

# Konzeption und Implementierung einer Trainingsanwendung für Stomapatienten in der virtuellen Realität

Masterarbeit  
von  
Florian Gensing

Erstprüfer: Prof. Dr. rer. nat. Rainer Brück  
Zweitprüfer: Dr.-Ing. Armin Grünewald  
Tanja Eiler, M. Sc.

# Plagiatserklärung

Ich versichere, dass ich die schriftliche Ausarbeitung selbständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach (inkl. Übersetzungen) anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle (einschließlich des World Wide Web sowie anderer elektronischer Datensammlungen) deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für angefügte Zeichnungen, bildliche Darstellungen, Skizzen und dergleichen. Ich nehme zur Kenntnis, dass die nachgewiesene Unterlassung der Herkunftsangabe als versuchte Täuschung gewertet wird.

---

Ort, Datum

---

Unterzeichner

# Danksagung

Zuerst möchte ich mich bei Prof. Dr. Rainer Brück, Dr. Armin Grünewald und Tanja Eiler dafür bedanken, dass sie mich bei dieser Arbeit betreut haben und jederzeit für Fragen bereitstanden.

Außerdem möchte ich mich bei Dr. Alexander Beham und Dirk Theis des Kreisklinikums Siegen für ihre fachliche Expertise und ihre Vorschläge bei der Entwicklung der Anwendung bedanken.

Ein besonderer Dank geht an meine Freunde Kai Schnieber, Katrin Schnieber und Vanessa Schmücker, die mich während des gesamten Studiums unterstützt haben. Zusätzlich möchte ich mich bei ihnen für die Hilfe und das Korrekturlesen dieser Arbeit bedanken.

Florian Gensing  
Siegen, 01.02.2021

# Inhaltsverzeichnis

<b>Akronyme</b>	<b>I</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation und Problemstellung . . . . .	1
1.2. Ziele der Arbeit . . . . .	2
1.3. Aufbau der Arbeit . . . . .	2
1.4. Typografische Konvention . . . . .	3
<b>2. Medizinische Grundlagen</b>	<b>4</b>
2.1. Was ist ein Stoma? . . . . .	4
2.2. Wann ist ein Stoma notwendig? . . . . .	4
2.3. Stomaversorgung . . . . .	6
<b>3. Technische Grundlagen</b>	<b>8</b>
3.1. Virtual Reality . . . . .	8
3.1.1. Bildschirmausgabe in VR . . . . .	8
3.1.2. Tracking/Navigation . . . . .	9
3.1.3. Immersion und Präsenz . . . . .	11
3.1.4. Körpertransfer-Illusion . . . . .	12
3.1.5. Interaktions- und Eingabemöglichkeiten . . . . .	13
3.2. Wahl der Entwicklungsumgebung . . . . .	16
3.2.1. Unreal Engine . . . . .	16
3.2.2. Unity Engine . . . . .	17
3.2.3. Entscheidung für die Engine . . . . .	17
3.2.4. Unity-Grundlagen . . . . .	18
3.3. Benutzte Software . . . . .	21
3.3.1. SteamVR . . . . .	21
3.3.2. Blender . . . . .	21
3.3.3. MakeHuman . . . . .	21
3.3.4. Andere Modelle und Texturen . . . . .	22
<b>4. State of the Art</b>	<b>23</b>
4.1. Konventionelles Stomatraining . . . . .	23
4.2. VR in medizinischen Anwendungen . . . . .	24
4.3. Forschungsbedarf . . . . .	28

<b>5. Konzeptentwicklung</b>	<b>29</b>
5.1. Anforderungen . . . . .	29
5.2. Konzept . . . . .	30
5.2.1. Gewählte Abläufe . . . . .	30
5.2.2. Trainingsphasen . . . . .	31
5.2.3. Grundstruktur der Anwendung . . . . .	32
5.2.4. Szenen . . . . .	33
5.2.5. Immersion und gewählte Hardware . . . . .	34
<b>6. Implementierung</b>	<b>36</b>
6.1. Übersicht der Skripts . . . . .	36
6.1.1. Skripts des Hauptmenü . . . . .	36
6.1.2. Skripts der Videoszene . . . . .	37
6.1.3. Skripts der VR-Szenen . . . . .	38
6.2. Design der Szenen . . . . .	41
6.2.1. Design des Hauptmenüs . . . . .	41
6.2.2. Design der Videoszene . . . . .	42
6.2.3. Design der Badezimmerzene . . . . .	43
6.2.4. Design der Schlafzimmerszene . . . . .	44
6.3. Implementierung des Hauptmenüs . . . . .	45
6.4. Implementierung der Videoszene . . . . .	50
6.5. Implementierung der VR-Grundsteuerung . . . . .	53
6.5.1. Steuerung der VR-Komponenten . . . . .	53
6.5.2. Controller-Skript . . . . .	56
6.5.3. SteamController . . . . .	60
6.5.4. Body Tracking . . . . .	61
6.6. Implementierung der Virtual Reality Szenen . . . . .	65
6.6.1. Gemeinsame Funktionen . . . . .	65
6.6.2. Leeren des Beutels . . . . .	69
6.6.3. Wechseln des Beutels . . . . .	70
6.6.4. Wechseln der Platte . . . . .	72
<b>7. Studie</b>	<b>76</b>
7.1. Ablauf der Testung . . . . .	76
7.2. Beobachtungen während der Testung . . . . .	77
7.3. Auswertung . . . . .	78

7.4. Bewertung der Anwendung . . . . .	84
<b>8. Schlussbetrachtung</b>	<b>85</b>
8.1. Zusammenfassung . . . . .	85
8.2. Ausblick . . . . .	85
<b>A. Anhang</b>	
A.1. Persönliche Angaben . . . . .	
A.2. Hygienekonzept . . . . .	
A.3. Einwilligungserklärung . . . . .	
A.4. Fragebogen . . . . .	

# Abbildungsverzeichnis

1.	Aufbau des Gastrointestinaltrakts [Unie] . . . . .	5
2.	Verschiedene Beutelmodelle: (a) Beutel für ein Dickdarmsstoma [Colb], (b) Beutel für ein Dünndarmsstoma [Cola] . . . . .	7
3.	CAVE der University of Illinois Chicago [Ill] . . . . .	9
4.	<i>Vive Tracker</i> [Vivb] und Datenhandschuhe der Firma Noitom Hi5 [Noi] .	14
5.	Datenhandschuhe der Firma Teslasuit [Mik] . . . . .	14
6.	Übersicht des <i>Unity</i> Interfaces . . . . .	18
7.	Zuordnen einer Methode zu einem Event in <i>Unity</i> . . . . .	20
8.	Stomapflege-Simulator von „Anatomy Online“ [Onl] . . . . .	23
9.	Anwendung „Itsy“: (a) Zeigt die Spinne zu Anfang der Anwendung (b) Realistischere Spinne des späteren Anwendungsverlaufs . . . . .	25
10.	Graphische Benutzeroberfläche der Arthroskopie-Anwendung [Mül+00] .	27
11.	UML des Hauptmenüs . . . . .	36
12.	UML der Videoszene . . . . .	37
13.	UML der Controllersteuerung . . . . .	38
14.	UML der VR-Szenen . . . . .	39
15.	Design des Hauptmenüs . . . . .	42
16.	Design der Videoszene . . . . .	42
17.	Design der Badezimmerszene . . . . .	43
18.	Design der Schlafzimmerszene . . . . .	44
19.	Hauptmenü: Fenster zur Auswahl des Patienten . . . . .	46
20.	Hauptmenü: Fenster zum Anlegen und Modifizieren von Patienten . . . .	47
21.	Hauptmenü: Einführungsfenster . . . . .	48
22.	Hauptmenü: Fenster zum Auswählen des Ablaufs und der Trainingsphase	49
23.	Übersicht der Videoszene . . . . .	50
24.	Videoszene Fehler . . . . .	51
25.	Videoszene: Abfrage zum Abschließen des Ablaufs . . . . .	52
26.	Videoszene: Anzeige der vergessenen Schritte . . . . .	53
27.	Das Fenster „SteamVR Input“ . . . . .	54
28.	<i>SteamVR</i> : Wahl der Geräte vor dem Einstellen der Tastenbelegung . . .	55
29.	Tastenbenennung der <i>Vive Controller</i> [Ske] . . . . .	55
30.	Einstellung der Tastenbelegung eines Controllers in <i>SteamVR</i> . . . . .	56
31.	Outline-Skript . . . . .	58

32.	Verschiedene einstellungen für <b>Outline Mode</b> : (a) Outline Visible, (b) Outline Hidden, (c) Outline All, (d) Silhouette Only und (e) Outline and Silhouette . . . . .	59
33.	<b>SteamController</b> -Skript . . . . .	60
34.	<i>SteamVR</i> -Interface . . . . .	61
35.	<i>SteamVR</i> Manage Tracker Ansicht . . . . .	62
36.	SteamVR-Controllerauswahl mit Trackern . . . . .	63
37.	Auswahl der SteamVR-Tracker . . . . .	63
38.	Importeinstellung der menschlichen Modelle . . . . .	64
39.	Körperhaltung des Patienten in Angehörigensicht: (a) Badezimmerszene (b) Schlafzimmerszene . . . . .	65
40.	Editoransicht des <b>SnapObject</b> -Skripts . . . . .	67
41.	Editoransicht des <b>SnapTarget</b> -Skripts . . . . .	68
42.	Visualisierung der Stellen, an welchen der Beutel festgedrückt wird . . . . .	71
43.	Übersicht der beim Wechseln der Platte verwendeten Objekte . . . . .	72
44.	Schneiden der Platte . . . . .	73
45.	Bewertung der örtlichen Präsenz . . . . .	78
46.	Bewertung der Involviertheit . . . . .	79
47.	Bewertung des Realismus nach <i>igroup</i> . . . . .	80
48.	Bewertung des Realismus nach <i>Witmer und Singer</i> . . . . .	80
49.	Bewertung der Interaktionsmöglichkeiten . . . . .	81
50.	Bewertung des Interfaces . . . . .	82

## Akronyme

**BMI** Body-Mass-Index. 34, 37, 47, 66, 85

**HMD** Head-Mounted-Display. 8, 9, 12, 13, 16, 76, 77

**UI** User Interface. 20, 53

**VR** Virtual Reality. 1, 2, 8–13, 15, 16, 21, 23–28, 31, 32, 34, 35, 39, 49, 50, 53, 54, 61, 65, 66, 70, 77, 83–85

# 1. Einleitung

Der Gedanke, in virtuelle Welten eintauchen zu können, hat Menschen schon seit langer Zeit fasziniert. Mithilfe moderner Technologien ist diese Idee heute umsetzbar und ist bereits in der Spieleindustrie verbreitet. Doch nicht nur im Bereich der Unterhaltung hat die virtuelle Realität ihre Nische gefunden. Da VR die Möglichkeit bietet, realitätsnah mit virtuellen Objekten zu interagieren, kann sie auch in vielen anderen Bereichen verwendet werden. Einer dieser Anwendungsbereiche ist das Training von medizinischen Abläufen.

In dieser Arbeit wird die virtuelle Realität verwendet, um Stomapatienten die Versorgung ihres Stomas beizubringen. Dazu ist die Stomaversorgung in drei Abläufe eingeteilt worden, welche zuerst anhand von Videos gelernt und anschließend in VR durchgeführt werden können. Diese Anwendung ist schließlich von Mitarbeitern des Kreisklinikums Siegen und des Lehrstuhls *Medizinische Informatik und Mikrosystementwurf* der Universität Siegen getestet worden.

## 1.1. Motivation und Problemstellung

Ein Stoma ist ein künstlicher Darmausgang. Dieser kann unter anderem bei Patienten mit Darmkrebs notwendig werden. Obwohl Stomata bei älteren Menschen häufiger notwendig sind, ist es dennoch ein Problem, dass alle Altersklassen betrifft. Viele Betroffene empfinden die Zeit mit dem Stoma als eine schwierige Phase in ihrem Leben. Allein in Deutschland leben insgesamt rund 160.000 Menschen mit einem Stoma [Int; BBr]. Manche haben Zeit, sich auf das Leben mit einem Stoma vorzubereiten, für andere kommt dies sehr überraschend.

Für viele Betroffene stellt der künstliche Darmausgang eine große Umstellung des Alltags dar und oftmals fühlen sich die Patienten wegen dem Stoma unwohl. Vor allem anfangs sind sie verunsichert, ob sie weiterhin Sport machen und ihre Hobbies ausleben können. Um der Angst der Patienten entgegenzuwirken, soll ihnen das Gefühl vermittelt werden, dass sie sich selbständig um das Stoma kümmern können. Daher ist die Idee aufgekommen, eine VR-Anwendung für das Training von Stomapatienten und Angehörigen zu entwickeln.

## **1.2. Ziele der Arbeit**

Das Ziel dieser Arbeit ist es, eine Anwendung zu entwickeln, welche angehende Stomapatienten mithilfe der virtuellen Realität auf das Leben mit einem Stoma vorbereitet. Die Anwendung soll dabei ergänzend zu existierenden Trainingsmethoden im Krankenhaus dienen und diese nicht ersetzen. Außerdem soll sie Patienten die Angst vor dem Stoma nehmen und ihnen beibringen, wie sie dieses versorgen. Dazu sollen die Patienten die Abläufe der Stomaversorgung erst einmal beobachten und anschließend selbst durchführen können.

Um die Erfahrung der Patienten realistischer zu gestalten, soll die Anwendung am eigenen Körper durchgeführt werden können. Dazu wird Körpertracking verwendet, um die Bewegungen des Patienten in die virtuelle Welt zu übertragen.

Da hauptsächlich ältere Menschen betroffen sind, muss die Anwendung so entworfen werden, dass sie auch für diese geeignet ist. Daher ist es wichtig, dass die Anwendung leicht zu benutzen ist und kein technisches Vorwissen benötigt wird.

## **1.3. Aufbau der Arbeit**

Um eine solche Anwendung zu entwickeln, ist es notwendig, einen Überblick über die Stomata und die Grundlagen der Stomaversorgung zu erlangen. Diese werden in Kapitel 2 erläutert. Damit die Anwendung die Vorteile der virtueller Realität voll ausnutzen kann, muss sich zu Beginn mit den Grundlagen von VR sowie der dafür verfügbaren Hardware auseinandergesetzt werden. Dies geschieht in Kapitel 3.1.

Danach wird in Kapitel 4.1 ein Blick auf aktuelle, bereits bestehende Trainingsangebote geworfen. Obwohl es noch keine Anwendungen zum Training von Stomapatienten mithilfe der virtuellen Realität gibt, gibt es viele Anwendungen, welche die virtuelle Realität zum Training oder zur Fortbildung benutzen. Aus diesen kann viel über die Vorteile der VR zum Training gelernt werden, weshalb einige davon in Kapitel 4.2 beispielhaft untersucht werden.

Anschließend kann aus dem gesammelten Wissen ein Konzept für die Anwendung entworfen werden, welches in Kapitel 5 vorgestellt wird. Dabei werden bereits Entscheidungen für die gewählte Hardware getroffen und die Szenen entworfen.

Genauere Details zur Umsetzung dieses Konzeptes befinden sich in Kapitel 6. Im Anschluss werden die Ergebnisse in Kapitel 7 evaluiert und die gesammelten Eindrücke in Kapitel 8 zusammengefasst.

## 1.4. Typografische Konvention

Da in dieser Arbeit viele Fachwörter verwendet werden, werden sie nach den folgenden Konventionen markiert:

- Firmennamen und Produktbezeichnungen werden *kursiv* geschrieben
- Fachbezeichnungen werden *kursiv* geschrieben, wenn sie nicht weiter erläutert werden
- Fachbegriffe, welche weiter erklärt werden, werden einmal in „Anführungszeichen“ gesetzt
- Methoden-, Variablen- und Skriptnamen werden in **dieser Schriftart** geschrieben

Um die Arbeit lesbarer zu gestalten, wird auf die gleichzeitige Verwendung der Sprachformen für alle Geschlechter verzichtet. Es werden demnach männliche Personenbezeichnungen verwendet, welche allerdings für alle Geschlechter gelten sollen.

## 2. Medizinische Grundlagen

### 2.1. Was ist ein Stoma?

Das Wort „Stoma“ kommt aus dem Griechischen und bedeutet „Öffnung“. Damit sind operativ angelegte Körperöffnungen, meist ein künstlicher Dünn- oder Dickdarmausgang, gemeint. Bei einem künstlichen Darmausgang wird Stuhl durch den künstlichen Ausgang in einen Beutel befördert und dort aufgefangen [Deu].

Bei Krankheiten, welche den Darm oder die Harnleiter betreffen, kann es notwendig werden, den betroffenen Abschnitt zu entfernen. In diesem Fall wird ein künstlicher Ausgang operativ auf die Hautoberfläche verlegt.

Mit dem künstlichen Ausgang kann das Ausscheiden nicht mehr kontrolliert werden. Daher werden spezielle Beutel benutzt, um den Stuhl oder Urin aufzufangen. Eine Platte mit einem passend geschnittenem Loch wird dazu auf der Haut um das Stoma herum platziert. Wenn das Loch zu groß oder zu klein ist, kann es zu Entzündungen kommen. Auf dieser Platte wird der Beutel befestigt, welcher die Ausscheidungen letztlich auffängt. In dieser Arbeit werden nur Dünn- und Dickdarmsomata betrachtet.

### 2.2. Wann ist ein Stoma notwendig?

Verschiedene Krankheiten können einen künstlichen Ausgang notwendig machen. Bevor diese weiter erläutert werden, werden ein paar Grundlagen benötigt.

Der erste Begriff, der hier erläutert wird, ist der Gastrointestinaltrakt, auch als Magen-Darm-Trakt bekannt, welcher in Abbildung 1 gesehen werden kann. Der Gastrointestinaltrakt beginnt am Mund und endet am Anus. Nahrung wird über den Mund aufgenommen und über die Speiseröhre an den Magen weitergeleitet. Als nächstes gelangt sie in den Dünndarm, welcher die Nährstoffe aufnimmt und an den Körper weitergibt. Danach wird sie an den Dickdarm weitergeleitet, welcher das Wasser aufnimmt. Abschließend gelangt sie in das Rektum, wo sie gesammelt wird, bevor sie den Körper verlässt.

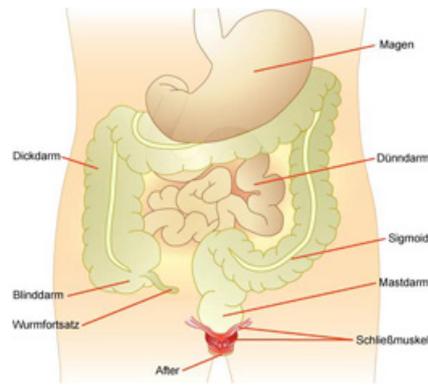


Abbildung 1: Aufbau des Gastrointestinaltrakts [Unie]

Als zweites werden einige Grundlagen zu Krebs benannt. Wenn sich Zellen im eigenen Körper unkontrolliert vermehren, bilden diese Tumore. Diese werden in gutartig (benigne) und bösartig (maligne) unterteilt. Gutartige Tumore zerstören das umliegende Gewebe nicht, sondern verdrängen es nur. Bösartige Tumore hingegen können in das umliegende Gewebe einwachsen und es somit beschädigen. Diese sind als Krebs bekannt [Krea]. Bösartige Tumore können sich auch im Rest des Körpers ausbreiten und dort weitere Tumore bilden. Diese werden als Metastasen bezeichnet [Kreb].

Die bösartigen Tumore werden abhängig von den Geweben, aus welchen sie entstehen, in verschiedene Kategorien eingeteilt. Sarkome entwickeln sich aus Bindegewebe oder Stützgewebe, also aus Fettgewebe, Muskeln, Sehnen und Knochen. Karzinome, die häufigste Art von Krebs, entstehen aus Gewebe, welches innere und äußere Oberflächen des Körpers, wie beispielsweise Schleimhäute, bedeckt. Außerdem gibt es Blastome, welche bei der Gewebeentwicklung entstehen. Diese kommen hauptsächlich bei Kindern vor.

Eine der häufigsten Krankheiten, bei welchen ein Stoma notwendig wird, ist das Kolonkarzinom, auch Dickdarmkrebs genannt, welches meist bei älteren Personen auftritt [Han].

Die Symptome hängen hier stark von der Position des Tumors ab. Zu den häufigsten zählen Blut im Stuhl, Durchfall, Verstopfung und Bauchschmerzen. Je weiter der Tumor wächst, desto schlimmer werden die Symptome.

Die Diagnose kann unter Verwendung verschiedener Methoden erfolgen. Oft gibt es eine Tastuntersuchung des Enddarms, welche allerdings erst in den späteren Stadien effektiv wird. Zusätzlich kann der Stuhlgang auf Blut untersucht werden. Dies kann aber nur nachweisen, dass es irgendwo im Gastrointestinaltrakt eine Blutung gibt. Die genaue Position sowie die Ursache können allerdings auf diese Weise nicht bestimmt werden.

Schließlich gibt es noch die Koloskopie, bei welcher ein Endoskop in den Dickdarm eingeführt wird. Mit diesem kann der untersuchende Arzt den Dickdarm betrachten [Han].

Ist der Krebs einmal entdeckt worden, kann die Behandlung beginnen. Der Tumor und der umliegende Darmabschnitt werden mittels einer Operation entfernt. Wenn der Tumor bereits zu groß ist, ist vor der Operation eine Chemo- und/oder Strahlentherapie notwendig. Wenn sich der Tumor im unteren Drittel des Darms befindet, muss ein Stück des Darms bis zum Schließmuskel entfernt werden. In diesem Falle wird ein Stoma notwendig, damit das Darmende verheilen kann, ohne, dass Stuhl es passieren muss. Das Stoma kann in der Regel nach ungefähr drei Monaten wieder zurückverlegt werden. Falls bei der Operation für ein Colostoma allerdings der Schließmuskel mit entfernt werden muss oder seine Funktion nicht mehr erfüllen kann, muss der Patient das Stoma permanent behalten [Han].

### **2.3. Stomaversorgung**

Um Infektionen zu vermeiden, ist es wichtig, dass Beutel und Platte regelmäßig ausgetauscht werden.

Wie bereits beschrieben, dient ein Stoma zum Abführen von Stuhl und besteht aus einer Platte, auf welcher ein Beutel befestigt wird.

Im Falle eines Dickdarmstomas ist der Stuhl bereits fest, weswegen die Beutel in der Regel nicht gelehrt, sondern ausgewechselt werden müssen, sobald sie voll sind [- Ea]. Die Beutel besitzen eingebaute Kohlefilter, um den Geruch zu neutralisieren. Bei Dickdarmstomata werden meistens zweiteilige Beutelsysteme benutzt. Das bedeutet, dass die Platte und der Beutel getrennt voneinander ausgetauscht werden können.

Beim Auswechseln der Platte muss beachtet werden, dass das ausgeschnittene Loch genau zur Stomaform passt. Wenn das Loch zu klein ist, könnte das Stoma beschädigt werden. Ist es allerdings zu groß, könnte Magensäure an die Haut gelangen, was zu starken Hautirritationen führen kann. Die Platte sollte also spätestens gewechselt werden, wenn sie nicht mehr richtig an der Haut haftet. In der Regel wird der Beutel gewechselt, wenn er annähernd voll ist.

Bei einem Dünndarmstoma muss der Beutel mehrmals am Tag ausgeleert werden. Dafür gibt es unten am Beutel eine Öffnung, durch welche der Stuhl entleert werden kann. Der Beutel kann dadurch mehrfach verwendet werden und muss nicht jedes mal ausgetauscht werden [- Eb]. Er muss spätestens gewechselt werden, wenn der Aktivkohlefilter seine Funktion nicht mehr erfüllen kann oder der Verschluss nicht mehr richtig funktioniert.

Bei Dünndarmstomata werden oftmals einteilige Beutelsysteme benutzt, bei welchen Beutel und Platte eine Einheit bilden. In diesem Fall muss das Beutelsystem auch gewechselt werden, wenn die Platte nicht mehr gut an der Haut haftet. Dies ist bei Dünndarmstomata noch wichtiger als bei Dickdarmstomata, da mehr Verdauungsenzyme in dem Stuhl enthalten sind, wodurch die Haut stärker gereizt werden würde.

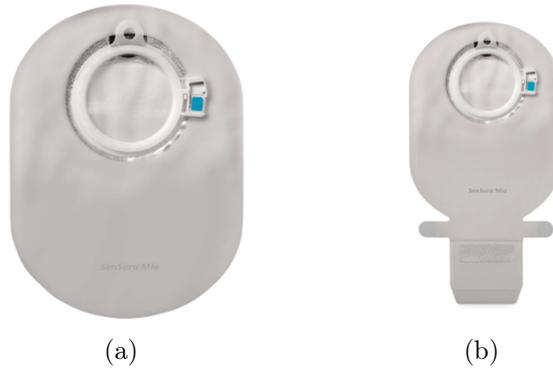


Abbildung 2: Verschiedene Beutelmodelle: (a) Beutel für ein Dickdarmstoma [Colb], (b) Beutel für ein Dünndarmstoma [Cola]

In Abbildung 2 kann je ein Beutel für ein Dickdarm- und ein Dünndarmstoma gesehen werden. In beiden Beispielen ist ein zweiteiliges Beutelsystem verwendet worden. Hier kann der Verschluss an dem Beutel für das Dünndarmstoma gesehen werden.

## 3. Technische Grundlagen

### 3.1. Virtual Reality

Der Begriff „Virtual Reality“ beschreibt eine Technologie, welche dem Benutzer eine computergenerierte Szene auf eine Weise präsentiert, die bei ihm den Eindruck erweckt, er würde sich tatsächlich in dieser befinden [Mye01].

Im Gegensatz zu normalen PC-Anwendungen wird die VR-Anwendung durch Bewegungen in der realen Welt gesteuert. Dabei kann der Grad von der Verwendung der Blickrichtung bis hin zu einem kompletten Körpertracking reichen. Im Gegensatz zur Augmented Reality (AR) wird bei Virtual Reality die tatsächliche Realität so weit es geht ausgeblendet. Um das VR-Erlebnis also möglichst realistisch zu gestalten, muss sowohl die Eingabe als auch die Ausgabe intuitiv und realitätsnah sein.

#### 3.1.1. Bildschirmausgabe in VR

Um die Immersion, welche in Kapitel 3.1.3 weiter erläutert wird, in der virtuellen Welt zu erhöhen, werden keine konventionellen Bildschirme benutzt. Stattdessen gibt es hauptsächlich zwei Optionen. Die erste Option sind „CAVE“-Systeme. Das sind spezielle Räume, bei welchen die Wände mit Monitoren ausgestattet sind. Die andere Option sind sogenannte Head-Mounted Displays (HMD). Das sind Brillen, bei denen direkt vor den Augen jeweils ein Monitor angebracht wird.

In beiden Fällen ist es wichtig, dass sowohl die Position des Kopfes des Benutzers als auch die Blickrichtung durch das Tracking verfolgt wird, sodass die angezeigten Bilder darauf angepasst werden. Sonst kann es zur VR-Krankheit (Cybersickness) kommen. Diese wird im Kapitel 3.1.2 genauer erläutert.

#### **Cave-Systeme**

„CAVE“ (Cave Automatic Virtual Environment) beschreibt einen speziellen Raum, bei welchem Wand-, Boden- oder Dachflächen durch Projektionsflächen oder Bildschirme ersetzt werden. Die Bildschirmmaße und Abstände im Raum müssen im Voraus gemessen werden. Dadurch wird es möglich, die virtuelle Welt richtig auf allen Bildschirmen darzustellen.

CAVE-Systeme haben den Vorteil, dass mehrere Personen gleichzeitig den virtuellen Raum sehen können, wodurch sie gut zum Visualisieren von Daten geeignet sind [CSD93]. Sie werden oft für Anwendungen eingesetzt, bei denen Umgebungen wie beispielsweise

Flugsimulatoren oder eine dreidimensionale Ansicht von großen Geräten oder Gebäuden betrachtet werden.

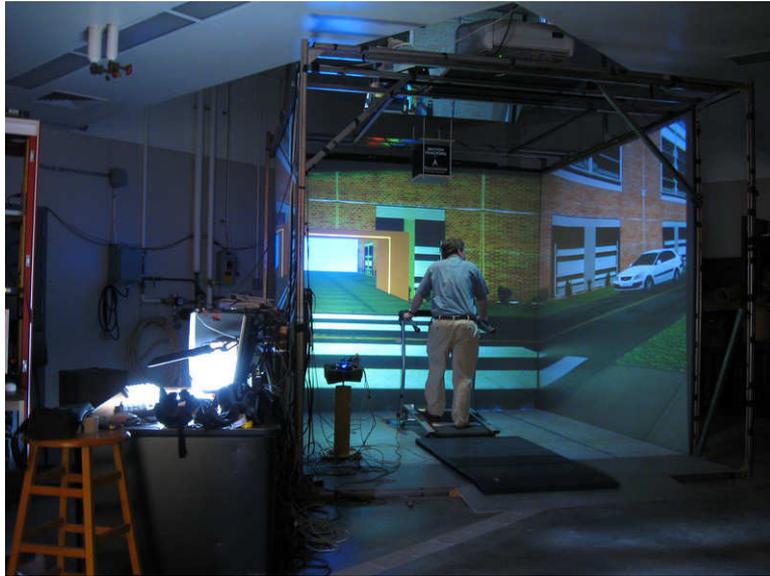


Abbildung 3: CAVE der University of Illinois Chicago [III]

### **Head-Mounted Display**

Head-Mounted Displays (HMD) sind spezielle Brillen, welche ein dediziertes Bild für jedes Auge anzeigen. Dadurch wird stereoskopisches Sehen der VR-Szene möglich. VR-Brillen ermöglichen Tracking, wodurch die Anzeige auf die Position und Blickrichtung angepasst werden kann. Das Tracking kann hierbei von der Brille selbst durchgeführt werden, häufiger aber werden dazu separate Tracking-Geräte benutzt, wie im folgenden Kapitel 3.1.2 beschrieben wird.

#### **3.1.2. Tracking/Navigation**

Für die Steuerung von VR-Anwendungen werden die Position und Orientierung von realen Objekten benutzt. So können beispielsweise auch die Positionen und Orientierungen der Hände durch Controller oder andere Geräte ausgelesen und benutzt werden. Dadurch können virtuelle Objekte in VR „gegriffen“ werden, indem zum Beispiel ein Controller an die entsprechende Position in der realen Welt bewegt und das Objekt dort durch einen Knopfdruck gegriffen wird. Bei Datenhandschuhen kann auch eine Greifbewegung benutzt werden. Dadurch wird die Immersion, welche in Kapitel 3.1.3 genauer erläutert wird, deutlich erhöht und die Anwendung intuitiver.

