

# Bachelorarbeit

Fabian Meinhardt

Entwicklung eines Exergames für Trainingseinheiten in  
Virtual Reality mit einem Rudergerät als  
Interaktionsmedium

Fabian Meinhardt

Entwicklung eines Exergames für Trainingseinheiten  
in Virtual Reality mit einem Rudergerät als  
Interaktionsmedium

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt  
Zweitgutachter: Prof. Dr. Philipp Jenke

Eingereicht am: 19. September 2019

**Fabian Meinhardt**

**Thema der Arbeit**

Entwicklung eines Exergames für Trainingseinheiten in Virtual Reality mit einem Rudergerät als Interaktionsmedium

**Stichworte**

Virtuelle Realität, Exergame, Gamification, Rudern, Training

**Kurzzusammenfassung**

Diese Arbeit befasst sich mit der Entwicklung eines VR-Exergames für das Rudertraining mit einem realen Rudergerät als Interaktionsmedium. Ziel ist es, die Motivation zum Training mit mehreren Trainingsmodi und einem intuitiven Bedienkonzept zu steigern. Sensoren, die eine Gewichtsverlagerung auf dem Rudergerät messen, erweitern die Möglichkeiten zur Steuerung. Ein abschließendes Experiment mit einigen Testpersonen zeigt, dass das Bedienkonzept trägt und einige Leute durchaus bereit wären, mit einem VR-Exergame zu trainieren.

**Fabian Meinhardt**

**Title of Thesis**

Development of an exergame for training units in virtual reality with a rowing machine as interaction medium

**Keywords**

Virtual Reality, Exergame, Gamification, Rowing, Workout

**Abstract**

This thesis deals with the development of a VR Exergame for rowing training, using a real rowing machine as interaction medium. The aim is to increase motivation with several training modes and an intuitive operating concept. Sensors, which measure a weight shift on the rowing machine, extend the possibilities for control. A final experiment with some test subjects shows that the operating concept is supportive and that some people would be quite willing to train with a VR Exergame.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>vi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	2
1.3 Gliederung . . . . .	2
<b>2 Grundlagen und vergleichbare Arbeiten</b>	<b>3</b>
2.1 Zentrale Begriffe . . . . .	3
2.2 Vergleichbare Arbeiten . . . . .	8
2.2.1 Virtuelle Umgebungen beim Rudern . . . . .	9
2.2.2 Exergaming Simulator . . . . .	11
2.2.3 Arcade Klassiker als VR Umsetzung . . . . .	13
2.2.4 Gamification im Heimtraining . . . . .	15
2.3 Zusammenfassung . . . . .	17
<b>3 Funktionale und nicht funktionale Anforderungen</b>	<b>18</b>
3.1 Funktionale Anforderungen . . . . .	18
3.1.1 Sprintlevel . . . . .	18
3.1.2 Parcourslevel . . . . .	19
3.1.3 Piratenlevel . . . . .	21
3.1.4 Interaktion . . . . .	22
3.1.5 Simulation eines Ruderbootes . . . . .	23
3.2 Nicht funktionale Anforderungen . . . . .	24
3.3 Zusammenfassung . . . . .	25
<b>4 Entwurf</b>	<b>26</b>
4.1 Systemübersicht . . . . .	26
4.2 Komponentenübersicht . . . . .	27

4.3	Abbildung auf das Unity Programmiermodell . . . . .	31
4.4	Umsetzung . . . . .	34
4.4.1	Interaktion . . . . .	34
4.4.2	Sensoreinheit . . . . .	38
4.4.3	Ruder . . . . .	40
4.4.4	Ruderboot . . . . .	41
4.4.5	Level . . . . .	42
4.4.6	Trainingsmodi . . . . .	45
4.5	Zusammenfassung . . . . .	47
<b>5</b>	<b>Experimentelle Überprüfung</b>	<b>48</b>
5.1	Versuchsaufbau . . . . .	48
5.2	Auswertung . . . . .	49
5.3	Zusammenfassende Einschätzung . . . . .	52
<b>6</b>	<b>Zusammenfassung</b>	<b>53</b>
	<b>Literaturverzeichnis</b>	<b>55</b>
	<b>Selbstständigkeitserklärung</b>	<b>58</b>

# Abbildungsverzeichnis

4.1	Verteilungsdiagramm zur Übersicht über die eingesetzte Hardware und der Verteilung der Software (Quelle: eigene Arbeit) . . . . .	27
4.2	Komponentendiagramm zur Übersicht des Zusammenspiels der verwendeten Komponenten (Quelle: eigene Arbeit) . . . . .	28
4.3	Kräfte und Richtungen die auf das Ruderboot wirken (Quelle: eigene Arbeit)	30
4.4	GameObject-Sicht einer Szene im Unity Editor . . . . .	33
4.5	Zustandsautomat über Zustände und Übergänge, zur Bestimmung, ob eine Ruderbewegung durchgeführt wird (Quelle: eigene Arbeit) . . . . .	34
4.6	Kurve einer quadratischen Funktion, welche die x-Achse an den Punkten Null und Eins schneidet. Die x-Achse steht für den Ruderfortschritt und die y-Achse für die Kraft. y ist normalisiert, damit es mit einem variablen Kraftwert multipliziert werden kann. (Quelle: eigene Arbeit) . . . . .	37
4.7	Verkabelung der Drucksensoren und Widerstände mit dem Arduino Uno (Quelle: eigene Arbeit) . . . . .	39
4.8	Verteilung der Sensoren auf dem Sitz des Rudergerätes . . . . .	39
4.9	Befestigung des Arduino und der Kabel auf der Rückseite des Sitzes vom Rudergerätes . . . . .	39
4.10	Klassisches Ruder . . . . .	40
4.11	Invertiertes Ruder . . . . .	40
4.12	Sprintlevel . . . . .	43
4.13	Parcourslevel . . . . .	43
4.14	Piratenlevel . . . . .	43
4.15	Beispielhafte Darstellung der Bereiche, in denen das Schiff Kanonenkugeln abfeuert . . . . .	44
4.16	UI während des Trainings im Sprintlevel . . . . .	46
4.17	UI nach Erreichen des Ziels im Sprintlevel . . . . .	46
5.1	Balkendiagramm über Fragen zur allgemeinen Einstellung zum Training .	49

5.2	Balkendiagramm über Fragen zur Nutzbarkeit der Anwendung . . . . .	50
5.3	Balkendiagramm über Fragen zur Lenkung . . . . .	51
5.4	Balkendiagramm über Fragen zur Eignung der Level . . . . .	52

# 1 Einleitung

## 1.1 Motivation

Virtual Reality birgt das Potenzial ein sehr hohes Maß an Immersion in der Präsentation einer virtuellen Welt zu erzeugen. So können etwa virtuelle Besuche eines Museums oder auch ein Horrorspiel wesentlich realistischer bzw. intensiver erlebt werden als bei einer klassischen Ausgabe über einen Monitor. Auch Exergames und Fitnessspiele können davon profitieren, da sich der Nutzer mehr auf die spielerischen Aspekte konzentrieren kann und die Wahrnehmung des Trainings weiter in den Hintergrund rückt. Auch die Art der Interaktion in der virtuellen Umgebung trägt ihren Teil dazu bei. So lassen sich die Controller eines VR-Systems im Raum orten und ermöglichen räumliche Freiheit und Interaktionsmöglichkeiten, die eine VR-Anwendung realistischer erscheinen lassen. Allerdings sind Controller nicht für jeden Einsatzzweck optimal, sodass spezielle Eingabemedien zu wählen oder entwickeln wären. So eignet sich beispielsweise ein Lenkrad besser für eine Rennsimulation oder ein Rudergerät besser für eine Rudersimulation, da sowohl die Haptik als auch das Feedback der realen Objekte besser zu den jeweiligen Anwendungsfällen passen.

Eine Einbindung von Sportgeräten in eine virtuelle Umgebung bietet die Möglichkeit eine immersive Welt zu präsentieren, welche sich auf das jeweilige Gerät abstimmen ließe. So könnten etwa reale wie auch fiktive Orte auf einem Fahrrad, einem Ruderboot oder auch zu Fuß mit einem Laufband bereist werden. Herausforderungen, Wettkampfsysteme und andere abwechslungsreiche Inhalte würden das Trainingserlebnis noch erweitern und könnten durch den Unterhaltungsfaktor möglicherweise die Motivation zum Training für einige Menschen erhöhen.

Durch die wählenden Fortschritte im Bereich der VR-Systeme können diese mittlerweile zu erschwinglichen Preisen erworben werden. Diese Situation bietet Potenzial, dass VR-Systeme auch im Heimtraining verstärkt für sportliche Zwecke eingesetzt werden könnten,



sodass Menschen, die weniger Sport treiben oder sich nicht für den klassischen Sport interessieren, neue Anreize für den Sport geboten würden.

### 1.2 Zielsetzung

Im Rahmen dieser Bachelorarbeit soll ein VR-Exergame entwickelt werden, das es ermöglicht mit einem Rudergerät in einer virtuellen Umgebung zu trainieren. Zu diesem Zweck soll eine Umgebung geschaffen werden, in welcher der Nutzer mit einem Ruderboot virtuelle Seen und Flüsse befahren kann. Hierbei soll es mehrere Level geben, die unterschiedliche Aufgaben und Herausforderungen bieten und bei der Durchführung des Trainings unterschiedliche Schwerpunkte setzen.

Die Trainingsbewegung auf dem Rudergerät soll getrackt und zum Antreiben des Ruderbootes genutzt werden. Darüber hinaus sollen Sensoren in der Sitzfläche des Rudergerätes verbaut werden, die messen sollen, wie stark sich jemand zur Seite neigt. Mit diesen Werten soll eine Lenkung in der Anwendung realisiert werden, die eine räumliche Bewegung (Translation entlang der x- und z-Achse sowie Rotation entlang der y-Achse) ermöglicht.

Zum Schluss soll eine Überprüfung mit einigen Testpersonen stattfinden, die einige Aspekte der Anwendung bewerten und eine Einschätzung über Fragen zur Motivation und Akzeptanz der Anwendung erlauben.

### 1.3 Gliederung

In Kapitel 2 werden zunächst einige Begriffe vorgestellt, welche für das Thema dieser Arbeit relevant sind. Eine anschließende Einordnung der Arbeit sowie eine Vorstellung und Abgrenzung von vergleichbaren Arbeiten, sollen ein genaueres Bild von der geplanten Anwendung vermitteln. In Kapitel 3 werden die funktionalen, sowie die nicht funktionalen Anforderungen an die Anwendung definiert. Kapitel 4 befasst sich mit der Umsetzung der Anwendung, bestehend aus einem Systementwurf und einer detaillierten Beschreibung der realisierten Komponenten. In Kapitel 5 werden die Fragen und Ergebnisse der Überprüfung vorgestellt und anschließend ausgewertet. Zum Schluss findet sich in Kapitel 6 eine Zusammenfassung, die nochmals die wichtigsten Inhalte der Arbeit hervorhebt und einen Ausblick auf mögliche weitere Entwicklungsschritte gibt.

## 2 Grundlagen und vergleichbare Arbeiten

In diesem Kapitel werden einige vergleichbare Arbeiten sowie Grundlagen vorgestellt, welche für das Thema dieser Arbeit relevant sind. In 2.1 werden zentrale Begriffe erläutert, welche häufiger Verwendung finden werden. In 2.2 werden vergleichbare Arbeiten vorgestellt, angefangen mit einer Zusammenfassung von Versuchs- und Studienergebnissen, gefolgt von Einzelvorstellungen relevanter vergleichbarer Arbeiten inklusive einer Beschreibung des Projektaufbaus und einer Abgrenzung zu dieser Arbeit.

### 2.1 Zentrale Begriffe

Im Folgenden werden die Begriffe *Exergames*, *Gamification*, *Virtual Reality*, *Immersion* und *Cybersickness* erläutert und deren Bezug zu dieser Arbeit dargelegt.

