Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Informatik

Immersive Engineering Collaboration Tool

Bachelorarbeit FS2019

Kevin Huber

Betreuer: Markus Zank

Hochschule Luzern - Departement Informatik
7. Juni 2019
Version 1.0.0

FH Zentralschweiz

EFQM
Recognised for excellence
4 star

Bachelorarbeit an der Hochschule Luzern - Informatik

Titel: Immersive Engineering Collaboration Tool

Student: Kevin Huber

Studiengang: BSc Informatik

Abschlussjahr: 2019

Betreuungsperson: Markus Zank

Experte: Diego Schmidlin

Codierung / Klassifizierung der Arbeit:

- ☒ **A: Einsicht (Normalfall)**
- ☐ **B: Rücksprache** (Dauer: Jahr / Jahre)
- ☐ **C: Sperre** (Dauer: Jahr / Jahre)

Eidesstattliche Erklärung
Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt haben, alle verwendeten Quellen, Literatur und andere Hilfsmittel angegeben haben, wörtlich oder inhaltlich entnommene Stellen als solche kenntlich gemacht haben, das Vertraulichkeitsinteresse des Auftraggebers wahren und die Urheberrechtsbestimmungen der Fachhochschule Zentralschweiz (siehe Markblatt «Studentische Arbeiten» auf MyCampus) respektieren werden.

Ort / Datum, Unterschrift: _____

Abgabe der Arbeit auf der Portfolio Datenbank

Bestätigungsvisum Student
Ich bestätige, dass ich die Bachelorarbeit korrekt gemäss Merkblatt auf der Portfolio Datenbank abgelegt habe. Die Verantwortlichkeit sowie die Berechtigungen habe ich abgegeben, so dass ich keine Änderungen mehr vornehmen kann oder weitere Dateien hochladen kann.

Ort / Datum, Unterschrift: _____

Verdankung
Ich bedanke mich bei meiner Betreuungsperson Markus Zank für die diversen Inputs, Denkanstösse und Ratschläge während meines Projekts. Ebenfalls bedanken möchte ich mich bei meinem Experten Diego Schmidlin für die Inputs nach meiner Zwischenpräsentation. Weiterhin bedanke ich mich bei allen Teilnehmern meiner beiden Nutzerevaluationen für die Teilnahme und Mithilfe in meinem Projekt. Zu guter Letzt bedanke ich mich bei allen Studierenden, welche mir während meiner Arbeit im Immersive Realities Lab immer wieder zur Seite standen, falls ich an einem Punkt im Projekt fragen hatte.

Eingangsvisum (durch das Sekretariat auszufüllen):

Rotkreuz, den _____ Visum: _____

Hinweis: Die Bachelorarbeit wurde von keinem Dozierenden nachbearbeitet. Veröffentlichungen (auch auszugsweise) sind ohne das Einverständnis der Studiengangleitung der Hochschule Luzern – Informatik nicht erlaubt.

Copyright © 2019 Hochschule Luzern - Informatik

Alle Rechte vorbehalten. Kein Teil dieser Arbeit darf ohne die schriftliche Genehmigung der Studiengangleitung der Hochschule Luzern - Informatik in irgendeiner Form reproduziert oder in eine von Maschinen verwendete Sprache übertragen werden.

Abstract

In der vorliegenden Arbeit wurde eine Multi-User taugliche Applikation in Virtual Reality entwickelt. Die Applikation ist darauf ausgelegt, Objekte in der virtuellen Realität gemeinsam zu betrachten und manipulieren und währendem miteinander über die Objekte zu diskutieren. Um das Projekt umzusetzen wurde die Virtual Reality Brille HTC Vive eingesetzt. Dank dem Einsatz des Steam VR Assets kann das Projekt aber ohne grosse Probleme mit anderen Virtual Reality Brillen gestartet werden.

Das Projekt wurde in zwei Phasen entwickelt. Die erste Phase widmete sich der Interaktion einer Person mit den Objekten in der virtuellen Realität. Anhand einer Nutzerevaluation wurde evaluiert, welche von drei Varianten sich am besten eignete, um dem Nutzer mitzuteilen, dass zwei Objekte miteinander kollidiert sind. Durchgesetzt hat sich die Variante, bei welcher dem Nutzer via Vibration am entsprechenden Kontroller ein haptisches Feedback gegeben wird, sobald er mit dem Objekt in seiner Hand ein anderes Objekt berührt.
Die zweite Phase widmete sich der Multi-User Funktionalität. Für die Netzwerkkommunikation zwischen den verschiedenen Usern wurde die Photon-Engine eingesetzt, welche eine einfache Anbindung an Unity erlaubte. Damit die Benutzer miteinander Kommunizieren können, wurde das Photon Voice Plugin verwendet. Falls zwei Benutzer gleichzeitig dasselbe Objekt manipulieren wollen, musste eine Lösung gefunden werden, damit keine Inkonsistenzen zwischen den Benutzern entsteht, falls einer das Bauteil an einen anderen Ort bewegt als der andere Benutzer. Es wurden zwei Varianten erstellt, welche anhand von Nutzertests verglichen wurden. Mit den Nutzertests hat sich herausgestellt, dass es reicht, wenn das Bauteil für Interaktionen von anderen Benutzern gesperrt wird, sobald es von einem Benutzer gepackt wird. Die andere Variante, bei welcher nur ein Benutzer in der virtuellen Umgebung mit den Bauteilen interagieren kann, hat sich für die Aufgabe, eine Maschine gemeinsam zusammenzubauen, nicht bewährt. Diese Variante könnte aber in einer Lern- oder Präsentationsumgebung eingesetzt werden, um unerwünschte Interaktionen von anderen Benutzern zu unterbinden.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Problem . . . 1
1.2 Fragestellung . . . 1
1.3 Vision . . . 1

2 Stand der Technik **2**
2.1 Virtuelles Engineering . . . 2
2.2 Zusammenarbeit in der virtuellen Realität . . . 3
2.2.1 Collaborative User Interaction . . . 3
2.2.2 Avatar Repräsentation . . . 4
2.2.3 Multi-User Anwendungen . . . 4

3 Methoden **5**
3.1 Vorgehensmodell . . . 5
3.2 Rahmenplan . . . 5
3.3 Meilensteine . . . 6
3.4 Anforderungskatalog . . . 6
3.5 Risikoanalyse . . . 7
3.5.1 Mögliche Risiken . . . 7
3.5.2 Risikoverminderung: Mögliche Massnahmen . . . 7
3.6 Verwendete Hardware . . . 8

4 Ideen und Konzepte **9**
4.1 Single-User Prototyp . . . 9
4.1.1 Highlight . . . 9
4.1.2 Snapping . . . 9
4.1.3 Kollision . . . 10
4.1.4 Feedback beim Zusammenbau / Manipulation der Objekte . . . 10
4.2 Multi-User Prototyp . . . 11
4.2.1 Avatar-Repräsentation in der virtuellen Umgebung . . . 11
4.2.2 Kommunikation zwischen den Benutzern . . . 12
4.2.3 Gleichzeitige Interaktion am selben Objekt . . . 12

5 Single User Interaktion **15**
5.1 Realisierung . . . 15

5.1.1 SteamVR-Plugin . . . 15
5.1.2 Highlight . . . 15
5.1.3 Snapping . . . 16
5.1.4 Kollision . . . 17
5.1.5 Baugruppen . . . 17
5.1.6 Feedback beim Zusammenbau / Manipulation der Objekte . . . 18
5.2 Evaluation . . . 20
5.2.1 Aufgabenstellung . . . 20
5.2.2 Resultat . . . 21
5.2.3 Schlussfolgerung . . . 22
5.3 Systemarchitektur . . . 22
5.4 Klassendiagramm . . . 23

6 Multi User Interaktion **24**
6.1 Realisierung . . . 24
6.1.1 Multi User . . . 24
6.1.2 Avatar-Repräsentation in der virtuellen Umgebung . . . 26
6.1.3 Kommunikation zwischen den Benutzern . . . 26
6.1.4 Gleichzeitige Interaktion am selben Objekt . . . 27
6.2 Evaluation . . . 28
6.2.1 Aufgabenstellung . . . 28
6.2.2 Resultat . . . 28
6.2.3 Schlussfolgerung . . . 30
6.3 Systemarchitektur . . . 30
6.4 Klassendiagramm . . . 31

7 Schlussfolgerung **32**

8 Ausblick **33**

1 Einleitung

1.1 Problem

Virtuelle 3D-Modelle werden heutzutage an einem Computer betrachtet oder umständlich modelliert, um einem anderen Benutzer gezeigt werden zu können. Die Betrachtung am Computer ist aber sehr umständlich und der Benutzer kann sich das Modell oftmals nicht vorstellen. Soll das Produkt frühzeitig einem Kunden gezeigt werden, muss meistens ein physisches Mockup des virtuellen Modells angefertigt werden. Ein physisches Mockup hilft dem Benutzer oder dem Kunden dann die Sachlage besser zu verstehen, kostet aber meist viel Geld und Zeit.

1.2 Fragestellung

Ziel der Arbeit ist es ein Multi-User taugliches Interaktionssystem in Virtual Reality zu entwickeln, um gemeinsam besser über virtuelle Modelle diskutieren zu können. Dabei soll der Fokus auf intuitiver Interaktion, sowie dem gemeinsamen Zerlegen bzw. Zusammenfügen des Modells liegen. Die Umsetzung dieser Arbeit wird am Beispiel einer technischen Baugruppe gezeigt.

Einerseits wird untersucht welche Art von Interaktion sich am besten eignet, um es dem Benutzer so intuitiv wie möglich zu machen, mit dem Modell zu arbeiten. Andererseits soll untersucht werden, welche Art von gemeinsamer und gleichzeitiger Interaktion sich für diese Applikation am besten eignet.

1.3 Vision

Aus der Arbeit soll ein Prototyp hervorgehen, mit welchem mehrere Personen in der virtuellen Realität über einfache Modelle diskutieren können. Diese Diskussionen sollen auch möglich sein, falls sich nicht alle Personen am gleichen Ort befinden. Mit dem Prototyp soll auch evaluiert werden können, ob eine solche oder ähnliche Applikation in der Wirtschaft eingesetzt werden kann.

2 Stand der Technik

Forschungen über Kollaborative Engineering Tools gibt es direkt keine. Die Forschungen können aber in zwei Bereiche aufgeteilt werden. Zum einen gibt es das virtuelle Engineering und zum anderen die Zusammenarbeit in der virtuellen Realität.

2.1 Virtuelles Engineering

Im Bereich des virtuellen Engineerings gibt es verschiedene Forschungen welche sich mit dem Thema Zusammenbau und insbesondere Kollision zwischen den Elementen beschäftigen. Eine Arbeit befasst sich mit der Berechnung der Kollisionen zwischen Elementen, welche aus einem CAD Model importiert wurden (Tching, Dumont und Perret, 2010). Wie in Abbildung 2.1 zu sehen ist, nimmt die Genauigkeit der importierten Modelle aber ab, da Game-Engines wie Unity nur geometrische Approximationen des CAD-Modells benutzen. Für die Kollision in Virtual Reality werden daher einfache “Collider“ empfohlen, da die Kollision sonst schnell unnatürlich wirkt.

