Technology Arts Sciences
TH Köln

Project Documentation

Interaction Lab

written by

Vera Brockmeyer	(Matrikelnr. 11077082)
Anna Bolder	(Matrikelnr. 11083451)
Britta Boerner	(Matrikelnr. 11070843)
Laura Anger	(Matrikelnr. 11086356)

Interactive Systems in SS 2017

Supervisor:
Prof. Dr. Stefan Michael Grünvogel
Institute for Media- and Phototechnology

Inhaltsverzeichnis

1 Introduction

Vera

The main idea of Virtual Reality (VR) is to ensure a totally immersive user experience. A first important milestone was the development of the first Head-Mounted-Display (HMD) in 1966 by Sutherland and Goertz. This HMD offers the possibility to walk around in each virtual scene and to look at the scene from every view point. Despite this possibility, a totally immersive experience asks for user interactions with virtual objects like selecting, grabbing, manipulating, movement and indirect controlling via widgets, gestures and voice input in virtual scenes. Especially, a realistic grabbing and positioning of these objects is required which should come close to human motion sequences and fits to the human cognition and experiences. These kind of immersive interaction methods should include further requirements as well. First, a high precision rate is unavoidable and must be guaranteed. Second, the method should not be tiring for the users. Finally, the methods should be easy to learn and offer an ergonomic usage.

The number of VR devices and applications increases heavily during this decade and the relevance of consumer and business VR applications rises steadily. Several suppliers currently offer various methods and devices for grabbing interactions. The most common are the *Oculus*-HMD, *HTC Vive*-HMD, *HTC Vive*-Controller, data gloves and motion capturing systems for hand-tracking like the *LeapMotion*-Controller. Most devices offer a system with two hand controllers at this state. These hand controllers enable the use of virtual menus and offer a robust hand tracking. The latter is urgently needed to implement common grabbing interactions. Due to this rising amount of VR devices, it is vital for the development and improvement of grabbing methods to evaluate their usability and performance in an adequate environment as well as to compare them with the state of art methods.

In the following sections, we describe and discuss an interactive VR system to compare different controller-based grabbing methods to evaluate their usability, error rate and time consume. This VR system runs with the *HTC Vive*-HMD and two *HTC Vive*-Controllers. Thus, all offered grabbing methods are controller based which could be learned in a special learning room and evaluated in a predefined VR supermarket with all required measurement tools.

1.1 Motivation

Vera

Currently, there exist no interaction laboratory that compares the different grabbing interaction methods. Similar laboratories [16][24][?] have been developed but they do not allow the user to compare different kind of grabbing methods. In case of [?] and [24], it provides only some VR applications or devices to experience the methods in different scenarios. Another [16] laboratory evaluates natural grabbing methods without a credible evaluation.

This circumstance claims a virtual laboratory, where different interaction methods could be compared, demonstrated, or tested in adequate virtual test scenes. Furthermore, all tests and compares should be based on scientific standards to allow factual and useful results. Thus, this laboratory should provide all required measurement tools to start credible and scientific studies of pre-implemented grabbing methods. This helps to standardise the evaluations and to increase their comparability of common studies. Another achievement is that the user friendliness of interaction methods which are not tiring and do not destroy the immersion could be improved by researcher.

Another aspect of those user-friendly methods is the increasing usability of VR applications which will yield to a higher consumer preference of devices with their implementation. Therefore, the profit of VR device suppliers will be squeezed and the relevance of those products will expand worldwide.

In an educational context, it could be used for demonstrations during lectures. This will give the lecturer an effective tool at hand to explain the importance of usability as well as the advantages and disadvantages of each grabbing method. Another aspect of this laboratory, is to give students a tool for the technical realisation of interaction studies in virtual (or augmented) reality environments.

1.2 Project Goal

Vera

Hence, a virtual interface laboratory with an environment to test and compare grabbing methods should be developed. Furthermore, it should offer a possibility to develop new ones as well and it offer a use for teaching purposes. Thus, one task is to develop sophisticated test scenes for testing the interaction methods. These scenes should implement test exercises in different difficulty levels and represent typical and well-known environments like shops. All relevant parameters for the evaluation of the methods should be measured automatically and saved in an output comma separated value file. The latter must be easily imported in common statistic tools. Additional, the laboratory should offer digital questionnaires to evaluate the usability of each provided method as well as one that enquire the system relevant properties like motion sickness, immersion, and latency.

2 State of the Art

Laura

In the following sections the fundamentals and scientific knowledge of all parts of the *Interaction Lab* will be described. Publications of similar projects are described in section 2.1. Furthermore, an overview of methods of (grabbing) interactions (compare section 2.2), as well as their evaluation methods (compare section 2.3) are given.

2.1 VR Labor

Anna

In the development process of the grabbing interactions there were also different evaluations of the differences between the interactions. There are a few papers and other articles. In the following three different works will be described shortly.
In the year 1999 Pouryrev and Ichikawa categorized and evaluated different grabbing interactions in their paper [22]. In their project they used hand tracking to measure the position of the real hand and to visualise a virtual representation of the hand. The participants than had to select different simple test objects, like cubes, spheres and cylinders. Also a positioning task were supplied. A few years later Lee et al. evaluated different raycast grabbing methods [14]. Lee at al. used a 3D mouse for tracking positions and orientation of the hand. With this mouse the participants selected spheres on various and random positions. For this the participants were ask not to change the position of their head.
In the last year Eriksson published his master thesis, which is relative similar to this project [8]. Erikkson used the Oculus Rift with the Oculus Touch to provide the tracking and visualisation. In this project the participants also had to complete tasks in a virtual shop. The tasks had similar requirements related to the selection but also refer to manipulation and translation of the objects.

2.2 VR Grabbing-Interactions

Laura

To create an immersive virtual environment it is, inter alia, necessary to make use of adequate grabbing methods [2]. As mixed reality was not taken into consideration for implementing the *Interaction Lab*, all objects, the user can interact with, are completely virtual.
The interaction methods provided by the *Interaction Lab* have no haptical feedback [1], which would allow a conclusion about the surface quality of the gripped object. Furthermore, they only use the hardware described in section 3.1.2 for the execution of the interactions.
Many of the common methods of interaction, which are available in the *Interaction*

Lab (compare section 4.3) are explained in the paper by Bowman and Hodges [2]. In their explanations they also mentioned the *Go-Go* technique, which presents a conceivable extension of the laboratory.

2.3 VR Grabbing-Interactions Evaluation

Britta

3 Materials

Britta

The following section will give an overview over all hard- and software that was used in this project.

3.1 Hardware

Britta

The hardware consists of a head-mounted display, the HTC Vive and a computer.

3.1.1 Computer

Britta

@Britta: Vielleicht möchtest du die Tabelle einfach übernehmen und nur die Daten des PCs ändern?

Die Hard- und Software-Voraussetzungen für die Ausführung der *Unity*-Anwendung in Verbindung mit der *HTC Vive*, welche in Tabelle 2 aufgelistet sind, werden von dem verwendeten Computer übertroffen.

3.1.2 HTC Vive

Britta

The *HTC Vive* is a head-mounted display which is being produced by *HTC* in cooperation with *Valve* [29]. It was introduced on the 1st of March 2015 at the *Mobile World Congress* [20].

Dieist ein Head-Mounted Display, welches von *HTC* in Kooperation mit *Valve* [29] produziert wird. Vorgestellt wurde dieses am 1. März 2015 im Vorfeld des *Mobile World Congress* [20].

CGPC6	**Beschreibung**
Prozessor	Intel Core i7 6700 CPU @ $4 \times 3.4 - 4.0$ GHz
Arbeitsspeicher	16 GB
Grafikkarte	NVIDIA GeForce GTX 980
Betriebssystem	Windows 10 Education 64 bit
Schnittstellen	2× USB 3.0, 5× USB 2.0, 1× HDMI

Tabelle 1: Übersicht der technischen Daten des Computers für die *Unity*-Simulation.

HTC Vive	**Systemvoraussetzungen**
Prozessor	mindestens Intel Core i5-4590 oder AMD FX 8350
Grafikkarte	mindestens NVIDIA GeForce^{TM} GTX 1060 oder AMD Radeon^{TM} RX 480
Arbeitsspeicher	mindestens 4 GB
Videoausgang	1× HDMI 1.4-Anschluss oder DisplayPort 1.2
USB	1× USB 2.0-Anschluss
Betriebssystem	Windows 7 SP1, Windows 8.1 oder Windows 10

Tabelle 2: *HTC Vive* Systemvoraussetzungen [11].

Die Auflösung des Displays beträgt insgesamt 2160×1200 Pixel, was 1080×1200 Pixeln pro Auge enstpricht. Die Brille bietet ein Sichtfeld von bis zu 110° bei einer Bildwiederholrate von 90 Hz [10]. Alle technischen Systemvoraussetzungen können in Tabelle 2 eingesehen werden.
Zur Positionsbestimmung im Raum wird die Lighthouse-Technologie [3] von *Valve* genutzt. Zusätzlich sind neben einem Gyroskop auch ein Beschleunigungssensor und ein Laser-Positionsmesser verbaut. Mittels proprietärer Hand-Controller wird bei der *HTC Vive* eine Interaktion mit virtuellen Objekten ermöglicht.