#### **Exergames**

Bei dem Begriff „Exergame“, bzw. der Tätigkeit des „Exergaming“, handelt es sich um eine Kombination der beiden englischen Wörter „exercise“ und „video games“.[24] Es geht also um Videospiele, bei welchen Training und Bewegung eine wichtige Rolle spielen. Es gab ursprünglich viel Inkonsistenz über den Begriff, etwa da die Definition von „exercise“ nicht auf alle üblichen Exergames zutreffen würde, da es sich nicht immer um ein direktes Training handelt. Auch wurde körperliche Aktivität häufig zur Beschreibung dieses Begriffes verwendet, obwohl dies auch auf u.a. sitzende Aktivitäten anwendbar wäre. Angesichts dieser Problematiken, wird in [14] das „Exergaming“ definiert als eine experimentelle Aktivität, in welcher das Spielen eine körperliche Anstrengung oder Bewegungen erfordert, die über das sitzen hinausgeht und außerdem Kraft-, Gleichgewichts- und Bewegungsaktivitäten beinhaltet.

Ein wichtiger Faktor für die Entwicklung eines Exergames ist, dass es für den Benutzer ansprechend gestaltet wird. Dazu sollte es dem Spieler z.B. eine gute Balance zwischen

Können und Herausforderung bieten. Ebenfalls ist es wichtig, dass das Spielen die Effektivität einer regulären Trainingsübung hat.[17]

Einige bekanntere Beispiele für Exergames sind etwa Wii Fit<sup>1</sup> und Just Dance<sup>2</sup>.

### **Gamification**

Der Begriff „Gamification“ beschreibt im allgemeinen meist die Verwendung von spielerischen Elementen in einem nicht spielerischen Kontext. So fallen beispielsweise Umsetzungen von Belohnungen, Bestenlisten und Punktesystemen, z.B. in einem sportlichen Umfeld, unter diesen Begriff. Das Ziel ist es hierbei, den Spaß an der Nutzung eines Produkts zu erhöhen, basierend auf der erhöhten Motivation und dem Spaß, den ein Benutzer oft beim Spielen eines Spiels zeigt.

Nach einer weiteren Abgrenzung in [5][4] kann eine Anwendung zusätzlich noch in die Kategorien „Gamification“, bzw. „Gameful Design“, „Serious Games“ und Playful Interaction“ eingeordnet werden. Gamification legt hierbei den Fokus darauf, nur einzelne Elemente für den nicht spielerischen Kontext zu nutzen, wohingegen es sich bei Serious Games eher um ganze Spiele handelt, welche für einen nicht unterhaltenden Kontext design sind. Bei der dritten Kategorie handelt es sich um Spielelemente, bzw. spielerische Interaktionen, welche Merkmale aufweisen die nicht charakteristisch für Spiele sind.

Im Allgemeinen wirkt sich der Einsatz von Gamification in der Regel positiv auf die Motivation des Nutzers aus, diese Erfahrung kann aber auch je nach Anwendungsbe- reich und Nutzer variieren. Zum Beispiel wollen einige Benutzer auf der Spitze einer Bestenliste erscheinen, wohingegen es anderen gerade mal darauf ankommt überhaupt auf der Liste vertreten zu sein, unabhängig vom Rang. [8] Aber auch die verwendete Gamification-Technik kann einen unterschiedlich starken Einfluss auf die Nutzer haben. So erwiesen sich in [21] in einer sportlichen Anwendung Herausforderungen als die effek- tivste Gamification-Technik, wohingegen Preise als Belohnung keinen relevanten Einfluss auf die Nutzer hatten.

### **Virtual Reality**

Der Begriff „Virtual Reality“ oder zu deutsch „virtuelle Realität“ besteht aus zwei Wörtern unterschiedlicher Bedeutung. So steht virtuell für etwas nicht echtes, bzw. etwas das

---

<sup>1</sup><https://www.nintendo.de/Spiele/Wii/Wii-Fit-283894.html> zuletzt aufgerufen am 05.05.2019

<sup>2</sup><https://www.ubisoft.com/en-us/franchise/just-dance/> zuletzt aufgerufen am 05.05.2019

nicht in der Wirklichkeit vorhanden ist, aber echt erscheint. Realität hingegen bedeutet Wirklichkeit. Eine Zusammensetzung dieser Begriffe ergäbe etwa: eine Wirklichkeit, die nicht in der Wirklichkeit vorhanden ist.[3] Der Begriff beschreibt also eine Wirklichkeit bzw. Realität, welche nicht echt ist, aber echt erscheint.

Virtual Reality wird heute meist als eine computergenerierte 3D-Umgebung bezeichnet, die Bildinhalte auf einem Head Mounted Display (HMD) oder in einer Umgebung mit Großleinwänden (CAVE) präsentiert. Entscheidend ist hierbei, dass die Umgebung den Nutzer umgibt und auf natürliche Weise auf ihn reagiert, weshalb Virtual Reality auch als Mensch-Maschinen-Schnittstelle bezeichnet wird. Für die Interaktion mit der virtuellen Umgebung dienen Eingabegeräte wie etwa Controller oder Datenhandschuhe. [6]

Eine Definition, welche sich nicht auf spezifische Wiedergabemedien bezieht und damit von der aktuellen Technik unabhängig ist, lautet wie folgt: „Virtual Reality (VR) ist ein Sammelbegriff für mediale Strategien, die darauf abzielen, bei Rezipient/innen die Illusion auszulöschen, sich tatsächlich an einem vom Computer simulierten Ort zu befinden (Ortsillusion).“[3] Hierzu sollte die Simulation die Position und Handlungen der Nutzer erfassen und Feedback geben, welches einen oder mehrere Sinne ersetzt oder ergänzt.[16] Ob die virtuelle Umgebung einer realen Umgebung gleicht oder rein fiktionalen Erfahrungen bietet ist jedoch irrelevant. Wichtig ist, dass ein gelungenes Maß an Immersion erreicht wird.

Eins der größten Anwendungsgebiete für Virtual Reality ist die Unterhaltungsindustrie. Hier gibt es für HTC Vive<sup>3</sup>, Oculus Rift<sup>4</sup> und Playstation VR<sup>5</sup> bereits ein großes Repertoire an Videospiele, die sich über viele Genres erstrecken. Darüber hinaus gibt es weitere Anwendungsgebiete, wie z.B. Simulationen in der Aus- und Weiterbildung, die Konstruktion und der Bau von Prototypen in der Industrie und viele Weitere.

### **Immersion**

Menschen sind in der Lage sich umfangreiche Geschichten und Welten auszudenken und diese in Medien wie etwa Büchern zu manifestieren, meistens mit dem Ziel sie mit anderen Menschen zu teilen. Abhängig von den Fähigkeiten des Autors, eine glaubwürdige und konsistente Welt zu erzählen, ist es dem Konsumenten möglich in die Geschichte und die dargestellte Welt einzutauchen. Dieses Eintauchen in eine Umgebung, bzw. das Gefühl

---

<sup>3</sup><https://www.vive.com/de/> zuletzt aufgerufen am 07.05.2019

<sup>4</sup><https://www.oculus.com/> zuletzt aufgerufen am 07.05.2019

<sup>5</sup><https://www.playstation.com/de-de/explore/playstation-vr/> zuletzt aufgerufen am 07.05.2019

sich in einer Umgebung zu befinden, sodass diese als real empfunden wird, ist es was der Begriff Immersion beschreibt.[16]

Darüber hinaus wird in [16] auch noch zwischen mentaler und physischer Immersion differenziert. Mentale Immersion bezeichnet den Zustand tief in etwas eingebunden zu sein, also sich gedanklich in einer anderen Umgebung zu befinden, ein Zustand den die meisten Unterhaltungsmedien zu erreichen versuchen. Bei physischer Immersion hingegen handelt es sich um das körperliche Betreten eines Mediums und das Stimulieren einiger körperlicher Sinne mithilfe von Technologie. Sie beschreibt also Eigenschaften welche mithilfe von VR-Technik erfüllt werden. Der Unterschied zwischen klassischen Medien und virtueller Realität ist also, dass man sich bei ersterem in die präsentierte Welt hineindenkt und bei zweiterem die Welt mithilfe physischer Sinne erfährt.

Laut einer technischeren Beschreibung in [18], beschreibt Immersion wie fähig eine Technik ist, eine Illusion der Realität zu liefern. Abhängig ist dies von 4 Punkten. So soll 1. eine Isolation von der echten Welt stattfinden, Sinneseindrücke sollen also von einem Computer erzeugt werden, außerdem sollen 2. möglichst viele Sinne angesprochen werden. 3. sollen Ausgabegeräte ein großes Panorama-Sichtfeld bieten, wie es etwa bei einer CAVE Umgebung oder einem HMD der Fall ist und 4. sollen Ausgabegeräte eine lebendige Darstellung bieten, etwa durch hohe Auflösung und gute Farbwiedergabe.

Abseits der Hardware ist es die Aufgabe der virtuellen Umgebung, den Benutzer in die simulierte Welt eintauchen zu lassen. Faktoren wie eine natürliche Interaktion mit der virtuellen Umgebung oder die Wahrnehmung der eigenen Bewegung beeinflussen den Grad der Immersion, den der Benutzer verspürt.[23]

Im Zusammenhang mit Immersion fällt auch öfter der Begriff „Präsenz“. Präsent sein bedeutet in diesem Kontext, dass das Bewusstsein denkt man befinde sich an dem Ort, welchen die virtuelle Umgebung einem präsentiert.[18] Aus einem stärkeren Gefühl von Immersion, folgt zudem ein höheres Level an Präsenz.[23]

### **Cybersickness**

Cybersickness beschreibt ein Unwohlsein welches bei der Nutzung von Virtual Reality als Begleiterscheinung auftreten kann und schon seit den Anfängen von VR ein Problem darstellt. Nach der „Sensory-Conflict-Theory“, der gängigsten Theorie in diesem Feld, tritt Cybersickness auf, wenn der visuelle Sinn und der Gleichgewichtssinn widersprüchliche Informationen an das Gehirn senden. So etwas kann beispielsweise auftreten, wenn man

sich in der virtuellen Szene bewegt, etwa bei einer Fahrt in einem Auto, sich aber nicht in der Realität bewegt und die Informationen die der Gleichgewichtssinn einem liefert mit den visuellen Informationen in Konflikt stehen. Dies kann zu Symptomen wie Übelkeit, Schwindelgefühl oder Augenbelastung führen.[2][10]

Das Auftreten von Cybersickness kann von Person zu Person stark variieren, es gibt jedoch mehrere Faktoren die einen Einfluss auf das Auftreten haben. Einen dieser Faktoren stellt der Nutzer selbst dar, so sind Menschen je nach Alter, Geschlecht und auch Krankheit unterschiedlich anfällig für Cybersickness.[2][10] Des Weiteren spielen das Design der Umgebung, sowie die auszuführenden Aufgaben eine Rolle für das Auftreten. Wenn eine Person etwa mehr Kontrolle in der virtuellen Umgebung hat, ist auch die Wahrscheinlichkeit für Cybersickness geringer. Zuletzt stellt noch die verwendete Technik einen wesentlichen Anteil an Risiken dar. So können schon geringe Latenzen oder Ungenauigkeiten beim Tracking zu starken Unterschieden der Sinneswahrnehmungen führen. Auch eine fehlerhafte Kalibrierung des Ausgabegerätes oder ein flackerndes Bild können zu Cybersickness führen. [2]

### **Bezug zur Arbeit**

Die in dieser Arbeit beschriebene VR-Ruderanwendung soll sportliche Aktivität und den Spaßfaktor einer spielerischen virtuellen Umgebung verbinden und erfüllt damit die Voraussetzungen, um als Exergame eingestuft zu werden. Zu diesem Zweck sollen Gamification Elemente, wie etwa unterschiedliche Spielmodi und Bestzeiten zum Einsatz kommen. Nach der Abgrenzung von Gamification in [5], wäre die Anwendung als Serious Game einzuordnen.