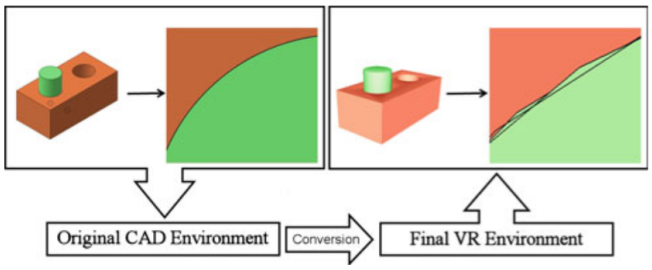

Abbildung 2.1: Verlust der Genauigkeit bei der Konvertierung eines CAD-Models zu VR

In der Arbeit “Virtual reality and augmented reality as a training tool for assembly tasks“ (Boud, Haniff, Baber und Steiner, 1999) wurde bereits 1999 untersucht, ob das virtuelle Assembly Training einen Mehrwert bringt gegenüber dem Training mit Modellen in der realen Welt. Nach ihrer Aussage hat das virtuelle Training Zukunft, da das Training bereits gestartet werden kann, ohne dass ein Prototyp vorhanden sein muss. Aktuell, im Jahre 1999, sei die Technik im Bereich der virtuellen Realität aber noch nicht bereit dafür.

VADE (Tirumali, Lyons und Hart, 1999) ist eine virtuelle Assembly Designumgebung. In dieser werden je nach Situation die Freiheitsgrade der Bewegung des Bauteils eingeschränkt um das Zusammenbauen diverser Bauteile zu erleichtern. In der Abbildung 2.2 kann das Bauteil in der rechten Hand nur auf den hervorgehobenen Achsen bewegt werden.

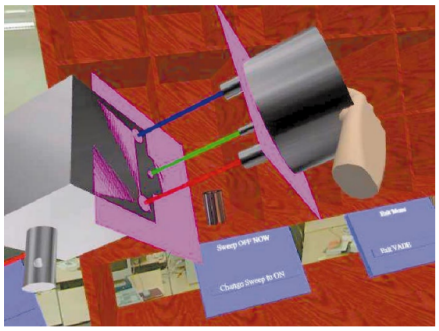

Abbildung 2.2: Einschränkungen beim Zusammenbau in VADE

2.2 Zusammenarbeit in der virtuellen Realität

Für die Zusammenarbeit in der virtuellen Realität gibt es drei Stufen. Die erste Stufe definiert wie sich die Benutzer in der virtuellen Realität wahrnehmen. Die zweite Stufe ist die alleinige Manipulation in der virtuellen Umgebung und die dritte Stufe die gleichzeitige Manipulation eines Objektes in dieser virtuellen Umgebung.

2.2.1 Collaborative User Interaction

Um eine gleichzeitige Interaktion am selben Objekt zu ermöglichen hat Márcio S. Pinho bereits 2002 (Pinho, Bowman und Freitas, 2002) eine Variante beschrieben, bei welcher die Freiheitsgrade der Benutzer separiert werden. Bei einem Würfel wäre das zum Beispiel wie folgt aufgeteilt. Ein Benutzer rotiert den Würfel während der andere Benutzer den Würfel in der virtuellen Welt bewegen kann.

Eine andere Arbeit befasste sich ebenfalls mit der gleichzeitigen Manipulation eines Objektes und beschreibt ein Lösungsansatz, bei welchem der Mittelwert aller Nutzer genommen und auf das Objekt übertragen wird (Ruddle, Savage und Jones, 2002).

Um der Realität so nahe wie möglich zu kommen gibt es diverse Arbeiten, welche sich damit beschäftigen, getrackte Objekte in der realen Welt zu verwenden um virtuelle Objekte zu manipulieren (He, Zhu und Perlin, 2017) (Podkosova et al., 2016). In der realen Welt befindet sich ein abstrahiertes Objekt, welches ein Objekt in der virtuellen Realität darstellt. Wollen mehrere Nutzer dieses Objekt manipulieren, greifen sie das Objekt in der realen Welt und haben so ein natürliches Verhalten. Das Objekt in der realen Welt sollte aber dem Objekt in der virtuelle Welt sehr ähnlich sein, da sonst die Immersion der virtuellen Realität zerstört wird (Simeone, Velloso und Gellersen, 2015). Für diese Variante müssen sich alle Nutzer im selben Raum befinden.

2.2.2 Avatar Repräsentation

Wie wird ein Benutzer einem anderen Benutzer in der virtuellen Realität dargestellt? Für viele Anwendungen reicht es den Kopf und die Hände des anderen Nutzers zu sehen. Andere Anwendungen verlangen einen komplexere 3D-Körper. Menschliche Körper besitzen mehrere hundert Muskeln und Gelenke. Einen solchen Avatar in VR umzusetzen ist daher sehr schwierig. Die Doktorarbeit von Weiya Chen beschreibt im Kapitel 1.4.1 User Representation die verschiedenen Varianten der Avatar Repräsentation (Chen, 2015).

Heutzutage gibt es Systeme, um das Gesicht und den Körper in der virtuellen Realität möglichst real zu animieren. Die Arbeit «Interactive Virtual Humans in Real-Time Virtual Environment» gibt eine gute Übersicht darüber was alles gemacht werden kann und möglich ist (Magnenat-Thalmann und Egges, 2015).

2.2.3 Multi-User Anwendungen

Studenten aus Deutschland haben eine Multi-User Anwendung im Bereich des Medical-Trainings geschrieben und dazu ein Paper veröffentlicht (Schild et al., 2018). Die Anwendung sollte demonstrieren, dass Virtual Reality für solche Zwecke eingesetzt werden kann.

DIVE ist ein sehr altes Multi-User System aus dem Jahre 1993 (Carlsson und Hagsand, 1993). Es diente als Plattform um verschiedene Experimente mit virtuellen Umgebungen durchzuführen.

ImmersiveDeck ist eine Applikation, in welcher bis zu drei Personen gleichzeitig sich frei bewegen und miteinander interagieren können (Podkosova et al., 2016). Für das genaue Tracking der Personen sind am Körper diverse Marker angebracht, welche die Personen präzise in der virtuellen Welt abbilden (zu sehen in Abbildung 2.3). Für die Interaktion mit den Objekten werden virtuelle Objekte abstrahiert und in der realen Welt mit Marker versehen. So können die Objekte auch in der realen Welt gepackt werden und die Interaktion fühlt sich natürlich an.

Abbildung 2.3: ImmersiveDesk: Tracking der Personen mithilfe von Markern

3 Methoden

3.1 Vorgehensmodell

Als Vorgehensmodell wurde SoDa gewählt, da im Projekt über mehrere Iterationen zwei Prototypen weiterentwickelt werden sollen. Im ersten Teil des Projektes wird ein Single-User Prototyp entwickelt und im zweiten Teil ein Multi-User Prototyp, welcher auf dem ersten Prototypen und dessen Erkenntnissen aufbauen wird.
Nach dem ersten sowie dem zweiten Teil des Projektes wird jeweils eine Nutzerevaluation durchgeführt, um die verschiedenen Varianten miteinander zu vergleichen und Rückmeldungen zum Prototypen selbst zu bekommen.

3.2 Rahmenplan

Das Projekt besitzt die folgenden Ereignisse:

Ereignis	Semesterwoche	Termin
Initialisierung	SW01 - SW03	18.02.19 - 08.03.19
Iterationen(Sprints)	SW04 - SW13	11.03.19 - 17.05.19
Zwischenpräsentation	SW07 - SW10	24.04.19
Projektabschluss	SW16	07.06.19

Tabelle 3.1: Rahmenplan

Die Iterationen werden in zwei Teile unterteilt. Die Sprints dauern jeweils zwei Wochen. Die Aufteilung ist in Tabelle 3.2 zu sehen.

Projektphase	Semesterwoche	Termin
Single-User (3 Sprints)	SW04 - SW09	11.03.19 - 19.04.19
Nutzerevaluation Single-User	SW10	22.04.19 - 26.04.19
Multi-User (2 Sprints)	SW10 - SW13	22.04.19 - 17.05.19
Nutzerevaluation Multi-User	SW14	20.05.19 - 24.05.19

Tabelle 3.2: Sprintplan

3.3 Meilensteine

Für das Projekt wurden zwei Meilensteine definiert welche in Tabelle 3.3 ersichtlich sind.

Meilenstein	Termin	Beschreibung
Single-User Applikation	SW09	**Single-User Prototyp** - Natürliche Interaktion mit dem Modell - Natürliche Kollision zwischen Objekten - Intuitive Handhabung der Applikation
Multi-User Applikation	SW14	**Multi-User Prototyp** - Mehrere User in der selben virtuellen Umgebung - Simultane Interaktion am selben Objekt - Avatar-Repräsentation der anderen Person - Kommunikation zwischen den Benutzern

Tabelle 3.3: Meilensteine

3.4 Anforderungskatalog

Anhand der Aufgabenstellung und den Erkenntnissen aus der Recherche wurden folgende Anforderungen definiert:

Single-User Anforderungen	Priorität
Objekt manipulieren	A
Objekt zerlegen	A
Objekt zusammensetzen	A
Intuitive Interaktion	A
Einschränkungen beim Zusammenbau / Manipulation	B

Tabelle 3.4: Single-User Anforderungen

Multi-User Anforderungen	Priorität
Gleichzeitige Interaktion mit der virtuellen Umgebung	B
Kommunikation zwischen den Benutzern	B
Avatar Repräsentation in der virtuellen Umgebung	B
Gleichzeitige Interaktion am selben Objekt	C

Tabelle 3.5: Multi-User Anforderungen

3.5 Risikoanalyse

Eintrittswahrscheinlichkeit (EW)

id	Beschreibung
1	Unwahrscheinlich
2	Wahrscheinlich
3	Sehr wahrscheinlich

Tabelle 3.6: Eintrittswahrscheinlichkeit

Schadensausmass (SA)

id	Beschreibung
1	Geringer Schaden (1h - 1 Tag Verzögerung)
2	Mittlerer Schaden (Mehrere Tage - Mehrere Wochen Verzögerung)
3	Grosser Schaden (bis zu Abbruch des Projektes)

Tabelle 3.7: Schadensausmass

3.5.1 Mögliche Risiken

id	Risiko	Beschreibung	EW	SA
R1	Netzwerk-Synchronisation Probleme	Die Synchronisierung der Bauteile über das Netzwerk ist zu langsam oder kann in VR nicht umgesetzt werden.	2	2
R2	Limitierter Testbereich	Da nur ein limitierter Testbereich zur Verfügung steht und sich dementsprechend beide Testpersonen im gleichen Tracking-Bereich aufhalten, kann es zu Kollisionen kommen.	3	1
R3	Hardware defekt	Während des Projektes kann ein Hardware defekt an einer der Virtual Reality Brillen auftreten.	1	3

Tabelle 3.8: Risiken vorher

3.5.2 Risikoverminderung: Mögliche Massnahmen

Nachfolgend sind mögliche Massnahmen zur Verminderung der Risiken sowie deren Auswirkungen auf die Eintrittswahrscheinlichkeit (EW) und das Schadensausmass (SA) beschrieben.

id	Risiko	Massnahme	EW	SA
R1	Netzwerk-Synchronisation Probleme	Es gibt diverse Frameworks welche eine Echtzeit Synchronisation erlauben. (zB. „Photon Engine“, 2019) Trotzdem könnte das Problem auch dann noch auftauchen.	1	2
R2	Limitierter Testbereich	Solange die Positionen der beiden Testpersonen auf wenige Zentimeter genau übertragen werden und die Tracking-Bereiche bei beiden exakt die gleichen sind, sollten die Testpersonen nicht mehr miteinander kollidieren. (Solange sie darauf achten, wo sie hingehen)	1	1
R3	Hardware defekt	Da im Projekt mit SteamVR gearbeitet wird, kann jederzeit auf eine andere Virtual Reality Brille zurückgegriffen werden.	1	1

Tabelle 3.9: Risiken nachher

3.6 Verwendete Hardware

Um die Prototypen entwickeln zu können wird die HTC Vive (Abbildung 3.1) verwendet („VIVE™“, 2019). Umgesetzt wird die Programmierung der Applikationen mit der Game-Engine Unity und dem SteamVR Plugin, welches eine einfache Integrierung der Virtual Reality Brillen in Unity erlaubt („Unity“, 2019). SteamVR stellt diverse Prototypen und Beispielapplikationen zur Verfügung um dem Programmierer den Einstig so einfach wie möglich zu machen („SteamVR Plugin“, 2019). Dank dem SteamVR Plugin ist es möglich, ohne grossen Aufwand, die verwendete VR-Brille gegen eine andere auszutauschen.

