Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science

Bachelor Thesis

Developing a prototype of a dynamically constructed virtual reality environment for E-Commerce experiences

Sebastian Cucerca

October 26, 2016

Advisor
Marco Speicher

Supervisor
Prof. Dr. Antonio Krüger

Reviewers
Prof. Dr. Antonio Krüger
Prof. Dr. Jürgen Steimle

Declarations

Eidesstattliche Erklärung / Statement in Lieu of an Oath:
Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.
I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Unterschrift / Signature:

Sebastian Cucerca

Einverständniserklärung / Declaration of Consent:
Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.
I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Unterschrift / Signature:

Sebastian Cucerca

Abstract

The domain of virtual reality grew radical and became more and more popular in the last years. The reason for that is the exponentially growing power of smartphones. Since most people own a smartphone and cardboards are very cheap to produce, it is possible for nearly everyone to use virtual reality even at home. Especially in the e-commerce domain, this is an interesting trend. Hence it is important to do research in this domain to increase the usability of virtual environments. To do so the VRProductFinder, a virtual reality e-commerce shop, was implemented. Users can interact with this VRProductFinder by using various interaction types like looking around and speaking. These allow the users to move around in the virtual environment to search for specific products. Different output devices can be used to utilize the scene. In the end, a study was conducted to test the usability of the VRProductFinder and the efficiency of the different search types. As it turned out, the search by speech input was indeed the most efficient and reached the highest usability in combination with VR.

Contents

1 Introduction

1.1 Motivation

The idea of diving into an imaginary world is hoary. Even in the ancient Rome, the philosophers tried to take people into another world by narrating their stories as thrilling and vivid as they could. The wish to experience this feeling hasn't change since then, which is one of the reasons why the popularity of immersive virtual reality(VR) systems has become so popular in a short period of time. Many big companies like Facebook, Samsung, HTC and Sony are working hard on the VR market, so expectations predict that it will be worth seven billion dollars by 2018[6]. The growing and good prospects make it all the more necessary to participate actively in this rising domain.

For a long time, the availability of immersive VR systems for private people was limited. The main reason for this was the cost factor. Additionally, such systems weren't comfortable to wear or even to use. That changed with the invention of cheap cardboards in combination with the popularity of modern smartphones, which are becoming more and more powerful nowadays. The performance of their hardware, as well as the accuracy of the built-in sensors, are improving continuously. Additionally, they are wireless and thus convenient for immersive VR purposes. The 3D computer graphics technologies evolved in the last years to the point that real-time rendered scenes look more realistic than ever[11].

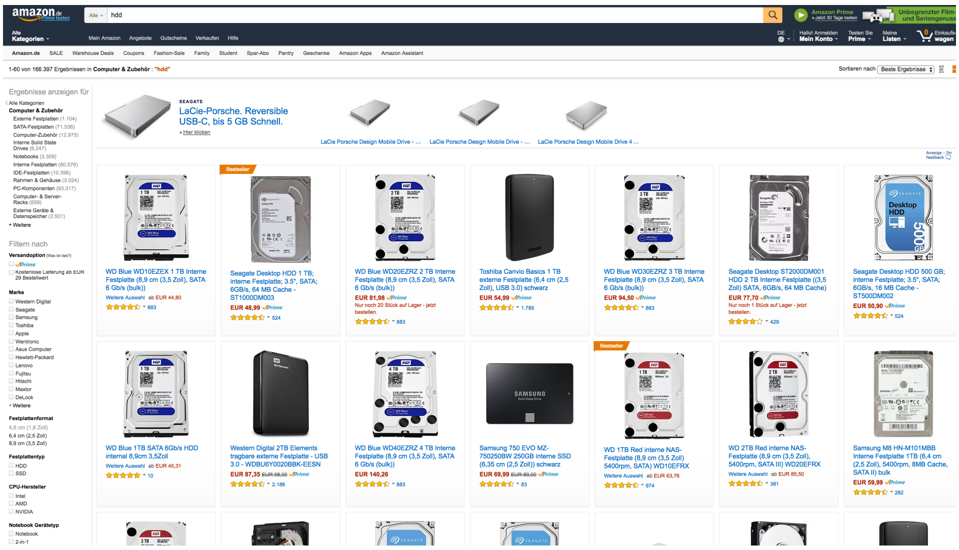


Figure 1: Amazon as an example for an ordinary online shop[3].

The widespread availability and fast growing speed of mobile Internet make it possible to use such applications nearly everywhere. Current e-commerce applications also use this fact, so that more and more users tend to shop on the go. This way of shopping is convenient and thus attractive, which makes it a fast-growing domain. Even though current e-commerce applications are functional, they do not provide a realistic shopping experience(see figure 1). They can be improved by combining them with immersive VR.

There already exist some approaches of creating realistic e-commerce applications by using VR. The problem is that they do not exploit the full potential of today's technologies regarding the possible rendering quality. Nor are they providing any mobility or immersion, although it would be possible with today's smartphones in combination with cardboards or VR glasses. The few provided interaction types of current virtual reality e-commerce applications are not very good.

The main task of this thesis is to combine the idea of mobile e-commerce and immersive VR in a way that it is efficient and realistic. For this purpose a prototype of a VR e-commerce shop named VRProductFinder was implemented. Thanks to the dynamically built virtual environment, there are nearly unlimited possibilities regarding its customizability. Users can move around in the store by using gaze interactions. With the aid of speech input, it is very easy to search for desired products. But a search is also possible by moving through the shop and looking around. A mobile phone in combination with a cardboard can be used as an output method to have a realistic shopping experience. But it can also be used on a desktop computer.

1.2 Research Questions

Since the VRProductFinder can be used on different output devices with multiple search types, the two following research questions came up:

- Does immersive VR provide a better shopping experience than Desktop VR?
- Is the search by speech input faster than the search by gaze?

To test the usability and answer these questions a study was conducted in the end, whose main task was to search for products in the virtual environments using different interaction types and output devices. The independent variables of the study arise from varying search type, the output device and searched product. For every trial, which represents a single search, the task completion time was measured. Moreover, every user had to fill out some questionnaires, which will be explained in the study section in detail. The results of these questionnaires, together with the measured times, were the dependent variables of the study.

1.3 Significance of the study

Like it was mentioned before the existing e-commerce applications do not use immersive VR as output possibility. The VRProductFinder implements exactly this feature but also provides Desktop as output. While most of the current e-commerce shops don't provide many interactions, the VRProductFinder has a search by speech input, so that it should be possible to find a product faster in addition to the common search by gaze. So, the user has the possibility to use the VRProductFinder with Desktop or VR as output. To search in the environment, speech search or gaze search can be utilized. These options result in four different scenarios, which define four tasks. One of the primary purposes of the study was to find the best out of these four tasks.

The conducted study gives insights into the usability of the implemented VRProductFinder. It will prove that the usability of the VRProductFinder is not bad. Particularly in the VR setup, the speech search provided an added value regarding the usability. Furthermore, the question about the efficiency of the speech search towards the general gaze search will be answered. Speech search is clearly more efficient than the search by gaze. One of the main reasons for the usage of immersive VR is to provide a more realistic feeling. It had to be tested if this is the case with the VRProductFinder. However, the study proved that the users had a sense of presence and thus a realistic shopping experience in VR. But the motion sickness was also higher in the VR setup. The question about the best out of the four tasks can be answered by taking all these findings together. It will be shown that speech search in combination with VR proved to be the best.

1.4 Outline

The related work chapter will present some concepts that were used as a base for the VRProductFinder. More precise, it will show some of the current smartphone sensors and explain how they can be used to interact with virtual reality environments. Additionally it discusses some approaches of shopping in virtual environments and how the consequential motion sickness can be reduced.

After that, the main concept of the VRProductFinder will be expounded, which is followed by an explanation of all technical aspects. The conducted study together with its results will be discussed. Due to the sustainability of the application, some of the main points for future work will be presented. Finally, the thesis is summed up in conclusion.

2 Related Work

There are multiple types of virtual reality(VR). Since this thesis uses only two of them, these will be explained in the following. The simplest one is Desktop VR. It uses a desktop computer and a monitor as an output device. Together they display the rendered images. The second is immersive VR. Here the output device is positioned in front of the user's eyes, while it displays stereoscopically rendered images. The viewing direction in the virtual environment(VE) adjusts to the user's head position and orientation.

2.1 Interaction using Smartphone Sensors and Speech Input

Today's smartphones have multiple built-in sensors. An article by Ming Liu[25] showed that the most often used sensors are the accelerometer and the gyroscope. This fact was explained with the high accuracy of these sensors. The accelerometer is used to measure the acceleration relative to the free fall. Unlike the gyroscope, which is able to measure the absolute device orientation[25]. By fusioning the data of both sensors it's possible to track the current orientation of the device quite accurate, with a small error rate. That is the reason why this thesis also uses these two sensors for the interaction with the smartphone.

Head-mounted devices(HMD) are worn like a helmet on the head. In the front, they contain a display. These devices are used for immersive virtual reality. Google Cardboard is an HMD created by Google[16]. A smartphone is put into the front side of the Cardboard and serves as a display. Since the Google Cardboard is made of cheap materials and most people possess a smartphone it is perfect for small purses. With the aid of the sensors mentioned above, the head rotation of the user can be measured in the real world and translated into a virtual world, which gives the user three degrees of freedom(DOF)[31]. The study that was conducted at the end of this thesis used VR glasses that were inspired by Google Cardboard.

Concerning the explanation on how this hardware can be used to interact with a virtual environment, Soojeong et al.[46] presented some controller-less interaction possibilities. Gaze is one of the simplest ones and comes with two types. Instant Gaze means that the action of the gazed object is triggered directly by looking at it. In contrast when using Dwelling Gaze the user has to look for a predefined interval on the object to trigger its action. While Instant Gaze has the "Midas Touch Problem"[20], Dwelling Gaze prevents it, which was the reason for using it. Tilt is another interaction type, but since the VR-ProductFinder doesn't provide a scenario where it can be useful, it wasn't implemented. A further method is a Magnetic switch, which was also presented by Smus et al.[39]. It is a small magnet located on the outside of the cardboard that makes use of the magnetometer, which measures the strength of magnetic fields around the smartphone[39]. The magnetometer can record the movement of the magnet and is therefore used to interact with the smartphone. One reason why the VRProductFinder does not use this interaction method is that not every HMD has this magnet. So a usage of it would reduce the compatibility of the application. Furthermore it is not hands-free, which is one of the big advantages of the VRProductFinder.

Sharma et al.[36] pointed out that the smartphone's internal sensors only provide limited interaction possibilities. For example, it is not possible to capture the real world movement of the user and map it to the VE. This limitation cannot be avoided, but it is possible to use an external device to make the movement easier for the user by providing further DOF's. There are some approaches to realize this. Soojeong et al.[46] presented a further interaction type that uses such an external device, which is connected to the smartphone via Bluetooth. Papaefthymiou et al.[31] used a further smartphone to interact with the scene. This smartphone is held by the user, who can use it as a button or joystick. Additionally, they used the built-in step sensor to move forward. Pointing is possible with this application, too. Steed et al.[40] also proposed to use a second smartphone. But with their application the user can use the second device to move a ray in the scene to point on objects with. At first glance, the usage of other devices is a good idea, but the application is not hands-free anymore by using them. Besides it cannot be used anywhere because the user has to carry an additional bulky device, while the first smartphone is usually available and the HMD lightweight. To provide a simple usage of the VRProductFinder, it does not use other devices.

Most of the smartphones have a built-in microphone[25]. It can also be used to interact with the VE by using speech input. Speech input is a very comfortable input modality because it is natural. Google has been researching for a long time in the field of speech recognition. With the increasing amount of available data and computational resources, it becomes more and more accurate[35, 34]. The Web Speech API is a speech recognition framework for the browser which uses Google's speech recognition web service. Adorf et al.[1] used the Web Speech API to show that an accurate recognition at word level is possible. The accuracy and browser compatibility are reasons for using the Web Speech API as an interaction type for the VRProductFinder.

2.2 Shopping in Virtual Environments

A lot of big online shops exist, and many people use them every day. One of the most famous is Amazon[3]. Their problem is that they are not very realistic and thus do not provide a high sense of presence(see Figure 1). 3D shops in virtual environments change that. Buffa et al.[5] stated some advantages of 3D stores. Customers can profit from daily opened warehouses, timesaving shopping and much information on the products. Traders have to spend less money and get a broad audience. It is easier to gather data about the customers to know their needs. There are already some approaches for 3D shops. Sanna et al.[33] implemented a dynamically generated 3D shop. The users can choose preferred products, and the 3D shop is generated out of them. It is possible to move around in this shop, too. Robles et al.[32] also implemented a 3D shop that is adapted to the user's preferences by predefined rules. The store manager defines the adaption rules. Roberto et al.[7] proposed AWE3D, an architecture to built adaptive 3D websites. They used this approach to build a dynamic 3D shop. Another dynamically created 3D store was ADVIRT, which was implemented by Chittaro et al.[9]. Here the store layout, organization, and look adapts to the personalization rules of the user. All these 3D shops have their focus on adaptability. The problem is that they are only available for Desktop VR. The technology to develop and use e-commerce in immersive VR already exists, as Khandewal et al.[22] explained. So, the VRProductFinder captures the concept of completely dynamic 3D shops and extends the existing ideas by immersive VR as output type.