Da Immersion für die Akzeptanz einer VR-Anwendung wichtig ist, soll eine virtuelle Umgebung entstehen, die natürliche Kulissen bietet und durch ausreichend visuelle und akustische Rückmeldungen ein realistisches Spielgefühl ergibt. Der Einsatz eines HMDs, sowie eines haptischen Trainingsgerätes, trägt durch das vollständige Umgeben des Spielers und der Ermöglichung der Interaktion, ihren Anteil zur physischen Immersion bei.

Cybersickness stellt beim Einsatz von VR ein mögliches Problem dar, wogegen möglichst gut vorgesorgt werden sollte. Um dieses Risiko auf Anwendungsebene möglichst gering zu halten, soll der Benutzer in der Anwendung möglichst viel Kontrolle über das virtuelle Fortbewegungsmittel erhalten. Darüber hinaus werden virtuelle Objekte wie Ruder an die realen Gegenstände ausgerichtet.

## 2.2 Vergleichbare Arbeiten

In diesem Kapitel werden einige Arbeiten vorgestellt, die vergleichbare Ansätze bezüglich des Themas dieser Arbeit zeigen oder Ähnlichkeiten bei der Beschreibung eines Aufbaus für die Praxis aufweisen. Hierbei werden insbesondere die Punkte Motivation der Arbeit, Aufbau, Funktionsweise und Interaktion bezüglich der virtuellen Realität aufgezeigt. Im Anschluss findet eine Abgrenzung dieser Publikationen zu der in dieser Arbeit beschriebenen VR-Ruderanwendung statt.

Bei der ersten vergleichbaren Arbeit (siehe Abschnitt 2.2.1) handelt es sich um eine Studie, welche sich damit beschäftigt, ob virtuelle Umgebungen einen Einfluss auf die Motivation beim Sport haben. Die Studie ergab, dass der Einsatz von VR für ein besseres Gesamterlebnis beim Training sorgt, aber keinen Einfluss auf die erbrachte Leistung hat[15].

Die zweite Arbeit in Abschnitt 2.2.2 handelt von einem Projekt, in dessen Rahmen ein Produkt entwickelt wurde, das es ermöglicht, einige Sportgeräte mit einer virtuellen Trainingsumgebung auszustatten. In dieser kann der Benutzer sein Training in verschiedenen Spielmodi durchführen, was somit einen ähnlichen Ansatz wie diese Arbeit verfolgt. Zudem ergaben Befragungen, dass sich dieses Konzept positiv auf die Motivation der Nutzer auswirkt[13].

In Abschnitt 2.2.3 geht es um eine Arbeit, in der ein Virtual Reality Exergame auf der Basis eines Arcade-Klassikers entwickelt wurde, welches unter Verwendung eines Fahrrads und einer Kinect<sup>6</sup> ein eigenes Interaktionskonzept umsetzt. Durch den Ansatz, ein Exergame mit einer dafür angepassten Interaktionsform auszustatten, besteht hierbei eine Relevanz für diese Arbeit.

Abschnitt 2.2.4 handelt von einer Arbeit, in welcher der Aufbau einer virtuellen Trainingsumgebung für einen Fahrrad-Heimtrainer beschrieben wird und welche sich auf den Einsatz von Gamification Techniken fokussiert. Diese Arbeit liegt durch die Kombination von VR-Sport und Gamification in einem vergleichbaren Themengebiet wie diese Bachelorarbeit. Eine anschließende Studie ergab zudem, dass Gamification-Techniken, wie etwa Herausforderungen, einen positiven Effekt auf die Motivation der Nutzer haben können[21].

---

<sup>6</sup><https://developer.microsoft.com/en-us/windows/kinect> zuletzt aufgerufen am 30.05.2019

### 2.2.1 Virtuelle Umgebungen beim Rudern

Während sich viele Arbeiten mit Ansätzen und Umsetzungen von VR als Hilfsmittel im professionellen Sportumfeld beschäftigen, gibt es auf der anderen Seite noch die normalen Nutzer, die nur in einem normalen Rahmen oder sogar gar keinen Sport betreiben. Wie sich der Einsatz von VR in Kombination mit Sport auf diese Personengruppe auswirken kann, insbesondere im Hinblick auf Motivation und Engagement für den Sport, wurde in der Studie von Schmidt u.a.[15] untersucht. Hierfür wurde anhand mehrerer Trainingsumgebungen untersucht, wie Testpersonen das Training in den jeweiligen Umgebungen bewerten.

#### Versuchsaufbau

Im Fokus der Studie standen die Fragen, ob der Einsatz von virtuellen Umgebungen die Motivation und das Engagement der Nutzer erhöhen kann und welche virtuelle Umgebung für diesen Zweck besser geeignet ist. Zu diesem Zweck, wurden drei Trainingssetups miteinander verglichen, zwei VR-Setups, einmal mit einem HMD und einmal mit einer CAVE Umgebung, sowie ein Setup ohne VR-Umgebung. Ausschlaggebend für die Untersuchung waren, dass 1. die virtuelle Umgebung nicht allzu viele Spielelemente hat, um das Training ohne VR-System besser vergleichbar zu machen, 2. die Verwendung verschiedener Wiedergabemöglichkeiten zur Darstellung der virtuellen Umgebung und 3. die Verwendung eines Gerätes, welches Interaktion ermöglicht. Als Trainingsgerät kam ein Ruderergometer zum Einsatz und als Repräsentanten für die VR-Systeme, wurden die HTC Vive als HMD und eine CAVE Umgebung mit dem Namen „Data Arena“ gewählt.

Das verwendete Ruderergometer verfügt über Sensoren sowie eine API, die über http den Zugriff auf Messdaten wie Geschwindigkeit und Stoßkraft ermöglicht. Die virtuelle Umgebung wurde in einem Unity-Projekt<sup>7</sup> umgesetzt, bei dem ein von Bergen umgebener Sommersee als Szenerie dient, auf welchem sich der Benutzer in einem Ruderboot befindet. Durch eine lineare Interpolation zwischen den Ergometerdaten und der Positionen der Spielobjekte, werden die Bewegungen der virtuellen Ruder und des virtuellen Bootes dargestellt. Eine rote Linie, welche sich auf den Benutzer zubewegt, soll die ersten zurückgelegten 50 Meter aufzeigen. Probleme mit Cybersickness sollen durch die übereinstimmenden Ruderbewegungen zwischen Ruderergometer und virtuellem Ruderboot stark reduziert sein.

---

<sup>7</sup><https://unity3d.com> zuletzt aufgerufen am 30.05.2019



### **Ablauf**

Für den Ablauf der Studie, wurde pro Person drei mal je drei Minuten trainiert, einmal auf jedem Setup. Die Wahl des Setups erfolgte in zufälliger Reihenfolge. Nach jeder Übung gab es eine 5 minütige Pause und die Testpersonen wurden zu ihren Training befragt. Zum Schluss wurde noch ein Interview durchgeführt, bei dem nach positiven, sowie negativen Erfahrungen mit den einzelnen Setups gefragt wurde und die Testpersonen angeben sollten, welches Training sie persönlich bevorzugen und empfehlen würden. Insgesamt nahmen 11 Personen teil.

### **Abgrenzung**

Die Ähnlichkeiten der Studie zu dieser Arbeit sind in erster Linie, dass es sich bei beiden um eine Kombination von VR mit einem Sportgerät handelt, genauer gesagt arbeiten beide Arbeiten mit einem Rudergerät und einem HMD. Auch die Idee der Umsetzungen, sich nicht an eine medizinische oder professionelle Zielgruppe zu richten, sondern stattdessen Leute anzusprechen die eher gelegentlich oder sogar keinen Sport machen, weist Ähnlichkeiten auf. So liegt der Fokus von [15] sehr stark darauf herauszufinden, ob es möglich ist die Motivation zum Sport mithilfe von VR-Exergames zu erhöhen und welche Technik dabei den besten Eindruck hinterlässt. Dazu wird jedoch eine Testumgebung gewählt, welche möglichst wenig Gamification-Elemente beinhaltet, um die Setups besser miteinander vergleichen zu können. Auch in dieser Arbeit soll überprüft werden ob ein VR-Exergame einen Einfluss auf die Trainingsmotivation haben kann, der Fokus liegt aber viel eher darauf, eine Auswahl von Trainingsinhalten zu bieten und das Training durch Einbindung von Gamification-Elementen interessanter zu gestalten.

Während die Interviews ein Kernpunkt der Studie darstellen um herauszufinden welche Vor- und Nachteile VR-Training gegenüber gewöhnlichem Training hat und welche Faktoren für eine Gesamterfahrung entscheidend sind, liegt der Schwerpunkt dieser Arbeit viel eher darin, ein Konzept für ein umfangreiches Trainingserlebnis in Form eines VR-Exergames zu entwickeln und dies beurteilen zu lassen. Hierbei soll durch unterschiedliche Trainingsmodi und die Möglichkeit das Ruderboot vollständig zu steuern, die Trainingsmotivation durch Unterhaltung und Immersion gefördert werden.

### 2.2.2 Exergaming Simulator

Exergaming hat viel Potential als Gegenmaßnahme zu Bewegungsmangel zu dienen und durch spielerischer Eigenschaften mehr Menschen zu Bewegung zu motivieren, was einen positiven Effekt auf die Gesundheit haben kann. Mit dem Thema der Kombination von Trainingsgeräten mit Exergaming, hat sich ein Team aus Finnland auseinandergesetzt und einen Exergaming Simulator entwickelt und vorgestellt [13], der es ermöglicht mehrere Trainingsgeräte mit einer virtuellen Umgebung auszustatten. Das Projekt startete 2013 vor dem Hintergrund, dass Exergame-Trainingsgeräte bis dahin meist zu wenig Immersion boten, was durch zu kleine Bildschirme und einen Mangel an Interaktionsmöglichkeiten begründet ist. Entstanden sind mehrere Prototypen, welche in unterschiedlichen Pilotversuchen getestet wurden.

#### **Aufbau**

Das Exergaming wurde durch eine virtuelle Umgebung realisiert, welche sich mit Trainingsgeräten wie etwa einem Laufband oder einem Fahrrad kombinieren lässt und mit welcher der Nutzer durch Einsatz von Bewegungssteuerung interagieren kann. Im Mittelpunkt standen dabei die sportliche Anwendung und der spielerische Charakter. Umgesetzt wurde die Software mit der Unity 3D Spiel-Engine. Als virtuelle Umgebungen dienten ein Waldgebiet, eine Stadt, eine tropische Insel, sowie eine Gebirgsumgebung.

Der Nutzer hat die Möglichkeit die virtuelle Umgebung entweder frei zu erkunden oder festen Routen unterschiedlicher Längen und Schwierigkeitsgrade zu folgen. An Trainingsmodi stehen Joggen, Fahrrad fahren, Orientierungslauf und Abenteuer zur Auswahl. Beim Orientierungslauf beispielsweise, muss der Nutzer in einer begrenzten Zeit möglichst viele Checkpoints in der virtuellen Umgebung ablaufen. Zur Orientierung kann hierfür eine reale Karte der Gegend verwendet werden. Darüber hinaus gibt es die Möglichkeit gegen andere Nutzer anzutreten oder auch gegen einen Avatar, basierend auf den eigenen vorherigen Trainingsergebnissen, zu spielen.

Die Spielinhalte werden je nach Prototyp entweder über einen einzelnen Monitor oder Projektor, über drei Monitore oder über drei Projektoren im Aufbau einer CAVE Struktur wiedergegeben. Entsprechend variiert die Größe des Aufbaus von einem Aufbau für begrenzte Platzverhältnisse hin zu der Boxstruktur eines CAVE Systems. Ein Computer auf dem die Software läuft ist Teil der Prototypen und die Anbindung der Trainingsgeräte

an das System wird durch ein Modul ermöglicht, welches als Kommunikationsschnittstelle dient. Zusätzlich können über das Modul Erweiterungen angeschlossen werden, wie etwa ein Herzfrequenzmonitor oder eine Oculus Rift. Die Bewegungserkennung erfolgt mithilfe einer Kinect.