Abbildung 3.1: HTC Vive

4 Ideen und Konzepte

4.1 Single-User Prototyp

Um dem Benutzer die Interaktion mit den Bauteilen so intuitiv und angenehm wie möglich zu machen, sind die folgenden Features in der Umsetzung geplant:

4.1.1 Highlight

Um dem Benutzer zu zeigen, mit welchem Bauteil er interagieren kann, soll das Bauteil, in welchem sich seine Hand aktuell befindet, markiert werden. Dabei soll die Markierung für den Benutzer gut sichtbar sein, nicht aber die Sicht auf das Bauteil selbst versperren.
Im SteamVR-Asset befindet sich eine Beispiel Szene (Interactions_Example), um die Handhabung mit dem Asset besser kennen zu lernen. In dieser Szene werden Objekte mit einer gelben Umrandung versehen, zu sehen in Abbildung 4.1, sobald sich ein Controller des Benutzers im Objekt befindet.

Abbildung 4.1: SteamVR Highlight

4.1.2 Snapping

Bauteile in der virtuellen Realität präzise zu platzieren kann oft sehr schwierig sein. Um dem entgegenzuwirken sollen Bauteile, welche sich in der Nähe ihres vorgesehenen Ortes befinden, beim loslassen automatisch an ihre richtige Position bewegt werden. Um dem Benutzer mitzuteilen, dass das Bauteil an eine Position “gesnappt“ werden kann, soll die Silhouette des Bauteils am vorgesehen Ort erscheinen. Um die Silhouette erscheinen zu lassen, könnte die gelbe Umrandung verwendet werden, welche im Kapitel 4.1.1 bereits erwähnt wurde.

4.1.3 Kollision

Beim Bewegen der Bauteile sollten diese mit anderen Bauteilen, welche sich in der virtuellen Umgebung befinden, kollidieren. Da alle Bauteile aber nur virtuell verfügbar sind und die Hand in der realen Welt nicht an etwas abprallen kann, muss dies in der virtuellen Realität simuliert werden.
Um dies zu Realisieren müssen beide Bauteile einen Collider besitzen. Sobald ein Kollisions-Event gesendet wird, wird das Bauteil von der Hand des Spielers losgelöst. Die Physik-Engine von Unity berechnet dann welche Auswirkungen die Kollision auf das Bauteil hat.

4.1.4 Feedback beim Zusammenbau / Manipulation der Objekte

Da Bauteile nicht durch andere Bauteile hindurch bewegt werden können, muss dem Benutzer eine Art Feedback gegeben werden, sobald zwei Bauteile kollidieren. Dies soll dem Benutzer auch beim Zusammenbauen der Objekte behilflich sein. In der Tabelle 4.1 sind drei Varianten aufgelistet, welche dieses Problem lösen könnten. Anhand eines Nutzertests soll evaluiert werden, welcher dieser Varianten sich am besten für dieses Problem eignet.

Variante	Pro	Contra
Visuelles Feedback, falls ein Bauteil mit einem anderen kollidiert	• Ist für alle Benutzer sichtbar	• Kann dem Benutzer sehr unnatürlich erscheinen
Einschränkungen der Freiheitsgrade sobald das Bauteil eine gewisse Position erreicht, wie in Abbildung 2.2 zu sehen ist	• Das Zusammenbauen des Bauteils wird dem Benutzer erleichtert	• Kein Feedback falls eine falsche Bewegung durchgeführt wird
Vibration Feedback bei einer falschen Bewegung	• Der Benutzer bemerkt die falsche Bewegung, ohne hinzusehen	• Nur der bearbeitende Benutzer bemerkt die falsche Bewegung

Tabelle 4.1: Varianten für das Feedback beim Zusammenbau

4.2 Multi-User Prototyp

Der Multi-User Prototyp soll auf dem Single-User Prototyp aufbauen und alle Features von diesem implementieren. Die Erkenntnisse aus der ersten Nutzerevaluation sollen ebenfalls in den zweiten Prototypen einfliessen.
Anhand der Erkenntnissen aus den Recherchen, sind die folgenden Features in der Umsetzung geplant:

4.2.1 Avatar-Repräsentation in der virtuellen Umgebung

Anhand der Recherchen, siehe Kapitel 2.2.2, wurden folgende Varianten für eine Avatar-Repräsentation gefunden:

Variante	Pro	Contra
Kopf und Hände der Benutzer werden in der virtuellen Umgebung abgebildet	• Gesten und die Rotation des Kopfes können dargestellt werden	• Den Kopf und die Hände ohne Körper zu sehen kann sehr unnatürlich wirken
Einfacher Avatar, welcher den Oberkörper abbildet	• Funktioniert ohne zusätzliche Detektoren	• Versperrt dem anderen Benutzer möglicherweise die Sicht auf Teile von relevanten Objekten
Ganzkörper Avatar	• Ganzkörper Repräsentation des Gegenübers, erscheint daher natürlicher	• Schwer umzusetzen bezüglich des Trackings, da zusätzliche Detektoren verwendet werden müssen. Falls diese nicht richtig funktionieren erscheint der Avatar dem Gegenüber unnatürlich

Tabelle 4.2: Varianten für die Avatar-Repräsentation

Für diesen Typ Applikation ist es nicht notwendig einen Ganzkörper Avatar zu verwenden. Dieser würde nur weitere Detektoren benötigen und bringt keinen Mehrwert. Die Gesten können auch mit einem einfachen Avatar dem Gegenüber gezeigt werden.

Ein einfacher Avatar bestehend aus Kopf, Oberkörper und den beiden Händen kann ohne zusätzlichen Detektoren umgesetzt werden. Dabei sehen die anderen Benutzer zusätzlich die Orientierung der anderen Benutzern, was für die Zusammenarbeit von Vorteil sein kann. Dementsprechend wird im Projekt ein Avatar mit einem Oberkörper verwendet.

4.2.2 Kommunikation zwischen den Benutzern

Variante	Pro	Contra
Audio-Aufnahmen mit dem Mikrofon und Wiedergabe über die Kopfhörer des Gerätes	• Kann über jegliche Distanzen verwendet werden	• Das Gerät muss ein Mikrofon sowie ein Lautsprecher besitzen
Direkte Kommunikation mit dem Gegenüber	• Benötigt keine technische Implementation	• Beide Benutzer müssen sich im selben Raum befinden
Kommunikation anhand der Körpersprache - Gesten und die Rotation des Kopfes	• Einfach und verständlich	• Eingeschränkte Kommunikation

Tabelle 4.3: Varianten für die Kommunikation zwischen den Benutzern

Da die Applikation auch verteilt über mehrere Standorte funktionieren soll, muss entweder Variante eins oder drei aus der Tabelle 4.3 umgesetzt werden.
Da die HTC Hive ein Mikrophon sowie Kopfhörer integriert hat, kann die erste Variante ohne zusätzliche Hardware umgesetzt werden. Um die Kommunikation zwischen den Benutzern so verständlich wie möglich zu machen, wird zusätzlich zu der ersten Variante noch die dritte Variante umgesetzt, welche dank dem geplanten Avatar, beschrieben in Kapitel 4.2.1, implementiert werden kann.

4.2.3 Gleichzeitige Interaktion am selben Objekt

Falls zwei Benutzer gleichzeitig dasselbe Objekt manipulieren wollen, muss eine Lösung gefunden werden, damit keine Inkonsistenzen zwischen den Benutzern entsteht, falls ein Benutzer das Bauteil an einen anderen Ort bewegt als der andere. Im Bereich der Gleichzeitigen Interaktion am selben Objekt, gibt es sehr viele Varianten, über welche Forschungen und Paper veröffentlicht worden sind (siehe Kapitel 2.2.1). Anhand dieser wurde die nachfolgende Tabelle erstellt:

Variante	Pro	Contra
Separierung der Freiheitsgrade – Ein Benutzer steuert die Position, ein anderer die Rotation des Objektes		• Umständlich für eine Engineering Applikation • Behindert die Zusammenarbeit • Maximal zwei Benutzer
Berechnung des Mittelwerts aller Inputs der Benutzer	• Gleichzeitige Manipulation	• Das Manipulierte Objekt verhält sich nicht natürlich und zerstört so die Immersion
Abstrahierte Objekte in der realen Welt	• Natürliches Verhalten • Natürliches Feedback bei der Manipulation	• Benutzer müssen sich im selben Raum befinden • Objekt und Hände müssen getracked werden
Zauberstab – Nur der Benutzer mit dem Zauberstab kann das Objekt manipulieren	• Natürliches Verhalten, da nur ein Benutzer das Objekt manipuliert	• Nur ein Benutzer kann die Umgebung bearbeiten
«First come, First grab» Der Benutzer welcher eine Interaktion als erstes beginnt hat so lange die Macht über das Objekt bis er dieses wieder loslässt.	• Natürliches Verhalten, da nur ein Benutzer das Objekt manipuliert	• Nur ein Benutzer kann das Objekt bearbeiten • Andere Benutzer müssen sehen, ob sie das Objekt aktuell manipulieren können oder nicht

Tabelle 4.4: Varianten für gleichzeitige Interaktion am selben Objekt

Die erste Variante macht in der umzusetzende Applikation die Zusammenarbeit umständlicher als es sein muss. Da ein Objekt oftmals nicht nur die Position ändern soll sondern auch die Rotation müssten bei den meisten Interaktionen mehrere Benutzer beteiligt sein.
Die Variante bei welcher der Mittelwert aller Inputs genommen wird, wird ebenfalls nicht umgesetzt, da sich die Bewegung des Objektes für alle beteiligten Benutzer unnatürlich anfühlen würde.

Die Variante mit den abstrahierten Objekten in der realen Welt würde den Benutzern das natürlichste Verhalten geben, kann aber nicht umgesetzt werden, da die Applikation auch verteilt funktionieren soll und sich die Benutzer für diese Variante im selben Raum befinden müssten.
Dementsprechend werden die letzten beiden Varianten implementiert und anhand von Nutzertests evaluiert, welche Variante sich für diese Applikation besser eignen würde.