3.2 Software

Britta

3.2.1 Unity

Britta

@Britta: Vielleicht möchtest du das nur auf Englisch übersetzen?

Unity ist eine sogenannte Spiel-Engine, also eine Entwicklungs- und Laufzeitumgebung, die speziell auf die Entwicklung von 3D-Spielen ausgelegt ist. Die Software wurde am 6. Juni 2005 veröffentlicht [9] und wird von *Unity Technologies* [25] entwickelt und vertrieben. In der Spieleentwicklung ist *Unity* weit verbreitet, so werden beispielsweise 34 % der kostenfreien Top-1000-Spiele im mobilen Sektor mit *Unity* entwickelt [27].

Unity bietet eine sehr breite Plattformunterstützung [26] und erlaubt ebenso die Entwicklung für Head-Mounted Displays, wie etwa die *Oculus Rift* [28] oder auch die in diesem Projekt verwendete *HTC Vive* [28].

3.2.2 Visual Studio 2015

Britta

@Britta: Vielleicht möchtest du das nur auf Englisch übersetzen?

Micosoft Visual Studio 2015 ist eine verbreitete integrierte Entwicklungsumgebung (IDE), welche unter anderem die Programmiersprachen Visual Basic, Visual C#, und Visual C++ unterstützt. Mit Hilfe dieser IDE kann ein Entwickler Win32/ Win64 Anwendungen sowie Web-Applikationen und Webservices [19] programmieren und anschließend kompilieren. Für *MArC* wurde mit der Version 14.0.25123.00 Update 2 gearbeitet.

3.2.3 Steam VR

Britta

@Britta: Vielleicht möchtest du das nur auf Englisch übersetzen?

Steam VR [23] ist die Schnittstelle zwischen der *HTC Vive* und *Unity*. Um das HMD nutzen zu können, musss *Steam VR* auf dem Computer installiert sein. Für den Nutzer ist ein kleines GUI Element auf dem Monitor sichtbar, welches den Status der Geräte der *Vive* darstellt. Hierdurch werden Fehlermeldungen kommuniziert, Kalibrierungen durchgeführt und eine Kommunikation mit dem HMD bereitgestellt, so dass das Gerät im Fall der Fälle neu gestartet werden kann.
Innerhalb von *Unity* stellt *Steam* ein Plugin zur Verfügung, welches direkt in Szenen in *Unity* eingebettet werden kann. Der Entwickler ist also in der Lage, eine vorhandene *Unity*-Szene um die VR Möglichkeit bequem per Drag-and-drop-Technik zu erweitern.
Das bereitgestellte *Unity*-Prefab beinhaltet alle notwendigen Elemente um mit der Hardware kommunizieren zu können. Dabei wird eine Positionsbestimmung ebenso wie ein Kamera Rig für die stereoskopische Bildwiedergabe bereitgestellt, wie auch die Controllereingabe und Weiterverwendung der Daten möglich gemacht.

4 System

Laura

The actual application can be divided into two types of VR rooms. First of all there is a learning room (compare section 4.1.1), where the user can get in touch with the different interaction methods and afterwards different tasks will be presented to him in a VR supermarket scenario (compare section 4.1.2). In the learning room the user will be supported in his learning process by a selfteaching system (compare section 4.4), which he can switch on and off, when he is in the supermarket. The user can make this setting among all other settings in a Menu (compare section 4.2), which is controllable with the *HTC Vive*-controller. In this menu he can also chose between all provided interaction methods, which are described in section 4.3.

4.1 VR Labor

Anna

In this section the different rooms of the VR Labor will be described.
There are two different kinds of rooms. First there is a simple room to get familiar with the system and the methods, which were implemented. This room will be specified in the section 4.1.1.
The second room is created to look like a small supermarket. Here the user has to solve different little tasks. This room will be specified in section 4.1.2.

4.1.1 Learning room

The learning room is the first room where the user will experience the system. On one side the user will get familiar with the virtual experience and the HTC Vive system (HMD and controller). On the other side the user will learn the different interactions.

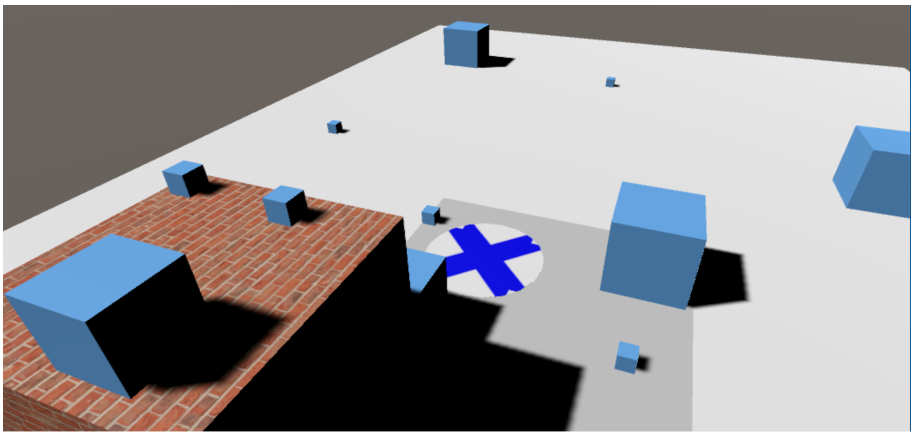

Abbildung 1: Overview of the learning room.

The room is designed very clean and simple. In different sizes and distances simple objects, in this case cubes, are distributed randomly, shown in Figure 1. The virtual room is slightly bigger than usable room in reality. So the user is forced to user interactions, which are designed for bigger distances, to move some objects. Due to different colors of the floor the user know how big the real room is approximately. Also the HTC Vive system offers a coloured wire, which is shown in the HMD, if the user gets close to the border of the calibrated area.
The user should already know the different interactions as well as the menu settings, when he/she enters the supermarket. So the menu and interactions are similar in both rooms. Also the labelling of the target object and the reaction of the target area to the target object will be established. This will be described in section **??**.
To get to know all settings and methods the user will be led to the entire learning room by a selfteaching system. This selfteaching will be described in section 4.4.

4.1.2 Supermarket

The second room is a small supermarket.

Abbildung 2: Overview of the supermarket.

Like it is shown in figure 2 in this supermarket are different sized objects in various distances. All objects are things that could be found in a real object, from fruits to milk. This objects were used from the asset store. [21] [15] [13]
In the supermarket is the size of the real room also shown by the color of the floor. The supermarket is bigger than the real room. So interactions for far distance have to be used according to the tasks.

Abbildung 3: Closer view of the shelves.

In some tasks the participant has to pick objects which are hard to pick. On one side they could be at a lower position, on the other side they could have to move other objects before they reach the labelled object. To implement this into a supermarket naturally shelves, like in figure 3, are appropriate for a supermarket. This shelves were build with different scaled cubes.

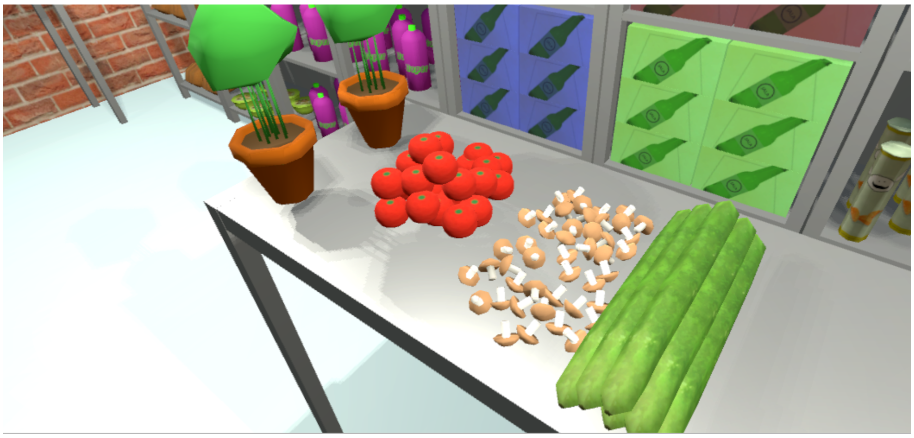

Abbildung 4: Closer view of the table.

In the real supermarket objects are not always in shelves. Also some objects where placed on tables as one bulk, for example fruits and vegetables. This is implied with two tables in this supermarket. The closer look to one of this tables is shown in figure 4.
Depending on the tasks different target areas can appear in the supermarket. In this situations different objects will be blend out. Also the target object of every tasks will be labelled. The first target area and target object are shown in figure 2.
The structure of the labelling of the target object are specified in section 4.4. The different tasks as well as the reaction of the target area to the target object will be described in section 5.1.

4.2 Controller Menu

Anna

The controller menu is the main menu of the system to change settings without leaving the virtual environment. Within this menu interaction methods can be changed and settings like snapping for the interactions can be enabled. In the learning room the user is able to turn the selfteaching informations on and off. In the same way the user can decide if the tasks will be shown or not in the supermarket. In the supermarket is also the option available to start the task again. With the reset button the scene will be reloaded. The main menu is shown in figure 5.

Abbildung 5: Main menu

If the button "select grab" is available, the user is able to choose every interaction method he or she likes. The button leads to an other menu. In this menu the methods are shown as icons, see figure 6. The user learns the meaning of the icons by the selfteaching in the learning room.