## Arten des Trackens

Zum Tracken können vier verschiedene Techniken verwendet werden: „Marker-based Inside-Out Tracking“, „Marker-based Outside-In Tracking“, „Markerless Inside-Out Tracking“ und „Markerless Outside-In Tracking“. Kann das Objekt selbst bestimmen, wo es sich im Raum befindet, so handelt es sich um „Inside-Out Tracking“. Wird das Objekt hingegen von außen verfolgt, handelt es sich um „Outside-In Tracking“. Die Begriffe „Marker-based“ und „Markerless“ geben hierbei an, ob das Tracking besondere Markierungen benutzt, um Objekte zu finden oder die Position in einem Raum zu bestimmen [Zie].

Ein Beispiel für „Marker-based Outside-In Tracking“ liefert die „Oculus Rift 1“ [Ocu]. Bei dieser VR-Brille verwendet das Tracking Kameras, welche in Richtung des nutzbaren VR-Bereichs ausgerichtet sind. Die Brille strahlt eine Infrarot-„Markierung“ aus, welche die Kameras aufzeichnen. Aus dieser Information und der im Voraus bekannten Positionen der Kameras kann die Position der Brille berechnet werden. Der PC berechnet daraus das neue Bild und schickt dieses weiter an die VR-Brille, welche es schließlich anzeigt.

Anders dazu ist das „Marker-based Inside-Out Tracking“. Bei der *HTC Vive* [Viva], einer häufig verwendeten VR-Brille, werden zum Tracking sogenannte „Lighthouses“ benutzt. Diese sind Basisstationen, welche in zwei gegenüberliegenden Ecken eines VR-Bereiches angebracht werden. Danach wird eine Raumausmessung durchgeführt, bei welcher aufgezeichnet wird, wo im Raum sich diese Lighthouses befinden. Zum Tracking strahlen hierbei die Lighthouses ein Infrarotmuster aus, welches von der VR-Brille gefunden wird. Danach kann die Brille aus ihrer relativen Position zu den Lighthouses ihre Position im Raum bestimmen und an den PC weitergeben. Da das Tracking Infrarotsensoren benutzt, können reflektierende Oberflächen oder andere Infrarotgeräte das Tracking stören.

Neben dieser Methode gibt es auch das sogenannte „Markerless Inside-Out Tracking“. Bei diesem werden keine externen Geräte zum Tracking benötigt. Dies wird beispielsweise von der „Oculus Quest“ [Ocu] verwendet. Die Brille erkennt hierbei selbstständig wichtige Punkte im Raum, an welchen sie sich orientieren kann, ohne, dass diese besonders markiert werden müssen [Xin]. Dies hat den Vorteil, dass neben der VR-Brille keine weiteren Geräte für das Tracking benötigt werden.

Da die VR-Brille in der Regel keine Sicht mehr auf die reale Welt zulässt, ist es wichtig, dass sich in diesem Raum keine Gegenstände oder Hindernisse befinden, da man sonst mit diesen kollidieren könnte. Um diesem entgegenzuwirken, sind Systeme wie beispielsweise das „Vive Chaperone-System“ entwickelt worden. Dieses System benutzt die Front-Kamera der VR-Brille, um es zu ermöglichen, einen Umriss der physischen Umgebung anzuzeigen [Viv; Har+19].

## Navigation

Neben dem Tracking gibt es eine weitere wichtige Komponente der Bewegung in VR: Die Navigation. Sie kann als die Bewegung des durch Tracking erreichbaren Bereichs durch die virtuelle Welt verstanden werden. Im Idealfall sollte der Anwender die Möglichkeit haben, alle Bereiche der virtuellen Welt durch Tracking zu erreichen. Es kann allerdings vorkommen, dass die virtuelle Welt größer als der Trackingbereich ist. In diesem Fall wird es notwendig, die Position in der virtuellen Welt zu verändern, ohne sich in der Realität zu bewegen. Dies kann an CAVE-Systemen gut erläutert werden. Hierbei kann sich der Anwender innerhalb der CAVE frei bewegen. Möchte er sich allerdings weiter in der Welt bewegen, muss die Position der gesamten CAVE relativ zum Rest der virtuellen Welt verschoben werden.

Hierbei müssen zwei Dinge beachtet werden. Zum einen muss die Steuerung weiterhin intuitiv und immersiv bleiben, zum anderen muss aber auch darauf geachtet werden, dass Benutzer keine VR-Krankheit (Cybersickness) bekommen. Diese tritt vor allem dann auf, wenn Benutzer in der virtuellen Realität fließend von einem Ort an einen anderen bewegt werden, ohne sich dabei in der Realität zu bewegen, wie es beim Tracking der Fall wäre. Da die visuell wahrgenommene Bewegung nicht mit der des Gleichgewichtssinns übereinstimmt, kann dies zu Übelkeit führen. Das Gleiche geschieht beim Lesen im Auto, nur dass dort die Bewegung gespürt, aber nicht gesehen wird [360b]. Um die Steuerung intuitiv zu gestalten, werden oftmals Joysticks oder Trackpads der Controller benutzt. In den meisten Fällen werden die Blickrichtung und die Rotation ausschließlich über das Tracking bestimmt, während die Position aus einer Kombination von Tracking und der Navigation berechnet wird [Rie+10]. Um VR-Krankheit zu verhindern, wird die Bewegung mittels Navigation entweder mit einer niedrigen Geschwindigkeit umgesetzt oder es wird eine Teleportation anstelle einer fließenden Bewegung benutzt.

### 3.1.3. Immersion und Präsenz

Damit VR-Anwendungen effektiv sind, müssen sich Anwender wohl fühlen und sollten den Eindruck bekommen, dass es sich um die reale Welt handelt. Dadurch kann das erlernte Wissen besser übertragen werden. Hierfür sind zwei Begriffe von Bedeutung: Immersion und Präsenz [360a].

Immersion beschreibt das Eintauchen in die virtuelle Welt. Der Benutzer soll das Gefühl bekommen, sich tatsächlich in dieser Welt zu befinden, und die Realität ausblenden. Immersion kann in mentale Immersion und physikalische Immersion unterteilt werden.

Mentale Immersion erleben wir beim Lesen oder beim Filme schauen und beschreibt den Zustand des Anwenders, in eine Geschichte vertieft und involviert zu sein. Physikalische Immersion wird auf der anderen Seite durch die Sinne erzeugt. Daher ist es wichtig, nicht nur möglichst viele Sinne anzusprechen, sondern auch die Realität auszublenden. Dies wird bei HMD dadurch erreicht, dass die VR-Brille die Sicht auf die reale Welt verdeckt. Neben der visuellen Komponente wird bei VR-Anwendungen auch oft eine Audiokomponente benutzt, welche die Immersion weiter steigert. Mithilfe spezieller Hardware kann auch haptisches Feedback benutzt werden. Allerdings ist diese Hardware noch vergleichsweise teuer und wird daher nicht so oft benutzt wie die anderen Komponenten.

Neben der Immersion ist auch der Begriff „Präsenz“ wichtig. Die Präsenz wird in Ortsillusion, Plausibilitätsillusion und Involviertheit eingeteilt [Dör+19]. Die Ortsillusion beschreibt das Gefühl, sich tatsächlich an dem computergeneriertem Ort zu befinden. Dies kann durch realistische oder in sich schlüssige Umgebungen hervorgerufen werden. Zusätzlich ist besonders wichtig, dass die Darstellung der Szene an die Bewegung des Anwenders angepasst wird. Die Plausibilitätsillusion beschreibt den Eindruck, dass sich simulierte und reale Welt gleich verhalten. Dies wird dadurch gefördert, dass Objekte sich gemäß den Erwartungen des Anwenders verhalten.

Mit Involviertheit werden die Aufmerksamkeit und das Interesse des Nutzers beschrieben. Die Involviertheit wird gesteigert, wenn der Nutzer aktiv mit seiner Umgebung interagieren kann oder ein Interesse an dem Thema hat. Bei der Interaktion mit der Umgebung ist es zudem wichtig, dass diese möglichst einfach gehalten und intuitiv ist, sodass der Benutzer sich auf die Aufgaben statt auf die Handhabung konzentrieren kann.

#### **3.1.4. Körpertransfer-Illusion**

Mit Körpertransfer-Illusion (Body Ownership) ist der Effekt gemeint, einen anderen Körper als den eigenen zu besitzen. In VR kann dies auftreten, wenn der Anwender einen Avatar steuert. Hierbei sollte beachtet werden, dass das Gefühl der Body Ownership und der Präsenz am stärksten ist, wenn der Avatar dem Anwender sehr ähnlich sieht [Coe+14]. Das Aussehen des Avatars kann aber nicht nur die Präsenz beeinflussen, sondern auch das Verhalten der Anwender. So kann es beispielsweise vorkommen, dass Anwender mit einem muskulöseren Avatar einige Aufgaben als weniger anstrengend empfinden [Koc+20]. In einer weiteren Studie ist herausgefunden worden, dass das Alter des Avatars auch nach dem Beenden der Anwendung einen Einfluss auf das Verhalten der Anwender haben kann [Rei+20].

### 3.1.5. Interaktions- und Eingabemöglichkeiten

Da die Eingabe, wie in Kapitel 3.1.3 beschrieben, sehr wichtig für die Immersion ist, müssen die Eingabemöglichkeiten gut gewählt werden.

Die erste Möglichkeit, mit der virtuellen Umgebung zu interagieren, besteht in dem Tracking der VR-Brille. Dieses kann nicht nur verwendet werden, um das Bild an die Bewegung anzupassen, sondern kann auch als Eingabe verwendet werden. So gibt es beispielsweise in dem Spiel „Beat Saber“ [Sab] Hindernisse, denen der Nutzer ausweichen muss. Zusätzlich ist es möglich, die Blickrichtung als Eingabe zu verwenden, indem beispielsweise die Anwendung reagiert, wenn bestimmte Objekte angesehen werden.

Eine weitere Form der Eingabegeräte sind die sogenannten *Vive Tracker* [Vivb]. Dies sind Geräte, welche auf realen Objekten befestigt werden können, sodass diese in der virtuellen Welt an der richtigen Position angezeigt werden und sich mit dem echten Objekt mitbewegen. Sie können auch am eigenen Körper befestigt werden, um diesen zu tracken. Dies ist in dem Artikel „Estimating Forearm Axial Rotation Using Vive Trackers for Interaction With Serious Games“ [Nai+19] benutzt worden, um Kindern mit Zerebralparese durch Spiele bei der Rehabilitation zu helfen.

Die am weitesten verbreitete Interaktionsmöglichkeit in Verbindung mit Virtual Reality sind kabellose Controller, welche oftmals gemeinsam mit dem HMD verkauft werden. Die am meisten benutzten Modelle sind die Controller der *HTC VIVE* und *Oculus Rift*. Diese werden über dieselbe Software wie die dazu gehörigen HMD gesteuert. Die Controller verfügen außerdem über ein eingebautes Trackingsystem und können daher die Handposition mitverfolgen und in der virtuellen Welt anzeigen.

Anstelle von Controllern können auch Datenhandschuhe verwendet werden. Diese haben in der Regel ein eingebautes Tracking oder werden in Kombination mit Trackern verwendet. Die Kombination zwischen Datenhandschuh und Tracker kann bei den Datenhandschuhen *Noitom Hi5 VR Glove* [Noi] in Abbildung 4 gut gesehen werden. Dort wird ein *Vive Tracker* an dem Datenhandschuh befestigt, um das Tracking mit den Daten des Handschuhs zu vereinen.

Datenhandschuhe haben gegenüber Controllern den Vorteil, dass nicht nur die Position des gesamten Controllers, sondern auch die Position und Orientierung der einzelnen Finger getracked werden, was es ermöglicht, die gesamte Handbewegung in der virtuellen Realität nachzubilden und somit die Immersion weiter zu erhöhen. Bei Datenhandschuhen gibt es außerdem die Möglichkeit, haptisches Feedback einzubauen. Dadurch wird es möglich, Objekte in VR nicht nur zu sehen, sondern auch zu fühlen. Dies wird meistens über ein sogenanntes Exoskelett an dem Handschuh möglich, welches in Abbildung 5

gesehen werden kann [Mik]. Es gibt allerdings auch die Möglichkeit, haptisches Feedback mithilfe von Vibrationen oder Strom zu erzeugen.



Abbildung 4: Vive Tracker [Vivb] und Datenhandschuhe der Firma Noitom Hi5 [Noi]



Abbildung 5: Datenhandschuhe der Firma TeslaSuit [Mik]

Eine weitere Methode, die Bewegungen der einzelnen Finger zu messen, sind spezielle Infrarotkameras, welche die Handbewegungen erkennen und in die virtuelle Welt übertragen. Ein Beispiel hierfür ist die *Leap Motion* [Ult]. Diese Infrarotkamera kann vorne an der VR-Brille befestigt werden. Die *Leap Motion* hat allerdings einige Nachteile. Einer davon ist, dass die Hände in VR verschwinden, wenn sie das Sichtfeld der Kamera verlassen. Auch wenn sich beide Hände im Sichtfeld befinden, kann es sein, dass sie nicht richtig erkannt werden können. Dies kann beispielsweise geschehen, wenn die Hände sich gegenseitig bedecken. Dadurch ist diese Methode der Eingabe für viele Anwendungen nicht geeignet, wenn sie nahe Zusammenarbeit der beiden Hände benötigen, da das Verschwinden der Hände die Immersion deutlich verringern würde.

## 3.2. Wahl der Entwicklungsumgebung

Bevor mit der Implementierung begonnen werden kann, muss zuerst entschieden werden, welche Entwicklungsumgebung benutzt wird.

Als Entwicklungsumgebung für VR-Anwendungen werden meist Spiele-Engines verwendet, da diese bereits viele Funktionen bereitstellen. Sie bieten beispielsweise die Möglichkeit, 3D-Objekte in einer Szene zu platzieren, ihnen Texturen und Materialien zuzuordnen und sie mit Animationen zu bewegen. Zusätzlich bieten Spiele-Engines auch eine eigene Physiksimulation, was eine wichtige Grundlage für VR-Anwendungen ist, da eine gute Physiksimulation für die Plausibilitätsillusion benötigt wird.

Darüber hinaus bieten Spiele-Engines auch viele vorgefertigte Funktionen wie Partikeleffekte, Audioquellen und viele weitere. Außerdem bieten sie die Funktionalitäten, welche für VR benötigt werden, wie das Berechnen der Bilder für HMD, das Tracking von VR-Komponenten und sonstige Unterstützung der VR-Geräte. Die beiden bekanntesten Spiele-Engines sind *Unreal* und *Unity*.

### 3.2.1. Unreal Engine

Die *Unreal Engine* ist eine Spiele-Engine von *Epic Games*, welche erstmals 1998 veröffentlicht worden ist. Insgesamt sind bisher vier Generationen der *Unreal Engine* veröffentlicht worden, über welche die Funktionalitäten der Spiele-Engine erweitert worden sind.

Die *Unreal Engine* ist für die Entwicklung von Videospielen bekannt und wird hauptsächlich in diesem Bereich benutzt. Im Vergleich zu *Unity* kann die *Unreal Engine* vor allem mit einer besseren Graphik punkten, weswegen sie auch häufiger von großen Spieleentwicklern benutzt wird. Im *Marketplace* der *Unreal Engine* gibt es viele hochwertige 3D-Modelle und andere Assets, welche kostenfrei benutzt werden können.

In der *Unreal Engine* gibt es zwei wichtige Ansätze zum Programmieren. Es kann entweder in der Programmiersprache C++ oder mithilfe des sogenannten **Blueprints Visual Scripting** [Gam] entwickelt werden. **Blueprints** sind ein System, dass in der vierten Generation der *Unreal Engine* eingeführt worden ist. Diese erlauben es, viele Funktionalitäten zu implementieren, ohne Code schreiben zu müssen. Allerdings ist es aufwendig, diese **Blueprints** abzuändern oder eigene **Blueprints** zu entwerfen, weswegen die mit **Blueprints** umsetzbare Funktionalität begrenzt ist.

Neben den **Blueprints** gibt es auch noch die herkömmliche Programmierung in C++. Dies ist zwar aufwendiger als die Entwicklung mit **Blueprints**, allerdings gibt es weniger Einschränkungen, sodass komplexere Funktionalität implementiert werden kann.

### 3.2.2. Unity Engine

Eine weitere bekannte Spiele-Engine ist die *Unity Engine* [Unia]. Diese ist 2005 von dem Unternehmen *Unity Technologies* veröffentlicht und seitdem stets erweitert worden.

Im Vergleich zur *Unreal Engine* liefert *Unity* bei gleichem Aufwand weniger eindrucksvolle graphische Ergebnisse. Dafür wird die *Unity*-Oberfläche oft als übersichtlicher und benutzerfreundlicher empfunden [Str]. Auch in *Unity* gibt es viele Modelle, Skripts, Texturen und Ähnliches, welche im *Asset Store* erworben werden können.

In *Unity* wird das Programmieren mithilfe von Skripts umgesetzt, welche in C# geschrieben werden. Diese Skripts werden in der Regel auf Objekten in der Szene angebracht, um deren Verhalten zu bestimmen. In ihnen können aber auch andere Objekte oder Skripts referenziert und beeinflusst werden, wodurch komplexeres Verhalten ermöglicht wird.

### 3.2.3. Entscheidung für die Engine

Beide dieser Spiele-Engines haben Vorteile.

Zwar liefert die *Unreal Engine* in der Regel Anwendungen mit ansprechenderer Graphik, diese kann aber auch in *Unity* erreicht werden [Str]. Es sollte allerdings beachtet werden, dass durch bessere Graphik auch ein höherer Rechenaufwand entsteht. Da Virtual Reality zurzeit die Hardware stark beansprucht, kann dies dazu führen, dass das Programm ruckelt, was die Immersion stören würde. In dieser Arbeit wird daher auf besonders realistische Graphik verzichtet, weshalb dieser Vorteil von *Unreal* nicht wirklich ausgenutzt werden kann.

Sowohl *Unreal Engine* als auch *Unity* stellen eine kostenfreie Version bereit, sodass auch hier keine der Spiele-Engines einen Vorteil bietet. Die Entscheidung ist letztendlich auf die *Unity Engine* gefallen, da mit *Unity* bereits weitreichende Vorkenntnisse bestanden haben.

### 3.2.4. Unity-Grundlagen

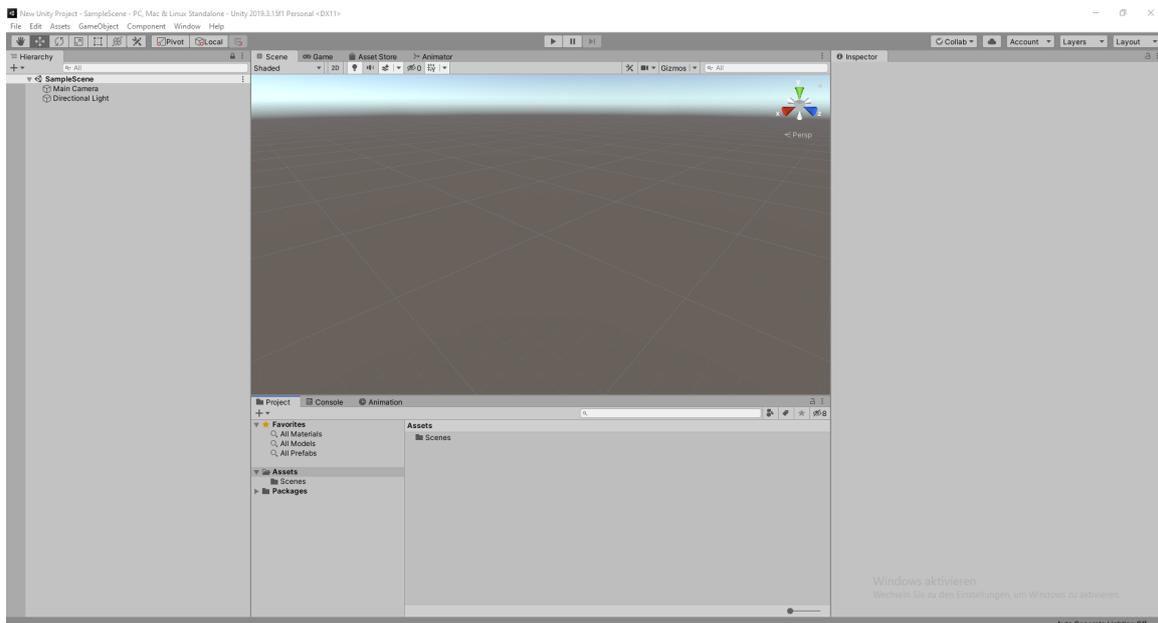


Abbildung 6: Übersicht des *Unity* Interfaces

Im Folgenden werden einige Grundbegriffe der Entwicklung mit der *Unity Engine* erklärt, die später benutzt werden. Eine Übersicht des *Unity* Interfaces kann in Abbildung 6 gesehen werden.

#### GameObject

**GameObjects** [Unib] in *Unity* sind die Grundklasse aller Objekte, welche in der Szene platziert werden. Alle **GameObjects** werden hierarchisch angeordnet. Veränderungen am Oberobjekt (Parent) beeinflussen auch die Unterobjekte (Children). Sie können aktiviert und deaktiviert werden, wodurch auch die Unterobjekte aktiviert oder deaktiviert werden. Des Weiteren haben sie ein **Transform** und können in der Szene sichtbar sein, wenn sie ein **Mesh** und Texturen haben. Außerdem können Skripts und andere Komponenten auf ihnen angebracht werden, um ihnen zusätzliche Funktionalitäten zu geben.

#### Empty

Ein **Empty** ist ein **GameObject** ohne **Mesh**, wodurch es keine sichtbare Form hat. Es hat lediglich ein **Transform**, ihm können aber auch andere Komponenten zugewiesen werden. **Empties** werden meistens dazu benutzt, Hierarchien zu organisieren oder Funktionalitäten zu implementieren, da sie keine nicht benötigten Komponenten besitzen.

## **MonoBehaviour**

`MonoBehaviour` ist die Grundklasse von Skripten in *Unity*. Nur Klassen, welche von `MonoBehaviour` erben, können Objekten zugewiesen werden.

### **Start()**

Die `Start`-Methode [Unid] wird am Anfang des Programms aufgerufen. Hier können Variablen initialisiert werden. Die Methode wird erst ausgeführt, nachdem die `Awake`-Methode aller Objekte ausgeführt worden ist.

### **Update()**

Nachdem die `Start`-Methode aller Skripts ausgeführt worden ist, wird für jeden Frame die `Update`-Methode aller Skripts ausgeführt. Da sie auf leistungsfähigeren Rechnern öfters ausgeführt wird, ist es wichtig, sie unabhängig von der Framerate zu implementieren.

## **Rigidbody**

Ein `Rigidbody` [Unid] (Starrkörper) wird in *Unity* Objekten zugewiesen, bei welchen die Physiksimulation benötigt wird. Dort kann beispielsweise mit dem `Boolean useGravity` eingestellt werden, ob das Objekt von der Schwerkraft beeinflusst wird. Ebenso bestimmt der `Boolean isKinematic`, ob die Bewegung des Objekts nur von der Bewegung des Elternobjekts abhängt. Zusätzlich können Masse und Widerstand angegeben werden.

## **Collider**

`Collider` werden in *Unity* benutzt, um Kollisionen zwischen Objekten zu detektieren und zu behandeln. `Collider` können dabei verschiedene Formen haben: `BoxCollider` mit Würfelform, `SphereCollider` mit Kugelform und `CapsuleCollider` mit Kapselform. Außerdem gibt es `MeshCollider`, welche die Form des Objekts annehmen oder konvex approximieren. Diese benötigen in der Regel allerdings mehr Rechenaufwand. `Collider` rufen in allen Skripten des Objekts die Methoden `OnCollisionEnter` und `OnCollisionExit` auf, wenn eine Kollision auftritt beziehungsweise aufhört, sodass im Code noch eine eigene Behandlung von Kollisionen eingeführt werden kann. Wenn bei `Collidern` die Option `isTrigger` angewählt ist, werden an ihrer Stelle die Methoden `OnTriggerEnter` und `OnTriggerExit` aufgerufen und es wird keine Kollisionbehandlung ausgeführt.

## Panel

Ein `Panel` ist eine UI-Komponente in *Unity*, welche benutzt wird, um Flächen der Benutzeroberfläche zu organisieren. Es besteht aus den zwei Komponenten `RectTransform` und `Image`. Das `RectTransform` bestimmt die Position auf dem Bildschirm, während das `Image` den Hintergrund für die angegebene Fläche bestimmt.

## Button

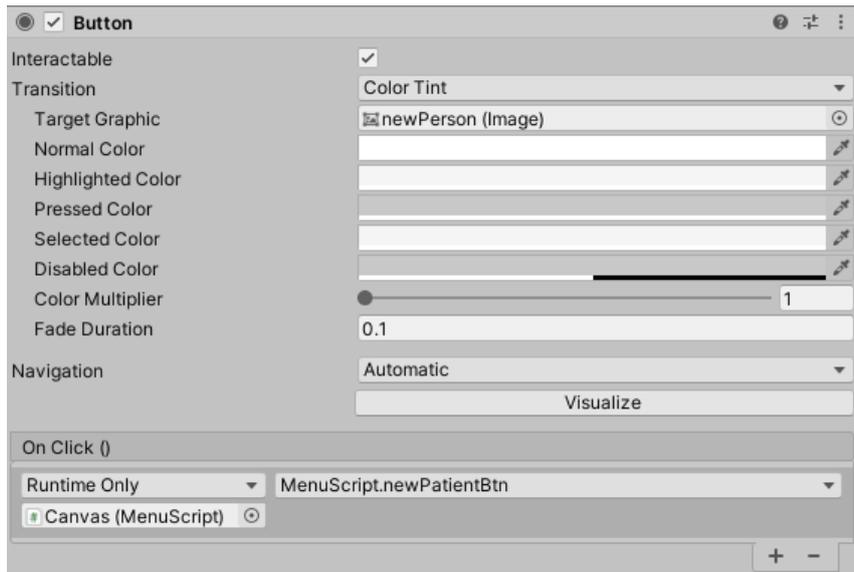


Abbildung 7: Zuordnen einer Methode zu einem Event in *Unity*

Knöpfe werden von *Unity* bereitgestellt und bieten bereits die Funktionalität, Methoden auszuführen, wenn sie angeklickt werden. Dazu werden diese Methoden dem `onClick`-Event des `Buttons` zugeordnet, wie in Abbildung 7 dargestellt. Neben dem `onClick`-Event gibt es noch weitere solche Events in *Unity*. Da diese aber in dieser Arbeit nicht eingesetzt werden, werden sie hier nicht näher erläutert.

## 3.3. Benutzte Software

### 3.3.1. SteamVR

Als Schnittstelle der verschiedenen VR-Geräte wird die *SteamVR*-Software benutzt. *SteamVR* erlaubt es, den betretbaren VR-Bereich festzulegen, und übernimmt die Eingabe und das Tracking der Geräte. Zusammen mit dem *SteamVR*-Asset in *Unity* werden die Position, Rotation und die Geräteeingabe der VR-Geräte an *Unity* übertragen. Das Konfigurieren der Geräte wird im Kapitel 6 genauer erklärt.

### 3.3.2. Blender

Blender [Ble] ist eine Open-Source-Software, mit welcher unter anderem 3D-Modelle erstellt und mit einem **Rigging** versehen werden können. **Rigging** bedeutet, dass ein Objekt mit Knochen versehen wird, welche es ermöglichen, das Objekt zu bewegen. Dies wird zum Beispiel bei menschlichen Modellen benutzt, damit diese animiert werden können. Diese Animationen können ebenfalls in Blender erstellt werden.

### 3.3.3. MakeHuman

Zum Erstellen der Menschenmodelle ist in diesem Projekt die Software „MakeHuman“ [Mak] verwendet worden. Diese Open-Source-Software bietet die Möglichkeit, menschliche Modelle zu erstellen und zu exportieren, ohne sie selbst modellieren zu müssen. Mithilfe von Schiebereglern können Alter, Körpergröße, Gewicht und Geschlecht eingestellt werden. Außerdem kann die Rasse mithilfe von drei Schiebereglern angegeben werden.

Zusätzlich ist es möglich, eine Frisur auszuwählen und dem Modell Kleidung zu geben. Hier bietet *MakeHuman* eine kleine Auswahl, welche mithilfe von Assets erweitert werden kann.

Letztlich kann das Modell mit einem **Rigging** versehen und exportiert werden. Für das **Rigging** wird die Option „Game Engine“ gewählt, da diese dafür sorgt, dass das **Rigging** bereits in dem von *Unity* benötigten Format ist. Danach wird das Modell als FBX-Datei exportiert.

Für dieses Projekt wurden 15 Modelle erstellt: Fünf männliche, fünf weibliche und fünf diverse, welches durch das Mitteln des Schiebereglers für Geschlecht erreicht worden ist. Die fünf Modelle pro Geschlecht unterscheiden sich durch verschiedene Gewichte. Die Körpergröße aller Modelle ist auf 1,60m festgelegt. Durch die einheitliche Modellgröße ist es möglich, die Modelle durch Skalierung in *Unity* auf die Körpergröße des Benutzers anzupassen, wodurch das Body Tracking verbessert wird.

### 3.3.4. Andere Modelle und Texturen

Andere Modelle, wie beispielsweise Tische, Betten oder andere Dekorationen, wurden von der Webseite *Sketchfab* heruntergeladen. *Sketchfab* ist eine Webseite, auf welcher viele 3D-Modelle bereitgestellt werden und teilweise kostenlos heruntergeladen werden können. Neben *Sketchfab* wurde auch die Webseite *3Dxo* verwendet, auf welcher Texturen erhältlich sind.

## 4. State of the Art

Da es zum aktuellen Zeitpunkt noch keine VR-Anwendung zum Training von Stomapatienten gibt, werden im Folgenden das Training von Stomapatienten und der Einsatz von VR in der Medizin unabhängig voneinander betrachtet.

