

Technologieupgrade für das StereoX Labor

Studienarbeit

Des Studienganges Informationstechnik
An der Dualen Hochschule Baden-Württemberg Ravensburg

Von
Nico Kubasta und Patrick Ziegeldorf

17.07.2017

Bearbeitungszeitraum
Matrikelnummer, Kurs
Betreuer der Studienarbeit

April – Juli 2017
4900123, 1259451
Prof. Dr. Andreas Judt

Erklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2015.

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

Technologieupgrade für das StereoX Labor

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ulm, den 14.07.2017

Nico Kubasta

Patrick Ziegeldorf

Abstract

Chapter 1

In summary, a wide range of structural work was done. Above all, the transition to VR, which is described in more detail in the second chapter of this thesis. Due to the change in technology, some rebuilding measures were carried out in the laboratory. However, with the stocktaking and the accompanying organization of the involved hardware, not only material clearing work took place. Two Systems, KeePass and Github, were introduced to organize all fields of work. Last but not least, the documentation of the introduction of these systems is particularly important in this report. Because students that are not aware that these systems are part of the study work can't use them.

Chapter 2

With the HTC Vive, a technology upgrade was found. A script was created to convert STL files to OBJ files. In addition, another script was created, which simplifies the use of the first script. Here, the required data is requested by the user. A presentable scene was also created. This can be compiled, but unfortunately the dynamic loading of models at runtime does not (see 2.5.3). For this reason, the application must be started from the Unity Editor.

The scene itself fulfills all requirements. It can be interacted with objects by lifting them with one controller. Using the second controller on an object scales it. The objects also interact with one another. Thus each object has its own body, which cannot be penetrated. For this reason, no objects can fall through the floor or be thrown through walls. In addition, each object has a simple physical model that simulates gravity.

Inhalt

Kapitel 1	Strukturelle Arbeiten	2
1.1	Einleitung.....	3
1.1.1	Motivation	3
1.1.2	Zielsetzung.....	3
1.2	Erfassung des Inventars	4
1.2.1	Gründe für die Inventur	4
1.2.2	Ergebnisse	4
1.3	Konfigurationsmanagement.....	6
1.3.1	Github	6
1.3.2	KeePass	6
1.3.3	Accounts	7
1.4	Umbau des Labors	9
1.4.1	Rückrüstung der Beamer und Leinwände	9
1.4.2	Aufbau der VR-Hardware.....	9
1.5	Zusammenfassung	11
Kapitel 2	Wissenschaftliche Arbeiten	12
2.1	Einleitung.....	13
2.1.1	Motivation	13
2.1.2	Zielsetzung.....	13
2.2	Bisheriges Konzept.....	14
2.2.1	Stereoskopische Projektion	14
2.2.2	Systemaufbau	14
2.2.3	Gestensteuerung	15
2.2.4	Fazit	16
2.3	Der Umstieg auf VR.....	18
2.3.1	Neue Technologien.....	18

2.3.2	Brillenauswahl.....	20
2.4	Grundlagen.....	25
2.4.1	Unity.....	25
2.4.2	Blender	26
2.4.3	Dateiformate	26
2.4.4	SteamVR	27
2.5	Demonstrationsfähige Szene	28
2.5.1	Anforderungen	28
2.5.2	Umwandlung der Dateien	29
2.5.3	Laden von Modellen.....	30
2.5.4	Interaktion	31
2.6	Zusammenfassung.....	33
2.7	Ausblick.....	34
2.7.1	Zusammenarbeit mit anderen Studienarbeiten	34
2.7.2	Erweiterungen.....	34
2.7.3	Upgrades	36

Abkürzungen

API *Application Programming Interface*

AR *Augmented Reality*

CAD *Computer Aided Design*

DVI *Digital Visual Interface*

GFR *Global Formula Racing*

GPL *GNU General Public License*

HTC *High Tech Computer Corporation*

MTL *Material Template Library*

OBJ *Wavefront Obejct*

STL *Standard Tessellation Language*

UK *United Kingdom*

USB *Universal Serial Bus*

VGA *Video Graphics Array*

Vorwort

Eine große und wichtige Rolle bei Softwareprojekten spielt eine präzise und ausführliche Dokumentation. Um die Dokumentation dieses Projektes auf ein angemessenes Niveau zu heben, wurde entschieden diesen Bericht in zwei Teile zu gliedern. Zum einen in das Kapitel 1 „Strukturelle Arbeiten“, welches sich ausschließlich mit strukturellen Arbeiten rund um das Projekt „Stereo X Labor“ befasst. Hier werden alle Arbeiten dokumentiert, die sich nicht direkt mit dem technologischen Fortschritt des Projektes befassen. Das Gegenstück zu strukturellen Arbeiten bilden wissenschaftliche Arbeiten, welche das Projekt vorantreiben. Diese werden im Kapitel 2 Wissenschaftliche Arbeiten behandelt. In diesem Umfeld werden speziell in dieser Arbeit die Überlegungen und Vorgänge zum Technologieupgrade im Projekt beschrieben.

Kapitel 1 Strukturelle Arbeiten

1.1 Einleitung

1.1.1 Motivation

Da Studienarbeiten von vielen Studenten bearbeitet werden und diese oft nur über einen relativ kurzen Zeitraum an diesen arbeiten, kommt es vor, und so trifft es auch auf diese Studienarbeit zu, dass viele verschiedene Dokumentationen und Systeme parallel existieren. So wird im Laufe der Zeit viel neue Hardware angeschafft, ohne einen genauen Überblick darüber zu haben, was bereits existiert oder eventuell auch fehlt, obwohl es eigentlich zum Projekt gehört. Auch Software, egal ob selbst entwickelt oder von anderen Anbietern, ist in Studienarbeiten oft ein leidiges Thema, da jeder Student eine andere Vorstellung von der idealen Verwaltung dieser hat. Dies führt dazu, dass im darauffolgenden Studienjahr zunächst viel Zeit damit verbracht werden muss um sich einen Überblick über den Stand der Arbeit zu machen. In den Berichten der Studienarbeiten wird oft auch wenig bis gar nicht dokumentiert, wo und ob sich die entwickelte Software auf dem System befindet. Selbst wenn diese dann gefunden wird, ist diese oft spärlich dokumentiert.

1.1.2 Zielsetzung

Aufgrund der oben angeführten Gründe soll in diesem Kapitel erläutert werden, welche Maßnahmen ergriffen wurden um den zukünftigen Studenten den Einstieg in diese Studienarbeit so einfach wie möglich zu gestalten. Dafür soll zunächst das gesamte Inventar der Studienarbeit erfasst und dokumentiert werden. Zudem sollen Tools eingeführt werden, um die Übergabe der Studienarbeit an den Betreuer und die Studenten zu erleichtern. Diese werden unter dem Thema „Konfigurationsmanagement“ zusammengefasst. Nicht zuletzt werden durch das Technologieupgrade benötigte Umbauarbeiten im Labor festgehalten. Dadurch soll es einfacher werden, diese wieder rückgängig zu machen, sollte das Projekt wieder in einen alten Stand versetzt werden.

1.2 Erfassung des Inventars

1.2.1 Gründe für die Inventur

Durch das ständig wachsende Inventar der Studienarbeit ist es mit der Zeit schwierig geworden den Überblick darüber zu wahren. Zudem ist es leider vorgekommen, dass Gegenstände ohne Hinweise auf ihren Verbleib verschwinden. Um für die Zukunft einen besseren Überblick zu haben und auch den Materialschwund einzudämmen, wurde zu Beginn eine Inventur durchgeführt. Hierbei wurden alle Gegenstände, welche eindeutig der Studienarbeit zugeordnet werden können, in einer Liste erfasst. Damit ist der aktuelle Bestand festgehalten. In Zukunft kann diese Liste aktualisiert werden, sollten sich Änderungen im Bestand der Studienarbeit ergeben. Durch eine gewissenhafte Führung dieser Inventarliste kann auch der Kreis der Verdächtigen eingeschränkt werden, sollten Teile der teilweise sehr teuren Ausstattung verschwinden.