Abbildung 6: select grab menu

For the measurement, see section 5.2, the user has to start and stop every tasks. The start button will be the first menu button in every scene. Only if this button

is pressed the user can interact with the environment and will be led to the next menu. When the user presses the stop button the switching area will appear and the corresponding box collider will be enabled. If the user reaches this area the next scene will be loaded.
The structure of the menu is built on the Plug-in "VRTK" (Virtual Reality Toolkit). [18] [17]
This is a free packet with different scripts and prefabs for interaction in the virtual reality. From this plug-in the radial menu is used for the controller menu. This creates menu buttons over the touchpad of the controller of the Vive. By scrolling over the touchpad the button can be selected. The selection will be shown by highlighting the button. If this button should be executed, the touchpad has to be pressed when the button is selected. The number of buttons per menu can be changed within the Unity editor. Each button can get a different icon. Through a link to a function of a script this function can be executed if the button is pressed. In this project there is a script named *Menu* in which all functions for the menu are collected. Due to this script different settings can be done.

4.3 Interaction Methods

Laura

Of course there were various different interaction methods required to make the *Interaction Lab* suitable for the testing described and evaluated in section 5. Also all interaction methods are implemented to realise the grabbing of virtual objects, they can be separated in the two categories, described in the next two paragraphs:

Close Range (CR) Interactions: The CR can be interpreted as a synonym for the natural interaction radius of the person. Due to this definition it is excluded that those interactions can be used outside an area, which the person can not reach with his arm, or to be more precise: with the controller in his hand. In other words: the CR combines all interactions which can be used to pick up objects in the direct reach of the user.

Far Range (FR) Interactions: Due to a limitation of the range of motion in VR applications, it is common to have grabbing interactions, which allow the users to grab objects which are normally seen as out of their reach [12]. Interaction methods allowing such an acting are called FR interactions. It is not excluded, that a FR interaction is used in the actual reach of the user.

Whereas the CR interactions differ mainly in the accuracy of the selection of an object while grabbing it, the FR methods differ in their usability. All characteristics of the various interaction methods can be traced in their descriptions (compare sections 4.3.1 - 4.3.6).

For a better understanding it should be mentioned, that all interaction methods can be controlled with the *HTC Vive*-controller. Even there are plenty of different possibilities to grab an object, all methods have in common, that the grabbing is caused by pressing the trigger on the *HTC Vive*-controller. The releasing of the object is than triggered by letting it go. Whenever there is a divergent Usability necessary, it is described in the respective section (compare 4.3.5).

Due to an easier integration into the learning room (compare section 4.1.1), as well as the actual supermarket scenes (compare section 4.1.2) all methods using a ray (compare sections 4.3.5, 4.3.4 and 4.3.3) are summed up in one script called *AllRaycastMethods.cs*. All other methods have their own script, in which the grabbing and releasing is implemented. Also the *Raycast Head Mounted Display*-method, described in section 4.3.6, is using a ray, it is not included into the script mentioned above. This is caused by remaining problems during the implementation of this method, which lead to an unfinished work. Further explanations on why this method is not available in the *Interaction Lab* can be found in the according section.

The two interaction methods, described in sections 4.3.1 or rather 4.3.2, can be used with snapping or without it. This technique is used to reassign the position and orientation of a grabbed object in hand. By assuming that the middle of the ring of the *HTC Vive*-controller is the new center of the grabbed object, the actual grab could appear more realistic to the user.

In the application all objects, which can be grabbed are tagged as moveable.

In the following sections all available interaction methods of the *Interaction Lab* are presented. To guarantee a better overview they are sorted by their interaction range.

4.3.1 Close Range: Touch Grab

Laura

When this interaction method is selected, the user can make use of the *HTC Vive*-controller to pick up objects directly by touching them and pulling the trigger. To release the object the trigger needs to be released as well. An object can be grabbed, whenever it is tagged as moveable and collides with the *HTC Vive*-controller. This collision is detected by giving the object a collider, which fits its form best [4][6] and applying a *BoxCollider* to the controller. The interaction can be seen in figure 7.
In the script *TouchGrab.cs* in which the interaction method is implemented is checked frequently, whether there is a overlap of the collider of the controller with the collider of a moveable object or not. Whenever they collide, the respective object is coloured green to show the user, that he could grab it by pulling the trigger.

Abbildung 7: Grabbing a virtual Object by using *Touch Grab*.

4.3.2 Close Range: Proximity Grab

Laura

When it comes to the CR interactions the *Proximity Grab* is by far the most inexact selection. Grabbing and releasing and object are realised by using the trigger of the *HTC Vive*-controller.
The functionality is provided by the script *ProximityGrab.cs*. The basic idea is that the object can be grabbed, whenever the object triggers the *BoxCollider* [4] placed at the end of the *HTC Vive*-controller. A more detailed description can be found in section 4.3.1. In contrast to the *Touch Grab* described in section 4.3.1 this *BoxCollider* is bigger than the actual size of the controller. To show the user which object collides with the controller, and can therefore be grabbed, the respective object is coloured green, as shown in figure 8. The small gap between the actual grabbed object and the controller shows the difference between the interaction method shown in figure 7.

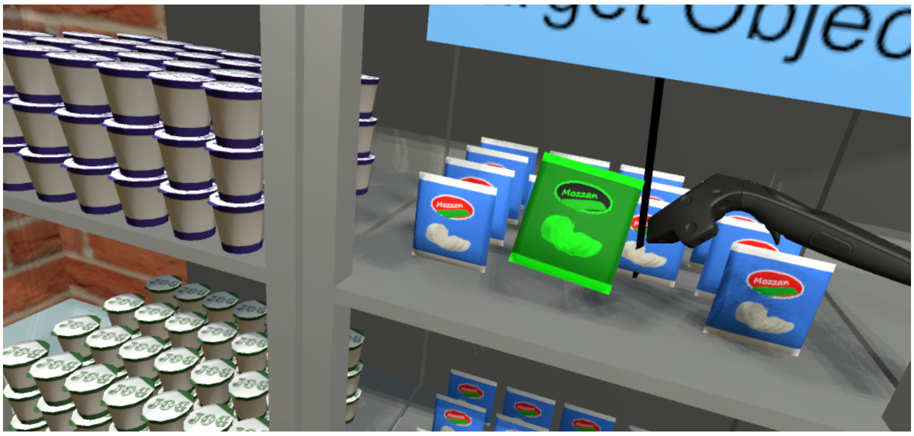

Abbildung 8: Grabbing a virtual Object by using *Proximity Grab*.

4.3.3 Close Range: Wand Grab

Laura

In contrast to the interaction method described in 4.3.2 this method can be used to grab very tiny objects. Thereby it is not needed that the target object is very isolated from other objects. To give the user such an high grade of accuracy a stick is added to the controller like shown on figure 9. The user can grab an virtual object he touches with the *HTC Vive*-controller by pulling the trigger. To place the object on the target area he simply release the trigger after he moved the object to its destination.
The implementation can be found in *AllRaycastMethods.cs*. The wand consists of two elements: a ray [5] and a cube. The cube is only for the visualisation and has a fixed size in all three dimensions. The collision detection, which is necessary for the actual grabbing, is done with the ray. That means, that an object can be grabbed, if the ray which has the same dimensions like the cube, touches this specific object. The cube will then turn from black to green to show the user, that there is an object, which can be grabbed.

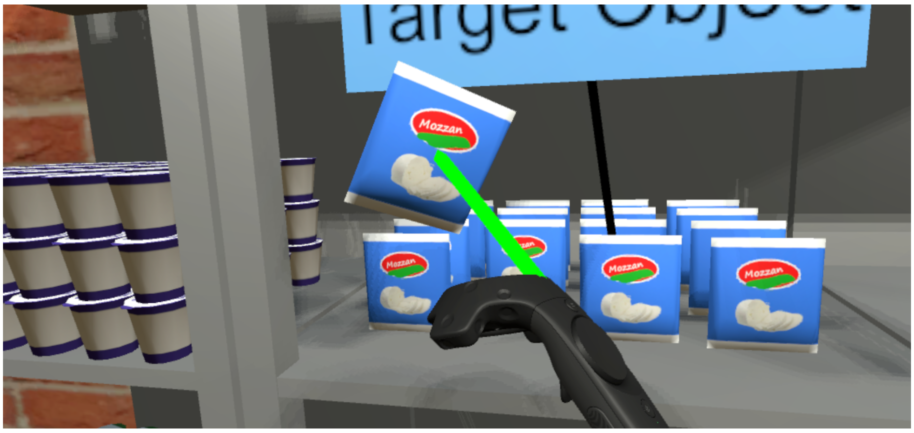

Abbildung 9: Grabbing a virtual Object by using *Wand Grab*.

4.3.4 Far Range: Raycast

Laura

By using the *Raycast* method, the user can grab virtual objects, which are further away, as well as objects in his CR. As shown in figure 10 a ray is coming out of the *HTC Vive*-controller pointing away from the user. At the end of the ray is a small sphere, which turns green, if it collides with an moveable object. Whenever the ray hits an objects, like for example the floor or a product in the supermarket (compare section 4.1.2), the ray is shortened to the distance between the controller and the respective object.
The implementation can be found in the *AllRaycastMethods.cs* script. As already explained in section 4.3.3 a cube and a ray are combined to reach the intended

functionality. In contrast to the *Wand Grab*, the ray and the visible cube have no fixed length and there is a sphere added to the end of the cube.