### 4.1. Konventionelles Stomatraining

Die Vorbereitung der Stomapatienten ist von Krankenhaus zu Krankenhaus unterschiedlich. In der Regel führen Ärzte oder Krankenschwestern den Patienten die Abläufe zur Stomaversorgung mehrfach vor. Darüber hinaus bieten viele Krankenhäuser regelmäßige Sprechstunden an, in welchen Fragen rund ums Stoma geklärt werden können. Ebenso gibt es die Möglichkeit, sich auch online zu informieren und beraten zu lassen.

Zusätzlich zum Vorzeigen der Abläufe gibt es auch die Möglichkeit, diese durch ein sogenanntes „Skills Lab“ zu erlernen. Bei einem Skills Lab werden spezielle Simulatoren verwendet, an welchen praktische Fertigkeiten erlernt oder vertieft werden können. Beispielsweise können besondere Körpermodelle verwendet werden, um Ärzten die Möglichkeit zu geben, Praxiserfahrung zu sammeln, ohne dafür an echten Patienten trainieren zu müssen. Diese Modelle haben im Vergleich zu echten Patienten den Vorteil, dass die Bedingungen für alle Lernenden gleich sind und beliebig oft reproduziert werden können [Fic13]. Daher werden sie vor allem bei der studentischen Ausbildung im medizinischen Bereich verwendet [Nik+05].



Abbildung 8: Stomapflege-Simulator von „Anatomy Online“ [Onl]

In Abbildung 8 wird ein Unterkörpermodell (Puppe) gezeigt, an welchem zwei verschiedene

Stomamodelle befestigt sind. An einem solchen Modell könnten nicht nur Ärzte, sondern auch Patienten trainieren. Allerdings sind diese Modelle meist sehr teuer, weswegen sie nur zur Ausbildung von Ärzten und Krankenhauspersonal verwendet werden, nicht aber für Patienten.

## 4.2. VR in medizinischen Anwendungen

Obwohl es bislang noch keine Anwendung zum Stomatraining in der virtuellen Realität gibt, wird VR dennoch in vielen anderen medizinischen Bereichen benutzt. Hier wird meist die im Vergleich zu regulären PC-Anwendungen erhöhte Immersion ausgenutzt.

### Therapie in VR

Als erstes werden therapeutische Anwendungen in VR vorgestellt. Hierzu zählen Anwendungen wie *ANTARES* [Ant]. Das Projekt der Universität Siegen benutzt die virtuelle Realität, um Menschen bei der Rauchentwöhnung zu unterstützen. Das Prinzip der Anwendung basiert auf dem Ansatz des „Approach-Avoidance Tasks“ (AAT), bei welchem der Nutzer positive Objekte oder Bilder zu sich heranziehen und negative von sich wegdrücken soll. Dieses Verfahren wurde zuvor mithilfe eines Joysticks durchgeführt, hatte dort aber hohe Abbruchraten. Durch die intuitivere Steuerung in VR sowie die erhöhte Immersion und den Einsatz von Gamification soll diese verringert werden.

Ein weiteres Beispiel für eine Therapie in VR ist von dem Unternehmen *ANDERS VR* [VR] aus Stuttgart entwickelt worden. Ziel ihrer Anwendung ist es, den Krankenhausaufenthalt der Patienten angenehmer zu gestalten. Lange Krankenhausaufenthalte, bei welchen Patienten das Krankenhaus oder sogar das Bett nicht verlassen können, können zu einer Isolation von der Außenwelt und damit zu psychischen Schäden führen. Mithilfe der Anwendung können Patienten in virtuelle Welten eintauchen, welche Abwechslung zum Krankenzimmer schaffen. In einer App können verschiedene Naturszenarien wie beispielsweise ein Wald oder eine Wiese für die VR-Anwendung ausgewählt werden. Diese virtuellen Szenen können zur Entspannung und Abwechslung dienen, es gibt aber auch Atemübungen oder leichte Bewegungsübungen, welche weiteren Begleiterscheinungen langer Krankenhausaufenthalte wie beispielsweise Muskelschwund entgegenwirken können. Die Anwendung soll die Patientenbetreuung in Krankenhäusern ergänzen und die negativen Folgen langer Krankenhausaufenthalte verringern. Außerdem soll die Behandlungszeit verkürzt und der Behandlungserfolg verbessert werden.

Die virtuelle Realität kann also verwendet werden, um Patienten von ihrer realen Situation abzulenken. Dies kann bei Patienten aber nicht nur für Abwechslung sorgen, sondern kann

auch zur Schmerztherapie verwendet werden. Wenn Patienten sich in einer virtuellen Welt befinden, wenden sie ihre Konzentration eher auf die virtuelle Umgebung und weniger auf den Schmerz. Dies ist beispielsweise bei der Wundversorgung eines Brandopfers verwendet worden [Hof+04]. Hierbei hat der Patient sich während der Wundenversorgung mit Fingerbewegungen durch eine virtuelle Schneewelt bewegen können. Er hat zuerst drei Minuten in VR verbracht, für die folgenden drei Minuten der Wundversorgung ist keine Ablenkung bereitgestellt worden. In den Pausen nach diesen Phasen hat der Patient Angaben zu den verspürten Schmerzen gemacht. Insgesamt hat er bis zu 50% weniger Schmerz verspürt, während er in die virtuelle Welt vertieft gewesen ist. Während diese Studie vielversprechende Ergebnisse geliefert hat, muss beachtet werden, dass nur ein Patient an der Studie teilgenommen hat und die VR-Phase der Studie nur drei Minuten lang gewesen ist.

Neben diesen Bereichen kann VR auch zur Therapie von Phobien [WKM19], wie beispielsweise bei einer Arachnophobie (Angst vor Spinnen) [Lin+20; Bou+06; Alm+18] oder bei Akrophobie (Höhenangst) [Coe+09], verwendet werden.

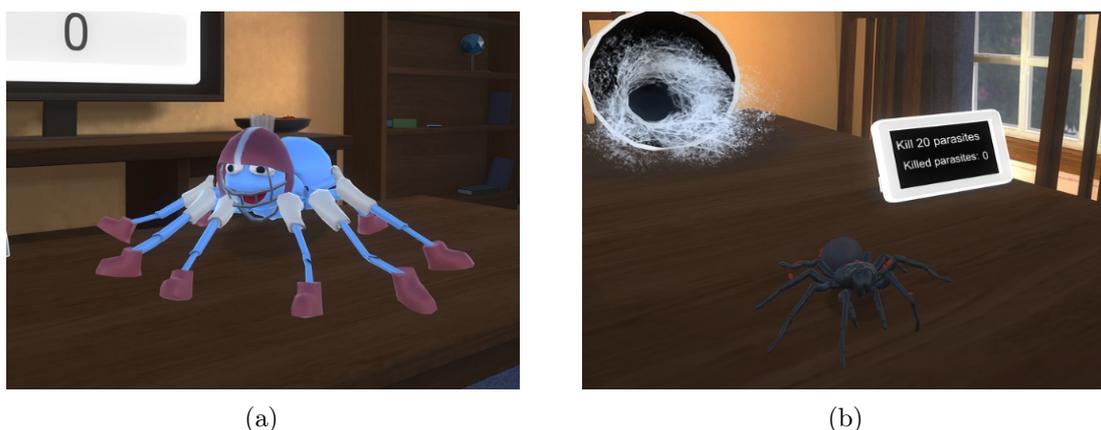


Abbildung 9: Anwendung „Itsy“: (a) Zeigt die Spinne zu Anfang der Anwendung  
(b) Realistischere Spinne des späteren Anwendungsverlaufs

Eine Studie zur Arachnophobie [Lin+20] hat dazu die Anwendung „Itsy“ [mim] verwendet. Diese Anwendung besteht aus mehreren Levels, in welchen der Anwender einer Spinne bei verschiedenen Aufgaben helfen muss. Zu Beginn der Anwendung wird eine sehr unrealistische Spinne gezeigt, welche mit jedem Level realistischer wird. Dadurch sollen die Patienten langsam an die Spinnen gewöhnt werden. Auch diese Studie hat vielversprechende Ergebnisse geliefert. Das Verwenden einer solchen Anwendung zur Therapie hat mehrere Vorteile. So kann auch Praxen die Therapie von Arachnophobie ermöglicht

werden, denen die artgerechte Haltung von Spinnen nicht möglich ist. Allerdings muss auch bei dieser Studie beachtet werden, dass es nur eine kleine Menge an Testpersonen gegeben hat.

### **Training in VR**

Ähnlich wie bei der Therapie kann VR auch zum Training verwendet werden. Dieses Training ist durch mehrere Studien als wirksam eingestuft worden [Sey08; Gun+18]. Die virtuelle Realität dient dabei nicht nur dazu, einen Ablauf zu lernen, sondern kann auch die Motivation der Lernenden fördern [Man+03]. Dies kann durch verschiedene Gamification-Elemente in den VR-Anwendungen erreicht werden. Neben diesen Aspekten hat VR den Vorteil, dass die Abläufe beliebig oft unter den gleichen Bedingungen wiederholt werden können.

Ein Beispiel hierfür ist die Arthroskopie [Mül+00]. Hierbei werden Instrumente über zwei Millimeter große Zugänge in das Knie des Patienten eingeführt. Eines der Instrumente dient zur Videoaufnahme und Beleuchtung, das andere ist ein Tasthaken, mit welchem abgetastet wird. Bei dieser Technik gibt es also keinen direkten Blick auf das Geschehen, sondern nur den indirekten Weg über die eingeführte Kamera. Das bedeutet, dass Chirurgen ihr eigenes Handeln nur über den Bildschirm verfolgen können. Damit Chirurgen dieses Verfahren lernen, hat es bisher zwei Trainingsmethoden gegeben: Das Lernen an echten Menschen unter Leitung eines erfahrenen Chirurgen sowie das Lernen mit Skills Labs. Das Lernen an Menschen hat den Nachteil, dass Fehler große Schäden verursachen können und das Training nur möglich ist, wenn auch Patienten zur Verfügung stehen. Obwohl diese Nachteile bei Modellen nicht mehr auftreten, sind diese oftmals sehr teuer und können nicht beliebig oft verwendet werden.

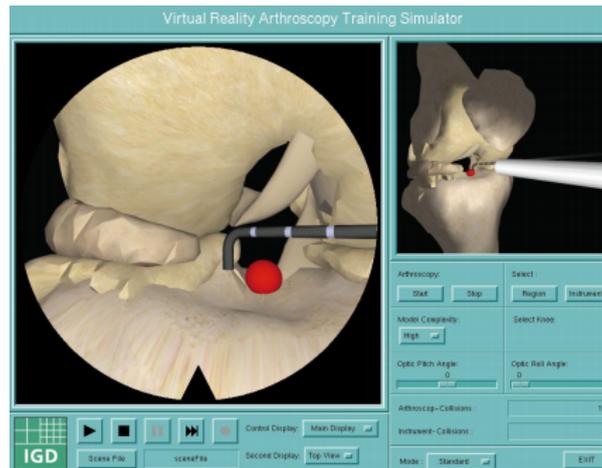


Abbildung 10: Graphische Benutzeroberfläche der Arthroskopie-Anwendung [Mül+00]

Um diese Probleme zu lösen und das Training um eine VR-Anwendung zu erweitern, ist von der Berufsgenossenschaftlichen Unfallklinik Frankfurt am Main und dem Fraunhofer Institut für graphische Datenverarbeitung eine VR-Anwendung entworfen worden, mit der Arthroskopie in VR gelernt werden kann. Die Benutzeroberflächen dieser Anwendung kann in Abbildung 10 gesehen werden. Dafür ist zuerst das dreidimensionale Modell eines Kniegelenks generiert worden, welches alle wichtigen Strukturen für das Training enthält. Zusätzlich ist ein Tracking-System für die Instrumente umgesetzt worden, sodass die Bewegung der Instrumente auch in die virtuelle Welt übertragen werden kann. Verschiedene Teile des Kniemodells sind so umgesetzt worden, dass sie von den virtuellen Instrumenten verformt oder bewegt werden können. Wenn der Tasthaken andere Objekte berührt, wird ein Ton abgespielt, um den Benutzer auf die Berührung aufmerksam zu machen.

Durch die Anwendung kann der Vorgang geübt werden, ohne Patienten zu gefährden oder teure Modelle zu benötigen. Jede der Sitzungen kann aufgenommen und automatisch bewertet werden. Dadurch muss auch nicht immer ein erfahrener Chirurg dabei sein.

Neben der Arthroskopie ist VR auch zum Training von Studenten der Radiologie verwendet worden [Gun+18]. Dazu haben die Studenten zuerst theoretisches Training in Form von Vorlesungen erhalten. Danach wurden sie in zwei Gruppen geteilt und es sind zwei Abläufe gewählt worden, welche die Studenten lernen sollen. Die erste Gruppe erhielt für den ersten Ablauf VR-Training und für den zweiten Ablauf konventionelles Training. Bei der zweiten Gruppe ist es anders herum gewesen. In dieser Studie hat sich herausgestellt, dass das Training in der virtuellen Realität mindestens so gute Ergebnisse liefert wie das

konventionelle Training. Zusätzlich gab es keinen Unterschied in der Leistung zwischen Testpersonen, welche bereits Erfahrungen mit Videospielen haben und den restlichen.

Ein weiteres Beispiel ist die Laparoskopie. In einer Analyse wurde das Training mithilfe von VR mit konventionellen Trainings verglichen [Por+20]. Dazu wurden die Ergebnisse mehrerer Studien verglichen. Dabei stellte sich heraus, dass die Operationszeit der Gruppe, die Training in VR hatte, kürzer war. Auch beim Umgang mit den Instrumenten lieferte diese Gruppe bessere Ergebnisse. Es muss allerdings beachtet werden, dass nicht bei allen dieser Studien angegeben worden ist, ob Testpersonen bereits vor der Studie VR-Training erhalten haben. Zusätzlich hatte das VR-Training die Einschränkung, dass kein haptisches Feedback verwendet worden ist. Dadurch kann nicht trainiert werden, wie viel Kraft benötigt wird, um die einzelnen Schritte auszuführen, ohne das Gewebe zu verwunden.

### **4.3. Forschungsbedarf**

In diesem Kapitel sind einige VR-Anwendungen beschrieben worden, welche Ärzten oder Medizinstudenten helfen verschiedene medizinische Abläufe zu erlernen. Zusätzlich sind Studien genannt worden, in welchen die virtuelle Realität zur Therapie verwendet wurde. Allerdings gibt es bisher kaum VR-Trainingsanwendungen für Patienten.

Da es bisher keine Möglichkeit für Stomapatienten gibt, die Stomaversorgung vor der Operation am eigenen Körper zu erlernen, soll in dieser Arbeit eine entsprechende Anwendung entwickelt werden. Diese wird anschließend in einer Pilotstudie auf ihre Usability überprüft. Die Durchführung einer Wirksamkeitsstudie ist im Rahmen dieser Arbeit aufgrund der Corona-Pandemie und der damit einhergehenden Einschränkungen nicht möglich, ist aber in der Zukunft geplant. Bei der Wirksamkeitsstudie sollen mit einer ausreichend großen Anzahl an Testpersonen die Wirkung und der Langzeiterfolg überprüft werden.

## 5. Konzeptentwicklung

Das Ziel dieser Arbeit ist es, in Kooperation mit der Allgemeinchirurgischen Klinik des Kreisklinikums Siegen [Sie], eine Anwendung zu entwickeln, welche Stomapatienten auf das Leben mit einem Stoma vorbereiten soll. Mithilfe dieser Anwendung soll den Patienten die Angst und Ungewissheit vor dem Stoma genommen werden. Zusätzlich sollen sie bereits vor der Operation lernen, wie sie ihr Stoma pflegen, sodass sie weniger ärztliche Betreuung benötigen. Hierzu sollen die Vorteile der virtuellen Realität ausgenutzt werden, um eine Trainingsanwendung zu erstellen.

### 5.1. Anforderungen

Anforderungen werden in drei Kategorien eingeteilt: Muss-, Soll-, und Kann-Anforderungen. Muss-Anforderungen sind die wichtigsten Aspekte der Anwendung, welche auf jeden Fall umgesetzt werden müssen. Soll-Anforderungen sind jene Eigenschaften, die nicht zwingend notwendig sind, aber dennoch große Vorteile für die Anwendung haben. Schließlich gibt es noch Kann-Anforderungen. Das sind Aspekte, welche die Anwendung erweitern würden, aber für die Grundfunktionalität nicht nötig sind.

#### **Muss-Anforderungen:**

- Umsetzbarkeit aller Trainingsschritte
- Angehörigenperspektive
- Patientenperspektive
- Bodytracking
- Verschiedene Patientenmodelle
- Klarer Aufbau der Anwendung
- Intuitive und immersive Gestaltung
- Erkennbarkeit aller verwendeten Objekte
- Eignung für ältere Personen
- Vorzeigen der Abläufe

- Prüfung auf Korrektheit der Schritte

#### **Soll-Anforderungen:**

- Speichern und Laden von Patientenprofilen
- Speichern der Ergebnisdaten

#### **Kann-Anforderungen:**

- Einbindung eigener Patientenmodelle

## **5.2. Konzept**

Da das Ziel der Anwendung die Vorbereitung der Patienten ist, sollen die einzelnen Abläufe in der virtuellen Realität nachgebaut werden. Dies soll den Benutzern die Möglichkeit bieten, die einzelnen Abläufe am eigenen Körper (Patientensicht) oder am Körper eines Angehörigen (Angehörigensicht) durchzuführen. Die Anwendung soll dabei in drei Abläufe und vier Trainingsphasen eingeteilt werden, welche im Folgenden beschrieben werden.

### **5.2.1. Gewählte Abläufe**

Wie bereits beschrieben, wird die Stomaversorgung in drei Abläufe eingeteilt: Leeren des Beutels, Wechseln des Beutels und Wechseln der Platte. Folgende Schritte müssen bei den Abläufen trainiert werden:

#### **Leeren des Beutels**

- Patient setzt sich verkehrt herum auf Toilette
- Beutel unten öffnen
- Beutel entleeren
- Verschluss mit Toilettenpapier reinigen
- Verschluss wieder verschließen

## Wechseln des Beutels

- Beutel an der Lasche packen und abziehen
- Beutel komplett entsorgen
- Neuen Beutel anbringen
- Neuen Beutel befestigen

## Wechseln der Platte

- Platte abziehen
- Kompresse befeuchten
- Stoma mit feuchter Kompresse reinigen
- Stoma mit einer trockenen Kompresse trocknen
- Platte zuschneiden
- Folie von der Platte abziehen
- Platte an die Haut kleben
- Beutel anbringen
- Beutel verschließen
- Platte 10 Minuten festhalten und mit Hand wärmen

### 5.2.2. Trainingsphasen

Die drei Abläufe werden jeweils in vier Phasen trainiert, welche im folgenden kurz erläutert werden. Die Trainingsphasen sind wie folgt:

- **Phase 1:** Es wird ein Video des gewählten Ablaufs abgespielt. Dazu ist die Versorgung eines Stomas in der Realität aufgezeichnet worden.
- **Phase 2:** Es wird eine Aufnahme der VR-Anwendung abgespielt. Zusätzlich wird zu jedem Zeitpunkt angezeigt, welcher Trainingsschritt gerade ausgeführt wird.

- **Phase 3:** Alle Aktionen werden angezeigt. Der Patient muss angeben, welche Aktion als nächstes ausgeführt werden soll.
- **Phase 4:** Die Patienten führen die Abläufe in VR selbstständig durch.

Die erste Trainingsphase soll dazu dienen, den Patienten den Ablauf beizubringen. Dazu wird lediglich die Videoaufnahme des Ablaufs mit einer erklärenden Tonspur abgespielt. In dieser Phase werden keine weiteren Informationen angezeigt und es wird keine Aktion auf Seiten der Patienten erfordert.

In der zweiten Trainingsphase sollen sich die Patienten die einzelnen Trainingsschritte besser einprägen. Dazu sollen neben dem Video zusätzlich die Trainingsschritte angezeigt werden, wobei der aktuell ausgeführte Trainingsschritt markiert wird. In dieser Phase soll die Videoaufnahme nur Bild in Bild innerhalb einer Bildschirmaufnahme der VR-Anwendung angezeigt werden. Dadurch sollen sich die Patienten nicht nur die Schritte einprägen, sondern diese bereits einmal in VR sehen, um ein Gefühl dafür zu bekommen, wie sie in VR umgesetzt sind. Auch hier wird die Tonspur verwendet, welche die Schritte erklärt.

Die dritte Trainingsphase soll die gleichen Videos wie die zweite Trainingsphase zeigen. In der dritten Phase sollen die Videos allerdings nicht automatisch abgespielt werden. Stattdessen soll der Patient nach jedem Trainingsschritt angeben müssen, welcher Schritt als nächstes ausgeführt werden soll. Da die Videos hier nur schrittweise abgespielt werden, wird hier keine Tonspur abgespielt.

In der vierten und letzten Trainingsphase sollen die Patienten die Abläufe schließlich selbstständig in der virtuellen Realität durchführen.

Das Festhalten der Platte soll in der Anwendung auf 30 Sekunden verkürzt werden, da das 10-minütige Festhalten der Platte nicht zwangsläufig einen erhöhten Lerneffekt hätte und die Motivation der Patienten belasten könnte. Die verbleibende Zeit soll mithilfe eines Timers angezeigt werden.

Nachdem die Patienten die Abläufe absolviert haben, sollen sie sich besser darauf vorbereitet fühlen, die Abläufe nach der Operation in der Realität selbst durchzuführen.

### **5.2.3. Grundstruktur der Anwendung**

Beim Starten der Anwendung soll ein Hauptmenü aufgerufen werden. In diesem sollen Patienten verwaltet werden können. Jeder Patient bekommt eine eindeutige Patienten-ID zugeordnet. Zusätzlich sollen das Geschlecht, die Körpergröße und das Gewicht angegeben werden. Diese werden benötigt, um die Modelle an die Patienten anzupassen.

Patientenprofile sollen angelegt, verändert oder gelöscht werden können. Nachdem der Patient ausgewählt worden ist, soll im Hauptmenü auch ausgewählt werden, ob der Ablauf aus der Patientensicht oder Angehörigensicht durchgeführt werden soll. Schließlich kann gewählt werden, welcher Ablauf trainiert und in welcher Phase das Training gestartet werden soll.

Die zu ladende Szene wird abhängig von dieser Auswahl bestimmt. Für die ersten drei Phasen des Trainings soll die sogenannte „Videoszene“ geladen werden. In der vierten Phase soll beim Wechseln der Platte die „Schlafzimmerszene“ und für das Wechseln und Leeren des Beutels jeweils die „Badezimmerszene“ geladen werden. Diese Szenen werden in Kapitel 5.2.4 genauer erläutert.

Nachdem der Ablauf abgeschlossen worden ist, sollen die dazugehörigen Daten gespeichert werden. Dazu zählen unter anderem der gewählte Patient, die benötigte Zeit, die gemachten Fehler und aus welcher Perspektive der Ablauf ausgeführt wurde.

#### **5.2.4. Szenen**

Für diese Anwendung sollen insgesamt vier Szenen erstellt werden. Diese Szenen sind: Hauptmenü, Video-, Badezimmer- und Schlafzimmerszene.

Im Hauptmenü sollen Patientenprofile verwaltet und ein Ablauf gewählt werden können. Das Menü wird in mehrere Fenster aufgeteilt. Im ersten Fenster sollen die Ärzte die Patienten verwalten können. Erst nachdem der richtige Patient ausgewählt oder erstellt worden ist, soll der Patient die Anwendung benutzen können. Dafür soll ein zweites Fenster erstellt werden, in welchem der Patient einen Ablauf und eine Trainingsphase auswählen kann. Nachdem ein Ablauf abgeschlossen worden ist, soll die Anwendung auch zu diesem Fenster zurückkehren und dabei die zuvor getroffene Auswahl erneut anzeigen. In der Videoszene werden die ersten drei Trainingsphasen aller Abläufe durchgeführt. Auf einem Großteil des Bildschirms soll dabei das Video angezeigt werden. Auf der rechten Seite sollen Knöpfe für die Trainingsschritte des Ablaufes angezeigt werden. In der zweiten Trainingsphase soll der aktuelle Trainingsschritt zusätzlich markiert werden. In der dritten Phase sollen die Knöpfe in einer zufälligen Reihenfolge angezeigt werden und der Patient soll den Knopf mit dem nächsten Schritt anklicken, bevor das entsprechende Video abgespielt wird.

In der vierten Trainingsphase sollen die drei Abläufe in zwei verschiedenen Räumen stattfinden. Das Leeren und Wechseln des Beutels soll in einem Badezimmer vor einer Toilette und das Wechseln der Platte in einem Schlafzimmer durchgeführt werden.

Im Badezimmer wird eine Toilette zum Leeren und eine Ablage zum Wechseln des Beutels

benötigt. Als Ablage wird ein kleiner Schrank verwendet, welcher so nahe an der Toilette platziert wird, dass der Benutzer nicht aufstehen muss, um den neuen Beutel zu greifen. Um die Präsenz in der Patientensicht zu erhöhen, soll zusätzlich ein Spiegel in der Szene platziert werden, in welchem Anwender das Spiegelbild ihres Modells betrachten können. Dadurch kann der Anwender direkt den Einfluss seiner Bewegungen auf das Modell beobachten, was die Involviertheit steigert. Außerdem könnte die Plausibilitätsillusion darunter leiden, wenn das Badezimmer keine spiegelnden Flächen enthält. Um die Szene weiter auszufüllen, können zusätzlich ein Waschbecken, eine Badewanne, ein Handtuchhalter und ein Badvorleger in der Szene positioniert werden. Schließlich sollen noch ein Fenster und eine Tür erstellt werden, um das Gefühl zu erwecken, dass das Badezimmer Teil eines größeren Gebäudes ist. In der Schlafzimmerszene werden ein Bett und eine Ablage benötigt. In der Angehörigensicht soll der Patient auf dem Bett liegen. Als Ablage wird ein Schreibtisch verwendet, auf welchem alle benötigten Gegenstände platziert werden. Dieser Schreibtisch soll nahe genug am Bett platziert werden, sodass beides durch das Tracking erreicht werden kann. Auch in dieser Szene wird ein Spiegel angebracht, welcher in der Patientensicht benutzt werden kann, um den Ablauf am eigenen Körper besser sehen zu können. Zusätzlich werden auch hier ein Fenster und eine Tür verwendet. Des Weiteren kann die Szene mit Schränken oder Regalen ausgefüllt werden, um sie lebendiger zu gestalten.

### **5.2.5. Immersion und gewählte Hardware**

Um den Lerneffekt zu maximieren, ist es wichtig, dass die Anwender einen möglichst hohen Grad an Immersion erfahren. Dies ist einer der Hauptgründe für das Benutzen der virtuellen Realität. In VR können Patienten die Stomaversorgung bereits am eigenen Körper durchführen, noch bevor die eigentliche Operation durchgeführt wurde. Daher soll viel Wert darauf gelegt werden, dass die Patienten sich bestmöglich mit den Menschenmodellen identifizieren können. Im Rahmen dieser Arbeit ist es nicht möglich, personalisierte Modelle der Patienten anzulegen. Stattdessen sollen 15 Menschenmodelle angelegt werden (5 männliche, 5 weibliche und 5 diverse), welche je einen verschiedenen Body-Mass-Index (BMI) besitzen. Zu Beginn der Anwendung soll mithilfe der Patientendaten das Modell ausgewählt werden, welches am besten zu dem Patienten passt. Daraufhin soll die Körpergröße des Modells angepasst werden, sodass sie mit der Körpergröße des Patienten übereinstimmt.

Um die Immersion weiter zu erhöhen, soll Bodytracking eingesetzt werden. Das Ziel des Bodytrackings ist es, dass sich das Modell in der virtuellen Welt mit den realen

Bewegungen des Patienten mitbewegt. Dies soll den Eindruck bei den Patienten erhöhen, am eigenen Körper zu arbeiten. Bodytracking soll immer dann benutzt werden, wenn die Anwendung aus der Patientensicht benutzt wird. Für die Angehörigensicht wird das Patientenmodell in der Szene platziert, sodass die Angehörigen die Abläufe an diesem trainieren können. Es wird also kein Körper für den Benutzer angezeigt.

Da die Anwendung hauptsächlich von älteren Personen benutzt wird, muss sie auf diese Personengruppe angepasst werden. Um die Handhabung für ältere Anwender möglichst angenehm zu gestalten und sie nicht zu verwirren, sollte die Steuerung intuitiv sein. Zusätzlich wurden die Größe der Platte und des Beutels angepasst, um die Handhabung zu erleichtern.

Für diese Anwendung ist die *HTC Vive Pro* als VR-Brille ausgewählt worden. Die *HTC Vive Pro* hat ihrem Vorgänger gegenüber den Vorteil, dass sie eine höhere Auflösung bietet. Allgemein ist diese VR-Brille kommerziell erhältlich, was den Vorteil hat, dass sie leicht zugänglich ist und es viel Unterstützung für Entwickler und Anwender gibt.

Die Interaktion mit der virtuellen Welt stellt einen zentralen Aspekt jeder VR-Anwendung dar, da sie einen großen Einfluss auf die Immersion hat. Daher wurde auf Geräte wie die *Leap Motion* verzichtet, da diese (wie in Kapitel 3.1.5 bereits genauer beschrieben) beim Überdecken der Hände oftmals zu Problemen führt, was zum Beispiel beim Öffnen der Beutellasche der Fall ist. Daher ist die *Leap Motion* für diese Anwendung ungeeignet.

Stattdessen wurden die *Vive Controller* benutzt. Diese können zwar nicht die Fingerbewegungen in die Anwendung übertragen, sind dafür aber zuverlässiger. Da die Controller nicht nur im Sichtfeld getracked werden, können die Controller immer angezeigt werden. Dies hat den Vorteil, dass das Bodytracking auch außerhalb des Sichtfeldes funktioniert. Dies ist beispielsweise für die Spiegelung notwendig, da dort das Spiegelbild der Hand gesehen werden kann, ohne dass sich die Hand selbst im Sichtfeld befindet. Außerdem können die Controller auch außerhalb des Sichtfeldes benutzt werden, um mit der Umwelt zu interagieren. Dies ist zum Beispiel sinnvoll, damit Gegenstände festgehalten werden können, während sich der Anwender im Raum umsieht. Durch ihre großen Knöpfe und einfache Handhabung sind sie außerdem sehr gut für ältere Anwender geeignet, welche keine Erfahrung mit VR-Anwendungen haben.

Im Rahmen dieser Arbeit ist auch geplant gewesen, Datenhandschuhe mit haptischem Feedback zu benutzen. Dies würde es den Patienten erlauben, die Objekte nicht nur zu sehen, sondern auch zu fühlen. Leider haben solche Datenhandschuhe nicht rechtzeitig bereitgestellt werden können, weswegen sie nicht in das Programm integriert werden konnten.