### **Zielgruppe**

Es wurden mehrere Pilotstudien durchgeführt um herauszufinden, wie gut der Simulator zu verschiedenen Zielgruppen passt, was die Stärken und Schwächen des Produktes sind und um die Geschäftsaussichten unterschiedlicher Zielmärkte zu analysieren. Durchgeführt wurden die Studien meist an Standorten mit genügend Platz für die größeren Aufbauten und mit einer Vielzahl an Testpersonen, wie etwa in einem Fitnesscenter oder bei einer Messe. Fitnesscenter stellen darüber hinaus die Hauptzielgruppe für die Entwicklung des Produktes dar.

Umfragen zu der Bewertung des Trainings ergaben dass die Testpersonen recht positiv auf das Training reagierten und ihnen Punkte wie der Spaßfaktor, die Präsentationsart und die Szenerie besonders gut gefielen. Lediglich der Anwendungsfall des Wendens und die Realitätsnähe der Bewegungsgeschwindigkeit wurden neutral bewertet. Die Umfragen haben zusätzlich noch festgestellt, dass es zwischen älteren (älter als 35) und jüngeren (35 oder jünger) Menschen keine wesentlichen Unterschiede beim Spaßfaktor gab.

### **Abgrenzung**

Gemeinsame Merkmale des Exergaming Simulator aus dem Artikel von [13] und dem Thema dieser Arbeit sind, dass beide Arbeiten das Ziel haben den Sport auf einem Sportgerät mit einer virtuellen Umgebung auszustatten. Dies ermöglicht es dem z.T. recht eintönigen Training eine neue Komponente hinzuzufügen, welche dem Nutzer unterschiedliche Welten präsentiert und ihn darüber hinaus noch mit ihnen interagieren lässt. Die Art der Interaktion ist dabei unterschiedlich, so wird bei dem Exergaming Simulator Bewegungssteuerung durch ein Kinect realisiert, beispielsweise um die Laufrichtung durch eine Neigung des Kopfes zu ändern. Bei dieser Arbeit hingegen soll die seitliche Neigung des Körpers mithilfe von Sensoren erkannt werden, um diese zur Lenkung in der virtuellen Umgebung zu nutzen. In beiden Fällen dienen diese Interaktionsarten dazu mehr Möglichkeiten zu haben um spielerische Konzepte umzusetzen, den Grad an Immersion zu steigern und das Erlebnis des VR Sports spannender zu gestalten. Des Weiteren sind

beide Konzepte darauf ausgelegt, mehrere unterschiedliche Spiel- bzw. Trainingsmodi anzubieten.

Unterschiede sind hingegen, dass der Exergaming Simulator universal für mehrere Arten von Trainingsgeräten einsetzbar sein soll, wohingegen sich diese Arbeit auf ein Gerät fokussiert und sich außerdem auf die Konzepte sowie die Umsetzung konzentriert. Auch die Zielgruppen unterscheiden sich, so soll sich diese Arbeit auf das Heimtraining fokussieren und mit dem Platzverbrauch eines Rudergerätes und einer HTC Vive auskommen. Der Exergaming Simulator hingegen ist in erster Linie auf den Einsatz in einem Fitnesscenter ausgelegt und benötigt je nach Ausführung viel Platz. Der Grund hierfür ist die Verwendung von Bildschirmen oder gar einer CAVE Struktur zur Wiedergabe der virtuellen Umgebung, im Gegensatz zu einem platzsparenden HMD. Außerdem soll in dieser Arbeit ein Steuerungskonzept entwickelt werden, welches universal für Rudergeräte einsetzbar ist, ohne dass diese selbst eine Schnittstelle bereitstellen müssen.

### 2.2.3 Arcade Klassiker als VR Umsetzung

Der Einsatz von virtueller Realität in Videospiele hat das Potenzial, ein immersiveres Erlebnis zu bieten als der Einsatz klassischer Wiedergabegeräte. Es besteht jedoch das Problem, dass viele Spiele nur für den Einsatz von HMDs nachgerüstet werden und die klassischen Eingabemethoden wie Tastatur, Maus und Controller weiterhin verwendet werden. Dies kann zu weniger Immersion und Symptomen von Cybersickness führen. Exergames hingegen nutzen meistens andere Eingabemöglichkeiten, jedoch mangelt es den Spielen oft an Immersion, was zu einer geringen Motivation führen kann. Auf Grundlage dieser Problematiken, haben Bolton u.a. ein Virtual Reality Exergame namens Paperdude entwickelt, welches alternative Interaktionsmöglichkeiten mit einer VR-Umsetzung eines Arcade-Klassikers vereint.[1]

#### Aufbau

Das Exergame ist dem Arcade-Spiel Paperboy<sup>8</sup> nachempfunden, in welchem man als Zeitungszusteller auf einem Fahrrad durch ein Vorstadtviertel fährt und Zeitungen in Briefkästen und auf Grundstücke wirft. Zum Einsatz kommen hierfür ein Fahrrad, dessen Hinterrad mit einem stationären Ersatz ausgetauscht wurde, eine Kinect zum Tracken des Körpers, und ein HMD als Wiedergabegerät. Der Hinterradersatz ermöglicht es, Werte

---

<sup>8</sup><http://www.atari.com> zuletzt aufgerufen am 30.05.2019

wie Geschwindigkeit und Stärke über Bluetooth an ein Mobilgerät zu senden, von dem aus die Informationen per UDP an das Spiel gesendet werden. Die Kinect wird verwendet, um die Position der Gelenke des Spielers zu ermitteln, sodass dessen Bewegungen erfasst und auf einen virtuellen Avatar abgebildet werden können. Dadurch können Nutzer virtuelle Arme und Hände sehen, welche sich ähnlich wie ihre realen Gegenstücke verhalten. Des Weiteren kommt die Kinect zur Gestenerkennung zum Einsatz.

### **Spielmechaniken**

In dem Spiel fährt der Spieler in der Ego-Perspektive eine Straße entlang, an welcher sich Häuser und Briefkästen befinden. Ziel ist es möglichst viele Zeitungen in die Briefkästen der Häuser zu werfen. Zum Werfen der Zeitung muss eine Wurfgeste vollführt werden, welche von der Kinect erfasst wird. Die Geschwindigkeit des virtuellen Fahrrads wird durch die Tretgeschwindigkeit des realen Fahrrads bestimmt, was zu einem natürlichen Fahrgefühl führen und das Risiko von Cybersickness reduzieren soll. Lenken ist nicht möglich, um aber dennoch für mehr Schwierigkeit zu sorgen, wurden Hindernisse und zerstörbare Objekte in das Spiel implementiert. Erstere erscheinen vor dem Fahrer auf der Straße und müssen durch den Treffer einer Zeitung zerstört werden. Darüber hinaus gibt es zerstörbare Objekte wie Fenster und Lampen, welche nicht zerstört werden sollten. Das Spiel ist mit einem Punktesystem ausgestattet, durch welches man Punkte für das Ausliefern der Zeitungen bekommt und Punkte verliert, wenn man ein Hindernis nicht beseitigt oder andere Objekte zerstört.

### **Abgrenzung**

Das Ziel des Spieles Paperdude von Bolton u.a.[1] ist es, ein immersives Exergame zu verkörpern, welches die Ansprüche hat, einen natürlichen Input zu bieten und zum Trainieren zu motivieren. Diese Motivation weist Ähnlichkeiten zu der Motivation dieser Arbeit auf, welche ebenfalls darauf abzielt, sportliche Aktivitäten durch die Nutzung spielerischer Aspekte in einer virtuellen Umgebung zu fördern und dabei eine natürliche Form der Interaktion mit der Anwendung zulässt. Auch die Umsetzung weist Gemeinsamkeiten auf, so kommen in beiden Arbeiten Trainingsgeräte zum Einsatz, welche durch ein HMD mit einer virtuellen Umgebung ausgestattet sind und darüber hinaus noch eine weitere Quelle für die Interaktion miteinbeziehen. Hierbei unterscheidet sich die verwendete Technik jedoch sehr stark.

Für Paperdude wird ein stationäres Fahrrad verwendet und es kommt eine Kinect zum Einsatz, mit welcher durch Gestensteuerung mit dem Spiel interagiert werden kann und die es ermöglicht, die Bewegungen des Nutzers auf einen virtuellen Avatar abzubilden. In dieser Arbeit wird dagegen ein Rudergerät eingesetzt, bei dem die Hände des Benutzers für das Sportgerät verwendet werden müssen. Um dennoch eine Interaktion zu ermöglichen, sollen statt einer Kinect Sensoren verwendet werden, die die Gewichtsverlagerung des Benutzers messen und dadurch die Möglichkeit zu lenken integrieren. Auch die Anwendungen unterscheiden sich stark, so handelt es sich bei Paperdude um ein Spiel, welches auf einem Arcade-Klassiker Paperboy basiert. Die Ruder-Anwendung hingegen wird eine Umsetzung mehrerer Spielmodi, die jeweils einen unterschiedlichen Trainingsfokus haben.

### 2.2.4 Gamification im Heimtraining

VR-Geräte stellen heutzutage eine bezahlbare Form der Unterhaltung dar, sodass sich diese auch zur Verwendung im Heimtraining eignen. Durch den Einsatz von Gamification bei einem solchen Training, besteht die Möglichkeit, bei dem Benutzer zu einem höheren Maß an Motivation zu führen. Mit dem Rift-a-bike genannten Projekt haben Tuveri u.a. unter Verwendung von Gamification Techniken eine virtuelle Trainingsumgebung für den Einsatz mit Fahrrad-Heimtrainern erstellt, mit dem Anspruch, nur Geräte auf Verbraucherebene zu verwenden.[21]

#### **Aufbau**

Der Aufbau von Rift-a-bike ähnelt dem Aufbau der vorangegangenen Arbeit, anstelle eines Fahrrads, kommt jedoch ein Fahrrad-Heimtrainer zum Einsatz, welcher mit einem Raspberry Pi zum Erfassen der Geschwindigkeit ausgestattet ist. Ansonsten kommen noch ein HMD als visuelle Schnittstelle zur virtuellen Umgebung, eine Kinect, sowie Kopfhörer zum Einsatz. Als virtuelle Umgebung dient eine Stadt, welche prozedural generiert und in Unity 3D importiert wurde. Durch diese Stadt führt ein vordefinierter Pfad, auf welchem der Nutzer sich auf einem virtuellen Fahrrad fortbewegt. Der Nutzer wird in der virtuellen Umgebung durch einen Avatar repräsentiert, der die Bewegungen des realen Benutzers mithilfe des Trackings der Kinect repliziert.

#### **Spielmechaniken**

Das Ziel des Spiels ist es, sich durch verschiedene Phasen bzw. Levels des Spiels zu

bewegen und sich dabei nach einer Idealgeschwindigkeit zu richten, welche durch die unterschiedlichen Trainingsphasen vorgegeben wird. Das Training gliedert sich in eine Aufwärm-, Trainings- und Abkühlphase. Ziel dieser Trainingsphasen ist es, den Körper auf Anstrengung vorzubereiten, bzw. ihn wieder zur Ruhe kommen zu lassen. Um stärker zum Training zu Motivieren, wurden mehrere Gamification Elemente eingebaut. Zum einen gibt es Punkte, welche eine virtuelle Belohnung darstellen und als quantitative Maßeinheit für die Leistung des Benutzers dienen. Zu diesem Zweck werden Münzen generiert, deren Häufigkeit des Auftretens davon abhängt, wie korrekt der Fahrer die Geschwindigkeit hält.