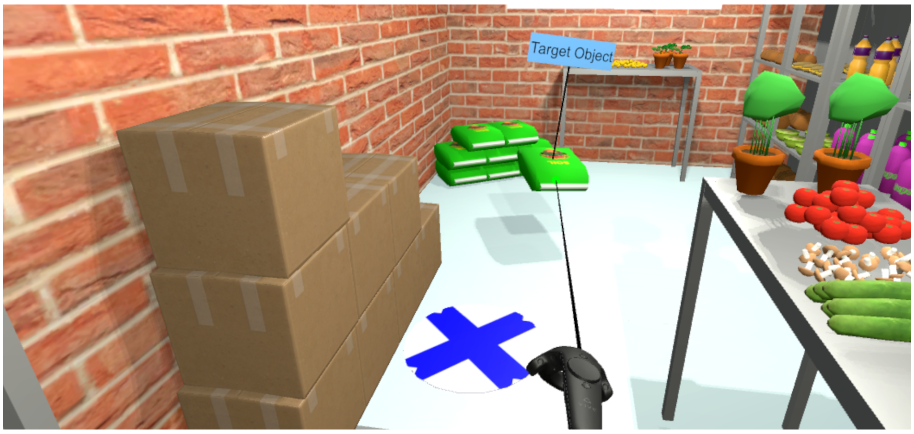

Abbildung 10: Grabbing a virtual Object by using *Raycast*.

4.3.5 Far Range: Extendable Ray

Laura

The actual ray is build as described in section 4.3.4. The ray [5] is complemented by a cube with a sphere at the end, to make it visible for the user. In contrast to the normal *Raycast* method, the length of the ray is set to a start value of 3 meters. The user can shorten and lengthen the ray by pressing the touchpad of the *HTC Vive*-controller in the lower or rather upper area. This subtracts or adds a constant value to the length of the visible ray. What remains is the behaviour of the sphere, which turns green, whenever a moveable virtual object is brushed. The *Extendable Ray* is one of the three interactions methods (compare sections 4.3.3 and 4.3.4), which are combined in the *AllRaycastMethods.cs* script. This interaction method is shown on figure 4.3.5.

Abbildung 11: Grabbing a virtual Object by using *Extendable Ray*.

4.3.6 Far Range: Raycast Head Mounted Display

Laura

It was planned to realise an interaction method, where there is a ray coming out of the HMD, which can be used similar to the method described in section 4.3.4. Due to the low accuracy of this method, it did not become a part of the *Interaction Lab*. There was a try to parent [7] the position of the HMD to the starting point of the ray. It turned out, that the parenting is not successful, when it comes to the position and rotation of the HMD. Even this method was effective for adding a ray to the *HTC Vive*-controller, there is a irregular shift when you try to implement it with the HMD. To proof that an easy example (compare source code 1) was observed. In this example a simple cube should be rendered at the front of the HMD. In reality this cube was rendered in a position, which can be seen as random.

```
cube.transform.SetParent(HMDEye.transform);
```

Source Code 1: Test on the parenting of *HTC Vive*-HMD and an virtual object.

All effort on this method, can be found in the script *RaycastingMethodeHMD.cs*.

4.4 Self-Teaching

Anna

To introduce the user to the system there will be a teaching in the learning room. When the program will be started informations will be shown next to the controller, like in the following figure.

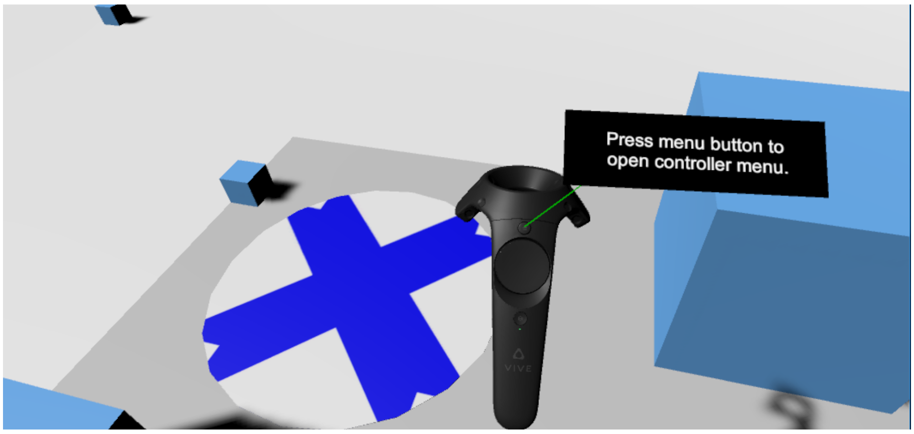

Abbildung 12: Start of the selfteaching

This informations led the user step by step to the system. He/She will get to know how the interaction methods can be changed and what settings can be made.

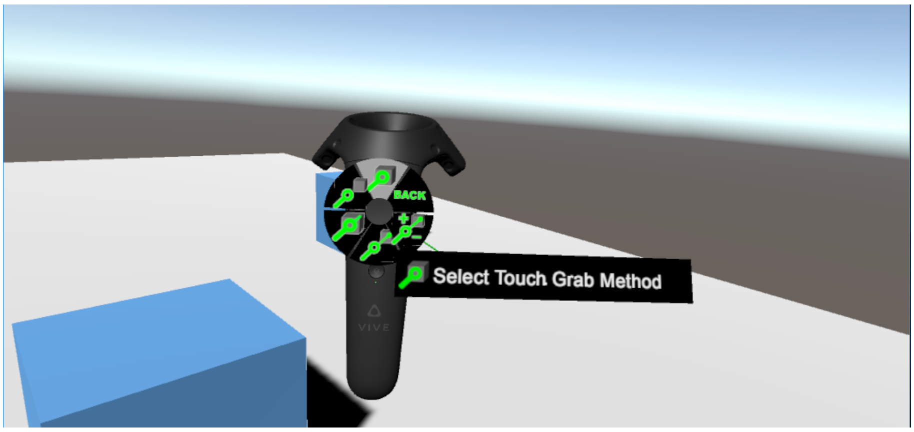

Abbildung 13: Selfteaching recommends to select touch grab method

This informations will just be shown in the learning room. The position of the information area changes depending on which button is important for the interaction. In the first grab method, the touch grab method, also two terms will be established which are important for all tasks. This terms are the labelling of the target object and the target area. Here the user learns how the target object is marked, see figure 14, and how the target area reacts to the target object.

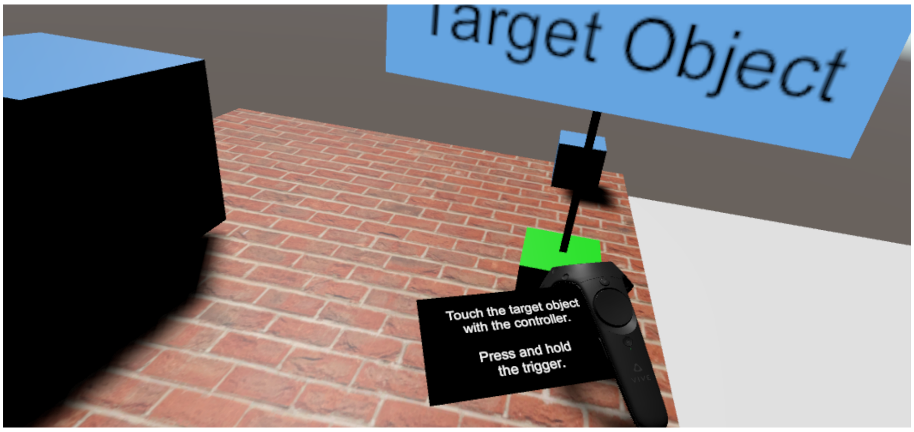

Abbildung 14: Selfteaching with target object

All informations and steps of the selfteaching are saved in an external CSV-file. Due to an additional script *WriteMeasureFile.cs* this file will be read line after line and be saved into different lists. In each line are three main informations. First there is the text which should be shown in this step. Second is the Button ID, which defines on which button the information should be fixed. Third and last is the hight of the information area. This hight changes depending on the amount of informations.
In the *SelfTeaching.cs* script is a counter implemented. This counter controls which line of the file will be displayed. This counter will be increased or set from different scripts depending on an action. This increasing is still a problem if the user does not follow the selfteaching exactly. For example, if the user grabs a object twice and the selfteaching plans to grab just once, the selfteaching will go on and the user misses

information.
The information canvas is also a prefab of the “VRTK” plug-in. Here they are called controller tips. This creates a area next to the controller and a line from this area to the selected button of the controller. This area is parented to the controller and moves according to it. Different settings can be changed within the Unity Editor or by script.
The “VRTK” plug-in has also a prefab for object tips. This is also a canvas in which information can be display but this can be fixed on any object you like. With this canvas the target objects are labelled.

