UNIVERSITY
OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Using virtual reality for a controlled evaluation of a haptic navigation wearable for people with a visual impairment

Bachelor's Thesis

Tim Yeung

Supervisors:
Dr Angelika Mader
Prof. Dr Jan Van Erp

Enschede, July 2021

Abstract

Visually impaired people are currently severely limited in their ability to navigate their surroundings. While they have the ability to detect certain objects within the range of their white cane, objects outside of this range are hard to detect and nearly impossible to identify.

For this purpose, a haptic navigation wearable is proposed which attempts to improve the mobility of these people by giving them more information on objects outside of their cane's reach. This wearable is capable of detecting and identifying objects and can be used by visually impaired people to find specific navigationally significant objects such as pedestrian crossings, doors, and stairs..

In order to evaluate such a device, a reliable, reproducible, and safe way of evaluation is required. This project presents the usage of VR peripherals, digital environments mapped to real life environments, and automation as a way to test and evaluate such a device.

The evaluation of the haptic wearable using this VR method has shown that, while limited in scope, this wearable shows a proof of concept, allowing users to better understand their environment. For future works, it is proposed that more development is done on the haptic language, the portability of the device, and the implementation of non-euclidean spaces to map large virtual environments to smaller real life test areas.

Contents

Chapter 1: Introduction

1.1 Context

Visually impaired people are currently limited in their use of senses to navigate around their environment. While they can use sensory and aural feedback through the use of white canes and their ears to get around safely, there has yet to be a more definitive technological solution to this problem.

Haptic wearable devices, using micro sensors and actuators, would have the possibility to both map out the information of their surroundings and communicate this information to the user, without further adding communication load to the auditory senses. Such haptic wearable solutions could be used to further help a visually impaired navigate their surroundings in an intuitive way and improve their navigation capabilities in situations where the sole use of conventional tools such as white canes fail.

This graduation project is done as part of a larger project in collaboration with other graduation students. The larger project contains the entire scope of developing this haptic wearable, with each student handling a different subdomain.

The device that will be developed for this project uses a sensor array, which generates and processes data on the environment of the user. This data is then passed on to an actuation system consisting of vibration motors which is able to convey this information to the user using different signals and patterns. However, the generation of such algorithms and the creation of a system for processing the data is a time intensive task. Furthermore, testing a novel sensor system in potentially dangerous environments such as pavements near ditches or train platforms, is undesirable. Due to the scope of this project, the finished sensor system will not be able to be tested by the end of this project. Therefore, in order to still be able to test the device during development in semi-realistic situations and in order to be able to test the concept of the device, a virtual environment system is to be constructed which will be used as a substitute for the sensor system, but will also be capable of generating data similar to the data gathered from the sensors of the sensor system. This data will then be able to be used as dummy data for testing the concept of the device in action.

In order to realistically simulate data and test the system, a virtual environment needs to act as a replica of the real world. Furthermore, by mapping this virtual world to the real world, it is possible to have users test the device in a natural way. As an example, one of the situations which poses problems for VI people is the occurrence of hanging objects such as extruding storefront signs [7]. Rather than testing the device in this situation in real life, a virtual environment can be built that copies this real situation 1 to 1. By having testers stand in an area cleared of obstacles and by tracking their movements, it is possible to simulate the position of the tester and the device in the virtual environment. Now, if the tester would accidentally walk into the location where the sign would be in the virtual world, they would not get hurt as the sign would not exist in the real world. In recent years, the usage of VR peripherals for this purpose has grown, with recorded academic usage of VR as an evaluation tool for projects such as autonomous vehicles[1] and eye controlled wheelchairs[2]. In using these virtual environments rather than real environments, situations can be simulated which might otherwise be dangerous to test in real life situations.

Furthermore, in iterating and designing a wearable, appropriate evaluation requirements, measurements of success, and experimental setups should be identified. This will aid the design process as a whole and adhere to the general standard practices in developing products.

This project specifically will deal with the construction of the virtual environment tool for the purposes which have been described above and how the end product can be evaluated and tested using this digital environment.

1.2 Problem Statement

The goal of this project is to develop a VR evaluation tool and to use this evaluation tool in testing the described haptic wearable and by extension, complete the development of this haptic navigation wearable. State of the art research needs to be conducted regarding previous projects in the domain of VI navigation and their methods of evaluation and the usage of VR in simulating real world environments. Furthermore, research should be conducted with VI people using surveys, interviews, and user tests in order to identify and test problematic situations in navigation for VI people. The wearable should be designed to not replace current tools (i.e. white canes), but act as an extension of the latter. Required functionalities include the detecting and communicating of the direction of objects relative to the user and object identification.

Due to the nature of the project, thought should be put into the procurement of data from potential users (i.e. visually impaired people). Through the client and their connections, it should be possible to get into contact with this group, although it should be noted that likely due to the Coronavirus pandemic, there can be issues and limitations in contacting and interacting with potential users. All equipment, i.e. sensors, actuators and VR peripherals are readily available through the EEMCS SmartXP lab and will be sourced from either SmartXP, the client, or personal inventories.

1.3 Research Questions

The goal of this collaborative project is to successfully design, implement and test an assistive navigation wearable or device, using haptic feedback. As such, the main research question is as follows.

How to design a wearable which improves the navigation capabilities of visually impaired people using haptics?

In order to effectively design the evaluation tool, research will be conducted on several fronts.

First, the evaluation methods of previous VI navigation tools will be researched, in order to gain an understanding of how the wearable could be evaluated. This knowledge will subsequently be applied for the design of the VR tool.

Second, the shortcomings of VI people in navigation will be identified in order to make the wearable useful, but also to be able to incorporate these situations into the evaluation. As a result the following sub research questions are identified.

SQ1:

What are the shortcomings of the current way people with a visual impairment navigate?

SQ2:

What are possible use cases of such a haptic navigation device for people with a visual impairment.

SQ3:

What are the testing methods and evaluation criteria for previous navigation devices for people with a visual impairment?

SQ4:

How can a real world environment be mapped to and interact with a virtual environment?

SQ5:

How to design a tool using a VR environment that allows for the evaluation of a haptic navigation device for people with a visual impairment?

Chapter 2: Exploration

The scope of this project concerns two things, the creation of a VR tool for evaluating the actuation and the design of a test/evaluation plan. In order to execute both, preliminary research is required.

First of all, shortcomings in daily navigation of people with a visual impairment should be identified. From these shortcomings, the possible use cases of the device should be investigated in order to determine the scope of the device and to develop a proper test strategy.

Second, previous projects concerning VI navigation should be explored in order to gain an understanding in common practices for evaluating such systems. This information can then be synthesized and used to create a more complete evaluation plan for this project and identify the requirements of both the tool and the plan.

Finally, state of the art should be explored on the usage of VR in HMI evaluations and particularly on the usage of tracking and mapping of spaces in VR. This information will serve as a guide to implementing the VR tool.

2.1 Shortcomings in VI navigation

The purpose of the proposed VR tool is to substitute the data from the sensors of the device, by simulating a real environment. Therefore, the requirements of this tool are directly related to the requirements of the sensors and the design and limitations of the sensors. In order to identify the requirements of the sensors, research will be conducted regarding the shortcomings in current navigation aids, useful features for new navigation aids, and situations where using only a white cane is not sufficient in effectively navigating. This is done with a focus group and conducted with 3 VI participants. Further information on the focus group and the result of the focus group is provided in the sections below.

2.1.1 Focus Group

The goal of the focus group is to identify the shortcomings in daily VI navigation. It consists of 3 people ranging from 28 to 65 years old. Levels of visual impairment range from complete blindness in a single eye, with 97% blindness in the other, to complete blindness. Main areas of interest include the problems faced in navigation, the usage and effectiveness of currently used navigation solutions, and design criteria for the to be designed device.

Problems in navigation which are found during the interviews are analysed and categorized into several categories; point to point navigation, orientational navigation, and obstacle avoidance.

Point to point navigation is defined as navigation which deals with long term navigation. An example of this would be aiding the user in navigating from their house to a bus stop a few streets down.

Orientational navigation is defined as navigational aspects which deal with navigation through visible clues. This includes the identification of navigationally significant objects such as doors and road signs within the field of view of the user.

Finally, obstacle avoidance navigation is defined as navigational aspects which deal with identifying and avoiding static or moving obstacles in their path.

The usage of navigation aids by the members of the focus group is inquired as well. This information is subsequently analysed to see if there are any patterns in devices which are deemed useful or unuseful. This information can then be used to decide whether specific functionalities should be included or avoided while designing the device.

Lastly, the focus group is asked about their preferences in using and wearing the device, such as possible locations for carrying the device, i.e. on the head or on the chest, and restrictions on weight.

The focus group is to be conducted using an unstructured interview format. From each interview, notes are taken and discussed among the project members. The full list of interview questions and notes for each interview can be found under Appendix A.

2.1.2 Conclusion use cases

People with a visual impairment are relatively capable of navigating known surroundings which are static and are within reach of a white cane. However, problems arise when they are unfamiliar with their surroundings. As a result, they will rarely travel to new locations, as the route to go to such a place will need to be learned with someone who can see. This is due to the fact that VI people tend to navigate using “orientation points” which they use to determine where on their route they are and when they need to change directions, etc. These orientation points are usually objects which can be felt using either the cane or their extremities. However, finding these objects without any pre knowledge would be impossible. This is also the case when navigating buildings and open spaces. For example when navigating towards a shop in a shopping centre, it tends to be hard for VI people to find the entrance to a shop . While navigation applications such as Google Maps can direct users to the general location of a shop, the limitations of gps and information gathering, usually result in Maps directing the user to the general location of the entrance, which is not precise enough for a person with a visual impairment to find an entrance.

Furthermore, there is also the issue of hard to detect obstacles. In general, hanging objects and any objects which are not attached at ground level are hard to detect as they cannot be found with a white cane. Furthermore, objects near the floor with a significant height difference can be hard to detect as well, such as a drop to the road from the pavement. These objects can result in painful accidents, which further demoralize these people in exploring the world.

Lastly, outside the realm of navigation, people with a visual impairment have trouble finding objects in general. In the case an object is placed on a table, and the person forgets where they put that object, it is hard for them to retrieve said object. The same situation can be found when they drop something on the floor.

In conclusion, the main areas of shortcomings in VI navigation concerns the finding and identification of objects which have navigational significance and finding hard to find obstacles.

2.2 Evaluation and Testing

In order to determine how the device should be evaluated and a test plan, a literature review is conducted in order to find patterns and categorizations in how previous academic projects relating to VI navigation were evaluated. The research questions followed during this research are focused on what the goals are of evaluating a navigation device, what type of data is measured when evaluating these goals, and how this data is obtained. This review has been compacted and summarized in the sections below.

2.2.1 Categorizing goals of evaluation

When looking at all of the articles, recurring goals can be abstracted from the evaluations described. To clarify, a goal is defined as the general purpose of an evaluation. Of course, since a paper can describe multiple evaluations, it is possible that a single paper will contain multiple evaluation goals.

Among the researched papers, a total of 4 recurring goals can be found. These will be further described as domains.

The first domain is the evaluation of sensors. Evaluations in this domain focus on the testing of sensors used in the navigation devices. This evaluation is used to determine whether the performance of the sensor system is capable enough to fulfill the system's requirements. An example of this domain of evaluation can be seen in a project report by Singh & Kapoor [5] which describes a smart cane using ultrasonic sensors to detect its surroundings. For this project, the ultrasonic sensors were tested. A project can also test multiple sensors separately such as a project by Khan et al. [14] which evaluated both ultrasonic sensors and cameras using object detection individually.

The second domain is the evaluation of the feedback of a device. These kinds of evaluations are used to test the method of feedback employed in the system. The goal of this evaluation is to test the intuitiveness and effectiveness of the feedback, not so much the performance of the actuators themselves. It does not matter what type of feedback is evaluated. For example, a smart cane devised by Nasser et al. (2020) [15] uses thermal feedback in order to communicate directional information. Another project by Alzighaibi et al. (2020) [12], uses haptic feedback on a foot sole to communicate directional information. Even though the method of feedback is different, both still aim to evaluate the factors concerning the feedback, therefore both evaluations are categorized as feedback evaluations.

The third domain is the evaluation of functional effectiveness. Evaluations in this domain concern the overall effectiveness of the device in real life situations and tests a combination of both sensors and feedback. Rather than looking directly at the sensors and feedback, the combined effectiveness is typically measured by looking at variables which are indirectly influenced by these components. Examples of this can be found in articles by Nair et al. [16] and Giudice at al. [18] where factors such as average walking speed and the amount of errors in navigation are measured.

The last domain is the evaluation of the user experience. The goal of evaluations in this domain is to find out what the user thinks of the device and to probe the perceived usefulness of the device by users. It mostly consists of subjective data collected through surveys and can be found in nearly all projects testing a full product or MVP.

2.2.2 Types of data measure per domain

In order to get a better overview of the relation between the domains and the type of data that has been measured, the types of data will be discussed for each previously mentioned domain.

2.2.2.1 Evaluation of Sensors

With sensors being one of the pivotal parts of any smart device, the evaluation of them would logically be common among projects involving object navigation.

When evaluating sensors, the goal is to gain insight into the reliability of the sensor. The accuracy of a device's sensors is evaluated by comparing the distance measured by the system and the actual distance as set by the experimental setup [5][14]. Furthermore, projects employing the evaluation of sensors will typically focus on the placement and combination of sensors on their device, in relation to the effectiveness of the sensors.

Apart from measuring the accuracy, sensors can also be evaluated in terms of performance speed. As an outlier, Khan et al. [14] are the only one to evaluate this, measuring the average frames per second achieved by their system. This could be explained by the fact that their sensor system uniquely includes an rgb camera and image recognition. Image recognition inherently takes up more processing power compared to processing sensor data using one dimensional data like distance measurements. As a result it is possible that this type of evaluation could be insignificant or redundant for systems employing less sensor data.

2.2.2.2 Evaluation of Feedback

Given the focus on VI people, the performance of the feedback is another logical thing to test. If feedback is unintuitive or hard to distinguish, users may make mistakes in navigation or react too late to the given signals.

Evaluation of feedback can be done by comparing the feedback as experienced by the tester and the actual feedback sent out by the device [12] [11]. By measuring the difference in perception of feedback and the actual feedback given, the distinguishability of the feedback and how accurate users interpret the feedback can be evaluated.

The other approach is to measure the duration of time between the start of the feedback and the reaction from the tester [6]. This will not test the correctness of the perceived feedback, but rather seeks to understand how quickly users can react to the signals.

2.2.2.3 Evaluation of Functional Effectiveness

By far the most popular type of measurements are from the domain of functional effectiveness. The frequent usage of testing the overall effectiveness can likely be attributed to the nature of the project which is very much akin to designing a user product. As a result, evaluations end up being similar to forms of user tests.

In order to gauge the effectiveness of the finished product, the most common method is to measure the time it takes for a tester to complete a predetermined course. Two variations of this approach can be found.

First, one can draw conclusions from the average walking speed which is calculated by dividing the total distance of a predetermined course by the time it takes a tester to complete the course [7] [14]. On the other hand, one can opt to only record the time it takes to complete a predetermined course, without calculating average speeds [16] [18]. However, all sources agree on the importance of testing the movement speed of users, as it gives an important indication on the improvement in mobility. With mobility being “important for activity and social participation” [7], it is only logical that mobility is central to the problem being addressed in VI navigation.

Lastly, another indicator of the improvement in mobility is the amount of “events” caused by the system [16]. Nair et al. define these events as “(1) bumps into walls and other obstacles, (2) wrong turns, and (3) needed interventions by the authors while using the app”. The amount of these events is subsequently counted and recorded while users complete the course.

2.2.2.4 Evaluation of User Experience

Lastly, part of any product development process is the usability test. Usability tests are used to evaluate the acceptability and are considered “paramount to the successful development” of a VI tool [16].

The performance of a device’s usability is typically defined using either a 5 point likert [14] [18] or 10 point [15] score based on survey questions. The questions probe a variety of topics and include the comfort of using/wearing the device [14], the perceived usefulness and helpfulness in mobility [14] [15] [16] [18], the preference compared to conventional navigation tools [14] [18], demand of the user [15], the amount of effort exerted or ease of use [15] [16] [18], the amount of frustration in using the device [15], the general score of the device [15], ease of navigation (with or without device) [16], and the confidence in using the device compared to without [18].

2.2.3 Measurement methods and setup

For testing objective aspects, systems can be tested in an experimental setup, as done by Alzighaibi et al. [12] and Bizon-Angov et al. [11], where testers were sat down in a room and were given feedback based on simulated input. In other words, rather than testing the feedback in a real life situation using input from the sensors, the feedback can be tested in a systematic fashion with researchers controlling when and which feedback signals are sent. However, systems might also be tested in a more realistic setup, where users will typically complete a course within a real life setting [18], or within a controlled environment which mimics a real life environment, using objects such as cardboard boxes to simulate obstacles [7].

In both cases there is a tradeoff between faithfulness to real life situations and the ethicality of potentially hurting the VI testers either physically or mentally. As Dos Santos et al. [7] explain, their experiments were designed in a specific way, prioritizing the ethicality of the experiment as "walking into these obstacles could have caused unpleasant embarrassment among the visually impaired participants" [7].

Finally, in order to test the subjective aspects, surveys can be conducted. In order to increase the efficiency of evaluating both the system performance and the user experience, surveys can be conducted before and after the objective tests [16].

Surveys before the test are used to gather information on the participants and their current state or situation, i.e. experience with using navigation tools, general difficulty in travel.

On the other hand, surveys after the test are used to gauge the experience of the user during the test and therefore will typically only contain questions regarding the system itself [15] or questions comparing the system and the current state of the art [14] [16] [18].

2.2.4 Summary of evaluation methods

In conclusion, there are a few patterns that emerge in evaluating object navigation systems for people with a Visual Impairment.

Evaluations of such systems typically aim to evaluate either individual aspects or the system as a whole. Individual evaluations include sensor evaluation, feedback evaluation, and user experience evaluation. Multiple types of evaluation can also be combined depending on the needs and focus of the project. A new project for developing a new navigation system should pick at least one of these goals of evaluation depending on the focus of the project.

Data measured depends on the goal of the evaluation. Evaluations aiming to test the sensors of a system should include measurements of accuracy and speed. Evaluations testing the feedback of a system should aim to test the intuitiveness through the accuracy at which users can identify different feedback signals and how quickly they can do so. User experience evaluation should focus on the additional value in using the system, the efforts of using the system, and optionally the experience of being a VI person. If the whole system is to be evaluated, developers should include measures that measure the increase in mobility, as this is the main problem addressed in VI navigation systems.

Finally, developers often have to choose whether they want to use a highly controlled environment such as a mock course, or a more realistic but less controlled environment. Ethical responsibility and faithfulness of recreating a realistic environment should be considered when making this decision.

2.2.5 Conclusion on Evaluation and Requirements

Using the knowledge gained from the literature review, the following conclusions are made regarding the design and requirements of testing and evaluating this project.

First of all, for the purpose of this project, the domain of functional effectiveness and the domain of user experience is most relevant, as the division of tasks in this shared project makes it impractical to evaluate the individual systems.

In the development of the device for this project, an iterative approach will be taken. This means that there will be several rounds of evaluations throughout the development process. Sensor systems and actuation systems will be tested separately in the early stages of development to ensure that the minimum requirements of those components are met. Then, in the later stages of development, the system as a whole will be evaluated together with the user experience.

Second, where possible, data should be collected by the systems themselves in order to determine the performance, reliability, and accuracy of the system. This will particularly apply to the sensor system, which should record data on what objects are identified and their relative location in order to be checked for accuracy.