Eine weiteres Gamification Element sind Herausforderungen. In Rift-a-bike zieht sich dieser Punkt durch das gesamte Training. Hierfür wurde ein Hase implementiert, welchem der Nutzer folgen und so lange wie möglich mit ihm Schritt halten soll. Der Hase stellt dadurch etwas gegenwärtiges dar, was einem durch das Training hindurch begleitet und gleichzeitig ein Feedback über die eigene Leistung liefert. Um gute Leistungen im Spiel zu belohnen wurden Preise und Trophäen implementiert. Einen Preis erhält man für den Abschluss jeder Phase und der Wert richtet sich nach der erbrachten Leistung. Trophäen gibt es zusätzlich noch für den Abschluss eines Trainings. Der Wert der Trophäe setzt sich dabei aus einem Gesamtwert der Preise zusammen. Dieses Belohnungssystem soll für mehr Motivation sorgen.

### **Abgrenzung**

Tuveri u.a. nutzten für ihr Projekt Rift-a-bike[21] verschiedene Gamification Mechaniken um herauszufinden welchen Einfluss diese auf das Training haben. Die verwendeten Mechaniken waren u.a. die Nutzung eines Punktesystems, Herausforderungen, Levels oder auch Preise. Auch diese Arbeit zielt darauf ab, durch den Einsatz von Gamification ein motivierendes Exergame zu erstellen. Einige dieser Mechaniken sollen ebenfalls Verwendung finden, jedoch in anderer Ausführung. Beispielsweise werden Herausforderungen daraus bestehen, Objekten auszuweichen oder Zeiten zu schlagen und Levels wird es in Form von unterschiedlichen Spielmodi geben, welche unterschiedliche Ziele verfolgen.

Beide Arbeiten befassen sich damit, eine Anwendung zu schaffen, welche das gewöhnliche Training um eine virtuelle Ebene erweitert. Neben den unterschiedlichen verwendeten Sportgeräten, gibt es auch Unterschiede in der Bedienung der Anwendungen. Bei Rift-a-bike wird die Geschwindigkeit des Sportgerätes gemessen, darüber hinaus wurden

aber keine weiteren Interaktionsmöglichkeiten beschrieben, welche sich auf die Anwendung auswirken. Jedoch kommt eine Kinect zum Einsatz, welche den realen Körper auf den virtuellen Avatar abbildet. Bei dieser Arbeit hingegen soll neben dem Steuern der Geschwindigkeit auch ein Lenken des virtuellen Bootes möglich sein, was in einigen Trainingsmodi von zentraler Bedeutung ist.

### 2.3 Zusammenfassung

In diesem Kapitel wurden zunächst die Begriffe *Exergames*, *Gamification*, *Virtual Reality*, *Immersion* und *Cybersickness* genauer erläutert. Anschließend wurde darlegt warum diese Begriffe im Kontext dieser Arbeit relevant sind. Dies ist im wesentlichen der Fall, da die VR-Ruderanwendung, welche in den folgenden Kapiteln noch genauer aufgeführt wird, den Kriterien eines Exergames entspricht und durch Gamification ein ansprechendes Trainingserlebnis bieten soll. Bedingt durch das Medium VR, spielen auch Immersion und Cybersickness eine Rolle.

In 2.2 wurden in den Unterkapiteln 2.2.1 bis 2.2.4 einige Arbeiten mit einem Sport-VR-Fokus vorgestellt. Dabei wurde auf den Aufbau, sowie die Funktionsweise der in den Arbeiten vorgestellten Projekte eingegangen. Im Anschluss fand eine Abgrenzung zu dieser Arbeit statt. Die Vorstellung dieser Arbeiten soll dazu dienen einen besseren Überblick über das Thema der VR-Exergames mit Trainingsgeräten zu erhalten. Die Abgrenzungen dienen dazu, sich ein besseres Bild von der in dieser Arbeit zu entwickelnden Anwendung zu machen und wie sich diese von anderen Arbeiten unterscheidet bzw. wo Gemeinsamkeiten liegen. Die wesentlichsten Unterschiede liegen darin, dass sich diese Arbeit ausschließlich auf ein Rudergerät fokussiert und Bewegungstracking über Sensoren im Sitz, statt über Body Tracking durch eine Kinect, erfolgt. Weiterhin besteht der Schwerpunkt eine trainingsgerechte Anwendung zu entwickeln, die durch Gamifikation ansprechender gestaltet werden soll, um die Motivation für das Training zu erhöhen sowie einer Überprüfung, ob dieser Zweck erfüllt wird.



## 3 Funktionale und nicht funktionale Anforderungen

In dieser Arbeit soll eine Anwendung entstehen, welche die Möglichkeit bietet, ein physisches Rudergerät in eine VR-Umgebung einzubinden, um eine Ruderumgebung zu simulieren, mit welcher der Benutzer interagieren kann. Das Ziel ist es, Heimsport und Technik zu kombinieren, um mehr Unterhaltung zu bieten und damit die Motivation für das Training zu erhöhen. Im Folgenden werden in Abschnitt 3.1 die funktionalen Anforderungen aufgeführt, welche die einzelnen Komponenten der Anwendung beschreiben und in Abschnitt 3.2 werden die nicht funktionalen Anforderungen erfasst.

### 3.1 Funktionale Anforderungen

Die Anwendung soll drei Level besitzen, die jeweils einen unterschiedlichen Trainingsmodus darstellen und sich auf verschiedene Aspekte des Anreizes zum Training konzentrieren. Die Funktionen und Gestaltungen der Level werden in den Abschnitten 3.1.1, 3.1.2 und 3.1.3 beschrieben. Einen zentralen Bestandteil der Anwendung soll ein Ruderboot darstellen, auf welchem der Benutzer die virtuelle Umgebung befahren kann. Dafür soll es möglich sein das Boot zu bedienen, indem mit dem Rudergerät interagiert wird. Die Anforderungen an die Interaktionsmöglichkeiten und an die Simulation des Ruderbootes sind in den Abschnitten 3.1.4 und 3.1.5 zu finden.

#### 3.1.1 Sprintlevel

Bei dem ersten Level soll es sich um einen Sprintmodus handeln, in welchem sich der Nutzer auf ein reines Training und seine erbrachte Leistung konzentrieren kann. Hierfür soll er in möglichst kurzer Zeit eine feste Distanz zurücklegen. Um dies zu erfüllen, muss der Nutzer eine gute Taktung seiner Bewegungen finden, um eine möglichst gute

Geschwindigkeit zu erreichen und er soll sich vollständig auf das Training konzentrieren können.

#### **Gestaltung**

Zu diesem Zweck soll die Spielwelt einen rein optischen Zweck erfüllen und keine ablenkenden Elemente, wie etwa Kollisionsobjekte, besitzen. Ein See soll als Kulisse dienen, da dieser einen großen Abstand zwischen dem Boot des Nutzers und dem Ufer erlaubt und so eine unbeabsichtigte Kollision mit dem Gelände verhindert wird. Der Benutzer soll allgemein frei in seiner Bewegung sein, es soll also auch möglich sein das Boot zu lenken. Um ihm jedoch die Orientierung zu erleichtern, soll es eine Hilfslinie geben, welche die Zielrichtung weist, sowie weitere Markierungen um feste Distanzen abzustecken.

#### **Levelinhalte**

Um dem Nutzer Rückmeldung über seinen aktuellen Fortschritt zu geben und ihm die Möglichkeit zu lassen, sein Training entsprechend anzupassen, sollen Daten wie die zurückgelegte Strecke, die verstrichene Zeit und die aktuelle Geschwindigkeit für den Nutzer über Anzeigen in der virtuellen Umgebung sichtbar sein. Um dem Level eine Herausforderung zu geben und damit die Motivation für das Training zu erhöhen, soll es eine Bestenliste geben, welche die Zeiten für die jeweilige Entfernung verzeichnet und nach dem Training die Bestzeiten sowie die benötigte Zeit des aktuellen Trainings anzeigt.

#### **3.1.2 Parcourslevel**

Das Parcourslevel soll einen wesentlich stärkeren Fokus auf Geschicklichkeit legen, um das Training durch Spaß und Herausforderung auf Dauer interessanter zu halten. In dem Level soll der Nutzer durch einen Fluss rudern und einen Zielpunkt in einer möglichst kurzen Zeit erreichen. Dieser Fluss soll dabei aufgrund seiner Bauweise und der Hindernisse, mit denen er ausgestattet ist, eine Herausforderung darstellen ihn zu durchqueren.

#### **Gestaltung**

Das Terrain spielt bei diesem Level eine sehr wichtige Rolle, da es sowohl richtungsweisend ist, als auch ein Hindernis darstellt und damit den Schwierigkeitsgrad wesentlich bestimmt. Zu diesem Zweck soll ein Flusslauf gewählt werden, welcher nicht nur gerade

Strecken beinhaltet, sondern besonders kurvenreich ist, evtl. auch einen mäandrierenden Lauf besitzt. Des Weiteren sollen Gabelungen verbaut werden, wodurch mehr Möglichkeiten zur Befahrung der Strecke geboten werden. Bei der Wiederausführung mehrerer Flussläufe ist darauf zu achten, dass der Verlauf der Fließrichtung gut sichtbar ist, um zu verhindern, dass der Spieler in die falsche Richtung rudert.

#### **Levelinhalte**

Als Parcours-Elemente sollen neben dem Terrain auch größere Felsen und Stromschnellen in dem Level verwendet werden. Die Felsen sollen an unterschiedlichen Stellen entlang des Flusslaufes platziert werden, um mehr Elemente zu schaffen, welchen der Spieler ausweichen muss. Stromschnellen oder auch seichtere Strömungen sollen entlang des Flusslaufes platziert werden und das Boot des Spielers in die Fließrichtung der Stromschnelle beschleunigen. Sie dienen als Elemente, welche der Spieler einerseits zu seinem Vorteil nutzen kann, um sich schneller fortzubewegen und Hindernissen auszuweichen, welche aber auch das Gegenteil bewirken können und ihn von seinem Weg abbringen können. Beide Elemente dienen dazu, sowohl die Herausforderung als auch den Schwierigkeitsgrad des Levels zu erhöhen. Nach Erreichen des Ziels soll auch hier eine Übersicht über die Bestzeiten, sowie die gefahrene Zeit angezeigt werden. Dies soll den Wiederspielwert erhöhen und den Spieler motivieren, sich in dem Spielmodus zu verbessern.

#### **Risiken**

Dieses Level besitzt aufgrund der erhöhten Kollisionsgefahr ein höheres Risiko für das Auftreten von Cybersickness, wenn auf den Körper nicht die Einflüsse wirken, die er bei einer Kollision und den einhergehenden visuellen Eindrücken erwarten würde. Um dieses Risiko zu minimieren, soll es keine Passagen geben in denen der Spieler einer Kollision nicht mehr ausweichen kann, etwa wenn er in einer Stromschnelle gefangen wäre. Optional sollen auch wesentliche Auswirkungen, welche bei einer Kollision auf das Boot wirken würden, deaktivierbar sein. Dies soll etwa das Schaukeln und Kippen des Bootes unterbinden.

Ein weiteres Risiko stellt eine schlechte Abwägung des Schwierigkeitsgrades dar, denn ein zu einfacher Schwierigkeitsgrad kann schnell dazu führen, dass das Level als langweilig empfunden wird und die Motivation für das VR-Training verloren geht, wohingegen ein zu hoher Schwierigkeitsgrad Anfänger abschrecken könnte. Entsprechend soll beim Erstellen

des Levels auf eine ausgeglichene Gewichtung des Schwierigkeitsgrades geachtet werden und es sollen genügend Möglichkeiten gegeben werden sich zu verbessern.