Da diese Studienarbeit sich das Labor mit anderen Studienarbeiten teilt wurden während der Inventur alle Gegenstände in Kisten verstaut. Dies trägt zu einem ordentlicheren Zustand des Labors bei, und macht es einfacher die Gegenstände im Labor den einzelnen Studienarbeiten zuzuordnen.

1.2.2 Ergebnisse

Aufgrund von bereits vorangegangener Modernisierungen fällt das Inventar der Studienarbeit relativ groß aus. So gehören unter anderem unzählige Kabel dazu. Mit insgesamt 14 Stück sind am meisten Kaltgerätestecker vorhanden. Allerdings sind hiervon zwei in der Schweizer Ausführung und einer in UK Ausführung. Auch diverse Bildschirmkabel sind vorhanden. Davon insgesamt fünf DVI-D Single Link zu HDMI Kabel und zwei DVI-D Single Link Kabel. Allerdings sind ebenfalls veraltete Kabel, wie vier VGA und vier Composite AV Kabel vorhanden. Diese sind höchst wahrscheinlich dem Lieferumfang der Xbox 360 zu verdanken. Von diesen sind nämlich neun in der Release Version vorhanden und weitere fünf Xbox 360 Slim. Zusätzlich zur „alten“ bereits modernisierten Technik sind noch vier Beamer der Marke Acer P5390W vorhanden. Ebenfalls sind einige PC-Komponenten einzeln vorhanden. Nicht zuletzt ist ein leistungsstarker PC vorhanden, welcher weiterhin benutzt werden soll. In diesem Jahr hinzugekommen ist eine HTC Vive. Zusätzlich gekauft wurden ein WLAN Empfänger für den PC, da der alte abhandengekommen

Technologieupgrade für das StereoX Labor

ist und zusätzlich ein kleines Vorhängeschloss, welches am PC befestigt wurde, um zu verhindern, dass weiteres Equipment daraus abhandenkommt.

1.3 Konfigurationsmanagement

1.3.1 Github

Nicht nur der Verlust und die Unordnung von Hardware macht es neuen Studenten schwer sich in die Thematik einer Studienarbeit einzuarbeiten. Auch digitales Chaos und fehlende Programmteile erschweren den Einstieg. Aus diesem Grund wurde entschieden, ein Tool zur Versionierung einzuführen. Die Wahl ist dabei auf Git gefallen, da dort mit Github eine Plattform geboten wird, auf der die digitalen Inhalte gespeichert werden können. Zudem bildet Git zusammen mit Github ein mächtiges Werkzeug für die Planung von Projekten und die kontinuierliche Arbeit an unterschiedlichen Aspekten dieser. Github bietet den Entwicklern die ihren Code und auch andere Dateien in einem Repository abzulegen. Andere Entwickler haben dann die Möglichkeit in verschiedenen Branches des Projekts zu arbeiten. So können Features in mehreren Branches mit verschiedenen Implementationen realisiert werden. Darüber hinaus können Issues erstellt werden, um Bugs zu verfolgen und zu beheben, oder geplante Erweiterungen zu organisieren und zu realisieren.

Daraus resultierend wurde für das Projekt ein eigener Git/Github Account angelegt (siehe 1.3.3 Accounts). Über diesen Account wurde ein öffentliches Repository [1] angelegt, welches unter der GPL-3.0 Lizenz steht. Um ein privates Repository anzulegen würden monatlichen Kosten von min. 7 \$ anfallen. Durch diesen Account können zentral über den Account der Studienarbeit weitere „Mitarbeiter“ zum Projekt hinzugefügt werden. Dadurch können dem Projekt dynamisch die Accounts der Studenten hinzugefügt werden, welche gerade an der Studienarbeit arbeiten. Durch das von Github bereitgestellte Wiki können Studenten eine Dokumentation für ihre Nachfolger hinterlassen, die leicht zu finden ist, und ebenfalls mit dem Fortschritt des Projektes angepasst werden kann.

1.3.2 KeePass

Möchte man bei der Entwicklung einer Anwendung für die HTC Vive VR-Brille auf den gesamten Wissensschatz der Community und sämtliche Aspekte der Brille zugreifen werden einige Accounts benötigt. Diese werden in 1.3.3 genauer beschrieben. Im ersten Schritt wurden diese zwar relativ simpel in einer Tabelle hinterlegt. Durch die große Anzahl an benötigten Accounts erschien es jedoch von Vorteil eine Software zur Verwaltung dieser zu Verwenden. Da in den Praxisphasen

bereits gute Erfahrungen mit KeePass gemacht wurden, wurde auch für die Studienarbeit die Verwendung von KeePass beschlossen. KeePass ist eine freie Verwaltungssoftware für Passwörter. Sie verwendet in der Classic Edition wahlweise den AES-Algorithmus oder den Twofisch-Algorithmus. Die Datenbank, in welcher die Passwörter hinterlegt werden, wird durch einen Hauptschlüssel und unterschiedliche sekundäre Schlüssel verschlüsselt. Im Fall der Studienarbeit wurde der Hauptschlüssel durch eine Schlüsseldatei und einen geheimen Schlüssel, welcher durch das Windows-Benutzerkonto abgeleitet wurde, erweitert. Durch die somit erhöhte Entropie Schlüssel ist es ohne die Schlüsseldatei und den exakten Schlüssel des Benutzerkontos nicht möglich die Datenbank zu entschlüsseln.

Allerdings muss sowohl von Schlüsseldatei und Benutzerkontenschlüssel eine Sicherung angelegt werden. Bei Verlust des Benutzerkontos kann, selbst bei identischen Einstellungen, kein identischer Schlüssel generiert werden, da der Schlüssel zwar mit diesen Daten geschützt ist, aber nicht direkt daraus abgeleitet werden kann.

1.3.3 Accounts

1.3.3.1 Steam

Steam stellt die grundlegende API und bereits fertige Komponenten zur Entwicklung von Anwendungen für die HTC Vive bereit. Zudem wird Steam als Software auf dem computer benötigt, um die Brille verwenden zu können (siehe 2.4.4 SteamVR). Auch bereits veröffentlichte Anwendungen und Spiele können über Steam, welche als Vertriebsplattform für Spiele bekannt ist, bezogen werden. Da Steam jedoch nicht ohne einen Account funktioniert und es impraktikabel ist Accounts von Studenten zu verwenden, wurde ein Account eigens für die Studienarbeit angelegt. Theoretisch könnten zwar auch Accounts von Studierenden verwendet werden, allerdings müssen diese dann immer mittels eines Codes die VR Inhalte freischalten, wobei nicht bekannt ist, ob dies unbegrenzt möglich ist.

1.3.3.2 Viveport

Viveport ist, ähnlich zu Steam, ein App-Store für die HTC Vive. Allerdings bietet er noch viele weitere Möglichkeiten für Entwickler. So kann man sich mit anderen Entwicklern in Foren vernetzen und Informationen und Wissen austauschen. Ursprünglich war wohl auch die Viveport-Software nötig um die HTC Vive verwenden

zu können. Im Laufe der Studienarbeit stellte sich jedoch heraus, dass Steam/SteamVR ausreichend sind. Nichts desto trotz existiert ein Account für Viveport und wird Vollständigkeitshalber ebenfalls hier aufgeführt.

Bei der Erstellung des Accounts bzw. dem Upgrade zu einem Entwickleraccount traten jedoch unerwartete Probleme auf. So muss zwingend ein Benutzerfoto hochgeladen werden. Dies war jedoch zum Zeitpunkt der Studienarbeit aus unbekannten Gründen nicht möglich.

1.3.3.3 Unity

Mit Unity nutzt HTC für die Vive eine mächtige, aber relativ Einsteigerfreundliche Grafikenengine. Für den privaten und schulischen Gebrauch ist die Engine kostenlos. Allerdings wird, um überhaupt ein Projekt im eigenen Editor zu erstellen, ein Account benötigt.