## 6. Implementierung

Ziel dieser Arbeit ist es, eine Anwendung zu entwickeln, welche die Anwender auf das Leben mit einem Stoma vorbereitet. Wie bereits in Kapitel 5.2.1 erläutert, wird die Stomaversorgung in drei Abläufe eingeteilt, welche in zwei verschiedenen Räumen erlernt werden.

Jeder dieser drei Abläufe wird außerdem in vier Phasen trainiert, welche bereits in Kapitel 5.2.2 beschrieben wurden. Zusätzlich müssen alle Abläufe sowohl aus der Sicht des Patienten, als auch aus der Sicht eines Angehörigen trainiert werden können.

Um dies zu ermöglichen, werden vier Szenen entworfen. Die erste ist ein Hauptmenü, welches dazu dient, die Anwendung zu steuern. Des Weiteren gibt es eine Videoszene, in welcher die ersten drei Trainingsphasen stattfinden. Letztlich gibt es die Badezimmerszene und die Schlafzimmerszene, in welchen die Anwender die Abläufe schließlich selbst in der virtuellen Realität ausführen.

### 6.1. Übersicht der Skripts

Im folgenden Abschnitt wird die Grundstruktur der verwendeten Skripts mithilfe von UML-Diagrammen erläutert. Zuerst werden in Abbildung 11 die Skripts des Hauptmenüs gezeigt.

#### 6.1.1. Skripts des Hauptmenüs

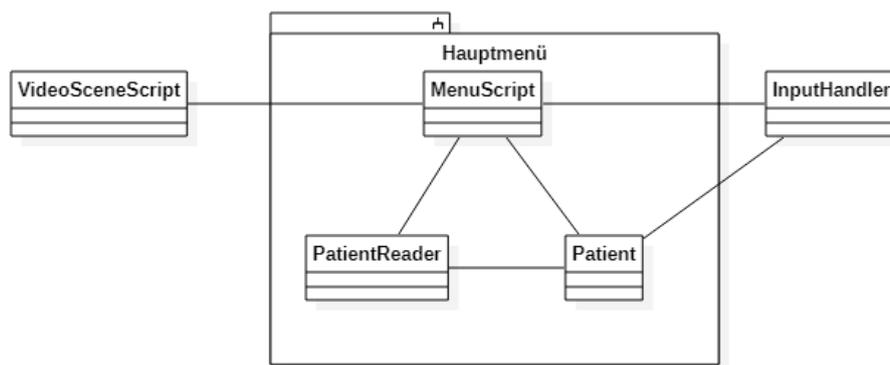


Abbildung 11: UML des Hauptmenüs

Das Hauptmenü besteht aus den Skripten `MenuScript`, `Patient` und `PatientReader`. Das `Patient`-Skript wird benutzt, um einen einzelnen Patienten zu beschreiben, so-

dass dieser verwaltet werden kann. Es beinhaltet die Patienten-ID, das Geschlecht, die Körpergröße, den BMI und eine Stomagröße. Zusätzlich besitzt die Klasse Methoden, um einen Patienten aus einem `string` einzulesen oder einen Patienten in einen `string` umzuwandeln.

Diese `strings` werden von der Klasse `PatientReader` verwendet, um die Patienten in einer Textdatei abzuspeichern, sodass sie nach erneutem Starten der Anwendung wieder geladen werden können.

Das `MenuScript` wird benutzt, um die Benutzeroberfläche zu steuern. Dort werden die Patientendaten angezeigt, neue Patienten angelegt oder existierende Daten verändert. Zusätzlich werden von diesem Skript aus die verschiedenen Trainingsszenen gestartet. Als Schnittstellen zu den verschiedenen Trainingsszenen dienen die Skripts `VideoSceneScript` und `InputHandler`. Das `VideoSceneScript` ist dabei für die ersten drei Trainingsphasen zuständig, während das Skript `InputHandler` die finale Trainingsphase steuert. Die Skripts werden im Abschnitt 6.4 beziehungsweise 6.6 weiter erläutert.

### 6.1.2. Skripts der Videoszene

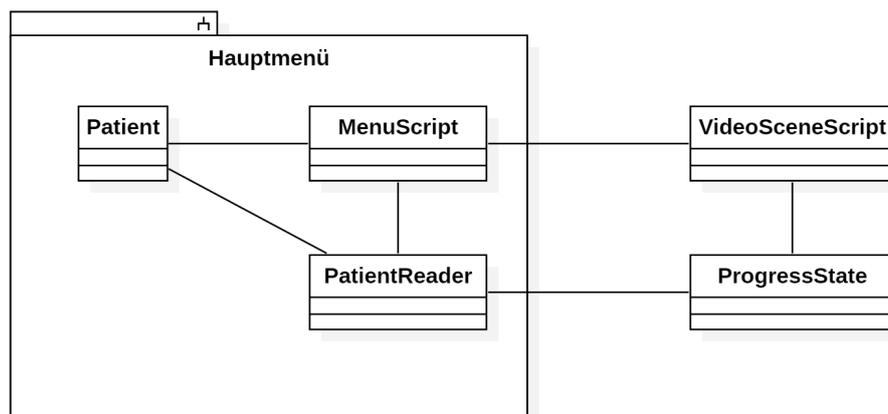


Abbildung 12: UML der Videoszene

In der Videoszene finden die ersten drei Phasen des Trainings statt. Die Videoszene wird durch das `VideoSceneScript` gesteuert. Wenn die Szene aufgerufen wird, werden die Einstellungen des Hauptmenüs aus dem `MenuScript` in das `VideoSceneScript` geladen. Außerdem werden die Skripts `ProgressState` und `PatientReader` verwendet, um die Trainingsergebnisse zu speichern.

In Abhängigkeit der Auswahl zwischen Patienten- und Angehörigensicht sowie des gewählten Ablaufs, werden die entsprechenden Videos in die Szene geladen. Für die zweite

und dritte Trainingsphase werden zusätzlich Knöpfe für die einzelnen Trainingschritte generiert.

Wenn ein Knopf gedrückt worden ist, soll farblich markiert werden, ob die Auswahl richtig oder falsch gewesen ist. Zusätzlich soll ein Ton abgespielt werden, um ein stärkeres Feedback zu liefern und dadurch den Lerneffekt zu steigern.

### 6.1.3. Skripts der VR-Szenen

Die Badezimmer- und Schlafzimmerszene werden für die vierte Trainingsphase verwendet. Da in diesen Szenen sehr viele Skripts verwendet werden, werden sie in mehrere Kategorien eingeteilt: Controllersteuerung, Szenensteuerung, Beutel-Skripts, Platten-Skripts und weitere Skripts.

#### Controllersteuerung

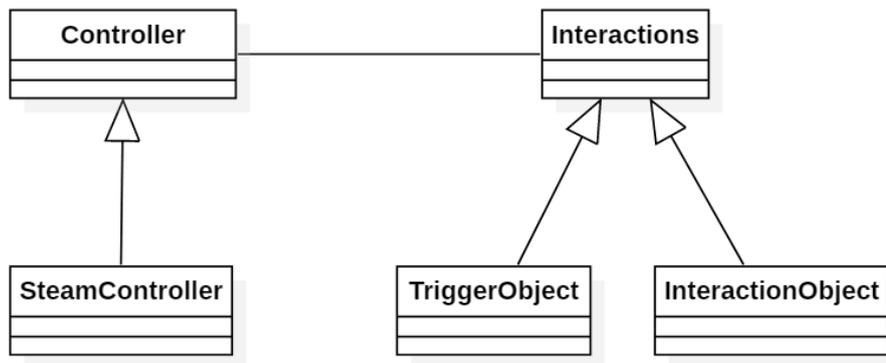


Abbildung 13: UML der Controllersteuerung

Die Controllersteuerung besteht aus zwei Grundklassen, welche in Abbildung 13 abgebildet sind. Die Grundklasse **Controller** stellt die Funktionalität für Controller bereit. Die Unterklasse **SteamController** ist auf die *Vive Controller* spezialisiert. Hier muss lediglich die Eingabe überschrieben werden. Dies bietet die Möglichkeit, weitere Controller oder Datenhandschuhe mit den gleichen Funktionalitäten einzubinden, ohne viel an dem Programm ändern zu müssen.

Zusätzlich zu den Skripts der Controller gibt es noch die Skripts der Objekte, mit welchen die Controller interagieren. Die Grundklasse dieser Objekte ist die Klasse **Interactions**. Von dieser Klasse erben die beiden Klassen **TriggerObject** und **InteractionObject**. **InteractionObjects** sind all solche Objekte, die mit den Controllern aufgehoben und bewegt werden können. **TriggerObjects** auf der anderen Seite sind jene Objekte, welche

nicht aufgehoben, sondern nur betätigt werden können. Dies wird zum Beispiel benutzt, um den Stomabeutel an der Platte festzudrücken.

## Szenensteuerung

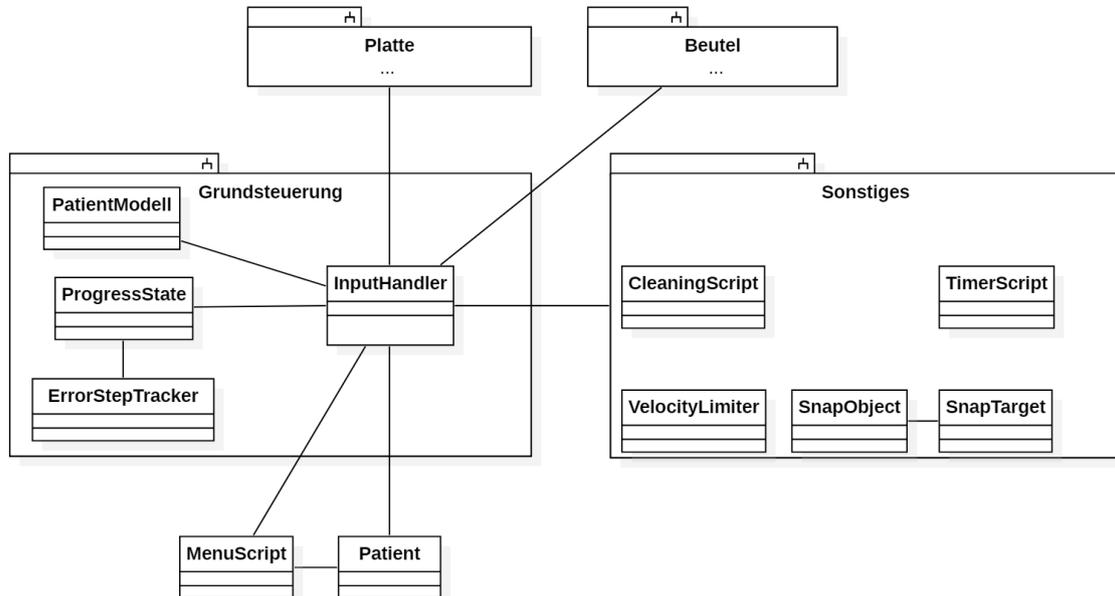


Abbildung 14: UML der VR-Szenen

Die Hauptkomponente der VR-Szenen ist das sogenannte **InputHandler**-Skript. Das Skript liest die Einstellungen aus dem Hauptmenü ein und aktiviert den entsprechenden Ablauf sowie das gewählte **PatientModell**. Außerdem regelt es die gesamte Kommunikation in der Szene. Es ist dafür verantwortlich, alle Eingaben zu bearbeiten, welche über den PC getätigt werden. So kann beispielsweise das Patientenmodell ausgetauscht oder der VR-Bereich verschoben werden.

Die einzelnen Patientenmodelle werden über Instanzen des **PatientModell**-Skripts gesteuert. Diese werden benutzt, um die Modelle zu aktivieren, beziehungsweise zu deaktivieren. Zusätzlich steuert es das Bodytracking und passt die Größe der Modelle an die Körpergröße des Patienten an.

Der Fortschritt des Ablaufes wird in dem **ProgressState**-Skript abgespeichert. In diesem Skript werden die Schritte des Ablaufes angegeben. Es kontrolliert, wann die einzelnen Schritte ausgeführt werden können, indem es Komponenten aktiviert und deaktiviert. Zusätzlich ist dieses Skript dafür verantwortlich, die Ergebnisse des Trainings abzuspeichern. Das letzte Skript der Szenensteuerung ist das **ErrorStepTracker**-Skript. Dieses wird von dem **ProgressState** angesprochen, um einen Erfolgston abzuspielen, wenn ein

Schritt erfolgreich ausgeführt worden ist. Ebenso wird ein Fehlerton abgespielt, wenn ein Fehler gemacht worden ist.

### **Beutel-Skripts**

Der Beutel wird über mehrere Skripts gesteuert. Die Hauptkomponente ist hierbei das `BagScript`, welches von der Klasse `InteractionObject` erbt. Diese Klasse steuert, wann der Beutel gegriffen und bewegt werden kann, und kontrolliert die `BagTrigger` und `BagSeal`-Skripts.

Sowohl `BagTrigger` als auch `BagSeal` erben von `TriggerObject`. `BagTrigger` wird benutzt, um das Befestigen des Beutels an der Platte zu steuern. Dazu soll der Anwender mit vier Objekten interagieren, um das Befestigen zu simulieren. Das `BagSeal`-Skript wird benötigt, um das Auf- und Zufalten des Verschlusses am Beutel zu ermöglichen. Der Anwender kann den Verschluss auf- beziehungsweise zufalten, indem abwechselnd beide Händen beziehungsweise Controller mit dem Verschluss interagieren. Dadurch sollen das Auf- und Zufalten realistischer als durch eine einzelne Interaktion gestaltet werden.

### **Platten-Skripts**

Ähnlich wie der Beutel, wird auch die Platte durch mehrere Skripts gesteuert. Diese Skripts sind: `PlateScript`, `FoilScript`, `CutPartManager`, `CutPart` und `ScissorScript`. Analog zum `BagScript` des Beutels, ist das `PlateScript` hierbei die Hauptkomponente und erbt von `InteractionObject`. Es entscheidet, wann die Platte bewegt werden darf. Das `FoilScript` wird ausschließlich dazu verwendet, die Folie von der Platte entfernen zu können. Die restlichen drei Skripts werden zum Schneiden der Platte verwendet. Das `ScissorScript` ist dafür zuständig, die Schere aufheben und ablegen zu können. Die Schere wird dazu benutzt, Objekte mit einem `CutPart`-Skript zu „zerschneiden“. Um die Platte zu schneiden, wird der zerschneidbare Bereich in viele Einzelobjekte geteilt, welche alle mit einem `CutPart`-Skript ausgestattet werden. In dem `CutPartManager` werden sie anschließend verwaltet. Die zerschneidbaren Objekte werden in drei Kategorien eingeteilt. Eine Kategorie sind die Objekte, welche geschnitten werden müssen. Eine weitere sind Objekte, welche auf keinen Fall geschnitten werden dürfen. Die dritte Kategorie sind Teile, welche geschnitten werden können, aber nicht müssen. Diese dritte Kategorie wird verwendet, um den Anwendern eine kleine Fehlertoleranz zu erlauben. Je nach eingestellter Stomagröße werden die schneidbaren Objekte den Kategorien zugeteilt, sodass bei einem größeren Stoma auch mehr ausgeschnitten werden muss.

## Weitere Skripts

Die restlichen fünf Skripts, welchen keine eigene Kategorie zugeordnet worden ist, sind: `SnapObject`, `SnapTarget`, `CleaningScript`, `VelocityLimiter` und `TimerScript`.

Die beiden Skripts `SnapObject` und `SnapTarget` werden verwendet, um zwei Objekte zusammenzufügen, wenn sie sich berühren oder nahe genug aneinander sind. Das Objekt mit dem `SnapObject`-Skript wird dabei an das Objekt mit dem `SnapTarget`-Skript gebunden. Diese Skripts werden verwendet, um die Platte an den Patienten und den Beutel an die Platte zu binden.

Das `CleaningScript` wird an mehreren Stellen der Anwendung verwendet. Dieses Skript kann benutzt werden, um ein Objekt mit einem anderen zu säubern und dies visuell anzuzeigen. Dieses Skript wird beispielsweise verwendet, wenn das Stoma mit den Kompressen gereinigt wird.

Das `TimerScript` wird verwendet, um einen Timer anzuzeigen, wenn die Platte nach einem Plattenwechsel mit der Hand gehalten und gewärmt wird. Das Skript `VelocityLimiter` ist in der Lage, die Bewegungs- und Rotationsgeschwindigkeit eines Objektes zu beschränken. Dies wird benötigt, um die Bewegung des Beutels zu kontrollieren.

## 6.2. Design der Szenen

Wie bereits in Kapitel 5.2.4 erläutert, wird die Anwendung in vier Szenen eingeteilt.

### 6.2.1. Design des Hauptmenüs

Im Hauptmenü wird eine Leiste mit den Logos aller Beteiligten angezeigt. Hierzu ist auch ein Logo für die Anwendung erstellt worden.

Als Hintergrund wird hier die Schlafzimmerszene verwendet, welche langsam rotiert wird. Darüber wird ein fast-transparentes Bild gelegt, um den Fokus weniger stark auf den Hintergrund zu lenken. Darauf werden die verschiedenen Hauptmenüfenster angezeigt.



Abbildung 15: Design des Hauptmenüs

In Abbildung 15 wird das Fenster zur Patientenauswahl dargestellt. Die anderen Fenster des Hauptmenüs werden in Kapitel 6.3 genauer beschrieben.

### 6.2.2. Design der Videoszene

Die Videoszene ist einem alten Fernseher nachempfunden worden, wie in Abbildung 16 ersichtlich ist. Auch hier wird die Schlafzimmerszene als Hintergrund verwendet. Allerdings wird sie in dieser Szene nicht mehr rotiert, sondern bleibt so stehen, wie sie beim Start der Szene im Hauptmenü gewesen ist.



Abbildung 16: Design der Videoszene

Auf dem Anzeigebereich (oben rechts) werden der gewählte Ablauf und die aktuelle Aufgabe angezeigt. Darunter werden Knöpfe angezeigt, welche die Schritte des Ablaufes angeben. Unter dem Video gibt es zusätzlich Knöpfe, mit welchen das Video pausiert oder der Ablauf abgebrochen werden kann.

### 6.2.3. Design der Badezimmerzene

Abbildung 17a zeigt die Toilette und die Ablage des Badezimmers. Diese sind nah aneinander platziert, sodass in der Patientensicht die Objekte auf der Ablage von der Toilette aus erreicht werden können. Neben den für die Abläufe benötigten Gegenstände sind noch weitere Objekte platziert, um die Szene auszufüllen. Diese können in Abbildung 17b gesehen werden. Durch das Fenster kann außerdem eine einfache Berggegend gesehen werden, welche den Eindruck erwecken soll, dass der Raum ein Teil einer größeren Welt ist.



(a)



(b)

Abbildung 17: Design der Badezimmerzene

#### 6.2.4. Design der Schlafzimmerszene

Die letzte Szene der Anwendung ist die Schlafzimmerszene. Wie in der Badezimmerszene sollen auch hier alle für die Abläufe benötigten Gegenstände durch Tracking erreichbar sein. Daher sind diese Objekte auf einem Schreibtisch platziert, welcher neben dem Bett steht.



(a)



(b)



(c)

Abbildung 18: Design der Schlafzimmerszene

Auch diese Szene ist durch weitere Gegenstände ergänzt. In Abbildung 18 wird die Schlafzimmerszene aus verschiedenen Perspektiven gezeigt.

### 6.3. Implementierung des Hauptmenüs

Wie in Kapitel 6.1.1 beschrieben, wird die Funktionalität des Hauptmenüs von den Skripten `Patient`, `PatientReader` und `MenuScript` übernommen. Das `Patient`-Skript wird benutzt, um alle anwendungsrelevanten Daten eines Patienten in einem Objekt zu speichern. Das `PatientReader`-Skript ist dafür zuständig, die `Patient`-Objekte in eine Textdatei zu schreiben und aus dieser wieder auszulesen. Die restliche Funktionalität des Hauptmenüs ist in `MenuScript` implementiert.

Beim Starten der Anwendung wird als erstes das Hauptmenü angezeigt. Dort werden einige Voreinstellungen gewählt. Neben dem Ablauf und der Trainingsphase wird auch ausgesucht, ob der Ablauf aus Patienten- oder Angehörigensicht trainiert werden soll. Das Hauptmenü wird in vier Fenster eingeteilt. Die ersten beiden Fenster dienen zur Patientenauswahl (Abbildung 19) und zum Anlegen oder Modifizieren von Patienten (Abbildung 20) und werden von dem Studienleiter bearbeitet. Das erste Fenster ist in Abbildung 19 zu sehen und dient dazu, die Patienten auszuwählen. In Abbildung 20 wird das zweite Fenster gezeigt, in welchem Patienten angelegt oder modifiziert werden können. Danach folgt das in Abbildung 21 abgebildete Fenster, in welchem die Anwender eine kurze Einleitung bekommen. Im Anschluss daran wird ein weiteres Fenster angezeigt, über welches der Ablauf gewählt werden kann. Dieses ist in Abbildung 22 zu sehen.

Jedes dieser vier Fenster ist das Unterobjekt von je einem eigenen `Panel`-Objekt. Dadurch ist es möglich, zwischen den Fenstern zu wechseln, indem das `Panel`-Objekt des gewünschten Fensters aktiviert und die `Panel`-Objekte der anderen Fenster deaktiviert werden. Dazu werden die Methoden `switchToPatientPickMenu`, `switchToPatientMenu`, `switchToWelcomeMenu` und `switchToMenu` verwendet, welche die Fenster in genannter Reihenfolge aktivieren. Dazu bekommen die Knöpfe des Hauptmenüs im Editor die entsprechende Methode zu ihrem `onClick`-Event zugeordnet.



Abbildung 19: Hauptmenü: Fenster zur Auswahl des Patienten

Das erste Fenster, welches beim Start der Anwendung aufgerufen wird, ist das Fenster zur Patientenauswahl.

Beim Start der Anwendung wird das in Abbildung 19 gezeigte Fenster aufgerufen, in welchem der Patient ausgewählt wird. Dabei werden alle vorhandenen Patienten in eine Dropdown-Liste geladen. Dazu wird in der `PatientReader`-Klasse die Methode `readAll` aufgerufen. Diese liest für jeden Patienten eine Zeile der Patientendatei ein. Diese Zeile kann wie folgt aussehen:

```
Name:MaxMustermann;BMI:18.5 - 24.9;StomaArt:Dünndarm;Geschlecht:Männlich;
KoerperGroesse:180;KoerperGewicht:70;#
```

Diese Zeilen werden an das `MenuScript` weitergegeben, wo für jede ein `Patient`-Objekt angelegt und in einer Liste gespeichert wird. Die Patienten-IDs werden in der Dropdown-Liste angezeigt, damit der Betreuer aus dieser den gewünschten Patienten auswählen kann. Mit „Auswahl bestätigen“ wird das in Abbildung 21 gezeigte Fenster zur Einleitung des Patienten aufgerufen. Mit den Knöpfen „Neuer Patient“ und „Patient modifizieren“ wird das in Abbildung 20 zu sehende Patientenfenster geöffnet.



Abbildung 20: Hauptmenü: Fenster zum Anlegen und Modifizieren von Patienten

In dem in Abbildung 20 zu sehenden Patientenfenster werden neue Patienten angelegt oder existierende Patienten angepasst. Hier können die gewünschte Patienten-ID, Körpergröße und Körpergewicht eingegeben und ein Geschlecht gewählt werden. Wenn ein Patient modifiziert werden soll, werden die bereits bestehenden Patientendaten aus dem gewählten `Patient`-Objekt eingelesen und in die einzelnen Felder eingetragen. Danach können die Daten angesehen und angepasst werden, falls sie im Voraus falsch eingetragen worden sind oder sich seitdem verändert haben. Wenn auf „Speichern“ geklickt wird, wird zuerst überprüft, ob die eingegebenen Patientendaten gültig sind. Das bedeutet, dass alle Felder ausgefüllt worden sind und die Patienten-ID eindeutig ist. Wenn ein Feld nicht ausgefüllt worden ist oder die Patienten-ID nicht eindeutig ist, werden diese Felder rot markiert, bis sie das nächste Mal ausgewählt werden. Wenn die Patientendaten gültig sind, wird ein neuer Patient erstellt oder der gewählte angepasst.

Dazu wird aus den eingegebenen Daten der BMI des Patienten berechnet, sodass das passende Menschenmodell für ihn ausgewählt wird. Danach werden die Patientendaten in die Textdatei geschrieben. Im Anschluss wird die aktualisierte Textdatei wieder eingelesen und die Patientenauswahl aktualisiert. Nachdem der Patient gespeichert worden ist, wird das Patientenauswahlfenster wieder geöffnet.

Mit „Löschen“ wird der gewählte Patient gelöscht. Dies kann nur dann geschehen, wenn zuvor „Patient modifizieren“ gewählt worden ist. Ansonsten ist der Knopf ausgegraut und es kann nicht mit ihm interagiert werden. Ist ein Patient gelöscht worden, wer-

den wie beim Speichern die Textdatei und die Patientenauswahl aktualisiert und das Patientenauswahlfenster wird geöffnet. Mit dem „Zurück“-Knopf wird das Fenster zur Patientenauswahl wieder aufgerufen, ohne, dass Änderungen gespeichert werden.



Abbildung 21: Hauptmenü: Einführungsfenster

Nachdem der Patient gewählt worden ist, wird ein kleiner Einleitungstext zu der Anwendung angezeigt, welcher in Abbildung 21 dargestellt ist. An dieser Stelle übergibt der Betreuer die Anwendung an den Patienten. Der Patient kann sich den Einführungstext durchlesen und danach auf „Weiter“ klicken.

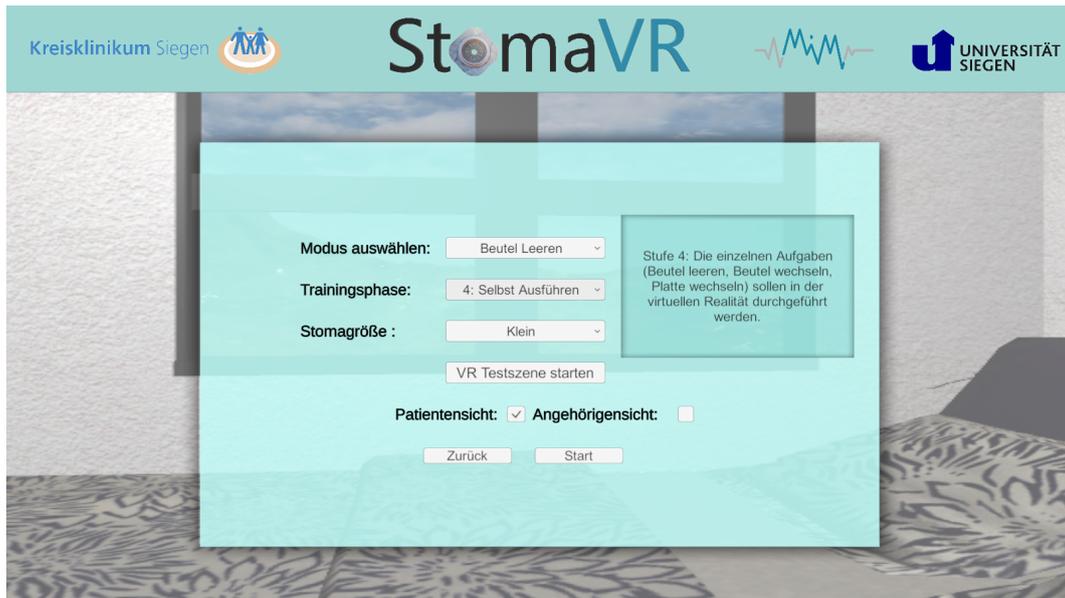


Abbildung 22: Hauptmenü: Fenster zum Auswählen des Ablaufs und der Trainingsphase

Daraufhin wird das in Abbildung 22 gezeigte Fenster geöffnet. Dort kann der Anwender einen der Abläufe aus Kapitel 5.2.1 und eine der Phasen aus Kapitel 5.2.2 auswählen. Um den Patienten die Anwendung zu erleichtern, wird zu der gewählten Phase eine kurze Beschreibung angezeigt. Wenn die vierte Phase ausgewählt ist, kann in diesem Fenster die Stomagröße gewählt werden. Ansonsten ist diese Auswahl ausgeblendet. Als Letztes kann der Anwender entscheiden, ob er den Ablauf aus Patienten- oder Angehörigensicht trainieren möchte. Wenn der „Start“-Knopf betätigt wird, werden alle relevanten Informationen in `static`-Variablen im `MenuScript` gespeichert und die gewählte Szene wird gestartet. Wenn eine der ersten drei Phasen ausgewählt ist, wird eine Videoszene gestartet. Ansonsten wird die entsprechende VR-Szene gestartet. Folgende `static`-Variablen werden gespeichert:

- `bool onSelf`: Gibt an, ob die Anwendung aus Patientensicht durchgeführt werden soll
- `int stomaSize`: Eine Zahl zwischen 0 und 2, welche die Stomagröße angibt
- `MainType mode`: Eine Variable vom Typ `enum` (Aufzählungstyp), welche den Ablauf angibt
- `int trainingPhase`: Eine Zahl zwischen 0 und 3, welche die Trainingsphase angibt
- `Patient startPatient`: Der gewählte Patient

Nachdem der Patient einen Ablauf abgeschlossen hat, gelangt er wieder zurück zu dem Fenster mit dem Einleitungstext. Nachdem er in diesem auf „Weiter“ geklickt hat, besteht die Möglichkeit, die Einstellungen zu ändern und den Ablauf anschließend erneut zu starten, ohne, dass der Betreuer erneut die Patientendaten auswählen muss.

#### 6.4. Implementierung der Videoszene

In der Videoszene finden die ersten drei Phasen aller Abläufe des Stomatrainings statt. Wie in Kapitel 5.2.2 beschrieben, werden in den ersten Trainingsphasen Videos der einzelnen Trainingsschritte des Ablaufes abgespielt.