#### 3.1.3 Piratenlevel

Das Piratenlevel soll einen Modus darstellen, in dem der Nutzer nach seinem eigenen Tempo trainieren kann, ohne aufgrund eines Timers Zeitdruck zu haben. In dem Level rudert der Spieler durch eine karibische Landschaft, hierbei kommen ihm immer wieder Piratenschiffe über den Weg, welche mit Kanonen auf das Boot des Spielers schießen. Ziel ist es, den Kanonenkugeln auszuweichen und dafür zu sorgen, dass die Piratenschiffe kentern, indem man sie durch ihr Fahrverhalten sich gegenseitig rammen lässt oder dafür sorgt dass sie auf Grund laufen.

#### Gestaltung

Die Welt muss für diesen Zweck entsprechend gestaltet werden, sodass sie genügend Inseln bietet, an welche der Spieler die feindlichen Schiffe heran lotsen kann. Mit typischer Vegetation wie etwa Palmen oder Karibik-Designelementen kann das Setting darüber hinaus glaubwürdiger gestaltet werden.

#### Levelinhalte

Die Piratenschiffe sollen, wie auch das Boot des Nutzers, schwimmende Objekte sein und eine entsprechende Physik im Bezug auf die Auftriebskraft und den Wasserwiderstand bekommen. Sobald der Spieler sich dem Schiff auf eine bestimmte Distanz nähert, soll dieses die Verfolgung aufnehmen und um das Ruderboot kreisen. Dabei sollen aus den Kanonen Kugeln auf den Spieler geschossen werden. Wird der Spieler von einer dieser Kugeln getroffen, verliert er ein Leben, sollte er alle Leben verlieren, mit einem der Piratenschiffe kollidieren oder auf Land auflaufen, verliert er und das Spiel endet. Gewonnen werden kann, indem alle Piratenschiffe ausgeschaltet werden.

#### Risiken

Auch in diesem Level könnte bei einer Kollision Cybersickness auftreten. Aus diesem Grund soll bei einer Kollision mit einem Piratenschiff das Spiel beendet und Steuerung, sowie das Piratenschiff deaktiviert werden um Folgen und Auswirkungen gering zu halten.

Das Risiko zur Kollision mit dem Terrain soll durch eine offene Bauweise der Welt gering gehalten werden.

#### 3.1.4 Interaktion

Die Steuerung der Anwendung soll aus drei wesentlichen Elementen bestehen, dem Antrieb, der Lenkung und der Levelauswahl. Für alle drei Elemente soll der Controller als Basis dienen, lediglich die Lenkung soll als zusätzliche Option die Möglichkeit bekommen, über Sensoren angesteuert zu werden. Der Controller wird an der Griffstange des Rudergerätes befestigt, sodass die wichtigsten Daten über die Bewegung des Nutzers hierüber abgerufen werden können.

##### **Antrieb**

Die erste Steuerungsfunktion ist der Antrieb. Hierbei soll regelmäßig die Position des Controllers abgefragt werden, um festzustellen, ob eine Ruderbewegung durchgeführt wird. Ist dies der Fall, soll eine Kraft in Form einer Beschleunigung auf das Boot wirken. Die Dauer der Ruderbewegung soll hierbei einen Einfluss haben, sodass eine zu langsame oder zu schnelle Bewegung die Beschleunigung reduziert.

##### **Lenkung**

Die Lenkung soll über zwei Wege möglich sein, zum einen über Neigung des Controllers, zum anderen durch Einbindung von Drucksensoren, welche die Körperneigung des Nutzers messen sollen. Die Steuerung über den Controller soll messen, wie weit der Controller geneigt ist und abhängig davon eine entsprechend starke Rotation auf das Boot des Spieler durchführen. Die Sensoren hingegen sollen auf der Sitzfläche des Rudergerätes fixiert werden und über einen Mikrocontroller ihre Messwerte an die Anwendung weiter leiten. Neigt sich der Nutzer nun in eine Richtung, wird dies in der Veränderung der Werte ersichtlich und es wird eine Rotation durchgeführt, die proportional zur Neigung ist.

##### **Levelauswahl**

Neben der Steuerung des Ruderbootes, soll der Nutzer auch die Möglichkeit haben während des Trainings zwischen den Trainingsinhalten zu wechseln. Hierfür sollen Funktionen

eingebaut werden, die das Wechseln und Neustarten der Leveln erlauben. Darüber hinaus soll es möglich sein, die Art des Ruderbootes während des Trainings zu ändern, also zwischen einem gewöhnlichen und einem invertierten Ruderboot zu wechseln.

#### 3.1.5 Simulation eines Ruderbootes

Das Ruderboot dient dem Nutzer als Transportmittel, um sich durch die virtuelle Umgebung zu bewegen. Eine Steuerung findet dabei über die bereits genannten Mittel statt. Darüber hinaus soll das Boot über ein Fahrverhalten verfügen, welches die wichtigsten physikalischen Eigenschaften eines schwimmenden Objektes umsetzt. Außerdem soll es zwei Ausführungen geben, ein Ruderboot mit Blick nach vorne und eins mit Blick nach hinten. Abschließend soll die Bewegung der Ruder simuliert werden.

##### **Fahrverhalten**

Das Fahrverhalten soll durch drei wesentliche Größen beeinflusst werden. Die erste ist der Widerstand bzw. der Wasserwiderstand, dieser soll als Gegenkraft zum Antrieb die Geschwindigkeit des Bootes allmählich verringern, wie auch in der Realität das Wasser eine Bremswirkung auf Wasserfahrzeuge hat. Die zweite Größe ist der Auftrieb. Dieser ist dafür verantwortlich, dass das Boot nicht versinkt, bzw. durch das Wasser fällt. Hierfür soll je nach Tiefe des Bootes und dessen Verdrängung von Wasser eine Aufwärtskraft auf das Fahrzeug wirken, sodass sich durch die Gegengewichte von Auftrieb und Schwerkraft ein Mittelmaß findet, auf welchem sich das Schiff einpendelt. Der dritte Faktor des Fahrverhaltens ist die Ausrichtung des Bootes nach dessen Schwerpunkt. Dadurch soll sichergestellt werden, dass das Fahrzeug bei einer Schiefelage, etwa nach einer Kollision, wieder in eine waagerechte Ausrichtung zurückkehrt.

##### **Ausführungen**

Das Ruderboot soll es einmal in einer klassischen rückwärtsgewandten Variante, sowie in Form eines Vorwärtsruderbootes geben. Da bei dem klassischen Ruderboot der Blick nach hinten gerichtet ist, könnte es bei einigen Leveln Probleme geben die Fahrtrichtung nicht dauerhaft im Blick zu haben. Das Vorwärtsruderboot soll sich lediglich in der Art der Ruder und in der Blickrichtung von der rückwärtsgewandten Variante unterscheiden.

#### **Ruder**

Die Bewegung der Ruder soll basierend auf der Bewegung auf dem Rudergerät simuliert werden. Als Grundlage soll die Position des Controllers dienen, welcher an der Griffstange des Rudergerätes befestigt ist. Die Griffenden der virtuellen Ruder sollen sich nach dieser Position ausrichten. Da die Bewegung der Griffstange auf einem Rudergerät in der Regel linear ist, die Ruder bei einem realen Ruderboot aber ins Wasser gesenkt werden, soll dieses Auf- und Absenken der Ruder entsprechend nachmodelliert werden.

Die Ruder für das Vorwärtsruderboot sollen aus zwei Teilstücken bestehen, welche sich an einem Angelpunkt treffen und sich in entgegengesetzte Richtung bewegen.

## **3.2 Nicht funktionale Anforderungen**

Im Folgenden werden einige nicht funktionale Anforderungen aufgeführt, welche für eine stärkere Akzeptanz und eine gute Benutzbarkeit der Anwendung sorgen sollen.

#### **Bedienbarkeit**

Um eine gute Bedienbarkeit zu erreichen, sollte die Steuerung der Anwendung möglichst unkompliziert und intuitiv sein. Zu diesem Zweck sollte es vermieden werden, dass der Benutzer für irgendwelche Aktionen seine Position auf dem Rudergerät aufgeben und sein Headset absetzen muss. Die Steuerung, wie der Wechsel des Levels, soll mit lediglich dem Controller auskommen, da dieser erreichbar ist ohne das Training zu unterbrechen. Auch soll die Anwendung ohne ein Menü auskommen und nur mit direkten Aktionen über den Controller steuerbar sein.

Eine intuitive Kontrolle des Ruderbootes soll dadurch gegeben werden, dass der Nutzer wie gewohnt auf dem Rudergerät trainieren kann ohne sich umgewöhnen zu müssen. Außerdem soll er die freie Wahl haben, auf welche der verfügbaren Steuerungsarten er zurückgreift, um die für sich angenehmste zu wählen.

#### **Performance**

Die Anwendung sollte eine möglichst geringe Latenz zwischen einer Bewegung und dem daraus resultierenden Bild auf dem VR-Headset gewährleisten. Bei einer für den Nutzer spürbaren Latenz ist die Wahrscheinlichkeit für das Auftreten von Cybersickness zu hoch und die Anwendung ist entsprechend nicht nutzbar. Ein Teil dieser Latenz ist bereits

durch das Tracking des VR-System gegeben, auf den die Anwendung keinen Einfluss hat. Dafür sollte die anwendungsbedingte Latenz auf Grund von Rendering und Ausführung von Skripten möglichst gering gehalten werden.

#### **Konfigurierbarkeit**

Der Benutzer soll die Möglichkeit haben einige Einstellungen an der Anwendung vorzunehmen, um sie besser auf das eigene Trainingsgerät und das eigene Trainingstempo anzupassen.

### **3.3 Zusammenfassung**

Im wesentlichen sollen in der Anwendung drei unterschiedliche Level entwickelt werden, welche jeweils einen anderen Trainingsmodus darstellen. Die Interaktion mit der Umgebung erfolgt dabei über ein Ruderboot, welches vom Benutzer gesteuert wird. Diese funktionalen Anforderungen wurden in 3.1 aufgeführt. In 3.2 wurden die nicht funktionalen Anforderungen zur Bedienbarkeit, Performance und Konfigurierbarkeit der Anwendung beschrieben.



## 4 Entwurf

In diesem Kapitel wird der Entwurf der VR-Ruderanwendung beschrieben. Eine Systemübersicht in Abschnitt 4.1 gibt einen Überblick über die Verteilung der Komponenten auf die verwendete Hardware. Abschnitt 4.2 stellt die Software-Komponenten und deren Schnittstellen vor. Da die Anwendung mit Unity als Entwicklungsplattform entwickelt wurde, ist in Abschnitt 4.3 eine Abbildung der Komponenten auf das Programmiermodell von Unity beschrieben. Abschnitt 4.4 beschreibt ausführlich, wie die einzelnen Komponenten des Projektes umgesetzt wurden.

### 4.1 Systemübersicht

Das Verteilungsdiagramm in Abbildung 4.1 zeigt die Verteilung der Hard- und Softwarekomponenten, sowie die Kommunikationswege zwischen den Hardwarekomponenten. Die verwendeten Bestandteile lassen sich in drei Einheiten unterteilen. Die erste ist der *Computer* des Benutzers, der das Zentrum der Hardwareverteilung darstellt. Damit verbunden ist eine *Sensoreinheit*, welche in das Rudergerät integriert wird, und ein *VR-System*.

Auf dem *Computer* befindet sich die Unity Laufzeitumgebung. Diese dient zur Ausführung der Ruderanwendung, welche gemäß den Anforderungen im Kapitel 3 entwickelt wurde und in den folgenden Abschnitten genauer beschrieben wird. Zusätzlich wird das SteamVR Plugin<sup>1</sup> für Unity verwendet, welches die Interaktion mit dem *VR-System* ermöglicht. SteamVR kommuniziert dazu mit der OpenVR API und ermöglicht die Verwendung verschiedener VR-Systeme über dieselbe Schnittstelle [22].