Olga De Troyer et al.[11] implemented the Shop-WISE, which is an individual generated virtual 3D shop. In their shop, the user can even pick up 3D objects to inspect them. Their product list is maintained separately and thus can be altered very easily. This idea inspired the VRProductFinder to read in the product list dynamically to provide an easy maintenance of the products. Another interesting aspect about Shop-WISE is the search function. The user can search for a product by text input and is brought to the desired product after it has been selected from a results list. The fact that the product has a static location makes the shopping experience more realistic. The problem about Shop-WISE's search is the presentation of the results. The user has to choose a product from a simple text list without even knowing if the product is the desired one. Here the approach proposed by Luca et al.[8] helped. They had the idea of "walking products", where animated products move through the shop going to their shelves(see figure 2). So the user just has to follow the product to get to its location or a shelf with more assortment. These two search approaches inspired the search of the VRProductFinder. Already during the search process the user can see 3D representations of the search results to get a better knowledge about it. Then the user is brought to the shelf of the desired product.

Figure 2: Walking products[8]

Some factors affect the buyer's behavior. Häubl et al.[19] argued that users tend to buy a product if they see a 3D representation of it. A good navigation can also help. "Massification" is another affecting point, which was introduced by Chittaro et al.[8]. If a shop has more than one variety of a product in the shelf, it appears richer, which affects the user positively. But it only takes effect if the products are ordinary. For exclusive products like paintings a massification would have an adverse effect, because the products are not exclusive anymore. The VRProductFinder uses 3D objects but not massification because too many products on the shelves make the market appear untidy.

Too large scenes can be problematic to render, especially for mobile devices. Mass et al.[26] proposed dividing such scenes into disjoint closed areas. The objects behind a door are preloaded just as the user comes closer to provide a continuous navigation through the areas. Distant objects are disabled. Thanks to the dynamic structure of the VRProductFinder it is possible to divide the shops into multiple areas. The products of the different areas can then be enabled and disabled during the runtime.

2.3 Reducing Motion Sickness in Virtual Environments

One of the biggest problems when using immersive virtual reality with an HMD is motion sickness. That means that the user develops some symptoms triggered by the virtual environment while being exposed to it or even afterward. These symptoms can be nausea, eye strain, headaches, difficulty focussing and blurred vision. A study conducted by Sharples et al.[37] proved that the symptoms are higher when using immersive VR. In contrast, the symptoms in Desktop VR are negligible.

By placing a fixed object within the field of view of the user, it is possible to reduce motion sickness. Whittinghill et al.[44] used a virtual nose for this purpose(see Figure 3). The users did not perceive the virtual nose, but a conducted study showed that the presence of the virtual nose reduces motion sickness. The VRProductFinder has such a fixed object also in the form of a nose in the VR setup.

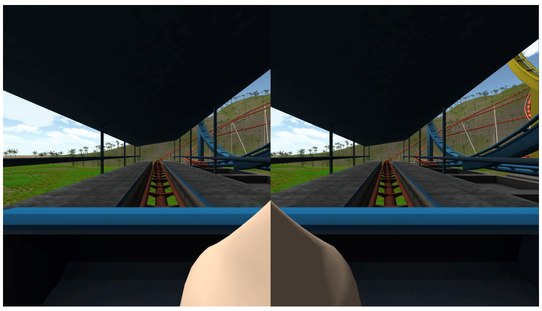


Figure 3: A nose as fixed object in a stereoscopic rendered image[44].

One reason for motion sickness is the velocity in the VE. Mourant et al.[28] showed that a high velocity could amplify the symptoms. Users that drive in a virtual environment on a highway develop stronger symptoms than users in the city. Tanaka et al.[41] found the same effect. Another important factor for motion sickness is the field of view(FOV). Seay et al. conducted a driving simulator study and found that a larger FOV gives the user a higher feeling of presence, but also motion sickness. They described a large FOV as a "double-edged sword". Lin et al.[24] also came to the result that motion sickness and feeling of presence are positively correlated. Furthermore, they found a negative correlation between motion sickness and enjoyment. So a trade-off between both has to be found, which was presented by Ajoy et al.[14]. They proposed a dynamically changing FOV(see Figure 4). A conducted study showed that this approach helps to keep the sense of presence while it reduces the symptoms of motion sickness. To reduce the motion sickness when using the VRProductFinder, both the default velocity and the FOV are set to a medium value so that it does not cause much motion sickness, but provides a good feeling of presence. Additionally, both values can be dynamically adjusted with parameters and thus adapted to every single user. For users who are susceptible to motion sickness, this helps to provide a higher enjoyment. By contrast, it helps users who do not develop motion sickness to get a greater feeling of presence.

Figure 4: The unchanged and the dynamically reduced FOV[14].

3 Concept

The main concept of this thesis is the VRProductFinder, which represents a scene of a virtual market and is strongly based on ThreeJS. Within this scene, the user can move around in this virtual market. The market is dynamically generated from JSON files, which are based on the data of a real market. On top of shelves, there are 3-dimensional versions of real products. The user can interact with the objects in the scene with simple gaze gestures and has the possibility to have a closer look at the products and its information just like in a real market. Desired products can be put into a cart by the user to be bought later. To make the search in the market more efficient and comfortable, the VRProductFinder provides a speech search.

3.1 Design

The design of the scene was kept as simple as possible but still comes close to the one of a real market. While in most of the real markets the shelves are overfilled with products and the market is decorated with advertising wherever the customer looks, the VRProductFinder only contains a small number of products and other objects. The reason behind this is that the user should orient himself as easily as possible in the market to find products pointedly. There were just a few decoration elements added like the logo on the long walls to make the market look more realistic, but these were also kept subtle and should not disturb the user.

The colors were chosen using the model of real markets, which means that the colors of elements such as the floor and shelves have dark tones of gray. By contrast, the walls and the ceiling have a light color, so that the market appears bigger and brighter. Still, all these elements have real textures to make the virtual scene look realistic. The attention of the user should always be attracted to the products. Therefore the products are colorful and natural, unlike the dark elements they are surrounded by so that a contrast arises and the products themselves catch the user's eye. The contrast is emphasized by a subtle glossy reflection of the products.

3.2 Elements

The scene consists of different types of elements, which are created by parsing the information from several JSON files. The functionality of the various elements, as well as the interaction with them, will be presented and explained in the following.

Shelves

Shelves are holding the different products, which are evenly distributed on it(see figure 5). Every shelf has to have at least one product on it. If not it is disabled. The space between the single boards is computed by the market height and the number of boards.

Figure 5: A Shelf with products

Product

A product represents a virtual 3D model of a real product. These products are placed on the different shelves. A product can be added to the cart if the user wants to reserve this product to buy it later. The user can interact with the products by the gazing gesture. By gazing on a product from a shelf for a long time, the product view is enabled(see figure 6). That means that the product is moved from the shelf towards the user. Simultaneously a panel is faded in next to the product. It displays information about the product, more precisely its name and price. At the bottom of the panel, there are buttons with different actions. A button can be pressed by gazing on it. The button with the cross symbol quits the product view by moving the product back to the shelf and hiding the panel. To add the product to the cart once or twice the user has to look at one of the buttons with the tick symbols on them respectively.

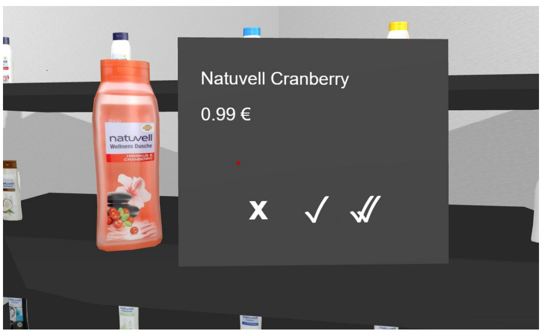


Figure 6: A product with its information panel.

Cart

The cart is always located in front of the user(see figure 7). It is holding all the desired products. Products can be put into the car by using the Product View. Added products can be removed from the cart by gazing at them. The panel at the back side of the cart contains several buttons. The button with the bubble symbol moves the user to the panel for the speech search. By gazing at the button with the speaker symbol, background music is played. The button with the door symbol resets the whole scene and sets the user back to the start position. On the left side of the panel, one can see how many products there already are in the cart and the price for all of them in total.

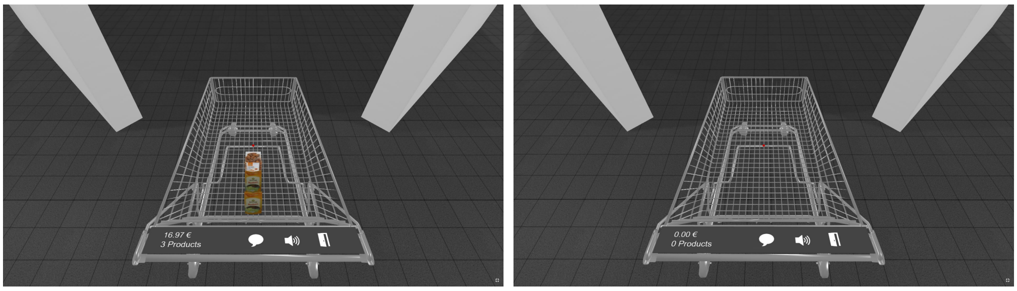


Figure 7: A cart with products(left) and without(right).

Search Interface

The search interface is localized on a wall in the scene(see figure 8). It manages the speech input and provides the functionality to search for products. On the idle state of the panel, the user can start the speech input by gazing at the button with the bubble symbol. After the search query is spoken into the microphone, the input will be displayed together with the found hits. Now the user can gaze at the desired product from the result list and thus will be moved to the shelf with this product by the defined move type. When the user is looking at a result, the front view of the respective product is displayed in the middle of the search interface. This helps to decide if this is the desired product.

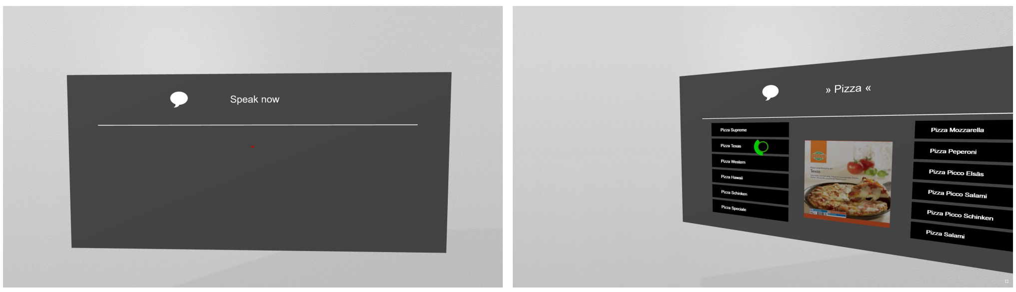


Figure 8: The Search Interface before(left) and after a search(right).

Move Plates

The Move Plates are spread over the whole market(see figure 9). They provide the functionality to move the player to their location. Only the Move Plates that are close to the user's position are displayed, while the ones that are too far away are hidden. For every position the user can occupy, it is ensured that there are always two or more Move Plates in range so that the player can move on and is never stuck in one position.

Now the user just has to gaze for a while on a nearby Move Plate to be moved to it by the defined move type. As soon as the final position is reached, the Move Plates that are too far away or below the user are faded out. The ones that are now in the predefined range are faded in.

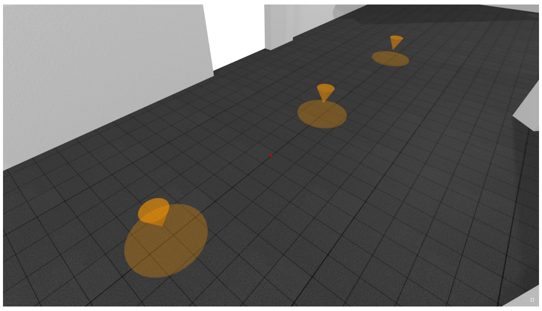


Figure 9: A row of Move Plates.

Nose

The nose is located in the middle of the visual field(see figure 10). It is supposed to simulate the real nose of the user and provides a familiar feeling. Additionally, it helps to reduce the motion sickness[44]. Since the use of a virtual nose just makes sense in immersive virtual environments, it is only enabled in the stereoscopic view.

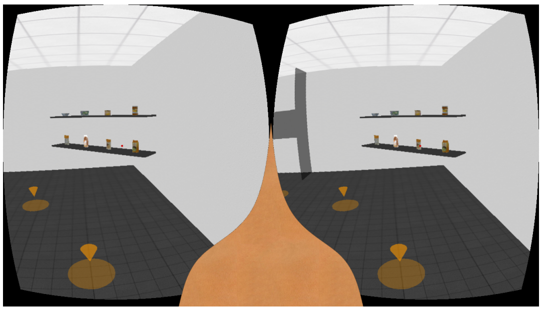


Figure 10: A stereoscopic view with the virtual nose.

3.3 Product Search

There are two possibilities to search for a desired product in the VRProductFinder. Both will be explained in the following.

Speech Input

The first search type is the search by speech input. The procedure of this search type will be explained in the following(see figure 11).

First, the user has to move to the search interface, which can be done by using the Move Plates or simply by gazing at the corresponding button on the cart panel. When standing in front of the search interface, the speech input can be activated by gazing at the bubble symbol on the search interface. Now the user has to say the name of the desired product or just a part of it. After the query has been recognized, all names of available products will be searched for the query. If there are any hits the 12 best results will be displayed on the search interface. In case that the desired product is among them, the user can gaze on it to be moved to the shelf of this product by the defined move type. Additionally, the product view for the chosen product is activated, so that the user does not have to search it in the shelf, which is convenient especially when all products on the shelf are very similar. The advantage of this search type is the speed. Thanks to the speech search the input of the search query is fast.

