

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	4
1.1	Motivation	4
1.2	Usage Context	4
1.3	Project Goal	4
2	State of the Art	5
2.1	VR Labor	5
2.2	VR Grabbing-Interactions	5
2.3	VR Grabbing-Interactions Evaluation	5
3	Materials	6
3.1	Hardware	6
3.1.1	Computer	6
3.1.2	HTC Vive	6
3.2	Software	7
3.2.1	Unity	7
3.2.2	Visual Studio 2015	7
3.2.3	Steam VR	8
4	System	9
4.1	VR Labor	9
4.2	Controller Menu	9
4.3	Interaction Methods	9
4.3.1	Close Range: Touch Grab	10
4.3.2	Close Range: Proximity Grab	10
4.3.3	Close Range: Wand Grab	10
4.3.4	Far Range: Raycast	11
4.3.5	Far Range: Extendable Ray	11
4.3.6	Far Range: Raycast Head Mounted Display	11
4.4	Self-Teaching	11
5	Evaluation	12
6	Project Management	13

6.1	Project Definition	13
6.1.1	??	13
6.1.2	??	13
6.2	Project Planning	13
6.2.1	??	13
6.2.2	??	13
6.3	Project Execution	13
6.3.1	??	14
6.3.2	??	14
6.4	Project Completion	14
6.4.1	??	14
6.4.2	??	14
7	Reflexion	15
8	Conclusion	16
9	Self-Assessment	17
9.1	Anna Bolder	17
9.2	Vera Brockmeyer	17
9.3	Britta Boerner	17
9.4	Laura Anger	17

1 Introduction

Vera

1.1 Motivation

Vera

1.2 Usage Context

Vera

1.3 Project Goal

Vera

2 State of the Art

???

2.1 VR Labor

Anna

2.2 VR Grabbing-Interactions

Laura

2.3 VR Grabbing-Interactions Evaluation

Britta

3 Materials

Britta

3.1 Hardware

Britta

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

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

Die *HTC Vive* ist ein Head-Mounted Display, welches von *HTC* in Kooperation mit *Valve* [14] produziert wird. Vorgestellt wurde dieses am 1. März 2015 im Vorfeld des *Mobile World Congress* [8].

Die Auflösung des Displays beträgt insgesamt 2160×1200 Pixel, was 1080×1200 Pixeln pro Auge entspricht. Die Brille bietet ein Sichtfeld von bis zu 110° bei einer Bildwiederholrate von 90 Hz [4]. Alle technischen Systemvoraussetzungen können in Tabelle 2 eingesehen werden.

Zur Positionsbestimmung im Raum wird die Lighthouse-Technologie [1] von *Valve* genutzt. Zusätzlich sind neben einem Gyroskop auch ein Beschleunigungssensor und

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

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™ GTX 1060 oder AMD Radeon™ 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 [5].

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 [3] und wird von *Unity Technologies* [10] 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 [12].

Unity bietet eine sehr breite Plattformunterstützung [11] und erlaubt ebenso die Entwicklung für Head-Mounted Displays, wie etwa die *Oculus Rift* [13] oder auch die in diesem Projekt verwendete *HTC Vive* [13].

3.2.2 Visual Studio 2015

Britta

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

Microsoft 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 [7] 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 [9] ist die Schnittstelle zwischen der *HTC Vive* und *Unity*. Um das HMD nutzen zu können, muss *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, 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). In the learning room the user will be supported in his learning process by a selfteaching system (compare section 4.4), which can get switched 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.

4.1 VR Labor

Anna

4.2 Controller Menu

Anna

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, there 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 [6]. Interaction methods allowing such an acting are called FR interactions.

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, as well as the actual supermarket scenes (compare section 4.1), 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 on script, where 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 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

4.3.2 Close Range: Proximity Grab

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* [2] placed at the end of the *HTC Vive*-controller. In contrast to the *Touch Grab* described in section 4.3.1 this collider is bigger than the actual size of the controller.

4.3.3 Close Range: Wand Grab

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 PICTURE. 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.

4.3.4 Far Range: Raycast

4.3.5 Far Range: Extendable Ray

4.3.6 Far Range: Raycast Head Mounted Display

4.4 Self-Teaching

Anna

5 Evaluation

Britta

6 Project Management

Vera

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

Dieses Kapitel beschreibt die Planung und das Management des Projektes *MArC*. Es ist in die einzelnen Projektphasen gegliedert, welche jeweils die wichtigsten Punkte der entsprechenden Phasen enthalten.

6.1 Project Definition

Vera

6.1.1 ??

Vera

6.1.2 ??

Vera

6.2 Project Planning

Vera

6.2.1 ??

Vera

6.2.2 ??

Vera

6.3 Project Execution

Vera

6.3.1 ??

Vera

6.3.2 ??

Vera

6.4 Project Completion

Vera

6.4.1 ??

Vera

6.4.2 ??

Vera

7 Reflexion

Vera

8 Conclusion

Vera

9 Self-Assessment

???

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

9.1 Anna Bolder

Anna

9.2 Vera Brockmeyer

Vera

9.3 Britta Boerner

Britta

9.4 Laura Anger

Laura

Literatur

- [1] Doc-Ok.org. Lighthouse tracking examined. <http://doc-ok.org/?p=1478>. Aufgerufen: 30. März 2017.
 - [2] Unity Documentation. Boxcollider. <https://docs.unity3d.com/ScriptReference/BoxCollider.html>. Aufgerufen: 12. July 2017.
 - [3] John Haas. *A History of the Unity Game Engine*. PhD thesis, Worcester Polytechnic Institute.
 - [4] HTC. Htc vive. <https://www.vive.com/>. Aufgerufen: 30. November 2016.
 - [5] HTC. HTC Vive – Für Vive geeignete Computer. <https://www.vive.com/de/ready/>. Aufgerufen: 18. März 2017.
 - [6] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery & Morgan und Claypool, New York, NY, USA, 2016.
 - [7] Microsoft. Introducing Visual Studio. [https://msdn.microsoft.com/en-us/library/fx6bk1f4\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/fx6bk1f4(v=vs.90).aspx). Aufgerufen: 18. März 2017.
 - [8] Mobile World Congress. Mobile World Congress. <https://www.mobileworldcongress.com/>. Aufgerufen: 30. November 2016.
 - [9] Steam. Steam VR Internetauftritt. <http://store.steampowered.com/steamvr?l=german>. Aufgerufen: 26. März 2017.
 - [10] Unity Technologies. Unity. <https://unity3d.com/de>. Aufgerufen: 8. März 2017.
 - [11] Unity Technologies. Unity – Multiplattform. <https://unity3d.com/unity/multiplatform>. Aufgerufen: 14. März 2017.
 - [12] Unity Technologies. Unity – Public Relations. <https://unity3d.com/public-relations>. Aufgerufen: 14. März 2017.
 - [13] Unity Technologies. Unity – VR Overview. <https://unity3d.com/de/learn/tutorials/topics/virtual-reality/vr-overview>. Aufgerufen: 14. März 2017.
 - [14] Valve. Valve Software. <http://www.valvesoftware.com/>. Aufgerufen: 30. November 2016.
-