2.3 Implementation of VR

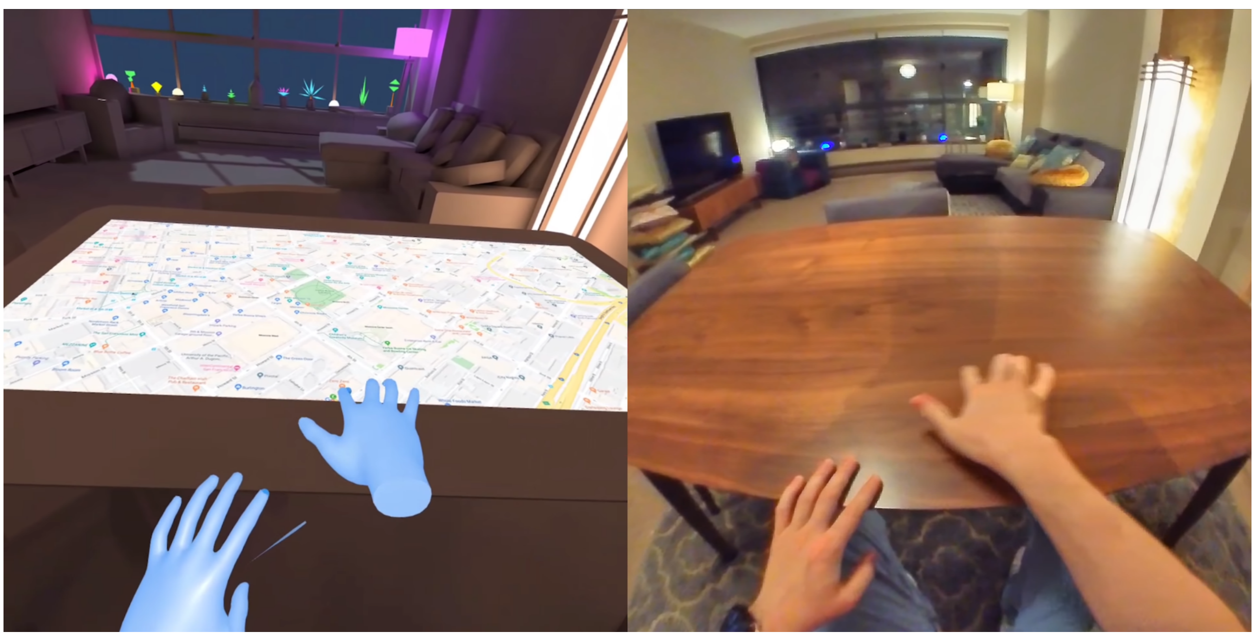
Given the requirements of the VR tool determined by the general requirements as identified in 2.1 and the evaluation plan as described in 2.2, the main functionality of the tool will include the identification of objects and obstacles in the virtual environment, the ability to track movement in the real world to the virtual environment, and the ability scale the virtual world to fit the real world, in order to best simulate the real sensors. In order to gain an oversight of current relevant technologies, the following section will contain a brief state of the art overview.

2.3.1 State of the art

In the academic world, the usage of VR has mainly been limited to the context of using the headset of a VR device to create an immersion as done by Shi et al. [1] and Diederichs et al. [3]. However the usage of tracking real life movement in a virtual environment for research purposes does not appear to have been practiced yet, or at the very least is still very obscure. On the other hand, there is quite some popularity of mapping VR to the real world in hi-tech communities. In order to gain an understanding of the possibilities and devices used, the state of the art will focus on the usage of VR to map the real world to the virtual world in non-academic settings.

2.3.1.1 Greg Madison - Hand tracking on flat surfaces [19]

Oculus Quest



Greg Madison is an Interaction and UX designer for Unity Technologies. Outside of his work, he uploads videos on his youtube channel which includes experiments on the mapping of his home environment to a matching one in VR. In his latest endeavours, he has made use of the Oculus Quest and its hand tracking capabilities in order to create interactive surfaces inside of his house on real life surfaces. Of interest is the fact that his entire apartment has been accurately mapped to VR, allowing him to move around freely (as the Oculus Quest does not require a connection to a PC) and interact with real life objects in VR.

This project greatly demonstrates the mobility and flexibility of the Oculus Quest 2. As the device is wireless and light, it will minimally impair the testers of the wearable. Together with the fact that the Oculus Quest 2 is readily available for this project, this makes a strong case in using the peripheral in implementing the tracking.

2.3.1.2 MediaMonks SP - Into The Wild (Singapore ArtScience Museum) [20]

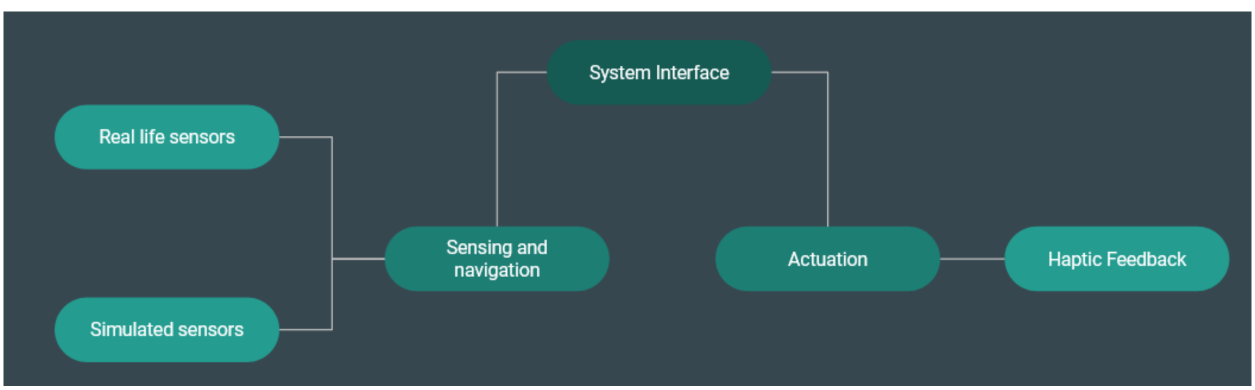
Lenovo Phab 2 Pro

Into The Wild was an interactive experience that was exhibited at the ArtScience Museum in Singapore. The experience saw visitors use a tablet issued by the museum as a viewfinder and explore a virtual world depicting a rainforest. This virtual world itself was mapped to the building of the ArtScience Museum, allowing visitors to walk around the museum while traversing the virtual world as well. In an article written by the Technical Director of the Team Rene Bokhorst [20], the following technical aspects were of interest to this project. First of all they needed a device that was capable of tracking their 3d position and orientation inside of a given space. The tracking was also required to be accurate and fast enough in order to maintain the immersion. Second, Unity3D was used to create and render the virtual environment onto the camera feed of the tablet. Lastly, they discuss the method of lining up the real world with the virtual world. In order to do this a scale must be set for the virtual objects to make sure that they are using the same measurements as real life. From there, both environments are "overlaid" by shifting the position of the virtual world over "anchors", points where the real world and the virtual world would take as an origin point.. These anchors needed to be multiple as a single anchor is not enough to determine a 3d plane such as the ground.

The approach and implementation of this tracking method is useful to this project as it allows the project to map the virtual environment to a physical space. This is helpful to be implemented as it will ensure the safety of the tester. By clearing a predefined space in real life and making sure that the virtual world is contained within that space. The tester will be able to walk around the virtual environment without having to worry about walking against obstacles.

2.5 In conclusion: implementation of the VR tool and the system

Given all of the specifications, the following design was proposed among the members of the project. The entire system consists of 3 sub-systems. First there are the sensory sub-systems, these include the real life sensor system and the VR sensor tool. These systems provide input data to the entire system. This input is sent to an interface connecting the sensors with the actuation. Each call to the input represents a "sentence" and is mapped to an appropriate output which calls the actuation system to create a feedback signal.



The entire system will be attached to a backpack which carries an Intel NUC mini desktop. This desktop directly connects to the sensor array consisting of the Intel RealSense D435 and will act as the processing unit for the sensors and the interface. The actuators will consist of an unspecified collection of vibration motors which will be controlled by a TinyPICO ESP32. The TinyPICO in turn will connect with the processing unit and interface through bluetooth.

For this project, VR will be used only as a substitute to real life sensors in the early stages of development. Due to popularity and accessibility, Unity3D will be used to implement the virtual environment. For the tracking of movement in the virtual world, the Oculus Quest 2 peripheral will be used. Subsequently, the virtual environment and the simulation of data will be run in Unity on the Quest itself. Since the processing unit and interface are housed in a separate hardware unit, the Quest will connect to the interface using TCP protocol.

The Virtual sensors will be attached to an approximation of a user in the virtual environment and will be used to supplement the dummy data. With the sensors currently being placed on the chest of the user, a strap will be created which can mount a VR controller to the chest of a user in order to track the position in the virtual world. These virtual worlds will contain scenarios which contain navigational points of interests such as doors and staircases. Also, in order to simulate the intended functionality of the sensors, the virtual environment will simulate the sensors by using raycasts within the field of view of the real sensors to check whether the point is visible by the simulated sensor. If so, this data will be processed and outputted in the same way as the real sensor system, which will cause the actuation to generate a signal. This setup will be used when testing the actuation of the system without the real life sensors. Full system requirements can be found in 2.1.3

2.5 In conclusion: evaluation plan

At least one evaluation should be done at the end of the project, containing both the evaluation of functional effectiveness and the evaluation of user experience (see Chapter 2.2). As described in the previous sections, measurements of time, the amount of incidents, and the user's opinion should be recorded to this extent.

In general, all evaluations should follow the same structure. In both cases, the tester will be placed in an empty space in the real world. From there, the virtual environment will be mapped to the constraints of the space and the tester will be blindfolded. Next, they will be led to a specific starting point and given a specific object/orientation point which they will need to find. While they are walking around looking for these orientation points, they will attempt to avoid other obstacles as indicated by the device. After each found object, the time between the finding this object and leaving the starting point will be recorded. Afterwards, users will fill in a survey pertaining to their thoughts on the experiment and the device. A full overview of all the measurements and survey questions is given below.

Measurements for experimental test simulating real life situations

- Measuring time between leaving the starting position and reaching the goal.
- Measuring the total number of incidents while navigating the course.
    - Incidents include collisions with obstacles, or walking outside of the designated test zone

Questions for evaluation Survey (7 point Likert scale) conducted after test

- Measuring the perceived usefulness of the device.
    - How would you rate the device overall?
    - How safe do you feel when using the device?
    - How useful is the device for navigating, compared to only using the white cane?
- Measuring the effort in using the device.
    - How much physical effort did it take to use the device?
    - How much mental effort did it take to use the device?
    - How tiring was it to navigate this situation?
    - How frustrating was it to use the device?
    - How confident are you in interpreting and recognizing information from the device?

Further questions specific to subparts of the project may be appended to this questionnaire by other project members.

Chapter 3: Ideation

3.1 Design of the product and Use Cases

From the shortcomings identified in 2.1, three use cases are proposed after careful consideration. These use cases determine the initial requirements of the sensor system and the actuation system. Subsequently, from the requirements of the sensor system in combination with the evaluation plan for the device, the requirements and the design for the VR tool are determined.

3.1.1 Use case ideation

From the focus group interview a total of 3 use cases are considered. Two of these use cases are tied to a specific subtype of navigational problems identified during the interviews. The last one is identified as a solution to a recurring problem related to the topic of visual impairment, but which is outside of the scope of navigation. A short overview of all the use cases will be given in the sections below.

3.1.1.1 Orientational context navigation and “Last few metres”

This use case is conceived as a result of a recurring problem within the focus group of identifying orientationally significant objects.

People with a visual impairment are unable to detect anything outside of their range, as they can only identify objects using touch and sometimes sound. As a result, objects which are outside of their reach are hard to find and identify. In many situations it is useful to know whether a specific object is in the vicinity of the user and if so, where that object is located in relation to the user. This can be useful as VI people will use certain objects for which they know the location of, in order to navigate. However, it can sometimes be difficult to find those objects if the context is lost, i.e. when the person loses their bearing. Furthermore, this situation is also relevant in the phenomenon of “the last few metres”. Due to the limitations of GPS and data storage, pedestrian navigation applications such as Google Maps will bring you to the general vicinity of an intended destination. However, if this destination is a shop or another building with an entrance. The user would still need to navigate towards this entrance. In this case, it is useful for users to be able to find doors in the vicinity as the nearest door would likely be the entrance to the intended location. In order to further increase the scope of the device, it is also proposed that the device would keep track of obstacles in front of the user in order to make sure that the user can avoid those on their way to the desired location.

3.1.1.2 Waypoint navigation

This use case is further tied to the topic of finding objects with orientational significance.

As mentioned earlier, VI people use "orientational objects" in order to keep track of their location. When they want to go to a certain location, this will mean that they first need to learn the route to this location by remembering objects en route which they can use to determine when to turn into a new street, etc. To this use, previous devices exist which can save the GPS coordinates of locations which the user can set. When the user then approaches one of these locations, and thereby the orientational object, the device will tell the user the name of the current location and possible instructions as programmed by the user. The goal of this use case is to further improve on this concept by combining the usage of setting GPS coordinates with object avoidance and guidance towards a location. Where the user was given no directions previously on where the location was, but only was told they were at a saved location if they arrived, this new device would be able to direct the user to a chosen GPS location. This would also be useful in situations where the user would be lost and need to navigate back to a known location.

3.1.1.3 Object location

Finally, this last use case is tied to a recurring problem outside of the navigational scope of this project. Nonetheless, this use case will be discussed as it is an interesting idea which would still allow for the development of a haptic sensor device.

Understandably, it is difficult for people with a visual impairment to find objects in their vicinity. This is the case in navigation, but also in more domestic areas. For example, accidentally dropping your keys, and subsequently having to pick them up is a relatively straightforward procedure for people with sight. However, for people without sight, this would be a problem as they would need to "scan" the ground with their appendages in order to feel and find the keys. The same applies for misplacing things. People with sight would be able to look around in order to look for the misplaced item. However, for people with a visual impairment, finding the object would be a tedious and time consuming task. For this use case, the device would specifically be aimed at tracking the position of objects and classifying them, subsequently outputting that information to the user.

3.1.2 Final use case

After deliberation with both the supervisors of this project and when considering the use case with the most potential for future development, the first use case of "Orientational context navigation" has been chosen.

While the second use case seems very interesting and relevant to the scope of the project, it has eventually been decided that this project did not have enough potential for furthering the current state of the art in navigational aids. The intended device for this use case, while effective, is deemed to be too similar to current state of the art solutions. Furthermore, it did not present any clear ideas on how to improve this solution. In conclusion, the orientational context currently seems as the most useful and promising solution in filling out the gap in the current state of the art.

Lastly, the non-navigation use case of finding objects has been dismissed as it is deemed that the usage of a haptic device for such usage would be too unintuitive as informing users on precise locations in 3d using only haptics would be very complex and better reserved for different types of actuation, outside the scope of this project.

3.1.3 Use case scenarios

From the selected use case, a number of use scenarios are constructed. These use scenarios are constructed by analysing the interviews from the focus group interviews mentioned in Chapter 2 and looking at recurring themes for problematic situations.

As mentioned before, while looking through the interviews, it becomes clear that there is a recurring theme of not being able to find "orientationally significant objects" or "orientational objects". These are objects which can be used by visually impaired people to navigate their surroundings. Orientational objects can be used and identified in different ways by people of varying visual impairment. In interviewees with a partial visual impairment, orientational objects are mostly things such as stairs and elevation changes such as transitions from road to pavement, city infrastructure such as traffic lights, and doors in large buildings. In interviewees with a full visual impairment, orientational objects are often objects which do not have any direct navigational significance, such as trash bins or bumps in the road. However, these objects are still useful to them, as they use these objects to tell when and where to perform certain actions such as turning around or changing direction.

However, finding these objects can be difficult, and it is possible that the user may lose track of them. It was found that this especially was the case in situations where incomplete information was given, most commonly when Google Maps was used to navigate to a specific location, such as a shop. While Maps is capable of bringing you close to the entrance of the shop, there is often still a few metres between the user and the entrance. This recurring theme serves as the base of the design goal of the device. As such, the chosen use case has also been dubbed as the “Last Few Metres” use case within this project. The user scenarios as shown in the following sections have all come from this starting point.

The following scenarios will be used as base scenarios on which to test the functionality of the device.

- A user walks along a street on the pavement and needs to cross the road at an unmarked traffic light pedestrian crossing.
- A user is at the entrance of a train station and needs to climb several stairs in order to reach the train platform.
- A user walks along a road which is adjacent to a steep ditch.
- A user navigates through a shopping centre, with benches and plants placed throughout the path and multiple entrances to different shops.
- A user walks in a park on a curved path.

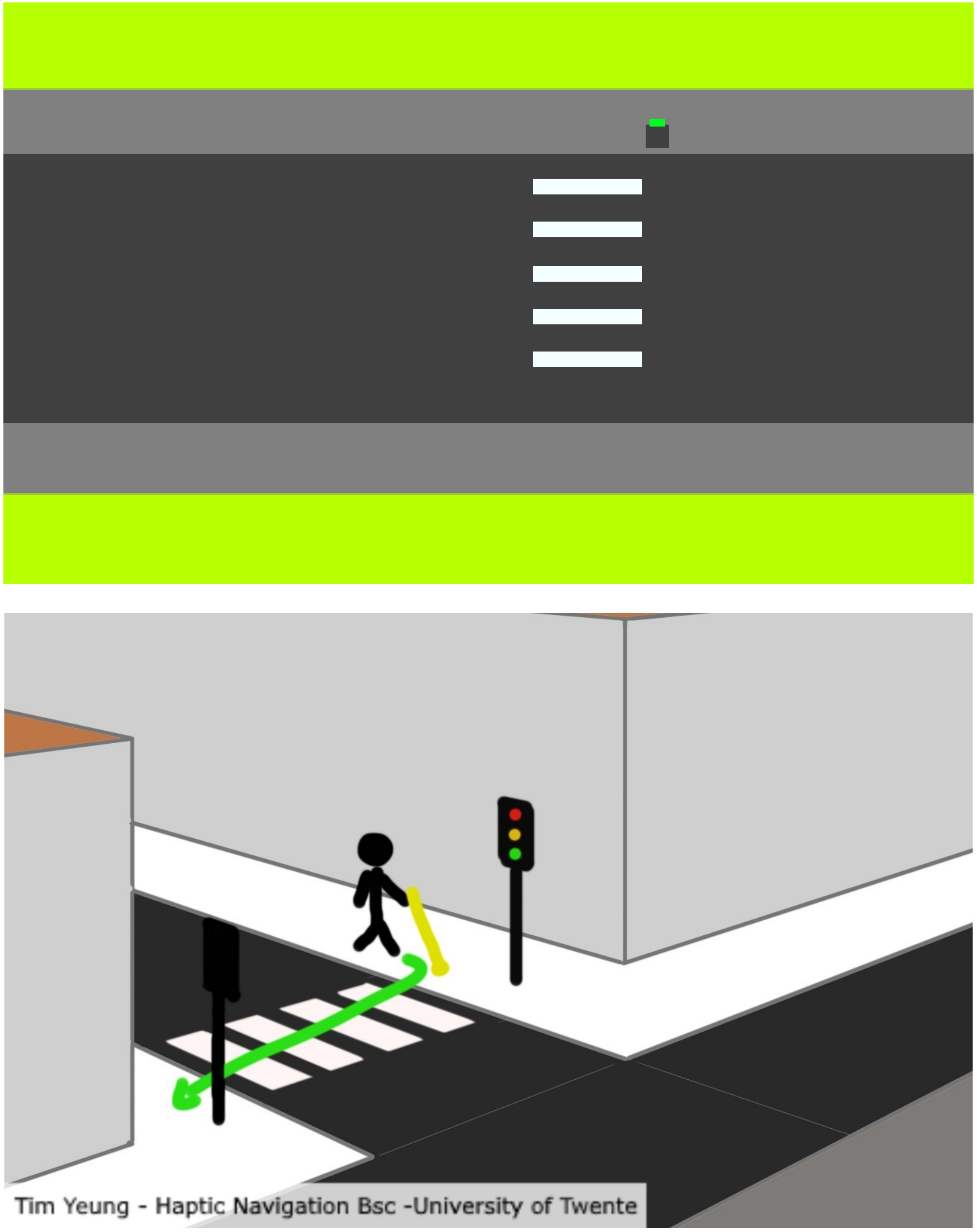


Fig. 3.1.3a: A user walks along a street on the pavement and needs to cross the road at an unmarked traffic light pedestrian crossing.