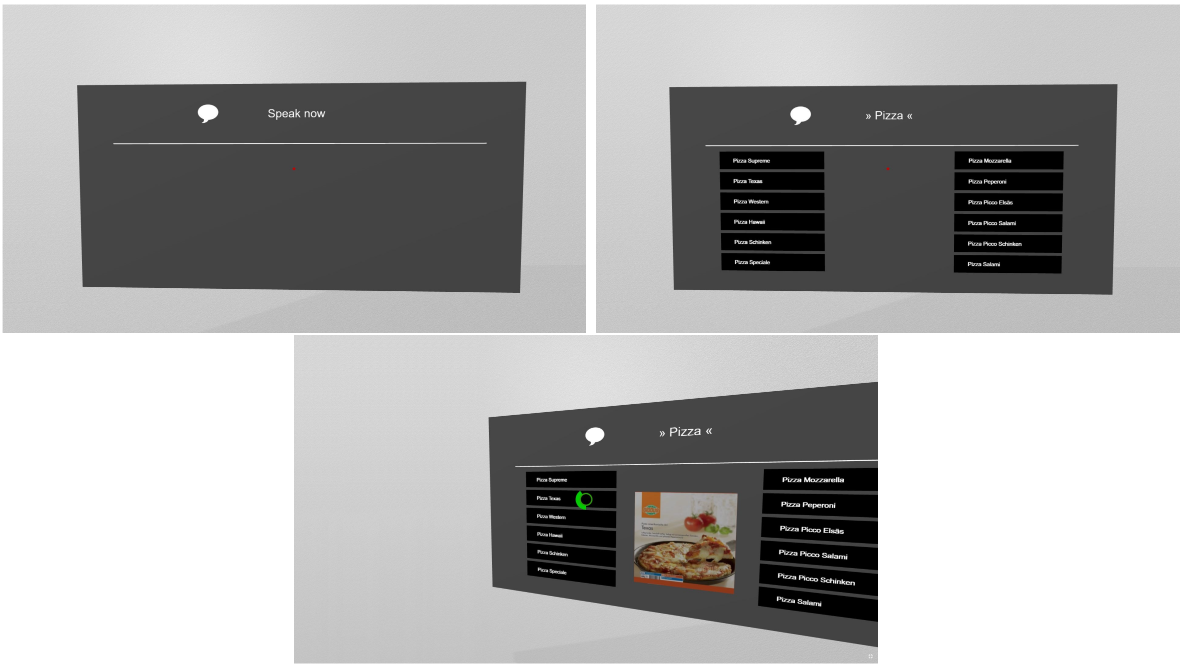


Figure 11: Example for the procedure of a search by speech input.

Move around

Another possibility to search for a product is moving around in the scene and look for it. The procedure of the search type is relatively straightforward(see figure 12). The user moves around in the virtual space by using the Move Plates looking around to find the product by checking all the products on the shelves.

This search type has the disadvantage that in the worst case scenario the user has to move through the whole virtual environment until the desired product is found. Thus it can be more inefficient than the search by speech input.

The advantage is that the user can learn the layout of the shop by moving around. Once the structure is known, it is easy to find products by just moving towards them. But it can still take some time until the desired product is reached even if the location is known.

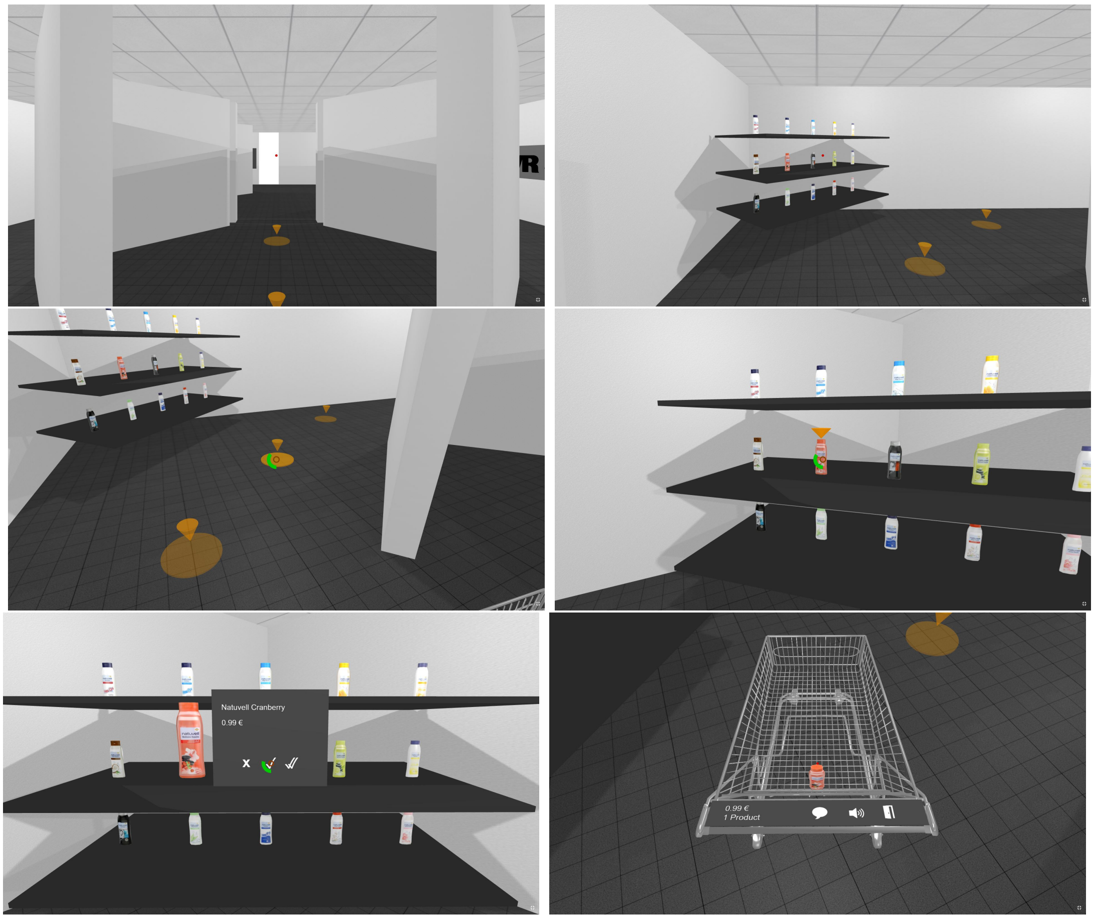


Figure 12: Example for a search by move around.

3.4 Movement Types

The player has two possibilities to move through the scene. By default, the path moving is selected, but it is possible to choose the clipping moving by setting the respective parameter. Both movement types will be explained in the following.

Path

By default, the user is moved over a minimal path to the target position, which is computed over the positions of the Move Plates(see figure 13). This kind of movement has the advantage that the user is moved in a natural way that is known from the real world. But it is not very efficient because the user has to walk slowly over a path that could be very long.

The path from the current position to the target position is computed by a pathfinding algorithm, which is called Dijkstra algorithm. The Dijkstra algorithm calculates a minimal path between two points in a two-dimensional graph, which is generated by adding every Move Plate as a node in the graph, while the connections between the Move Plates represent the edges. Although the scene is three-dimensional, it is enough to compute a two-dimensional path because the height of the user is never changed so that it can be discarded in this case.

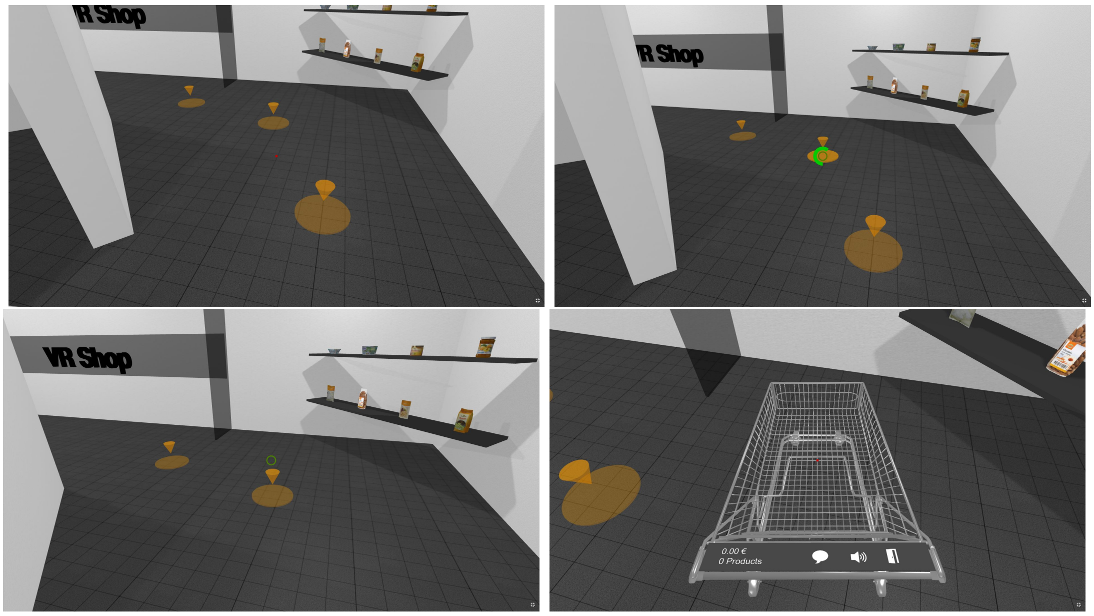


Figure 13: Example for a movement by path.

Clipping

Another moving type is the clipping teleportation(see figure 14). It works in the following way. There are two black bars outside of the visual field - one at the top and one at the bottom. After the user has selected the desired product, both bars are faded in, by being moved to the middle of the screen. Now that a black screen is displayed the current position is switched to the target position. After that, the two bars are moved out again from the visual field.

This movement type has the advantage that it is fast and gets the user straight to the target position. The duration of the teleportation is always the same, independent of the distance between the two points. But it can be problematic because the environment is switched very quickly.

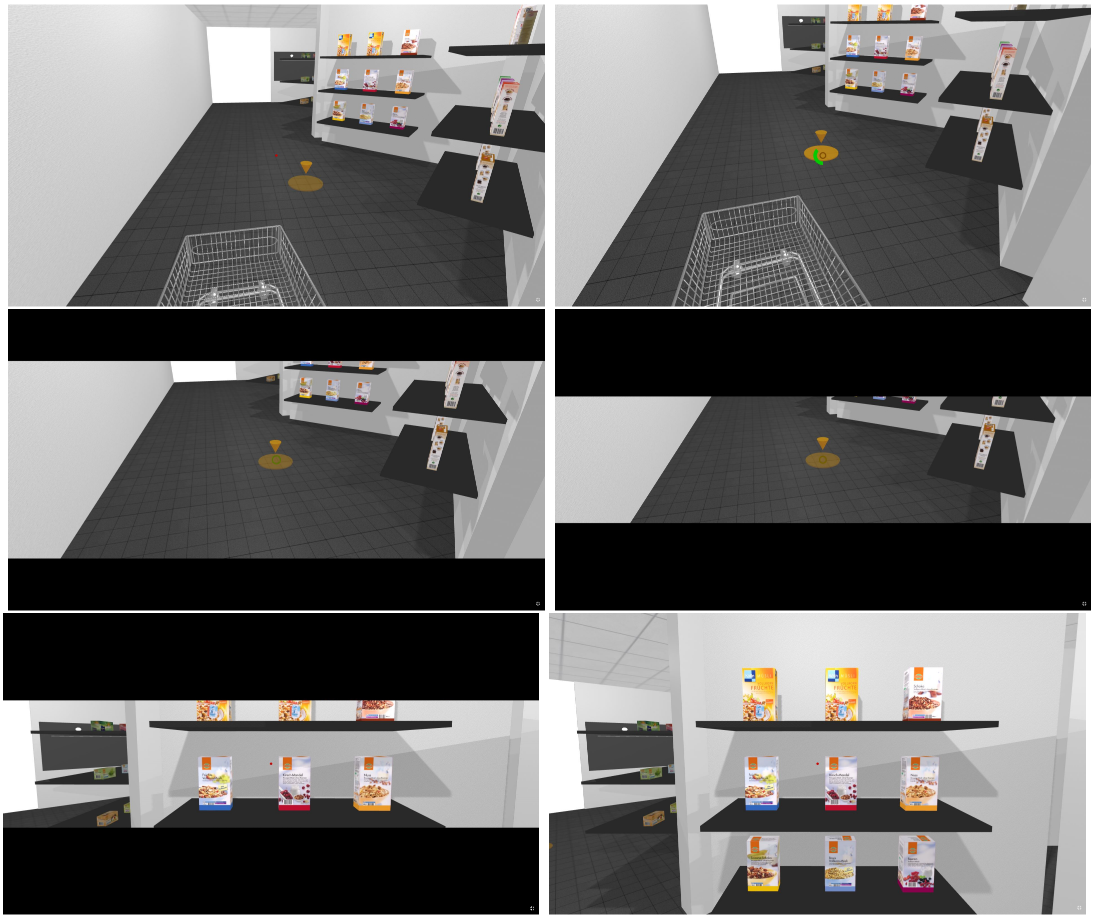


Figure 14: Example for a movement by path.

3.5 Highlight types

There are two types of highlighting products. By using the first one, a small reversed cone is placed over the product the user is currently gazing at. The second one highlights the product by fading out all other products. A comparison of both is shown in figure 15.

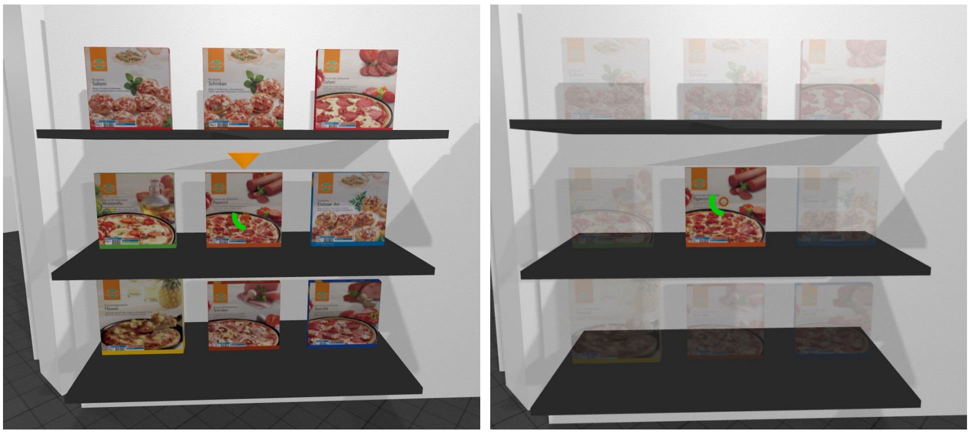


Figure 15: Reversed cone(left) and fading(right).