1.3.3.4 Github

Wie bereits in 1.3.1 Github erläutert, wurde entschieden, die Arbeiten der Studienarbeit auf einem Github Repository abzulegen. Auch hierzu ist ein Account nötig. Es wurde entschieden hier ebenfalls einen eigenen Account für die Studienarbeit anzulegen. Zum einen können Accounts von Studierenden zwar, durch ein Angebot von Github, kostenfreie private Repositories anlegen, zum anderen müssten diese dann auch von Studierenden verwaltet werden, die eventuell bereits lange Zeit nichtmehr an der Studienarbeit mitarbeiten.

Durch den eigenen Account für die Studienarbeit entfällt zwar die Möglichkeit private Repositories anzulegen, allerdings kann das Projekt nun zentral über diesen Account verwaltet werden. Das öffentliche Repository der Studienarbeit kann zwar von jedem angesehen werden, jedoch nur von zugelassenen Mitgliedern bearbeitet werden. Des Weiteren wurde das Projekt unter die GNU General Public License gestellt.

1.3.3.5 Google

Um einen Account bei einem Anbieter anlegen zu können wird inzwischen nahezu immer eine aktuelle und gültige E-Mail Adresse benötigt. Da im privaten Umfeld bisher nur gute Erfahrungen mit Google bzw. Gmail gemacht wurde, wurde entschieden eine E-Mail Adresse bzw. einen Account bei selbem anzulegen.

1.4 Umbau des Labors

1.4.1 Rückrüstung der Beamer und Leinwände

Die Studienarbeit um das Stereo X Labor hat bisher auf das Prinzip der Stereoskopischen Projektion gesetzt. Hierfür wurden insgesamt vier Beamer und zwei Leinwände benutzt (siehe 2.2 Bisheriges Konzept). Da jedoch durch den Umstieg auf eine VR-Brille weder Beamer noch Leinwände benötigt werden, wurden diese abgebaut. Die Beamer waren, jeweils als Paar, an einer Halterung an einer Stange, welche an der Decke befestigt ist, montiert. Diese Halterungen wurden von den Beamern und der Stange entfernt. Die Stange an sich ist jedoch immernoch an der Decke befestigt (siehe 1.4.2 Aufbau der VR-Hardware). Die Leinwände waren ohne bauliche Maßnahmen im Labor aufgestellt worden. Die Beamer wurden in die dafür vorgesehenen Taschen verstaut und lagern in einem abgeschlossenen Raum in der Hochschule. Bei Bedarf kann auf diese über Herrn Lau zugegriffen werden. Deshalb konnten sie einfach abgebaut und in den dafür vorgesehenen Taschen verstaut werden. Diese Taschen lagern im Labor.

1.4.2 Aufbau der VR-Hardware

Nachdem die alte Hardware abgebaut wurde, kann die für die VR-Brille benötigte Hardware aufgebaut werden. Diese besteht aus zwei Lightboxen. Dies sind kleine Boxen in denen Infrarot LEDs verbaut sind. Die Boxen werden über ein separates Netzteil mit Strom versorgt und können frei im Raum platziert werden. Zu beachten ist allerdings, dass sie über Kopfhöhe, also mindestens 2 m, befestigt werden sollten. Zur Befestigung der Boxen liefert HTC jeweils einen Halter mit, welcher mit zwei Schrauben an einer Wand befestigt werden kann. Auf diesem Halter wird die Lightbox mittels einer Stativschraube befestigt. Die Stativschraube weist jedoch keine metrisches ISO-Gewinde auf, sondern ein Zollgewinde in der Größe $\frac{1}{4}$ ". Aus diesem Grund können auch beliebige andere Stative zur Positionierung der Lightboxen verwendet werden, solange sie ebenfalls über eine $\frac{1}{4}$ " Schraube verfügen.

Bei der Montage der Halterungen für die Lightboxen wurde zuerst, vor dem Abbau der Stereo Hardware, versucht, diese mittels Kabelbindern und Drähten an den im Labor vorhandenen Rohren zu befestigen. Allerdings stellte sich diese Methode als ungeeignet heraus, da selbst die kleinste Bewegung der Lightbox zu einem Ruckeln in der VR-Brille führen und diese wiederum zu Motion-Sickness. Deshalb wurde

letztlich entschieden die Lightboxen samt Halterungen an den Rohren bzw. Halterungen der Beamer zu befestigen. Aus diesem Grund wurden sowohl Beamer als auch Leinwände abgebaut. Die Lightbox wurde dann an einem Halterungsarm, an welchem zuvor ein Beamer befestigt war, mit Kabelbindern befestigt und am Rohr, welches an der Decke befestigt ist, montiert. Die Kabel des separaten Netzteils wurden ebenfalls mit Kabelbindern befestigt, um Unfällen vorzubeugen.

Bei der Montage der Lightboxen muss darauf geachtet werden, dass diese sich gegenseitig „sehen“ können. Stellt man sich ein Quadrat vor, welches die Fläche darstellt, auf der der Anwender getrackt werden soll, sollten die Lightboxen in den Ecken dieses platziert werden. Im Idealfall werden die Boxen in gegenüberliegenden Ecken platziert. Aufgrund der bereits montierten Halterungen der Beamer konnte dies leider nicht berücksichtigt werden, so dass die Boxen nun in zwei nebeneinander liegenden Ecken dieses imaginären Quadrates montiert sind.

Zum Lieferumfang der HTC Vive zählt zusätzlich noch eine Connector-Box, welche die Kabel, die zum Computer führen, verlängert und physisch vom Headset trennen. Das Headset besitzt zwar die gleichen Anschlüsse wie die Kabel, welche zum Computer führen, und könnte somit direkt am Computer angeschlossen werden, allerdings ist das Netzteil, welches benötigt wird um die Brille zu verwenden, nur dazu gedacht in Verbindung mit dieser Box verwendet zu werden. Somit sollten die Kabel des Headsets in die Connector-Box gesteckt werden. Dies ermöglicht zudem, dass das Headset, sollte die Rückseite des Computers schwer erreichbar sein, vom Computer getrennt werden kann und verstaut werden kann.

Des Weiteren werden neben den bereits genannten Komponenten noch zwei Controller, Netzteile zum Aufladen dieser und Kopfhörer mitgeliefert.

1.5 Zusammenfassung

Zusammenfassend wurden verschiedenste strukturelle Arbeiten an der Studienarbeit vorgenommen. Allem voran der Umstieg auf VR, welcher im zweiten Kapitel dieser Arbeit genauer beschrieben wird. Durch den Technologiewechsel bedingt wurden einige Umbaumaßnahmen im Labor der Studienarbeit vorgenommen. Allerdings fanden mit der Inventur im Labor und der einhergehenden Organisation der zur Studienarbeit gehörenden Hardware nicht nur materielle Aufräumarbeiten statt. Vielmehr wurde, mit KeePass, ein System zur Account und Passwortverwaltung und, mit Github, eine Möglichkeit zur besseren Verwaltung der entwickelten Software eingeführt. Nicht zuletzt ist die Dokumentation über die Einführung dieser Systeme in diesem Bericht besonders wichtig, damit Studierende, welche die Studienarbeit fortführen, einen einfachen Start in die Arbeit im Stereo X Labor haben. Denn sollte den Studierenden nicht bewusst sein, dass diese Systeme Bestandteil der Studienarbeit sind, können sie auch nicht genutzt werden.

Kapitel 2 Wissenschaftliche Arbeiten

2.1 Einleitung

2.1.1 Motivation

Die Technik im Bereich der Visualisierung schreitet immer schneller voran. So geschehen mit der 3D-Technik. Das Konzept des Stereo X Labors beruht auf dem Prinzip der stereoskopischen Projektion. Hierbei werden den Augen unterschiedliche Bilder gezeigt und durch den Abstand der Augen entsteht der Eindruck eines dreidimensionalen Objekts. Im Stereo X Labor wird dieses Prinzip momentan mit mehreren Beamern und Leinwänden auf die Spitze getrieben, um ein im Raum schwebendes Hologramm zu erschaffen. Jedoch ist diese Technik durch den Fortschritt in den letzten Jahren doch etwas veraltet. Aus diesem Grund soll für das Stereo X Labor untersucht werden, wie die Technologie modernisiert werden kann, um diese auch in Zukunft gut nutzen zu können.