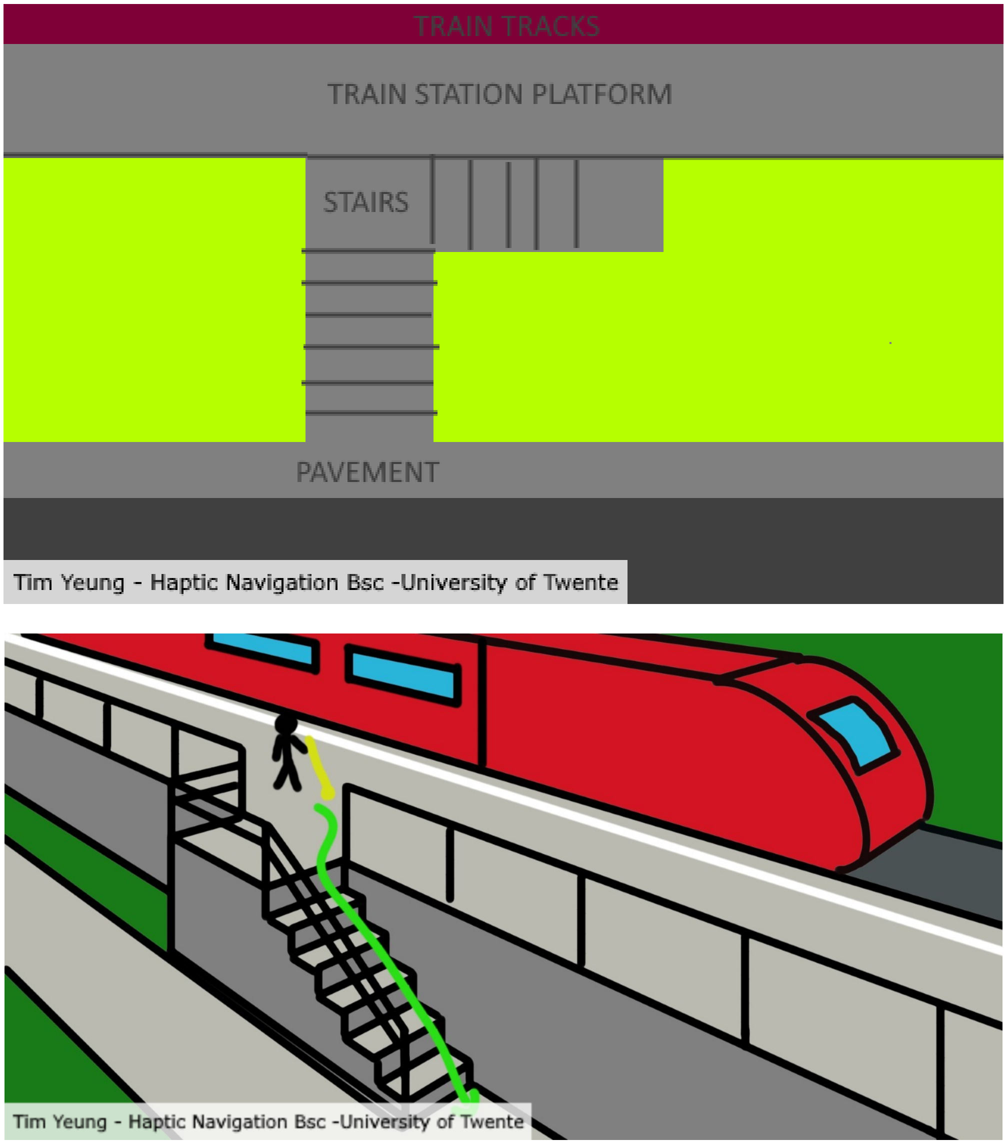
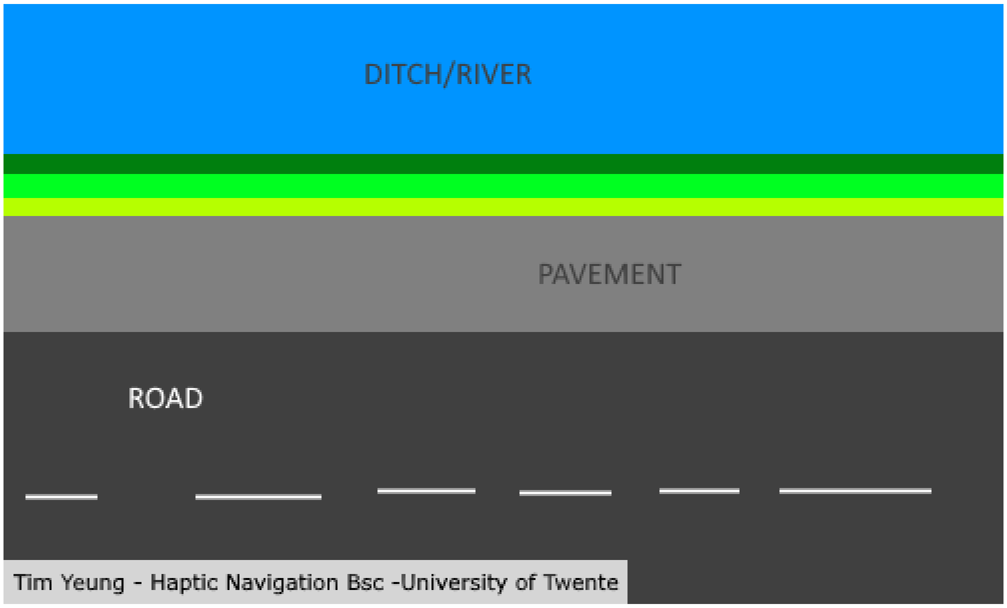


Fig. 3.1.3b: A user is at the entrance of a train station and needs to climb several stairs in order to reach the train platform.



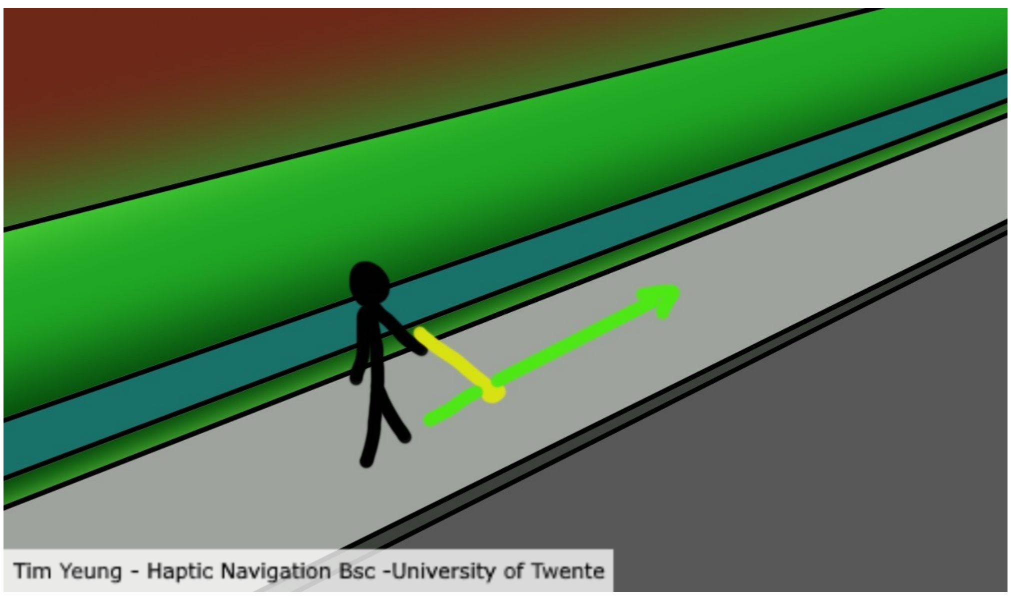


Fig. 3.1.3c: A user walks along a road which is adjacent to a steep ditch.

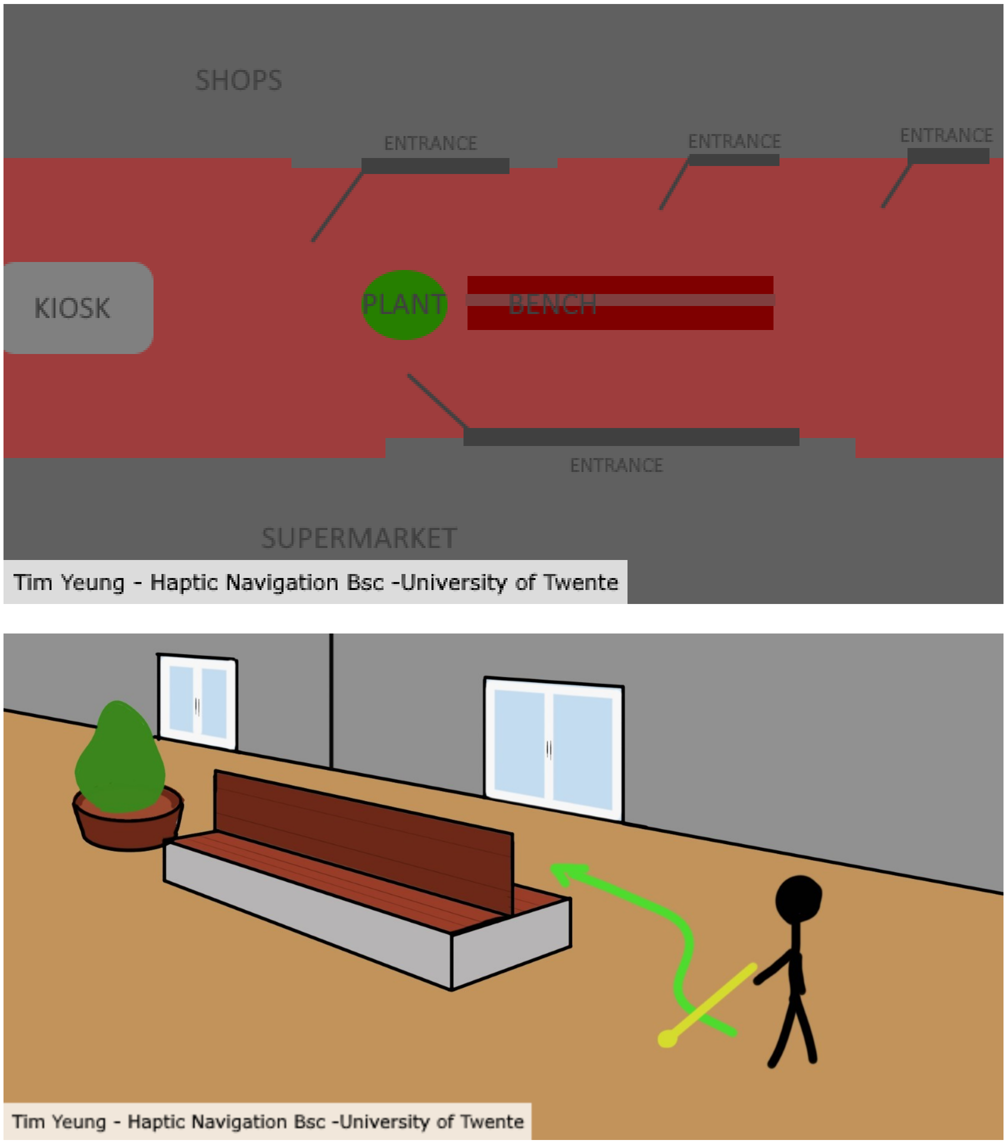


Fig. 3.1.3d: A user navigates through a shopping centre, with benches and plants placed throughout the path and multiple entrances to different shops.

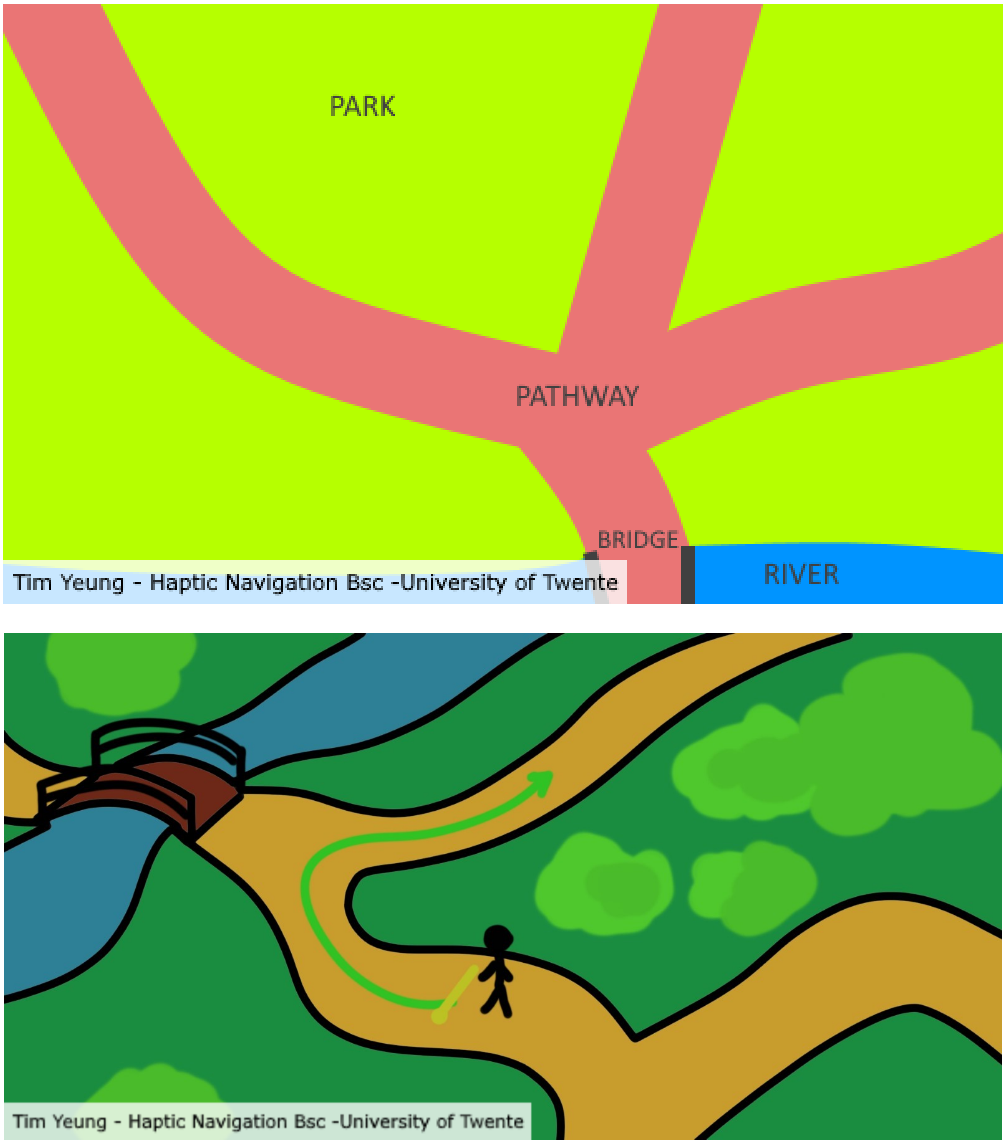


Fig. 3.1.3e: A user walks in a park on a curved path.

Chapter 4: Specification

4.1 Sensor system Requirements

Due to the scope of the project, only three of the user case scenarios described above are taken into considerations for the specification. These scenarios are scenario 1, 2, and 4. From the use case and these user scenarios, it becomes clear that in order to better let the user understand their surroundings, they require knowledge of two things. First, they require knowledge of where they cannot go. Therefore, the system needs to be able to identify and locate obstacles in a 3d environment. Second, users should be able to identify and locate orientationally significant objects. Therefore, the system should be able to identify and classify specific objects which can hold orientational significance. The final requirements of all scenarios are described below.

- Sensors should be able to detect obstacles and orientationally significant objects which include:
    - Pedestrian traffic lights
    - Roads
    - Stairs going up
    - Stairs going down
    - Doors
    - Normal obstacles
    - Hanging obstacles at head height.
- Sensors should be able to determine the location and distance to the detected objects.

In order to effectively communicate this information to the actuation subsystem, an interface and corresponding protocol is designed by all members of the entire project team. This interface dictates that data on obstacles is stored using a two-dimensional grid, which represents the area in front of the user. When projected onto the floor, the location of these grids is akin to a cone, with divisions made along the circumference of the cone. A circular part at the origin of the cone up till a certain radius is ignored as this area is covered by the white cane, rendering the sensing of obstacles there redundant.

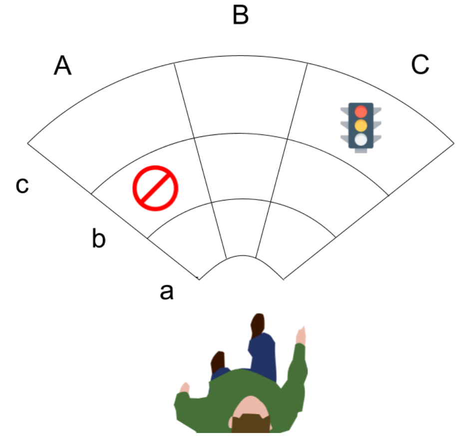


Fig 4.1a: visual representation of the two dimensional grid and the location and size of its cells relative to the user.

Each cell inside the grid stores information on any objects that intersect with the cell, or the absence of objects.. This information includes:

- The type of object, which includes
  - Nothing
  - An obstacle
  - An orientationally significant object
- The classification of an object, if it is orientationally significant, which includes
  - Pedestrian traffic lights
  - Roads
  - Stairs going up
  - Stairs going down
  - Doors
  - Normal obstacles
  - Hanging obstacles at head height.

While initially it was thought that normal obstacles and hanging obstacles would be useful to distinguish, tests showed that showing anything but orientationally significant objects would act as noise. Users would typically get confused by the many obstacles in their surroundings, while there was actually no need to distinguish between those non-significant obstacles. Therefore, the last two classifications of objects, “Normal obstacles” and “Hanging obstacles at head height” were subsequently removed.

4.2 Requirements of VR tool

From these requirements, the project concerning the sensor system designed a system which included an Intel RealSense D435, which is a normal colour camera combined with an infrared sensor, allowing for the capture of depth data. Furthermore this camera has a vertical field of view of 57 degrees for colour images and 42.5 degrees for depth images, both at an aspect ratio of 16:9. Finally, the initial design is for the sensors to be worn on the chest. Therefore, in order to faithfully recreate the capabilities of the sensors, the following requirements were identified for the VR tool in addition to the requirements stated above.

- The VR tool is able to provide data in the same format as the sensor system.
- The VR tool is able to emulate a virtual environment on the same scale as a real life environment
- The VR tool is able to track movement in real life in order to move the position of the sensors in the virtual environment.
- The VR tool is able to simulate vertical movement in the virtual environment (for example when the user is walking at a location which should contain stairs in the virtual environment)
- The VR tool is able to simulate the location, rotation and field of view of the sensors in VR to mimic the limitations of the real sensors.

In iterating the design of the actuation part of the device, an extra functionality was requested for the VR tool, which was the inclusion of a “pointer sensor”, which would be able to tell where the user was pointing, using the controllers of the VR headset. Therefore, a few additional requirements were added. The iteration process is further described in Chapter 6.

- The VR tool is able to track what the controller of the headset is pointing at.
- The VR tool is able to send information on what the controller is pointing at through the existing interface.

4.2 Testing requirements

For the testing of the device, two distinct domains should be considered. First of all, the creation of new environments should be facilitated. Second, the automated collection and evaluation of data should be facilitated. Both of these domains should be in the scope of the project as both functionalities further justify the usage of digital tools for evaluation. Both of these domains can be split up into separate tools which are discussed below.