3.6 Interactions

As was being shown in the related work section, there are not many interaction types with virtual environments in combination with mobile devices and card boards. This thesis implements two of them that are used to manipulate the scene. They will be explained in the following.

Head Tracking

By using head tracking the users head rotation of the real world is mapped to the virtual environment, which makes it possible to interact with objects in the scene by just gazing at them. This interaction type is easy to use and makes the use of the virtual reality more immersive.

The disadvantage of this interaction type is a possible motion sickness, which can make it unpleasant for the user to use it. Fast motions or motions that are not synchronous with the movements in the real world can cause it.

Speech Input

The speech input is also a compelling option to interact with the scene. It can be employed to search for a particular product with the aid of the speech search. This interaction type has the advantage that it is a natural way of interaction and requires no additional input devices to use it. All smartphones have already a built-in microphone, and the user can simply use it. Another advantage is that saying the name of the product is fast. The difficulty is to recognize the spoken content of the user correctly.

3.7 Output Devices

There are three output devices that a user can claim to utilize the virtual reality mall. The WebVR framework automatically detects which output devices are currently available and displays the rendered scene to it. If an Oculus Rift is attached, apparently a computer monitor is also connected to the system. First WebVR shows the scene to the monitor. The user has the possibility to render the scene to the Oculus Rift by activating the stereoscopic view, which can be done by clicking on the symbol on the right lower corner of the monitor. On mobile devices, this symbol can also be used to start the stereoscopic rendering. The different output devices will be explained in the following.

Computer Monitor

One way to use the VRProductFinder is to view it via a computer and an attached monitor. The user can use a mouse in combination with a keyboard as input devices to interact with the scene, which is an easy way to use it, but it comes with some disadvantages. Of course, the render quality is good because a computer is powerful enough for this purpose. But the user is not able to have an immersive virtual reality experience.

Mobile Device

Another way is to use the mobile device in combination with a cardboard. They are put on the head of the user. Now the gazing gesture can be used as interaction type to manipulate the scene, which replaces the mouse/keyboard input of the computer. Smartphones have some advantages. The most important one is handiness because the smartphone does not need a tethered connection, so it can be used nearly everywhere without much effort. The problem is that a mobile device is not very powerful, so the rendered scene does not look as good as the same one rendered on a computer.

Oculus Rift

The last output device is the Oculus Rift. Just like the mobile device the Oculus Rift is put on the user's head. Again the user can utilize the gazing gesture to interact. The advantage of this output type is the accuracy of the built-in sensors. They are more accurate than the ones obstructed in the smartphones. Another advantage is the output quality of the displays, which have a repetition rate of 90Hz and an overall resolution of 2160x1200 pixels. The render quality is also good because the scene is rendered on a computer. Thus rendered images look way better than the ones one smartphones. The Oculus Rift also has some disadvantages. The most important one is clearly the need of a powerful computer. Another one is the unhandiness because of the tethered connection to the computer.

3.8 Website

A website was implemented to view the log files in an easily readable format. It also contains some other information about the VRProductFinder, but it is particularly useful for the logs. They are stored in a MySQL database and thus not comfortable to study. The user's ID must be chosen from a list to see the log. Then all interactions done with the scene by this user will be displayed. Additionally, it is possible to see the time elapsed between two interactions by clicking on them in sequence.

3.9 Dynamic scene

The whole scene is constructed dynamically by parsing three JSON files that contain all needed information about the different elements.

- The first JSON file is for the shelves and contains information about the size and the location of the different shelves as well as the products they are holding. Every product has its own information about the 3D files and textures as well as general properties like pricing and naming.

- The second JSON file is responsible for the positions of all Move Plates as well as the connections between them.

- The third JSON file contains information about building elements like floor, ceiling, and walls. It consists of their size, location, and textures.

4 VRProductFinder

In this section, the technical parts of the VRProductFinder will be discussed. After that, all external frameworks that were used to implement it will be listed. The VRProductFinder has a variety of advantages compared with real or typical online shops. The most significant of them will be presented at the end along with some limitations.

4.1 Architecture

The architecture of the VRProductFinder requires a client-side and a server-side. The server is the place where the VRProductFinder is installed. Besides a running PHP-Compiler, it also needs access to a MySQL database. Obviously, the server has to be accessible from outside its local network.

The connection between server and client has to be secured over HTTPS. This is necessary to be able to use the speech input correctly. If the connection is secured over HTTPS Google Chrome asks once for the permission to use the microphone as soon as the user uses the speech input. Without a secured connection, Google Chrome would ask for the permission every time the user starts the speech input and thus make the speech input nearly impossible to use.

The dynamically constructed virtual scene is constructed on the server, which is the main reason why HTML alone is not enough and PHP has to be used. Another reason is the adaptability to mobile devices. The server recognizes if the client is a mobile device and adjusts the parameters so that the scene can be rendered on these devices, which have a much lower performance than current computers. Once the server has generated and composed the content, it uploads it to the client. The main part of the server is done and the VRProductFinder runs on the client via javascript.

The MySQL database is needed to store specific data. While the user interacts with the VRProductFinder, it constantly logs the interactions of the user and sends them to the server. These logs are stored in the MySQL database and can be consulted on a separate website.

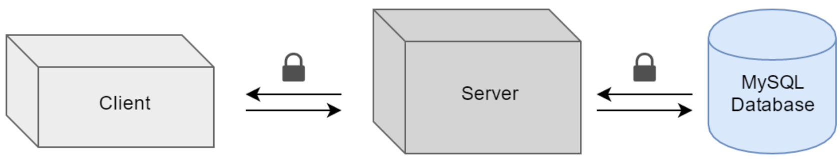


Figure 16: The architecture of the VRProductFinder.

4.2 Technical Requirements

Hardware

The VRProductFinder supports several output devices. The user can access it via a cardboard along with a mobile device. By using a computer, the VRProductFinder can be viewed in combination with an attached Oculus Rift or simply with mouse and keyboard over the monitor.

When using a computer, there has to be a microphone connected to the system to use the speech input. The mobile device should also have a built-in microphone, but this is almost always given. Independent on the device the user always has to grant the system rights to use the microphone once.

The mobile device needs a built-in gyroscope and an accelerometer. These sensors are required so that the javascript library can track the movement of the head. By using mouse and keyboard as input, the computer can not track head position and movement and thus does not need these sensors. The Oculus Rift has already built in the required sensors.

Software

To access the virtual scene from a computer with mouse and keyboard input the user has to use a full version of Google Chrome as browser, because it is the only browser that supports all the external frameworks. Other browsers can be used to move around and do most of the other things in the scene, but the speech input is not supported. It is possible to use either Windows or MacOS as operating system, because they both have a full version of Chrome.

For using the Oculus Rift as output device obviously the required drivers of the Oculus Rift have to be installed on the system used. Additionally, it is not enough to use the full version of Google Chrome. At the moment it has to be the latest unofficial version of Chromium in order to support the transmission to the Oculus Rift. Unfortunately this method only works on Windows computers due to lack of missing Oculus Rift support on Mac computers.

The mobile device must run with Android as operating system. It would also run on an iOS mobile device, but there is no browser available for these devices that supports the speech input. All other functionalities can be used without problems.

4.3 External Frameworks

ThreeJS

ThreeJS is a lightweight 3D javascript library for rendering and displaying 3D scenes in the browser[42, 45]. It is the core framework which was used to build the virtual scene. ThreeJS provides a renderer with all standard components such as different shaders and materials. Many extensions can be included to the renderer, which is the main advantage because due to its modularity threeJS can be extended very easily by external frameworks. It also works on all modern browsers without any additional plugins, because it is based on WebGL.

WebVR

WebVR is a javascript API that makes the usage of virtual reality with threeJS easier[30]. It works as a wrapper around the standard threeJS renderer. It detects which type of output device is available for the current setup and adjusts the rendering options depending on that. WebVR handles the stereoscopic rendering for mobile devices. It also can detect other output devices like Oculus Rift.

TweenJS

TweenJS is an animation manager for threeJS[10]. It handles all the animations and movements in the scene. It animates objects by interpolating between two values for a given variable. The advantage of TweenJS is that it can do this for every given variable of a threeJS object regardless of whether it is a position or not.

Reticulum

Reticulum is a virtual reality gaze interaction manager for threeJS[38]. This library creates the reticle in the middle of the visual field. It fills and colors the reticle depending on the current dwell time. When the user looked onto an object for a specific time interval, Reticulum executes a predefined function. The advantage of this library is that it is possible to define a particular action and dwell time for every object that the user can interact with.

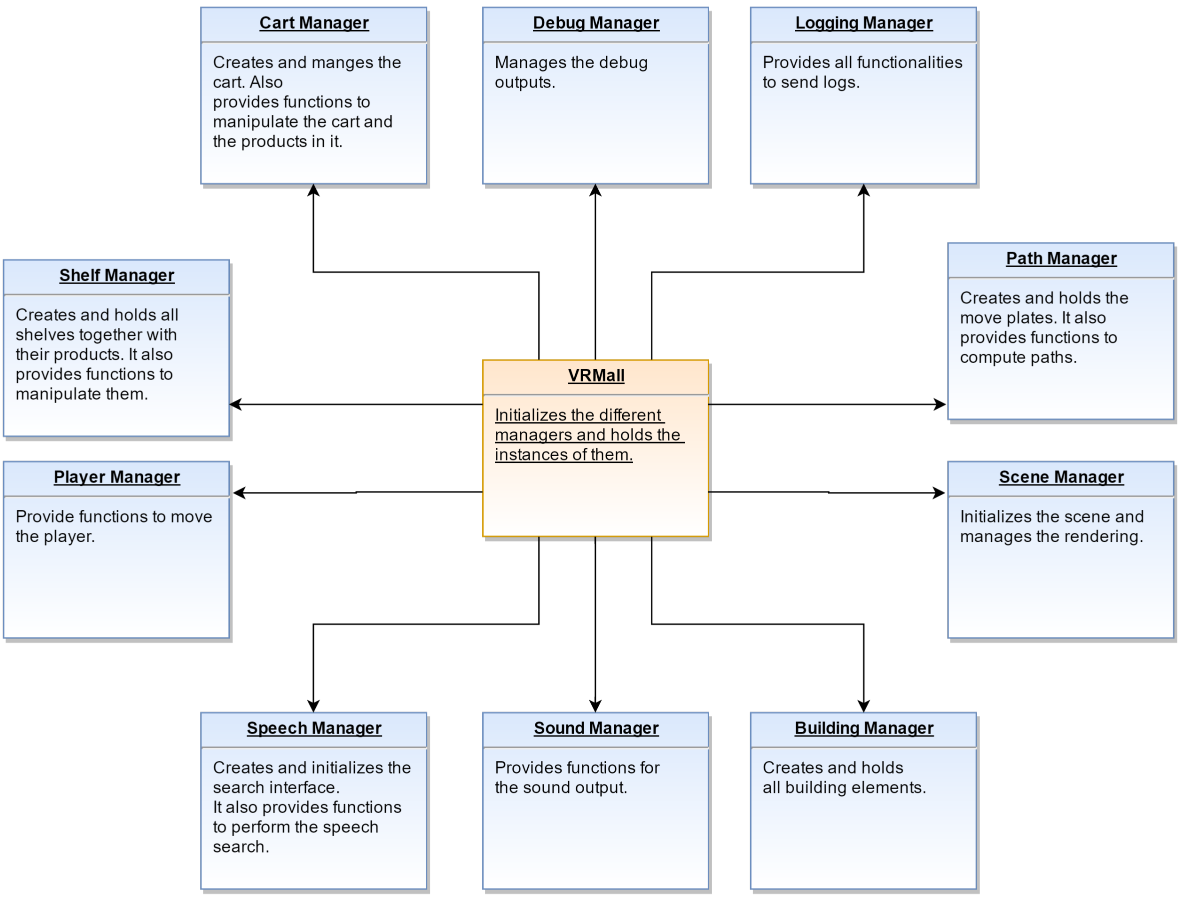
jQuery

The javascript library jQuery was used to handle and manipulate the HTML code of the website[21]. It can be done very easily and more efficient by using jQuery.

4.4 Implementation

The implementation of the VRProductFinder was strongly based on sustainability. It was kept modular so that it can be extended very easily without the need to understand the whole code previously written. It is split into several managers, which gives further developers the possibility to add other managers that are independent of the existing ones. The current managers have public functions, which are self-explaining. They can just be called from new managers without the need to understand what they do internally. Figure 4.4 represent the different managers and explains their main functionalities.

To offer a high compatibility, the whole VRProductFinder was implemented in web-based programming languages. Web browsers are available in most of today's devices. While the server-side is implemented in PHP, Javascript was used for the client-side.