Abbildung 23: Übersicht der Videoszene

In Abbildung 23 wird die Videoszene gezeigt. Auf der rechten Seite gibt es eine Anzeige mit einer kurzen Beschreibung der Aufgabe. Unter dieser werden die Schritte des Ablaufs aufgelistet. In dem `VideoSceneScript` wird für jeden Ablauf eine Liste mit Knöpfen angelegt. Zusätzlich werden 12 Listen mit `Videoclips` gespeichert. Für jeden der drei Abläufe werden Realaufnahmen und Bildschirmaufnahmen der VR-Anwendung sowohl aus Patientensicht als auch aus Angehörigensicht gespeichert. Bei den Bildschirmaufnahmen der VR-Anwendung wird zusätzlich in der unteren Ecke die Realaufnahme des gleichen Schrittes angezeigt, sodass Nutzer einen direkten Vergleich zwischen der Realität und der VR-Anwendung bekommen.

Beim Start der Szene wird aus dem Hauptmenü ausgelesen, welcher Ablauf, welche Phase

und welche Ansicht gewählt worden ist. Die gewählten Knöpfe werden zu Unterobjekten eines `Panel`s mit einem `VerticalLayoutGroup`-Skript. Das `VerticalLayoutGroup`-Skript verteilt die Unterobjekte gemäß ihrer Anordnung in der Hierarchie gleichmäßig in der angegebenen Fläche, welches für die Knöpfe mit den Trainingsschritten verwendet wird. Für die dritte Phase werden die gewählten Knöpfe außerdem zu Unterobjekten des `Panel`s mit zufälliger Anordnung, sodass sie nicht immer in der gleichen Reihenfolge angezeigt werden. Allerdings sind sie in der Liste im `VideoSceneScript` weiterhin in der richtigen Reihenfolge gespeichert.

Um herauszufinden, ob ein betätigter Knopf zu dem richtigen nächsten Trainingsschritt gehört, wird in dem `VideoSceneScript` ein `int currentStep` angelegt. Wenn die Listenposition des Knopfes mit dem in `currentStep` gespeicherten Wert übereinstimmt, ist der richtige Knopf ausgewählt worden. Um die Listenposition zu ermitteln, wird beim Betätigen eines Knopfes die Methode `buttonPressed` mit dem betätigten Knopf als Parameter aufgerufen. Das `VideoSceneScript` durchsucht daraufhin die Liste nach dem übergebenen Knopf und erhält dadurch seine Position in der Liste. Ist der gedrückte Knopf der nächste Schritt, wird der Knopf kurzzeitig grün markiert, das Video des Schrittes abgespielt und `currentStep` um eins hochgezählt. Ist der gewählte Knopf allerdings ein falscher Knopf, wird er kurzzeitig rot markiert und es ertönt ein Fehlerton, wie in Abbildung 24 zu sehen ist. Zusätzlich wird in dem `ProgressState`-Skript der Fehler in Textform gespeichert, sodass er beim Abschließen des Ablaufes in einer Textdatei gespeichert werden kann.

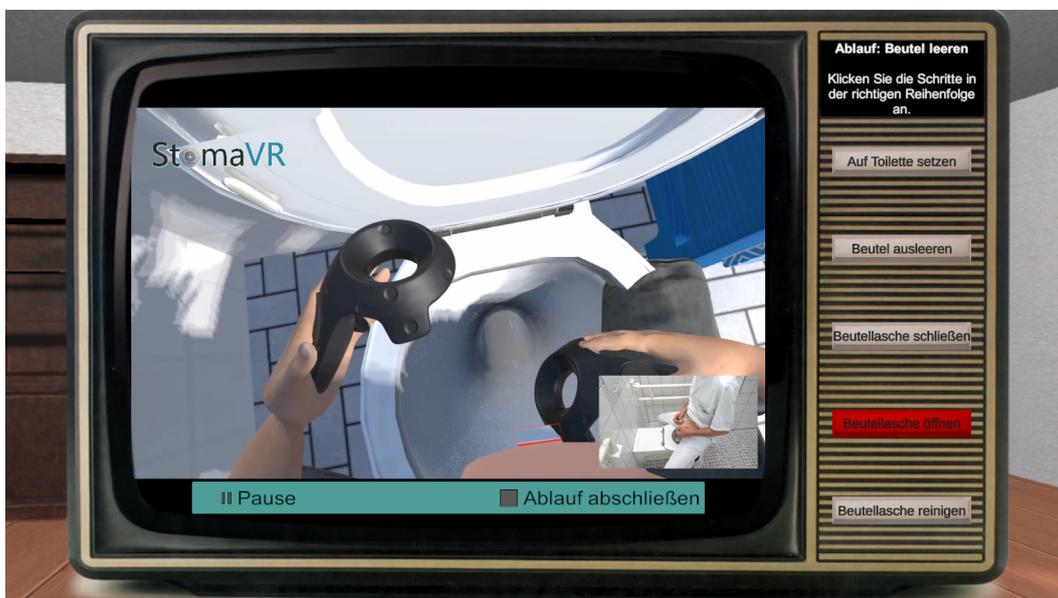


Abbildung 24: Videoszene Fehler

Die einzelnen Videos werden abgespielt, indem der gewählte VideoClip mithilfe eines VideoPlayers auf eine RenderTexture geladen wird. Diese kann wie eine gewöhnliche Textur in das Image des großen Panels auf der linken Seite aufgetragen werden.

Wenn der Nutzer den Ablauf vorzeitig beenden möchte, kann er den Knopf „Ablauf abschließen“ drücken. Dadurch wird ein Fenster aufgerufen, in welchem der Nutzer bestätigen muss, ob er den Ablauf wirklich beenden möchte. Dies wird in Abbildung 25 gezeigt.



Abbildung 25: Videoszene: Abfrage zum Abschließen des Ablaufs

Wenn der Nutzer sich dafür entscheidet, die Anwendung tatsächlich zu beenden, wird das in Abbildung 26 dargestellte Fenster geöffnet. In diesem Fenster werden alle Schritte angezeigt, welche der Nutzer vergessen hat.



Abbildung 26: Videoszene: Anzeige der vergessenen Schritte

An dieser Stelle hat der Nutzer keine Möglichkeit mehr, den aktuellen Ablauf weiter auszuführen, und muss mithilfe des Knopfes „Zurück zum Hauptmenü“ zum Hauptmenü zurückkehren. Von dort kann er einen neuen Ablauf starten.

Während eines der in Abbildung 25 und Abbildung 26 angezeigten Fenster zum Abschließen des Ablaufes sichtbar ist, können keine Knöpfe außerhalb des Fensters betätigt werden. Um sicherzustellen, dass dies der Fall ist, ist in beiden dieser `Panel`-Objekte ein transparentes `Image`-Objekt erstellt worden, welches die anderen Knöpfe bedeckt. Dazu wird ausgenutzt, dass später gezeichnete UI-Elemente bereits vorhandene bedecken. Die UI-Elemente werden in *Unity* in der Reihenfolge gezeichnet, in welcher sie im Editor angezeigt werden. Es spielt demnach keine Rolle, ob sie Unterobjekte der vorherigen Objekte sind oder nicht. Es muss darauf geachtet werden, dass sich das transparente `Image`-Objekt im Editor über den Buttons befindet, die weiterhin benutzt werden, und unter allen, die nicht mehr benutzt werden sollen.

## 6.5. Implementierung der VR-Grundsteuerung

### 6.5.1. Steuerung der VR-Komponenten

Zur Steuerung der VR-Komponenten wird das Asset „SteamVR Plugin“ der *Valve Corporation* benutzt. Dieses Asset bietet bereits eine Grundlage für die meisten VR-Funktionalitäten. Die Hauptkomponente ist hierbei das sogenannte `CameraRig`. In Kom-

ination mit der *SteamVR*-Software bietet das *CameraRig* das Tracking der VR-Geräte in *Unity*.

Für die Controllersteuerung muss zuerst eine Tastenbelegung für die einzelnen Controller-tasten angelegt werden. Dazu muss in *Unity* unter dem Reiter „Window“ der Unterpunkt „SteamVR Input“ angeklickt werden. In dem in Abbildung 27 gezeigten Fenster „SteamVR Input“ kann nun eine neue Tastenbelegung, ein sogenanntes **Action Set**, angelegt werden.

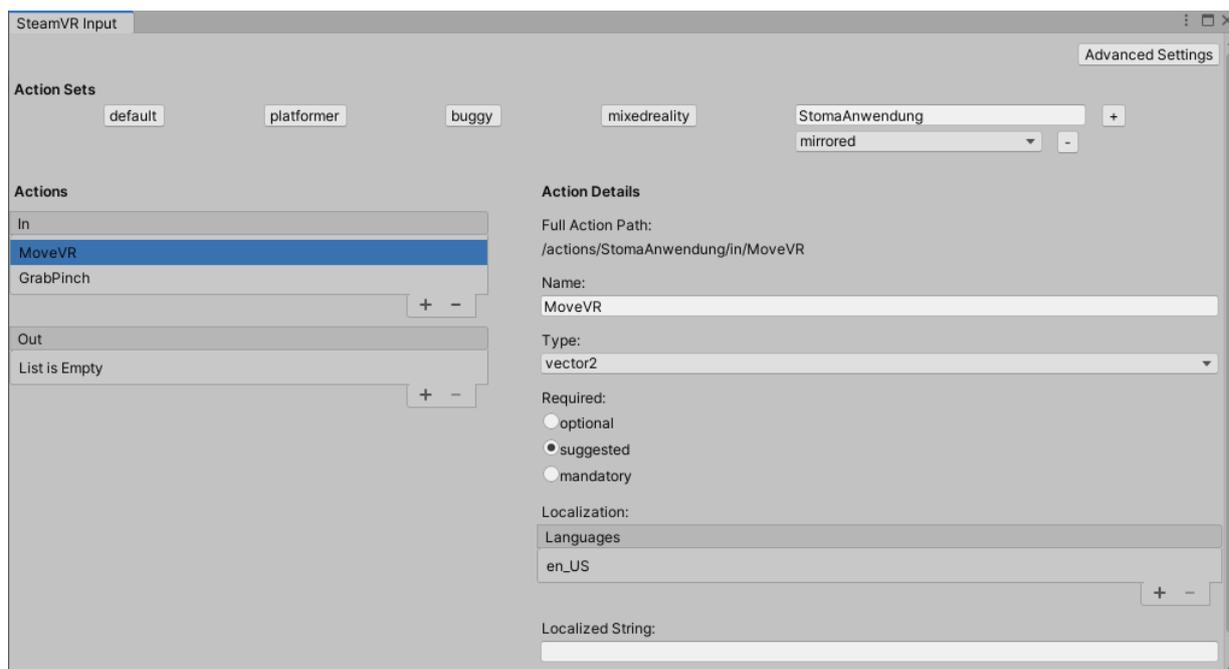


Abbildung 27: Das Fenster „SteamVR Input“

Zuerst wird ein **Action Set** mit dem Namen „StomaAnwendung“ erstellt. Danach werden die zwei Aktionen „MoveVR“ und „GrabPinch“ angelegt. „MoveVR“ dient dazu, den VR-Bereich mit den Controllern zu bewegen, falls der Benutzer sich nicht im realen Raum bewegen kann oder möchte. Daher wird der Typ **Vector2** gewählt, um eine zweidimensionale Richtung angeben zu können. Zum Greifen von Objekten wird die Aktion „GrabPinch“ verwendet. Da nur entschieden werden muss, ob aktuell gegriffen wird oder nicht, reicht hier ein **Boolean** als Typ. Um diese Aktionen mit Funktionalität zu hinterlegen, muss im Fenster „SteamVR Input“ auf „Open binding UI“ geklickt werden. Danach muss der richtige Controller ausgewählt werden. Für diese Anwendung ist es der *Vive Controller*, dessen Tastenbenennung in Abbildung 29 zu sehen ist.

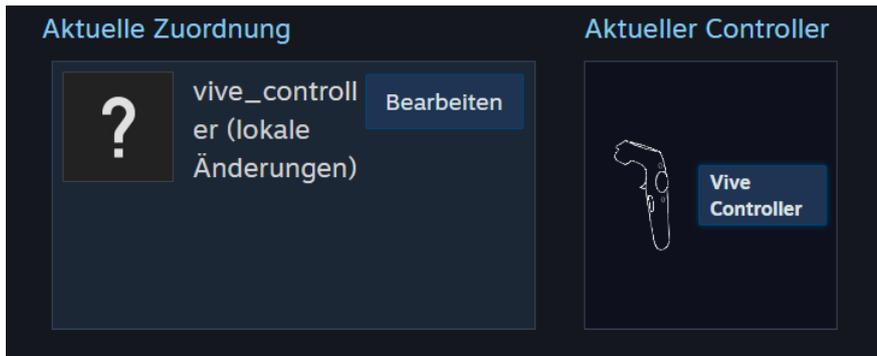


Abbildung 28: *SteamVR*: Wahl der Geräte vor dem Einstellen der Tastenbelegung

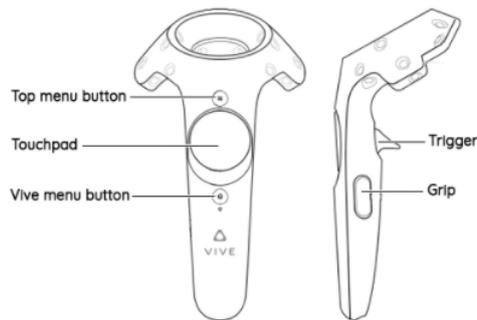


Abbildung 29: Tastenbenennung der *Vive Controller* [Ske]

Nachdem der *Vive Controller* ausgewählt worden ist, kann nun in dem in Abbildung 30 gezeigten Fenster „Bearbeiten“ ausgewählt werden. Als erstes wird über den Reiter „stomaanwendung“ das neu angelegte **Action Set** ausgewählt. Unter „Trigger“ wird für „Klick“ „grabpinch“ ausgewählt. Greifen wird also immer ausgeführt, wenn der Trigger-Knopf betätigt wird. Beim „Trackpad“ wird bei „Position“ „movevr“ gewählt. Wenn Benutzer das „Trackpad“ berühren, wird erkannt, an welcher Stelle die Berührung ist. Dies wird als zweidimensionaler Vektor in „movevr“ angegeben.

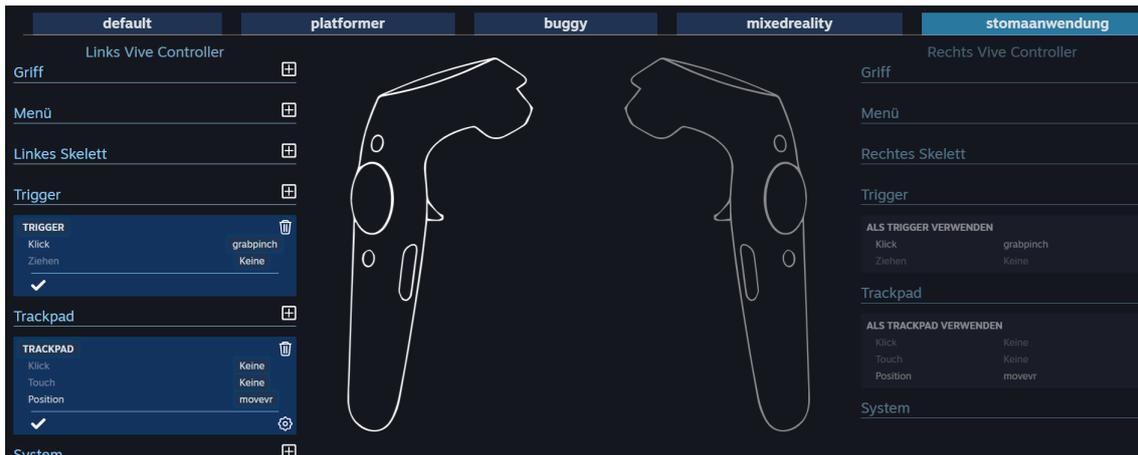


Abbildung 30: Einstellung der Tastenbelegung eines Controllers in *SteamVR*

Nachdem die Konfiguration gespeichert worden ist, muss sie innerhalb der Anwendung aktiviert werden. Dazu wird das Skript `SteamVR_ActivateActionSetOnLoad` benutzt. Dies muss auf einem beliebigen aktiven Objekt in der *Unity*-Szene angebracht werden. In dieser Anwendung wird dazu das `Prefab`-Objekt benutzt, da es das Oberobjekt aller funktionellen Objekte der Anwendung ist und dadurch garantiert aktiv ist. Nun muss die Variable „Action Set“ des Skripts auf das erzeugte `Action Set` „StomaAnwendung“ gesetzt werden. Dadurch wird dies beim Starten der Szene aktiviert und die Tastenbelegung kann benutzt werden.

### 6.5.2. Controller-Skript

Nachdem das `Action Set` eingerichtet ist, kann das Controller-Skript geschrieben werden. Als erstes wird eine Oberklasse `Controller` erzeugt, welche auch ohne die *Vive Controller* funktioniert. Diese Klasse kann danach durch andere Klassen erweitert werden, um sie auf verschiedene Geräte, beispielsweise die *Vive Controller*, zu spezialisieren.

Um angeben zu können, ob der linke oder rechte Controller verwendet wird, wird das `enum ControllerHand` erstellt. Zusätzlich wird eine Variable `hand` des Typs `ControllerHand` angelegt, um diesen Wert für jeden Controller zu speichern.

Danach werden drei `Booleans` `triggerOnce`, `triggerHold` und `isTriggerHeld` festgelegt und mit `false` initialisiert. Der `Boolean` `triggerHold` wird von der Unterklasse immer dann auf `true` gesetzt, wenn mit dem Controller gegriffen wird. Ansonsten wird es wieder auf `false` gesetzt. `triggerHold` wird bei jeder Änderung in `isTriggerHeld` gespeichert, wodurch es immer den beinhaltet, den `triggerHold` beim letzten Aufruf der `Update`-Methode gehabt hat.

Außerdem werden die Methoden `triggerDown` und `triggerUp` erzeugt. Die `triggerDown`-Methode soll immer dann aufgerufen werden, wenn gegriffen wird. Die `triggerUp`-Methode hingegen soll immer ausgeführt werden, wenn losgelassen wird. Dazu wird in der `Update`-Methode geschaut, ob `triggerHold` und `isTriggerHeld` verschiedene Werte haben. Wenn `triggerHold false` und `isTriggerHeld true` ist, bedeutet dies, dass beim letzten Update noch gegriffen worden ist, beim aktuellen Update aber nicht mehr gegriffen wird. In diesem Fall wird das Loslassen durch Aufrufen der `triggerUp`-Methode ausgelöst. Analog dazu wird die `triggerDown`-Methode aufgerufen, wenn `isTriggerHeld false` und `triggerHold true` ist.

Als nächstes wird dafür gesorgt, dass Objekte beim Greifen an die Hand gebunden und beim Loslassen wieder gelöst werden. Dafür muss zunächst bestimmt werden, welches Objekt gegriffen wird. Danach kann es an den Controller gebunden werden. Zuerst wird ein Skript `Interactions` angelegt, welches auf allen Objekten, mit denen interagiert wird, angebracht wird. Auch diese Klasse wird später für spezielle Objekte, mit denen interagiert wird, durch andere Klassen erweitert. Ein Beispiel für eine solche Erweiterung ist der Beutel. Als nächstes wird eine Methode implementiert, welche bestimmt, welches Objekt gegriffen wird. Dafür wird in `Controller` eine Liste von `Interactions` mit dem Namen `interactables` erzeugt. Zusätzlich werden die zwei `Interactions heldObject` und `nearest` angelegt. Diese speichern, welches Objekt aktuell gehalten wird beziehungsweise welches gegriffen werden soll. Wenn keines gehalten wird, hat `heldObject` den Wert `null`. Das Selbe gilt für `nearest`, wenn sich kein Objekt in der Nähe des Controllers befindet. Um zu bestimmen, welches das nächste Objekt ist, wird ein `SphereCollider` mit `isTrigger = true` auf dem Controller-Objekt angebracht. Die Reichweite des Controllers wird durch den Radius dieses `SphereColliders` stellt dargestellt. Mithilfe der `Unity`-Methoden `OnTriggerEnter` und `OnTriggerExit` wird erkannt, wenn andere Objekte dem Controller nahe genug sind und ab wann sie zu weit entfernt sind. Wenn ein Objekt nahe genug am Controller ist, wird überprüft, ob dieses Objekt ein `Interactions`-Skript besitzt. Wenn ja, wird es der Liste `interactables` zugefügt. Wenn nicht, wird es ignoriert. In `OnTriggerExit` hingegen werden Objekte mit `Interactions`-Skript aus der Liste entfernt. Dadurch befinden sich alle Objekte in Controller-Reichweite in der Liste `interactables`.

Um das nächste Objekt der Liste zu finden, wird eine Methode `getNearestObj` definiert. Wenn noch kein Objekt gehalten wird, wird in dieser Methode durch die Liste `interactables` iteriert und das nächste auswählbare Objekt wird als `nearest` gespeichert und markiert. Zum Markieren wird das „Quick Outline“-Asset aus dem *Unity Asset*

*Store* benutzt. Dieses Asset bietet ein Skript `Outline`, welches Objekte mit einer farbigen Umrandung markiert. In Abbildung 31 werden die Variablen des `Outline`-Skripts gezeigt. `Color` und `Width` geben die Farbe und Dicke der Umrandung an. `Outline Mode` erlaubt es, die Sichtbarkeit der Umrandung einzustellen. Dies wird in Abbildung 32 gezeigt. Für diese Anwendung ist `Outline All` gewählt worden.

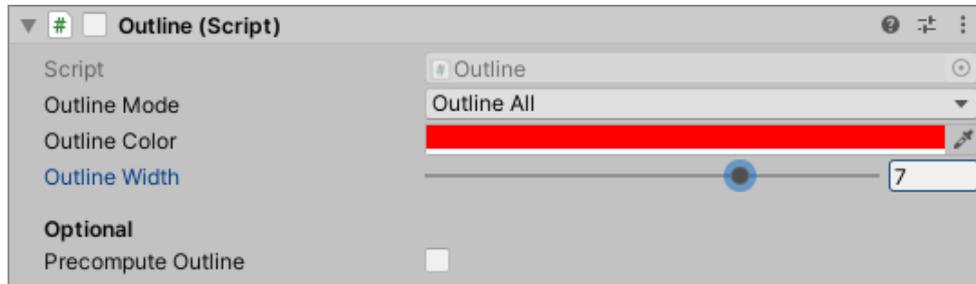


Abbildung 31: Outline-Skript

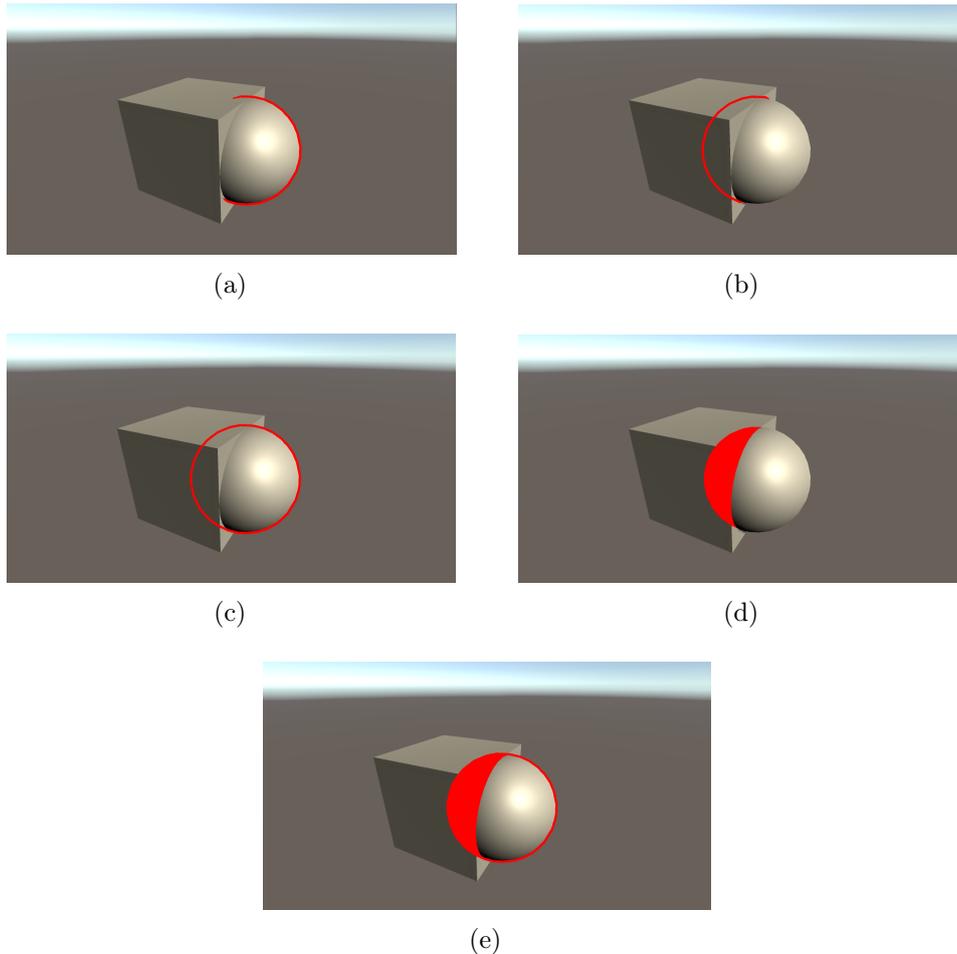


Abbildung 32: Verschiedene Einstellungen für **Outline Mode**: (a) **Outline Visible**, (b) **Outline Hidden**, (c) **Outline All**, (d) **Silhouette Only** und (e) **Outline and Silhouette**

In **Interactions** wird ein **Boolean selectable** definiert, welches angibt, ob das Objekt aktuell ausgewählt werden kann. Außerdem werden auch hier die Methoden **onTriggerDown** und **onTriggerUp** definiert, welche beide einen **Controller** als Parameter bekommen. Dies ist notwendig, da beim Greifen mancher Objekte spezielle Aktionen ausgeführt werden sollen. Die beiden Unterklassen von **Interactions** sind **InteractionObject** und **TriggerObject**. **InteractionObjects** sind Objekte, welche aufgehoben und bewegt werden können. Ihre **onTriggerDown**-Methode ruft die **grabObject**-Methode des **Controllers** auf und übergibt sich selbst als Parameter. Ihre **onTriggerUp**-Methode hingegen ruft die Methode **releaseObject** des **Controllers** mit sich selbst als Parameter auf. Dadurch kann kontrolliert werden, wann Objekte beim Greifen aufgehoben werden und wann mit ihnen interagiert wird. Für das Binden an den **Controller** wird eine

FixedJoint-Komponente auf dem Controller-Objekt angebracht. Diese bindet zwei Rigidbodies aneinander, sodass sich bei der Bewegung des einen das andere mitbewegt. In der grabObject-Methode wird der Rigidbody des gegriffenen Objektes in die connectedBody-Variable des FixedJoints des Controllers geschrieben. Außerdem wird das gegriffene Objekt in die heldObject-Variable des Controllers geschrieben. In der releaseObject-Methode werden heldObject und connectedBody wieder auf null gesetzt.

Es muss darauf geachtet werden, dass sowohl der Controller als auch alle Interaktionsobjekte einen Rigidbody besitzen. Außerdem müssen die Interaktionsobjekte auch das Outline-Skript bekommen, damit sie markiert werden können. Dieses Skript wird bei allen Objekten ausgeschaltet und immer nur für das Objekt angeschaltet, welches dem Controller am nächsten ist.

### 6.5.3. SteamController

Als nächstes wird die Klasse SteamController definiert, welche von Controller erbt. In dieser Klasse soll das Greifen speziell für die Vive Controller implementiert werden. Dazu werden drei Variablen angelegt: public SteamVR\_Action\_Boolean grabGrip, public SteamVR\_Action\_Vector2 moveVR und private SteamVR\_Behaviour\_Pose m\_Pose.

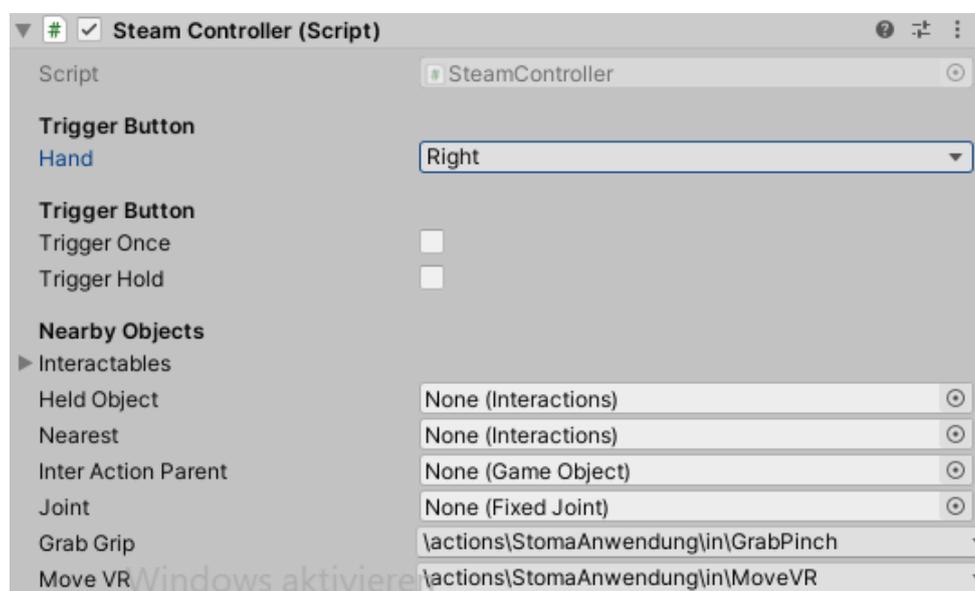


Abbildung 33: SteamController-Skript

In Abbildung 33 wird die Editoransicht des SteamController-Skripts des rechten Controllers gezeigt. Es müssen nur hand, grabGrip und moveVR eingestellt werden. Als hand

ist in diesem Beispiel `Right` gewählt worden. Für `grabGrip` werden alle `Actions` aller `Action Sets` als `Option` angegeben, welche `SteamVR_Action_Boolean` als Typ haben. Es wird die `Action GrabPinch` aus dem `Action Set StomaAnwendung` gewählt. Für `moveVR` wird `MoveVR` aus dem `Action Set StomaAnwendung` gewählt. Dadurch wird im `SteamController` jederzeit angegeben, ob der Trigger oder das Trackpad betätigt werden. Um die Funktionen der Oberklasse zu verwenden, muss der Wert der Variable `triggerHold` überschrieben werden. Dazu wird der Wert aus `grabGrip` mit der Methode `GetState` ausgelesen und in `triggerHold` gespeichert.