2.1.2 Zielsetzung

Um in einem Projekt etwas zu erreichen, müssen die Ziele klar abgesteckt sein. Diese sind zum einen die Auswahl einer Technologie, um das Labor zu modernisieren. Nach der Auswahl der Technologie muss zudem ein spezifisches Produkt dieser Technologie ausgewählt werden, um im Projekt eingesetzt zu werden. Dies ist jedoch ein kleiner Teil der geplanten Ziele. Gesamtziel ist es, eine präsentierfähige Szene zu erstellen, welche die Vorteile der gewählten Technologie aufzeigt und dem Nutzer ermöglicht nicht nur die Technologie an sich, sondern auch die Möglichkeiten mit dieser zu erleben.

Des Weiteren sollen mögliche Konzepte für den Einsatz der gewählten Technologie, unter Umständen auch in Kooperation mit anderen Studienarbeiten, erörtert werden.

2.2 Bisheriges Konzept

2.2.1 Stereoskopische Projektion

Die vorangehenden Studienarbeiten beruhen auf dem Prinzip der Stereoskopischen Projektion oder auch Stereoskopie. Dieser Effekt ist beispielsweise aus dem Kino bei 3D Filmen bekannt. Dabei werden zwei Bilder auf die Leinwand projiziert, welche eine unterschiedliche Polarisierung haben. Der Nutzer bekommt eine Brille bei welcher die Gläser ebenfalls unterschiedlich aber zum Bild passend polarisiert sind. Durch die Brille erreicht jeweils nur ein Bild ein Auge. Da das menschliche Gehirn darauf ausgelegt ist, jederzeit zwei ähnliche, aber durch den Abstand der Augen nicht ganz gleiche, Bilder zu sehen, nehmen wir Menschen beim Sehen die Tiefe wahr. Dieser Effekt wird durch die beiden durch die Brille empfangenen Bilder auch erzeugt. Da diese auch einen leichten perspektivischen Unterschied haben, empfindet der Nutzer auch eine Tiefe. Der Hologramm Effekt wird durch die Kombination zweier dieser 3D Bilder erlangt.

2.2.2 Systemaufbau

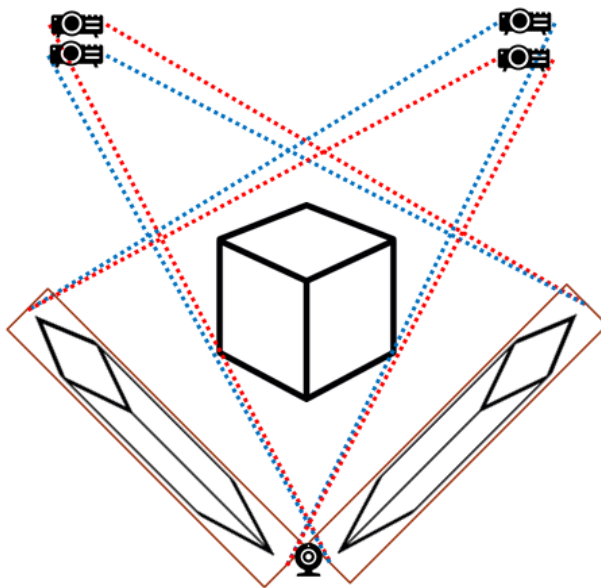


Abbildung 1: Schema des Sytemaufbaus

In obigem –nicht maßstabgetreuen– Schema lässt sich der Aufbau der vorherigen Stereo X Installation erkennen. Es gibt zwei Leinwände, die jeweils von zwei Beamern angestrahlt werden. Vor jeden Beamer wurde eine Polarisationsfolie gehängt. Die beiden Bilder, die auf die gleiche Leinwand gestrahlt werden, also von

Beamern, die übereinander hängen, sind dabei unterschiedlich polarisiert. Es sind aber nur zwei unterschiedliche Polarisierungen für die vier Bilder vorhanden. Mit einer entsprechend polarisierten Brille können dann sowohl das linke Bild als auch das rechte Bild mit einem 3D Effekt gesehen werden. Im richtigen Betrachtungswinkel entsteht durch die Überlagerung der 3D Effekte ein Hologramm Effekt.

Zusätzlich zu diesem Aufbau ist zwischen den beiden Leinwänden eine Microsoft Kinect angebracht. Der Zweck dieser wird im nachfolgenden Abschnitt 2.2.3 Gestensteuerung erklärt.

2.2.3 Gestensteuerung

Zusätzlich zu dem oben beschriebenen System wurde noch eine Gestensteuerung von einer anderen Studienarbeit entwickelt.



Abbildung 2: Kinect 2 for Windows

Die Gestensteuerung funktioniert über eine Kinect v2 von Microsoft, wie sie in obiger Abbildung 2¹ zu sehen ist. Über einen aktiven Infrarot Sensor, werden Objekte anhand ihrer Oberflächenrückstrahlung bestimmt. Dies kann sowohl im hellen wie auch im dunklen geschehen. Aus den empfangenen Daten errechnet Kinect automatisch ein Skelettmodell und gibt dem Entwickler dieses über eine API von Microsoft zurück. In diesem Modell sind die wichtigsten Gelenke für die Körperhaltung als Knoten definiert, mit diesen kann entweder selbst eine Geste programmiert werden oder eine von Microsoft übernommen werden.

¹ <https://1.f.ix.de/imgs/18/1/2/7/0/2/7/8/Kinect-5d0e062ea99ee253.jpeg> - 13.07.2017



Abbildung 3: Kinect Erkennung

In Abbildung 3² sind deutlich Menschen und die zugehörigen, von der Kinect erkannten Skelettmodelle zu erkennen. Microsoft gibt an bis zu sechs Personen gleichzeitig erkennen zu können.³ Von diesen könnten dann einzeln Gesten ausgewertet werden, das System der vorherigen Studienarbeit beruht jedoch auf einer einzigen Person.

2.2.4 Fazit

Insgesamt war das ganze System für den Zeitpunkt der Erstellung sehr einfallsreich und gut. Dies ist hauptsächlich den fehlenden alternativen Technologien geschuldet. Probleme waren, dass das System Fehler anfällig war, vor allem in der Umgebung, in der es aufgebaut war. Der Raum hat eine hohe Luftfeuchtigkeit, wodurch die Beamer angegriffen wurden, so waren zu Beginn dieser Studienarbeit auch zwei Beamer kaputt. Des Weiteren war das System nur von einer begrenzten Position aus zu betrachten, weswegen der Nutzer das Modell vor sich bewegen und selbst an der gleichen Position bleiben musste. Auch gibt es beim Aufbau des Systems einiges zu beachten, wie die Position der Beamer, der Leinwände und der Kinect. Wenn dort

² <https://1.f.ix.de/imgs/18/1/2/7/0/2/7/8/Track-b4e19a7f169eda66.png> - 13.07.2017

³ [1]

Technologieupgrade für das StereoX Labor

etwas abweicht, entsteht nicht das für den Nutzer gewünschte Hologramm Effekt. Zu guter Letzt ist auch noch zu beachten, dass der Raum nicht zu hell sein darf, da die Bilder der Beamer sonst nicht zu sehen sind.

Zusammenfassend lässt sich sagen, dass es ein gutes und technisch fortgeschrittenes Projekt war, aufgrund neuer Technologien aber dennoch reif für ein Upgrade.

2.3 Der Umstieg auf VR

2.3.1 Neue Technologien

2.3.1.1 Augmented Reality

Die erste betrachtete Möglichkeit ist Augmented Reality. Darunter wird das einblenden von virtuellen Objekten in die reale Welt verstanden. Bekannte Beispiele dafür sind zum Beispiel Pokémon GO, Head-up Displays –auch HUD genannt– oder die Microsoft Hololense. Für das Stereo X Labor käme von diesen am ehesten eine Hololense in Erwägung, da eine Smartphone App problematisch bei der Berechnungszeit und der Tiefendarstellung wäre. Auch die Möglichkeit mit diesen Objekten dann zu interagieren ist begrenzt.