5 Evaluation

Britta

5.1 Tasks

Anna

In the supermarket are four different tasks the user has to do. In every tasks a object has to be selected and placed on a target area. If the correct object is placed, the target area will change the color to signal that the tasks is successful done. In the script *TargetTest.cs*, which will be a component of the target object, will be recognized when the target object hits the target area. At this moment the texture of target area will be changed and the measurement will be stopped. More information about the measurement will be mentioned in section 5.2.
In the first three tasks the user has to select different objects. In this tasks the user decides which methods he/she uses. The selected methods should differ depending on the tasks. In this tasks objects far away as well as closely should be picked.
The tasks will be shown on the controller similar to the selfteaching which the user is already aware of.

Abbildung 15: Task shown on the controller

The text for the tasks are saved in a CSV-file, similar to the selfteaching, see section 4.4. The implementation is also related to the selfteaching and implemented in the script *showTasks.cs*.

Abbildung 16: Additional informations on the wall

To give the user some more information a information board within the supermarket is established, see figure 16. In the first tasks there are only some basic informations shown.
The last tasks will be repeated with every method. That means, that the user has to pick up the same object five times. This object is so placed that the user could use close as well as far range methods. The methods are fix implemented and already activated as soon as the user presses start.
So help the user to figure out what method will be activated the name of the method will be shown on the information board as soon as he/she is in the new scene, like in the following figure.

Abbildung 17: Touch grab is activated

In the last tasks after every method the user has to answer an usability questionnaire about this interaction before he/she does the tasks with the next method.

5.2 Measurement

Vera

6 Project Management

Vera

This chapter describes the project planning and management of the *Interaction Lab*. It is divided into the different project phases. Each division includes all important facts of its project period.

6.1 Project Definition

Vera

This section describes the results of the project definition phase in detail. This includes a problem analysis, a list of objectives and requirements, a solution concept as well as a workability analysis.

6.1.1 Problem Analysis

Vera

The demand for Virtual Reality (VR) devices and applications increased heavily since the first consumer devices like *HTC Vive* and *Oculus Rift* were released during last years. One main difficulty of the current development of VR-applications is the lack of standardisation of the Software Development Kit (SDK) and interfaces. The most acknowledged suppliers *HTC* and *Oculus* do not work together or force standards for VR application development. Thus, all applications are system related and incompatible with other devices. Accordingly, each device offers different opportunities of interaction methods. These methods could be divided in the acknowledged categories selecting, grabbing, manipulating, movement and indirect controlling via widgets, gestures and voice input. Several suppliers currently offer different devices for interaction. And with focus on the grabbing and positioning methods, the most common are the *Oculus*-HMD, *HTC Vive*-HMD, *HTC Vive*-Controller, data gloves and motion capturing systems for hand-tracking like the *LeapMotion*-Controller.

As mentioned in section 1.1, there exist currently no interaction laboratory which compares the different interaction methods in a scientific and credible way. Hence, the development of a virtual laboratory is highly requested to compare and test different interaction methods in adequate test environments. Thus, user friendly interaction methods which nearly full-fill usability requirements could be improved by researcher which yields to a higher demand of VR devices and application. That will squeeze the profit of VR device suppliers which include those user friendly interaction methods.

6.1.2 Usage Context

Vera

Hence, the required laboratory has mainly two usage contexts. First, it could be used to run scientific studies in VR research or development. Second, it could demonstrate and exemplify the differences of grabbing interactions in education proposes or support the students to develop and test grabbing methods on their own during lectures.

6.1.3 Objective and Requirements

Vera

At least two scenes should be realised to provide a laboratory which allows to run scientific and reliable study as well as is useful for the education of students. In the first scene, the user will be able to learn the offered grabbing methods. Therefore, this room is will provide simple cubes of various sizes which are in different distances from the users. Every cube is moveable and could be placed at every place. Each user is forced to follow the introductions of a self teaching before every offered method could be tested independently. The current user can only begin with the actual study after every method is trained to ensure equal preconditions.

The second room will be modelled after a supermarket because this model offers various options of grabbing and positioning tasks. In this room, the participant will get different tasks which will differ by complexity, distance of grabbing and size of the objects. The user will be able to change the options of grabbing independently but not choose the current method. An optional extension of the project will be another type of task where the user decides which type of method is preferred for this task.

The grabbing methods can be categorised into close range and far range and include the grabbing, rotating, and positioning of an object. Possible types of close range methods, are the actual touching of a movable object to select it or by holding the controller in the proximity of it but without touching it. Another more precise option is the selection with a thin wand in front of the controller. This of collection of methods that includes close human cognition methods as well as less or very accurate ones. The far range interaction will have different options as well. One will be a ray that shoot out of the controller, another one will extend a ray from the head and the third one will extend the arm in the pointed direction. This means the user will be able to point at an object with the controller or to look in the direction of it.

The system offers two measurements and the related saving of the different parameters. First, the duration time is measured for every performed task to compare and validate the performance of the different interaction methods. Second, every single grabbing try of a task will be counted and saved to get a conclusion about the learn-ability, accuracy, and performance.

Furthermore, there will be a questionnaire designed to give the users of *Interaction Lab* an usability evaluation tool at hand. This questionnaire will test parameters as tiring, learnability, self-descriptiveness and fulfilling expectations.

6.1.4 Solution Concept

Vera

An interaction laboratory for grabbing and positioning interactions at close or far range will be developed in *Unity*. It includes two test rooms e.g. scenes, where the first is a learning room, in which the users can get familiar with the interaction methods. The second room is designed as a supermarket. This environment was chosen because it offers various possibilities of exercises under changing difficulties like grabbing small mushrooms, fetching distantly placed tins or putting goods on provided target areas. The exercises are offered in form of a tasks that tells the participant what goods have to be grabbed and repositioned. These various tasks are predefined and cover all difficulties that a type of grabbing method could have. They are displayed on tables which are connected to the controller and could be shown or hide in the controller menu.

All rooms are implemented in Unity and the VR components are controlled by the same framework. Further, the *HTC Vive*-HMD and the corresponding controllers are used to run the interactions, imaging and orientation in the environment. It is planned to realise at least six interaction methods of grabbing and positioning. Additional, the complete framework should be compatible with new test scenes and other interaction categories.

The system offers a measurement of the accuracy as well. A time measuring of duration and an error rate for every performed task is planned. Each measuring of every room is automatically saved in an output file which could be easily imported in common statistic tools. Furthermore, there will be a validated questionnaire designed to give the users a usability and simulator sickness evaluation tool at hand. This usability questionnaire will test parameters as tiring, learn-ability, self-descriptiveness and fulfilling expectations of each method. Whereas the simulator sickness evaluation asks for motion sickness and other system properties of the complete system. All questionnaires are validated and fit the requirement of VR systems and application. The results of each questionnaire will be saved in an output file as well.

6.1.5 Workability Analysis

Vera

There are several risks according to the concept in section 6.1.4. First, the measurements could be implemented incompletely or inaccurately. This can be avoided by a thorough testing before the final release with some external test persons. The tasks could be incomprehensible for them as well which should be prove as well. The system integration future extensions could cause trouble. Therefore, the systems architecture should be designed wisely and consequently to avoid incompatibilities. Another risk of the implementation is that they might be more costly and complex as recommended but this is widely acknowledged. After the implementation is finished the interaction method performance or validation could be too expensive which results in a higher latency. These circumstances must be observed during the implementation and testing. Due to the high workload of the testing, the time slot for it and the trouble shooting might be underestimated. Another time risk is that there is limited access to the facilities and VR laboratory because of the huge number of running project at the current time.

Nevertheless, the concept is feasible and the project goals could be achieved during the time schedule because all the risk seems to manageable and could be observed during the scheduled testing.

The demand of the students project are satisfied and a financial profitability check is not necessaries due to the fact that the facilities of the university can be used and no further purchases are affordable.

6.1.6 Project Organisation

Vera

The project manager is Vera Brockmeyer who mainly should manage the appointments and facilities as well as to communicate to the outside. The latter is done via email or in a meeting with the concerned persons. Another task is to create and maintain the project plans that includes to keep the overview of the complete project progress and to ensure the milestones. The current state should be hand out weekly to the team in form of an email or a team meeting.

All other team members have their own responsibilities. Anna is head of the scene building which includes the definition of the general scene design, research, and to inform the project manager about current problems and timing. The latter two points concern each head of a section. The other section is split into the close and far range interactions. The head of close range interaction is Britta Boerner and the other is Laura Anger. Both manage the implementation of their section.

The formal an informal non-verbal communication in the team is done via email with the subject *VR Interface Lab* and a *Google Calendar* is exclusively maintained by the project manager where all team appointments are intercalated. This calendar shows the availability of all team members and the VR laboratory, too. More complex problems or team decisions are made in the weekly team meeting with stringently required appearance. Due to the requirements and availability of the team members, the meeting is held via *Skype* or in personal.

All files belonging to the project are organized in a cloud folder of *Google Drive* or in two *GitHub* repositories. The first one is for all *LaTex* files and the second manage the complete framework. Whereas the cloud folder contains presentation files, graphics and images, To-Do-List, papers and more.

The required facility is one of the VR laboratory of the faculty which should have a minimum size of 15qm and be located in the university building. These laboratories have a complete *HTC Vive* system and a compatible computer (see Table 1).