4.2.1 Requirements of Environment Creation Tool

As multiple scenarios have been identified which will serve as the starting point for the device, these scenarios should also be used when testing and evaluating the device. Therefore, multiple virtual environments need to be created. Furthermore, in order to maintain flexibility and better accommodate the iterative process of design, it makes sense for anyone to be able to easily and quickly create new virtual environments which can be used for testing. This also allows for better reproducibility, as the virtual environments can be stored digitally and be used for future usage. This tool should be separate from the VR tool so that people do not need VR headsets (which can be scarce) to create test scenarios. Users should also be able to easily transfer the environment files to the VR headset and the VR tool should be able to read these files without needing to alter any of the program itself. The above section gives the following requirements.

- The creation tool is able to create files which can be edited again by the tool and which can be read by the VR tool in order to generate a testing environment.
- The creation tool is able to read files created by itself in order to load in previously created environments.
- The creation tool is able to set and save the scale of the scene.
- The creation tool is able to set and save objects in the scene with custom shapes.
- The creation tool is able to set and save roads and walls in the scene by drawing lines.
- The creation tool is able to set and save small objects with predetermined shapes.
- The creation tool is able to set the scale of objects with predetermined shapes.
- The creation tool is able to set the 3D position and height of objects with predetermined shapes, roads and walls.

Furthermore, in order to integrate the functionality of the VR headset with the environment tool, the following additional requirements are identified for the VR tool.

- The VR tool is able to load files created by the creation tool, without alteration of the VR tool itself.
- The VR tool is able to create environments, from said files which contain objects and obstacles which can be sensed with the VR tool.

4.2.2 Requirements of the Evaluation Tool

As the VR tool allows for the automated collection of digital physical data, a strong case can be made for the inclusion of a specific tool for evaluating this data within the scope of this project. In order to maximize the flexibility of analysing later data, it makes sense to store the location and rotation of the headset and the controller, rather than any specific measurements. By storing this data in combination with which environment is being tested, it is possible to recreate the scene virtually after the tests. This decreases the processing load of the VR tool, as it does not require live processing of data and also allows for new measurements which can be applied after having tested the device using the VR tool. Subsequently, the measurements defined in Chapter 2 for the evaluation of the device should be implemented in a tool that can read out the stored locations and rotations and subsequently derive the measurements. It is also important that researchers are able to replay the experiments exactly in order to get better insights into the test results and the user's behaviour. All of this gives the following requirements.

- The evaluation tool should be able to read out positions and rotations from a file.
- The evaluation tool should be able to read out the environment which is being tested.
- The evaluation tool should be able to recreate and replay the test as it played out by showing the headset and controller inside of the tested environment in 3D.
- The evaluation tool should be able to automatically derive the duration of the test.
- The evaluation tool should be able to automatically derive the amount of incidents that occurred during the test.

Subsequently, in order to integrate this with the VR tool, the following requirements are added to the VR tool.

- The VR tool should be able to start and stop recordings.
- The VR tool should be able to create recordings of the position and rotation of the headset and controller.
- The VR tool should be able to save these recordings in a file which can be read by the evaluation tool.

Chapter 5: Implementation

The implementation of all tools developed will be discussed in the following sections. A total of 3 separate tools will be discussed as indicated in the previous chapter, with two of those tools corresponding to the implementation of the VR tool, one corresponding to the implementation of the Environment tool, and one corresponding to the implementation of the Evaluation tool..

5.1 VR Tool

The VR tool consists of two programs, the first one running on the Oculus Quest 2 and the second running on the pc or device which is connected to the actuation's ESP32. In the following sections, a brief overview will be given of the implementation of both programs.

5.1.1 Unity3D on the Oculus Quest 2

For the program running on the Oculus Quest 2, the unity game engine is used to implement the simulation of the sensors. Within this scope, the program deals with identifying and storing objects, assigning objects to the corresponding grid, loading in objects and environments, and safety measures.

5.1.1.1 VR in Unity

Unity has been chosen to implement the simulation of the sensors as it is a capable 3D engine which has excellent support for VR implementations. As the VR headset used for this project is the Oculus Quest 2, it is possible to use the Oculus Integration SDK, which is available for Unity for free on the Oculus Developer website[21]. This package allows for easy implementation of VR spaces, and already implements some of the more rudimentary necessities for VR tracking, such as creating the virtual cameras and making sure that all the movements in real life are tracked to the virtual environment.

In order to make sense of some of the implementation details for the following sections, a brief overview of the way Unity handles coordinates is given. Unity uses 3-dimensional vectors to express coordinates in the virtual world. All objects inside of Unity will have a position. Furthermore all objects have a "parent object", which can be either null or another object. Child objects will automatically move with their parent object, if a parent object is moved. Any object that exists in the environment will have a "Global Position", which is the position relative to the world origin, and a "Local Position", which is the position relative to the parent object of the object. If the parent object is null, the local position will be equal to the global position. Lastly, in order to implement functionalities in Unity, one generally needs to write a script deriving from the Monobehaviour implementation and then "attach" an instance of this script to an object in the scene. Effectively, this means that almost all functionalities in Unity can be linked to an object, and subsequently a position.

The first step to being able to map the real world to the virtual environment, is by matching the "floors" of both environments. Oculus will already track the floor of the real world environment, so this process can be done quite easily. Using the Oculus SDK, we can set it so that all of the positions of the VR peripherals are calculated from the floor's origin. The Oculus SDK is implemented through a Monobehaviour script attached to an object. So, by making sure that the coordinates of the object related to this script are on the same height as the virtual floor, it is possible to synchronize the floors of the real and virtual spaces.

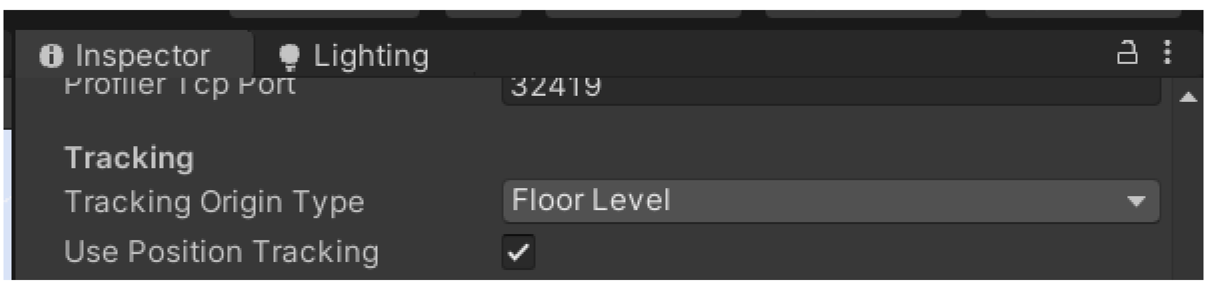


Fig 5.1.1.1a: Tracking Origin Type is set to "Floor Level" in order to ensure that the real life floor aligns with the virtual floor.

Secondly, we need to be able to align the boundaries of the virtual environments to the boundaries of the real life test space. By making it so that all of the objects in the environment are children of a single object, it is possible to easily move, scale, and rotate all of the objects in a scene at once by only manipulating the parent object. From there, we only need to determine how the parent object should be changed in order to align with the real life environment. This is done by setting the origin of the real life environment in the virtual environment and then setting the first connected corner of the real life environment going clockwise. In practice this is done by having a script attached to the right controller of the Quest 2 and detecting presses to the Primary Button (A). In practice, the researcher will walk over to a set corner of the testing space and put their controller in the middle of this corner. They then press the A button to set the origin. The process is repeated after putting the controller in the middle of the first clockwise corner. A third press confirms the position of both corners and will instruct Unity to position the parent object of all the objects in the environment at the origin point, and then rotate it by the angle of the vector going from the origin to the first clockwise corner. As the floor is already set by the Oculus SDK, the parent object is only rotated in the Y-axis.

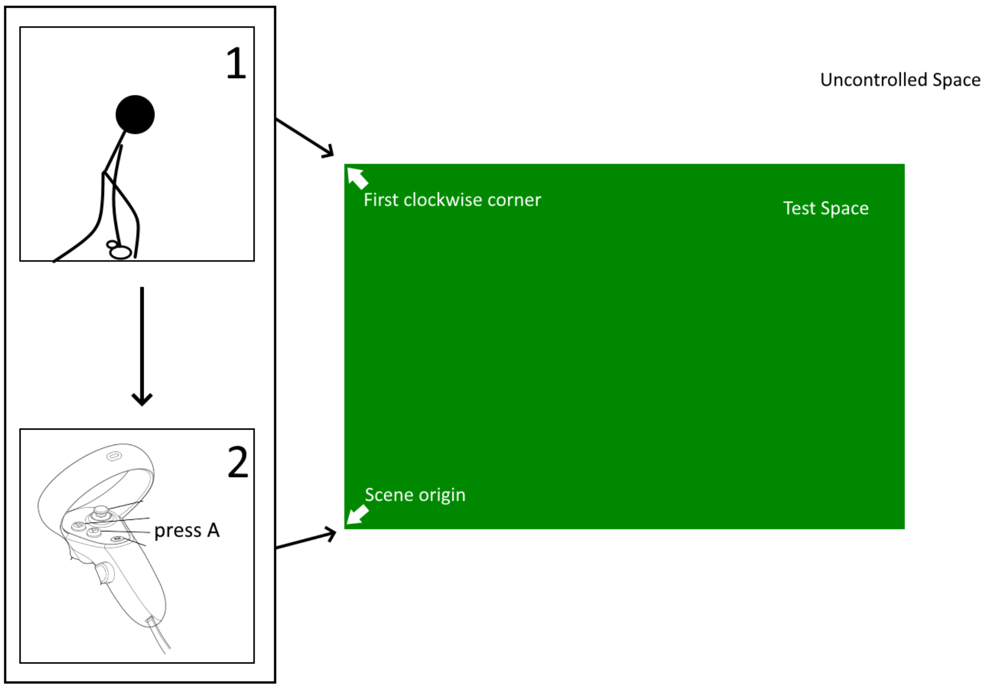


Fig 5.1.1.1b: Schematic representation of the corners used for calibrating the environment. User puts the controller right on the center of each corner and presses the A button of the right controller.

5.1.1.2 Simulation of objects in the scene

In order to simulate objects, Unity objects are simply used. This way, the position, rotation, and scale of box-shaped objects are already implemented. Each object has its own collider, which determines the bounds of the object. For box-shaped objects, this collider simply has the same size as the scale of the Unity object. For other objects, this collider is determined by the mesh that is attached to the Unity object. In order to store what kind of objects are, each object has an “Environment Object” script attached to it which stores what the shape type of the object is (either complex or box-shaped) and what the object type is (i.e. Traffic lights, Doors, etc.). This script also automatically adds an icon in the middle of the object's bounds to indicate the object type.

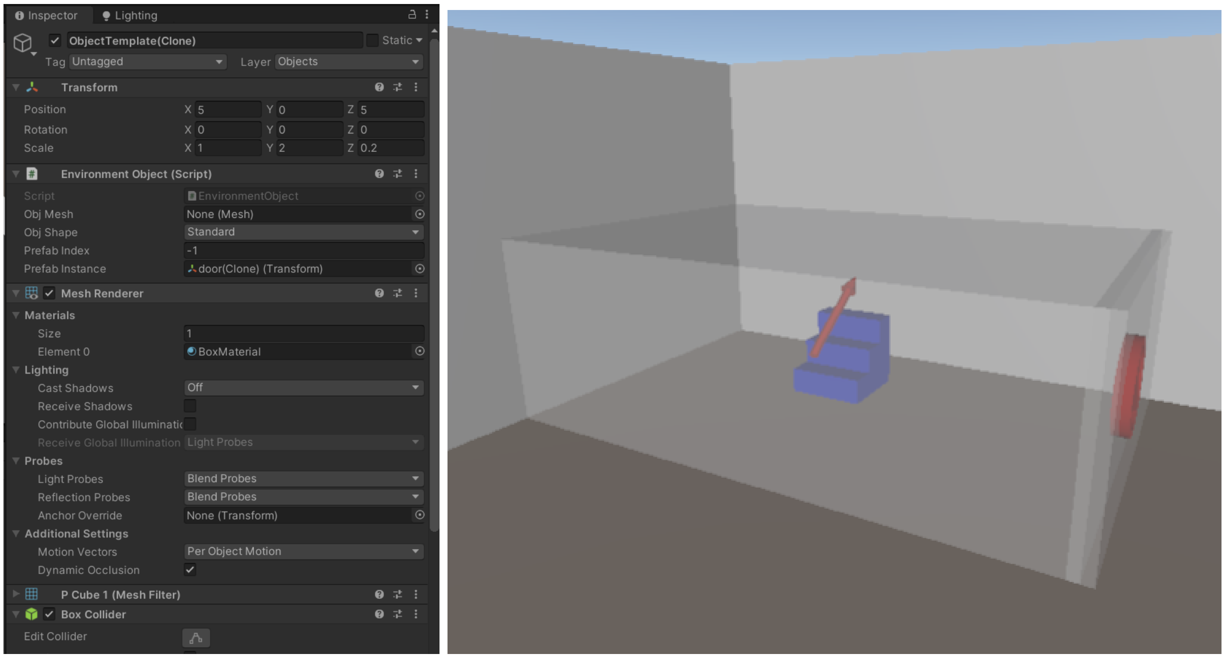


Fig 5.1.1.2: Inspector view of an Environment Object object on the left and an example of a staircase and railing expressed using box-shaped objects on the right.

5.1.1.3 Simulation of sensors in the scene

In order to simulate the sensors and fill a 2-Dimensional grid with objects a relatively simple approach is taken. Unity allows for something called an "OverlapBox" which is part of its physics implementation. This method checks for any objects with colliders within a box given by the implementer.

Initially, a similar approach to the real life sensor system was taken. This approach would use a big OverlapBox to detect all objects within the range of the grid and subsequently calculate the angle and the distance between the object and the user's current position. From there the object could be assigned to the appropriate cell. However, it was found that larger objects would often span multiple cells. This, combined with the initial implementation meant that objects would only be assigned to one cell based on their center point, while they could span multiple objects. This would cause "accidents" since the user would walk towards a cell that was considered to be clear, while in actuality the object would still span that cell. Therefore, it was chosen to change the implementation to the one described below.

Instead of assigning objects based on the position of their origin, it is possible to create an array of positions, rotations, and sizes of boxes which can represent the grid which is to be filled, and subsequently call Physics.OverlapBox for each cell to determine whether there is an object inside that cell. Furthermore, the necessary details of the box such as object type are loaded into the grid at the same time. General specifications of the sensors such as the angle of detection, the detection range, and the size of the grid, can be set through the Unity UI.

In order to accommodate the programming of the actuation, it was agreed that the amount of objects that are assigned to one cell can be limited. This parameter can be set in the UI as well.

There is also an implementation for storing the object which the user is pointing at, which was added after iteration (see Chapter 4.2). This is done by simply casting a rectangular ray from the front of the controller and looking at whether it collides with any environment objects. The object which acts as the controller is assigned through the UI.

Finally, it was found during tests that participants would often become confused if they were standing inside of an object as the system would just ignore that object. This was a problem unique to the VR tests as preliminary tests with the actual sensor system were done with real boxes. Therefore, if testers would collide with a box, they would already understand that they had hit an obstacle. Since there is no physical object in the real world for the VR tests, testers would have no feedback on whether they had just hit an obstacle. To solve this issue, an implementation has been added which checks whether the user of the headset is currently standing inside of any objects. This works by using the Physics.OverlapCapsule method to check for objects within a certain radius of the headset, as it is assumed that generally, the majority of the user's body is contained below their head. For the radius of this check, an average adult male's radius has been estimated at around 36 centimeters. The radius of the check can be set in UI in meters. If a user has been determined to be inside of an object, the actuation sends a special signal to indicate that they have hit an object.

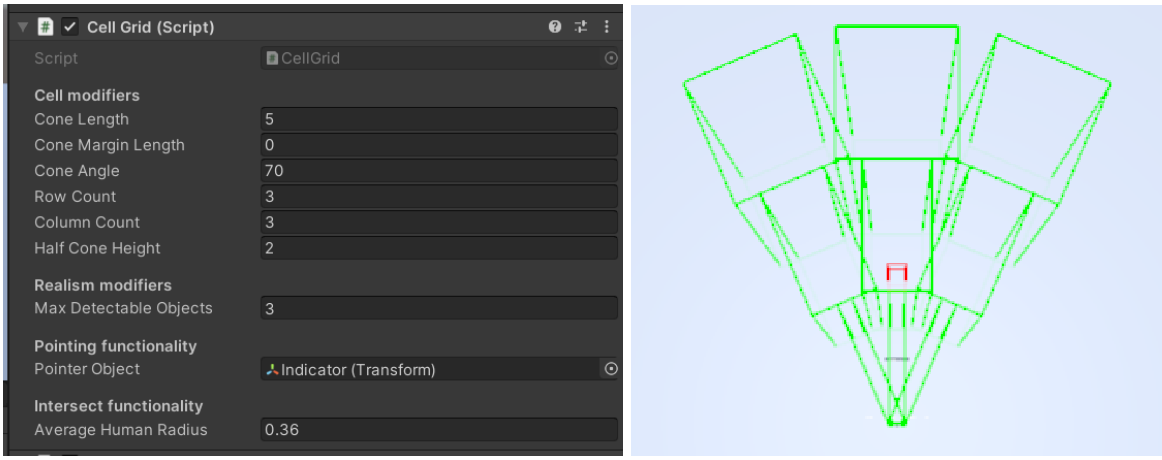


Fig 5.1.1.3: a view of the Cell Grid script which implements the sensors and an aerial view of the boxes which are used to detect objects.

5.1.1.4 Loading and generating environments from files (further discussed in 5.2)

The VR Tool implements a “summonable” menu which can be called by pressing the joystick of the right controller. This menu contains buttons, with each button representing a scene that can be loaded from the local data folder. Users can then use their controllers to “press” buttons, which will load the corresponding scene. Further explanation on how environments are stored and loaded will be given in 5.2.

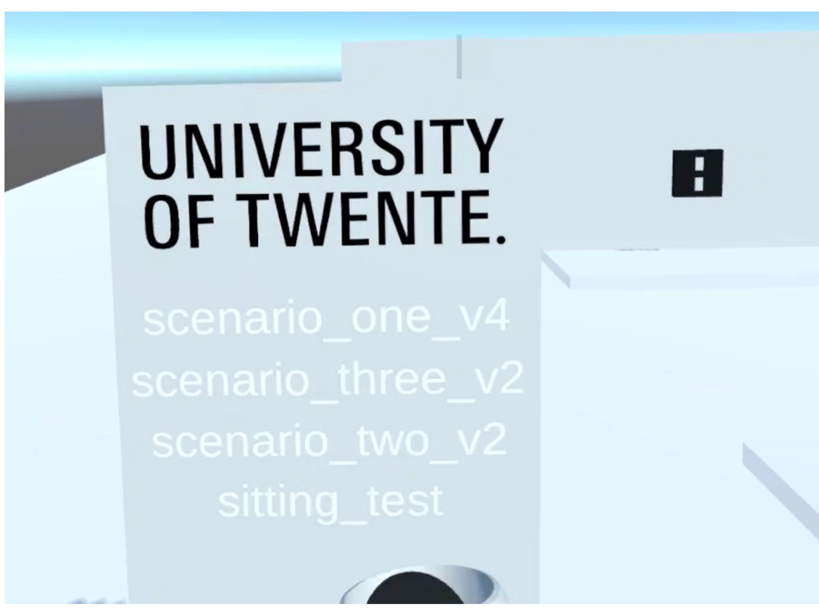


Fig 5.1.1.4: Main menu with buttons for loading scenes in the VR tool

5.1.1.5 Safety Measures

To ensure the safety of people using the VR tool, a virtual blindfold was implemented. The Oculus Quest 2 uses the Oculus Guardian system, which allows users to set the floor and the boundaries of their usable space. If any of the peripherals come too close to the edge of the spaces, a rastered wall representing the edge will appear, warning users that they are close to exiting the safe space. For testing purposes, researchers can set these boundaries to the edges of the testing space.