Als letztes wird ein zweidimensionaler Vektor erzeugt, in welchem die Position des Fingers auf dem Trackpad gespeichert wird. Wenn der Betrag dieses Vektors größer als ein Schwellwert ist, wird im `InputHandler` die Methode `moveVRRoom` mit dem Vektor als Parameter aufgerufen. Diese Methode verschiebt den betretbaren VR-Bereich in der virtuellen Welt. Es ist ein Schwellwert von 0.25 gewählt worden, um zu verhindern, dass ein „Zittereffekt“ auftritt, wenn sich der Finger in etwa in der Mitte des Trackpads befindet. Dieser würde durch viele minimale Änderungen zustande kommen, die hintereinander ausgeführt würden. Diese Methode wird in Kapitel 6.6.1 genauer erläutert.

#### 6.5.4. Body Tracking

Für das Bodytracking werden die *Vive Tracker* benutzt. Es werden drei dieser Tracker am Körper angelegt: Einer an der Hüfte und je einer an den Ellenbogen. Zuerst muss definiert werden, welcher Tracker welche Rolle bekommt. Hierfür wird zuerst die *SteamVR*-Software benutzt. In Abbildung 34 werden die verbundenen VR-Geräte angezeigt.

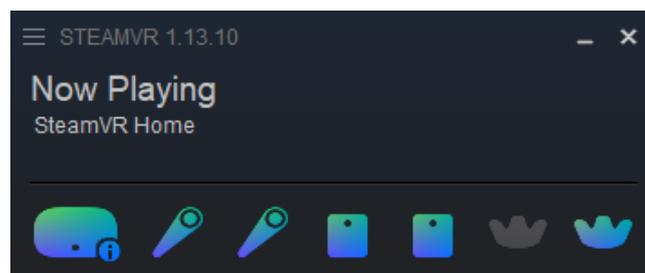


Abbildung 34: *SteamVR*-Interface

Oben links kann das Menü aufgerufen werden. Hier muss zuerst „Manage Vive Trackers“ ausgewählt werden. In dem Fenster, welches daraufhin geöffnet wird, muss dies nochmal ausgewählt werden. Anschließend wird das Menü geöffnet, welches in Abbildung 35 gesehen werden kann. Hier kann für jeden Tracker eine Rolle ausgewählt werden.

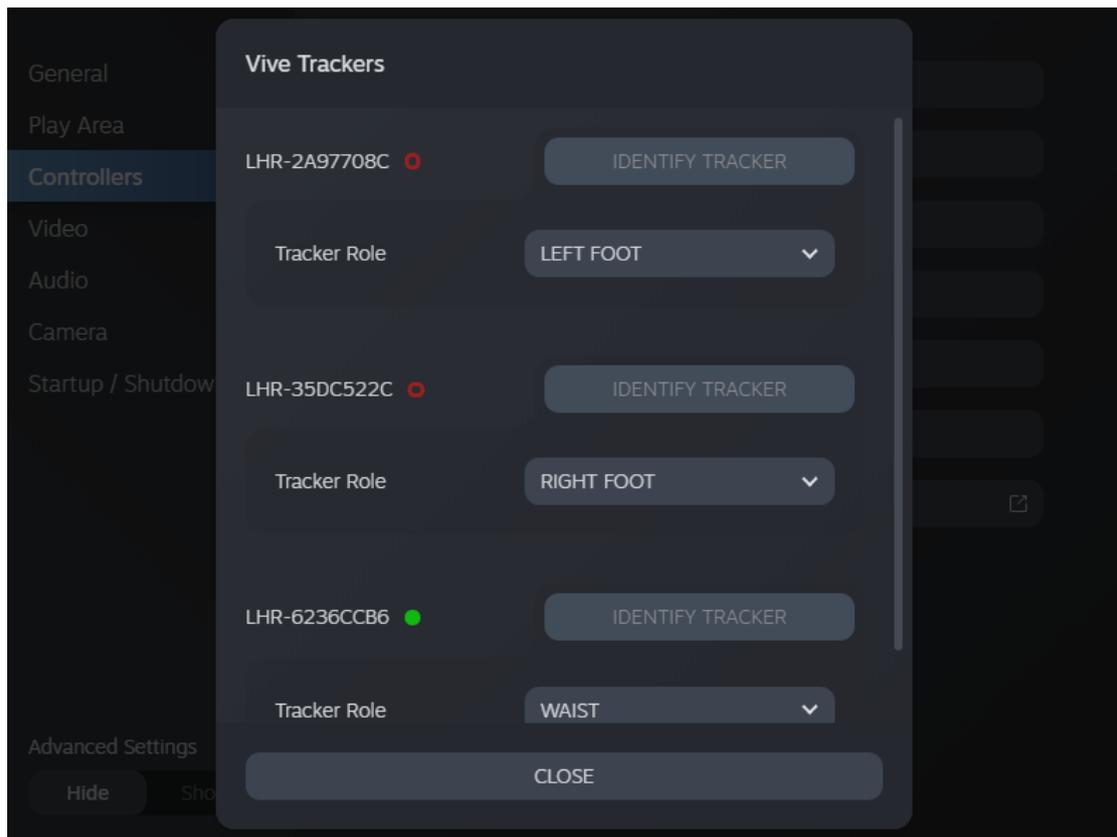


Abbildung 35: *SteamVR* Manage Tracker Ansicht

Für diese Anwendung werden die Rollen „Right Foot“, „Left Foot“ und „Waist“ gewählt. „Right Foot“ und „Left Foot“ werden anstelle der Ellenbogen benutzt, da das *Unity*-Asset die Ellenbogen nicht bereitstellt und diese also nicht benutzt werden können.

Als nächstes muss die Konfiguration des *SteamVR*-Assets in *Unity* vorgenommen werden. Dafür muss, wie in Kapitel 6.5.1 beschrieben, das Fenster „SteamVR Input“ aufgerufen werden. Von dort muss über „Open Binding UI“ wieder die Controllersteuerung aufgerufen werden. Danach öffnet sich das Menü aus Abbildung 28. Nachdem den Trackern die Rollen zugeordnet worden sind, werden sie nun, wie in Abbildung 36 gezeigt, als Controller angezeigt.



Abbildung 36: SteamVR-Controllerauswahl mit Trackern

In diesem Menü wird zuerst „Vive Tracker on Waist“ gewählt. Danach muss die persönliche Konfiguration für dieses Gerät aktiviert und bearbeitet werden. In dem darauf folgenden Menü wird „Haltungsaktionen bearbeiten“ gewählt. In dem Fenster „Haltunugszuordnungen“ muss auf der linken Seite auf „Pose“ geklickt werden. Danach wird „Pose“ ausgewählt, damit das Fenster wie in Abbildung 37 aussieht.

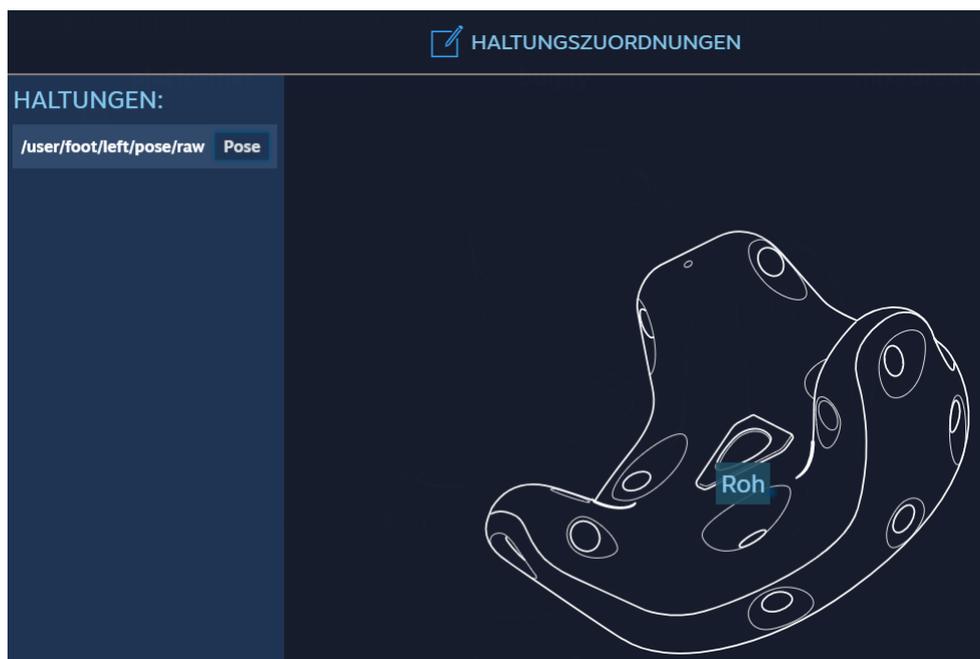


Abbildung 37: Auswahl der SteamVR-Tracker

Diese Konfiguration wird gespeichert. Danach wird das Gleiche für die Tracker „Right Foot“ und „Left Foot“ gemacht. Anschließend werden in *Unity* drei *Empty*-Objekte als

Unterobjekte des `CameraRig` angelegt, welche dazu benutzt werden, die Tracker Positionen in *Unity* zu verfolgen. Auf diese Objekte wird das Skript `Steam_VR_Behaviour_Pose` gelegt. Bei diesem Skript werden unter „Input Source“ „Waist“, „Left Foot“ und „Right Foot“ für die entsprechenden Objekte gewählt. Dies reicht bereits, um die Tracker in *Unity* benutzen zu können.

Jeder der drei Tracker, beide Controller und auch die Kamera bekommen je ein `Empty` als Unterobjekt, welches benutzt wird, um Feinanpassungen an den Positionen vorzunehmen. Diese Objekte werden „HipPos“, „LeftFootPos“, „RightFootPos“, „RightHandPos“, „LeftHandPos“ und „HeadPos“ genannt. „LeftFootPos“ und „RightFootPos“ werden trotz ihres Namens für die Ellenbogen verwendet. Die Namen stammen lediglich von der Rolle, die den Trackern in *SteamVR* zugeordnet ist.

Als nächstes wird in *Unity* das Asset „Final IK“ benutzt. Dieses stellt Skripts zur Verfügung, welche mittels inverser Kinematik die Körperhaltung aus den Positionen und Rotationen der Hände, Ellenbogen und des Kopfes berechnet. Dies ermöglicht es, das Menschenmodell zur Laufzeit an die Bewegungen des eigenen Körpers anzupassen, sodass die Immersion erhöht wird.

Dafür muss zuerst bei den Menschenmodellen eingestellt werden, dass es sich beim Rigging um ein menschliches Rigging handelt. Hierzu muss in den Importeinstellungen des Modells unter dem Reiter „Rig“ bei der Einstellung „Animation Type“ anstelle von „Generic“ „Humanoid“ ausgewählt werden, wie in Abbildung 38 gezeigt.

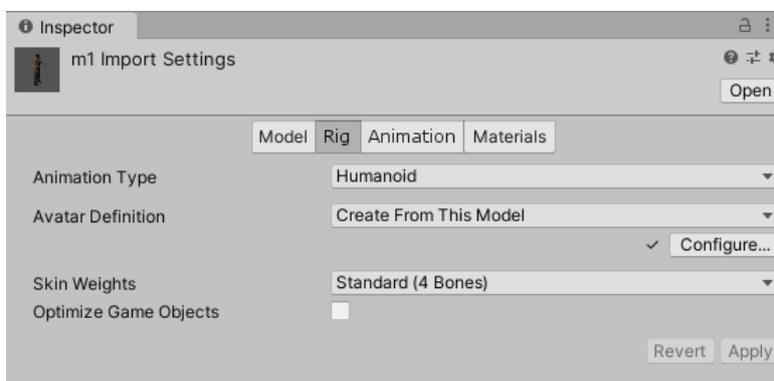


Abbildung 38: Importeinstellung der menschlichen Modelle

Anschließend muss auf dem Modell das Skript `VRIK` angebracht werden. Die einzelnen Knochen des Modells werden automatisch in der Liste `References` eingetragen. Unter der Kategorie „Solver“ gibt es Kategorien für die einzelnen Körperteile. Als nächstes werden die Unterobjekte der Tracker, der Controller und der Kamera in die entsprechenden Stellen des Skripts eingetragen. Die Gewichtung der Positionen wird auf eins gesetzt. Die

Rotation wird bei den Unterobjekten der Kamera und der Controller auf eins, bei den anderen auf 0.5 gesetzt. Dies liegt daran, dass die Tracker an ihrer Halterung gedreht werden können, weswegen die Rotation weniger stark gewichtet werden soll.

## 6.6. Implementierung der Virtual Reality Szenen

Bevor die Implementierung der Abläufe genauer beschrieben wird, werden ein paar gemeinsame Funktionen der beiden VR-Szenen beschrieben. Danach wird die Implementierung der Skripts in die einzelnen Abläufe eingeteilt und erläutert.

### 6.6.1. Gemeinsame Funktionen

Beim Starten der Szenen werden zuerst die Einstellungen aus dem Hauptmenü übernommen. Hierzu werden die verschiedenen Informationen aus den in Kapitel 6.3 beschriebenen `static`-Feldern des `MenuScripts` ausgelesen. Wie beschrieben, wird in diesen Feldern angegeben, ob das Programm aus dem Hauptmenü gestartet worden ist, ob die Anwendung aus der Patienten- oder Angehörigensicht durchlaufen wird sowie alle Informationen zum gewählten Patienten selbst.

Nachdem diese Informationen in die neue Szene geladen worden sind, wird als erstes das Patientenmodell angepasst. Hierzu sind im Voraus 15 mögliche Patientenmodelle mit der „MakeHuman“-Software angelegt worden. Alle 15 Menschenmodelle befinden sich gleichzeitig in der Szene, sind jedoch zu Beginn deaktiviert. Jedes dieser Modelle bekommt ein `PatientModel1`-Skript. In diesem Skript werden das `VRIK`-Skript und die Körpergröße des Modells angegeben. Zusätzlich werden zwei weitere Menschenmodelle angegeben.

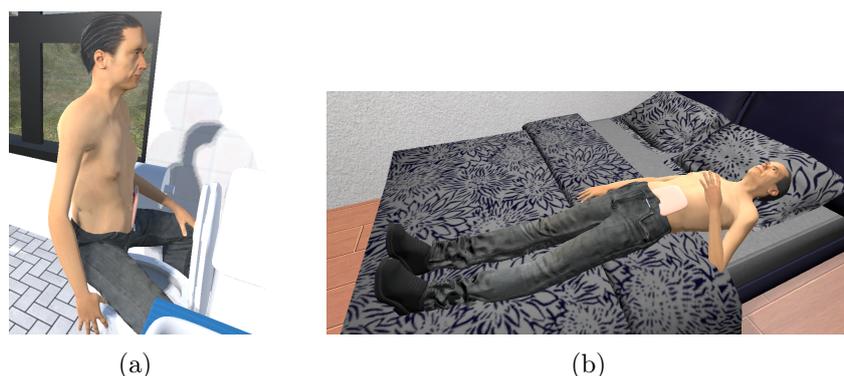


Abbildung 39: Körperhaltung des Patienten in Angehörigensicht: (a) Badezimmerszene  
(b) Schlafzimmerszene

Diese Modelle werden mit der Körperhaltung in der Szene platziert, die der Patient in der Angehörigensicht haben soll. Eines dieser Modelle wird für die Badezimmerzene verwendet, das andere für die Schlafzimmerszene. Beide dieser Posen können in Abbildung 39 gesehen werden. Dementsprechend, welcher Ablauf gewählt worden ist, wird die Körperhaltung eines dieser beiden Modelle auf das gewählte Patientenmodell übertragen. Dementsprechend, welches Geschlecht und welchen BMI der gewählte Patient hat, wird eines der 15 Modelle als aktives Modell gekennzeichnet und sichtbar gestellt. Mithilfe der Größe des Modells und der Körpergröße des Patienten wird anschließend die Skalierung des Modells angepasst, sodass die Gesamtgröße des Modells zur Größe des Patienten passt. Durch die Tasten „O“, „P“, „K“ und „L“ kann auch während der Anwendung das Modell gewechselt werden. Mit „O“ und „P“ wird ein Modell des gleichen Geschlechts, aber anderem BMI gewählt. „K“ und „L“ auf der anderen Seite können verwendet werden, um das Geschlecht zu ändern. So kann der Anwender das Modell anpassen, während er es sieht.

Neben dem Laden der Einstellungen können im „InputHandler“ auch Anpassungen an der Position des VR-Raumes vorgenommen werden. Mit den Tasten „W“, „A“, „S“ und „D“ wird der VR-Bereich durch den Anwendungsraum bewegt. Dadurch ist es nicht notwendig, den VR-Raum auf diese Anwendung spezifisch zu kalibrieren. Die Bewegung des VR-Bereichs ist dabei unabhängig von der Bildrate, damit die Anwendung auf allen Rechnern gleich funktioniert.

Neben diesen Funktionen dient das `InputHandler`-Skript auch dazu, den aktuellen Stand des Ablaufes zu aktualisieren. Wenn beispielsweise Platten und Beutel aufgelegt oder entfernt werden, wird eine Methode im `InputHandler`-Skript aufgerufen, welche dazu führt, dass der Stand aktualisiert wird. Zum Speichern des Standes dient das `ProgressState`-Skript. Dieses Skript besitzt Variablen für die einzelnen Ablaufschritte, welche durch das `InputHandler`-Skript aktualisiert werden. Zusätzlich werden einzelne Komponenten aktiviert oder deaktiviert, wenn einzelne Schritte erfüllt worden sind. So kann ein Beutel beispielsweise erst befestigt werden, nachdem er angelegt worden ist.

Ein weiteres Skript, welches bei allen Abläufen verwendet wird, ist das `CleaningScript`. Es wird immer dann verwendet, wenn Flächen oder Objekte gesäubert werden. Bei diesem Skript werden zwei `GameObjects` und vier `Materials` angegeben. Zusätzlich werden eine `float`-Variable und ein `Boolean` angelegt. Die `float`-Variable gibt an, wie nah die Objekte aneinander sein müssen, damit das eine das andere reinigt. Die `Boolean`-Variable gibt an, ob das Objekt bereits gereinigt worden ist.

Die vier `Materials` stellen hierbei jeweils die saubere oder verschmutzte Version eines

der `GameObjects` dar. Zu Beginn der Anwendung wird das `GameObject Cleaner` als sauber und das `GameObject Obj` dreckig initialisiert. Sobald die Entfernung zwischen den beiden Objekten kleiner als der angegebene Abstand ist, wird der Gegenstand `Obj` gesäubert. Wenn das geschieht, wird das Material von `Obj` mit dem Material für „sauber“ überschrieben und das Material von `Cleaner` mit dem für „dreckig“. Zusätzlich wird der Wert des `Booleans` auf `true` gesetzt und die Methode `onClean` des `InputHandlers` aufgerufen, sodass das `InputHandler`-Skript registrieren kann, dass ein Schritt ausgeführt worden ist.

Neben diesen Skripten gibt es noch die `SnapTarget`- und `SnapObject`-Skripts. Diese werden in Kombination verwendet, um zwei Objekte miteinander zu verbinden. Beim Verbinden wird das Objekt mit dem `SnapObject`-Skript an das Objekt mit dem `SnapTarget`-Skript angepasst. Die Editoransicht der beiden Skripts wird in den Abbildungen 40 und 41 angezeigt.

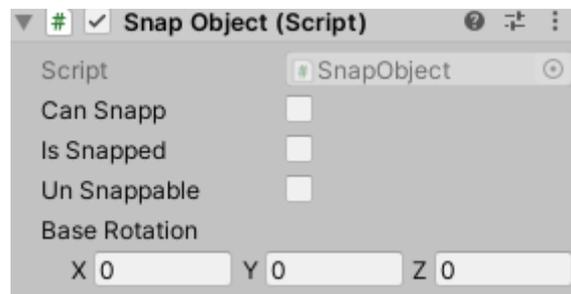


Abbildung 40: Editoransicht des `SnapObject`-Skripts

Die `Boolean`-Variable `canSnap` gibt an, ob das Objekt zu diesem Zeitpunkt an das `SnapTarget` gebunden werden kann, während `isSnapped` angibt, ob es aktuell gebunden ist. `UnSnappable` auf der anderen Seite gibt an, ob das Objekt wieder von dem `SnapTarget` gelöst werden kann. Letztlich verfügt dieses Skript noch über einen dreidimensionalen Vektor, welcher benutzt wird, um die Rotation des Objekts zu bestimmen, wenn die Objekte verbunden werden. Neben diesen Variablen besitzt das `SnapObject` keine weitere Logik, da diese vom `SnapTarget` übernommen wird.

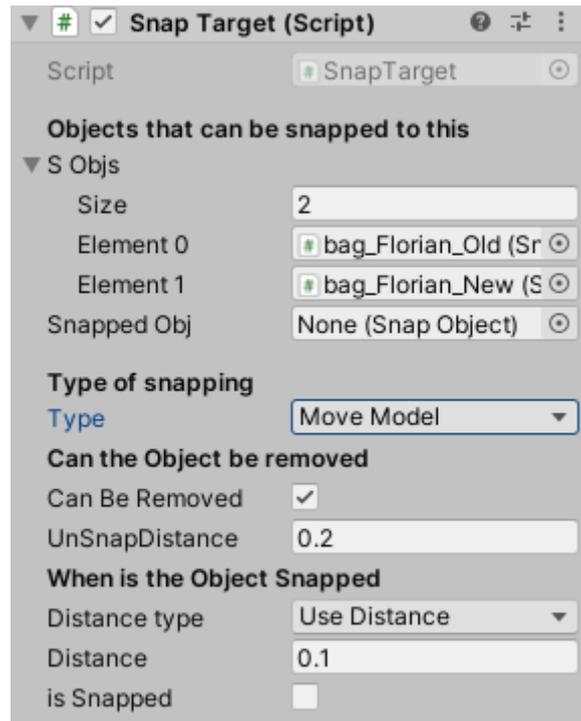


Abbildung 41: Editoransicht des SnapTarget-Skripts

Das Zusammenfügen oder Trennen der Objekte wird von dem SnapTarget-Skript übernommen. Dazu verwendet es eine Liste `sobjs` von `SnapObjects`, welche alle Objekte beinhaltet, welche an dieses `SnapTarget` gebunden werden können sowie ein einzelnes `SnapObject`, welches das aktuell gebundene Objekt darstellt.

Neben den möglichen Objekten kann auch ausgewählt werden, auf welche Art und wann die Objekte verbunden werden. Es gibt vier Möglichkeiten die Objekte zu verbinden. Es gibt die Möglichkeit, das `SnapObject` an die Position des `SnapTarget` zu teleportieren oder es mit einem anderen angegebenen Objekt zu ersetzen, welches bereits an der richtigen Stelle positioniert ist. Die zweite Methode bietet die Möglichkeit, nach dem Verbinden andere Komponenten zu verwenden, ohne das ursprüngliche Objekt zu verändern. Bei beiden dieser Arten der Verbindung kann das Objekt entweder seine Rotation behalten oder die im `SnapObject` angegebene Rotation übernehmen.

Neben der Art der Verbindung muss auch angegeben werden, wann die Objekte verbunden werden und wann diese Verbindung aufgelöst werden kann. Um zu bestimmen, wann das Objekt gelöst werden kann, wird die `Boolean`-Variable `canBeRemoved` verwendet. Das Objekt kann nur gelöst werden, wenn `canBeRemoved` und `UnSnappable` des `SnapObjects` beide `true` sind. Wenn dies der Fall ist und die Objekte mehr als `unSnapDistance` voneinander entfernt sind, werden die Objekte voneinander getrennt. Analog dazu können

die Objekte nur aneinander gebunden werden, wenn das `SnapObject` gebunden werden kann und noch kein Objekt als `snappedObj` angegeben ist. Auch hier gibt es zwei Möglichkeiten: Entweder die Objekte müssen weniger als die angegebene Distanz voneinander getrennt sein oder es kann überprüft werden, ob sich ihre `Collider` berühren.

Schließlich gibt es noch das `ProgressState`-Skript. Dies wird immer aufgerufen, wenn ein Trainingsschritt abgeschlossen worden ist. Wie bereits in Kapitel 6.1.3 angesprochen, ist es dafür zuständig, die Skripts für die einzelnen Trainingsschritte zu aktivieren und deaktivieren. Zusätzlich wird es dazu verwendet, die Schritte, welche bereits erfüllt worden sind, und die gemachten Fehler zu speichern. Die Methode `setProgress` wird von verschiedenen Skripts aufgerufen, wann immer ein Trainingsschritt erfüllt worden ist. Danach prüft das `ProgressState`-Skript, ob dieser Schritt bereits hätte ausgeführt werden sollen oder ob Schritte vergessen worden sind und speichert dies intern, bis es beim Abschließen des Ablaufs in eine Datei geschrieben wird.

### 6.6.2. Leeren des Beutels

In diesem Ablauf soll das Leeren des Beutels trainiert werden. Hierzu sollen folgende Schritte durchgeführt werden:

- Patient setzt sich falsch herum auf die Toilette
- Verschluss öffnen
- Beutel leeren
- Verschluss mit Toilettenpapier säubern
- Verschluss wieder verschließen

Im ersten Schritt soll sich der Patient falsch herum auf die Toilette setzen. Dieser Schritt wird von der Anwendung nicht überprüft, da eine genaue Detektion von Position und Haltung des Patienten den Rahmen dieser Arbeit übersteigen würde und eine ungenaue Detektion zu verwirrendem Feedback führen könnte.

Falls die Anwendung aus Angehörigersicht ausgeführt wird, befindet sich der Patient bereits sitzend auf der Toilette. Wird die Anwendung aber aus Patientensicht ausgeführt, so muss dem Patienten ein Hocker bereitgestellt werden, auf welchen er sich setzen kann. Das Öffnen des Verschlusses ist so implementiert, dass beide Hände abwechselnd mit den einzelnen Teilen interagieren müssen. Dadurch soll das Auffalten besser simuliert werden,

da auch dort die Hände abwechselnd benutzt werden. Mit jeder Betätigung des Trigger-Knopfs in der Nähe des Objekts wird ein weiterer Teil der Öffnungsanimation abgespielt. Die Entscheidung, eine Animationen zum Öffnen zu verwenden, ist getroffen worden, da die einzelnen beweglichen Teile recht klein sind und es mit den VR-Controllern sehr schwierig wäre, diese richtig zu bewegen. Das Öffnen des Verschlusses wird von dem `Bagseal`-Skript gesteuert, welches eine Unterklasse des `TriggerObjects` ist. Dadurch kann die Beutellasche mit den Controllern nicht bewegt, sondern nur betätigt werden. Bei jeder Betätigung wird überprüft, ob die Lasche aktuell geöffnet beziehungsweise geschlossen werden kann. Wenn dies der Fall ist, wird überprüft, mit welcher Hand beziehungsweise welchem Controller die Lasche betätigt worden ist und ob dies richtig ist. Danach wird die nächste Stufe der Animation zum Öffnen oder Schließen der Lasche abgespielt. Nachdem die Lasche vollständig geöffnet oder geschlossen worden ist, wird die Methode `setProgress` im `ProgressState` mit dem entsprechendem Parameter aufgerufen, um dort zu vermerken, dass der Schritt erfüllt worden ist.

Nachdem der Beutel geöffnet worden ist, beginnt eine kurze Animation, welche das Ausleeren des Beutels darstellt. Wenn der Beutel vollständig ausgeleert ist, wird der vorderste Teil des Verschlusses mit einer anderen Textur gefärbt, um anzuzeigen, dass er gesäubert werden muss. An dieser Stelle wird ein `CleaningScript` aktiviert, welches zum Reinigen des Verschlusses benutzt wird.

Anschließend kann die Öffnung mit Toilettenpapier gereinigt werden, wofür ein Stück Toilettenpapier von der Rolle gerissen werden kann. Dieses Stück muss danach gegen die Öffnung des Beutelverschlusses gehalten werden. Dadurch wird das `CleaningScript` ausgeführt und der Verschluss gereinigt. Der Verschluss wird daraufhin als sauber und das Toilettenpapier als dreckig dargestellt.

Abschließend muss der Beutel wieder verschlossen werden. Analog zum Öffnen des Verschlusses wird hier das `Bagseal`-Skript verwendet. Der Verschluss wird abwechselnd mit beiden Händen geschlossen. Nachdem der Beutel wieder vollständig geschlossen ist, ist der Ablauf abgeschlossen.

### 6.6.3. Wechseln des Beutels

In dem zweiten Ablauf wird das Wechseln des Beutels trainiert. Dazu zählen die folgenden Schritte:

- Beutel an der Lasche packen und abziehen
- Beutel komplett entsorgen

- Stoma reinigen
- Neuen Beutel anbringen
- Neuen Beutel befestigen

Als erstes muss der alte Beutel entfernt werden. Hierfür wird mit dem Controller die Lasche des Beutels gegriffen und der Beutel abgezogen. Dabei wird im `SnapTarget`-Skript der Platte die `Boolean`-Variable `canBeRemoved` und die `Boolean`-Variable `unSnappable` des Beutels auf `true` gesetzt, sodass der Beutel von der Platte entfernt werden kann. Danach kann er einfach abgenommen und entsorgt werden. Wenn er entfernt wird, wird im `SnapTarget` die Methode `unSnap` aufgerufen. Dadurch wird im `ProgressState`-Skript das `CleaningScript` für den nächsten Schritt aktiviert.

Nachdem der alte Beutel entfernt worden ist, muss die Gegend um das Stoma herum gesäubert werden. Dazu wird das `CleaningScript` verwendet und es muss nur das Stück Toilettenpapier auf das Stoma gehalten werden.



Abbildung 42: Visualisierung der Stellen, an welchen der Beutel festgedrückt wird

Nachdem das Stoma gereinigt worden ist, kann der neue Beutel angebracht werden. Hierzu werden die Skripts `SnapTarget` und `SnapObject` verwendet. Wenn der neue Beutel nahe genug an die Platte gehalten wird, wird er an die richtige Position teleportiert und von der Hand gelöst. Gleichzeitig werden vier rote Kreise an dem Beutel sichtbar, welche in Abbildung 42 gesehen werden können. Diese dienen zur Visualisierung der Stellen des Beutels, welche noch fest gedrückt werden müssen. Dadurch soll sichergestellt werden, dass die Patienten lernen, den Beutel von allen Seiten festzudrücken, da der Beutel sonst auslaufen könnte.