Gegen die Hololense im Vergleich zu den später genannten Möglichkeiten, sprachen der hohe Preis, der beschränkte Zugang und das eingeschränkte Blickfeld, da der Bildschirm (noch) recht klein ist. Für zukünftige Anwendungen des Stereo X Programms, wäre ein Einstieg in Augmented Reality mit der Hololense oder Produkten von anderen Anbieter durchaus denkbar, wenn diese auf dem Normalverbrauchermarkt zu finanzierbaren Preisen angeboten werden.

2.3.1.2 Virtual Reality

Virtual Reality ist mit der Veröffentlichung der Oculus Rift und der HTC Vive bekannt geworden. Hierbei hat der Nutzer mittels zweier in eine Brille verbaute Bildschirme das Gefühl, sich direkt in einer virtuellen Szene zu befinden. Der Nutzer hat hierbei die Möglichkeit sich in der Szene umzuschauen, wie in der realen Welt. Mithilfe externer Controller kann er sich außerdem in der Szene bewegen und mit Objekten interagieren. Die Möglichkeiten dazu sind sehr von den verwendeten Controllern abhängig. Nachfolgend werden einige Beispiele vorgestellt.



Abbildung 4: Zwei Möglichkeiten des Motion Tracking

In Abbildung 4: Zwei Möglichkeiten des Motion Tracking⁴ ist links der Sensor der HTC Vive zu sehen. Dieser sendet ein Signal, welches die Brille empfangen und somit ihre Position im Raum bestimmen kann. Rechts ist ein Gerät zu sehen, welches Bewegungen des Nutzers erkennt und somit Bewegungen im Raum simulieren kann. Dabei befindet sich der Nutzer immer an der gleichen Stelle, während er sich beim Vive Sensor selbst im Raum bewegt.



Abbildung 5: Vive Kontroller

⁴ https://www.vive.com/media/filer_public/f7/e1/f7e1eb16-6a14-4707-a82b-c9f7024ba161/base_station_01.jpg - http://www.virtuix.com/wp-content/uploads/2014/12/omni_product.png - 09.07.2017

In Abbildung 5⁵ ist der Controller der HTC Vive abgebildet, welcher die Position der Hände im Raum bestimmt und mit Buttons Interaktionen zulässt.

Im Nachfolgenden Abschnitt der Brillenauswahl wird genauer auf die verschiedenen Möglichkeiten eingegangen.

2.3.1.3 Mixed Reality

Die auch als Augmented Virtuality bekannte Mixed Reality beschreibt Anwendungen, die der Virtual Reality sehr ähnlich ist, jedoch werden Objekte aus der realen Welt in der virtuellen Szene dargestellt. Dies ermöglicht die Interaktion mit der realen Welt innerhalb der Szene. So ist es zum Beispiel möglich für den User sich auf ein Sofa zu setzen, das aus der realen Welt an der gleichen Stelle übernommen, oder etwas auf einem Tisch zu platzieren. Gleichzeitig können auch virtuelle Objekte dargestellt und mit diesen interagiert werden.

Bisherige Anwendungen dafür sind sehr begrenzt vorhanden und im Rahmen der Studienarbeit konnte kein größerer Nutzen dafür erkannt werden. So hat zum Beispiel die HTC Vive die Möglichkeit den Raum um den Nutzer mittels einer Kamer in der Brille anzuzeigen, jedoch können nicht einzelne Objekte übernommen werden. Auch die Interaktion mit vitruellen Objekten ist dann nicht mehr möglich.

2.3.2 Brillenauswahl

2.3.2.1 Samsung Gear VR

Die in Abbildung 6⁶ zu sehende Samsung Gear VR ist ein Virtual Reality Headset, in dem keine Recheneinheit oder Bildschirm verbaut ist, sondern eines der Samsung-Handys aus Galaxy S6, Note 5 oder Galaxy S7 platziert wird. Auf dem Handy wird dann mit einer entsprechenden App der gewünschte Inhalt angezeigt. Jegliche Rechenleistung übernimmt dabei das Handy selbst. Bewegungen des Nutzers im Raum werden nicht erkannt, jedoch Bewegungen des Kopfes, welche über den Gyrosensor wahrgenommen werden und verarbeitet werden können.

⁵ https://www.vive.com/media/filer_public/ac/85/ac8560e4-8d7f-42b6-9394-8fa6d5064b4e/controller_01.jpg - 09.07.2017

⁶ http://www.samsung.com/common/img/galaxynote4/micro/feature/gearvr/imagination_0.jpg - 10.07.2017



Abbildung 6: Samsung Gear VR

Trotz geringerer Kosten als bei anderen VR Headsets, wurde gegen das Samsung Gear VR entschieden, da die rechenintensiven VR Anzeigen den Akku schnell entleeren und das Handy teilweise zum Stocken bringen.

2.3.2.2 Playstation VR

Mit der Playstation VR, welche in Abbildung 7⁷ zu sehen ist, lassen sich Anwendungen der Playstation 4 in einer virtuellen Realität wiedergeben. Die Brille wird mit der Konsole verbunden und dient nur als Bildschirm. Jegliche Berechnungen nimmt die Konsole vor. Bewegungen im Raum werden nicht erkannt, jedoch wird über die Playstation Kamera die Bewegung des Kopfes anhand von Lichtern an der Brille erkannt. Die so empfangenen Daten lassen sich in den Anwendungen verarbeiten.

⁷ http://picscdn.redblue.de/doi/pixelboxx-mss-70945115/fee_786_587_png/SONY-PlayStation-VR-Virtual-Reality-Brille - 10.07.2017



Abbildung 7: Playstation VR Headset

Die Playstation VR wäre mit zusätzlicher Playstation im selben Preissegment wie andere VR Lösungen gewesen, wurde aber aufgrund der komplizierten Vorgehensweise, die nötig gewesen wäre um Anwendungen für Playstation zu entwickeln, und wegen fehlendes Motion Tracking im Raum wieder verworfen.

2.3.2.3 Oculus Rift

Für die Oculus Rift wird ein leistungsfähiger PC benötigt, welcher die Berechnungen der Anzeige übernimmt. Die Rift übernimmt dabei die Anzeige und das wiedergeben von Sounds. Des Weiteren die Erkennung von Bewegungen des Kopfes durch einen Gyro- und ein Beschleunigungssensor. Bewegungen im Raum erfassen externe Sensoren, die zusätzlich platziert werden müssen und Infrarotlicht, das vom Headset ausgesendet wird, empfangen. Die Informationen werden an den PC weitergeleitet, welcher daraus die Position errechnet.



Abbildung 8: Oculus Rift

Die Rift, welche in obiger Abbildung 8⁸ zu sehen ist, ist in der Grundversion günstiger als die Vive, jedoch sind bei der Rift weder Controller noch Sensoren inkludiert, sodass bei zusätzlicher Bestellung dieser der Preis insgesamt höher gewesen wäre. Deswegen wurde gegen die Oculus Rift entschieden.

2.3.2.4 HTC Vive

Die HTC Vive ist der Oculus Rift ähnlich. Auch sie benötigt einen PC, der die Berechnungen der Anzeige übernimmt. Ebenfalls gleich ist die Ausstattung mit Gyro- und ein Beschleunigungssensoren zur Erfassung der Kopfbewegungen. Im Gegensatz zur Rift wird bei der HTC Vive Infrarotlicht von externen Sensoren ausgesandt und die Brille und Controller empfangen diese und geben es an den PC weiter, welcher die Positionen errechnet. Dieser Unterschied sorgt bereits dafür, dass die Vive auch mit zwei externen Geräten problemlos ihre Position erkennt, wobei bei der Rift drei benötigt werden. Des Weiteren sind bei der HTC Vive bereits die Sensoren und Controller beigelegt und müssen nicht zusätzlich bestellt werden.