6.2 Project Planning

Vera

This section list all required projects plans like a work breakdown structure, the work packages, a capacity and cost plan as well as a quality plan.

6.2.1 Work Breakdown Structure

Vera

The work breakdown structure (see Figure 19) is split in four groups. First, the project planning which contains the actual project management as well as the research work packages Second, the implementation group includes the development of the laboratory environment and the six interaction methods. Third, the evaluation contains the work packages of tasks including measurements and the questionnaires. Finally, the last group includes the documentation and project profile packages.

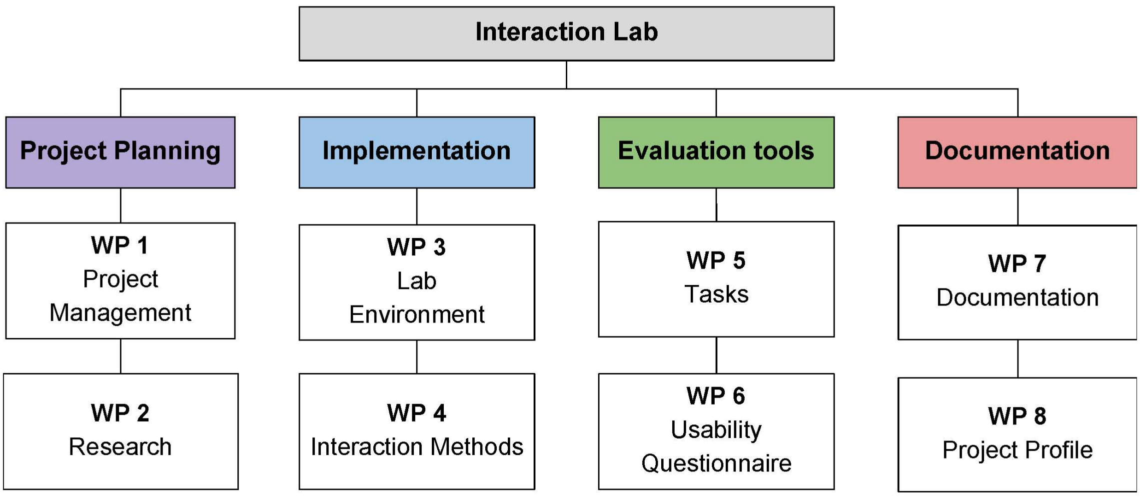

Abbildung 18: Work Breakdown Structure

6.2.2 Workpackages

Vera

A complete listing of the work packages according to Figure 19 could be seen in the Appendix A 10.1.

6.2.3 Project Schedule

Vera

Figure 19 shows the project schedule with the timing of all work packages. The most important deadlines are the project plan, research, first and second prototype as well as the release deadline. First, the project plan should be finished until 13th April 2017. Second, the research deadline is two and a half weeks later at the 27th April 2017. Followed by both prototypes which are set for the 31st may and 30th June 2017. The final release deadline is at the 15th July 2017.

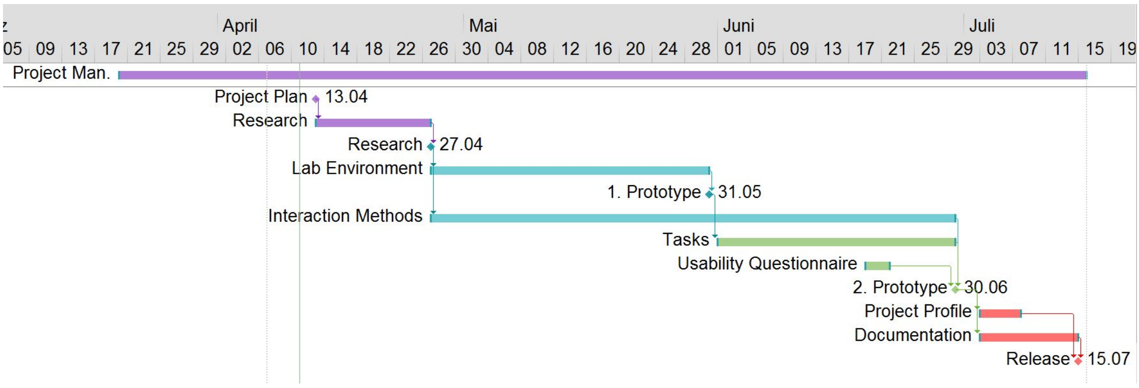

Abbildung 19: Project Schedule

6.2.4 Capacity Plan

Vera

The capacity plan lists the estimated time resources of each team member, the availability of the required laboratory with the *HTC Vive* system and the SLR camera for the documentation.

Resource	Estimated [h]	Actual [h]	Buffer [%]
Laura	166	180	7.78
Anna	170	180	5.56
Vera	182	180	-1.11
Britta	168	180	6.67
Total Personal	686	720	4.72
VR Lab 1	173	152	-13.82
SLR Video Camera	8	8	0.00

Tabelle 3: Capacity Plan of personal and facility resources.

6.2.5 Cost Plans

Vera

The cost plan of personal resources in Table 4 gives a more detailed overview of the estimated time resources per work package. It is structured into the cost of each team member and the total cost of a work package. Whereas, the cost plan of the facility resources offers an overview of the material costs and their providers.

Name	Laura [h]	Anna [h]	Vera [h]	Britta [h]	Cost WP [h]
Project Management	12	12	45	12	81
Research	10	10	10	10	40
Lab Environment	0	55	32	10	97
Interaction Methods	65	0	0	65	130
Tasks	15	45	30	15	105
Usability Questionnaire	8 0	0	0	8	
Documentation	45	45	55	45	190
Project Profile	8 0	0	8	16	
Total Cost	**163**	**167**	**172**	**165**	**667**

Tabelle 4: Cost Plan of personal resources.

Ressourcen	Quantity	Unit Price [EUR]	Cost [EUR]	Act. Cost [EUR]	Comments
HTV Vive	1	899	899	0	Price of Vive Online Shop, provided by TH Cologne
SLR Camera	1	500	500	0	estimated, provided by team member
VR Computer	1	1800	1800	0	offer from Computer Shop, provided by TH Cologne
MS Office	1	149	149	0	provided by TH Cologne
MS Project	1	195	195	0	provided by TH Cologne
3D assets of Food Beverages	2	15	30	0	Steam
Total			**3573**	**0**	

Tabelle 5: Cost Plan of facility resources.

6.2.6 Quality Plan

Vera

Quality Goal	Criteria	Method	Controlling
Immersion and Presence	no distraction by real world	Presence questionnaire by Witmer and Singer; questions according to sound and haptic will left out	use and evaluate brief questionnaire during functionality test
Low Latency	under 20ms	keep the scenes as simple as possible	show the frames per second in the Unity console
No dropouts	no black frames or errors in the unity project	no expensive or parallel calculation	visual testing
Understandability of tasks	correct task performance by test user during functionality tests	use common objects and tasks; brief and precise task descriptions	use brief questionnaire during functionality test
robust system	Vive System does not crash during process	no expensive calculations; control and calibration of hardware; analyse and solve directly problems	observe during development
Correct saving of Usability Questionnaire Response	entire answers of study participants are correctly saved in an file	Use of Google Forms	functionality test and controlling of completeness and correctness
Understandability of Usability Questionnaire	no questions or uncertainties of test users	validated Usability Questionnaire	use and evaluate brief questionnaire during functionality test

Tabelle 6: Quality Plan

6.3 Project Execution

Vera

The following sections summarizes the project progress as well as the problems that caused a delay of the schedule.

6.3.1 Project Progress

In general, the project progress went off quite well and most of the project goals were achieved and are running as expected. Therefore, the *Interaction Lab* is a complete and running application that includes all required tools for a scientific study. The final system contains a learning room, a supermarket test scene with implemented task and measuring, the implementation of five different interaction methods to test, completely set of validated usability questionnaires and excel templates for the evaluations.

However, some of the project goals were too ambitious for the resources capacity in the scheduled time window. Thus, not all project goals were completely achieved. First, the Gogo interaction method and the HMD raycast could not be developed due to the problems mentioned in section 6.3.2. Second, the number of tasks needed to be reduced significantly because it turned out that they are too time consuming during testing. Hence, only one task was implemented with all provided interaction methods and three tasks to figure out the preferences of the users for various kind of ranges and object sizes. Third, an additional goal was set after the first prototype. A self teaching mode has been added to the learning room to ensure an equal introduction of the methods and task for each test user.

The actual workload of most work packages covers their estimation beside some problems which are described in detail in section 6.3.2. All team members had a similar work load and the task allocation work efficient and well. Each team member knew their current tasks at every time and had an overview of the complete progress. This overview was achieved with weekly meetings via *Skype* or if affordable in personal. Furthermore, every few weeks we had a meeting with Prof. Grünvogel to give him an update and discuss current problems and their solution.

6.3.2 Problems and Solutions

Vera

A first problem was the additional work effort because all project plans needed to be overworked which causes a one week delay. In the end it should also be noted the planning was to detailed in the beginning which cause some unnecessary time. Thus, the research phase need to shorten and place on hold for this week but this step brought the project back on time. Another delay of one and a half week causes the inevitable unity and *HTC Vive* system that were more difficult that expected. This kind of problem occurs several times when changing the laboratory due to limited capacities of the designated VR Lab 1 or the periphery of the laboratories were not present. Additional, every time the VR systems needs to be recalibrated when changing the laboratory.