#### 6.6.4. Wechseln der Platte

Im letzten Ablauf wird das Wechseln der Platte trainiert. Dazu zählen die folgenden Schritte:

- Alte Platte entfernen
- Kompresse befeuchten
- Stoma mit feuchter Kompresse reinigen
- Stoma mit trockener Kompresse trocknen
- Neue Platte zuschneiden
- Folie entfernen
- Platte platzieren
- Beutel platzieren
- Beutel befestigen
- Beutel halten

Alle bei diesem Ablauf benötigten Objekte werden auf dem Tisch neben dem Bett platziert und sind in Abbildung 43 zu sehen.



Abbildung 43: Übersicht der beim Wechseln der Platte verwendeten Objekte

Als erstes muss die alte Platte mitsamt des Beutels entfernt werden. Hierfür werden die Skripts `SnapTarget` und `SnapObject` benutzt. Die Platte besitzt ein `PlateScript`, welches eine Unterklasse des `InteractionObjects` ist. Sie kann also aufgehoben und frei bewegt werden.

Dabei wird das `CleaningScript` der Wasserschale aktiviert. Dadurch wird es möglich, die vordere der beiden Kompressen zu befeuchten. Für die Wasserschale wird zweimal dasselbe Material verwendet, sodass es keinen optischen Unterschied zwischen dem sauberen und dreckigen Objekt gibt. Das Material der Komresse wird mit einem nass aussehenden Material ersetzt, um anzuzeigen, dass die Komresse befeuchtet worden ist. Nachdem die Komresse befeuchtet worden ist, kann sie benutzt werden, um das Stoma zu reinigen. Auch hier wird ein `CleaningScript` benutzt. Dazu wird ein Objekt angelegt, welches die Gegend um das Stoma herum darstellt. Dieses ist zu Beginn der Anwendung dreckig und erhält beim Reinigen ein feucht aussehendes Material. Das Material der Komresse wird mit einem dreckig aussehenden Material ersetzt. Die Komresse kann anschließend entsorgt werden.

Danach kann die zweite Komresse benutzt werden, um das Stoma zu trocknen. Das Trocknen des Stomas führt dazu, dass das Material um das Stoma herum mit einem durchsichtigen Material ausgetauscht wird, während das Material der Komresse zu einem dreckig aussehenden wird. Im Anschluss kann auch diese Komresse entsorgt werden.

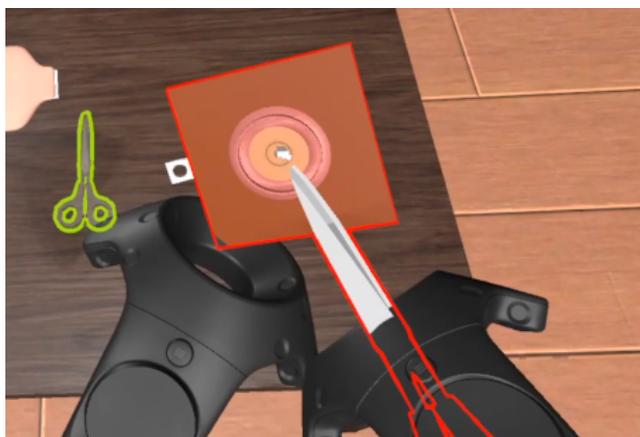


Abbildung 44: Schneiden der Platte

Im Anschluss kann die neue Platte zurechtgeschnitten werden, wie in Abbildung 44 abgebildet ist. Hier muss beachtet werden, dass die Größe des Stomas je nach Einstellung angepasst wird. Passend zu den drei möglichen Stomagrößen werden zu Beginn der Anwendung drei Schablonen bereitgelegt. Wenn der Anwender die passende Schablone auf

die neue Platte legt, wird auf dieser ein Kreis mit der entsprechenden Größe eingezeichnet. Danach kann mithilfe des `ScissorScripts` die Schere in die Hand genommen werden und an ihrer Stelle erscheint eine Silhouette. Die Schere kann wieder abgelegt werden, indem der Controller mit der Schere auf die Silhouette bewegt und der Trigger betätigt wird. Mit der Schere in der Hand kann die Platte schließlich geschnitten werden. Zum Schneiden werden die Skripts `ScissorScript`, `CutPart` und `CutPartManager` verwendet. Um das Schneiden der Platte zu ermöglichen, ist die schneidbare Fläche der Platte in mehrere kleine Objekte eingeteilt worden. Jedes dieser Objekte bekommt ein `CutPart`-Skript zugeordnet. Wenn die Schere eines dieser Objekte berührt, wird das Objekt deaktiviert und die Methode `checkCut` des `CutPartManagers` aufgerufen. In dieser Methode wird überprüft, ob die Platte richtig geschnitten worden ist. Dazu werden die einzelnen Objekte in drei Kategorien eingeteilt: Teile, welche geschnitten werden müssen, Teile, die geschnitten werden können und Teile, die nicht geschnitten werden dürfen. In welche Kategorie die einzelnen Objekte eingeteilt werden, hängt von der gewählten Stomagröße ab. Je nach gewählter Stomagröße werden ein minimaler und ein maximaler Abstand festgelegt. Alle Objekte mit `CutPart`-Skript, welche weniger als der minimale Abstand von dem Zentrum der Platte entfernt sind, müssen ausgeschnitten werden. Alle, welche weiter als der maximale Abstand entfernt sind, dürfen nicht ausgeschnitten werden. Alle dazwischen können ausgeschnitten werden, müssen es aber nicht.

Falls beim Schneiden der Platte zu viel ausgeschnitten worden ist, wird ein Fehlerton abgespielt. An dieser Stelle muss die Platte in den Mülleimer geworfen werden und eine neue Platte erscheint. Ist die Platte richtig geschnitten worden, wird im `CutPartManager` das weitere Schneiden verhindert und das `FoilScript` wird aktiviert.

Dieses Skript ist für die Steuerung der Folie verantwortlich. Die Folie besteht aus einem verformbaren Würfel, welcher mit einem `Rigging` versehen worden ist. Wenn das `FoilScript` aktiviert worden ist, kann die Folie an einem der vier Eckpunkte gegriffen werden. Sobald dies geschehen ist, werden die `isKinematic`-Werte der Knochen auf `false` gesetzt. Dies führt dazu, dass sich die Folie verformen kann und von der Platte gelöst wird.

Nachdem die Folie entfernt worden ist, kann die neue Platte auf dem Stoma platziert werden. Hierfür werden die Skripts `SnapTarget` und `SnapObject` verwendet. Auf dieser Platte kann der Beutel wie bereits beschrieben platziert und befestigt werden.

Schließlich muss die Platte festgehalten werden, um den Klebstoff zu wärmen. Hierzu ist auf dem neuen Beutel ein `TimerScript` platziert worden. Dieses überprüft, ob sich die Controller in der Nähe des neuen Beutels befinden. Solange dies der Fall ist, läuft

der Timer. Für die visuelle Darstellung an einer der Wände wird das `Image`-Objekt eines roten Kreises verwendet, auf welchem ein kleinerer weißer Kreis platziert wird. Dadurch erscheint nur ein roter Ring. Für den größeren Kreis wird als `Image Type` der Wert `Filled` und als `Fill Method` der Wert `Radial 360` ausgewählt. Das sorgt dafür, dass ein neuer Wert `FillAmount` angegeben werden kann, welcher angibt, wie sehr der Ring gefüllt ist. Der Wert `FillAmount` wird also mit der Zeit erhöht, sodass der Ring nach insgesamt 30 Sekunden vollständig gefüllt ist. Danach ist das Wechseln der Platte abgeschlossen.

## 7. Studie

Das Ziel dieser Arbeit ist es, eine Anwendung zu erstellen, welche angehende Stomapatienten auf das Leben mit einem Stoma vorbereitet. Dazu sollen sie lernen, ihr Stoma zu versorgen, um ihnen die Angst davor zu nehmen.

Um zu überprüfen, ob die Anwendung dieses Ziel erfüllt, wird eine Studie mit einer großen Anzahl angehender Stomapatienten benötigt. Dies ist im Rahmen dieser Arbeit aus Zeitgründen und insbesondere wegen der andauernden Corona-Pandemie nicht möglich gewesen. Stattdessen ist die Anwendung von einer kleinen Anzahl Mitarbeitern des Kreis-klinikums Siegen und des Lehrstuhls *Medizinische Informatik und Mikrosystementwurf* der Universität Siegen getestet worden.

### 7.1. Ablauf der Testung

Zu Beginn der Testung sind die Probanden gebeten worden, eine Einverständniserklärung auszufüllen. In dieser stimmen die Probanden zu, dass ihre Angaben zu Alter, Größe, Gewicht, Geschlecht und einem Pseudonym, welche in einem weiteren Formular angegeben werden, für diese Studie benutzt werden dürfen. Diese Formulare befinden sich im Anhang. Diese Daten werden von den Studienleitern dazu verwendet, um die Patienten in der Anwendung anzulegen.

Nachdem die Patienten angelegt worden sind, sind die Probanden gebeten worden, sich an den Rechner zu setzen und die Anwendung durchzuführen. Zuerst haben sie für jeden der drei Abläufe die ersten drei Trainingsphasen durchführen sollen.

Im Anschluss sind die Probanden gebeten worden, das bereitgestellte HMD sowie die drei Tracker anzulegen. Danach sind ihnen die Controller angereicht worden. Anschließend ist eine Testszene gestartet worden, in welcher die Probanden sich mit der Steuerung haben vertraut machen können.

Sobald die Probanden bereit gewesen sind, ist die vierte Phase der Abläufe in der Angehörigensicht durchgeführt worden. Als erstes ist das Leeren des Beutel, danach das Wechseln des Beutel und schließlich das Wechseln der Platte durchgeführt worden. Nachdem ein Proband einen Ablauf abgeschlossen hatte, ist der nächste Ablauf gestartet worden.

Erst nachdem ein Proband alle drei Abläufe in der Angehörigensicht durchgeführt hatte, sind sie in der Patientensicht durchgeführt worden. Auch hier ist mit dem Leeren des Beutels begonnen worden. Da sich die Patienten bei diesem Ablauf auf eine Toilette setzen sollen, ist hierfür ein Hocker in dem Trackingbereich platziert worden. Die Position

des Hockers im Trackingbereich ist vor Beginn der Studie erprobt und festgelegt worden, sodass der Hocker an der Stelle der Toilette in der virtuellen Realität steht. Zu Beginn des Ablaufs setzt sich der Proband auf den Hocker und führt danach den Ablauf aus. Nachdem das Leeren des Beutels abgeschlossen worden ist, wird das Wechseln des Beutels gestartet. Aus Komfortgründen bleibt der Proband hierfür auf dem Hocker sitzen und steht zwischen den beiden Abläufen nicht auf. Nachdem dieser Ablauf ausgeführt worden ist, kann der Proband aufstehen und der Hocker wird entfernt. Danach wird das Wechseln der Platte gestartet, welches im Stehen durchgeführt wird.

Nachdem alle Abläufe abgeschlossen worden sind, können die Probanden das HMD wieder abnehmen und dem Studienleiter übergeben. Auch die Tracker können nun abgelegt werden. Zum Abschluss des Versuchs haben die Probanden schließlich zwei Fragebögen ausgefüllt. Der erste Fragebogen befasst sich mit der virtuellen Realität und ist von den Betreuern bereitgestellt worden. Der Fragebogen besteht aus Fragen, welche aus einem Fragebogen der *igroup* [igr] und einem Fragebogen von *Witmer und Singer* [Sin] übernommen worden sind. Der zweite Fragebogen ist von dem Kreisklinikum Siegen bereitgestellt worden und befasst sich mit der Angst der Patienten. Dieser Fragebogen dient allerdings nur zur Kontrolle für eine spätere, größere Studie und wird in dieser Arbeit daher nicht ausgewertet.

## 7.2. Beobachtungen während der Testung

Während der Testung sind keine größeren Schwierigkeiten aufgetreten, allerdings haben bereits erste Eindrücke der Probanden gesammelt werden können.

Das Auffälligste hierbei sind die Eindrücke zum Schneiden der Platte gewesen. Hier haben viele der Probanden Schwierigkeiten gehabt. Dies liegt vor allem daran, dass beim Schneiden kein Widerstand zu fühlen ist. Dadurch geschieht es sehr schnell, dass das Loch in der Platte zu groß geschnitten wird, wenn die Hände nicht still genug sind. Vor allem bei diesem Schritt spielt die VR-Erfahrung eine große Rolle und Probanden, welche bereits Erfahrung mit VR haben, haben wesentlich weniger Schwierigkeiten gehabt.

Zusätzlich ist nicht allen Probanden klar gewesen, wie das Auffalten des Verschlusses ausgeführt wird. Daher sind Anweisungen während der Studie benötigt worden, die allerdings die Immersion beeinträchtigen.

### 7.3. Auswertung

Um die Anwendung zu bewerten, ist der Fragebogen verwendet worden, welchen die Teilnehmer ausgefüllt hatten. Insgesamt haben 9 Teilnehmer die Fragebögen ausgefüllt. Die verschiedenen Fragen des Fragebogens können mit einem Wert zwischen -3 und 3 bewertet werden. Manche Fragen sind so formuliert, dass niedrige Zahlen eine bessere Bewertung darstellen. Diese werden in den folgenden Diagrammen mit orange markiert, während die restlichen in blau angegeben werden. Bei der Auswertung wird die Summe der in orange markierten Fragen von der Summe der in blau markierten Fragen abgezogen. Das Ergebnis ist anschließend durch die Anzahl der Fragen geteilt worden und wird in den folgenden Diagrammen mit gelb markiert.

#### Präsenz

Unter Präsenz wird das Gefühl verstanden, in der virtuellen Welt anwesend zu sein. Mit der generellen Präsenz setzt sich nur die erste Frage des Fragebogens auseinander. Mit 1.3 ist die Präsenz der Anwendung als positiv bewertet worden.

#### Örtliche Präsenz

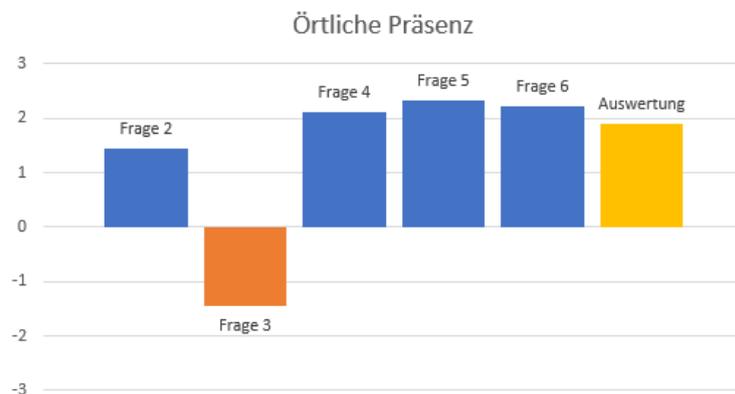


Abbildung 45: Bewertung der örtlichen Präsenz

Neben der allgemeinen Präsenz wird auch die örtliche Präsenz abgefragt. Dabei soll erfragt werden, wie stark die Wirkung der Ortsillusion, welche in Kapitel 3.1.3 erläutert worden ist, auf den Anwender wirkt. Dazu werden die Fragen zwei bis sechs verwendet. Die örtliche Präsenz ist mit 1.9 bewertet worden, was ein gutes Ergebnis ist. Die Bewertung der einzelnen Fragen kann Abbildung 45 entnommen werden.

## Involviertheit

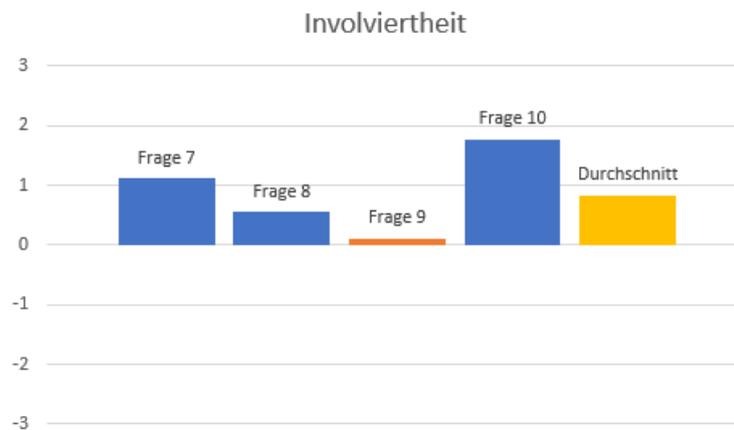


Abbildung 46: Bewertung der Involviertheit

Bei der Involviertheit wird überprüft, wie sehr sich der Benutzer auf die Anwendung konzentriert und dabei die reale Welt ausblendet. Dazu sind die Fragen sieben bis zehn verwendet worden. Mit einem Ergebnis von 0.8 ist die Involviertheit schlechter als die Präsenz bewertet worden. Dies kann daran liegen, dass sich die Probanden beim Leeren und Wechseln des Beutels in der Patientensicht haben auf einen Hocker setzen müssen. Obwohl der Hocker an der gleichen Stelle wie die Toilette in der virtuellen Welt ist, müssen sich die Probanden dennoch sehr vorsichtig auf den Hocker setzen und sich dabei auf die reale Welt konzentrieren.

## Realismus

Da der Realismus sowohl im Fragebogen der *igroup* als auch im Fragebogen von *Witmer und Singer* erfragt wird, werden die Ergebnisse zuerst einzeln bewertet und im Anschluss wird ein Durchschnitt berechnet.

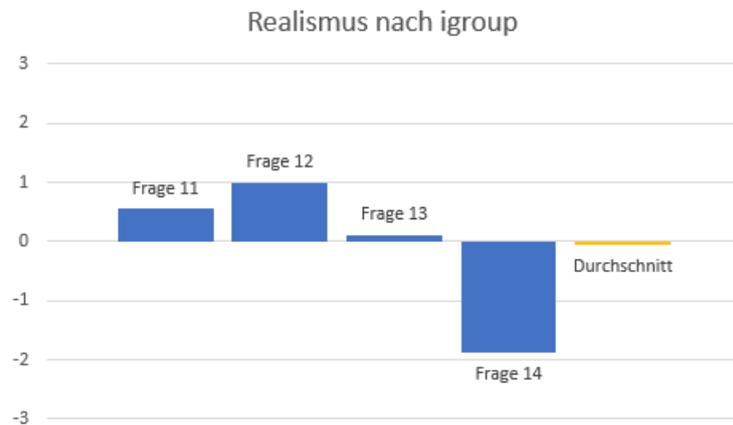


Abbildung 47: Bewertung des Realismus nach *igroup*

Die Fragen 11 bis 14 befassen sich mit dem Realismus nach *igroup*. Hier ist als Ergebnis ein Wert von -0.1 erreicht worden. Dies könnte unter anderem daran liegen, dass die Modelle nicht realistisch genug aussehen und die Patientenmodelle in der Angehörigen-sicht keinerlei Bewegung zeigen. Auf der anderen Seite hat die Frage „Die virtuelle Welt erschien mir wirklicher als die reale Welt.“ auch für Verwirrung bei den Probanden gesorgt.

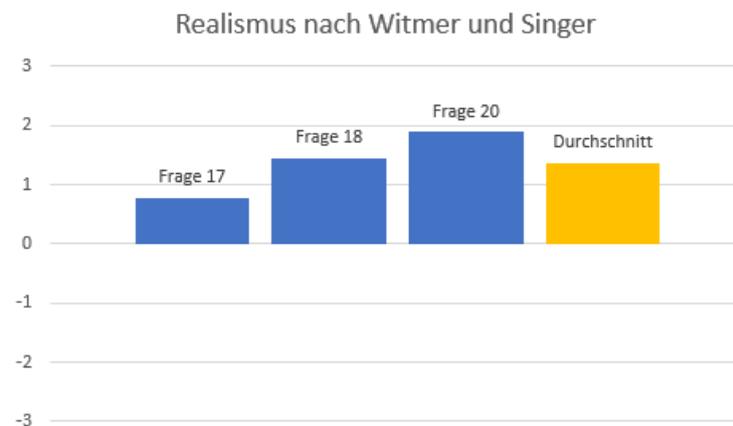


Abbildung 48: Bewertung des Realismus nach *Witmer und Singer*

Der Realismus nach *Witmer und Singer* ist mit 1.3 deutlich besser als der Realismus nach der *igroup* bewertet worden. Hierzu sind die Fragen 17, 18 und 20 verwendet worden.

### Durchschnittlicher Realismus

Im Durchschnitt ist der Realismus daher mit 1.0 bewertet worden. Während dies ein positiver Wert ist, ist es auch ein Zeichen dafür, dass die Anwendung in einigen Aspekten noch realistischer gestaltet werden kann.

### Interaktionsmöglichkeiten

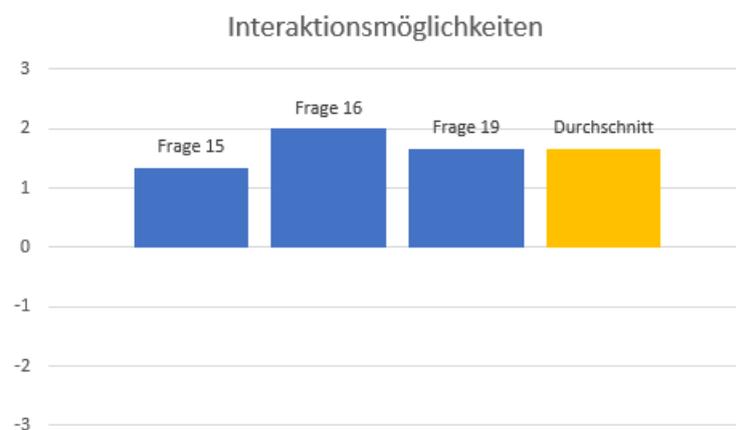


Abbildung 49: Bewertung der Interaktionsmöglichkeiten

Nach dem Realismus wird auch die Möglichkeit bewertet, mit der Umgebung zu interagieren. Dazu werden die Fragen 15, 16 und 19 verwendet. Dabei wird darauf eingegangen, wie natürlich die Interaktion mit der Umgebung ist. Dies ist von den Probanden mit 1.7 als gut bewertet worden.

## Qualität des Interfaces

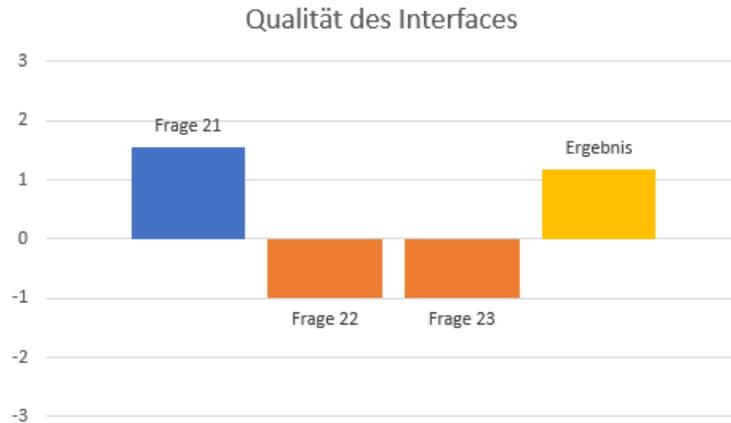


Abbildung 50: Bewertung des Interfaces

Die Qualität des Interfaces wird von den Fragen 21 bis 23 bewertet und beschreibt, wie sehr die Anwendung den Nutzer beim Ausführen der Aufgabe unterstützt oder beeinträchtigt. Mit 1.2 ist auch dies von den Probanden als positiv bewertet worden.

### Bewertung der Audioelemente

Die Bewertung der Audioelemente wird ausschließlich durch Frage 25 bestimmt und ist mit 1.6 ebenfalls als positiv bewertet worden.

Schließlich werden noch einige Fragen ausgewertet, welche nicht in eine der anderen Kategorien fallen.

### Frage 24: „Wie gut konnten Sie sich auf die zugewiesenen Aufgaben oder erforderlichen Aktivitäten konzentrieren und nicht auf die Mechanismen, die zur Durchführung dieser Aufgaben oder Aktivitäten notwendig sind?“

Diese Frage dient dazu zu erfahren, wie intuitiv die Steuerung der Anwendung ist. Je intuitiver die Steuerung, desto weniger brauchen die Probanden sich auf die Steuerung konzentrieren. Stattdessen können sie sich auf die Aufgabe selbst konzentrieren. Diese Frage ist mit 1.4 bewertet worden. Dies weist darauf hin, dass die Steuerung noch verbessert werden könnte.

**Frage 26: „Haben Sie sich während des gesamten Durchlaufs ausreichend geleitet gefühlt?“**

Bei dieser Frage wird bewertet, wie klar die Anweisungen der Studienleiter sind. Mit 2.3 wurde dies von den Probanden als sehr gut bewertet. Dies weist darauf hin, dass alle Fragen der Probanden ausreichend von den Testleitern beantwortet werden können.

**VR-Erfahrung und Nebenwirkungen**

Die Fragen 27 bis 29 befassen sich damit, ob die Probanden bereits Erfahrungen mit VR gehabt haben, diese regelmäßig benutzen und ob sie bei der Anwendung Nebenwirkungen verspürt haben. Sieben von neun Probanden haben bereits Erfahrung mit VR gehabt und fünf davon haben zu dem Zeitpunkt der Studie sogar regelmäßig VR benutzt. Nur einer der Probanden hat ein leichtes Unwohlsein während der Verwendung der Anwendung verspürt.

**Frage 30: „Wobei hatten Sie den größten Lerneffekt für die einzelnen Stoma-Schritte?“**

Zur Auswahl stehen bei dieser Frage folgende Optionen: „Die Videos“, „Das Anklicken der Reihenfolge“ und „Durchlauf in VR“. Diese Frage ist allerdings erst nach den ersten beiden Probanden hinzugefügt worden, weswegen hier nur für sieben der neun Probanden ein Ergebnis vorliegt. Von diesen sieben Testpersonen hat nur eine angegeben, dass sie beim Ansehen der Videos den größten Lerneffekt gehabt hat. Die restlichen sechs Probanden haben angegeben, durch das Durchlaufen in VR am meisten gelernt zu haben.

**Frage 31: Teilen sie uns Ihre Meinung mit: Haben sie noch Vorschläge oder Kritiken?**

Einer der Probanden hat kritisiert, dass das Schneiden der Platte sehr schwierig gewesen sei und die Handhabung der Controller nicht ausreichend klar gewesen sei. Mit der Handhabung ist wahrscheinlich das Aufklappen der Beutelöffnung gemeint, da dies nicht ausreichend im Voraus erklärt worden ist. Ein weiterer Kritikpunkt ist gewesen, dass es mit der VR-Bille etwas schwierig sei, in der Patientensicht an sich herab zu schauen und dort die Aufgaben zu erledigen. Zusätzlich ist vorgeschlagen worden, am Ende der Abläufe einen Audioton abzuspielen, um das Ende der Abläufe klarer zu signalisieren. Als letztes ist noch angemerkt worden, dass die Controller kein ausreichend guter Ersatz für die Hände seien, da die Fingerbewegung nicht in die virtuelle Welt übertragen werden kann.

## 7.4. Bewertung der Anwendung

Insgesamt ist die Anwendung gut bewertet worden. Alle Probanden haben jeden der Abläufe erfolgreich abschließen können. Aus den Fragebögen wird deutlich, dass sich die Patienten während der Abläufe wohlfühlt und gut geleitet gefühlt haben. Hier ist allerdings auffällig, dass Probanden, welche bereits Erfahrung mit VR gehabt haben, deutlich weniger Hilfe benötigt haben. Vor allem das Schneiden der Platte und das Auf- und Zufalten der Beutelöffnung ist für einige der Probanden nicht intuitiv genug gewesen und hat daher erläutert werden müssen.

Da die meisten Probanden den größten Lerneffekt beim eigenen Durchlaufen der Anwendung in VR gehabt haben, kann darauf geschlossen werden, dass die Anwendung zum Lernen eine gute Ergänzung zum einfachen Beobachten der Abläufe darstellt. Inwiefern die Anwendung angehenden Stomapatienten die Angst vor dem Stoma nimmt, kann allerdings erst durch eine größere Studie mit tatsächlichen Patienten evaluiert werden.

## 8. Schlussbetrachtung

### 8.1. Zusammenfassung

Das Ziel dieser Arbeit ist gewesen, eine Anwendung zu entwickeln, welche angehende Stomapatienten und deren Angehörige auf den Umgang mit einem Stoma vorbereitet. Dazu ist eine Anwendung entwickelt worden, bei welcher die Anwender die verschiedenen Abläufe erst mithilfe von Videos erlernen können, bevor sie die Abläufe selbst in der virtuellen Realität durchführen. Dabei haben die Anwender die Wahl zwischen der Patienten- und Angehörigensicht, sodass nicht nur die Patienten, sondern auch deren Angehörige die Abläufe lernen können. Dies ist wichtig, da viele Patienten ihre Stomaversorgung nicht selbst durchführen, sondern dies von Angehörigen übernommen wird.

Die Anwendung ist von neun Probanden getestet und insgesamt als gut bewertet worden. Daher kann davon ausgegangen werden, dass die Anwendung funktional ist und alle Abläufe durchgeführt werden können. Allerdings kann die Anwendungen noch in einigen Aspekten verbessert oder angepasst werden, welche im folgenden Kapitel beschrieben werden.

### 8.2. Ausblick

Eine Option, die Anwendung zu verbessern, wäre die Verwendung von Datenhandschuhen. Diese würden haptisches Feedback ermöglichen, wodurch Objekte in VR nicht nur sichtbar, sondern auch spürbar wären. Dadurch könnten unter anderem das Schneiden der Platte oder das Befestigen des Beutels verbessert werden, da beim Schneiden auch ein Widerstand spürbar wäre. Zusätzlich würden Datenhandschuhe es auch ermöglichen, die Fingerbewegung in VR zu übertragen, wodurch das Bodytracking und damit einhergehend die Immersion verbessert werden würde.