4.5 Parameters

Parameters set the settings of the VRProductFinder. These parameters all have predefined default values. In case that the user wants to change a setting, its value can be overridden by setting it as a GET-parameter. Values that specify if something is enabled or not, need to be committed in the format "true" or "false". The most important parameters together with their default value will all be explained in the following.

guid This is the ID of the current user, which is used for the log files. As the default value, a unique ID is generated every time the scene is loaded.

market The market that should be viewed. The default value is "IRL".

noseEnabled Specifies if the virtual nose is enabled. By default it is.

speechInputEnabled Defines if the speech input is enabled. By default it is.

moveType Defines the move type. 0 is the movement by path, while 1 is the movement by clipping. The default value is 0.

highlightType Defines which highlight type should be used. 0 is the highlighting by fading, while 1 is the highlighting by the cone. The default value is 1.

language This is the language that is used for the texts in the scene. At the moment there are only German and English available, but they can be extended very easily over a language file. The default value is "English".

fieldOfView This is the field of view in degrees. The default value is 60.

startX/startZ These values define the two-dimensional starting point of the user. The default point is (450,900).

minMoveRadius/maxMoveRadius These values define the movement radius of the user. Only Move Plates that are in this radius around the user are enabled. The radius is by default between 100 and 400.

minGazeRadius/maxGazeRadius These values define the gaze radius of the user. The user can only interact with objects that are in this radius around him. Thus the radius should be adjusted to the moving radius. In case that the gaze radius is too low, it can happen that the user cannot interact with Move Plates if they are too far away. The gaze radius is by default between 5 and 400.

moveplateColor This value defines the color of the Move Plates in hex format. By default it is "#FF9900".

decorationColor This value defines the color of the decoration elements in hex format. By default, it is "#000000".

productHighlighterColor This value defines the color of the cone that is used for the product highlighting, in hex format. By default it is "#FF9900".

shelfColor This value defines the color of the shelf boards in hex format. By default, it is "#333333".

ambientLightColor The value defines the color of the ambient light in the scene in hex format. By default, it is "#cccccc".

moveSpeed This value defines the moving speed of the user when moving through the scene by path movement. It should not be 0. By default it is 10.

dwellTime This value defined the dwell time for the interaction by gazing in seconds. The default value is 2.

antialias This value specifies if anti-aliasing is enabled. It makes sense to disable it for devices with low performance. By default, it is disabled on mobile devices and enabled on all others.

spotLightsEnabled The value defines if the spotlights in the scene are enabled. They should be disabled on devices with low performance. Respectively they are disabled on mobile devices and enabled on all others.

shadowsEnabled Specifies, if shadows should be rendered. They should not be rendered on low-performance devices and in case that the spotlights are not enabled. By default, shadows are disabled for mobile devices and enabled for all others.

resolutionReduction By adjusting this value, the resolution of the scene can be reduced. The value can be between 0 and 100, while 0 means that the scene is rendered in the device resolution and 100 means that the resolution is halved. By default, this value is 0 and should only be used if the frames per second are extremely low.

resolutionIncrease By adjusting this value, the resolution of the scene can be increased. The value can be between 0 and 100, while 0 means that the scene is rendered in the device resolution and 100 means that the resolution is doubled. On high-performance devices, the use of the feature can improve the render quality. The default value for mobile devices is 0 and 100 for computers.

lowTextures Specifies if textures with a low resolution should be used. This improves the frames per second on low-performance devices. By default, low textures are never used.

4.6 Advantages

An advantage of the VRProductFinder is that it is easy to use with multiple types of input devices. For that reason, the user has the possibility to access it from nearly everywhere with an Oculus Rift or a cardboard together with a mobile device. Alternatively, a computer can be used when none of the other devices is in range.

An advantage over the known way of online shopping is that the VRProductFinder is copied from a real market and therefore it provides a more realistic possibility of online shopping. It makes the user feel familiar with a market that is already known from the real world. It is not necessary to use the traditional online shops if it is not possible to leave the house. People can just sit on the couch and use one of the numerous usable devices to dive into a realistic shopping experience.

The VRProductFinder is a dynamically created virtual environment, which offers a huge number of possibilities. For example, the user can visit various realistic shops without even doing a real step. This customizability of the virtual scene makes nearly everything possible. The shops can be personalized for the user to make them more attractive. Every user has own interests and thus it is almost impossible to provide only products in a real shop that are interesting for everyone. These uninteresting products bore users in real markets. With the dynamically VRProductFinder this belongs to the past. The design can also be adjusted to the taste of the individual users.

Searching in real shops can be frustrating just when they are too huge and overloaded with products. To search for a product, the customer has to run around and look for it. With the VRProductFinder users can just query for a product by putting in its name with the possibility of the simple speech input, which makes the shopping more efficient and comfortable compared to real shops.

4.7 Limitations

Mouse and Keyboard as input

If the user uses just the computer with mouse and keyboard as an input device, the head tracking is not possible. So the user has to change the viewing direction by dragging it around with the mouse. Head tracking is not possible here because there are no sensors that can track the current position or movement of the head. So manipulating the virtual environment is not very convenient for the user.

Speech Input

The speech input uses the Web Speech API from Google[1, 18]. This API is supported on desktop computers by Google Chrome from version 33 but is not available for Firefox, Safari, Internet Explorer and Opera. That means that the speech input can only be used on desktop computers in combination with Chrome. The upcoming version 49 of Firefox is said to support the API and therefore the speech input[29].

The support for mobile devices is limited. For the reason that Google Chrome for iOS is not compatible with the API, the speech input cannot be used on iOS devices. By contrast, the Android version of Google Chrome does support the API. Therefore it is possible to use the speech input only on Android devices via Google Chrome.

Orientation changing

After porting the user to a queried product, it helps a lot if the queried product is in the field of view. To achieve this the orientation has to be changed by a script. On devices that provide the orientation by sensors, like the Oculus Rift or mobile devices, it is not possible to override it manually, because it is prescribed. On a computer with mouse and keyboard as input the overriding of the orientation is no problem.

Lighting

The virtual scene contains several spotlights. At this moment these spotlights are entirely omitted on mobile devices. Only when using it with a computer, it is possible to enable them. The reason behind this limitation is a performance issue. With every light that is added to the scene, the rendered frames per second decrease on mobile devices. Their performance is not good enough to perform such scenes.

But this limitation is not only true for mobile devices. For a large virtual environment like a mall, even a computer hits the wall because such a scene consists of many more lights than current devices can handle. In that case, an intelligent light management that dynamically disables and enables lights is required.

Shadow Rendering

Due to the limited usage of lights on mobile devices, the shadow rendering with Shadow mapping also becomes a problem. To have shadow maps at all, it is first necessary to have lights in the virtual scene. So shadows on mobile devices are currently not possible until the lighting limitation is not solved for them. On computers, the rendering with Shadow mapping is no problem, because they can render the spotlights and also have enough performance to compute the shadows.

A reasonable possibility to render shadows without having lights in the scene is Screen Space Ambient Occlusion(SSAO). Unfortunately, SSAO is not rendered correctly on mobile devices, but on Desktop it is a cheap way to make the scene look a little bit more realistic. A comparison of the different shadow levels is shown in figure 17.

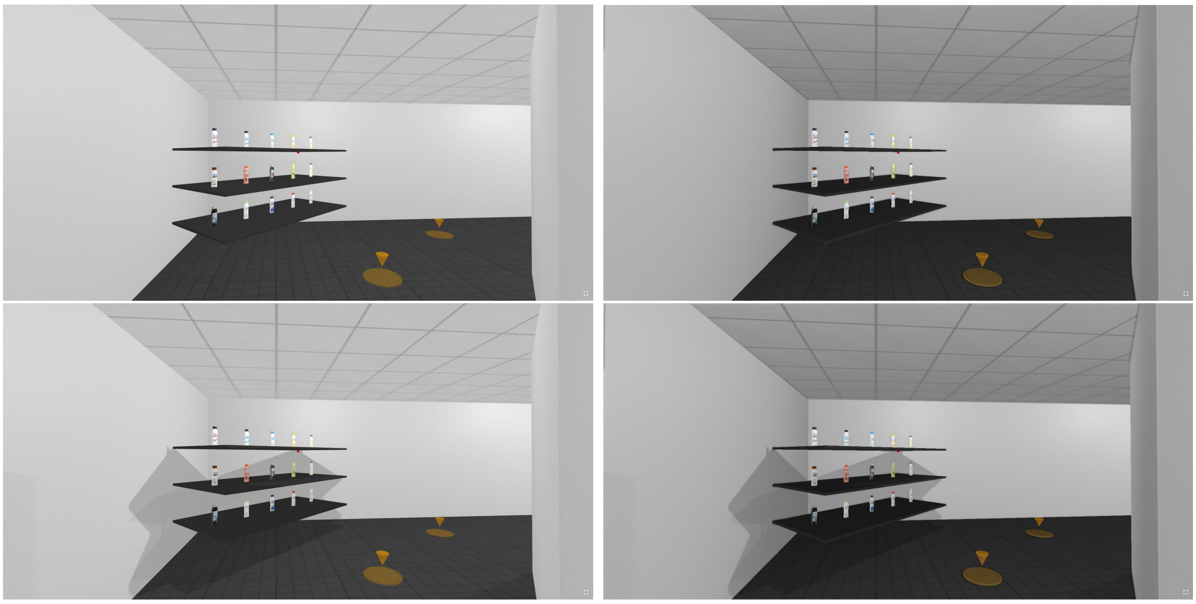


Figure 17: No shadows(upper left). Only SSAO(upper right). Shadow mapping(lower left). Shadow mapping and SSAO(lower right).

5 User Study

A study was conducted to test the usability and efficiency of the VRProductFinder and its interaction types. In this section, the used equipment and design of the study will be explained. At the end, the results are presented and discussed.

5.1 Prestudy

A prestudy was conducted with some people to determine some of the conditions for the study. The prestudy also showed some improvements that were applied before the study was conducted. One of them concerned the speech interface, which initially was dynamically enabled at the user's position. The users complained about the missing realistic aspect about this feature, so the speech interface got a fixed position. The idea of providing the functionality for adding a product twice to the cart also came from a participant.

Since there are two types of movement, one had to be chosen for the study. The participants preferred the path movement, so this one was used for the study. The highlighting by the cone was also preferred by all of the test persons, so it was chosen as the standard highlighting.

5.2 Participants

The study was conducted with 16 test people(13 male, three female), who participated by choice. They were aged between 17 and 55 years. Most of them were students and had very little or no experience with immersive virtual reality applications. The buying behavior of the test persons was widespread. But most of them tend to buy more often in real than in online shops. The average rate of online purchases per month was 2.6. It was significantly higher for offline purchases with a mean rate of 7.4. All participants had at least some experience with the usage of computers.

5.3 Apparatus

For the first task an Alienware[2] computer with an i7 processor and a Nvidia GeForce GTX 980TI graphics card was used. The attached Dell P2210HC[12] monitor had 22 inches with a resolution of 1920x1080 pixels. As input devices, a tethered mouse and a tethered keyboard were used. This system had to be connected to the Internet all the time because the VRProductFinder permanently sent logs to the server and was initially loaded from it. The VRPRoductFinder was displayed with the aid of Google Chrome as browser and Microsoft Windows as operating system. The second task was performed with the same setup as in the first one. Additionally, a Microsoft Kinect 2[27] was attached to the system, because of the built-in microphone.

While the third and fourth task shared the same setup, it will be explained once for both. The mobile device was a Nexus 5X[17] with a display size of 5.2 inches, the Snapdragon 808 as processor and an Adreno 418 graphics unit. This device was put into the Elegiant 3D VR glasses[13]. As sensors, the built-in gyroscope, accelerometer, and microphone of the Nexus were used. This device was just like the computer always connected to the Internet for the same reasons. To display the VRProductFinder on the mobile device, Google Chrome was used as Browser beneath Android 6 as the operating system. The whole setups are shown in Figure 18.

Figure 18: The used setup for the first two tasks(left) and the last two tasks(right).

5.4 Design

The study had a 2 x 2 x 4 within subjects factorial design. The factors were the output device, search type, and queried product. According to this, every participant had to execute 16 single trials, namely four trials per single task. In total 256 trials were performed. The four products were spread over the whole shop, so the difficulty to find them differed for each. While the user moves through the shop and searches for one product the position of all following products should not be learned. It cannot be avoided entirely, but with the aid of the spreading, it can be reduced. To ensure equal conditions to every participant, all trials started at the same position.

The users received only minimal instructions about the functionality of the different interaction types so that no explicit conceptual model was assigned to them. For example, the users were not told that the results of the speech search are found by comparing the input of the user with the names of the products in the shop. So the users had to figure out by themselves how to choose the search query. It helped to find out how users want to use the speech search.

Every trial had to be performed within 120 seconds. Otherwise, the trial was finished and counted as a fail. The average duration of the study for every participant was 60 minutes, including the introduction together with five questionnaires and the performed trials.

5.5 Task

Based on the two different output devices along with two separate search types, the study consisted of four different main tasks which every participant had to perform. The main goal was the same for every task, namely to find four specific products in the virtual environment and put it into the cart.

In the first task, the participants had to use the Desktop computer as output device together with mouse and keyboard as input. They were not allowed to use the speech search, so they had to search the product by moving around in the shop via the Move Plates and look for the desired products. The second task also used the same input and output devices. The microphone was added as an additional input device so that the participants could use the speech search. Here they were not allowed to move over the move plates through the shop anymore. They had to use the speech search to get to the desired product. Figure 19 shows a participant using the setup for these two tasks.

For the third task, the participants had to use the mobile device with the VR glasses as the output device. Like in the first task again they were only allowed to use the move plates to move around in the shop and look out for the searched products. Finally, the fourth task used the same setup as in task 3. But now the users were not allowed to use the Move Plates. They were supposed to utilize the microphone and thus use the speech search to find the desired product. Figure 19 shows a participant using this setup for the last two tasks.