5 Single User Interaktion

5.1 Realisierung

5.1.1 SteamVR-Plugin

Für die Anbindung der VR-Brille und der beiden Controller wurde das Unity Plugin SteamVR verwendet („SteamVR Plugin“, 2019). Das SteamVR Plugin beinhaltet verschiedene vorgefertigte Objekte, auch Prefabs genannt, welche dem Programmierer den Einstieg und die Arbeit mit Virtual Reality in Unity erleichtern. Eines dieser Prefabs namens “Player“ ist dafür da, die VR-Brille mitsamt den beiden Controllern in Unity abzubilden, so dass diese ohne weiteres verwendet werden können. Das verwendete “Player“-Prefab beinhaltet die folgenden Punkte, um welche sich der Programmiere nicht mehr kümmern muss:

- Regelt die Inputs der Controller und generiert daraus Events, auf welche der Programmierer sich abonnieren kann
- Handhabt die Kamera
- Zeichnet die Grenzen der virtuellen Umgebung, in welcher der Spieler sich sicher bewegen kann, ohne in der realen Welt in ein anderes Objekt zu laufen

5.1.2 Highlight

Dank der Beispiel Szene im SteamVR Asset, beschrieben im Kapitel 4.1.1, war sehr schnell klar, wie das Highlight der Objekte im Asset selbst gemacht wird. Um die Silhouette des Objektes gelb umrahmen zu können, wird das Material aller MeshRenderer (in Unity zuständig für die visuelle Darstellung eines GameObjects) des Objektes ersetzt, mit einem Material, auf welchem sich ein spezifischer Shader befindet. Dieser Shader zeichnet nur die Aussenlinie des Meshes und macht alles innerhalb dieser Linie transparent. Um die originale Farbe des Objektes nicht zu verlieren, wird das Objekt erst kopiert und anschliessend das Material auf dem kopierten Objekt geändert. Die Kopie des Objektes muss erstellt werden, da das originale Material beibehalten werden soll. Würde das Material direkt auf dem Objekt angepasst, würde nur noch die gelbe Aussenlinie zu sehen sein. Diese Lösung ist sehr ineffizient und könnte besser gelöst werden. Mehr dazu im Kapitel 8.

Das Highlight in der Single-User Applikation wurde genau wie im vorherigen Abschnitt beschrieben implementiert. An den beiden Controllern des Benutzers wurde jeweils ein Trigger hinzugefügt. Ein Trigger ist ein Collider, welcher aber keine Kollision verursacht sondern lediglich ein Event feuert, sobald er mit einem anderen Collider oder Trigger in Berührung kommt.

Dieses Event wird von der Applikation abgefangen und löst das Highlight des kollidierten Objektes aus. Dies ist in Abbildung 5.1 dargestellt.

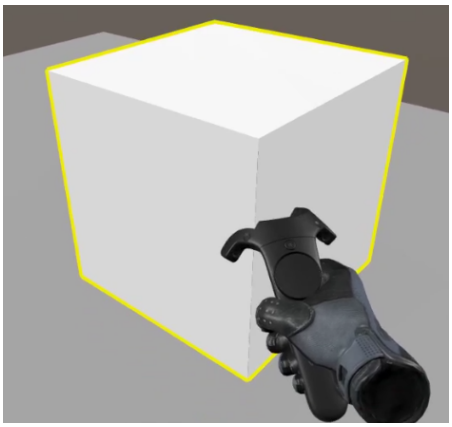

Abbildung 5.1: Highlight in der Single User Applikation

5.1.3 Snapping

Zwischen den Bauteilen, bei welchen das Snapping stattfinden soll, wurden Trigger angebracht, welche genau aufeinander passen (Abbildung 5.2). Sobald sich diese Trigger berühren wird ein Event ausgelöst, welches das Bauteil in der Hand des Benutzers als Silhouette am vorgesehenen Ort zeichnet (Abbildung 5.3).
Dafür wird eine Kopie des Bauteils mitsamt dem Trigger erstellt, so positioniert, dass die beiden Trigger übereinstimmen und anschliessend das Material durch den Highlight-Shader ersetzt (beschrieben in Kapitel 5.1.2).

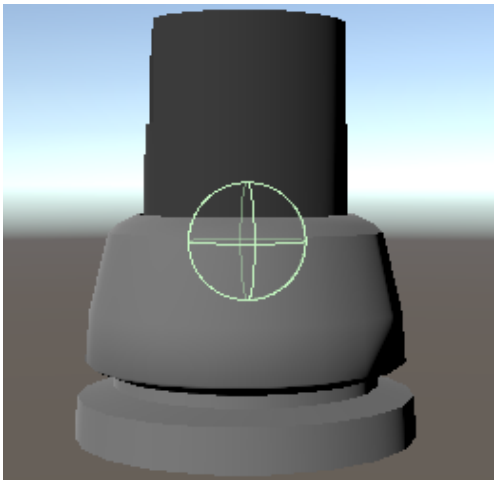

Abbildung 5.2: Trigger zwischen Bauteilen

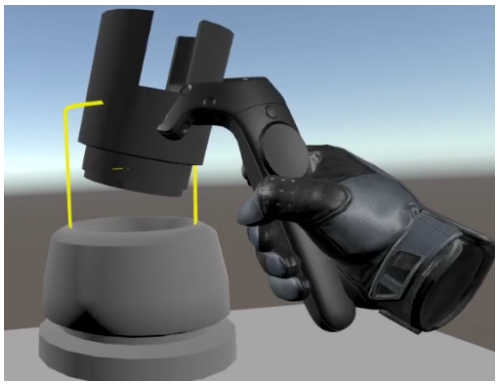

Abbildung 5.3: Highlight des Snappings

5.1.4 Kollision

Falls das Bauteil, wie in Kapitel 4.1.3 beschrieben, nur von der Hand losgelöst wird, kann es nicht ohne weiteres wieder an die Hand des Benutzers gemacht werden beim verlassen der Kollision. Auch andere Ansätze funktionierten nicht direkt so wie gedacht.

Um die Kollision der Bauteile so natürlich wie möglich zu machen, mussten den Bauteilen ein Rigidbody angehängt werden. Dieser ist für die Physik-Berechnungen zuständig und kann konstante Beschleunigungen oder eine konstante Geschwindigkeit auf das Bauteil geben. Sobald nun zwei Bauteile kollidieren, wird dem Rigidbody auf dem Bauteil, welches sich aktuell an der Hand des Benutzers befindet, mitgeteilt, dass es nicht mehr Kinematisch sein soll. Damit wird bewirkt, dass dieses Bauteil sich mithilfe der Physik bewegen kann (Die Gravitation ist deaktiviert). Da das Bauteil so aber langsam in eine Richtung gleiten würde, muss ihm eine konstante Geschwindigkeit zum Controller hin gegeben werden. Die Geschwindigkeit errechnet sich durch die Distanz des Bauteils zum Controller des Benutzers. Diese Distanz ist ein dreidimensionaler Vektor, welcher in eine bestimmte Richtung zeigt. Das heisst, dass der Vektor grösser wird umso grösser die Distanz zwischen dem Bauteil und dem Controller wird. Sobald das Bauteil mit keinem anderen Bauteil mehr kollidiert, wird der Rigidbody wieder Kinematisch und die Geschwindigkeit auf null gesetzt. Dieses Verhalten ist in Abbildung 5.4 dargestellt.

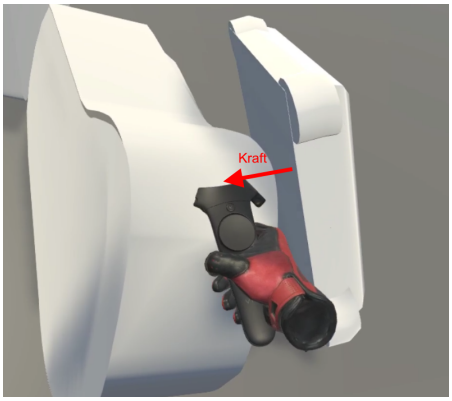

Abbildung 5.4: Kollision zwischen zwei Objekten und der konstanten Kraft

5.1.5 Baugruppen

Während der Arbeit wurde bemerkt, dass es sehr umständlich sein kann, mehrere Bauteile miteinander zu transportieren. Alle Bauteile mussten an einem Ort auseinandergenommen werden nur um diese an einem anderen Ort wieder zusammenzubauen. Um dies zu erleichtern wurde jedem Bauteil eine Liste von Kindern angehängt, welche besagt, welche Bauteile mit dem aktuellen Bauteil mitbewegt werden sollen. Dies resultiert darin, dass es bei jeder Maschine ein Bauteil gibt, an welchem alle anderen Bauteile direkt oder indirekt über andere Bauteile befestigt sind.

Um dem Benutzer mitteilen zu können, in welcher Situation er welche Bauteile zusammen bewegen kann, wurde das Highlight, beschrieben in Kapitel 5.1.2, wie folgt angepasst. Für jedes Kind des Bauteils wurde überprüft, ob es zum aktuellen Zeitpunkt verbunden ist und falls dies der Fall ist, wird auch von diesem Bauteil eine Kopie erstellt und der Shader für das Highlight hinzugefügt. Somit werden alle Bauteile, welche miteinander bewegt werden können, wie in Abbildung 5.5 ersichtlich, mit einem Highlight versehen.

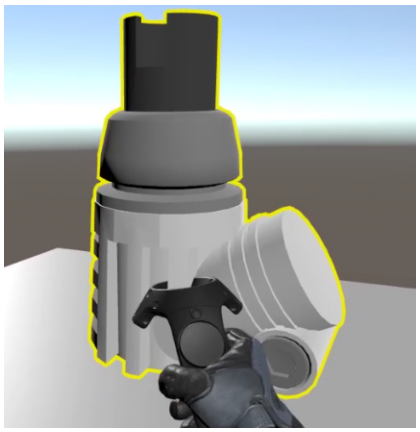

Abbildung 5.5: Highlight einer Baugruppe

5.1.6 Feedback beim Zusammenbau / Manipulation der Objekte

Da drei verschiedene Varianten, beschrieben in Kapitel 4.1.4, implementiert wurden, wurde für jede Variante ein eigener Prototyp mit einer eigenen Maschine erstellt.

Die Variante, bei welcher dem Benutzer ein visuelles Feedback gegeben wird, wurde wie folgt implementiert. Bei einer Kollision zwischen zwei Bauteilen, wird das Bauteil in der Hand des Benutzers, wie in Abbildung 5.6 zu sehen, rot eingefärbt. Sobald die Kollision endet, wird die Farbe des Bauteils wiederhergestellt.

Abbildung 5.6: Visuelles Feedback bei einer Kollision

Für die Umsetzung der zweiten Variante, bei welcher die Freiheitsgrade des Benutzers eingeschränkt werden, wurde die Ausrichtung des Triggers verwendet, welcher für das Snapping gebraucht wird. Sobald sich das Bauteil im Bereich des automatischen Snappings befindet, wird das Bauteil in der Hand des Benutzers anhand des Triggers ausgerichtet. Zur gleichen Zeit wird zwischen den beiden Bauteilen ein Joint platziert. Ein Joint ist eine Verbindung zwischen zwei Bauteilen und hat je nach Art des Joints unterschiedliche Eigenschaften. Im Falle dieser Applikation wird ein Joint verwendet, bei welchem zwei Achsen sowie die Rotation in allen Achsen fixiert werden. Somit lässt sich das Bauteil, wie in Abbildung 5.7 zu sehen, nur noch auf einer Achse bewegen.