Since the device is only tested with non visually impaired people, it is possible to only disable the rendering of all objects, but still allow the user to see the Oculus Guardian boundary. This way users can still see the edges, warning them from exiting the safe space.

5.1.1.6 Starting and stopping a test run and recording positions and rotations

When a user starts a test run, the blindfold is activated using the Secondary Button (B) of the right controller. This starts the recording of the positions and rotations of the headset and the controller used for pointing. This data is then stored inside of a single file, at an interval of 25 times per second, and can be copied individually from the headset back to a local computer. At the end of the test run the user presses the B button again to disable the blindfold and stop the recording. More details on this procedure can be found in Chapter 5.3.2.

5.1.2 C# Console application for the device computer and communication

In order to communicate with the ESP32 used for the actuation, serial communication is used. However, as the headset only has a single micro USB port and the ESP32 needs a 5V charge to function, it is more practical to have the ESP32 communicate with a laptop which can be put in a backpack to be carried by the user. This means that the headset needs to communicate with the laptop through some other means in order to communicate with the actuation. For this purpose, a small C# console application has been programmed which communicates with the ESP32 through serial communication and with the Quest 2 through a rudimentary TCP client server connection. In short, the VR tool acts as a TCP server, with the IP address being displayed on the left controller. The laptop can then fill in the IP and port arguments when running the program, and attempt a connection with both the VR tool and with the ESP32. Once a connection has been established, the console application will await requests from the ESP32 over serial communication and relay these to the VR tool through TCP. The VR tool then sends the corresponding data back to the console application, which in turn relays the exact same message to the ESP32.

5.1.2.1 Communication protocols

As determined through discussion with the actuation project, the information on the cells in the grid in the final version are communicated through single strings which represent one of the columns in the grid. Columns of the grid are enumerated from left to right in capital letters. I.e the leftmost column is represented by an “A”, with the next column directly to the left being represented with “B”, then “C”, etc. Each row in the grid is represented by small letters, starting from the row closest to the user, moving outwards. Also see Fig. 4.1a for a diagram of the cells in the grid and their names. The actuation has been programmed to request columns one by one from the sensors by sending the capital letter representing which column should be sent. The sensors then send back all of the cells in the column from closest to the user to furthest. Each representation of a cell starts with the letter representing the row of the cell, followed by a number denoting whether the cell is empty “0”, filled with an obstacle “1”, or whether the user is standing inside of an object “2”. If there are any objects which are considered to be orientationally significant, the types of the objects will be appended behind the number representing the type as a 3 digit number. In later versions of the program, the object type is only appended if the user points at the object with their controller. Orientationally significant objects and their enumerations are the following.

- Pedestrian traffic light 001
- Road 002
- Stairs up 003
- Stairs down 004
- Doors 005

An example of the communication is given below for different scenarios with a grid size of 3 columns by 3 rows.

Actuation requests “A” column, obstacle in middle cell and traffic light in furthest cell
Actuation: “A”
Sensors: “a0b1c1001”

Actuation requests "B" column, traffic light and door in middle cell
Actuation: "B"
Sensors: "a0b1001005c0"

Actuation requests "C" column, user is clipping through an object
Actuation: "C"
Sensors: "a2b0c0"

Alternate versions were also programmed for several specific tests requested by the actuation project, such as a version where the entire grid is sent at once, and a version where instead of columns, rows are requested. They in essence work with the same protocol as the standard version with only slight modifications which are discussed in Chapter 6.

5.2 Environment Creation Tool

The environment creation tool is a program made with Unity 3D for Windows systems. It is used to create, load, and edit test environments which can be read by the VR Tool to generate test environments.

Fig 5.2: the main menu of the environment tool.

5.2.1 Creating new environments

Initially the tool was intended to utilize 3D map APIs such as the Google Maps SDK or the WRLD SDK which are compatible with Unity, in order to generate 3D environments automatically. However, initial tests showed that unfortunately, most of the time the APIs were either too limited in functionality, which hindered implementation of adding additional environment objects, or were inaccessible behind a paywall. A workaround solution was devised to still allow for using real life locations as references for creating scenes, using Google Maps to serve as a reference image. Users would be able to pan and zoom the image, and then save and rotate it to match the specifications of the environment. However, due to time constraints and the fact that most available test spaces were not big enough to support environments the size of a street, this approach was abandoned in favour of the focus on the environment creation process itself.

Fig 5.2.1a: original workflow concept for creating environments using Google Maps from left to right top to bottom: "Choose an environment and right click to choose measure distance", "Left click to draw a line representing the horizontal wall of the environment", "Rotate the image to align the horizontal wall with the screen", "Finally, draw a rectangle by dragging the right mouse button to set the test area and set the scale from the measured distance"

When a new environment is created, the program will request the intended size of the test environment. Once this has been filled in, it will generate a green plane representing the new environment. From there, the user is able to add new objects. The user is also able to move and rotate the camera to get a better view of the environment by dragging with the left mouse button and the left alt key.

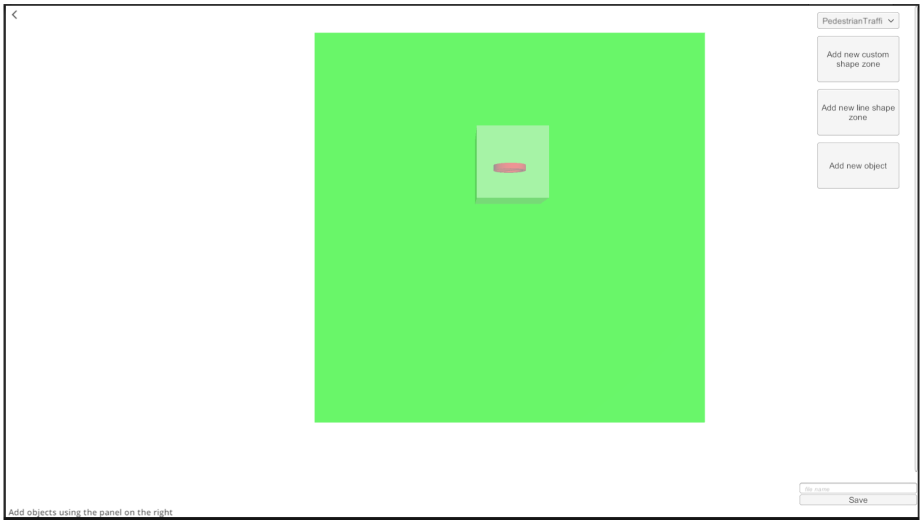


Fig 5.2.2: snapshot of the environment creator after having created a 6 by 6 meter environment and adding an obstacle.

5.2.3 Adding and editing objects

There are 3 different types of object shapes which can be used to define the bounds of an object.

- Custom objects are objects which have a custom, rectangular shape. While the edges of the rectangle can be set manually, it is currently not possible to adjust the height or position of the custom objects.
- Line objects are objects which consist of one or multiple lines of which the heights and widths can be adjusted.
- Box objects are the standard object type which consist of a box object which can be moved, scaled and rotated. Each new object's initial object type can be set using the dropdown menu at the top of the UI.

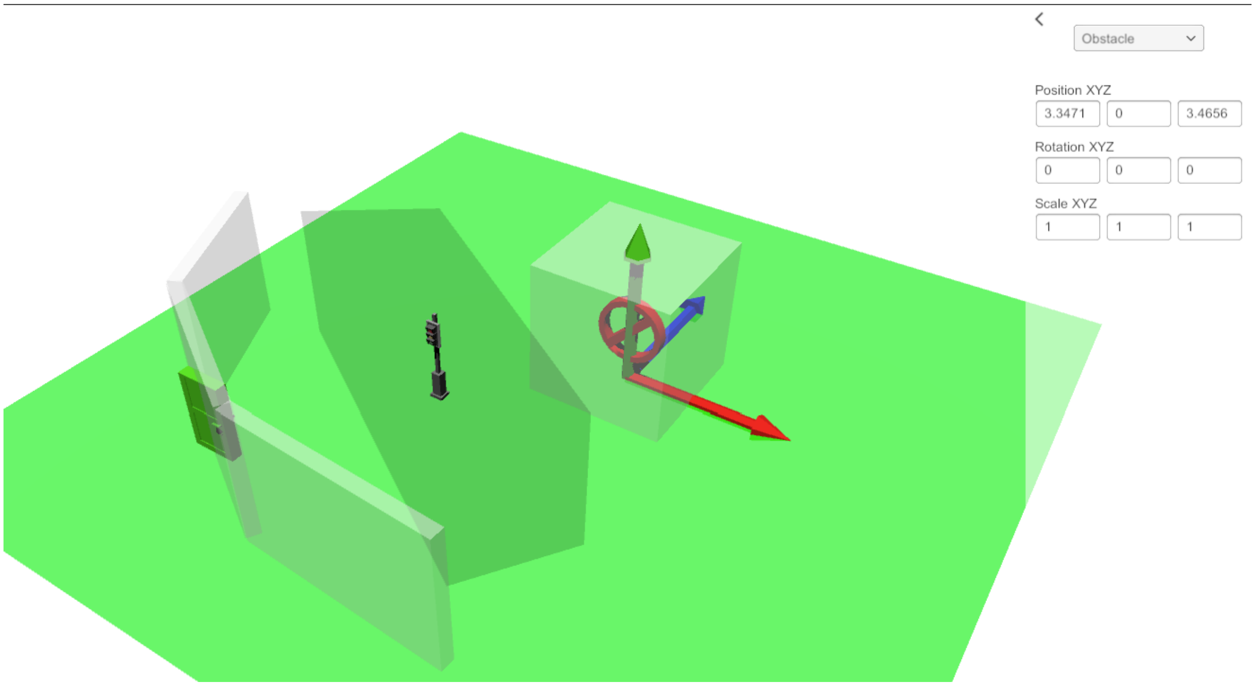


Fig 5.2.3a: from left to right, a line shaped door object, a custom shaped traffic light object, and a box shaped obstacle object.

Objects can be added through the UI buttons on the right side of the screen (see Fig 5.2.2) by clicking the button and following the instructions in the bottom left of the screen. Mainly, the right mouse button is used to select the position of the object or corners. Once an object has been placed in the scene it can be selected by pressing the right mouse button when hovering over said object. From there, a new UI appears which shows the position, rotation, scale, and type of the object. These UI elements are also interactable, allowing the user to set those parameters through the UI manually. Arrows will also appear over the selected object, allowing the user to drag the object.

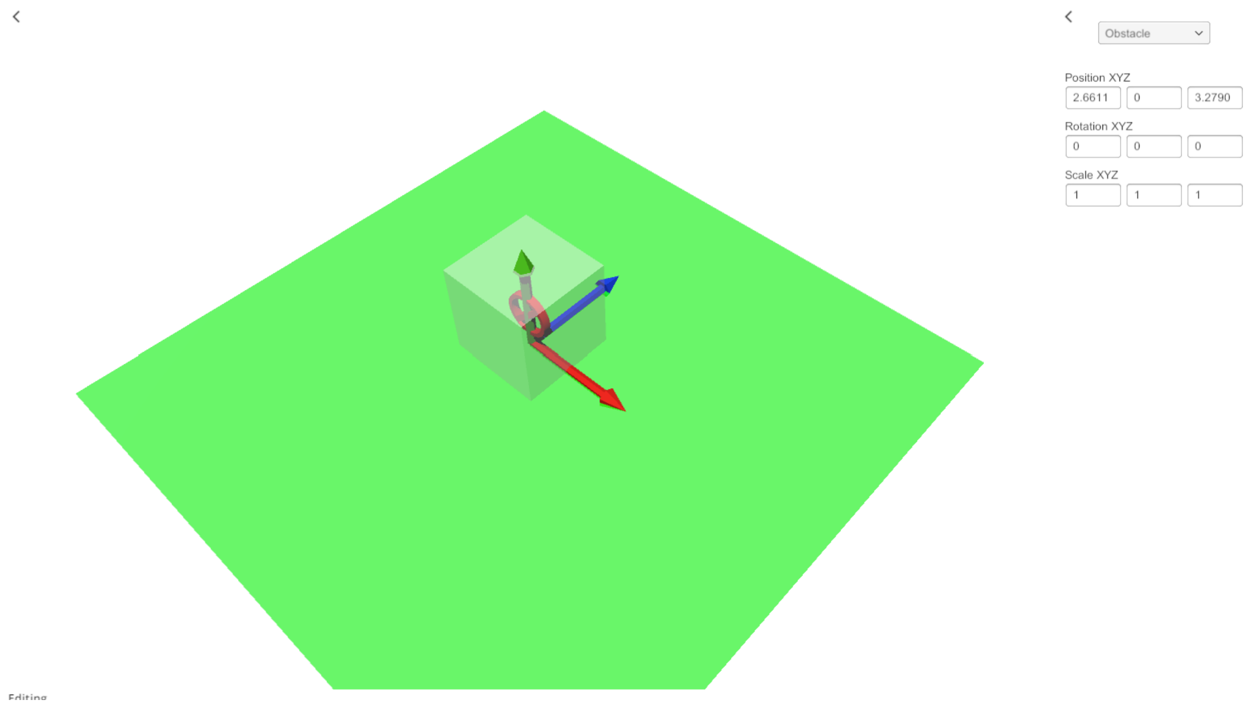


Fig 5.2.3b: snapshot of the program after having selected an obstacle object.

5.2.3 Storing and loading environments to files

In order to store environments with their objects, a rudimentary file writer has been implemented. Each environment is saved in a single file which contains the following information.

- Width of the test environment
- Length of the test environment
- The scale of the test environment
- All of the objects in the test environment

The width, length and scale are stored using converted float values. Each object that is stored contains the following information.

- The object type (i.e. Traffic Light, Door, etc.)
- The shape type (i.e. Custom, Line, or Box shaped)
- The coordinates of the

    - corners for custom shapes
    - line points for line shapes
    - origin for box shapes
- The
    - height for custom and line shapes
    - width for line shapes
    - scale for box shapes
- The rotation for box shapes

a b c d

6?6?1?2|2|(3.0, 0.0, 2.0)/(0.0, 0.0, 0.0)/(3.5, 0.1, 4.0)|-1?2|2|(3.0, 0.0, 5.5)/(0.0, 0.0, 0.0)/(3.5, 0.1, 1.0)|-1?1|2|(5.0, 1.0, 4.5)/(0.0, 0.0, 0.0)/(0.2, 2.0, 0.2)|-1?1|2|(1.0, 1.0, 4.5)/(0.0, 0.0, 0.0)/(0.2, 2.0, 0.2)|-1

a: width
b: length
c: scale
d: object type, object shape, position, rotation, scale, override prefab

Fig 5.2.3: example of file containing a digital version of the environment presented in use scenario 1 (see Fig 3.1.3a)

All data is stored on a single line, with different parameters being delimited with several different delimiters, depending on the type of parameter. It should be noted that while this implementation is fully functional, a better approach is to save the environment by using Unity's JSON Serialization implementation. However, the existence of this implementation was only found after the project ended.

Loading files works by reading out these files and parsing the test environment and environment objects from these files and then spawning these into the Unity world. The test environment is confined with 4 walls in order to contain the users within the test environment. These walls have been marked as obstacles so that the actuation will also tell the users to stay clear from the walls. For each environment object stored an environment object is spawned using the structure described in Chapter 5.1.1.2.

5.3 Evaluation tool

Finally, the evaluation tool is a part of the program for the Environment Creation Tool. It is used to evaluate the data containing positions and rotations stored by the VR tool during test runs, and allow for the analysis of individual and multiple test runs and the export of analyses to CSV tables.

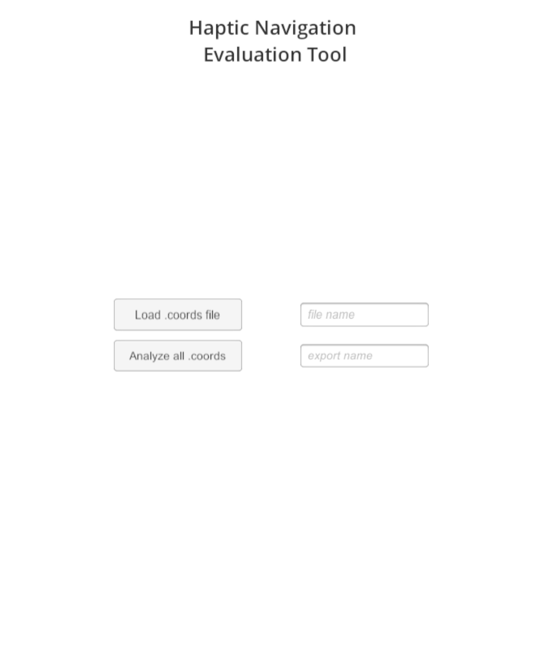


Fig 5.3: Main menu of the evaluation tool

5.3.1 Loading and storing position and rotations from files.

For storing positions and rotations, a simple implementation is used which stores the positions and rotations of the headset and the controller used for pointing. However, since the VR tool usually needs to be calibrated using the methods described in 5.1.1.1, positions and rotations are saved relative to the parent object which contains all the environment objects, i.e. the origin of the test area. If only the world position is saved, it is possible that the coordinates do not match up with the environment, as the environment is always spawned in the world origin in the environment and evaluation tools, while the position of the environment in the VR tool can be changed during calibration.

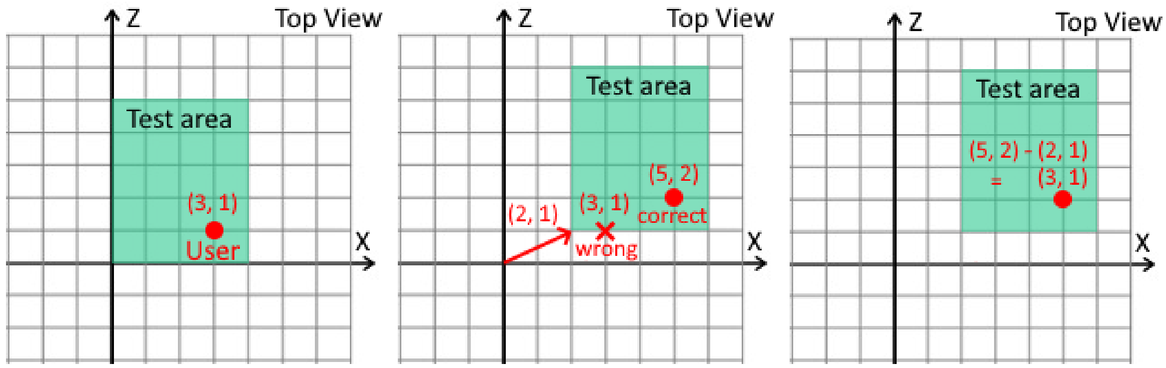


Fig 5.3.1a: Diagram showing the issue with saving global positions. In order to ensure that coordinates are correct regardless of the position of the Test area, the relative position to the test area's origin should be stored.

The positions and rotations are written to the file using the default Vector3.ToString() and Quaternion.euler.ToString() methods at an interval of 25 times per second. This set interval has been chosen as a compromise between the size of the files stored and the performance of the VR tool. Furthermore, it ensures that the timing of the positions and rotations remain consistent, as the interval is not timed to the framerate of the program, which can be susceptible to lag spikes. Each position and rotation is stored on a new line, with each new frame of data being appended on a new line. The following structure is maintained.