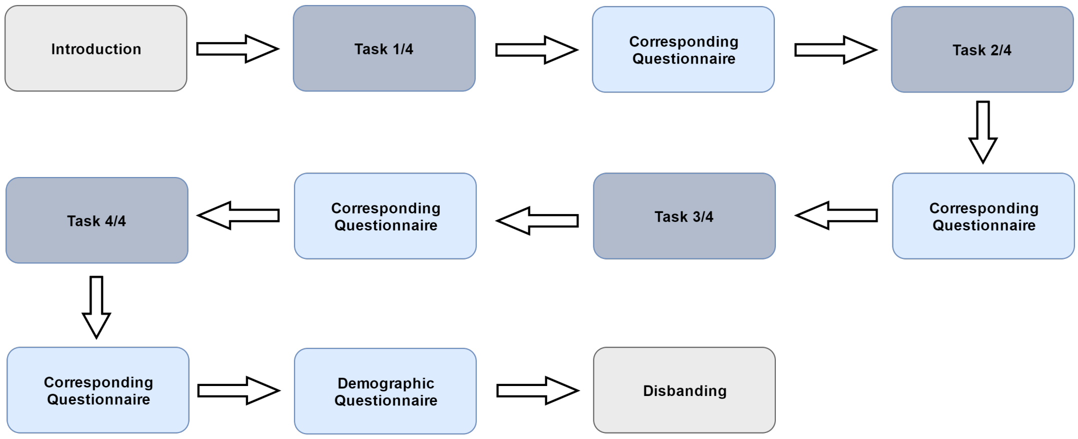
Figure 19: A test person using the computer(left) or VR glasses(right) as output device.

5.6 Procedure

In the beginning, every participant was welcomed. After the general idea of the VRProductFinder had been explained to them, the participants did each of the four tasks once. The order of the tasks was mixed respectively to a Latin Square. Due to the identical basic procedure of every task, the main process will be explained only once in the following.

At the beginning of every task, the examiner explained it to the participant. Then the participant was given a short period of time to get familiar with the techniques that should be used for the particular task. As soon as the participant was ready, the examiner reset the virtual environment, so that the test person was located at the start position. Now the participant had to perform four trials. That means, searching and finding four products in the virtual shop by using the task-specific search type and output device. A trial ended as soon as the participant found the product and put it into the cart. If this procedure could not be completed within a period of 120 seconds, the trials also ended and counted as a fail. After the four trials had been conducted, the participant had to fill out a questionnaire that contained a System Usability Scale[4], a User Experience Questionnaire[23] and a Motion Sickness Assessment Questionnaire[15]. The questionnaires for tasks which were performed with the VR glasses additionally contained a Presence Questionnaire[43].

In the end, every participant had performed 16 trials. A final questionnaire was then filled out. This questionnaire included demographic questions and some others on the opinion about the VRProductFinder. Finally, the participant was disbanded.



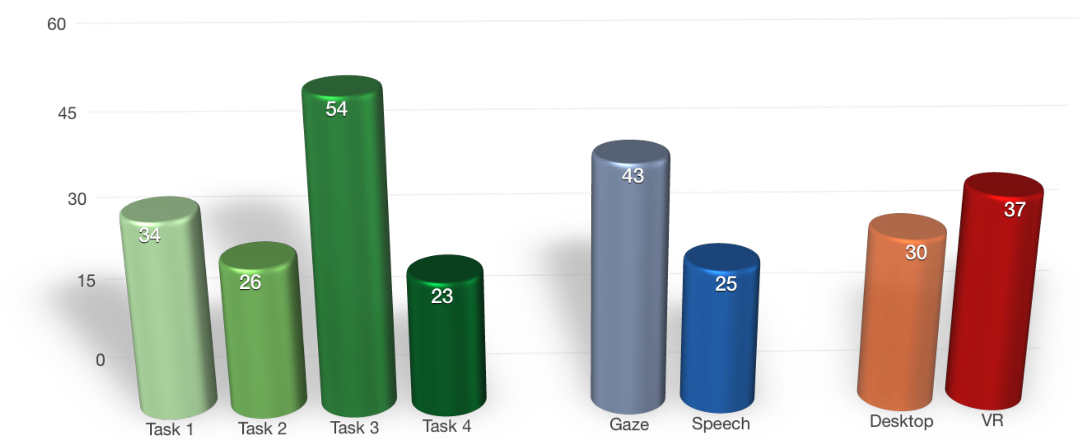
5.7 Results

In the following, the results gathered by the study will be shown. For practical reasons the task numbers defined in the Task section will be used further on.

Task Completion Time

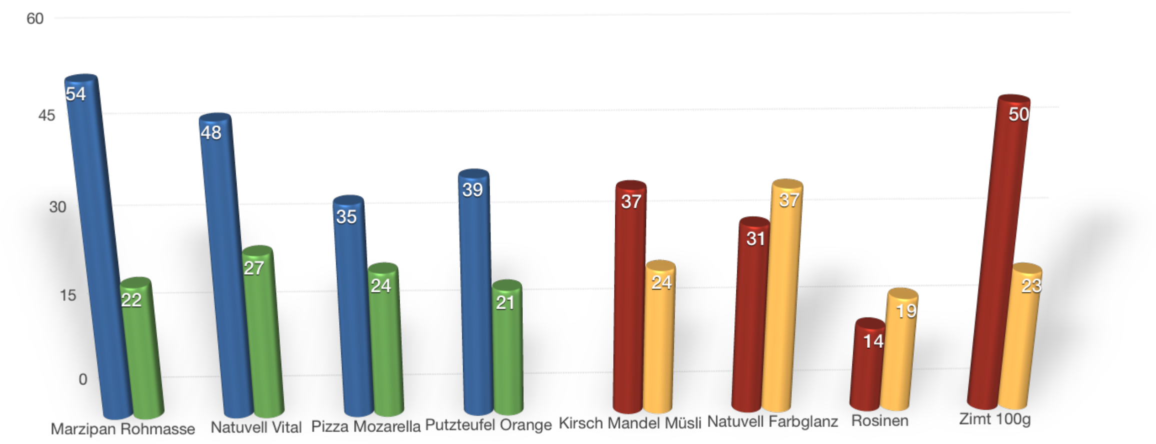
The task completion time is the elapsed time for the user to complete a single search task. For the gaze search the task completion time is defined as the elapsed time between the first gaze interaction with a Move plate and the moment the sought product was added to the cart. It differentiated a little bit for the second search type, namely the speech search. Here the time measurement started with the first gaze on the bubble symbol on the search interface. It ended when the searched product was added to the cart. For every trial, the task completion time was measured. The following times are all mean values.

The average task completion time of all trials was 33.49s (SD=24.83). Regarding each of the different four tasks, task 4 was the fastest with a mean time of 23.45s(SD=8.29), followed by task 2(M=25.78s, SD=16.03). While task 1 was in average(M=34.37s, SD=28.11), task 3 came off the worst mean time(M=54.12s, SD=30.4). When comparing the task completion time by search type, the speech search was clearly faster(M=24.60s, SD=12.73) than the gaze search(M=43.38s, SD=30.68). The tasks performed on Desktop were 30.04s(SD=23.15) quicker than the ones in VR(M=37.20s, SD=26.12).



Additionally, univariate ANOVA analysis was conducted for the task, search type, and output type. A significance was found regarding the task ($p<0.01$, $F(3,237)=8.605$). The analysis of the input type showed a significant effect ($p<0.05$, $F(1,237)=4.542$). Furthermore, an interaction was found between the input and output ($p<0.01$, $F(1,237)=14.606$).

Regarding the mean task completion times of the different products, some of them had interesting times. First the VR setup will be analysed. The two products "Marzipan Rohmasse" and "Natuvell Vital" had a much longer mean time for the gaze search than for the speech search. In the Desktop setup, it stands out that the product "Rosinen" had the shortest mean time by far for both search types. A huge difference between the times of the search type was also measured for the product "Zimt 100g".

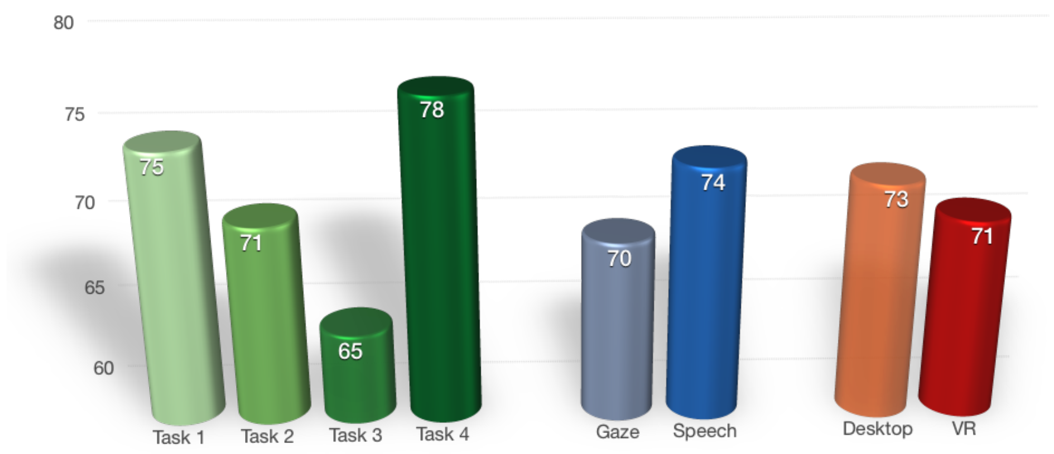


System Usability Scale

The System Usability Scale(SUS) was used to gather the following data. It is a good questionnaire to measure the perceived usability. The average SUS score for all four tasks together was 72.19(SD=15.92). Task 4 had the best SUS score(M=77.5, SD=13.213), closely followed by task 1(M=75.00, SD=17.182), task 2(M=70.938, SD=12.844) and last task 3(M=65.313, SD=17.386).

Regarding the search types, the speech search(M=74.219, SD=13.213) was higher than the gaze search(M=70.156, SD=17.89).By comparing the SUS scores of the output types, Desktop had a slightly higher score(M=72.969, SD=15.246) than VR (M=71.406, SD=16.409).

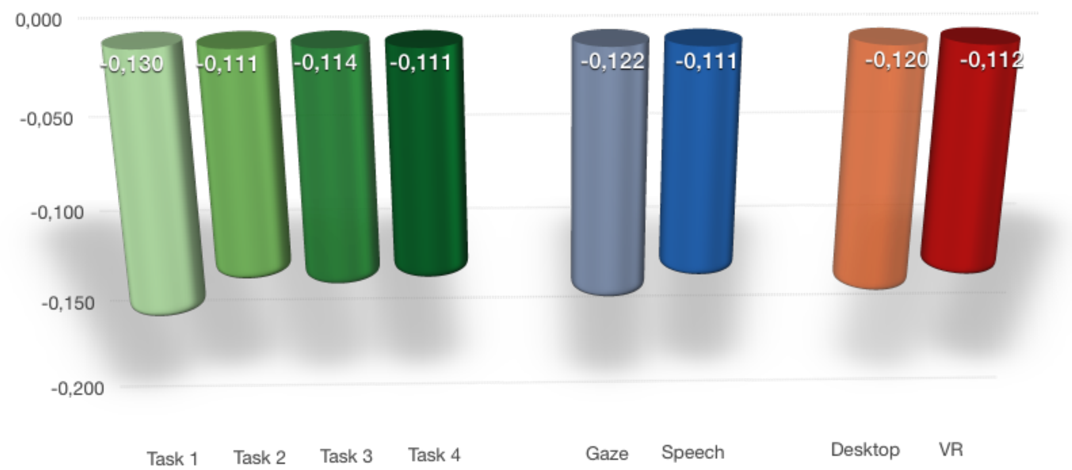
An univariate ANOVA showed a significance regarding the task($p<0.01$, $F(3,252)=7.816$). Besides, the input type has a significant effect ($p<0.05$, $F(1,252)=4.555$). Furthermore, an interaction between input and output was found($p<0.01$, $F(1,252)=18.219$).



User Experience

The user experience was measured with the User Experience Questionnaire(UEQ), which has a scale from -3 to 3. It contains questions from six categories, which are Attractiveness(A), Perspicuity(P), Efficiency(E), Dependability(D), Stimulation(S) and Novelty(N). The average value of the user experience was -0.116(SD=0.14). Task 2 had the highest average value(M=-0.111, SD=0.131) together with task 4(M=-0.111, SD=0.094). Task 3(M=-0.114,SD=0.108) and task 1(M=-0.130, SD=0.200) had only negligible lower values.

When regarding the two search types, the speech search had a higher value(M=-0.111, SD=0.113) than the gaze search(M=-0.122, SD=0.160). Among the output devices, the VR device reached a minimal higher value (M=-0.112, SD=0.101) than Desktop device(M=-0.120, SD=0.169).



