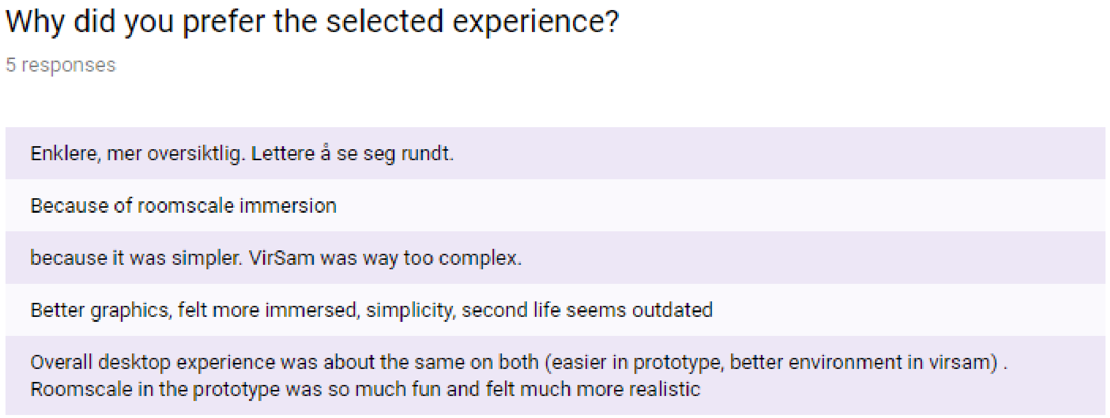
VirSam was "too" big, and you had to pass areas unrelated to the case to get to areas of interest. I didn't really know exactly what was going on when I first entered VirSam, but that might change if I play it again.

It's nice that the area is compact. Impossible to walk to a "wrong" place

Overall, I preferred using

5 responses





Why did you prefer the selected experience?

5 responses

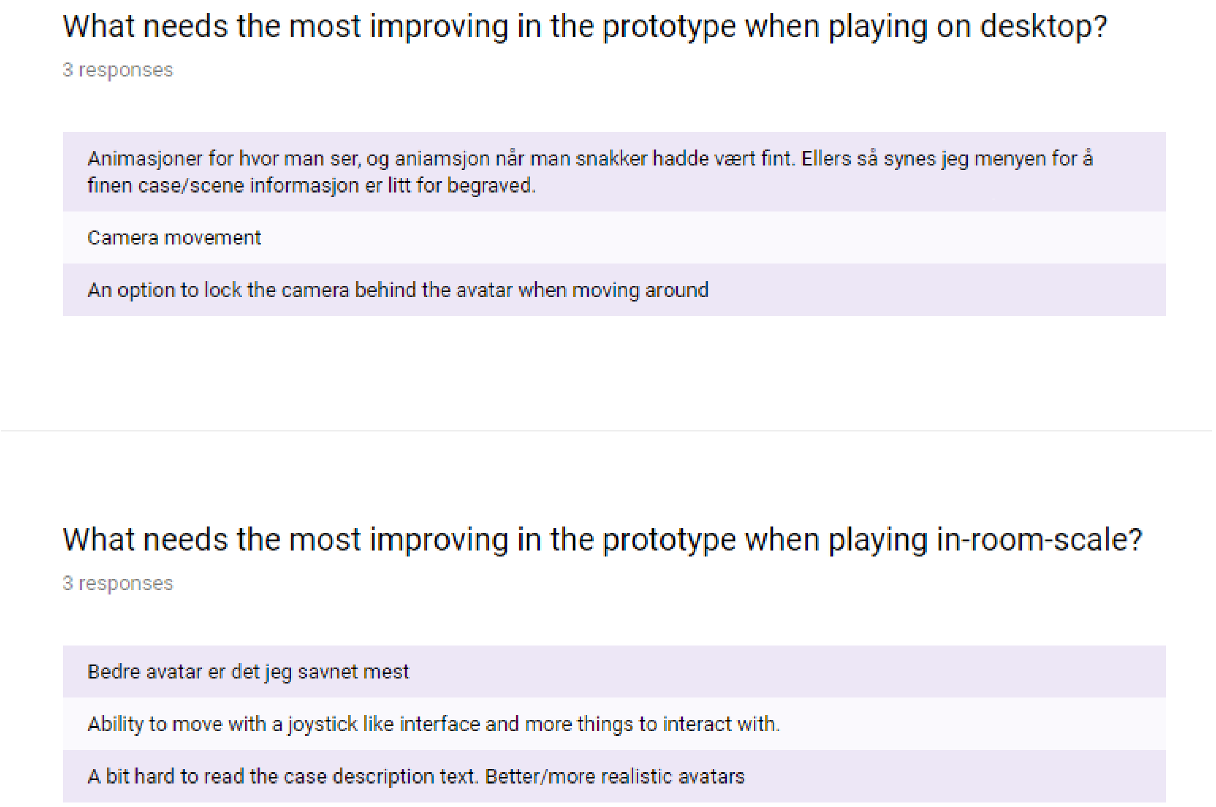
Enklere, mer oversiktlig. Lettere å se seg rundt.

Because of roomscale immersion

because it was simpler. VirSam was way too complex.

Better graphics, felt more immersed, simplicity, second life seems outdated

Overall desktop experience was about the same on both (easier in prototype, better environment in virsam) . Roomscale in the prototype was so much fun and felt much more realistic



What needs the most improving in the prototype when playing on desktop?

3 responses

Animasjoner for hvor man ser, og aniamsjon når man snakker hadde vært fint. Ellers så synes jeg menyen for å finen case/scene informasjon er litt for begraved.

Camera movement

An option to lock the camera behind the avatar when moving around

What needs the most improving in the prototype when playing in-room-scale?

3 responses

Bedre avatar er det jeg savnet mest

Ability to move with a joystick like interface and more things to interact with.

A bit hard to read the case description text. Better/more realistic avatars