- Line 1: Headset position
- Line 2: Headset rotation
- Line 3: Controller position
- Line 4: Controller rotation

```
scenario_one_v4
(0.6, 1.7, 0.8)
(9.9, 183.7, 359.3)
(0.5, 1.3, 0.6)
(358.6, 170.5, 339.8)
(0.6, 1.7, 0.8)
(10.0, 183.5, 359.5)
(0.5, 1.3, 0.6)
(359.5, 169.7, 340.9)
(0.6, 1.7, 0.8)
```

Fig 5.3.1b: excerpt from the coordinates file of one of the test runs.

Additionally the environment that is being used for the test run is written at the start of the file. This way, it is possible to recreate the obstacles in the environment as well by simply loading in the corresponding environment and its objects.

5.3.2 Recreating test environments

Files that have been stored by the VR tool can be transferred to the PC using USB or other methods such as file sharing applications. These files can then be placed in the work folder for the evaluation tool, allowing the program to read the files. In order to recreate an environment, first the objects in the scene and the test area are loaded in by loading the environment file that corresponds to the filename at the start of the coordinates file. This requires the evaluation tool and the VR tool to both have the same environments in their work folder, as the evaluation tool is only able to load in environments which are also saved in the working directory and the coordinates file does not store any environment data. Afterwards, all of the positions and rotations of the coordinates file are loaded into arrays.

Once the entire file has been read, an UI will appear which allows the user to replay the test run as it played out by pressing the play button on the UI. Users also have the option of playing through the test run backwards, pausing the playback, skipping ahead or back, and moving and rotating the camera in the same way as with the environment tool..

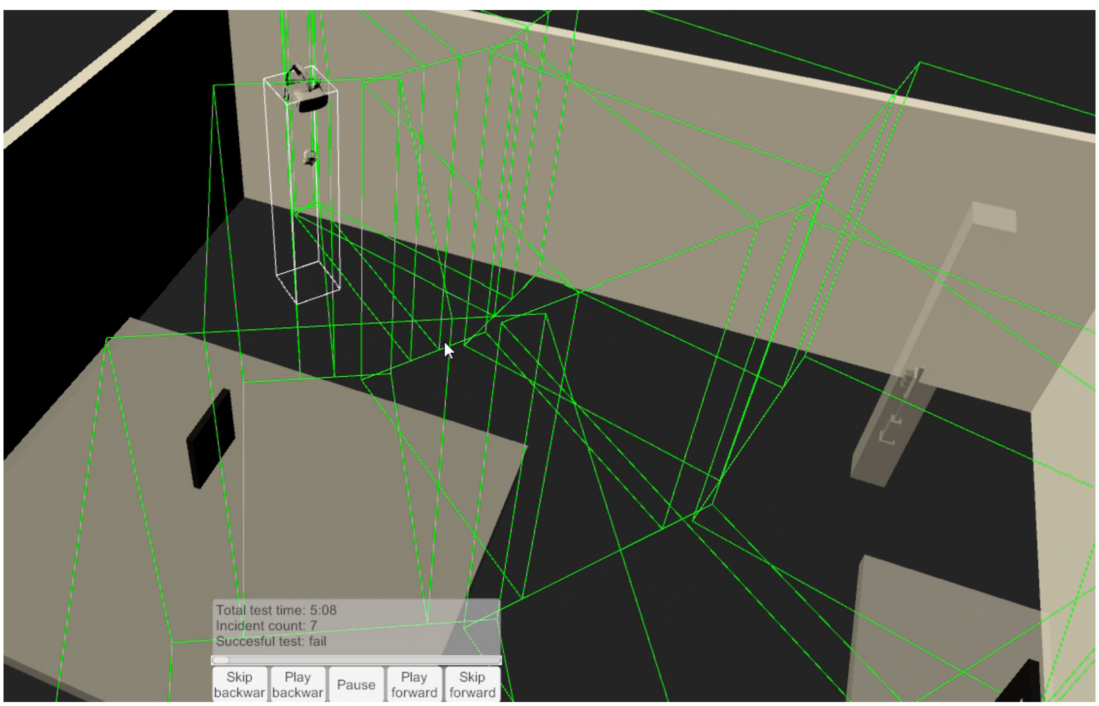


Fig 5.3.2: snapshot of the evaluation tool when analysing a single test run.

5.3.3 Measuring data from recreated scenes

When a coordinates file is being loaded into the program for replay, the program simultaneously analyses the test run with the measurement methods described in Chapter 4.2.2, namely the measurement of time in order to determine the mobility and effectiveness and the measurement of incidents in order to determine the safety and reliability.

Time measurement is done in quite a rudimentary manner. Since it is known that positions and rotations are stored at an interval of 25 fps, it is possible to calculate the length of the test run in seconds by multiplying the amount of positions stored in the file by 1 divided by 25.

Incident measurement is done by checking whether the headset collides with any object within a certain radius for each stored position. This radius can be set within the Unity UI and represents the average radius of a human body. When a collision occurs, the system counts the amount of frames that the sensor is still colliding with the object. If this count exceeds a certain threshold, which can be set through the UI, the system registers this as a single incident. The system will also create an indicator at the location of the collision in the scene.

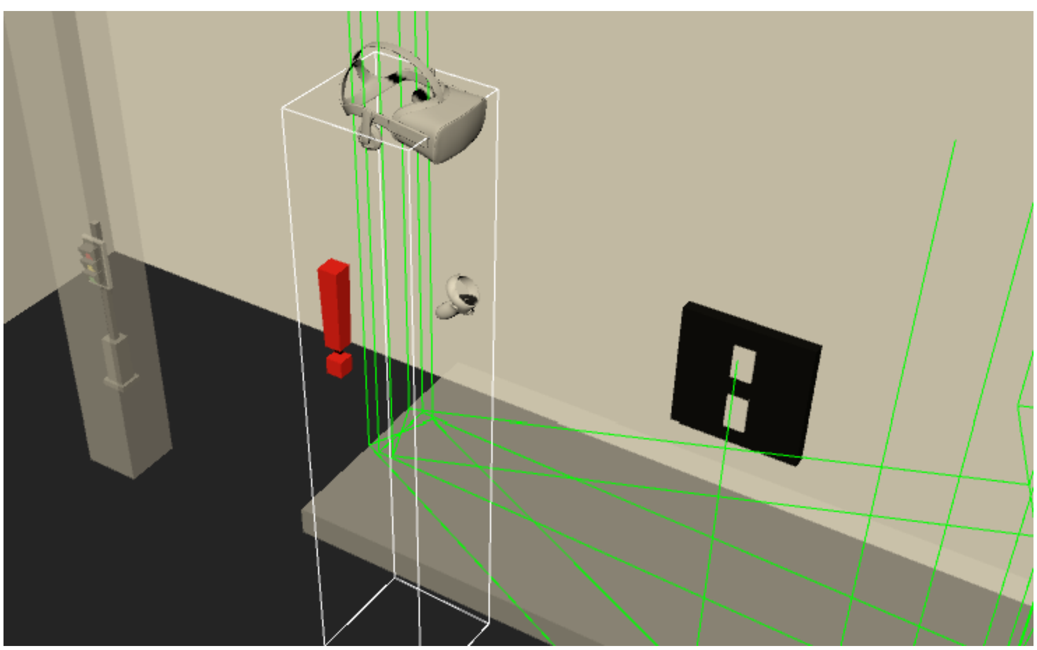


Fig 5.3.3a: snapshot of test run with an incident where the user walked into the road. The red exclamation mark marks the position of the collision.

After the entire coordinate file has been read, the details of the test run can be found in the console at the bottom of the screen. A test is marked as a successful test if the tester has finished the course within the maximum allotted time, which can be set in the UI.

Fig 5.3.3b: UI with controls for playback and console showing test run details.

5.3.4 Analysing multiple test runs and data export

As part of the implementation for automating data measurements, a functionality has also been implemented which allows for the analysis of all available data in the work folder in order to generate insights and statistics for each scene. This is done by effectively running through all of the test runs one by one and performing the same measurement process described for analysing single test runs. This data is then grouped into data for each scene and exported to a csv file in the work folder.

	A	B	C	D	E
1	Environment	AVG Test Time	AVG Incidents	Pass	Fail
2	scenario_one_v4	307.02	2	0	2
3	scenario_three_v2	229.36	0	2	0
4	scenario_two_v2	195.66	1	2	0

Fig 5.3.4: CSV file exported from analysis of final test runs

Chapter 6: Evaluation and Testing

As part of the iterative development, multiple tests have been conducted throughout the project to informally test the device, with a final formal test being conducted at the start of July. Each test that has been performed using the VR tool or the Environment Creation tool will be discussed in the following sections.

6.1 Actuation tests

One of the main purposes of this project is to enable quick implementation of sensor systems, allowing the actuation project to test their device and the haptic language during earlier stages of the project. These tests will be described in the following sections, outlining their purpose, the setup, and the changes made to the VR tool that came from observations during these tests or changes that were made before to accommodate this test. The results of these tests will not be discussed, as they fall under the scope of the actuation project.

6.1.1 Stationary Resolution test

In order to determine whether the current proof of concept worked, the first test conducted was a test where the tester was sat down on a chair and only allowed to move their head to look around. Two objects were placed on set locations in front of the user with two different setups required, with each setup being used for one test. Each test would compare two versions of the sensor system, with different field of view angles, number of columns, number of rows, and radiuses.

Setup one:

- Box 1: 1 meter to the left and 2 meters to the front
- Box 2: 2 meters to the right and 3 meters to the front

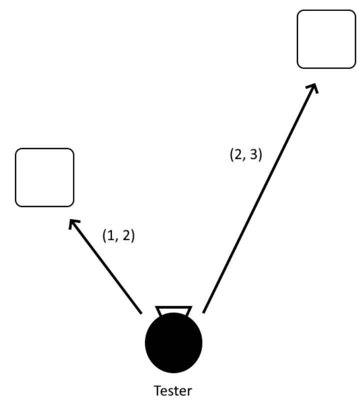


Fig 6.1.1a: setup one

Setup two:

- Box 1: centered and 2 meters to the front
- Box2: centered and 4 meters to the front

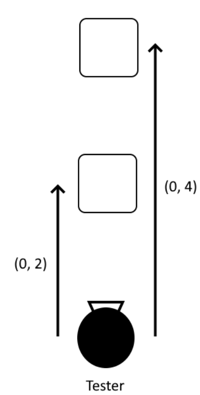


Fig 6.1.1b: setup two

Furthermore, the different setups required different parameters for the sensor system for each test. As a result the design of the VR tool was altered to allow for the quick tweaking of values, by implementing these values in the Unity UI.

6.1.2 Stationary Continuous haptic feedback test

After the previous test, the actuation project sought to find out whether the chosen approach of haptic feedback could be improved. Originally, the feedback would trigger for each column of the grid one by one. As a result, data from the cells was requested per column. However, in an effort to try something different, it was decided to test out a feedback which would trigger for each column, but simultaneously. In order to do this, it was requested by the actuation project that the protocol be reworked in order to minimize the number of requests from the actuation. As a result, a new protocol was devised. Using this protocol, the actuation would send a request for the entire grid, and the VR tool would output the 1-based index of the closest object in each column, with 0 being reserved for empty columns.

Actuation requests grid with content:

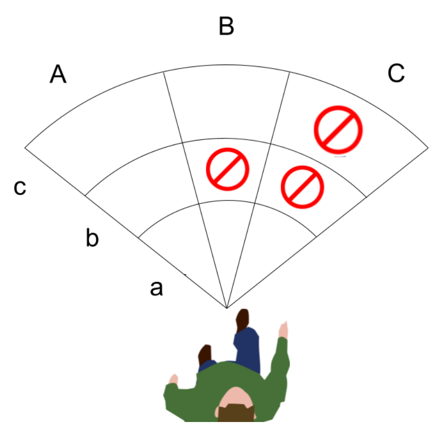


Fig 6.1.2: diagram representing the example grid state. All empty cells represent an absence of objects and all forbidden signs represent an obstacle.

Actuation: “Request”

Sensors: “A0B2C2”

While this test was successful, it was later decided by the actuation project that this approach was too unreliable in terms of haptic perception. Subsequently, the old communication protocol was restored.

6.1.3 Walking box obstacles test

The first test done with a fully functional portable prototype was used to test the concept of the device and probe the intuition. In order to achieve this, a setup was devised that was different from the final test in order to test the calibration functionality of the VR tool. It was decided that in order to test the accuracy of the calibration, real life cardboard boxes were scattered around the test environment. These boxes were then measured and an implementation was used to place virtual boxes with the same measurement manually inside of the VR tool. As the measurement of the boxes were the same, the positioning of the boxes would be the same if the calibration of the VR tool worked as expected. This was indeed the case and this environment with boxes in both the real and the virtual environment was used for the test with the actuation.

The goal of the test was simply to have a participant walk from one side of the testing area to the other while avoiding the boxes. A person would be standing at the opposite side and would periodically call out, giving the tester an indication of where they needed to go.

During the test it was observed that participants would often be unable to sense objects once they got too close to the object. This was due to the fact that the system was designed to be used in combination with a walking cane, which is why there is a set amount of space in front of the user which is ignored by the VR tool. Since this test was not done with VI people, and due to later tests being unable to be done with VI people, it was decided to remove this margin and stop ignoring the objects directly in front of the participant.

It was also found that due to the fact that boxes could span multiple columns, testers would often run into boxes as the system was designed to only store one obstacle in one cell of the grid. This meant that users would only feel the box in one column, while in reality it spanned multiple. To combat this, a new implementation was used for the VR tool sensors, of which more information can be found in Chapter 5.1.1.3.

Fig 6.1.3: combined footage from real life and virtual environment during one of the test runs.

6.1.4 Stationary device variations test

Finally, a test was requested by the actuation project to test out different configurations for the current feedback concept. The current concept used a vest with haptic motors on the back to indicate objects. However, multiple other configurations were to be tested, one of which required an implementation of a protocol where columns, rows were to be requested. Subsequently, this implementation was made for this test only, as later devices reverted back to the vest design using the original protocol.

6.2 Final Evaluation

For the final evaluation of the entire device, the original test setup as described in Chapter 2.5 has been executed. It should be noted that the final evaluation has been attempted twice, as the first attempt yielded unusable results, due to the fact that a bug in the actuation device caused users to experience the wrong feedback signals. A second test has subsequently been run and completed, of which the results and setup will be discussed below.

6.2.1 Evaluation design

As discussed in Chapter 2.5, the evaluation of the entire device is focused on 2 components. Firstly, the overall effectiveness is evaluated by measuring the amount of time it takes to run through the test and the number of incidents and collisions that have occurred. Secondly, the user experience is evaluated using a post-survey with as goal the measurement of perceived usefulness and effort in using the device. As a result, the entire evaluation consists of a practical part, where the user has to walk through a test environment, and a non-practical part, where users fill in a survey concerning their experience during the practical part.

6.2.2 Participants

Participants for this final evaluation include two non visually impaired students from the UT, due to issues in contacting visually impaired people. Testers have no prior experience with the device and are instructed on the usage for the first time before starting the practical part of the evaluation. There are no further requirements for participants.

6.2.3 COVID-19 Coronavirus Disease considerations

In light of the Coronavirus pandemic, measures were taken per recommendations by the government, in order to prevent the spread of COVID-19. Participants are to wear masks, only being able to take them off during the practical part of the evaluation. Disinfectants and sanitizers will be present on site, with the headset being disinfected and wiped clean after each use.

6.2.4 Procedure

At the start of the test, the participant is seated with the actuation project member who explains the usage of the actuation vest and glove. The user is then instructed on the test procedure for the practical part and the safety features. In the following sections the test procedure is explained for each part of the evaluation.

6.2.4.1 Functional Effectiveness: Environment simulation

The practical part consists of three test runs, with each run being dedicated to one test environment. The three test environments are based on the three chosen user case scenarios as mentioned in Chapter 4. For each environment, the user has 5 minutes to move from their starting position to a given goal, while trying to avoid obstacles. The given goal is a specific type of environment object, such as a door or a traffic light. If 5 minutes have passed without the participant having round the goal, the test is ended and considered a failed run. Before the test run, the participant will be led to the starting point, and the environment and the goal will be described to the participant in words. Then, they will put on the VR headset facing away from the environment. Once the participant is ready to start the run, they will press the B button on their controller, enabling the blindfold and the storing of coordinates, and starting the test run. If the participant has successfully reached their destination, they will be instructed to press the B button again, and the run will end. From the data stored during the test run, the measurements for the test time and number of incidents will be taken afterwards. All participants run through the environments in the same order; scenario one, scenario two, and last scenario four (see Chapter 4).

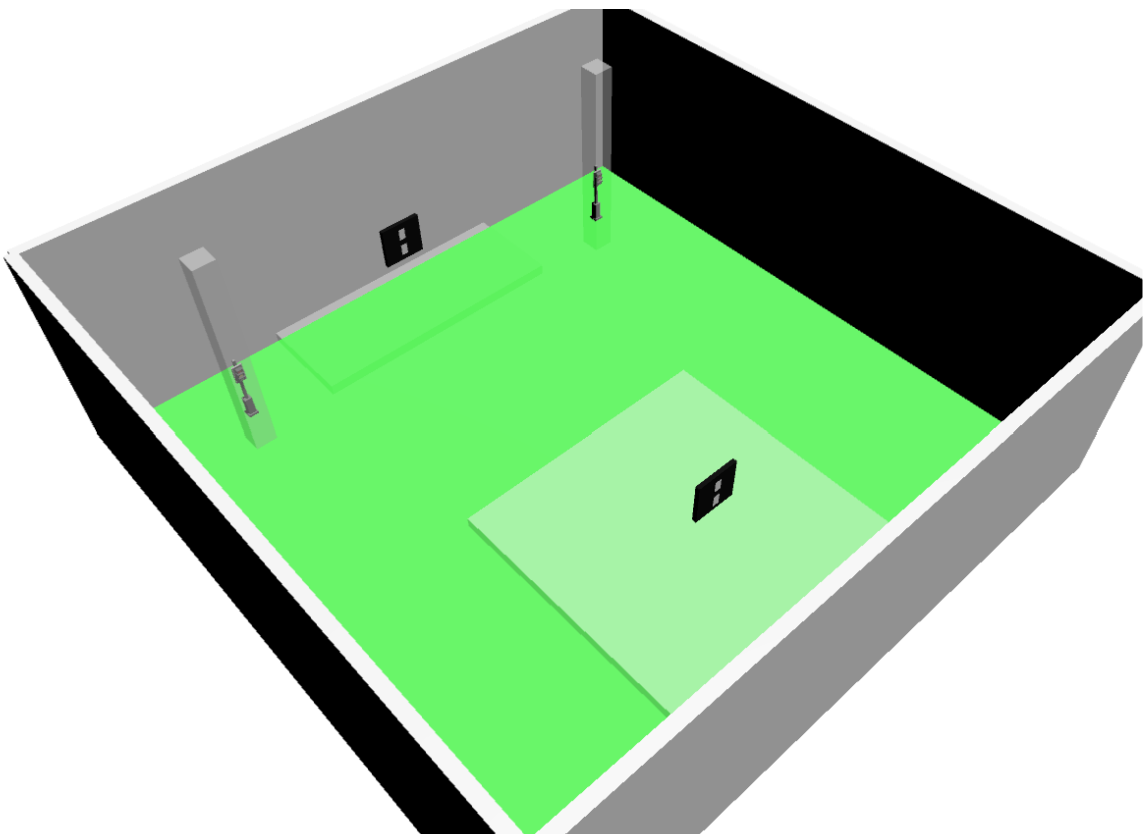


Fig 6.2.4.1a: Scenario one, contains a road with a pedestrian crossing and two traffic lights on each side.