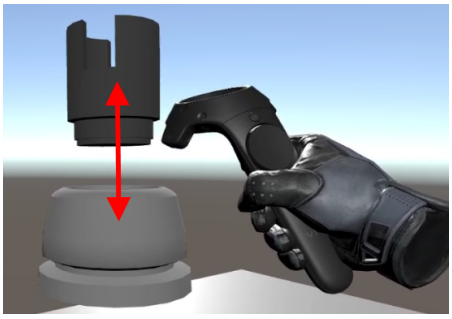

Abbildung 5.7: Einzige bewegbare Achse durch die Einschränkung der Freiheitsgrade

Die dritte Variante, mit dem Vibrations Feedback, konnte mithilfe des SteamVR Assets implementiert werden. Um auf die Vibration des Controllers zugreifen zu können, muss lediglich bekannt sein, auf welchem Controller die Vibration ausgeführt werden soll. Die Übergabeparameter der Funktion des SteamVR Assets erlauben dann die Vibration wie folgt anzupassen:

- Startzeit der Vibration in Sekunden ab dem aktuellen Zeitpunkt
- Dauer der Vibration
- Frequenz der Vibration in Hertz
- Stärke der Vibration

5.2 Evaluation

Nach Abschluss der Realisierungsphase des Single-User Prototypen wurde eine Nutzerevaluation durchgeführt um den Prototypen zu evaluieren und um entscheiden zu können, welcher der drei beschrieben Varianten in Kapitel 5.1.6 sich für diese Applikation am besten eignet.

5.2.1 Aufgabenstellung

Während der Nutzerstudie mussten die Teilnehmer zwei Aufgaben erledigen. Bei beiden Aufgaben wurden ihnen drei Maschinen gezeigt, mit jeweils einer anderen Art des Feedbacks beim Zusammenbau / Manipulation der Objekte.

1. In der ersten Aufgabe ging es darum, eine Maschine zusammenzubauen, welche sich in einer explosionsartigen Darstellung vor dem Nutzer befand.

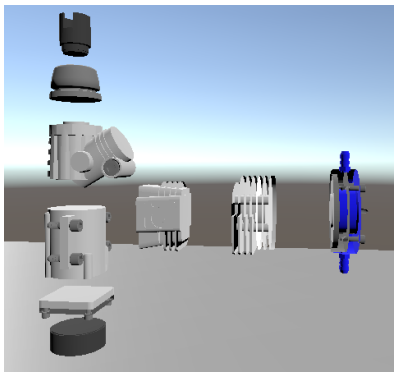

Abbildung 5.8: Erste Aufgabe der ersten Nutzerevaluation

2. Die zweite Aufgabe bestand darin, ein rot eingefärbtes Bauteil aus der Maschine auszubauen, dieses in einen gelben Bereich zu halten, um es automatisch reparieren zu lassen, und anschliessend dieses nun grün gefärbte Bauteil wieder in der Maschine einzubauen.

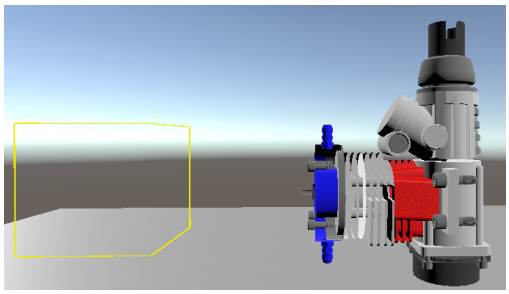

Abbildung 5.9: Zweite Aufgabe der ersten Nutzerevaluation

5.2.2 Resultat

Insgesamt haben sechs Teilnehmer an der Nutzerevaluation teilgenommen. Alle Teilnehmer hatten bereits Erfahrungen mit Virtual Reality.

Den Teilnehmern wurden nach dem Abschluss der Aufgaben die folgenden fünf Fragen gestellt:

1. Frage: Wie empfanden Sie die Steuerung mit dem Kontroller?
   Auswertung: Die Teilnehmer fanden die Steuerung mit dem Controller sehr angenehm und intuitiv. Nur eine Person bemängelte, dass die verwendeten Buttons beschriftet hätten sein müssen um die Interaktion noch klarer zu machen.
2. Frage: Wie verständlich ist die Manipulation der Objekte?
   Auswertung: “Man wusste immer mit welchem Objekt man interagiert“, meinte einer der Testpersonen. Dies fasst ziemlich gut die allgemeine Rückmeldung zur Manipulation der Objekte zusammen.
3. Frage: Was hat Ihnen an der Applikation besonders gefallen?
   Auswertung: Das Highlight sowie das Snapping hat sehr vielen Teilnehmern besonders gefallen. Zwei Teilnehmer haben geschrieben, dass ihnen das einfach gehaltene Interface sowie die einfache Handhabung der Applikation gut gefallen hat.
4. Frage: Wie natürlich empfanden Sie die Interaktion mit den Objekten?
   Auswertung: Die Teilnehmer fanden die Interaktion mit den Objekten sehr natürlich. Dank den Features wie das Snapping, die Baugruppen und das Highlight fiel es den Teilnehmern sehr leicht mit den Objekten zu interagieren und dementsprechend wirkte die Interaktion natürlicher.
5. Frage: Welche Variante der Rückmeldung einer nicht erlaubten Bewegung hat Ihnen am besten gefallen und wieso?
   Auswertung: Fünf von den sechs Teilnehmern fanden die Variante mit dem Vibrations-Feedback am besten geeignet für diese Applikation. Es sei die natürlichste Variante und der Benutzer bemerke eine unerlaubte Bewegung auch dann, falls er das Bauteil gerade nicht im Blickfeld hat.
   Die Variante mit dem Einfärben der Bauteile passe nicht. Möglicherweise wurde die Farbe falsch gewählt und deswegen erscheine es unnatürlich, meinten zwei Teilnehmer.
   Die dritte Variante, bei welcher die Freiheitsgrade eingeschränkt wurden, könnten sich zwei Teilnehmer bei gewissen Teilen der Applikation vorstellen. “Beim Anbringen einer Schraube könnte ich mir diese Variante mit den Freiheitsgraden vorstellen“, meinte einer dieser beiden Teilnehmern. Der andere schrieb, dass er sich eine Kombination der beiden Varianten vorstellen könnte: “Kombination von Variante 1 und 2 würde gut passen. Weiter weg zum fahren, sobald nah genug dann Snapping und je nachdem Vibration.“

5.2.3 Schlussfolgerung

Die Teilnehmer der Nutzerevaluation waren sehr zufrieden mit dem Single-User Prototyp. Bemängelt wurde einzig die Implementation der Variante, bei welcher die Freiheitsgrade der Benutzer eingeschränkt wird. Diese Variante hatte öfters Probleme, da sich die Objekte anders verhalten hatten als sie sollten oder diese gar in der Szene umhergesprungen sind. Dementsprechend ist die Bewertung dieser Variante insgesamt schlechter ausgefallen. Bei der Variante, bei welcher die Bauteile eingefärbt werden wurde vermutlich die Farbe falsch gewählt. Das rot symbolisierte eher, dass etwas falsche gemacht wurde, anstatt der Kollision mit anderen Bauteilen. Mit einer anderen Farbwahl, oder einer transparenteren Farbe, hätte die Nutzerevaluation eventuell anders ausfallen können.

Die meisten Teilnehmer fanden die Vibrations-Feedback Variante am besten geeignet für diese Applikation. Dementsprechend wird diese Variante für den zweiten Prototypen verwendet. Falls es die Zeit zulässt, könnte sich überlegt werden, gewisse Bauteile, wie zum Beispiel Schrauben, mit der Kombination von Vibration und Einschränkung der Freiheitsgrade zu implementieren.

5.3 Systemarchitektur

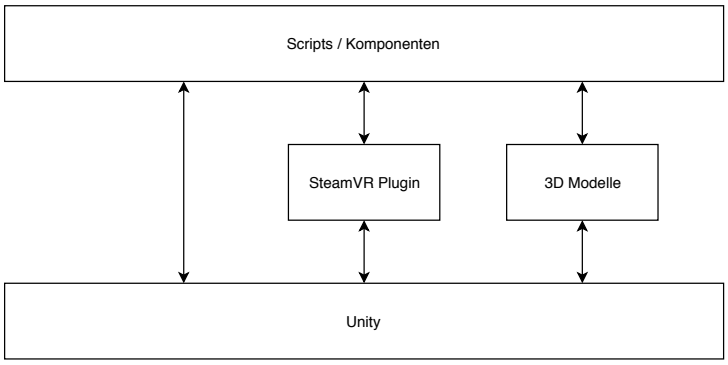

Abbildung 5.10: Systemarchitektur des Single-User Prototypen

Die Systemarchitektur des Single-User Prototypen ist in Abbildung 5.10 dargestellt. Als Basis dient die Game-Engine Unity und alle zur Verfügung gestellten Features. Das für die Virtual Reality Anbindung verwendete SteamVR Plugin kommuniziert mit Unity und stellt eine Schnittstelle für eigene Skripte dar. Die 3D Modelle werden von Unity importiert und in ein nutzbares Format innerhalb von Unity konvertiert.
Die erstellten Skripte und verwendeten Komponenten, beschrieben in Kapitel 5.4, befinden sich auf den konvertieren 3D Modellen oder verwendeten SteamVR Komponenten. Die Skripte verwenden Funktionen des SteamVR-Plugins und von Unity selbst.

5.4 Klassendiagramm

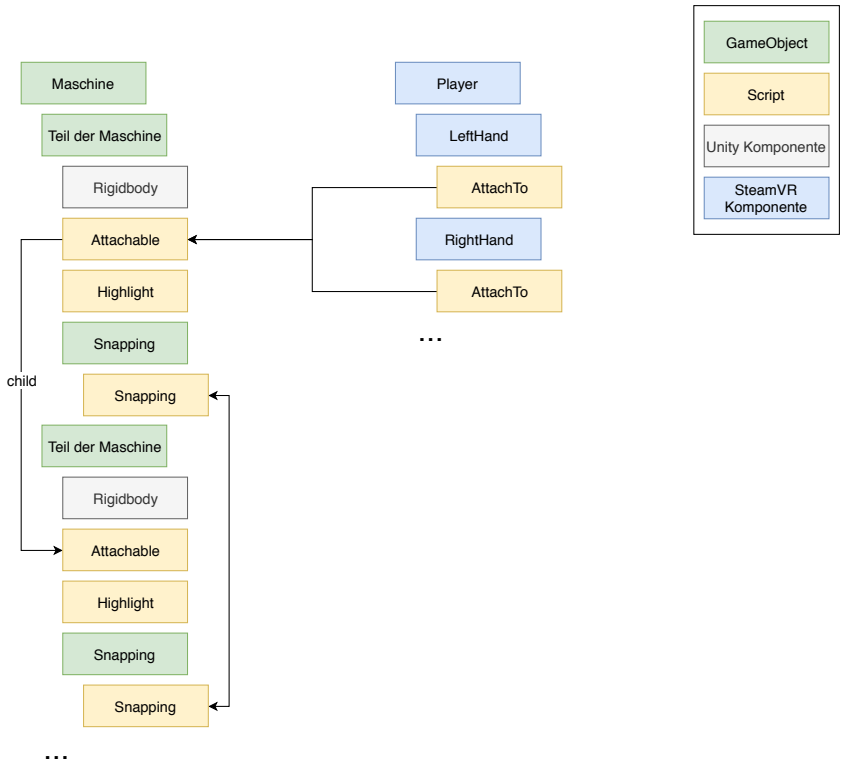

Abbildung 5.11: Klassendiagramm des Single-User Prototypen

Der “Player“ ist ein Prefab, welches vom SteamVR-Plugin zur Verfügung gestellt wird, beschrieben in Kapitel 5.1.1. An diesem Prefab wurde lediglich der linken und rechten Hand ein Skript “AttachTo“ hinzugefügt. Dieses Skript fängt die Inputs des Benutzers ab und teilt beim Greifen eines Objektes dem “Attachable“ Skript mit, dass es nun gegriffen wurde.