⁸ http://picscdn.redblue.de/doi/pixelboxx-mss-70863644/fee_786_587_png/OCULUS-Rift-VR-Virtual-Reality-Headset - 10.07.2017



Abbildung 9: HTC Vive

Die HTC Vive –beispielhaft in Abbildung 9⁹– war die Brille, die am Ende gewählt wurde, da sie günstig und mit Unity einfach zu programmieren ist.

⁹ http://picscdn.redblue.de/doi/pixelboxx-mss-72183776/fee_786_587_png/HTC-Vive-Virtual-Reality-Brille-inkl.-Star-Trek%3A-Bridge-Crew - 10.07.2017

2.4 Grundlagen

2.4.1 Unity

Unity ist eine Laufzeit- und Entwicklungsumgebung für Spiele. Die Zielplattformen sind nicht nur PCs, sondern auch mobile Geräte und Webbrowser. Der Unity Editor ist gängigen 3D-Animationsprogrammen nachempfunden. Unity stellt mit dem Editor eine Möglichkeit zum einfachen Einstieg in die Entwicklung bereit. In diesem wird standardmäßig die Szene aus der Sicht des Bearbeiters angezeigt. Die Sicht des Bearbeiters ist komplett unabhängig von der Sicht des Kameraobjektes. So kann durch die Szene navigiert werden, ohne Einfluss auf die eigentlich Kamera, aus deren Sicht der Anwender später die Szene sieht, zu nehmen. Zudem werden einige Menüs zur Manipulation der Szene angezeigt. Darunter eine Baumstruktur mit bereits enthaltenen Objekten. Diese sind in der Standard-Szene eine Kamera und eine Lampe. Durch den Editor können dann nach Belieben verschiedenste andere Objekte in die Szene hinzugefügt werden. Auch komplett neue Szenen können erstellt werden. Objekte könne in der Sicht des Bearbeiters mit der Maus skaliert, verschoben und gedreht werden. Zudem verfügt der Unity-Editor über einen Assetstore, über den andere, aufwändigere, Objekte, eventuell auch mit Animationen oder Lichtern, heruntergeladen und importiert werden können.

Für die Entwicklung einer VR Anwendung für die HTC Vive muss aus dem Assetstore das SteamVR Asset heruntergeladen und importiert werden. Dieses enthält unter anderem bereits vorgefertigte Objekte, sogenannte Prefabs. Diese Prefabs ändern die Standardkamera zu der der VR-Brille und fügen die Objekte für die Controller hinzu. Auch Skripte zum Tracking bzw der Bewegung der Kamera sowie der Controller in der Szene sind bereits enthalten. Lediglich die Einstellung, welche die Hauptkamera repräsentiert muss noch händisch geändert werden.

2.4.2 Blender

Blender ist eine freie 3D-Grafiksuite. Mithilfe dieser Software können Körper modelliert, texturiert und animiert werden. Diese können dann später exportiert werden und in anderen 3D-Programmen verwendet werden. Blender kann jedoch, im Gegensatz zu Unity nicht nur 3D-Dateiformate wie OBJ importieren, sondern auch Dateien, welche aus CAD Systemen stammen. Aus diesem Grund wird Blender im Rahmen dieser Studienarbeit im Hintergrund genutzt. So spricht das Skript, welches zur Umwandlung von STL zu OBJ Dateien verwendet wird, eine Schnittstelle von Blender an, welche eine STL Datei importiert und als OBJ Datei wieder exportiert.

2.4.3 Dateiformate

Die unterschiedlichen Dateiformate im Umfeld von Visualisierungen spielen eine große Rolle. So basieren Dateiformate, welche in CAD-Systemen verwendet werden, oft auf der geometrischen Beschaffenheit eines Objektes. Ein Kreis wird so beispielsweise über seinen Mittelpunkt und einen Durchmesser beschrieben. Im Gegensatz dazu sind Formate, welche von Grafikengines verwendet werden fast immer Vertex basiert. Ein Kreis wird hier also durch eine Vielzahl von Vektoren beschrieben. Je höher die Anzahl der Vektoren, desto runder wirkt der Kreis. Im Falle eines CAD-Formates übernimmt die Umrechnung von geometrischen Eigenschaften zu Vektoren das CAD-System.

Aus diesem Grund müssen zwei Dateiformate für dieses Projekt ausgewählt werden. Zum einen ein Format, welches die Grundlage für die weiteren Objekte bildet. Zum anderen ein Format, welches von Unity importiert werden kann.

Das Format für die Grundlage muss von Catia exportiert werden können, da Catia an der Hochschule als CAD-System eingesetzt wird. In diesem Zuge wurde das STL-Format ausgewählt. Der Vorteil ist, dass dieses Format Objekte über Dreiecksfacetten beschreibt, ähnlich zu den Vektoren von 3D-Formaten. Der Nachteil jedoch ist, dass gekrümmte Flächen durch die Eigenschaften eines Dreiecks nur angenähert werden können.

Unity könnte zwar nativ verschiedenste CAD-Formate importieren, allerdings nicht dynamisch zur Laufzeit. Aus diesem Grund musste ein Weg gefunden werden, Dateien zur Laufzeit zu importieren. Der einfachste Weg dies zu realisieren ist, die STL-Dateien mit Blender in OBJ-Dateien zu konvertieren, da es einige OBJ-Importer

im Assetstore gibt, die versprechen, auch zur Laufzeit laden zu können. Aus diesem Grund wurde das zweite Dateiformat auf OBJ festgelegt.

2.4.4 SteamVR

Mit Steam hat sich HTC einen starken Partner an ihre Seite geholt. Dieser bietet mit SteamVR eine der zwei von der HTC Vive unterstützten Softwareschnittstellen an. SteamVR bildet die Schnittstelle vom Entwickler zur HTC Vive. So können über diese Schnittstelle alle erdenklichen Parameter und Zustände der VR-Brille abgerufen werden. Beispielsweise können die Koordinaten der Brille im Raum abgefragt, oder einzelne Funktionen gesteuert werden. SteamVR ist jedoch nicht nur eine Softwareschnittstelle. Vielmehr bietet es ein Forum mit einer breiten Wissensbasis zur Spieleentwicklung an sich und ein privates Unterforum für die Entwickler der HTC Vive.

Zur Nutzung der HTC Vive muss das SteamVR Plugin für Steam heruntergeladen und installiert werden. Dieses definiert dann über ein Tutorial die Größe des Raumes und dessen Ausrichtung. Auch Einstellungen, ob der Nutzer im Raum mit Hilfe der Lightboxen getrackt werden kann oder ob nur stationäre Bewegungen möglich sind werden hier festgelegt.

Zudem bietet SteamVR eine Art Desktop für den virtuellen Raum. Dieser „Home room“ ist vom Nutzer anpassbar und ermöglicht ihm, Anwendungen direkt in der virtuellen Realität zu starten, ohne ständig die VR-Brille an bzw. ausziehen zu müssen.

2.5 Demonstrationsfähige Szene

2.5.1 Anforderungen

Um die Nutzung für andere Projekte zu ermöglichen, soll eine Beispielszene erstellt werden in der die Möglichkeiten der VR Lösung demonstriert werden. Im Hinblick auf eine Kooperation mit dem GFR Team soll ermöglicht werden, dass Objekte in die Szene geladen werden können, welche zuvor aus der CAD Software Catia exportiert wurden. Diese sollen in der Szene selbst bewegt und skaliert werden können. Da in der Szene noch andere Objekte existieren könnten, und sich diese, um die Immersion zu erhöhen, realistisch verhalten sollen, ist es notwendig, dass alle Objekte in der Szene miteinander interagieren. Darunter ist zu verstehen, dass Gegenstände einen festen Körper haben sich nicht durch andere hindurch bewegen können. Auch eine vereinfachte Form eines physikalischen Modells ist wünschenswert.

An den Grundgedanken einer Kooperation mit dem GFR Team anknüpfend ist davon auszugehen, dass sich die Modelle im Laufe der Zeit ändern und nicht fest in die Szene integriert werden können. Aus diesem Grund wurde die Möglichkeit gefordert, Objekte dynamisch zur Laufzeit aus einem zu definierenden Speicherort laden zu können.