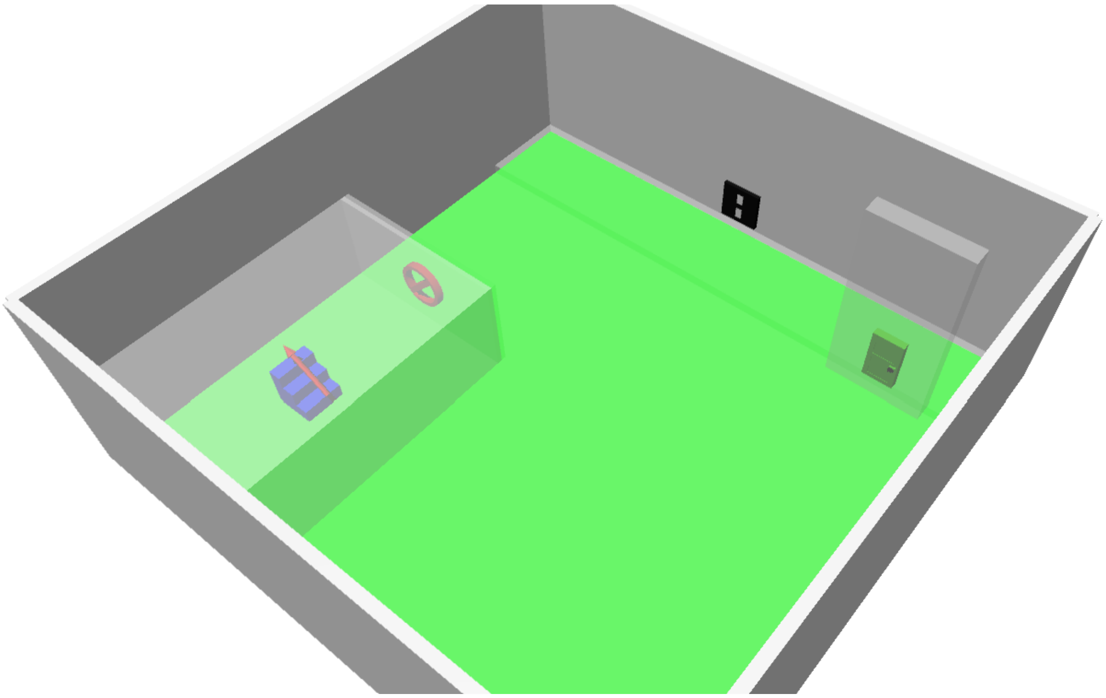


Fig 6.2.4.1b: Scenario two, represents a train station with tracks, a door representing the train door, and a staircase.

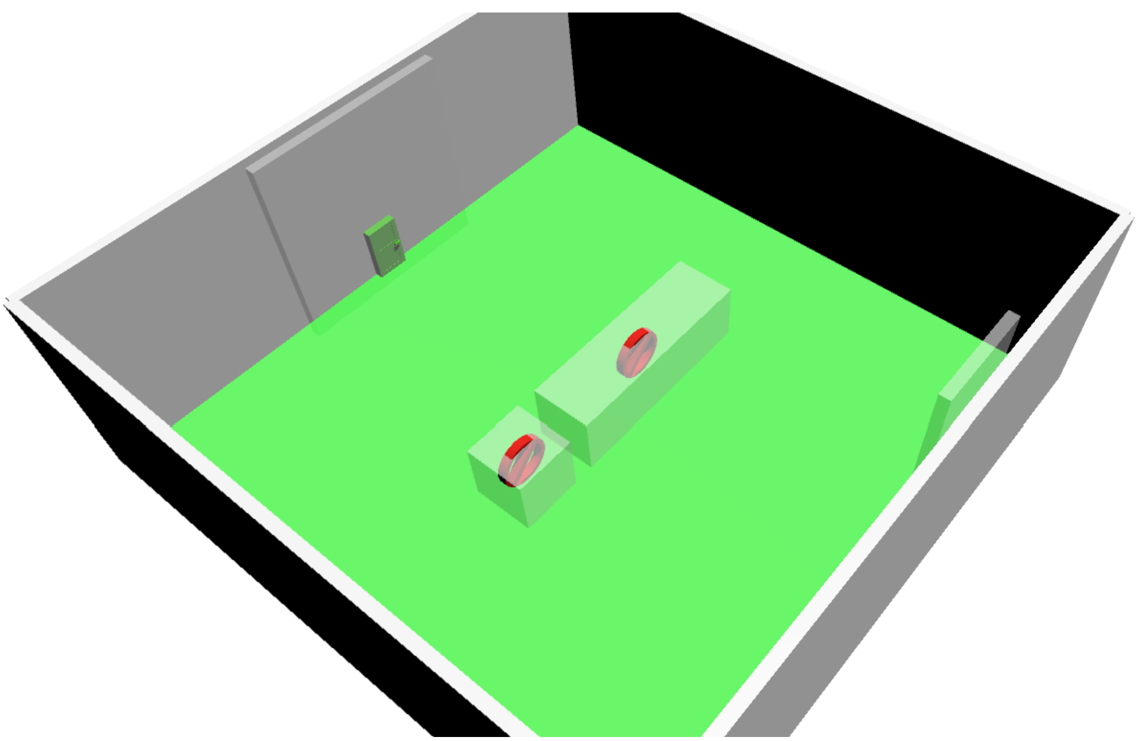


Fig 6.2.4.1a: Scenario three, consists of a street with two wide obstacles in the middle and entrances to shops on either side of the street..

6.2.4.2 User Experience: User survey

After the last test run, the participant is seated next to a laptop and given a survey to fill in through Google Forms. The survey consists of a mixture of open and multiple choice questions and takes approximately 10 minutes to fill out. A snapshot of the entire survey including questions belonging to the actuation project is included in Appendix A5.

6.2.5 Results

6.2.5.1 Functional Effectiveness results

Environment	AVG Test Time	AVG Incidents	Pass	Fail
scenario_one_v4	307.02	2	0	2
scenario_three_v2	229.36	0	2	0
scenario_two_v2	195.66	1	2	0

Fig 6.2.5.1a: raw output from the evaluation tool showing statistics for each environment

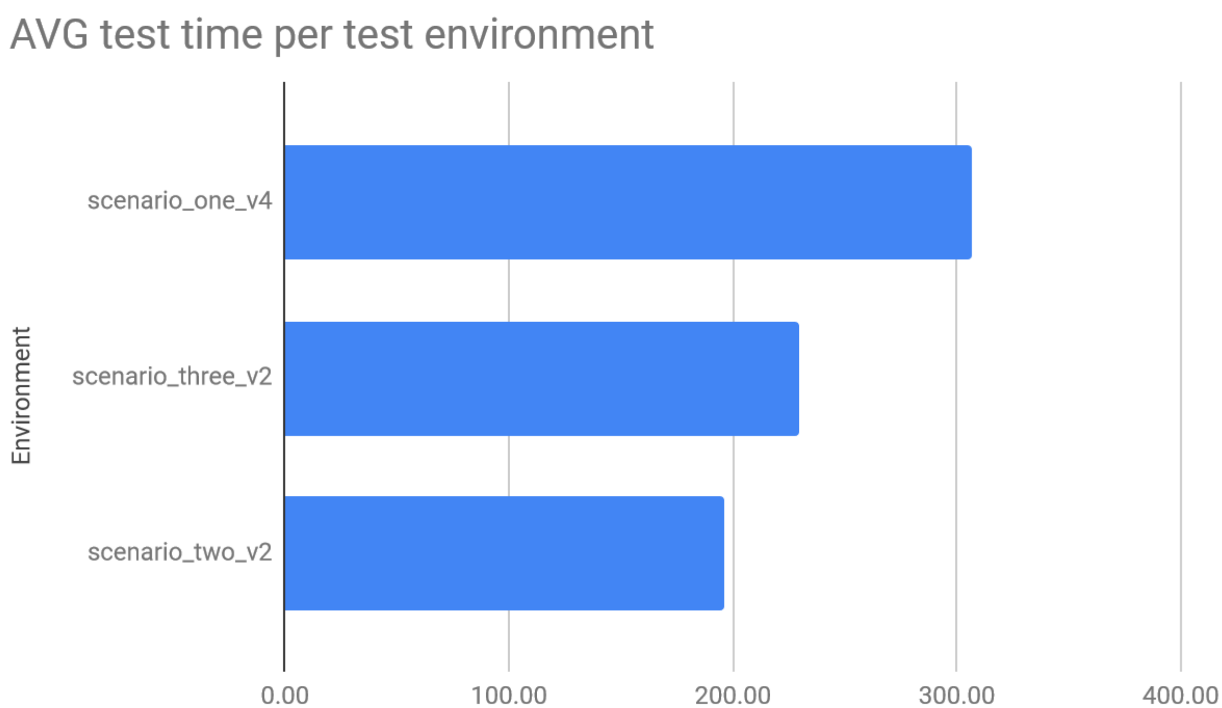


Fig 6.2.5.1b: average test time across all runs per scenario in seconds

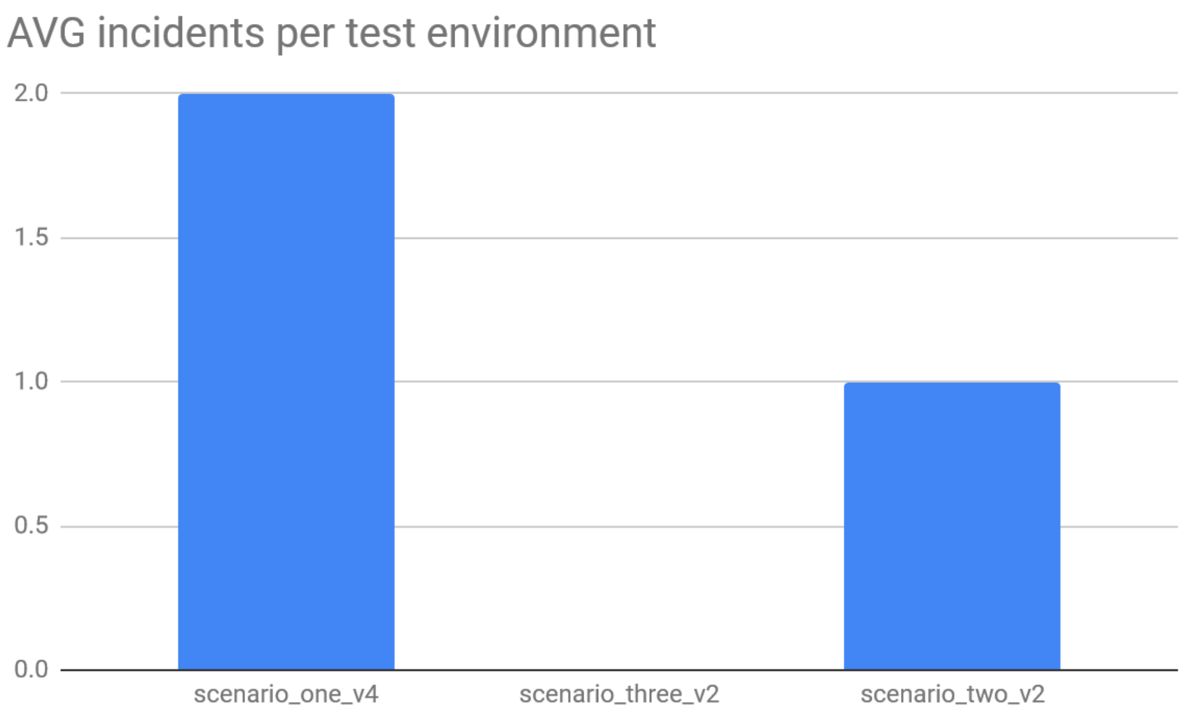


Fig 6.2.5.1c: average number of incidents across all runs per scenario

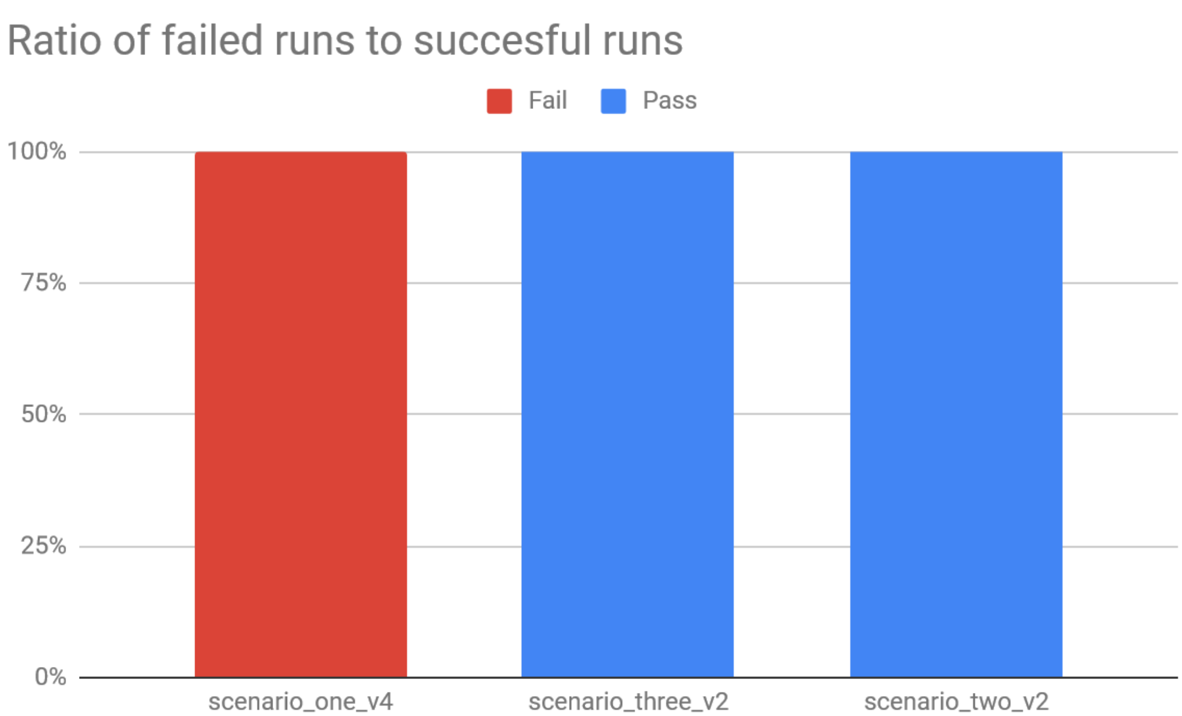


Fig 6.2.5.1d: ratio of failed runs to successful runs of all runs per scenario in percentages

Results show that during the practical part of the experiment, the first environment that was tested generally has the worst performance with none of the users reaching the intended goal and the highest number of average incidents being 2. However, the two scenarios that were tested afterwards show an increase in performance with all of the subsequent runs being successful and an average number of incidents being 0 and 1 respectively for scenario three and scenario two.

6.2.5.2 User Experience results

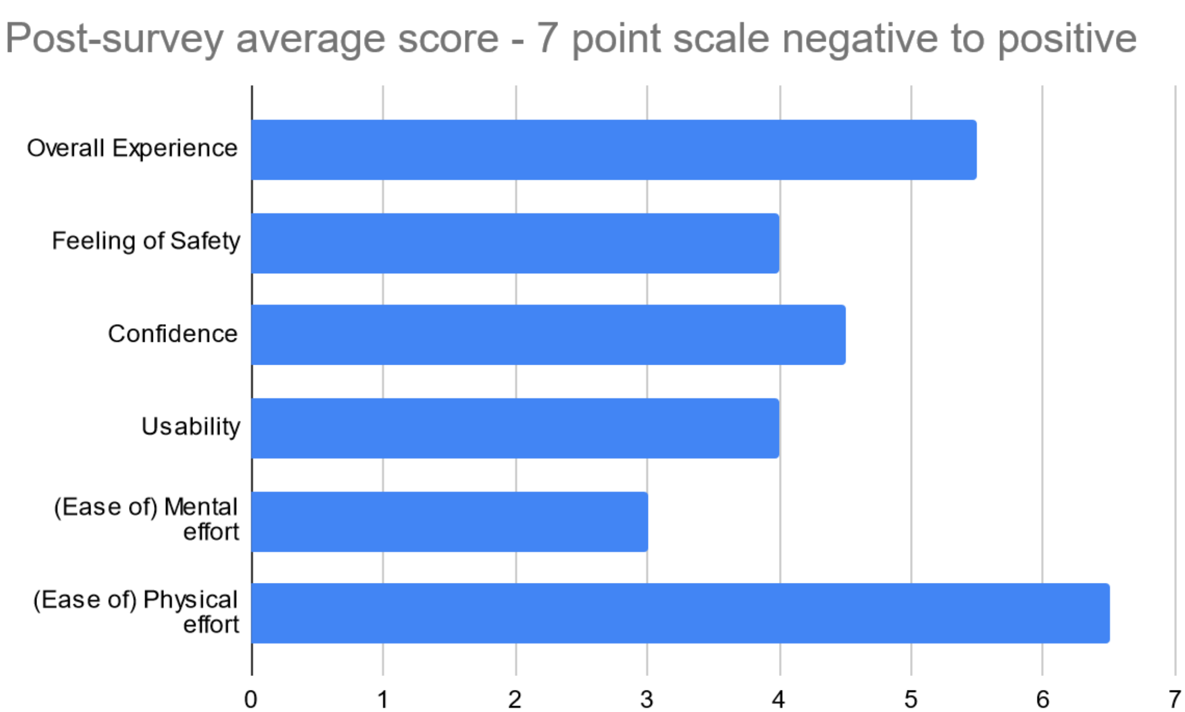


Fig 6.2.5.1: average of results from evaluation survey. Scores range from 0, denoting negative experiences, to 7, denoting positive experiences.

In general the device scored slightly above average. Overall experience and ease of physical effort are rated considerably well, with safety, confidence, and usability rating okay. Mental effort is considered to be the biggest flaw of the system, being rated at an average score of 3 towards high effort. In open questions, participants mainly commented on the difficulty of understanding and recognizing patterns on the hand device and sometimes the vest. They also noted that especially at first the device can be overwhelming and that it “takes time to properly understand all the information that the device gives you”.

6.2.6 Conclusion

In general the evaluation of the device shows that it can be effective, with a relatively low number of accidents and a solid proof of the fact that people are able to recognize objects through haptic signals. However, considering the usability and the effort of the device, many improvements can still be made, especially in lightening the mental load and improving understanding of the feedback. It should also be noted that for such a complex device, a case could be made that it takes time to properly learn the device. This could also be indicated by the fact that the first test run went relatively bad compared to the later runs for both of the participants.

Chapter 7: Discussion

7.1 Limitations of this project

7.1.1 COVID-19 and test participation

At the start of the project, much time was spent and lost on finding visually impaired people for the focus group. This led to a decreased scope in the project as implementation and design decisions were moved forward, being delayed by the lack of user input. Eventually, visually impaired people were contacted through personal channels, though the majority of these people subsequently would not be able to participate in the final evaluation as they did not live in the Netherlands. Due to this, it was decided to only use students at the UT for testing purposes, as this would also minimize risk of potential COVID-19 spread outside of the University bubble.

7.2 Reflection on current state and improvements

Due to the diminished scope of the project, some features could not be implemented into the final product. A compilation of suggestions for future improvements is given below.

7.2.1 Usage of non-euclidean virtual environments

In using an enclosed real life environment for testing, problems can arise when needing to test larger virtual environments. If an entire street or route were to be tested in the real world, this would require an enormous amount of space which would need to be cleared and monitored. Instead, the usage of virtual environments allows for non-euclidian environments, which would be able to increase the perceived real life space by warping the space around the user. A practical example can be seen in the VR game Tea For God[22] which uses this technique to allow the user to walk without artificial locomotion. In practice, this effect can be achieved by creating certain areas at the edges of the test area which force the user to walk past a corner. The moment the user passes through that corner, the scene can then be altered so that another part of the scene is loaded in front of the player.

7.2.2 Usage of bluetooth communication between device and actuation

Due to the scope of the project, it was chosen to use Serial communication between the sensors and the actuation. However, the initial plan was to use bluetooth communication for this purpose. Using bluetooth would remove the need for physical connections between the actuation and the sensors, allowing the system to be more modular and portable.