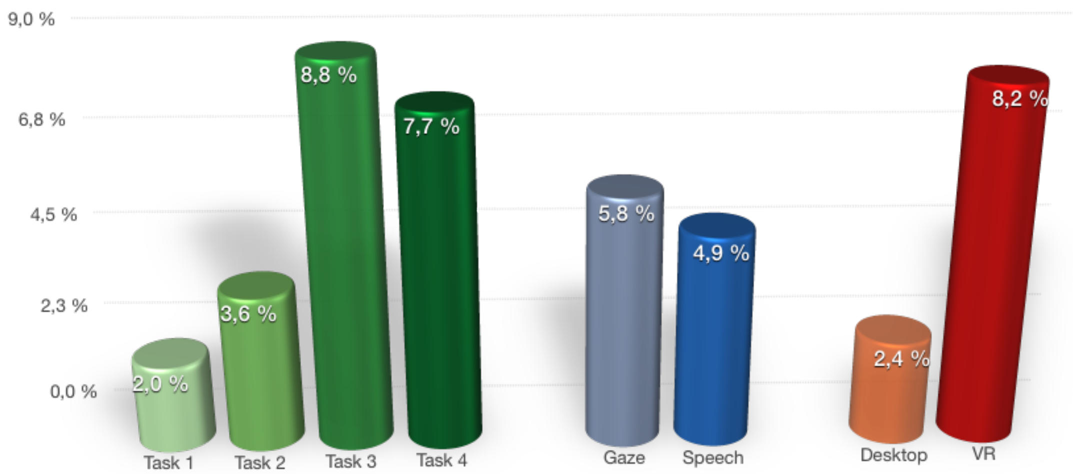
A multivariate ANOVA with the six categories as dependent variables was conducted. Regarding the different tasks, it showed no significance of A, N, D and E. But significances of P($p<0.01$, $F(3,252)=4.716$) and S($p<0.01$, $F(3,252)=5.871$) were found. Regarding the input types, a significance of P($p<0.01$, $F(1,252)=9.400$) and a significant effect of N($p<0.05$, $F(1,252)=4.692$) were found. The output had no significance for any factor. Additionally, an interaction between input and output regarding P($p<0.05$, $F(1,252)=3.925$) and S($p<0.01$, $F(1,252)=13.154$) were found.

Motion Sickness

A Motion Sickness Assessment Questionnaire(MSAQ) was used to measure the motion sickness of the participants in percent. It verifies four categories of motion sickness, which are defined as gastrointestinal, central, peripheral, and sopite-related. The highest scores were reached by task 3(M=0.088, SD=0.096) and task 4(M=0.077, SD=0.083). Much lower scores were reached by task 2(M=0.028, SD=0.044) and task 1(M=0.020, SD=0.036).

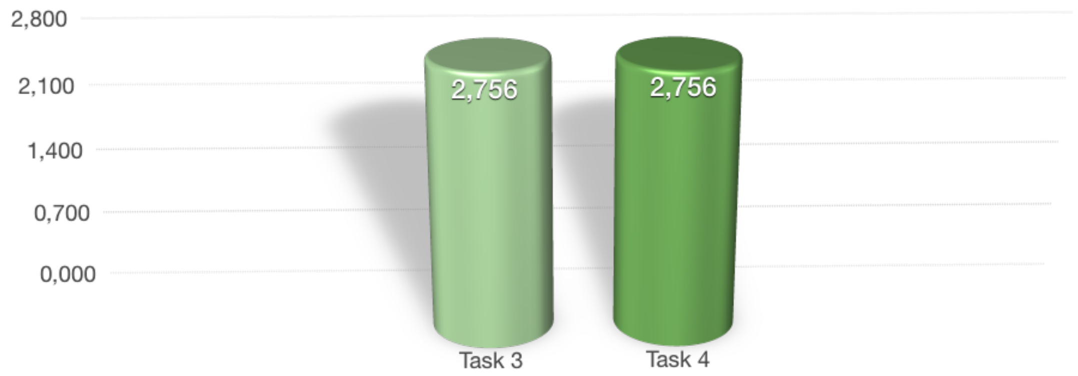
Regarding the search types, gaze search had a marginal higher value (M=0.058,SD=0.080) than speech search(M=0.049, SD=0.070). However, the difference between the values of Desktop(M=0.024, SD=0.040) and VR(M=0.082, SD=0.089) was much bigger.

A multivariate ANOVA with the four categories as dependent variables was conducted. It pointed out that there are overall significances concerning the different tasks. Apart from that, there were no significances found regarding the input. There was also no interaction found between the input and output types.



Immersion

To measure the immersion a Presence Questionnaire with values between 1 and 7 was used. The users filled it out after the VR tasks, so it is only possible to analyze the immersion values for the last two tasks. Task 3(M=2.756, SD=0.856) reached nearly the same immersion score as task 4(M=2.756, SD=0.909).



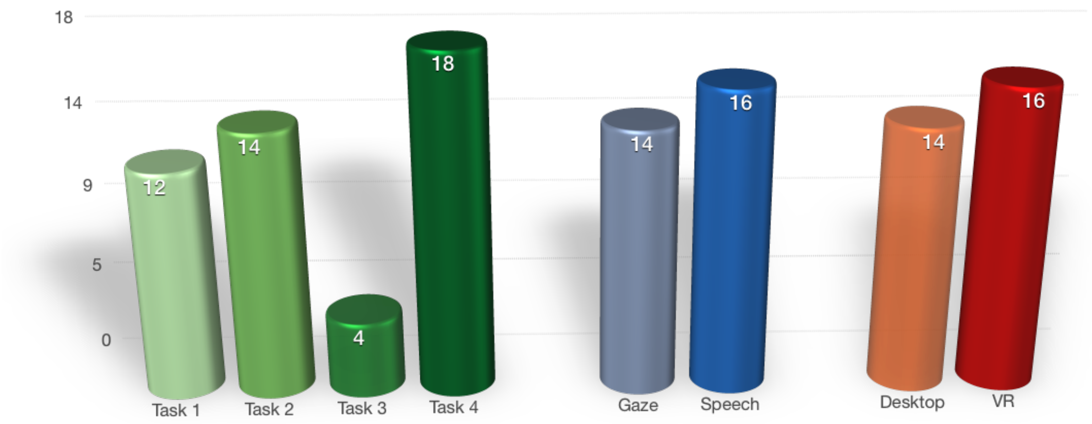
Dynamic Intercategorical Score

The Dynamic Intercategorical Score(DIS) was invented to make an overall comparison and combine all the gathered results. The idea of this method will be explained in the following.

The DIS gives every variable a specific amount of points. Thus, the variables can be compared with each other. The ranking is dependent on some chosen categories. For example, a category could be the task completion time. The variables that should be compared are ranked in every of these categories. The best variable in a category gets 6 points, the second one 4 points, the third one 2 points and the fourth one 0 points. Now the achieved points in every category are summed up for every variable. The variables can now be ordered in descending order by its obtained amount of points, while the variable with the most points is the best.

This ranking method has some further advantages. Since there is no limit to a count of categories, it is simple to add another category without destroying the validity of the previous ranking. It is also possible to add weights for specific categories by simply multiply all achieved points in this category with the weight. It offers the opportunity of taking some categories more into account than others.

First, the DIS will be applied to the different tasks as variables. They are rated in the categories task completion time, system usability, user experience and motion sickness. Task 4 had the highest score(P=18), followed by task 2(P=14), task 1(P=12) and task 3(P=4). When applying the MRK to the search type, speech search reached 16 points, closely followed by gaze search(P=14). Regarding the output devices, VR had more points (P=16) than Desktop(P=14).



5.8 Discussion

Task Completion Time

The question about the efficiency of the speech search can be answered by looking at the results of analysis of the task completion time. Both, task 2 and 4 make use of the speech search instead of the gaze search. The other two tasks that use the gaze search had a worse mean task completion times. The direct comparison of the search types also showed that the speech search is faster. An explanation for this outcome is that it is easy and fast to say the name of the queried product. The user is brought directly to the searched product and thus can put it into the cart directly. With the gaze search, the exact position of the queried product has to be known. If it is not, the user has to move around and look for it, which can take a long time, especially when the shop's map is not known.

However, it must be said that users that learned about the shop's map over time were also able to use the gaze search efficiently. Due to the small size of the shop, this is possible. In a much bigger one, they would still need much time to find a product even if they know its plan because they first need to move to the desired product.

When regarding the task completion times of the output devices, Desktop was faster, which makes sense because the gaze search on Desktop is much more efficient than the same in VR. The main reason for this is that the user has a bigger field of view on Desktop. The already small display of the mobile device is split into two parts, so the field of view gets much smaller here. The user can see more of the shop on Desktop and thus can find products faster. The fact that in the comparison of the task completion times for the different tasks, task 4 was slower than task 3 confirms this argumentation.

Last the salient task completion times regarding the various products will be explained. The short mean time for "Rosinen" is easy to explain. This product is located very close to the start point. So most of the users saw the product directly and did not have to move around. The products "Zimt 100g" and "Marzipan Rohmasse" are tiny, so the users overlooked them when moving through the scene. Here there is a significant advantage of the speech search because the searched product gets highlighted and thus cannot be overlooked so easily.

For "Natuvell Vital" there was a significant difference between the times of the two search types. All products from this shelf share the same geometry. So they only differ in their textures. There is also the fact that all textures have a lower resolution in the VR setup. So with the gaze search the user has to go through all products on this shelf and check them one by one for the searched product, which takes a lot of time. With the speech search, the user just has to find the product in the result list on the search interface. Additionally, the product gets highlighted after the user is ported to it, so not all of them have to be checked.

System Usability Scale

The overall SUS score is relatively high, which proves an excellent usability for the general VRProductFinder. When regarding the usability scores of the different tasks, it was shown that task 4 had the best usability score. By comparing this with the much lower score of task 3, which uses VR with gaze search, it shows that the speech search is a huge enrichment for the VR setup. In contrast, in the Desktop setup the gaze search reached a better SUS score than the speech search, which is due to the fact that the Desktop display is bigger than the small one in the mobile device. Additionally, the resolution of the Desktop display is greater, because it is not halved for the stereoscopic view. These circumstances result in a bigger field of view for Desktop, so that the user can see more and sharper products, which makes it easier to find products by gaze search. The minimal difference between the scores of the output devices confirms this. Desktop has its strengths in gaze search and VR in speech search.

Regarding the search types, speech search had a higher score than gaze search, which can be explained with the efficiency of the speech search. In the discussion about the task completion time, it was already shown that speech search is faster then gaze search. So, the participants got more rapidly to the desired product without the frustrating need to move around and look after it. The result of the univariate ANOVA also confirms these outcomes. It shows that the SUS score depends on the chosen task and the selected input type.

User Experience

The scores reached for the user experience regarding the different tasks distinguished slightly. So did the scores regarding the search or output types. All scores were marginal below zero which means that the user experience was neither good nor bad. But the multivariate ANOVA had some interesting findings. When regarding the perspicuity for the different tasks, task 2 and 4 had much better values than the others, which shows again that speech input is easier to use. The analysis of the perspicuity for the input types also confirms this.

Motion Sickness

The analysis of the motion sickness scores showed that VR causes much more motion sickness than Desktop, which is obvious because Desktop applications should not cause motion sickness. The users are not in an immersive virtual reality and thus should not have motion sickness symptoms. This fact can be obtained from the scores of the two VR tasks and also when comparing the VR and Desktop scores. Additionally, it is emphasized by the found significances in the multivariate ANOVA.

The gaze search reached higher motion sickness values when comparing the search types directly. Also, the comparison of task 3 and 4 is interesting because it shows that gaze search causes more motion sickness unless they both use VR. The reason for this is that the users have to move around when using the gaze search. As was beign shown in the discussion of the task completion time, they needed more time to find the products and thus were in the virtual environment for a longer time. So they had more time to get motion sickness. The frequently moving of the head in the gaze search is also a reason.

Immersion

The results of the immersion scores showed that the intensity of immersion is not affected by the search type because both analyzed tasks had the same value. The reached values are located around the middle of the scale, which shows that the users had a good feeling of presence when using the VR output. There are several reasons why the values are not higher. On the one hand the render quality on the mobile device is not as good as on a Desktop. Small textures and restricted lighting reduce the immersion, which could be solved by using more powerful devices. On the other hand, the inaccuracy of the sensors can cause a temporary latency of the head movement. It could be avoided by using a device with better sensors like the Oculus Rift.

Dynamic Intercategorical Score

The result of the DIS regarding the different tasks showed that speech search in the VR setup did best by a large margin. The speech search on Desktop also got many points, while the gaze search in VR came off poorly, which reflects the assumption that speech search is better and more efficient than the gaze search. The comparison of the two search types additionally emphasizes this hypothesis, because the speech search has a slightly higher score. It also confirms the previously mentioned disadvantages of the gaze search in VR. When regarding the scores of the two output types there is only a minor difference between them, so none of them can be favored.

Observations

During the study, some interesting observations were made. These observations will be explained in the following together with the comments that the participants made on the last questionnaire.

Positively mentioned were the innovative idea of online shopping. Many of the participants liked the fact that users do not have to leave the house anymore to have a realistic experience of shopping. It was considered as an advantage for disabled and senior citizens or just users that do not have much time for shopping. Another positive aspect was the bright and tidy design of the shop. But the biggest advantage that was mentioned by nearly every participant was the improvement that the speech search did. In general, the VRProductFinder has been well received by most of the test persons. More than 60 percent of them stated that they would use the VRProductFinder at home.

Of course, negative aspects were also observed. First, the points regarding speech search will be looked at. The fixed position of the speech interface was liked and disliked. The proponents liked the fact that it is realistic because in real shops the elements also have set positions. But some of them disliked that they first have to move to it to use it. They proposed to enable the speech interface dynamically at the current position. There were participants who tried to use the speech search by searching for the category of the desired product, which does not work at the moment but could be implemented in a future version.

Some participants had a problem when using the search interface in combination with the Desktop setup. They gazed on the bubble symbol and said the products name. Due to the wide field of view, they watched the appearing results without moving away from their viewing direction. So they gazed again on the bubble symbol and restarted the speech search accidentally. The problem should be reconsidered in future versions.

In general, the gaze search was not very pleasant. Almost all of the test persons complained about its efficiency. The participants did not know the shop's map and thus had problems to orientate themselves in the shop. So some of them proposed to add a map or at least some signposts for a better orientation. It was observed that the participants who did not know the shop's plan lost much time with running around aimlessly. Some of them mentioned that this is the main reason why they prefer the speech search.

There were some negative facts regarding the VR setup. The buttons on the cart were positioned too low. Because of this, some people had problems to gaze at them. In general, there were problems with the fixing of elements, because the interaction radius of some objects is too small, so the gaze moved too easily out of this area. Besides, the low render quality and blurry textures amplified this problem. It was requested to improve this.