Insgesamt sollte die fertige Szene möglichst einfach gestartet werden können, um schnelle Präsentationen zu ermöglichen.

2.5.2 Umwandlung der Dateien

Wie bereits in den Grundlagen erwähnt, wird zum Umwandeln der STL-Dateien aus Catia Blender verwendet. Blender hat die Eigenschaft, dass es sowohl CAD-Dateien, als auch 3D-Dateien lesen kann. Somit kann eine CAD-Datei, in diesem Fall eine STL-Datei importiert werden, und als OBJ-Datei wieder exportiert. Da es äußerst mühselig wäre, jede Datei von Hand in Blender zu importieren und umgewandelt wieder zu exportieren, wurde die Python Schnittstelle von Blender genutzt. Über ein Python Skript können verschiedene Komponenten von Blender im Hintergrund aufgerufen werden. Im Skript zur Umwandlung von STL-Dateien zu OBJ-Dateien werden zwei verschiedene Schnittstellen von Blender verwendet. Zum Importieren der STL-Datei wird die *stl* Funktion der *import_mesh* Schnittstelle, welcher der Dateipfad der STL-Datei übergeben wird, verwendet. Diese Funktion lädt, wie es der Name bereits sagt, ein Mesh, in diesem Fall aus einer STL-Datei. Zum Exportieren der Szene als OBJ-Datei wird die Funktion *obj* der *export_scene* Schnittstelle verwendet. Diese Exportiert die Szene in eine OBJ-Datei, welche im übergebenen Dateipfad erstellt wird.

Um die Verwendung des Python Skripts so komfortabel und einfach wie möglich zu gestalten, wurde ein weiteres Skript erstellt. In diesem Batch-Skript werden die Dateinamen der STL-Datei, sowie der gewünschten OBJ-Datei abgefragt. Im Anschluss daran wird im Unterordner OBJFiles ein Ordner mit dem Namen der OBJ-Datei angelegt. Hier werden die von Blender erstellten OBJ-Datei, sowie die dazugehörige MTL-Datei abgelegt.

Die Skripte und auch OBJ-Dateien befinden sich im Dateiconverter Ordner des Projektes. Zudem ist eine ReadMe.txt beigelegt, welche die Voraussetzungen der Skripte und die Anwendung erklärt.

2.5.3 Laden von Modellen

2.5.3.1 Unity-Unterstützung

Unity bietet bereits nativ die Möglichkeit, Objekte zur Laufzeit nachzuladen. Dieser Ansatz, die von Unity unterstützten Methoden zum Laden von Modellen zur Laufzeit zu verwenden, wurde leider schnell verworfen. Das Problem hierbei ist, dass Unity nur Dateien zur Laufzeit laden kann, welche bereits im Vorfeld bekannt sind. Diese müssen in einem Verzeichnis mit dem Namen „Resources“ im Projektordner liegen. Beim Kompilieren des Projektes werden diese Ressourcen dann in eine Datei verpackt, welche der ausführbaren Anwendung beiliegt. Diese Datei weist eine Baumstruktur auf, welche den Zugriff beschleunigt. Aus dieser Datei können nun beliebig Modelle zur Laufzeit nachgeladen werden [1]. Da die Anforderung allerdings vorschreibt, dass Modelle dynamisch zur Laufzeit geladen werden sollen, ist diese Lösung nicht geeignet. Dynamisch heißt in diesem Fall, dass die Modelle in OBJ-Format von der ausführbaren Anwendung geladen werden können. Mit der von Unity angebotenen Lösung müsste jedes Mal, wenn sich die OBJ-Dateien ändern, das gesamte Projekt neu kompiliert werden.

2.5.3.2 Third-Party Bibliotheken

Da Unity keine eigene verwendbare Lösung zum Laden von Objekten zur Laufzeit anbietet, muss auf Bibliotheken von Drittanbietern zurückgegriffen werden [2]. Hierzu wurden zwei verschiedene Lösungswege verfolgt. Zum einen wurde eine von einem Mitglied der Community veröffentlichte Klasse verwendet. Die von *el anónimo* veröffentlichte Klasse *ObjImporter* [1] ermöglicht es, OBJ-Dateien zur Laufzeit zu laden. Allerdings können, bedingt durch die von Unity auf ca. 65.000 limitierte Anzahl von Vertices pro Objekt, nur bestimmte Objekte geladen werden. Sollten die Objekte in den OBJ-Dateien größer als das Limit sein, können diese nicht geladen werden. Aus diesem Grund wurde ein Modul aus dem Assetstore getestet. Der *Runtime OBJ Loader* [2] von „aaro4130“ ermöglicht auch das Laden von sehr großen Objekten. Zwar dauert dies eine Weile, aber, vorausgesetzt die Datei weist keine Fehler aus, das Objekt wird korrekt geladen. Hierbei werden viele kleine Objekte erstellt, welche später in einem leeren Objekt zum eigentlichen Modell wieder zusammengefügt werden. Leider hat auch dieser Lösungsweg eine Schattenseite. So kann dieses Modul zwar problemlos Objekte laden, solange es im Editor ausgeführt wird, allerdings funktioniert dies nichtmehr, sobald das Projekt kompiliert wurde. Der

ausgegebene Fehler konnte leider nichtmehr im zeitlichen Rahmen der Studienarbeit behoben werden.

Aus diesen Gründen existieren zum Zeitpunkt der Abgabe zwei verschiedene Versionen des Projektes. Zum einen mit dem *ObjImporter* von *el anónimo*, welcher allerdings keine großen Objekte laden kann, und zum anderen die Version mit dem *Runtime OBJ Loader* von *aaro4130*. Welcher der beiden Lösungsansätze in Zukunft weiter verfolgt wird, hängt von den Studierenden ab, welche diese Studienarbeit übernehmen.

2.5.4 Interaktion

Die Interaktion der Objekte untereinander basiert auf der Standard Physics Engine von Unity. Unity berechnet automatisch für alle Objekte, welche einen sogenannten Rigidbody besitzen, die Schwerkraft und lässt diese so nach unten fallen. Kollisionen mit anderen Objekten kann die Engine ebenfalls selbst berechnen, wenn ein Collider zum Objekt hinzugefügt wurde. Mit diesen Ausgangsbedingungen kann bereits ein Raum erstellt werden, in welchem Objekte auf den Boden fallen und dort liegen bleiben.

Für die Interaktion des Nutzers mit den Objekten wurde als Grundlage das Virtual Reality Toolkit benutzt. Ausgehend von diesem wurde ein Skript entwickelt, welches zu einem Körper hinzugefügt werden kann und in welchem dann abgefragt wird, ob der Controller der Vive sich innerhalb dieses Körpers befindet. Wenn sich der Controller im inneren befindet, dann wird abgefragt ob ein Button am Controller gedrückt wird. Welcher Button das sein soll ist innerhalb von Unity mit einem Dropdown-Menü auswählbar. Dies geschieht über ein Enumeration mit allen verfügbaren Buttons und einer Variable mit Sichtbarkeit public mit diesem Enumeration. Dadurch lässt sich im Inspector von Unity ein Wert der Liste auswählen. Wenn der ausgewählte Button innerhalb eines Objekts gedrückt wird, so wird der in dem Moment vorhandene Offset zwischen dem Zentrum des Controllers und dem Zentrum des Objekts gespeichert. Anschließend wird das Objekt so lange mit dem Controller mitbewegt, wie der Button gedrückt bleibt. Der gespeicherte Offset wird dabei kann dabei je nach Einstellung beibehalten oder ignoriert werden. Sobald das Objekt losgelassen wird, kann es auch wieder nach unten fallen, da beim Tragen die Schwerkraft ignoriert wird.

Um Objekte vergrößern und verkleinern zu können, wurde ein zweites Skript geschrieben, welches aufgerufen wird, wenn sich zwei Controller innerhalb des Objekts befinden und bei beiden der entsprechende Button gedrückt ist. Dieses zweite Skript speichert sich dann den Abstand zwischen beiden Controllern und vergrößert beziehungsweise verkleinert das Objekt anschließend um einen Faktor der Längenveränderung der Distanz zwischen den Controllern.