The VR system causes trouble during the implementation of the HMD Raycast method, too. It turned out that the tracking of the HMD lacks of precision and the ray drifts at the border of the view field. At this point, the number of tasks that must be done to guarantee a working application until the release was challenging. Thus, we concentrate on a working application and discard the HMD Raycast. However, it seems that the implementation of the Gogo method was too complex for this time schedule, too. Hence, it was changed to the Extenable Ray method which could be implemented successfully during the calculated period.

This decision and the resulting choice of methods had the advantage that their basic structure is adoptable. Regarding to this realisation, it could be said that the split of responsibility for Far and Close Range in the project organisation was a mistake. Accompanied with the decision to create a single C# script per method. This plenty of scripts caused a lot of trouble during their integration in the final application but these problems were solves by using only one script for similar methods like the ray methods, for example.

7 Reflexion

Vera

Finally, a running and nearly complete application was developed during this project. The test environments in form of the learning room and supermarket are working without a latency or blackouts. They provide an effective virtual laboratory that could be used by researchers or for educational proposes. In both environments are five perfectly working grabbing interaction methods integrated. First, the close range methods Touch , Proximity and Wand Grab where the first two had an optional snapping mode are implemented. Second,the two far range methods Raycast and Extenable Ray are successfully integrated into the application.

All measurements of the tasks are saved automatically in output files that could be evaluated with the provided *MS Office Excel* template. The usability questionnaire forms could be accessed by every mobile device or computer and their results are saved automatically as well. They are offered in English and German language. Certainly, they could be evaluated with the template, too. Additional, all required consent forms are delivered with the application as PDF file.

As mentioned in section 6.3.2, the HMD Raycast and Gogo method could not be realised but unfortunately there are more goals that are not yet achieved or do not work completely without faults. First, currently the self teaching mode only works if the introductions were strictly followed. In case of a mistake, it will recover only when a new method is selected. However, there is no reminder of the usability questionnaires during the last task for the study superviser. Therefore, he is forced to be totally focused during a test. Another failure is the missing question of the VR experience in these questionnaires and the lack of the general input of the test id in the application.

There are some technical mistakes as well which are not yet fixed. A main problem is the potential lost of script links in the unity project when the laboratory is changed or a git repository is pulled. A further problem is the lack of an easy method or scene extension in the application due to the complex measuring and interaction script architecture. Other small and rare troubles are the visible ray in front of transparent surfaces or object that fall sometimes through the ground.

8 Self-Assessment

???

Laura:Hier müssen wir uns am besten eine gemeinsame Struktur überlegen, oder? Vielleicht beschreibt jeder was er gemacht und dann eine kurze Selbstrefelktion?

8.1 Anna Bolder

Anna

Working Ours: 116

Written Sections: 2.1, 4.1 (subsections included), 4.2, 4.4 and 5.1

Responsibilities:

My main tasks in this project was the menu as well as the displaying of different informations, selfteaching and tasks, for the user.
At the beginning I created a shared Unity project and added the Steam VR plug-in. I created different rooms for the testing of the close and far range interactions. Also I designed the learning room, described in 4.1.1. Also I discussed with V. Brockmeyer the structure of the supermarket. Additional I researched the options to switch between the rooms and created the basis for the script *TargetTest.cs*.
After the creation of the basic scenes I researched for a solution of an radial menu on the controller. I implemented this with the "VRTK" plug-in. To collect all functions behind the buttons in one script I created the script *menu.cs*. As soon as all interactions were implemented L. Anger and I integrated the interactions in the menu and all scenes. Here I helped to simplify the switching between different interactions with a ray.
In the last weeks I implemented the selfteaching together with V. Brockmeyer. This was time-consuming because the counter has to be increased or set on different parts of the projext and the height of the canvas has to be set for every step.
With the knowledge of the selfteaching I and L. Anger implemented also the tasks in the supermarket. Here we had to add a board on the wall and to short the actual tasks texts.
Also I did the test surveys together with L. Anger.

Self-Reflexion:

In my opinion the project ran quiet well. Everybody had her part of the project and we also helped each other.
I underestimated some parts of my responsibilities. First the selfteaching took much more time than I expected. There were a lot more steps to do than I thought. Second the combination of all interactions in one menu and scene was more complicated. We did not think about that they could affect each other or that parts of the interaction

have to be loaded at the start of the scene to work correctly.

8.2 Vera Brockmeyer

Vera

Working Ours: 170

Written Sections: 1(subsections included), 5.2, 6 (subsections included), 7, 9

Responsibilities:
- Umsonst Fragebögen übersetzt

Self-Reflexion:

+ wenig in falsche richtung + Arbeitsaufteilung sinnvoll und gut - weniger Skype meetings, mehr Rückmeldung mit email - zusätzliche Recherchezeit einplanen

8.3 Britta Boerner

Britta

8.4 Laura Anger

Laura

Working Ours: 97

Written Sections: 2, 2.2, 4 and 4.3 (subsections included)

Responsibilities:

Besides mandatory tasks, like for example doing research on specific problems and helping with the project management, I was the head of far range interactions. It should be obvious that I put my main effort into the various interaction methods of the *Interaction Lab*. As it turned out it was not wise to separate between the interaction methods into two sections and letting two people (B. Boerner and me) work on each part. This is caused by the overlap of the different interactions. For example the *Wand Grab* (compare section 4.3.3), which is a CR interaction is very similar to the FR interactions *Raycast* (compare section 4.3.4) and *Extendable Raycast* (compare section 4.3.5).
To cut a long story short, I implemented the FR as well as the CR interactions together with B. Boerner. A brief description of all completed, as well as one unfinished, methods can be found in section 4.3.
Together with Anna Bolder it I integrated the implemented interactions into the actual scenes (compare section 4.1) of the *Interaction Lab*. We also took care of the visualisation of the tasks for the user. Therefore we created a display on the wall of the supermarket as well as next to the controller. On both of the displays the corresponding tasks are presented for the user.
I was also involved in creating the project video and all images for this documentation.

Self-Reflexion:

All in all I felt like the project run smoothly. Of course I was confronted with unknown terrain, as I never had implemented interactions of any kind. Maybe this was why I first started to create one script per interaction method. In the end of the project I noticed, that both Raycast methods and the *Wand Grab* work after the same rules, so I combined them in one script. If I had done that earlier I could have avoided a lot of trouble for me and Anna Bolder as we needed to integrate them in to all scenes, separately first.

9 Conclusion

Vera

10 Appendix

Vera

10.1 Appendix A

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
1</td><td>WP-Name:
WP 1: "Project Management"</td></tr>
<tr><td>Start:
20.03.2017</td><td>End:
15.07.2017</td><td>Responsible Team Member:
V. Brockmeyer</td></tr>
<tr><td colspan="3">Results:
• Project definition
• Project plans
• Resource and material reservation
• Weekly reports
• Milestone presentation
• Project process evaluation and reflexion</td></tr>
<tr><td colspan="3">Tasks:
• Organisation of meetings
• create project definition, project order, list of requirements, project organization
• prepare milestone and final presentations
• make Work breakdown structure (WBS), Project schedule, Capacity plan, Cost plan and Quality plan
• manage internal and external communication
• give weekly reports
• delegate and supervise tasks
• team meeting to reflex project process, compare plans and suggested cost with actually effort
• organise all required resources</td></tr>
<tr><td colspan="3">Requirements:
• MS Project
• Internet
• team meetings</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Member</td></tr>
</table>

Abbildung 20: First Workpackage: Project Management

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
2</td><td>WP-Name:
WP 2: "Research"</td></tr>
<tr><td>Start:
13.04.2017</td><td>End:
26.04.2017</td><td>Responsible Team Member:
L. Anger, A. Bolder, B. Boerner und
V. Brockmeyer</td></tr>
<tr><td colspan="3">Results:
● Draft of implementation
● knowledge of the required theory
● Measurement methods
● UML of Unity Project</td></tr>
<tr><td colspan="3">Tasks:
● Research Far Range Interaction Methods
● Research Close Range Interaction Methods
● Research Test Scene Implementation and Assets
● Research Measurement Methods and related Tasks
● team meeting to plan integration of total system
● create UML of Unity Project</td></tr>
<tr><td colspan="3">Requirements:
● Project Definition
● Project Plans
● Internet
● team meeting</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Member</td></tr>
</table>

Abbildung 21: Second Workpackage: Research

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
3</td><td>WP-Name:
WP 3: "Implementation Environment"</td></tr>
<tr><td>Start:
27.04.2017</td><td>End:
30.05.2017</td><td>Responsible Team Member:
A. Bolder</td></tr>
<tr><td colspan="3">Results:
● Setup of the Unity Project
● Setup of the GIT repository
● one simple test room to learn and test the different interactions
● one test room build as supermarket to implement tasks for user
● GUI
● First functionality testing</td></tr>
<tr><td colspan="3">Tasks:
● Setting of the concrete Unity version
● Download and integrate SteamVR
● Build simple test room with cubes as big as the real room
● Build supermarket, which is bigger than the actual real room to force the user to use far range interaction
● ensure the user will not walk further than possible
● Implement switching between rooms
● Implement structure and position of the menu
● Implement different buttons and the functions behind
● Integration of the interaction methods
● Implement interface for further test scenes
● upload to GIT repository
● Perform first functionality test with 5 persons</td></tr>
<tr><td colspan="3">Requirements:
● WP 1 Project Management
● WP 2 Research
● Plugins for Unity (Steam VR, 3D assets)
● VR Laboratory 1
● GIT repository
● Supporter: Vera Brockmeyer</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Member</td></tr>
</table>