Eine weitere Möglichkeit, die Anwendung zu verbessern, wäre, für die zweite Trainingsphase eine andere Tonspur zu verwenden, bei welcher die einzelnen Schritte mit besserem Bezug auf die VR-Umsetzung erläutert werden. Dies würde die geäußerte Kritik der Testpersonen, die Anwendung sei nicht ausreichend gut erklärt worden, berücksichtigen. Neben diesen Aspekten können auch die Patientenmodelle weiter verbessert werden. So könnten weitere Modelle mit verschiedenen BMI-Werten verwendet werden. Somit können mehr Körpertypen abgedeckt werden und die Patienten bekommen immer ein Modell, das zu ihrem Körpertyp passt. Neben dem BMI könnten auch für Haut-, Haar- und Augenfarbe mehrere Alternativen bereitgestellt werden. Diese könnten durch einfaches

Durchwechseln der Texturen in der Anwendung verändert werden, sodass keine weiteren Modelle nötig wären. Die dadurch entstehende Variation des Aussehens der Modelle würde dafür sorgen, dass sich Patienten besser mit ihrem Avatar identifizieren können, wodurch die Immersion und der Lernfaktor erhöht werden. Des Weiteren könnten die Patientenmodelle in der Angehörigensicht mit Animationen versehen werden. Diese Animationen sollen nur leichte Bewegungen der Patienten wie beispielsweise Atmen, Blinzeln und leichte Kopfbewegungen sein. Sie sollten die Probanden nicht an den Aufgaben hindern, sondern nur dazu dienen, den Patienten etwas lebendiger erscheinen zu lassen.

Letztlich könnten auch einige der verwendeten Modelle und Texturen verbessert oder ausgetauscht werden. Es könnten auch weitere Objekte und Hintergrundgeräusche hinzugefügt werden, um die Umgebung lebendiger wirken zu lassen. Zusätzlich könnte auch *Postprocessing* verwendet werden. Dies soll dazu führen, dass die Umgebung realistischer wirkt.

Bevor diese Änderungen allerdings umgesetzt werden, wäre es zunächst nötig, die Wirkung der Anwendung mit echten Patienten zu testen. Insgesamt ist die Anwendung, sowohl von den Probanden, als auch von den beteiligten Ärzten des Kreisklinikums als positiv bewertet worden. Daraus kann geschlossen werden, dass die Anwendung einen großen Mehrwert für Stomapatienten bietet.

## Literatur

- [– Ea] PROLIFE enteron – Experten für Stoma. *Colostoma - Was ist das?* URL: <https://www.prolife-stoma.expert/colostoma#gsc.tab=0> (besucht am 10.09.2020).
- [– Eb] PROLIFE enteron – Experten für Stoma. *Ileostoma - Das sollten sie wissen!* URL: <https://www.prolife-stoma.expert/ileostoma#gsc.tab=0> (besucht am 10.09.2020).
- [360a] Omnia 360. *Immersion in Virtual Reality? Was ist das eigentlich?* URL: <https://omnia360.de/blog/was-ist-immersion/> (besucht am 14.09.2020).
- [360b] Omnia 360. *Motion Sickness: 6 Tipps, damit VR nicht zum Kotzen wird.* URL: <https://omnia360.de/blog/motion-sickness-in-virtual-reality/> (besucht am 17.09.2020).
- [Alm+18] Jonathan Almeida u. a. „Use of Virtual Reality Using Render Semi-realistic as an Alternative Medium for the Treatment of Phobias. Case Study: Arachnophobia“. In: *Applied Informatics*. Hrsg. von Hector Florez, Cesar Diaz und Jaime Chavarriaga. Cham: Springer International Publishing, 2018, S. 144–154. ISBN: 978-3-030-01535-0.
- [Ant] Antares. *ANTARES-Projekt*. URL: <https://antares.fokos.de/projekt/> (besucht am 18.01.2021).
- [BBr] BBraun. *Stoma: Was passiert mit mir und meinem Körper?n*. URL: <https://www.bbraun.de/de/patienten/stomaversorgung/mein-stoma.html#> (besucht am 29.11.2020).
- [Ble] Blender. *Blender.org - Home of the Blender project*. URL: <https://www.blender.org/> (besucht am 04.08.2020).
- [Bou+06] Stéphane Bouchard u. a. „Effectiveness of virtual reality exposure in the treatment of arachnophobia using 3D games“. In: *Technology and Health Care* 14 (2006). 1, S. 19–27. ISSN: 1878-7401. DOI: 10.3233/THC-2006-14103. URL: <https://doi.org/10.3233/THC-2006-14103>.
- [Coe+09] Carlos M. Coelho u. a. „The use of virtual reality in acrophobia research and treatment“. In: *Journal of Anxiety Disorders* 23.5 (2009), S. 563–574. ISSN: 0887-6185. DOI: <https://doi.org/10.1016/j.janxdis.2009>.

01.014. URL: <http://www.sciencedirect.com/science/article/pii/S0887618509000280>.

- [Coe+14] Tiago Coelho u. a. „Body Ownership of Virtual Avatars: An Affordance Approach of Telepresence“. In: *Innovative and Creative Developments in Multimodal Interaction Systems*. Hrsg. von Yves Rybarczyk u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 3–19. ISBN: 978-3-642-55143-7.
- [Cola] Coloplast. *SenSura Mio Click Ausstreifbeutel*. URL: <https://produkte.coloplast.de/coloplast/stomaversorgung/sensura-mio/sensura-mio-click/sensura-mio-click-ausstreifbeutel/> (besucht am 04.01.2020).
- [Colb] Coloplast. *SenSura Mio Click geschlossener Beutel*. URL: <https://produkte.coloplast.de/coloplast/stomaversorgung/sensura-mio/sensura-mio-click/sensura-mio-click-geschlossener-beutel/> (besucht am 04.01.2020).
- [CSD93] Carolina Cruz-Neira, Daniel J. Sandin und Thomas A. DeFanti. „Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE“. In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: Association for Computing Machinery, 1993, S. 135–142. ISBN: 0897916018. DOI: 10.1145/166117.166134. URL: <https://doi.org/10.1145/166117.166134>.
- [Deu] GHD GesundHeits GmbH Deutschland. *Stomaversorgung - wir sind für Sie da*. URL: <https://www.gesundheitsgmbh.de/stoma/#faq> (besucht am 10.09.2020).
- [Dör+19] Ralf Dörner u. a. „Einführung in Virtual und Augmented Reality“. In: *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Hrsg. von Ralf Dörner u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 1–42. ISBN: 978-3-662-58861-1. DOI: 10.1007/978-3-662-58861-1\_1. URL: [https://doi.org/10.1007/978-3-662-58861-1\\_1](https://doi.org/10.1007/978-3-662-58861-1_1).
- [Fic13] Andreas Fichtner. „Lernen für die Praxis: Das Skills-Lab“. In: *Simulation in der Medizin: Grundlegende Konzepte – Klinische Anwendung*. Hrsg. von Michael St.Pierre und Georg Breuer. Berlin, Heidelberg: Springer Berlin

- Heidelberg, 2013, S. 105–114. ISBN: 978-3-642-29436-5. DOI: 10.1007/978-3-642-29436-5\_10. URL: [https://doi.org/10.1007/978-3-642-29436-5\\_10](https://doi.org/10.1007/978-3-642-29436-5_10).
- [Gam] Epic Games. *Unity vs Unreal – which game engine is best for you?* URL: <https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html> (besucht am 03.08.2020).
- [Gun+18] Therese Gunn u. a. „The use of virtual reality simulation to improve technical skill in the undergraduate medical imaging student“. In: *Interactive Learning Environments* 26.5 (2018), S. 613–620. DOI: 10.1080/10494820.2017.1374981. eprint: <https://doi.org/10.1080/10494820.2017.1374981>. URL: <https://doi.org/10.1080/10494820.2017.1374981>.
- [Han] Prof. Dr. Dr. Ernst Hanisch. *Dickdarmkrebs (Kolonkarzinom)| Symptome und Therapie*. URL: <https://www.oncology-guide.com/erkrankung/dickdarmkrebs/> (besucht am 10.09.2020).
- [Har+19] Jeremy Hartmann u. a. „RealityCheck: Blending Virtual Environments with Situated Physical Reality“. In: Apr. 2019, S. 1–12. DOI: 10.1145/3290605.3300577.
- [Hof+04] Hunter G. Hoffman u. a. „Water-friendly virtual reality pain control during wound care“. In: *Journal of Clinical Psychology* 60.2 (2004), S. 189–195. DOI: <https://doi.org/10.1002/jclp.10244>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jclp.10244>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jclp.10244>.
- [igr] igroup. *igroup presence questionnaire (IPQ)*. URL: <http://www.igroup.org/pq/ipq/download.php#German> (besucht am 29.12.2020).
- [Ill] Beckman Institute Illinois Simulator Laboratory: University of Illinois at Urbana-Champaign. *CAVE*. URL: <http://www.isl.uiuc.edu/Labs/CAVE/CAVE.html> (besucht am 21.12.2020).
- [Int] ONKO Internetportal. *Der künstliche Darmausgang für Darmkrebspatienten*. URL: <https://www.krebsgesellschaft.de/onko-internetportal/basis-informationen-krebs/krebsarten/darmkrebs/anus-praeter-der-kuenstliche-darmausgang.html> (besucht am 29.11.2020).

- [Koc+20] Martin Kocur u. a. „Flexing Muscles in Virtual Reality: Effects of Avatars’ Muscular Appearance on Physical Performance“. In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. New York, NY, USA: Association for Computing Machinery, 2020, S. 193–205. ISBN: 9781450380744. URL: <https://doi.org/10.1145/3410404.3414261>.
- [Krea] Deutsches Krebsforschungszentrum. *Krebs: Was ist das? Wie entstehen Metastasen?* URL: <https://www.krebsinformationsdienst.de/tumorarten/grundlagen/krebsentstehung-faq.php> (besucht am 10.09.2020).
- [Kreb] Deutsches Krebsforschungszentrum. *Krebs: Was ist das? Wie entstehen Metastasen?* URL: <https://www.krebsinformationsdienst.de/tumorarten/grundlagen/krebsentstehung-faq.php#metastasen> (besucht am 10.09.2020).
- [Lin+20] Philip Lindner u. a. „Gamified, Automated Virtual Reality Exposure Therapy for Fear of Spiders: A Single-Subject Trial Under Simulated Real-World Conditions“. eng. In: *Frontiers in psychiatry* 11 (März 2020). PMC7069224[pmcid], S. 116–116. ISSN: 1664-0640. DOI: 10.3389/fpsy.2020.00116. URL: <https://doi.org/10.3389/fpsy.2020.00116>.
- [Mak] MakeHuman. *MakeHuman*. URL: <http://www.makehumancommunity.org/> (besucht am 02.08.2020).
- [Man+03] Fabrizia Mantovani u. a. „Virtual Reality Training for Health-Care Professionals“. In: *CyberPsychology & Behavior* 6.4 (2003). PMID: 14511451, S. 389–395. DOI: 10.1089/109493103322278772. eprint: <https://doi.org/10.1089/109493103322278772>. URL: <https://doi.org/10.1089/109493103322278772>.
- [Mik] Dimitri Mikhalechuk. *TESLASUIT Introduces its Brand-New VR-Gloves*. URL: <https://teslasuit.io/blog/teslasuit-introduces-its-brand-new-vr-gloves/> (besucht am 22.09.2020).
- [mim] mimerse. *Itsy*. URL: [https://www.oculus.com/experiences/gear-vr/1046954465328733/?locale=de\\_DE](https://www.oculus.com/experiences/gear-vr/1046954465328733/?locale=de_DE) (besucht am 24.01.2021).
- [Mül+00] W. Müller u. a. „VRATS – Virtual-Reality-Arthroskopie-Trainingssimulator“. In: *Der Radiologe* 40.3 (März 2000), S. 290–294. ISSN: 0033-832X. DOI: 10.1007/s001170050671. URL: <https://doi.org/10.1007/s001170050671>.

- [Mye01] Myeong-Sook Yoh. „The reality of virtual reality“. In: *Proceedings Seventh International Conference on Virtual Systems and Multimedia*. 2001, S. 666–674.
- [Nai+19] W. Nai u. a. „Estimating Forearm Axial Rotation Using Vive Trackers for Interaction With Serious Games“. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.9 (2019), S. 1893–1900.
- [Nik+05] C. Nikendei u. a. „Integriertes Skills-Lab-Konzept für die studentische Ausbildung in der Inneren Medizin“. In: *Deutsche Medizinische Wochenschrift* 130 (Apr. 2005), S. 1133–1138. DOI: 10.1055/s-2005-866799.
- [Noi] Noitom. *Hi5 VR GLOVE*. URL: <https://hi5vrglove.com/> (besucht am 22.09.2020).
- [Ocu] Oculus. *Oculus Quest*. URL: <https://www.oculus.com/quest/> (besucht am 13.12.2020).
- [Ocu] Oculus. *Oculus Rift*. URL: <https://www.oculus.com/rift/> (besucht am 13.12.2020).
- [Onl] Anatomy Online. *Stomapflege-Simulator*. URL: <https://www.anatomy-online.com/stomapflege-simulator-1112.html> (besucht am 29.11.2020).
- [Por+20] M Portelli u. a. „Virtual reality training compared with apprenticeship training in laparoscopic surgery: a meta-analysis“. In: *The Annals of The Royal College of Surgeons of England* 102.9 (Nov. 2020), S. 672–684. DOI: 10.1308/rcsann.2020.0178. URL: <https://doi.org/10.1308/rcsann.2020.0178>.
- [Rei+20] René Reinhard u. a. „Acting your avatar’s age: effects of virtual reality avatar embodiment on real life walking speed“. In: *Media Psychology* 23.2 (2020), S. 293–315. DOI: 10.1080/15213269.2019.1598435. eprint: <https://doi.org/10.1080/15213269.2019.1598435>. URL: <https://doi.org/10.1080/15213269.2019.1598435>.
- [Rie+10] Bernhard E. Riecke u. a. „Do We Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice“. In: *Spatial Cognition VII*. Hrsg. von Christoph Hölscher u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 234–247. ISBN: 978-3-642-14749-4.
- [Sab] Beat Saber. *Beat Saber*. URL: <https://beatsaber.com/> (besucht am 21.12.2020).

- [Sey08] Neal E. Seymour. „VR to OR: A Review of the Evidence that Virtual Reality Simulation Improves Operating Room Performance“. In: *World Journal of Surgery* 32.2 (Feb. 2008), S. 182–188. ISSN: 1432-2323. DOI: 10.1007/s00268-007-9307-9. URL: <https://doi.org/10.1007/s00268-007-9307-9>.
- [Sie] Kreisklinikum Siegen. *Klinik für Allgemein- und Viszeralchirurgie*. URL: <https://www.kreisklinikum-siegen.de/medizin-pflege/fachkliniken/klinik-fuer-allgemein-und-viszeralchirurgie/> (besucht am 15.01.2021).
- [Sin] Witmer und Singer. *PRESENCE QUESTIONNAIRE*. URL: <https://marketinginvolvement.files.wordpress.com/2013/12/pq-presence-questionnaire.pdf> (besucht am 29.12.2020).
- [Ske] VR Sketch. *Controls and input*. URL: <https://vrsketch.eu/docs-drawing.html> (besucht am 24.01.2021).
- [Str] Circuit Stream. *Unity vs Unreal Engine for XR Development: Which One Is Better?* URL: <https://circuitstream.com/blog/unity-vs-unreal/> (besucht am 06.10.2020).
- [Ult] Ultraleap. *Leap Motion Controller*. URL: <https://www.ultraleap.com/product/leap-motion-controller/> (besucht am 22.09.2020).
- [Unia] Unity. *2D 3D Game Creator & Editor*. URL: <https://unity.com/de/products/core-platform> (besucht am 03.08.2020).
- [Unib] Unity. *Unity - Scripting API:GameObject*. URL: <https://docs.unity3d.com/ScriptReference/GameObject.html> (besucht am 17.08.2020).
- [Unic] Unity. *Unity - Scripting API:MonoBehaviour.Start()*. URL: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.Start.html> (besucht am 17.08.2020).
- [Unid] Unity. *Unity - Scripting API:Rigidbody*. URL: <https://docs.unity3d.com/ScriptReference/Rigidbody.html> (besucht am 03.08.2020).
- [Unie] LMU Klinikum der Universität München. *Anatomie des Magen-Darm-Traktes*. URL: <http://www.klinikum.uni-muenchen.de/Klinik-fuer-Allgemein-Viszeral-Transplantations-Chirurgie/de/0600-fachgebiete/kolorektale/proktologie/MagenDarmTrakt/index.html> (besucht am 01.01.2021).
- [Viva] Vive. *HTC Vive*. URL: <https://www.vive.com/de/product/vive/> (besucht am 13.12.2020).

- [Vivb] Vive. *SPRENGE DIE GRENZEN DEINER VR CONTROLLER*. URL: <https://www.vive.com/de/accessory/vive-tracker/> (besucht am 02.01.2021).
- [Vivc] Vive. *Zertifiziertes vorgefertigtes VIVE VR-SYSTEM*. URL: <https://www.vive.com/de/product/vive/> (besucht am 13.12.2020).
- [VR] Anders VR. *Virtuelle Realität: Training. Atmung. Entspannung*. URL: <https://www.anders.life/> (besucht am 18.01.2021).
- [WKM19] Theresa F. Wechsler, Franziska Kämpers und Andreas Mühlberger. „Inferiority or Even Superiority of Virtual Reality Exposure Therapy in Phobias?—A Systematic Review and Quantitative Meta-Analysis on Randomized Controlled Trials Specifically Comparing the Efficacy of Virtual Reality Exposure to Gold Standard in vivo Exposure in Agoraphobia, Specific Phobia, and Social Phobia“. In: *Frontiers in Psychology* 10 (2019), S. 1758. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2019.01758. URL: <https://www.frontiersin.org/article/10.3389/fpsyg.2019.01758>.
- [Xin] XinReality. *Markerless inside-out tracking*. URL: [https://xinreality.com/wiki/Markerless\\_inside-out\\_tracking](https://xinreality.com/wiki/Markerless_inside-out_tracking) (besucht am 13.12.2020).
- [Zie] Dennis Ziesecke. *VR-Durchblick: Die Unterschiede zwischen Inside-Out und Outside-In*. URL: <https://vr-legion.de/hintergruende/vr-durchblick-die-unterschiede-zwischen-inside-out-und-outside-in/> (besucht am 13.12.2020).

# A. Anhang

## A.1. Persönliche Angaben



VR- Studie



Projektleitung:  
Universität Siegen, Lebenswissenschaftliche Fakultät  
Professur Medizinische Informatik und Mikrosystementwurf  
Prof. Dr. rer. nat. Rainer Brück  
Am Eichenhang 50  
57076 Siegen

Durchführung der Studie:  
Florian Gensing, Tanja Eiler, Dr.-Ing. Armin Grünewald

### *VR-FRAGEBOGEN*

#### Persönliche Angaben:

Alter: \_\_\_\_\_

Geschlecht: m  f  d

Größe: \_\_\_\_\_

Gewicht: \_\_\_\_\_

Pseudonym: \_\_\_\_\_

## A.2. Hygienekonzept

### Nutzungs- und Hygienekonzept zur Durchführung einer Studie mit Virtueller Realität (VR), XR-Labor AE - B 003

#### Übersicht

Um die Benutzerfreundlichkeit, Handhabung und Wirksamkeit von in Forschungsprojekten sowie in Projekten in der Lehre entwickelten VR-Anwendungen zu testen, sind Studien vor Ort mit einer unterschiedlichen Anzahl von Teilnehmer/innen notwendig. Da die VR-Anwendungen spezifische Hardware voraussetzen (leistungsstarker Rechner, VR-Brille), die die Proband/innen in der Regel nicht zu Hause besitzen, muss eine solche Studie im XR-Labor der Professur Medizinische Informatik und Mikrosystementwurf durchgeführt werden.

#### Maßnahmen

Um die in der Vorgabe „Regelungen für Forschung, Lehre und Verwaltung im Übergangsbetrieb“ vom 8. Mai 2020 genannten und daraufhin laufend aktualisierten Regeln umzusetzen, wurden für die Durchführung von Studien mit VR-Anwendungen folgende Maßnahmen getroffen:

##### Hygienemaßnahmen

Das Tragen eines Mundschutzes ist während der gesamten Dauer der Durchführung der Studie sowohl für die Studienleitung als auch für die Teilnehmenden verpflichtend. Die Hände werden beim Betreten des Gebäudes desinfiziert, wenn die Teilnehmer/innen am Eingang des Gebäudes abgeholt werden. Um für einen kontinuierlichen Luftaustausch im Raum zu sorgen, werden die Fenster während der Studie gemäß der Vorgaben vom 9. November 2020 alle 20 Minuten für die Dauer von 3 Minuten vollständig geöffnet.

Nach jeder Benutzung wird das verwendete Equipment (VR-Brille, VR-Controller) von der Studienleitung desinfiziert. Die Proband/innen müssen zudem Einmal-Masken (für den Augenbereich) tragen.

##### Sicherheitsabstand zwischen den Personen

Zwischen der Studienleitung und den Proband/innen wird jederzeit ein Mindestabstand von 1,5 m eingehalten. Damit dies möglich ist, müssen die Teilnehmenden das VR-Equipment selbstständig auf Anweisung anlegen und ablegen.

Während der Studie befindet sich neben der Studienleitung immer nur ein Proband/eine Probandin im Raum.

##### Protokollierung der Teilnehmer/innen

Die Anwesenden werden zu Beginn einer jeden Sitzung protokolliert und in die „Kontaktliste zur Teilnehmer-/Besucherdokumentation während der Corona-Pandemie“ eingetragen. Die gesammelten Listen werden an die dafür eingerichtete E-Mail-Adresse des Dekanats geschickt und anschließend vernichtet.

##### Durchführungszeitraum der Studie

Die Studie zur Stoma-Trainingsanwendung wird vom 07.12. bis 16.12. im Raum AE - B 003 durchgeführt werden.

## Kontaktmöglichkeiten für Rückfragen

**Lehrstuhlleitung:**

Name: Prof. Dr. Rainer Brück

Professur: Medizinische Informatik und  
Mikrosystementwurf

E-Mail: [Rainer.Brueck@uni-siegen.de](mailto:Rainer.Brueck@uni-siegen.de)

Telefon: 0271/ 740 2375

Büro: AE-D 206

**Laborverantwortlicher:**

Name: Dr.-Ing. Armin Grünewald

Professur: Medizinische Informatik und  
Mikrosystementwurf

E-Mail: [Armin.Gruenewald@uni-siegen.de](mailto:Armin.Gruenewald@uni-siegen.de)

Telefon: 0271/ 740 2670

Büro: AE-C 204

## A.3. Einwilligungserklärung



VR- Studie



Projektleitung:  
Universität Siegen, Lebenswissenschaftliche Fakultät  
Professur Medizinische Informatik und Mikrosystementwurf  
Prof. Dr. rer. nat. Rainer Brück  
Am Eichenhang 50  
57076 Siegen

Durchführung der Studie:  
Florian Gresing, Tanja Eiler, Dr.-Ing. Armin Grünewald

# Einwilligungserklärung zur Erhebung und Verarbeitung personenbezogener Daten im Rahmen eines Forschungsprojekts

Sehr geehrte/r Proband/in,

die nachfolgend in Eckpunkten beschriebene Studie wurde Ihnen von den Projektmitarbeiter/innen erläutert. Es werden noch einmal die Ziele und der Ablauf aufgezeigt. Bitte sprechen Sie den Projektleiter/die Projektleiterin oder die Mitarbeiter/innen gerne an, um alle Punkte anzusprechen, die Ihnen unklar und/oder wichtig sind.

### Warum wird diese Studie durchgeführt?

Wir möchten herausfinden, inwieweit der Umgang mit Stoma mit Hilfe einer Trainingsanwendung erleichtert werden kann. Das Ziel der Anwendung ist, das Beutel leeren, Beutel wechseln und das Wechseln der Platte sowohl aus Sicht der Patientinnen und Patienten als auch der Angehörigen zu erlernen. Das Training besteht aus vier verschiedenen Stufen, die von Erklärvideos bis hin zum Durchlaufen einer Virtual Reality (VR) Anwendung reichen.

### Datenschutzinformation

Mir ist bekannt, dass bei dieser Studie persönliche Daten (Geschlecht, Alter, Größe Gewicht) über mich erhoben, gespeichert und ausgewertet werden sollen. Die Daten werden u.a. dafür verwendet werden, um einen zu Ihnen passenden virtuellen Charakter zu generieren. Die Verwendung der Daten setzt vor der Teilnahme an der Studie die nachfolgende Einwilligungserklärung voraus, das heißt, dass mit der Teilnahme an der Studie auch die nachfolgende Einwilligung verbunden ist.

1. Ich erkläre mich damit einverstanden, dass im Rahmen dieser Studie Angaben über meine Person erhoben und verarbeitet werden und sowohl in Papierform sowie auf elektronischen Datenträgern aufgezeichnet werden. Soweit erforderlich, dürfen die erhobenen Daten im Rahmen einer wissenschaftlichen Auswertung und Veröffentlichung in anonymer Form (es werden nur die genannten Daten aufgenommen) weiterverwendet werden.
2. Ich bin darüber aufgeklärt worden, dass ich jederzeit die Teilnahme an der Studie beenden kann und die hier gegebene Einwilligung jederzeit mit Wirkung für die Zukunft widerrufen kann (z.B. per Mail an: [Armin.Gruenewald@uni-siegen.de](mailto:Armin.Gruenewald@uni-siegen.de)). Ich weiß, dass im Falle eines Widerrufs zur Teilnahme an der Studie die

bis zu diesem Zeitpunkt gespeicherten Daten weiterhin verwendet werden dürfen, soweit dies erforderlich ist, um statistische Auswertungen zu generieren.

3. Ich bin über folgende gesetzliche Regelung informiert: Falls ich meine Einwilligung, an der Studie teilzunehmen, widerrufe, müssen alle Stellen, die meine personenbezogenen Daten gespeichert haben, unverzüglich prüfen, inwieweit die gespeicherten Daten für die in Nr. 2 genannten Zweck noch erforderlich sind. Nicht mehr benötigte Daten sind unverzüglich zu löschen.

Mit Ihrer Unterschrift bestätigen Sie, dass Sie die Informationen, die über Zweck und Inhalte der Studie informieren sowie die Informationen bezüglich des Datenschutzes, gelesen, den Ablauf der Studie verstanden haben und mit diesen Vorgängen einverstanden sind.

Des Weiteren bestätigen Sie, dass keine medizinischen Bedenken gegen Ihre Teilnahme bestehen und dass Sie eigenverantwortlich und freiwillig an der Studie teilnehmen.

Außerdem bestätigen Sie mit Ihrer Unterschrift, dass Sie Kenntnis davon erlangt haben, dass für die Folgen Ihrer Teilnahme an der Studie keinerlei Haftung übernommen wird.

Die Teilnahme an dieser Studie ist freiwillig. Sie können jederzeit ohne Angaben von Gründen die Teilnahme beenden und ohne dass Ihnen dadurch ein Nachteil entsteht.

Sollten Sie weitere Fragen bezüglich der Studie haben, können Sie sich gerne an die Testleiter/innen wenden.

Teilnahme an der Studie?  Ja  Nein

Vor- und Nachname .....

Siegen, den .....

.....  
Unterschrift Proband/in

## A.4. Fragebogen

Pseudonym: \_\_\_\_\_

Hier sind einige Vorschläge, die auf die Erfahrungen, die Sie soeben gemacht haben, angewendet werden können. Bitte geben Sie an, ob jeder dieser Vorschläge auf Ihre Erfahrungen zutrifft oder nicht. Es gibt keine richtige oder falsche Antwort, nur Ihre Meinung ist wichtig.

Sie werden feststellen, dass einige Fragen ähnlich sind. Dies ist aus statistischen Gründen notwendig. Denken Sie daran, dass diese Fragen nur mit Bezug auf die Erfahrung beantwortet werden sollen, die Sie soeben gemacht haben.

**In der computererzeugten Welt hatte ich den Eindruck, dort gewesen zu sein...**

Überhaupt nicht                      -3    -2    -1    0    +1    +2    +3                      Sehr stark

**Ich hatte das Gefühl, dass die virtuelle Umgebung hinter mir weitergeht.**

Trifft gar nicht zu                      -3    -2    -1    0    +1    +2    +3                      Trifft völlig zu

**Ich hatte das Gefühl, nur Bilder zu sehen.**

Trifft gar nicht zu                      -3    -2    -1    0    +1    +2    +3                      Trifft völlig zu

**Ich hatte das Gefühl, in dem virtuellen Raum zu sein.**

Hatte nicht das Gefühl                      -3    -2    -1    0    +1    +2    +3                      Hatte das Gefühl

**Ich hatte das Gefühl, in dem virtuellen Raum zu handeln statt etwas von außen zu bedienen.**

Trifft gar nicht zu                      -3    -2    -1    0    +1    +2    +3                      Trifft völlig zu



**Wie real erschien Ihnen die virtuelle Welt?**

Wie eine vorgestellte Welt Nicht zu unterscheiden  
von der realen Welt

-3   -2   -1   0   +1   +2   +3

**Die virtuelle Welt erschien mir wirklicher als die reale Welt.**

Trifft gar nicht zu Trifft völlig zu

-3   -2   -1   0   +1   +2   +3

**Wie viel konnten Sie kontrollieren?**

Nichts Alles

-3   -2   -1   0   +1   +2   +3  
Etwas

**Wie reagierte die Umgebung auf Aktionen, die Sie initiiert (oder durchgeführt haben)?**

Gar nicht Vollkommen

-3   -2   -1   0   +1   +2   +3  
Mäßig

**Wie natürlich wirkte Ihre Interaktion mit der Umwelt?**

Sehr künstlich Vollkommen natürlich

-3   -2   -1   0   +1   +2   +3

**Wie überzeugend war Ihr Gefühl, dass sich Objekte durch den Raum bewegen?**

Gar nicht Sehr überzeugend

-3   -2   -1   0   +1   +2   +3

**Konnten Sie voraussehen, was als Reaktion auf die von Ihnen durchgeführten Aktionen als nächstes passieren würde?**

Gar nicht Vollkommen

-3   -2   -1   0   +1   +2   +3  
Mäßig

**Wie überzeugend war Ihr Gefühl, sich innerhalb der virtuellen Umgebung zu bewegen?**

Gar nicht Sehr überzeugend

-3   -2   -1   0   +1   +2   +3  
Mäßig



Verspüren Sie Nebenwirkungen, welche durch die VR-Erfahrung ausgelöst wurden (z.B. Kopf- oder Augenschmerzen, Übelkeit etc.)?

Ja

Nein

Falls ja, welche:

---

---

---

Hatten Sie bereits Erfahrung mit VR?

Ja

Nein

Falls ja, nutzen Sie VR regelmäßig?

Ja

Nein

Wobei hatten Sie den größten Lerneffekt für die einzelnen Stoma-Schritte?

Die Videos

Das Anklicken der Reihenfolge

Durchlaufen in VR

Teilen Sie uns Ihre Meinung mit: Haben Sie noch Vorschläge oder Kritiken?