Die Maschine wurde in verschieden Teile unterteilt, jeder Teil der Maschine kann bewegt und auseinandergenommen werden. Die Funktionalität der Skripts wurde in folgenden Kapiteln beschrieben:

- Highlight: Zuständig für das Highlight der Bauteile, falls sich eine Hand des Benutzers in einem Bauteil befindet sowie das Highlight des Snappings. Beides ist beschrieben im Kapitel 5.1.2.
- Snapping: Zuständig für das Snapping der Bauteile, beschrieben in Kapitel 5.1.3.
- Attachable: Zuständig für das Kollidieren der Bauteile, beschrieben in Kapitel 5.1.4, für die Baugruppen, beschrieben in Kapitel 5.1.5 und das Feedback beim Zusammenbau, beschrieben in Kapitel 5.1.6.

6 Multi User Interaktion

6.1 Realisierung

Mit dem Projekt und den Erkenntnissen des Single-User Prototypen begann die Erstellung eines Multi-User tauglichen Prototypen.

6.1.1 Multi User

Um die Applikation Multi-User tauglich zu machen, wurde die Photon-Engine verwendet („Photon Engine“, 2019). Entwickelt wurde diese von der Firma Exit Games. Diese Firma stellt ein Unity-Asset zur Verfügung, mit verschiedenen Beispielen wie das Asset zusammen mit der Photon-Engine einzusetzen ist.

Auf der Webseite von Photon kann nach einem erfolgreichen Login eine neue Applikation erstellt werden. Für den Multi-User Prototyp wurde eine Realtime-Applikation erstellt mit bis zu 20 gleichzeitigen Benutzern. Soll eine Applikation mit mehr als 20 gleichzeitigen Benutzern erstellt werden, wird dies kostenpflichtig. Nach dem erstellen kann die erhaltene App ID, zu sehen in Abbildung 6.1, in Unity verwendet werden.

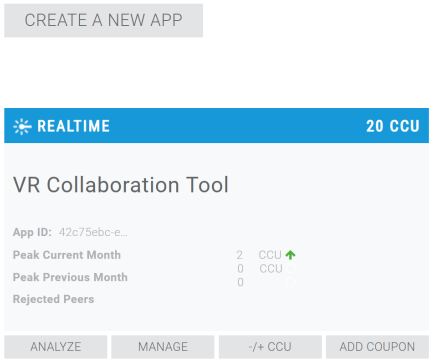

Abbildung 6.1: Photon Dashboard

Mit der App ID kann Unity sich nun auf die Applikation verbinden. Die erste Instanz, welche sich verbindet wird automatisch als Master deklariert. Um nun mit anderen Instanzen kommunizieren zu können, müssen diese sich mit einen virtuellen Raum verbinden. Sobald sich beide Instanzen im gleichen Raum befinden, kann die Kommunikation beginnen.

Jeglichen Objekten, welche nun untereinander synchronisiert werden sollen, muss die Komponente Photon View hinzugefügt werden. Diese besitzt die folgenden Parameter:

- Owner: Der aktuelle Besitzer der Objekts
- View ID: Eindeutige Identität des Objektes
- Observed Components: Welche Komponenten des Objektes mit den anderen Instanzen synchronisiert werden sollen

Für die Synchronisation vieler standardmässigen Unity-Komponenten stellt die Photon-Engine schon Komponenten zur Verfügung. Eine dieser Komponenten, die Photon Transform View, wurde im Prototyp verwendet, um die Position und die Rotation des Objektes zu synchronisieren. Somit werden diese beiden Eigenschaften mit allen Instanzen synchronisiert, welche nicht Besitzer des Objektes sind. Es ist aber nicht möglich als Nicht-Besitzer ein Objekt zu bewegen.
Die Master-Instanz wird bei der Initialisierung Besitzer von allen Objekten, welche eine Photon View besitzen. Um als Nicht-Master-Instanz nun ein Objekt bewegen zu können, stellt die Photon-Engine die Funktion *RequestOwnership()* zur Verfügung. Diese Funktion macht die Instanz zum neuen Besitzer der Photon View, welche die Funktion aufgerufen hat. Somit kann nun die Nicht-Master-Instanz das Bauteil bewegen und alle anderen Instanzen sehen diese Bewegung.

Da die Bauteile im Prototyp einen Rigidbody angehängt haben, muss auch dieser synchronisiert werden. Die Photon-Engine stellt aber für die Synchronisation eines Rigidbody keine Komponenten zur Verfügung. Sie stellt aber die Methode *OnPhotonSerializeView(PhotonStream stream, PhotonMessageInfo info)* zur Verfügung, mit welcher eigene Attribute zwischen den Instanzen synchronisiert werden können. Die Methode wird standardmässig 10 mal in der Sekunde aufgerufen, kann aber in den Einstellungen angepasst werden. Nachfolgend ein Ausschnitt wie der Aufbau dieser Methode aussehen sollte.

```
Input: PhotonStream, PhotonMessageInfo
if Falls die aktuelle Instanz auf den Stream schreiben kann then
    Schreibe alle Informationen des Rigidbody auf den Stream;
    Schreibe weitere Attribute auf den Stream;
else
    Lese die Informationen aus dem Stream und speichere diese in den Rigidbody;
    Lese die weiteren Attribute aus dem Stream;
end
```

Eine Instanz kann nur dann auf den Stream schreiben, wenn sie in besitzt der Photon View ist. Die Komponente Photon Transform View funktioniert sehr ähnlich wie der beschriebene Pseudocode.

6.1.2 Avatar-Repräsentation in der virtuellen Umgebung

Wie in Kapitel 4.2.1 beschrieben, wurde für diesen Prototyp ein einfacher Avatar bestehend aus Kopf, Oberkörper und den beiden Händen umgesetzt. Dieser ist in Abbildung 6.2 zu sehen mit Controllern als Hände. Die Position und Rotation des Oberkörpers werden anhand der Position sowie der Rotation des Kopfes berechnet.

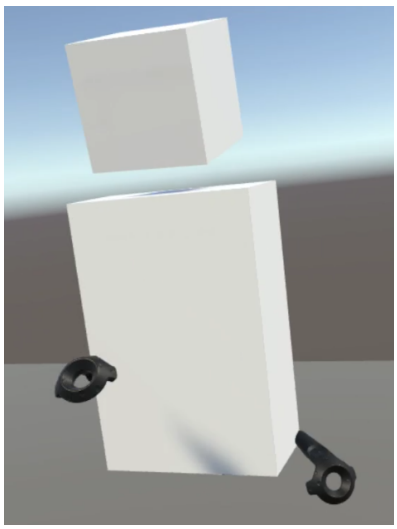

Abbildung 6.2: Avatar leicht nach links blickend

Der Benutzer selbst sieht von seinem eigenen Avatar nur die Hände mit den Controllern. Der Oberkörper sowie der Kopf sind für ihn unsichtbar, da sie sonst in der Quere des Sichtfelds sein würden. Um den Avatar mit den anderen Instanzen zu synchronisieren. wurde ebenfalls, wie in Kapitel 6.1.1 beschrieben, auf jede Komponente des Avatars eine Photon View hinzugefügt. Somit sehen sich die Benutzer untereinander und können mithilfe von Gesten miteinander Kommunizieren.

6.1.3 Kommunikation zwischen den Benutzern

Für die Kommunikation zwischen den Benutzern wurde ebenfalls auf die Photon-Engine zurückgegriffen. Im Photon Dashboard, zu sehen in Abbildung 6.1, wurde eine Voice Applikation erstellt und die erhaltene App ID konnte dann in Unity, zusätzlich zu der App ID für die Realtime-Applikation, eingeben werden.

Um die Voice Applikation nutzen zu können, muss einem leeren GameObject in der Szene eine “Photon Voice Network“ Komponente sowie ein “Recorder“ hinzugefügt werden. Diese Komponenten sind ebenfalls aus einem Plugin der Firma Exit Games namens Photon Voice („Photon Voice 2“, 2019). So wird der Sound über das Mikrophon des Benutzers aufgenommen. Für die Wiedergabe wurde dem Avatar zusätzlich eine “Speaker“-Komponente, welche ebenfalls aus dem Photon Voice Plugin stammt, hinzugefügt. Somit können die Benutzer miteinander über grosse Distanzen kommunizieren.

6.1.4 Gleichzeitige Interaktion am selben Objekt

Durch die Erkenntnisse während der Arbeit mit der Photon-Engine wurde klar, dass eine simultane Interaktion am selben Objekt nicht möglich ist, da immer nur ein Benutzer Besitzer eines Bauteils sein kann. Da nur der Besitzer das Bauteil verschieben kann, müssten andere Technologien verwendet werden. Im Kapitel 4.2.3 wurde entschieden, die beiden Varianten «First come, First grab» und Zauberstab zu implementieren. Beide Varianten verlangen keine simultane Interaktion am selben Objekt und konnten daher umgesetzt werden.

Die Variante «First come, First grab» funktioniert wie folgt. Alle Benutzer können Bauteile greifen und diese verschieben, solange kein anderer das Bauteil aktuell am manipulieren ist. Der genaue Ablauf ist im nachfolgenden Pseudocode veranschaulicht.

```
Der Benutzer versucht das Bauteil zu greifen;
if Ist das Bauteil noch greifbar? then
    RequestOwnerShip() aufrufen um Besitzer des Bauteils zu werden;
    Das Bauteil für anderen Benutzer sperren;
    Das Bauteil an die Hand des Benutzers anhängen;
else
    Dem Benutzer mitteilen, dass das Bauteil aktuell nicht greifbar ist;
end
```

Für die zweite Variante wurde dem Prototypen ein Zauberstab hinzugefügt, welcher ebenfalls greifbar und mit einer Photon View sowie einer Photon Transform View ausgestattet ist. Somit wird die Position wie auch die Rotation zwischen allen Benutzern synchronisiert. Nur derjenige Benutzer, welcher aktuell in Besitz des Zauberstabs ist, kann Bauteile manipulieren. Um den Zauberstab nicht immer in der Hand halten zu müssen, wird dieser beim greifen rechts bzw. links neben der Hand platziert und bewegt sich mit dieser mit, siehe Abbildung 6.3. Möchte ein anderer Benutzer die virtuelle Umgebung manipulieren muss er den Zauberstab vom aktuellen Benutzer übernehmen, indem er den Zauberstab an der Hand des anderen Benutzers greift.