Abbildung 22: Third Workpackage: Implementation of the Environment

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
4</td><td>WP-Name:
WP 4: "Implementation Interaction Methods"</td></tr>
<tr><td>Start:
27.04.2017</td><td>End:
29.06.2017</td><td>Responsible Team Member:
L. Anger (Far Range) and B. Boerner (Close Range)</td></tr>
<tr><td colspan="3">Results:
● Implementation of Close Range Methods
○ grab by contacting the object with controller
○ controller is in proximity with the object
○ controller is in proximity with the object and object snaps to the controller
● Implementation of Far Range Methods
○ Pointer to grab and control object
○ Pointer from the HMD to the Object
○ extension of the user’s arm</td></tr>
<tr><td colspan="3">Tasks:
● implement scripts for each interaction methods</td></tr>
<tr><td colspan="3">Requirements:
● WP Research
● VR Room 1
● Git Repository</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Member</td></tr>
</table>

Abbildung 23: Forth Workpackage: Implementation of the Interaction Methods

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
5</td><td colspan="2">WP-Name:
WP 5: "Tasks"</td></tr>
<tr><td>Start:
01.06.2017</td><td>End:
29.06.2017</td><td colspan="2">Responsible Team Member:
A. Bolder</td></tr>
<tr><td colspan="4">Results:
● implementation of three tasks
● integration of all task in the system
● completed functionality test with 5-10 persons
● output of measurements
● evaluation of prototype and questionnaire
● solution concepts</td></tr>
<tr><td colspan="4">Tasks:
● implement training in first test room
● implement three tasks in supermarket scene
● design questionnaire of functionality test
● run functionality test
● evaluate questionnaire
● implement solution concepts
● implement final solutions</td></tr>
<tr><td colspan="4">Requirements:
● VR Laboratory 1
● 1. Prototype
● Research
● 5-10 persons
● Test schedule
● brief questionnaire for test persons
● Supporter: Vera Brockmeyer</td></tr>
<tr><td colspan="2">Signature Project manager</td><td colspan="2">Signature Responsible Team Member</td></tr>
</table>

Abbildung 24: Fifth Workpackage: Tasks

Project name: Interaction Lab	WP- Nr.: 6	WP-Name: WP 6: "Usability Questionnaire"
Start: 19.06.2017	End: 21.06.2017	Responsible Team Member: V.Brockmeyer
Results: ● Standard Usability Questionnaire for Interaction Methods		
Tasks: ● research common Usability Studies ● make Google Form to save automatically the results		
Requirements: ● Google Form ● WP 2: Research		
Signature Project manager		Signature Responsible Team Member

Abbildung 25: Sixth Workpackage: Usability Questionaire

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
7</td><td>WP-Name:
WP 7: "Documentation”</td></tr>
<tr><td>Start:
03.07.2017</td><td>End:
15.07.2017</td><td>Responsible Team Member:
L. Anger, A. Bolder, B. Boerner und
V. Brockmeyer</td></tr>
<tr><td colspan="3">Results:
• Two printed versions of the documentation of the “Interaction Lab”-Project
• Final presentation</td></tr>
<tr><td colspan="3">Tasks:
• Introduction and Motivation
• State of the art
• Description of Hard- and Software
• detailed description of complete System
• Evaluation / Results
• Project Management
• Reflexion
• Conclusion
• prepare final presentation</td></tr>
<tr><td colspan="3">Requirements:
• Latex
• MS PowerPoint
• This work package needs to be finished after all other work packages are ready
• It might be possible and useful to start before the final completion of all other work packages
• GitHub Repository for Latex files</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Members</td></tr>
</table>

Abbildung 26: Seventh Workpackage: Documentation

<table>
<tr><td>Project name:
Interaction Lab</td><td>WP- Nr.:
8</td><td>WP-Name:
WP 8: "Project profile, project video and images"</td></tr>
<tr><td>Start:
07.07.2017</td><td>End:
07.07.2017</td><td>Responsible Team Member:
L. Anger and B.2 Boerner</td></tr>
<tr><td colspan="3">Results:
• brief Project description and profile as Word document
• 2-5 expressive image in adequate Quality
• Image and Tutorial Video of System</td></tr>
<tr><td colspan="3">Tasks:
• write brief Project description and profile in Word document
• capture expressive images of project
• make an image and tutorial video of system</td></tr>
<tr><td colspan="3">Requirements:
• SLR Video camera
• 2. Prototype
• VR Laboratory 1
• MS Word</td></tr>
<tr><td colspan="2">Signature Project manager</td><td>Signature Responsible Team Member</td></tr>
</table>

Abbildung 27: Eighth Workpackage: Project Profile, Video and Images

Literatur

[1] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, Jun 1999.

[2] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, pages 35–ff., New York, NY, USA, 1997. ACM.

[3] Doc-Ok.org. Lighthouse tracking examined. http://doc-ok.org/?p=1478. Aufgerufen: 30. März 2017.

[4] Unity Documentation. Boxcollider. https://docs.unity3d.com/ScriptReference/BoxCollider.html. Aufgerufen: 12. July 2017.

[5] Unity Documentation. Ray. https://docs.unity3d.com/ScriptReference/Ray.html. Aufgerufen: 16. July 2017.

[6] Unity Documentation. Spherecollider. https://docs.unity3d.com/ScriptReference/SphereCollider.html. Aufgerufen: 16. July 2017.

[7] Unity Documentation. Transform.setparent. https://docs.unity3d.com/ScriptReference/Transform.SetParent.html. Aufgerufen: 16. July 2017.

[8] Mikael Eriksson. Reaching out to grasp in virtual reality: A qualitative usability evaluation of interaction techniques for selection and manipulation in a vr game, 2016.

[9] John Haas. *A History of the Unity Game Engine*. PhD thesis, Worcester Polytechnic Institute.

[10] HTC. Htc vive. https://www.vive.com/. Aufgerufen: 30. November 2016.

[11] HTC. HTC Vive – Für Vive geeignete Computer. https://www.vive.com/de/ready/. Aufgerufen: 18. März 2017.

[12] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery & Morgan und Claypool, New York, NY, USA, 2016.

[13] Hit Jones. Unity asset store: Realistic cardboard boxes (pbr, hq). https://www.assetstore.unity3d.com/en/#!/content/58749, 03 2016.

[14] Sangyoon Lee, Jinseok Seo, Gerard J Kim, and Chan-Mo Park. Evaluation of pointing techniques for ray casting selection in virtual environments. In *Third international conference on virtual reality and its application in industry*, volume 4756, pages 38–44, 2003.

[15] Lucie Lescuyer. Unity asset store: Free food pack. https://www.assetstore.unity3d.com/en/#!/content/85884, 04 2017.

[16] Jonathan Lin and Jürgen P Schulze. Towards naturally grabbing and moving objects in vr. *Electronic Imaging*, 2016(4):1–6, 2016.

[17] Sysdia Solutions Ltd. Vrtk - virtual reality toolkit (read me).

[18] Sysdia Solutions Ltd. Unity asset store: Vrtk - virtual reality toolkit - [vr toolkit]. https://www.assetstore.unity3d.com/en/#!/content/64131, 06 2016.

[19] Microsoft. Introducing Visual Studio. https://msdn.microsoft.com/en-us/library/fx6bk1f4(v=vs.90).aspx. Aufgerufen: 18. März 2017.

[20] Mobile World Congress. Mobile World Congress. https://www.mobileworldcongress.com/. Aufgerufen: 30. November 2016.

[21] OneCoinGames. Unity asset store: Food & grocery items - low poly. https://www.assetstore.unity3d.com/en/#!/content/75494, 11 2016.

[22] IVAN POUPYREV and TADAO ICHIKAWA. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages & Computing*, 10(1):19 – 35, 1999.

[23] Steam. Steam VR Internetauftritt. http://store.steampowered.com/steamvr?l=german. Aufgerufen: 26. März 2017.

[24] Technische Hochschule Chemnitz. Human-Machine Interaction Lab. https://www.tu-chemnitz.de/mb/ArbeitsWiss/forschung/labore/human_machine_interaction_lab. Visted: 28. March 2017.

[25] Unity Technologies. Unity. https://unity3d.com/de. Aufgerufen: 8. März 2017.

[26] Unity Technologies. Unity – Multiplatform. https://unity3d.com/unity/multiplatform. Aufgerufen: 14. März 2017.

[27] Unity Technologies. Unity – Public Relations. https://unity3d.com/public-relations. Aufgerufen: 14. März 2017.

[28] Unity Technologies. Unity – VR Overview. https://unity3d.com/de/learn/tutorials/topics/virtual-reality/vr-overview. Aufgerufen: 14. März 2017.

[29] Valve. Valve Software. http://www.valvesoftware.com/. Aufgerufen: 30. November 2016.