7.2.3 Using Oculus Guardian functionality for audio warnings

Since the tests were performed with non visually impaired people, it was possible to have the blindfold implementation, which allowed testers to still see safety boundaries of the Oculus Guardian. However, for blind people this is different. Instead it is possible through the Oculus SDK to get the boundaries from the Oculus Guardian manually and implement a safety feature based on audio. This would be possible by playing a certain audio cue when testers are approaching the edge of the test environment.

Chapter 8: Conclusion

SQ1

In conclusion, people with a visual impairment have trouble navigating new environments as they are unfamiliar with new routes and rely on orientationally significant objects to navigate.

SQ2

One of the ways to alleviate this problem is by helping visually impaired people better understand their environment by giving them a bigger range of vision and allowing them to find and identify objects of interest.

SQ3

In designing and developing a device to solve this problem, one should measure the effectiveness of the device through measures that directly relate to the problem and the people involved. In this case, the mobility, safety, and experience that comes from such a device is most important to the users. This can be measured by looking at the time it takes to complete certain courses, counting the times where the device fails, and probing users through surveys.

SQ4

In order to test such a device, Virtual Reality peripherals can be used to map virtual environments onto real environments by tracking the floor of both environments, aligning these and then subsequently using points in the environment which act as the axes of the environment. By aligning all of these axes, it is possible to reliably track a virtual environment onto a real environment.

SQ5

Lastly, by gathering information on the limitations of real life sensors and careful communication it is possible to rebuild both real life environments and real life sensors and their functionality into a virtual environment.. Furthermore by automating the evaluation process as much as possible and allowing for flexibility through the easy changing of variables, the external creation of scenes, and cross platform support, it is possible to effectively use digital tools and VR together to evaluate such haptic devices.

This project, while flawed, has been proven to be functional during testing and evaluation of the device. Considering its limitations, more focus should be put on developing a haptic language itself in order to minimize the mental effort in understanding haptic signals. Great potential lies in finding ways to abstract and minimize the data that is sent to the user. All in all this project should be considered as a proof of concept for the usage of haptics in assisting the visually impaired to navigate their surroundings.

Appendix A

A1

Interview 2 Questions:

Introduction:

1. Interviewee
    a. Age
    b. Intensity of visual impairment
    c. How long have you had the impairment?
2. Use case
    a. What would you describe are some of the key issues you experience about navigation?
    b. What type of obstacles can you recognize? Which not?
    c. What can't you currently do in navigation, but would you like to be able to do?
    d. If you could develop a device to help you with that, what would that device do?
    e. If you could pick one situation in which you were not blind, what would that situation be?
    f. What type of situations do you typically are oblivious to?
3. What are your thoughts on the following scenarios?
    a. Finding a designated crossing point on a street (i.e. Traffic light, zebra crossing)
    b. Finding and successfully navigating stairs in outdoor spaces (i.e. Train platform)
    c. Staying on the sidewalk when it is not clearly marked (i.e. Not raised, no clear border pattern)
    d. Navigating through wide streets or places with obstacles in the middle of the path (i.e. Benches in the middle of the path, a mall with kiosks in the middle)
    e. Navigating on strongly curved paths without clearly marked path edges (ie. Park)
4. Aids
    a. What devices do you currently use for navigations?
    b. Why did you choose them?
    c. What are the shortcomings of these devices?
5. Design criteria
    a. How wearable (Size, comfort, weight)
    b. How pervasive (how much should it be noticeable/in the foreground
    c. How discriminatory (what about the exclusivity/ branding/price)

A2

Problems with guiding cane

Cane can bounces over smaller holes in the ground

- ➔ Brain can't always react to such a small detail that is barely noticeable
- ➔ Leads to stepping into hole possible twisting ankle or worse

Cane does not provide enough range

- ➔ Not enough reaction time

Current Devices

Fledermaus – attached to cane; not water resistant

RTE - Ultrabodyguard – worn around the neck; ultrasonic range finder; light; water resistant; includes 3 large buttons, gives haptic feedback on neck (the closer to object the harder), 250grams

Smartphone with voice assist (IPhone) - lots of functionality; voice over can give to much information -> overwhelms user

Cane – 1.25meters long; 0.9m range; golf ball sized ball at bottom -> detects different surfaces; attracts dogs

Sünfon – Barcode reader - a bit heavy, can save new barcodes for home use

Orcam -> reads written signs; needs slight visual abilities for aimin; cant read handwriting; great for shopping; long startup delay (1 minute); very light; 90min charge; multiple languages; very small buttons

Vistac - Color detector - detects color 95% correct, light and small

Measuring help for placing pots on a stove

Magnifying glasses

Feelware – home products without touchscreens

Navigation belt – worn on skin -> leads to excessive sweating; overwhelming input

Problematic Objects

Thin objects -> Street lights

Small indoor items -> water glass

Problematic Situations

Unknown surroundings

Height differences

(People tend not to be a problem)

Large open spaces -> individual loses orientation to easy

Construction sites

Downward leading stairs - > easily missed by cane and tripping hazard

Uneven ground

Dogs see the cane as a toy -> individual can't tell what the dog is doing or if it is aggressive

Doors and stairs

Large risk

Sliding doors are great

Half open doors are dangerous -> doorway might be detected but door isn't

Finding specific Objects

Difficult to impossible

Done entirely by feeling

Putting down thing on a table poses problems

Finding keys

Street signs

No current devices

Can detect sign but not tell which it is

Orcam can read out street names

Street awareness

Large problems with not raised sidewalks

Moving objects

E-scooters, Electric cars, bikes -> very quite and fast -> hard to react to

Often doesn't lead to crash but creates a large scare

Point to Point

Works well with navigational helps (Google maps etc.)

Should be more precise in its specifics (when to turn exactly)

Wearability

Very important (9/10)

Can't be too heavy or to big (phone size is good)

Sweat creation is a problem

Mental load/Signals

Signals should be clear

Must be able to bring you back from a wandering mind

Exclusivity

Can be visible -> might depend on social surrounding, more likely problematic for teenagers

Can be specifically made for visually impaired

Does not have to be a general use item

No consensus if all functions should be in one device or split over multiple

Price

Between 100 – 3000$

Orcam ~3000$

Wishes for any device

Must be light (max 250g)

Water resistant

Not acoustic

Haptic feedback

At least 3m range

Deterring to dogs (high pitched noise or smell)

Few functions for simple use

No touch screens

Large "scanning" radius

Hands free

Should provide a general feel for the surrounding

Other

Anything attached to cane has to be very light to keep it usable

Glasses are used as eye protection

Thin gloves protect from cuts but still give tactile feedback (textile is better than rubber)

White writing on black ground is more readable

Watch style is very practical

Many devices are not supported by health care provider

Podcast - Apfel Fleger -Herr Fleger

A3

Interviewee

65 years old

Right eye completely blind

Left eye only peripheral left

Used Devices

Guiding Cane

Smart phone

Alexa/Amazon Echo (not in use but wanted)

Problematic Situations

Dogs, Cyclists

Smaller holes in the ground -> Holes with little contrast especially I.e. bad lighting/colors

Bus stands with glass sides

Street signs with grey posts -> yellow posts are easier to see

Doors/Stairs

Finding doors are not a big problem themselves

Doorhandles/Doorbells are hard to find

Downward stairs are harder than upward

Contrast on the edge of stairs helps -> edge markings help a lot

Search Items

Slow but possible

Start in the center and go in circles of increasing size

Traffic Signs

Barely legible

Yellow signposts are great

Street awareness

Raised sidewalks help a lot

Clear color contrast helps

P2P

Phone for navigation is not enough

Info on where traffic lights are would help

Wearable/Functionality

One hand occupied by Cane

Should be easy to find but also not to big

Should be very intuitive -> to old to learn new things

Needs clear/intense signals -> sense of touch is not as good anymore

A4

Interviewee

Daan Rutgers
28 years old
Completely blind since ½ years old

Used devices

I-Cane
White Cane
Smart Phone (but doesn't really use Google Maps)
GPS orientation point tracker (device which saves gps coordinates of known orientation points and will say the name of the point when the point is reached.

Problematic Situations

Suspended / protruding objects.
Navigating through unknown environments with no guides.
Finding their way through unknown public transport stations

Suspended / protruding objects

Main problems arise with objects which protrude or otherwise unexpected at eye height. Specifically, an example was given where there was a sign from a construction site which was attached to a building and not to a signpost. These types of obstructions cannot be detected with a white cane.

Navigating through unknown environments

Going to new places is tricky as it requires the learning of a new path. Mostly navigates by learning "orientation points" which he uses to determine when to turn etc. These orientation points can be found using either the cane (for example small hills to the side of the road) or using aids such as GPS devices. The problem in new environments is that there is no pre knowledge of orientational points, which is why new routes are typically walked with someone else first to assist. Main examples include missing a bus or train stop and not being able to know where to go to take a train or bus back and navigating the station (crossing roads, finding the route towards the station).

Thoughts on the user scenarios

Liked the idea of the pedestrian crossing. There are crossings which can be indicated with profiled tiles on the floor, but those are not necessarily present at all crossings.
He notes for the scenario with detecting the ditch, that in some cases, being able to feel objects/terrain next the pavement can be useful as they can be used as orientation points. He does not mention it explicitly, but given the context we will be assuming that he does not always want to be guided away from "differences in height next to the pavement" as they could act as an orientation point.

Also liked the idea of being able to detect entrances etc. Before this, he mainly used a GPS device (which is no longer in production to his knowledge), which he could use to "save" GPS locations which he could use as orientation points. The user would also be able to input a name and an instruction for each saved location, so they could for example save a signpost and save the instruction "turn right here" for that location. However, this system was limited and would not take direction into account, thus it would just say turn "right here", regardless from which direction you arrived. He also doesn't want to rely too much on devices as he wants to be able to handle himself in case a device fails, or is forgotten and left at home etc.

On the I-Cane

He absolutely loathed using the I-Cane as it was not precise and mostly just annoying. In particular, he mentioned an instance where he used the I-Cane in a shopping centre and it just kept beeping (overload of information). This really gave him no useful information as it wasn't really clear what the cane was seeing, which is why he stopped using it quite quickly. Furthermore, the I-Cane would also see orientation objects as obstacles, which was quite annoying as he needed to feel those orientation objects to know where he was.

On other technological aids

On the computer he mostly uses the accessibility functionality of the pc, which will read text for him. More interestingly he mentioned that he also uses something called a "refreshable braille display". This device acts as a dynamic braille display using mechanical pins in order to form braille sentences dynamically. According to him, blind people will have varying degrees of skill with braille and will usually use either text to speech or braille more, depending on their skill and preference.

Final thoughts

Especially mentions the concern of alienation through the device, but otherwise he has no further remarks on the design specifications. Also is willing to help out with testing, pending the corona restrictions.

A5

Evaluation of a haptic navigation device

How would you describe your impression of the device you just tested?

	1	2	3	4	5	6	7	
I did not like it	○	○	○	○	○	○	○	I liked it very much

How would you rate the ease of understanding and using the device?

	1	2	3	4	5	6	7	
I found it difficult	○	○	○	○	○	○	○	I found it easy

How would you describe the amount of mental effort it took you use the device?

	1	2	3	4	5	6	7	
very challenging	○	○	○	○	○	○	○	very light

Which elements in particular would you say contributed to that view on the mental effort?

Your answer

How would you describe the amount of physical effort it took you to use the device?

	1	2	3	4	5	6	7	
very challenging	○	○	○	○	○	○	○	very light

Which elements in particular would you say contributed to that view on the physical effort?

Your answer

How would you rate the usability of the device?

	1	2	3	4	5	6	7	
Not usable at all	○	○	○	○	○	○	○	Very usable

Do you have any notes about the usablity ?

Your answer

How would you rate the wearability of the Vest and Gauntlet?

	1	2	3	4	5	6	7	
not very comfortable	○	○	○	○	○	○	○	very comfortable

How would you rate the amount of sweating expericed while wearing the Vest/Gautlet

	1	2	3	4	5	6	7	
A lot	○	○	○	○	○	○	○	Not much at all

Do you have any notes about the wearablity of the Vest or Gauntlet ?

Your answer

What did you like or dislike about the Vest or Gauntlet? Do you have suggestions for improvements ?

Your answer

How would you rate the addition of the gauntlet to the vest

	1	2	3	4	5	6	7	
it does not fit well together	○	○	○	○	○	○	○	it fits well togehter

How would you rate your experience of the device overall ?

	1	2	3	4	5	6	7	
very bad	○	○	○	○	○	○	○	very good

Which elements, positvely or negatively, influenced your experince of the device ? Were there any points of frustration ?

Your answer

During the test, were you able to imagine or understand your surroundings?

Your answer

How would you rate the level of confidence you had in your surroundings during the test ?

	1	2	3	4	5	6	7	
very unsure	○	○	○	○	○	○	○	very confident

How would you rate your level of confidence in understanding and interpreting the device?

	1	2	3	4	5	6	7	
very unsure	○	○	○	○	○	○	○	very confident

How would you rate the clarity of the decice ?

	1	2	3	4	5	6	7	
not very clear	○	○	○	○	○	○	○	very clear

Which elements did you find difficult or easy to interpret?

Your answer

How well were you able to discern the distance of objects ?

	1	2	3	4	5	6	7	
very difficult	○	○	○	○	○	○	○	very well

How well were you able to discern the position of objects ?

	1	2	3	4	5	6	7	
very difficult	○	○	○	○	○	○	○	very well

How would you rate your percieved level of safety with the device, given that you were blindfolded ?

	1	2	3	4	5	6	7	
very unsafe	○	○	○	○	○	○	○	very safe

Which elements did influence you perception of safety in either way ?

Your answer

How would you rate the level of intuiton for the vest ?

	1	2	3	4	5	6	7	
not intuitive	○	○	○	○	○	○	○	very intuitive

Did you asssociate the vibration of the vest with anything?

Your answer

How would you rate the level of intuition for the gauntlet ?

	1	2	3	4	5	6	7	
not intuitve	○	○	○	○	○	○	○	very intuitive

Did you associate the vibration of the gauntlet with anything ?

Your answer

To your knowledge, did you have any intuitive responses to any of the vibrations from the device ?

Your answer

Do you have any notes about the intuitiveness of the device?

Your answer

Do you have any further notes about the device, the test or the idea in general ?

Your answer

If you could change/add one thing, what would you want to change/add?

Your answer

Do you have any suggestions about the device, the test or the idea in general?

Your answer

References

[1] Shi, Y., Azzolin, N., Picardi, A., Zhu, T., Bordegoni, M., & Caruso, G. (2020). A virtual Reality-based platform to Validate HMI design for INCREASING USER'S trust in autonomous vehicle. *Computer-Aided Design and Applications, 18*(3), 502-518. doi:10.14733/cadaps.2021.502-518

[2] Maule, L., Fornaser, A., Leuci, M., Conci, N., Da Lio, M., & De Cecco, M. (2016). Development of innovative hmi strategies for eye controlled wheelchairs in virtual reality. *Lecture Notes in Computer Science,* 358-377. doi:10.1007/978-3-319-40651-0_29

[3] Diederichs, F., Niehaus, F., & Hees, L. (2020). Guerilla evaluation of Truck HMI with VR. *Virtual, Augmented and Mixed Reality. Design and Interaction,* 3-17. doi:10.1007/978-3-030-49695-1_1

[4] Chew, J. Y., Okayama, K., Okuma, T., Kawamoto, M., Onda, H., & Kato, N. (2019). Development of a virtual environment to realize human-machine interaction of forklift operation. *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA).* doi:10.1109/ritapp.2019.8932837

[5] Singh, B., & Kapoor, M. (2021). A framework for the generation of obstacle data for the study of obstacle detection by ultrasonic sensors. *IEEE Sensors Journal, 21*(7), 9475-9483. doi:10.1109/jsen.2021.3055515

[6] Saha, S., Shakal, F. H., & Mahmood, M. (2021). Visual, navigation and communication aid for visually impaired person. *International Journal of Electrical and Computer Engineering (IJECE), 11*(2), 1276. doi:10.11591/ijece.v11i2.pp1276-1283

[7] Dos Santos, A. D., Medola, F. O., Cinelli, M. J., Garcia Ramirez, A. R., & Sandnes, F. E. (2020). Are electronic white canes better than traditional canes? A comparative study with blind and blindfolded participants. *Universal Access in the Information Society, 20*(1), 93-103. doi:10.1007/s10209-020-00712-z

[8] Chen, Z., Liu, X., Kojima, M., Huang, Q., & Arai, T. (2021). A wearable navigation device for visually impaired people based on the real-time semantic visual slam system. *Sensors, 21*(4), 1536. doi:10.3390/s21041536

[9] Ntakolia, C., & Iakovidis, D. K. (2021). A route planning framework for smart wearable assistive navigation systems. *SN Applied Sciences, 3*(1). doi:10.1007/s42452-020-04038-6

[10] Khan, S., Nazir, S., & Khan, H. U. (2021). Analysis of navigation assistants for blind and visually impaired people: A systematic review. *IEEE Access,* 1-1. doi:10.1109/access.2021.3052415

[11] Bizoń-Angov, P., Osiński, D., Wierzchoń, M., & Konieczny, J. (2021). Visual echolocation concept for the colorophone sensory substitution device using virtual reality. *Sensors, 21*(1), 237. doi:10.3390/s21010237

[12] Alzighaibi, S., Bouaziz, R., & Kammoun, S. (2020). An haptic display for visually impaired pedestrian navigation. *Advances on Smart and Soft Computing,* 635-642. doi:10.1007/978-981-15-6048-4_55

[13] Akilandeswari, J., Naveenkumar, A., Iyyanar, P., Paramasivam, M. E., & Jothi, G. (2021). An Investigation on Indoor Navigation Systems. *Advances in Intelligent Systems and Computing, 1176*, 115-124. doi:10.1007/978-981-15-5788-0_11

[14] Khan, M. A., Paul, P., Rashid, M., Hossain, M., & Ahad, M. A. (2020). An ai-based visual aid with integrated reading assistant for the completely blind. *IEEE Transactions on Human-Machine Systems, 50*(6), 507-517. doi:10.1109/thms.2020.3027534

[15] Nasser, A., Keng, K., & Zhu, K. (2020). ThermalCane: Exploring Thermotactile directional cues ON Cane-grip for non-visual navigation. *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility.* doi:10.1145/3373625.3417004

[16] Nair, V., Olmschenk, G., Seiple, W. H., & Zhu, Z. (2020). Assist: Evaluating the usability and performance of an indoor navigation assistant for blind and visually impaired people. *Assistive Technology,* 1-11. doi:10.1080/10400435.2020.1809553

[17] Paiva, S., & Gupta, N. (2019). Technologies and systems to improve mobility of visually Impaired people: A state of the art. *Technological Trends in Improved Mobility of the Visually Impaired,* 105-123. doi:10.1007/978-3-030-16450-8_5

[18] Giudice, N. A., Guenther, B. A., Kaplan, T. M., Anderson, S. M., Knuesel, R. J., & Cioffi, J. F. (2020). Use of an indoor navigation system by sighted and blind travelers. *ACM Transactions on Accessible Computing, 13*(3), 1-27. doi:10.1145/3407191

[19] M. Wilson, "In the home of the future, every surface is an interface," *Fast Company*, 22-Jan-2020. [Online]. Available: https://www.fastcompany.com/90453867/in-the-home-of-the-future-every-surface-is-an-interface. [Accessed: 16-Apr-2021].

[20] "A Technical Look into Mapping Virtual Worlds for Real Spaces," *MediaMonks*. [Online]. Available: https://www.mediamonks.com/blog/2017/07/a-technical-look-into-mapping-virtual-worlds-for-real-spaces. [Accessed: 16-Apr-2021].

[21] "Unity Packages", *Oculus*. [Online]. Available: https://developer.oculus.com/downloads/unity/. [Accessed: 13-Jul-2021]

[22] "tea for god", *void room*. (n.d.). https://voidroom.com/tea-for-god/. [Accessed: 13-Jul-2021]