Abbildung 6.3: Zauberstab angehängt an der rechten Hand

6.2 Evaluation

Nach Fertigstellung des Multi-User Prototypen wurde ebenfalls, wie schon nach dem ersten Prototypen, eine Nutzerstudie durchgeführt. Anhand dieser Nutzerstudie sollte evaluiert werden, welche der beschriebenen Varianten im Kapitel 6.1.4 sich besser für die Anwendung eignet.

6.2.1 Aufgabenstellung

Für diesen Nutzertest wurden immer zwei Teilnehmer gleichzeitig benötigt. Falls einer von beiden noch keine Erfahrungen mit Virtual Reality hatte, oder sein Partner noch nicht Verfügbar war, wurde ihnen das Bogenschiessen im Spiel “The Lab“ gezeigt („The Lab“, 2019). Hier ging es darum, dass sich die Teilnehmer, vor dem Benutzen der Multi-User Applikation mit Virtual Reality vertraut machen konnten.

Anschliessend war es die Aufgabe zusammen vier verschiedene Motoren zusammenzubauen. Zweimal mit der Variante «First come, First grab» und zweimal mit der Variante mit dem Zauberstab.
Dafür hat immer einer der beiden Teilnehmer gesehen an welche Position das Bauteil muss (Abbildung 6.4) und der andere hat gesehen, welches Bauteil an diese Stelle muss (Abbildung 6.5). Dies wechselte immer zwischen den Teilnehmer. So wurde die Zusammenarbeit forciert, dass nicht einer alles alleine machen konnte.

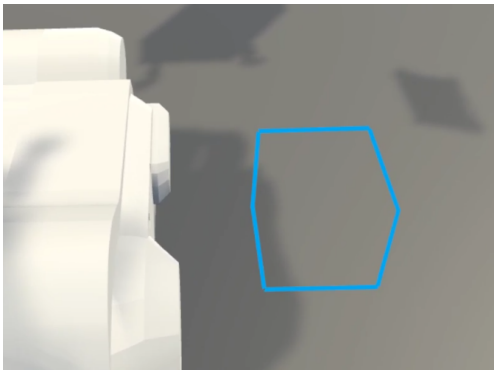

Abbildung 6.4: Wohin muss das Bauteil?

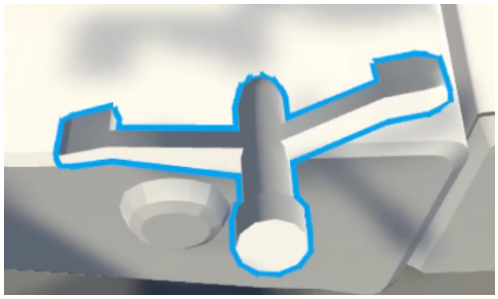

Abbildung 6.5: Welches Bauteil ist es?

6.2.2 Resultat

An der zweiten Nutzerstudie haben 13 Nutzer teilgenommen. Alle Teilnehmer bekamen für die Teilnahme an der Nutzerstudie einen 10.- CHF Gutschein für Digitec Galaxus. Einige Teilnehmer hatten noch keine Erfahrungen mit Virtual Reality.

Den Teilnehmern wurden nach dem Abschluss der Aufgaben die folgenden vier Fragen gestellt:

1. Frage: Wie gut funktionierte die Interaktion mit der anderen Person?
   Auswertung: Alle Teilnehmer fanden die Interaktion mit der anderen Person sehr angenehm und intuitiv. “Anderer Avatar zu sehen ist sehr hilfreich“, meinte ein Teilnehmer und viele andere schrieben, dass der Avatar ein sehr wichtiger Punkt für die Interaktion mit anderen war. Insbesondere, da die Teilnehmer untereinander mithilfe von Gesten kommunizieren konnten, zusätzlich zu der Stimme.
   Eine Person bemängelte, dass der Avatar den anderen Bauteilen zu ähnlich sah, da beide eine weisse Farbe hatten.

2. Frage: Sehen Sie einen Mehrwert in der Applikation gegenüber traditionellen Methoden über ein Bauteil / eine Maschine zu diskutieren?
   Auswertung: Alle Teilnehmer sahen einen Mehrwert in der Applikation gegenüber traditionellen Methoden. Viele fanden, dass es so einfacher sei sich etwas vorstellen zu können, da der Benutzer um das Objekt rumlaufen oder es gar bewegen kann. Ein Benutzer meinte, dass die Applikation für Schulungen bei heikleren Objekten verwendet werden könne: “Für Schulungen bei heikleren Objekten“
   “Aber solange teure Hardware benötigt wird, eher unrealistisch“, schrieb ein Teilnehmer, da für die meisten VR-Brillen heutzutage noch ein teurer Computer benötigt wird.

3. Frage: Welche der beiden Varianten fanden Sie hilfreicher, um den Task auszuführen und wieso?
   Auswertung: “Erste Variante war besser, da man besser zusammenarbeiten konnte“, oder ähnliche Aussagen wurden von allen Teilnehmern gemacht. Es sei, bei dieser spezifischen Aufgabe besser gewesen, wenn beide Teilnehmer die gleichzeitig mit den Bauteilen interagieren konnten.
   Viele Teilnehmer können sich aber vorstellen, dass die Variante mit dem Zauberstab vor allem in Lernumgebungen oder Präsentation mit mehreren Personen hilfreich sein könnte. So interagieren nicht ungewollt Benutzer mit Objekten, falls dies nicht erwünscht ist. Die Aussage eines Teilnehmers fasst dies gut zusammen: “Zweite Variante eher beim Vorzeigen, oder Fachpersonal bei welchem Input gegeben werden können. (zur Veranschaulichung)“

4. Frage: Wenn Sie etwas an der Applikation ändern könnten, was wäre es?
   Auswertung: Viele Teilnehmer bemängelten den Zusammenbau, da der Teilnehmer mit dem blauen Würfel, zu sehen in Abbildung 6.4, die Rotation des Bauteiles nicht sehen konnte. Dementsprechend kam es da zu grossen Verwirrungen wie das nächste Bauteil nun angemacht werden musste. Eine gute Rückmeldung bezüglich des Problems kam von einem Teilnehmer: “Tisch auf welchem die Objekte zusammengebaut werden können. Das erste Teil sollte fix sein.“
   Ein Teilnehmer fand, dass die Bauteile leicht verschiebbar sein sollten, ohne diese packen zu müssen: “Teile sollten leicht stoss bar sein.“

6.2.3 Schlussfolgerung

Die Teilnehmer der zweiten Nutzerevaluation waren mit dem Multi-User Prototyp sehr zufrieden. Leider kam es bei den gestellten Aufgaben öfters zu Problemen, da zum Beispiel richtig zusammengesetzte Bauteile als falsch zusammengebaut erkannt und der nächste Schritt nicht angezeigt wurde. Aber obwohl diese Probleme aufgetreten sind, haben die Teilnehmer die Maschine mithilfe von “Trial and Error“ zusammenbauen können und hatten bemerkbar Freude dabei.

Anhand der Antworten der Teilnehmer haben beide Varianten Anwendungsgebiete. Die Variante “First come, First grab“ kann für das gemeinsame arbeiten mit Objekten in der virtuellen Realität verwendet werden. Sobald es in den Bereich der Schulung geht oder ein Benutzer anderen was zeigen will, ohne dass diese mit den Objekten interagieren, würde sich die zweite Variante mit dem Zauberstab empfehlen.

Viele Teilnehmer können sich vorstellen, dass diese Methode über ein Bauteil zu diskutieren, in Zukunft in der Wirtschaft eingesetzt wird. Dazu muss die Hardware aber noch einige Fortschritte machen und unabhängig von einem Computer nutzbar sein.

6.3 Systemarchitektur

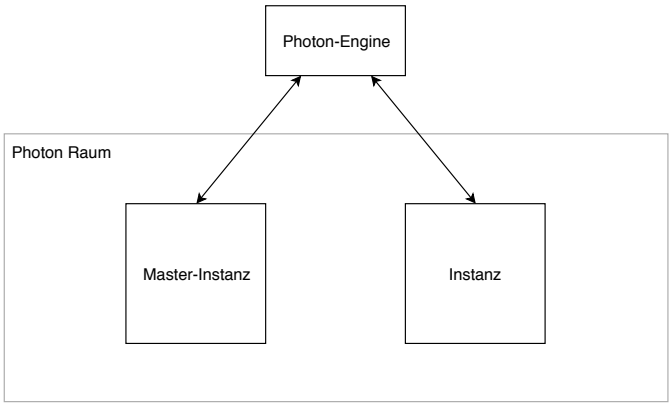

Abbildung 6.6: Systemarchitektur des Multi-User Prototypen

Die beiden Instanzen sind grundsätzlich genau gleich aufgebaut wie in der Single-User Applikation, beschrieben in Kapitel 5.3. Um mit der Photon-Engine kommunizieren zu können musste zusätzlich noch das Photon-Engine Plugin sowie das Photon-Voice Plugin hinzugefügt werden. Die erste Instanz, welche sich mit der Photon-Engine verbindet, wird automatisch zur Master-Instanz. Nach dem Verbinden mit der Photon-Engine wird die Instanz mit einem Photon-Raum verbunden. Sobald sich nun zwei Instanzen im selben Raum befinden, werden alle Objekte mit einer Photon View zwischen ihnen synchronisiert, beschrieben in Kapitel 6.1.1.

6.4 Klassendiagramm

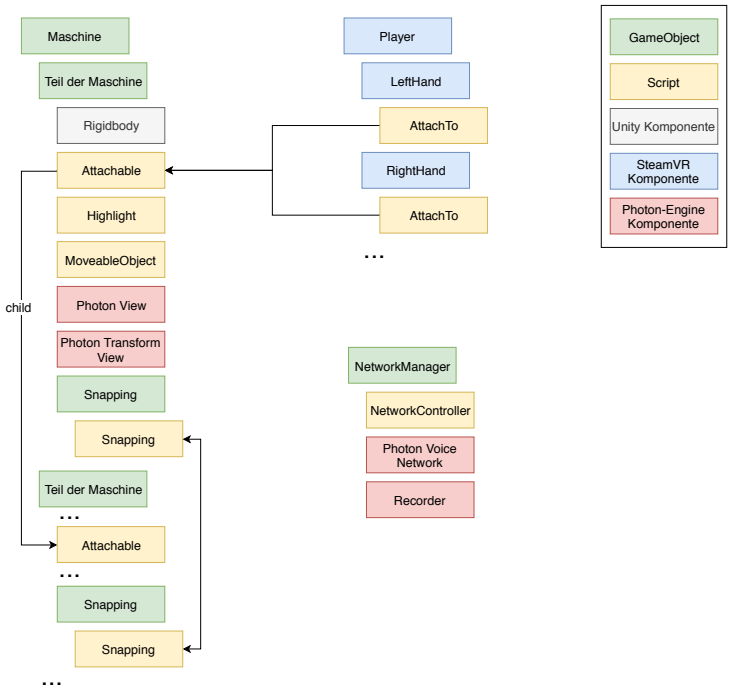

Abbildung 6.7: Klassendiagramm des Multi-User Prototypen

Viele Skripte und Komponenten sind die gleichen wie beim Single-User Prototypen, beschrieben in Kapitel 5.4. Dementsprechend werden hier nur die neuen oder geänderten Komponenten beschrieben.
Der “NetworkController“ ist für die Verbindungsaufnahme zu der Photon-Engine verantwortlich. Die Komponenten “Photon Voice Network“ und “Recorder“ stammen aus dem Photon-Voice Plugin und sind für die Kommunikation zwischen den Benutzern zuständig, beschrieben in Kapitel 6.1.3.
Das Skript “Attachable“ funktioniert ähnlich wie bisher. Einziger Unterschied ist der Ablauf beim Greifen eines Objektes. Hier ruft das Skript erst noch die Funktion *RequestOwnership()* auf um Besitzer des Objektes zu werden, beschrieben in Kapitel 6.1.4.
Das Skript “MoveableObject“ zusammen mit den beiden Komponenten “Photon View“ und “Photon Transform View“ sind dafür zuständig, dass die Objekte zwischen den Instanzen synchronisiert werden, beschrieben in Kapitel 6.1.1.