Additionally, to all previously mentioned improvement suggestions, a user had the idea to show the product name when gazing on it already.

6 Conclusion

Head-mounted displays become cheaper with the growing market of immersive virtual reality. With this popularity, the computation power of mobile devices also increases. It is possible to use immersive virtual reality applications nearly everywhere since the availability of mobile Internet also rises. Another growing sector is e-commerce, which is used by nearly everyone nowadays. Current e-commerce shops may be functional, but do not provide an immersive shopping experience. This thesis focused on developing a shopping application for immersive VR that includes all the advantages of e-commerce.

To do this the VRProductFinder was implemented, which realizes exactly this. It provides the possibility to search by gaze but also by speech and so offers the opportunity to search more efficiently for products in the virtual environment. The several output devices make the user independent and provide the freedom to use the VRProductFinder nearly everywhere. In conclusion, with the VRProductFinder a solid basic framework was implemented, which can easily be extended with further functionalities. At the same time, it is entirely dynamical, so that many different shops can be used and personalized for each customer.

At the end, a study was conducted with participants, who were supposed to search several products in the virtual environment using different output and search types. They filled some questionnaires, which were used to answer the question about the usability, efficiency, and immersion of the VRProductFinder. It was confirmed that the overall usability of the application is good. Especially for the VR setup speech search should be preferred. In contrast, the Desktop setup can be used with both search types for a good usability. Regarding efficiency, the speech search was faster than the simple gaze search. Here it is important that the data quality is good to provide a proper recognition. The study also proved an immersive shopping experience for VR. Naturally, it should be kept in mind that VR causes a higher motion sickness than Desktop, but there are many ways to reduce these symptoms. This relatively positive outcome was accentuated by the great resonance of the participants.

7 Future Work

The architecture of the VRProductFinder permits to extend it unproblematically. There are some clear points for future work, which will be presented in the following.

It would be pragmatic if the user does not have to search for every single product. Therefore it would be possible that the user uploads a buying list with all the desired products. After a path with all desired products on its way would be generated and the user will be lead along it. It is an advantage because it costs less time for searching the products. Another improvement regarding searching would be a search by text input. At this moment the only way to search for a product by its name is the speech search. There is already a text search for products implemented as an HTML page. This HTML page could be integrated into the scene as a simple plane. Thus a user could enter the query by text input. It is practical, if for some reasons the speech input is currently not possible or accurate enough.

Machine learning algorithms are often used for recommendation systems. These algorithms could analyze the buying behavior of users and as a result of this recommend specific products to them. The products could also be arranged in such a way that they rather see products that are interesting for them. But not only a derivation of the interests of other customers can be used. It would also be an improvement if own interests could be defined in order to hide uninteresting products.
The cart has a capacity of just very few products depending on their size. The main reason for this is that they are currently arranged in a line. Newly added products are just pushed behind the last added product without rotating or adjusting them. When there no space left behind the last one, the user cannot add new ones to the cart anymore even if there is free space left beside already added products. It would be a huge improvement if this open space would be used to add small products that fit there so that the capacity is increased many times.

This thesis did not test the usability and efficiency of the Oculus Rift. It would be interesting to do this to compare it with the results that were gathered using the Desktop and VR setup. Due to the DIS, it is very easy to compare the results of new studies with the results of the already conducted ones. It must only be minded to use the same categories for a valid comparison.

Acknowledgement

I especially want to thank Marco Speicher for his assistance during the process of my bachelor thesis. He always helped me issues from the elaboration of the idea via the implementation of it through to the writing of this thesis. I would like to thank Simon Pähler for providing the models and textures of the products that were used in the scene. I also want to thank Prof. Antonio Krüger for giving me the opportunity to write my bachelor thesis at the German Research Center for Artificial Intelligence in Saarbrücken.

Thanks to Gian-Luca who posed for the example images and his friendship in the lab. Another thanks goes to my parents, who always were behind me and supported me during my entire study days.

Bibliography

[1] Julius Adorf. Web speech api. *KTH Royal Institute of Technology*, 2013.

[2] Alienware. Alienware. `http://www.alienware.de/`. Accessed: 2016-09-06.

[3] Amazon. Amazon. `https://www.amazon.de`. Accessed: 2016-09-06.

[4] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[5] Michel Buffa and J-C Lafon. 3d virtual warehouse on the web. In *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pages 479–484. IEEE, 2000.

[6] Sophic Capital. Virtual reality: A virtual goldmine for investors. `http://sophiccapital.com/wp-content/uploads/2014/11/Download-Full-Virtual-Reality-Report-Here.pdf`. Accessed: 2016-09-06.

[7] Luca Chittaro and Roberto Ranon. Dynamic generation of personalized vrml content: a general approach and its application to 3d e-commerce. In *Proceedings of the seventh international conference on 3D Web technology*, pages 145–154. ACM, 2002.

[8] Luca Chittaro and Roberto Ranon. New directions for the design of virtual reality interfaces to e-commerce sites. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 308–315. ACM, 2002.

[9] Luca Chittaro and Roberto Ration. Adding adaptive features to virtual reality interfaces for e-commerce. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 86–97. Springer, 2000.

[10] Createjs. Tweenjs. `http://www.createjs.com/tweenjs`. Accessed: 2016-09-06.

[11] Olga De Troyer, Frederic Kleinermann, Haithem Mansouri, Bram Pellens, Wesley Bille, and Vladimir Fomenko. Developing semantic vr-shops for e-commerce. *Virtual Reality*, 11(2-3):89–106, 2007.

[12] Dell. Dell p2210. `http://www.dell.com/ed/business/p/dell-p2210/pd`. Accessed: 2016-09-06.

[13] Elegiant. Elegiant 3d vr glasses. `http://www.elegiants.com/de/ELEGIANT-360-Viewing-Immersive-Virtual-Reality-3D-VR-Glasses-Google-Cardboard-3D-Video-Games-Glasses-VR-Headset-Compatible-with-3_5-6_0-inches-Android-and-Apple-Smartphones-for-3D-Movies-and-Games-p-167.html` . Accessed: 2016-09-06.

[14] Ajoy S Fernandes and Steven K Feiner. Combating vr sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 201–210. IEEE, 2016.

[15] Peter J Gianaros, Eric R Muth, J Toby Mordkoff, Max E Levine, and Robert M Stern. A questionnaire for the assessment of the multiple dimensions of motion sickness. *Aviation, space, and environmental medicine*, 72(2):115, 2001.

[16] Google. Google cardboard. `https://vr.google.com/cardboard/`. Accessed: 2016-09-06.

[17] Google. Nexus 5x. `https://www.google.com/intl/de_de/nexus/5x/`. Accessed: 2016-09-06.

[18] Google. Web speech api instruction. `https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API`. Accessed: 2016-09-06.

[19] Gerald Häubl and Pablo Figueroa. Ineractive 3d presentations and buyer behavior. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 744–745. ACM, 2002.

[20] Howell Istance, Richard Bates, Aulikki Hyrskykari, and Stephen Vickers. Snap clutch, a moded approach to solving the midas touch problem. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 221–228. ACM, 2008.

[21] jQuery. jquery. `https://jquery.com/`. Accessed: 2016-09-06.

[22] Archa Khandewal, Abhinav Shah, and Arasil Vahora3 Gopi Bhatt. Virtual commerce: The future. *International Journal of Advance Research in Engineering, Science & Technology*, 2016.

[23] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and Usability Engineering Group*, pages 63–76. Springer, 2008.

[24] JJ-W Lin, Henry Been-Lirn Duh, Donald E Parker, Habib Abi-Rached, and Thomas A Furness. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 164–171. IEEE, 2002.

[25] Ming Liu. A study of mobile sensing using smartphones. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[26] Yosi Mass and Amir Herzberg. Vrcommerce—electronic commerce in virtual reality. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 103–109. ACM, 1999.

[27] Microsoft. Microsoft kinect 2. https://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-for-Windows-Developer-Bundle/productID.314513600. Accessed: 2016-09-06.

[28] Ronald R Mourant and Thara R Thattacherry. Simulator sickness in a virtual environments driving simulator. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 44, pages 534–537. SAGE Publications, 2000.

[29] Mozilla. Web speech api restrictions. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. Accessed: 2016-09-06.

[30] Mozilla. Webvr. https://webvr.info/. Accessed: 2016-09-06.

[31] M Papaefthymiou, K Plelis, D Mavromatis, and G Papagiannakis. Mobile virtual reality featuring a six degrees of freedom interaction paradigm in a virtual museum application. *Foundation for Research and Technology - Hellas (FORTH)*, 2015.

[32] Antonio Robles, José P Molina, Víctor López-Jaquero, and Arturo S García. Even better than reality: the development of a 3-d online store that adapts to every user and every platform. In *HCI International*, pages 10–20, 2005.

[33] Andrea Sanna, Bartolomeo Montrucchio, Paolo Montuschi, and C Demartini. 3d-dvshop: a 3d dynamic virtual shop. In *Multimedia 2001*, pages 33–42. Springer, 2002.

[34] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. "your word is my command": Google search by voice: A case study. In *Advances in Speech Recognition*, pages 61–90. Springer, 2010.

[35] Mike Schuster. Speech recognition for mobile devices at google. In *Pacific Rim International Conference on Artificial Intelligence*, pages 8–10. Springer, 2010.

[36] Prashant Sharma. Challenges with virtual reality on mobile devices. In *ACM SIGGRAPH 2015 Talks*, page 57. ACM, 2015.

[37] Sarah Sharples, Sue Cobb, Amanda Moody, and John R Wilson. Virtual reality induced symptoms and effects (vrise): Comparison of head mounted display (hmd), desktop and projection display systems. *Displays*, 29(2):58–69, 2008.

[38] Skezo. Reticulum. https://github.com/skezo/Reticulum. Accessed: 2016-09-06.

[39] Boris Smus and Christopher Riederer. Magnetic input for mobile virtual reality. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 43–44. ACM, 2015.

[40] Anthony Steed and Simon Julier. Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 43–46. IEEE, 2013.

[41] Nobuhisa Tanaka and Hideyuki Takagi. Virtual reality environment design of managing both presence and virtual reality sickness. *Journal of physiological anthropology and applied human science*, 23(6):313–317, 2004.

[42] ThreeJS. Threejs. `http://threejs.org/`. Accessed: 2016-09-06.

[43] Martin Usoh, Ernest Catena, Sima Arman, and Mel Slater. Using presence questionnaires in reality. *Presence: Teleoperators and Virtual Environments*, 9(5):497–503, 2000.

[44] DM Whittinghill, Bradley Ziegler, T Case, and B Moore. Nasum virtualis: A simple technique for reducing simulator sickness. In *Games Developers Conference (GDC)*, 2015.

[45] Wikipedia. Threejs. `https://en.wikipedia.org/wiki/Three.js`. Accessed: 2016-09-06.

[46] Soojeong Yoo and Callum Parker. Controller-less interaction methods for google cardboard. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pages 127–127. ACM, 2015.

Appendices

User Experience Questionnaire

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

Motion Sickness

***Instructions.* Using the scale below, please rate how accurately the following statements describe your experience**

Not at all Severely

1——2——3——4——5——6——7——8——9

1. I felt sick to my stomach (G)
2. I felt faint-like (C)
3. I felt annoyed/irritated (S)
4. I felt sweaty (P)
5. I felt queasy (G)
6. I felt lightheaded (C)
7. I felt drowsy (S)
8. I felt clammy/cold sweat (P)
9. I felt disoriented (Q
10. I felt tired/fatigued (S)
11. I felt nauseated (G)
12. I felt hot/warm (P)
13. I felt dizzy (C)
14. I felt like I was spinning (C)
15. I felt as if I may vomit (G)
16. I felt uneasy (S)

SUS VR

1. Please rate *your sense of being in the* office space, on the following scale from 1 to 7, where 7 represents your *normal experience of being in a place*.
   I had a sense of "being there" in the office space:
   1. Not at all ... 7. Very much.
2. To what extent were there times during the experience when the office space was the reality for you?
   There were times during the experience when the office space was the reality for me...
   1. At no time ... 7. Almost all the time.
3. When you think back about your experience, do you think of the office space more as *images that you saw*, or more as *somewhere that you visited*?
   The office space seems to me to be more like...
   1. Images that I saw ... 7. Somewhere that I visited.
4. During the time of the experience, which was strongest on the whole, your sense of being in the office space, or of being elsewhere?
   I had a stronger sense of...
   1. Being elsewhere ... 7. Being in the office space.
5. Consider your memory of being in the office space. How similar in terms of the *structure of the memory* is this to the structure of the memory of other *places* you have been today? By 'structure of the memory' consider things like the extent to which you have a visual memory of the office space, whether that memory is in colour, the extent to which the memory seems vivid or realistic, its size, location in your imagination, the extent to which it is panoramic in your imagination, and other such *structural* elements.
   I think of the office space as a place in a way similar to other places that I've been today...
   1. Not at all ... 7. Very much so.
6. During the time of the experience, did you often think to yourself that you were actually in the office space?

System Usability Scale

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

Final

Participant ID

Your answer

Age

Your answer

Job

Your answer

Gender

- ◯ Male
- ◯ Female

How experienced are you with Virtual Realtiy?

	0	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	◯	much

How often do you shop online each month?

Your answer

How often do you go shopping in supermarkets each month?

Your answer

How experienced are you with computers?

	0	1	2	3	4	5	
Not at all	○	○	○	○	○	○	Much

What did you like about the VRProductFinder?

Your answer

What did you dislike about the VRProductFinder?

Your answer

What would you change

Your answer

How likely would you use the VRProductFinder?

	0	1	2	3	4	5	
Unlikely	○	○	○	○	○	○	Likely

What are reasons for it or against it?

Your answer

Further notes

Your answer