Bei dem verwendeten *VR-System* handelt es sich um die HTC Vive. Das System setzt sich im Wesentlichen aus einem Headset, Controllern und Basisstationen zusammen. Das Headset dient als visuelle Schnittstelle zur Anwendung und ist über eine Anschlussbox

---

<sup>1</sup>[https://github.com/ValveSoftware/steamvr\\_unity\\_plugin](https://github.com/ValveSoftware/steamvr_unity_plugin) zuletzt aufgerufen am 25.07.2019

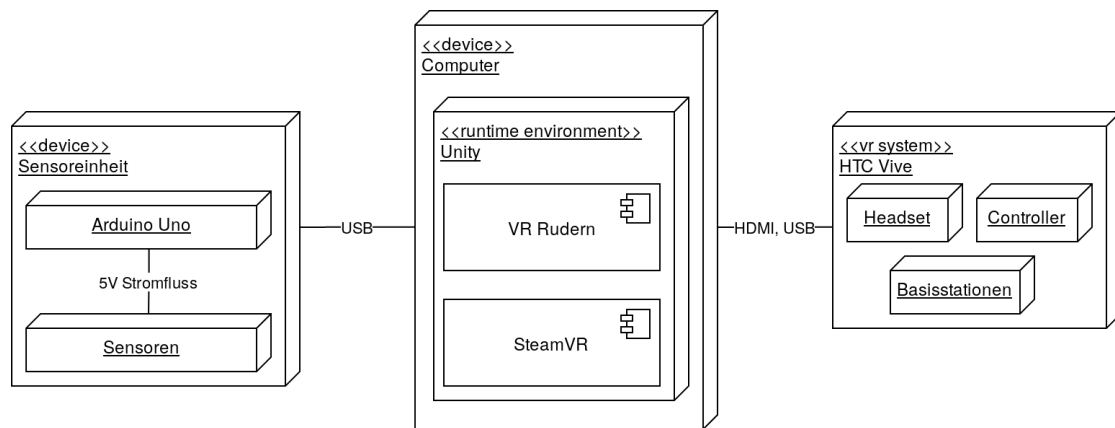


Abb. 4.1: Verteilungsdiagramm zur Übersicht über die eingesetzte Hardware und der Verteilung der Software (Quelle: eigene Arbeit)

über HDMI und USB mit dem *Computer* verbunden. Die Controller dienen zum Interagieren mit der virtuellen Welt und die Basisstationen ermöglichen die räumliche Erfassung von Headset und Controller. Die Kommunikation zwischen den Geräten funktioniert drahtlos.

Zuletzt kommt eine *Sensoreinheit* zum Einsatz, bestehend aus einem Arduino Uno<sup>2</sup> und vier Drucksensoren<sup>3</sup>. Durch die Sensoren fließt Strom, dessen Spannung durch die Belastung der Sensoren bestimmt wird. Der Strom wird von einem Arduino Uno bereitgestellt und über dessen Analogschnittstellen gemessen. Die gemessenen und verarbeiteten Ergebnisse werden daraufhin über einen seriellen Port per USB an den *Computer* übertragen.

## 4.2 Komponentenübersicht

Im Folgenden werden die einzelnen Komponenten der Anwendung aus dem Komponentendiagramm 4.2 in ihrer Funktionsweise und ihren Abhängigkeiten genauer erläutert. Im Gegensatz zum Verteilungsdiagramm 4.1 stehen hier die Softwarekomponenten, insbesondere die eigentliche Ruderanwendung selbst, sowie die Steuerung durch den Benutzer im Vordergrund. Die drei wesentlichen Komponenten sind die *Sensoreinheit*, das *VR-System*

<sup>2</sup><https://www.arduino.cc/> zuletzt aufgerufen am 25.07.2019

<sup>3</sup><https://www.interlinkelectronics.com/fsr-406> zuletzt aufgerufen am 25.07.2019

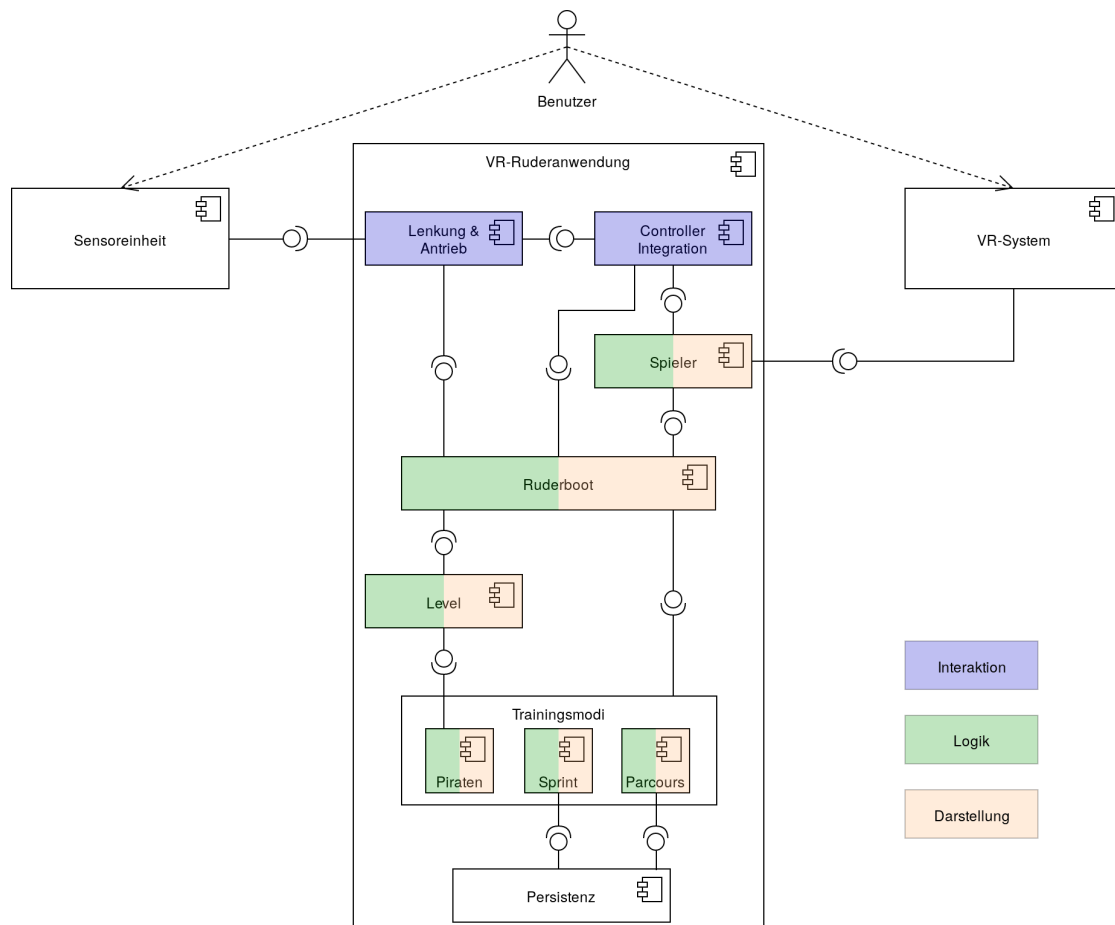


Abb. 4.2: Komponentendiagramm zur Übersicht des Zusammenspiels der verwendeten Komponenten (Quelle: eigene Arbeit)

und die *VR-Ruderanwendung*, welche sich selbst nochmal in weitere Komponenten unterteilen lässt.

Das *VR-System* stellt durch SteamVR eine Schnittstelle bereit, welche Zugriff auf das Headset und die Controller innerhalb der Anwendung erlauben. Diese werden durch virtuelle Gegenstände repräsentiert, wobei das virtuelle Headset eine Kamera beinhaltet, die einen Blick innerhalb der Spielwelt ermöglicht. Innerhalb eines Trackingbereiches des VR-Systems, werden die Positionen und Ausrichtungen der Hardwarekomponenten getrackt, sodass die virtuellen Controller und das Headset dementsprechend im virtuellen Raum bewegt und gedreht werden. Zusätzlich stellt SteamVR eine virtuelle Repräsentation dieses Trackingbereiches bereit, damit der Nutzer seine Grenzen auch innerhalb der Anwendung sieht. Diese virtuellen Repräsentationen finden sich in der Anwendung unter

der *Spieler-Komponente* wieder, die somit die einzige Komponente ist, die eine direkte Abhängigkeit zum *VR-System* hat. Zugriffe anderer Komponenten auf Informationen des *VR-Systems* finden somit über den *Spieler* statt. Der *Spieler* selbst hat noch eine Abhängigkeit zum *Ruderboot*, da es sich nach dessen Position und Rotation zu richten hat um sich aus Sicht des Benutzers auf dem Ruderboot zu befinden.

Die Steuerung der Anwendung findet über zwei Wege statt, zum einen über den Controller des *VR-Systems* und zum anderen durch die Benutzung der *Sensoreinheit*. Dafür werden die Drucksensoren, auf welche der Nutzer einwirkt, ausgelesen. Diese Informationen werden verarbeitet und die resultierenden Ergebnisse werden an die Anwendung weitergeleitet. Die Übertragung der Daten verläuft dabei asynchron und die *Sensoreinheit* sendet unabhängig vom Status der *Ruderanwendung*.

Dadurch, dass die Entwicklung der *Ruderanwendung* mit einer Spiel-Engine stattfindet, sind Komponenten nicht immer klar nach Logik und Darstellung getrennt, sondern besitzen häufig beides. Eine farbliche Einordnung soll deshalb aufzeigen, welche Komponenten neben der Logik noch eine visuelle Darstellung in der Anwendung finden. Zusätzlich sind die Komponenten, welche für die Interaktionsverarbeitung verantwortlich sind, separat gekennzeichnet.

Die *Controller Integration* ist zuständig für die Auswertung des Controller-Inputs des Benutzers, die Verwaltung welcher Controller in Benutzung ist und die Bereitstellung von Informationen für andere Komponenten. Die Auswertung beinhaltet eine Beurteilung, ob gerade eine Ruderaktion durchgeführt wird, wie weit diese fortgeschritten ist und die Verwaltung der Tastenbelegung, sodass beim Drücken einer zugeordneten Taste eine verknüpfte Aktion ausgeführt wird. Da immer nur ein Controller für das Training verwendet werden kann, ein VR-System aber in der Regel mindestens zwei Controller hat, verwaltet die Komponente, welcher Controller gerade aktiv ist. Die nach außen bereitgestellten Informationen umfassen die Position und Rotation des aktiven Controllers und die Information ob gerade eine Ruderaktion stattfindet und wie weit diese fortgeschritten ist. Die Komponente benötigt Zugriff auf die virtuellen Controller, sodass eine Abhängigkeit zur *Spieler-Komponente* besteht.

Das *Ruderboot* ist ein zentraler Bestandteil der Anwendung, da es durch die Aktionen des Benutzers gesteuert werden kann und als Mittel zur Erkundung der virtuellen Welt dient. Die Darstellungselemente der Komponente bestehen aus einem hölzernen Ruderboot und zwei Rudern. Die logischen Elemente machen eine Pseudophysik und die Simulation der Bewegung der Ruder aus. Die Pseudophysik ist für die Verarbeitung von Kollisionen,

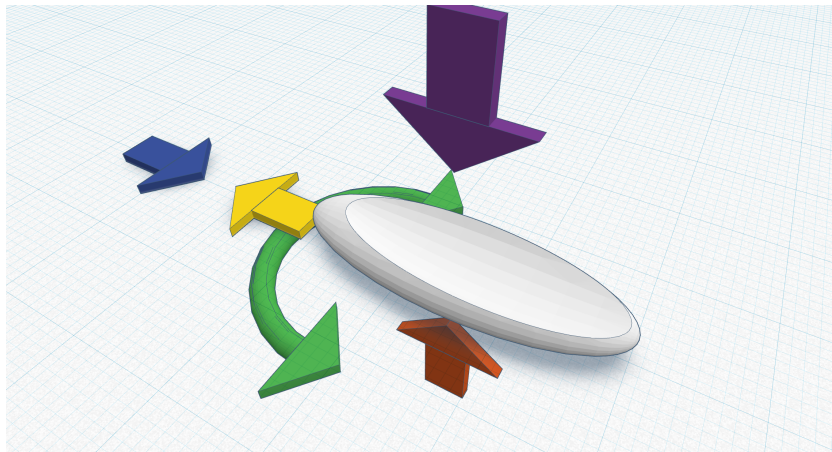


Abb. 4.3: Kräfte und Richtungen die auf das Ruderboot wirken (Quelle: eigene Arbeit)

Schwerkraft, Auftrieb und Strömungswiderstand zuständig. In Abbildung 4.3 ist die Verteilung der Kräfte, die auf das *Ruderboot* wirken, zu sehen, wobei der lilafarbene Pfeil für die Schwerkraft, der orangefarbene Pfeil für den Auftrieb und der blaue Pfeil für den Strömungswiderstand stehen. Um die Pseudophysik effektiv ausspielen zu können, muss sich das Boot in der vom *Level* bereitgestellten Umgebung befinden. Des Weiteren wird eine Schnittstelle nach außen angeboten, damit Kräfte auf das *Ruderboot* wirken können. Die Ruder benötigen in ihrer logischen Funktion Zugriff auf die Positionsdaten des Controllers, der am Rudergerät befestigt ist, um eine Bewegung zu simulieren, die mit dem Bewegungsablauf auf dem Rudergerät synchron ist.