7 Schlussfolgerung

Alle im Projekt definierten Anforderungen konnten umgesetzt und implementiert werden. Dank den beiden Nutzerevaluationen und dem konstanten Feedback der Betreuungsperson sowie anderen Studierenden konnte der Prototyp von Sprint zu Sprint überarbeitet und weiterentwickelt werden.

Die erste Nutzerevaluation ergab, dass das Vibrations-Feedback die beste Variante ist, um dem Benutzer ein Feedback beim Zusammenbau zu geben. Die umgesetzten Features halfen den Teilnehmern die Bauteile ohne grosse Vorkenntnisse und sehr intuitiv zusammenzubauen.

Der zweite Teil des Projektes verlief dank der Photon-Engine und dem Unity-Asset, welches von der Firma Exit Games zur Verfügung gestellt wurde, nahezu reibungslos. Die Erkenntnisse aus dem ersten Teil des Projekts konnten erfolgreich umgesetzt werden. Während der Nutzerevaluationen des ersten Single-User Prototypen wurden diverse Bugs gefunden, welche im Multi-User Prototyp behoben werden konnten. Aus zeitlichen Gründen wurde nur die Vibrations-Feedback Variante umgesetzt, nicht aber die Kombination zweier Varianten, welche im Kapitel 5.2.3 beschrieben wurde.

Der vorliegende Prototyp einer Multi-User Applikation beinhaltet beide Varianten, welche in der zweiten Nutzerevaluation verglichen wurden. Die “First come, First grab“ Variante eignet sich besonders für gemeinsame Arbeiten in der virtuellen Realität. Soll jedoch ein Produkt präsentiert werden oder wird die Applikation für Schulungszwecke verwendet, wird von Vorteil die Variante mit dem Zauberstab verwendet. Dank der gut dokumentierten Photon-Engine konnte der Multi-User Prototyp innerhalb sehr kurzer Zeit und ohne grosse Probleme umgesetzt werden.

8 Ausblick

Mit den vorliegenden Prototypen wurde eine solide Grundlage für das gemeinsame Arbeiten in der virtuellen Realität geschaffen. In einem weiteren Schritt würden nun die Erkenntnisse aus der zweiten Nutzerevaluation in die Applikation eingebracht und die folgenden zwei Anpassungen gemacht:

- Im zweiten Teil des Projektes wurde aus zeitlichen Gründen nur die Vibrations-Rückmeldung umgesetzt. Anhand der Rückmeldungen der ersten Nutzerevaluation würden diverse Teilnehmer eine Kombination dieser Variante mit der Variante bevorzugen, bei welcher die Freiheitsgrade eingeschränkt werden. Um bei einer Maschine zum Beispiel eine Schraube einzusetzen, würde die Kombination dieser beiden Variante eine grosse Hilfe sein.
- Damit eine neue Maschine in der Applikation integriert werden kann, sind aktuell viele Schritte notwendig. Um dies zu erleichtern müsste die Integration entweder automatisiert oder die Anzahl Schritte erheblich verringert werden. Zudem können zum aktuellen Zeitpunkt nur Modelle implementiert werden, bei welchen jedes Bauteil eine Rotation von 0° auf allen Achsen hat. Ansonsten funktioniert das «Snapping» nicht richtig.

In Kapitel 5.1.2 wurde beschrieben, dass für das Highlight eine Kopie des Bauteils gemacht und auf diese Kopie dann das neue Material gepackt wird. Dies ist sehr ineffizient und könnte einfacher gelöst werden. Jeder “Mesh Renderer“ ist in der Lage mehrere Materials gleichzeitig zu besitzen. Da das Highlight-Material nichts vom original Material verdeckt, würden so diese beiden Materialien ohne Probleme dargestellt werden.

Ein Folgeprojekt könnte sich darauf fokussieren, CAD oder ähnliche Modelle automatisch für Unity zu konvertieren. Diese Modelle könnten dann (semi-) automatisch in die bestehende Multi-User Applikation integriert und vorbereitet werden, so dass im besten Falle ein Modell direkt aus einem 3D-Programm in die Multi-User Applikation exportiert und verwendet werden kann.

Abbildungsverzeichnis

2.1 Verlust der Genauigkeit bei der Konvertierung eines CAD-Models zu VR . . . 2
2.2 Einschränkungen beim Zusammenbau in VADE 3
2.3 ImmersiveDesk: Tracking der Personen mithilfe von Markern 4

3.1 HTC Vive 8

4.1 SteamVR Highlight 9

5.1 Highlight in der Single User Applikation 16
5.2 Trigger zwischen Bauteilen 16
5.3 Highlight des Snappings 16
5.4 Kollision zwischen zwei Objekten und der konstanten Kraft 17
5.5 Highlight einer Baugruppe 18
5.6 Visuelles Feedback bei einer Kollision 18
5.7 Einzige bewegbare Achse durch die Einschränkung der Freiheitsgrade 19
5.8 Erste Aufgabe der ersten Nutzerevaluation 20
5.9 Zweite Aufgabe der ersten Nutzerevaluation 20
5.10 Systemarchitektur des Single-User Prototypen 22
5.11 Klassendiagramm des Single-User Prototypen 23

6.1 Photon Dashboard 24
6.2 Avatar leicht nach links blickend 26
6.3 Zauberstab angehängt an der rechten Hand 27
6.4 Wohin muss das Bauteil? 28
6.5 Welches Bauteil ist es? 28
6.6 Systemarchitektur des Multi-User Prototypen 30
6.7 Klassendiagramm des Multi-User Prototypen 31

Tabellenverzeichnis

3.1 Rahmenplan 5
3.2 Sprintplan 5
3.3 Meilensteine 6
3.4 Single-User Anforderungen 6
3.5 Multi-User Anforderungen 6
3.6 Eintrittswahrscheinlichkeit 7
3.7 Schadensausmass 7
3.8 Risiken vorher 7
3.9 Risiken nachher 8

4.1 Varianten für das Feedback beim Zusammenbau 10
4.2 Varianten für die Avatar-Repräsentation 11
4.3 Varianten für die Kommunikation zwischen den Benutzern 12
4.4 Varianten für gleichzeitige Interaktion am selben Objekt 13

Literatur

Boud, A. C., Haniff, D. J., Baber, C. & Steiner, S. J. (1999). Virtual reality and augmented reality as a training tool for assembly tasks. In *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)* (S. 32–36). doi:10.1109/IV.1999.781532

Carlsson, C. & Hagsand, O. (1993). DIVE A multi-user virtual reality system. In *Proceedings of IEEE Virtual Reality Annual International Symposium* (S. 394–400). doi:10.1109/VRAIS.1993.380753

Chen, W. (2015). *Collaboration in Multi-user Immersive Virtual Environment* (Diss., Université Paris-Saclay). Zugriff 20. Februar 2019 unter https://tel.archives-ouvertes.fr/tel-01340364/document

He, Z., Zhu, F. & Perlin, K. (2017). PhyShare: Sharing Physical Interaction in Virtual Reality. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology* (S. 17–19). UIST '17. event-place: Québec City, QC, Canada. doi:10.1145/3131785.3131795

Magnenat-Thalmann, N. & Egges, A. (2015). Interactive Virtual Humans in Real-Time Virtual Environment. *International Journal of Virtual Reality*, *06*(3), 15–24. Zugriff 21. Februar 2019 unter https://hal.archives-ouvertes.fr/hal-01530607

Photon Engine. (2019). Zugriff 29. Mai 2019 unter https://www.photonengine.com/

Photon Voice 2. (2019). Zugriff 30. Mai 2019 unter https://assetstore.unity.com/packages/tools/audio/photon-voice-2-130518

Pinho, M. S., Bowman, D. A. & Freitas, C. M. (2002). Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (S. 171–178). VRST '02. event-place: Hong Kong, China. doi:10.1145/585740.585769

Podkosova, I., Vasylevska, K., Schoenauer, C., Vonach, E., Fikar, P., Bronederk, E. & Kaufmann, H. (2016). Immersivedeck: a large-scale wireless VR system for multiple users. In *2016 IEEE 9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)* (S. 1–7). doi:10.1109/SEARIS.2016.7551581

Ruddle, R. A., Savage, J. C. D. & Jones, D. M. (2002). Symmetric and Asymmetric Action Integration During Cooperative Object Manipulation in Virtual Environments. *ACM Trans. Comput.-Hum. Interact. 9*(4), 285–308. doi:10.1145/586081.586084

Schild, J., Misztal, S., Roth, B., Flock, L., Luiz, T., Lerner, D., ... Herpers, R. (2018). Applying Multi-User Virtual Reality to Collaborative Medical Training. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (S. 775–776). doi:10.1109/VR.2018.8446160

Simeone, A. L., Velloso, E. & Gellersen, H. (2015). Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In *Proceedings of the 33rd Annual*

ACM Conference on Human Factors in Computing Systems (S. 3307–3316). CHI '15. event-place: Seoul, Republic of Korea. doi:10.1145/2702123.2702389

SteamVR Plugin. (2019). Zugriff 29. Mai 2019 unter https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

Tching, L., Dumont, G. & Perret, J. (2010). Interactive simulation of CAD models assemblies using virtual constraint guidance. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *4*(2), 95–102. doi:10.1007/s12008-010-0091-7

The Lab. (2019). Zugriff 30. Mai 2019 unter https://store.steampowered.com/app/450390/The_Lab/

Tirumali, a. H., Lyons, K. & Hart, P. (1999). VADE: a Virtual Assembly Design Environment. *IEEE Computer Graphics and Applications*, *19*(6), 44–50. doi:10.1109/38.799739

Unity. (2019). Zugriff 29. Mai 2019 unter https://unity.com/

VIVE™. (2019). Zugriff 29. Mai 2019 unter https://www.vive.com/eu/

Anhang

Die folgenden Anhänge sind im Anhang.zip zu finden:

- A: Aufgabenstellung
- B: Single-User Prototyp
- C: Nutzerevaluation Single-User Prototyp
- D: Multi-User Prototyp
- E: Nutzerevaluation Multi-User Prototyp