2.6 Zusammenfassung

Abschließen kann gesagt werden, dass viele der gesteckten Ziele erfüllt wurden. Mit der HTC Vive wurde ein Technologieupgrade gefunden und auch bereits verbaut. Es wurde ein Skript erstellt, mit dem STL-Dateien in OBJ-Dateien konvertiert werden können. Dazu wurde ein weiteres Skript erstellt, welches die Nutzung des ersten Skripts vereinfacht. Hier werden nach und nach die benötigten Daten vom Nutzer abgefragt. Auch eine präsentationsfähige Szene wurde erstellt. Diese kann zwar kompiliert werden, allerdings funktioniert dann leider das dynamische Laden von Modellen zur Laufzeit nicht (siehe 2.5.3). Aus diesem Grund muss momentan die Anwendung aus dem Unity Editor gestartet werden.

Die Szene an sich erfüllt alle Anforderungen. Es kann mit Objekten interagiert werden indem man sie mit einem Controller aufhebt. Greift man mit dem zweiten Controller nach einem Objekt kann dieses skaliert werden. Auch untereinander interagieren die Objekte. So hat jedes Objekt einen eigenen Körper, welcher nicht durchdrungen werden kann. Aus diesem Grund können keine Objekte durch den Boden fallen oder durch Wände geworfen werden. Zudem besitzt jedes Objekt ein einfaches physikalisches Modell, welches die Schwerkraft simuliert.

2.7 Ausblick

2.7.1 Zusammenarbeit mit anderen Studienarbeiten

2.7.1.1 GFR

Es war ursprünglich geplant worden mit dem GFR Team des Campus Friedrichsafen zusammenzuarbeiten. Diese Zusammenarbeit konnte jedoch aufgrund zeitlicher Probleme nicht durchgeführt werden.

Das GFR Team plant momentan ihre Rennwagen mit Catia. Da ein reales Modell nicht zu viel Zeit, sondern auch viel Geld kosten würde, könnte das Team die Möglichkeiten der VR-Technik nutzen, um sich vorab das Modell in Originalgröße anzusehen und eventuell Anpassungen vornehmen.

2.7.1.2 HeliSim

Auch eine Zusammenarbeit mit dem Helikopter Simulator wäre denkbar. Der HeliSim wird zurzeit mit einer Vielzahl an Beamern betrieben, die sich um ein ausrangiertes Helikopter Chassis befinden. Die Beamer könnten mit einer VR Brille für den Piloten und idealerweise einer weiteren für den Copiloten ersetzt werden. Dabei müsste ein Übergang von reiner Virtual Reality zu Augmented Reality gemacht werden, da der Pilot seinen Copiloten neben sich, wie auch das Cockpit um sich herum wahrnehmen können sollte. So kann auch der Umgang mit den Instrumenten am echten Helikopter geübt werden, wenn diese innerhalb der VR Umgebung ebenso dargestellt sind.

2.7.1.3 Lego Mindstorms

In der Zusammenarbeit mit den Lego Mindstorms Projekten wäre es beispielsweise denkbar, einen Roboter mit Kamera zu konstruieren, der von einem Nutzer mit VR Brille gesteuert wird. Bewegungen des Roboters könnten über die Bewegungen des VR-Brillenträgers gesteuert werden. Die Bewegungen der Kamera könnten über die Kopfbewegungen kontrolliert werden.

2.7.2 Erweiterungen

Außer diesen Zusammenarbeiten mit anderen Studienarbeiten, von denen noch weitere denkbar wären und hier nur einige Beispiele genannt werden, könnte auch das Stereo X Projekt an sich weiter entwickelt werden. Dabei sollen hier auch nur Beispiele genannt werden, da sich die Technik in den kommenden Jahren noch

deutlich weiterentwickeln wird und somit viele weitere Anwendungsgebiete erschlossen werden.

Eine zweite Brille könnte beschafft werden um somit mehreren Personen gleichzeitig das Betrachten einer Szene zu ermöglichen. Die Personen könnten dann auch die Möglichkeit haben sich untereinander zu sehen und miteinander zu interagieren. Damit würde auch gleichzeitig der Übergang zu Augmented Reality geschaffen und es könnten auch externe Objekte in die Szene eingebunden werden und somit als Unterlage oder andere Hilfsmittel zu dienen. HTC arbeitet bereits an einer Möglichkeit Objekte in die Szene zu übernehmen. In der unten stehenden Abbildung ¹⁰ ist ein sogenannter Vive Tracker zu sehen, der auf Objekten befestigt werden kann, somit der Position ermittelt und diese in einer Szene verwendbar macht.



Abbildung 10: Vive Tracker

Eine zusätzliche kleine Verbesserung, wäre die Möglichkeiten aus der Szene heraus Objekte von egal welchem Ort auf dem PC zu laden, auch von externen Datenträgern wie USB-Sticks.

¹⁰ https://www.vive.com/media/filer_public/d1/ec/d1ecfd2e-4bed-479f-9c18-2e81c8b952ad/tracker-s03.png - 11.07.2017

2.7.3 Upgrades

Der Einsatz neuer Technologien, um eine portablere Möglichkeit zu haben, wäre auch denkbar. So entwickelt HTC zurzeit eine VR Brille, die keinen PC mehr benötigt und deswegen an vielen Orten benutzt werden könnte. Auch die Hololense wäre ein denkbare Upgrade, wenn sie bezahlbarer und frei verfügbar wird, um Portabilität zu gewährleisten. Mit der Hololense hätte man außerdem den Umstieg zu Augmented Reality, womit man alles in der realen Welt um sich herum wahrnimmt während man immer noch virtuelle Objekte einfügen kann. In den kommenden Jahren dürften sowohl in dem Bereich der portablen VR wie auch AR Brillen noch weitere Anbieter Produkte entwickeln, sodass die Auswahl größer und die Preise höchstwahrscheinlich geringer werden. Auch die Entwicklung ganz neuer Technologien könnten Möglichkeiten für ein Upgrade bieten.

Abbildungsverzeichnis

Abbildung 1: Schema des Sytemaufbaus	14
Abbildung 2: Kinect 2 for Windows	15
Abbildung 3: Kinect Erkennung	16
Abbildung 4: Zwei Möglichkeiten des Motion Tracking	19
Abbildung 5: Vive Kontroller	19
Abbildung 6: Samsung Gear VR.....	21
Abbildung 7: Playstation VR Headset.....	22
Abbildung 8: Oculus Rift	23
Abbildung 9: HTC Vive	24
Abbildung 10: Vive Tracker.....	35

Literaturverzeichnis

- [1] N. Kubasta und P. Ziegeldorf, „StereoX - HTC Vive“, 14 Juli 2017. [Online]. Available: <https://github.com/StereoXDHBWFN/HTC-Vive>. [Zugriff am 14 Juli 2017].
- [2] Unity, „Unity3D Dokumentation“, Unity Technologies, 11 Juli 2017. [Online]. Available: <https://docs.unity3d.com/Manual/LoadingResourcesatRuntime.html>. [Zugriff am 14 Juli 2017].
- [3] R. M. L, „Unity Answers“, Unity Technologies, 24 März 2014. [Online]. Available: <http://answers.unity3d.com/questions/671833/load-3d-model-from-external-drive-in-runtime-witho.html>. [Zugriff am 14 Juli 2017].
- [4] e. anónimo, „Unity3D Wiki“, Unity Technologies, 10 Januar 2012. [Online]. Available: <http://wiki.unity3d.com/index.php?title=ObjImporter>. [Zugriff am 14 Juli 2017].
- [5] aaro4130, „Unity3D Forum“, Unity Technologies, 15 Juli 2010. [Online]. Available: <https://forum.unity3d.com/threads/free-runtime-obj-loader.365884/>. [Zugriff am 14 Juli 2017].
- [6] Microsoft, „Kinect-Hardware“, Microsoft, 13 Juli 2017. [Online]. Available: <https://developer.microsoft.com/de-de/windows/kinect/hardware>. [Zugriff am 13 Juli 2017].