Die *Lenk- und Antriebskomponente* ist für die Auswertung von Interaktionsdaten in Bewegungsdaten zuständig. Die Lenkung kann entweder über die Sensoren oder über den Controller erfolgen. Dementsprechend wird entweder auf die bereitgestellten Daten der Sensoren oder auf die Rotationsdaten des Controllers zugegriffen. Für den Antrieb ist der Ruderstatus der Controller Integration von Bedeutung. Die Lenk- und Antriebsdaten werden ausgewertet, sodass diese Informationen genutzt werden können, um das *Ruderboot* nach Abbildung 4.3 gemäß des gelben Pfeils für den Antrieb und des grünen Pfeils für die Lenkung zu bewegen. Dazu wird der Zugriff auf die Physik des *Ruderbootes* benötigt.

Die verschiedenen Level, zusammengefasst als *Level-Komponente*, sind die Umgebungen welche der Nutzer befahren kann. Dies umfasst ein Parcourslevel, ein Sprintlevel und ein Piratenlevel. Sie können jeweils für sich alleine stehen und umfassen das Wasser, das Terrain, inkl. der dekorativen Elemente wie Bäumen, sowie levelspezifische Elemente wie

Stromschnellen und Piratenschiffe. Das Terrain und die Piratenschiffe besitzen pseudo-physikalische Eigenschaften, wodurch es etwa zu Kollisionen mit dem *Ruderboot* kommen kann. Zwar reagieren die Stromschnellen und Piratenschiffe auf das *Ruderboot*, sind von dessen Existenz aber nicht abhängig und werden erst tätig wenn es sich innerhalb einer gewissen Reichweite befindet.

Die *Trainingsmodus-Komponente* ist für die Überwachung und Darstellung des Trainingsfortschrittes verantwortlich. Für jedes Level ist ein eigener Trainingsmodus vorgesehen, weshalb diese auch unterschiedliche Abhängigkeiten besitzen. Die einzelnen *Trainingsmodus-Komponenten* bestehen jeweils aus einem Logik Teil und einem Userinterface, welches als diegetic UI als Teil des Ruderbootes zum Einsatz kommt. Die *Piraten-Komponente* soll im Piratenlevel die Leben des Spielers überwachen. Diese werden bei einem Treffer einer Kanonenkugel oder bei einer Kollision verringert. Auch wird überprüft ob die Piratenschiffe im Level besiegt wurden. Die UI-Elemente bestehen aus einer Anzeige der verbleibenden Leben und der Information ob man gewonnen oder verloren hat. Die Aufgabe der *Sprint-Komponente* ist es, Informationen während des Trainings anzuzeigen, wie etwa der verstrichenen Zeit oder der aktuellen Geschwindigkeit. Außerdem soll nach Erreichen des Ziels eine Bestenliste, sortiert nach der benötigten Zeit, eingeblendet werden. Auch die *Parcours-Komponente* besitzt eine solche Bestenliste, verzichtet aber auf Anzeigen während des Trainings. Alle *Trainingsmodus-Komponenten* besitzen eine Abhängigkeit zum *Ruderboot*, da sie sowohl für Messungen als auch für die Positionierung der UIs die Position des *Ruderbootes* kennen müssen. Der *Piraten-Trainingsmodus* besitzt eine Abhängigkeit zur *Level-Komponente*, da sie auf die Piratenschiffe zugreifen muss. Die anderen beiden Trainingsmodi benötigen für die Bestenliste, welche Sessionübergreifend verfügbar sein soll, eine *persistente Datenhaltung*.

### 4.3 Abbildung auf das Unity Programmiermodell

Für die Entwicklung der Anwendung wurde Unity als Entwicklungsplattform gewählt. Um die Komponenten aus dem Komponentendiagramm 4.2 auf das Programmiermodell von Unity abzubilden, werden zunächst die wichtigsten Elemente für die Entwicklung in Unity erläutert. Anschließend wird aufgezeigt, in welcher Form die Komponenten der Ruderanwendung Verwendung in diesem Modell finden.

In Unity ist ein *Projekt* die Bezeichnung der Anwendung, an welcher gearbeitet wird. Dieses Projekt setzt sich aus mehreren *Szenen* zusammen. Eine solche Szene kann etwa

ein Level sein und umfasst eine Menge von Elementen, welche in einer hierarchischen Struktur geordnet sind. In dieser auch Szenengraph genannten Struktur, herrscht eine Eltern-Kind Vererbung bei Elementen, die anderen Elementen untergeordnet sind, welche in Form einer Vererbung von Positions- und Rotationsdaten auftritt. Diese Elemente werden in Unity *GameObjects* genannt und stellen Container dar, welche mit Komponenten bestückt werden oder als Sammelobjekt für weitere untergeordnete *GameObjects* dienen können. Eine *Komponente* bietet in Unity eine Funktionalität und muss an ein *GameObject* gebunden werden. Zu diesen Funktionalitäten gehören etwa Kameras, welche einen Blick in eine Szene ermöglichen, Audioquellen, Lichter, Skripte oder auch die Transformationskomponente die jedes *GameObject* standardmäßig besitzt. Funktionen, die nicht mit vordefinierten Komponenten umsetzbar sind, können mit Hilfe von *Skripten* eigenständig entwickelt werden. Diese Skripte werden in der Programmiersprache *C#* geschrieben und müssen an ein *GameObject* gebunden sein, um ausgeführt zu werden. Dadurch erhalten sie auch Zugriff auf das *GameObject* und dessen Komponenten, sodass etwa das Objekt bewegt werden kann. Medieninhalte wie Grafiken oder auch 3D-Modelle werden in Unity *Assets* genannt und können u.a. benutzt werden um Objekten Form und Farbe zu verleihen. Mithilfe von *Prefabs* können *GameObjects* mit ihren Komponenten und Konfigurationen gespeichert werden, sodass sie beliebig oft wiederverwendet werden können. Erstellte Instanzen eines *Prefabs* können darüber hinaus auch noch erweitert werden ohne dass sich dies auf andere Instanzen auswirkt, werden jedoch Änderungen an dem *Prefab* vorgenommen, wirkt sich dies auf alle Instanzen aus.[7]

In der VR-Ruderanwendung wird jeweils eine Szene für ein Level verwendet. In Abbildung 4.4 ist die Struktur der *GameObjects* für eine der umgesetzten Szenen zu sehen. Die einzelnen Elemente des Terrains sind in verschiedene Untergruppen zusammengefasst, z.B. sind alle Steine und Felsen einem Sammel-*GameObject* untergeordnet. Interaktive *GameObjects* wie die Stromschnellen oder die Piratenschiffe besitzen Komponenten die ihnen ihre Funktionalität geben, so hat das Schiff etwa Skripte für die Steuerung, die Schwimmphysik und die Kanonen, sowie eine Komponente für die Grundphysik, wie etwa die Schwerkraft. Diese Komponente wird in Unity *Rigidbody* genannt. Auch das Ruderboot ist ein *GameObject* und besitzt diesen *Rigidbody* für die Grundphysik, sowie die erweiterte Physik für das Schwimmverhalten auf dem Wasser. Weitere *GameObjects*, welche dem Ruderboot untergeordnet sind, sind die Ruder und ein Mesh Objekt. Die Ruder haben keinen physikalischen Körper sondern sind rein optische Objekte, welche ein Skript besitzen, wodurch sie eine Ruderbewegung nachahmen. Das *BoatMesh* beinhaltet ein Modell für die grafische Ausgabe und einen Kollisionskörper. Des Weiteren

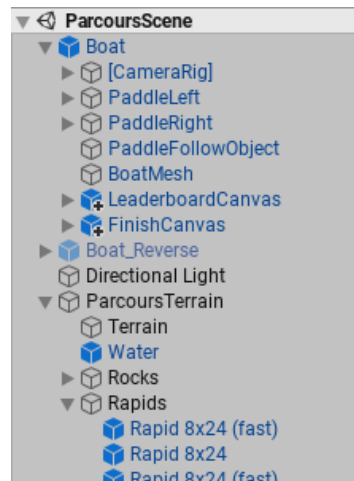


Abb. 4.4: GameObject-Sicht einer Szene im Unity Editor

ist dem Ruderboot die Spieler-Komponente für die Integration des VR-Systems angeheftet, sodass sich der Trackingbereich beim Fahren mit bewegt. Das Ruderboot gibt es in zwei Ausführungen, welche jeweils unterschiedliche Ruder besitzen, außerdem sind beide Varianten als Prefab gespeichert, damit diese grundlegend in jedem Level gleich sind. Erweitert werden diese durch die Komponenten des Trainingsmodus. Hierbei erhält das Spielerboot jeweils pro Level ein anderes UI und ein weiteres Skript, welches für das Überwachen des Trainingsfortschrittes verantwortlich ist. Die Verwaltung für die persistente Datenhaltung besteht aus statischen Funktionen und benötigt deshalb keine Bindung an ein GameObject um aufgerufen zu werden.

Die beiden Interaktionskomponenten bestehen aus insgesamt drei Skripten, eins für die Integration der Controller in die Anwendung und zwei weitere für die Lenkung und den Antrieb. Diese setzen eine Schnittstelle zu den Sensoren um und sind für die Auswertung der Interaktionsdaten in Lenk- und Antriebswerte zuständig. Diese werden daraufhin über einen Funktionsaufruf an den RigidBody des Spielerbootes weitergegeben. Da Zugriffe auf das Spielerboot und die angehefteten VR-GameObjects nötig sind, sind die Skripte an das Spielerboot gebunden und sind außerdem Teil des Prefabs, da sie sich in jedem Level gleich verhalten.



## 4.4 Umsetzung

### 4.4.1 Interaktion

Für die Interaktion mit der virtuellen Umgebung sind die Schnittstellen zu den Eingabegeräten, sowie die Verarbeitung der Informationen von zentraler Bedeutung. Zu diesem Zweck gibt es drei Skripte, welche eine Input-Verarbeitung für den Controller, eine Schnittstelle zu den Sensoren und eine Weiterverarbeitung der Informationen in Lenk- und Antriebsdaten umsetzen.

#### Interaktion - Controller Integration

Das Skript welches für die Integration der Controller in die Anwendung zuständig ist, besitzt drei wesentliche Aufgaben. Zum einen ist dies die Bereitstellung der Information, ob gerade eine Ruderaktion durchgeführt wird und wie weit diese fortgeschritten ist. Weiterhin werden die Positions- und Rotationsdaten des Controllers bereitgestellt und zuletzt wird überwacht, ob einer der Knöpfe des Controllers gedrückt wird, sodass eine zugewiesene Aktion ausgeführt werden kann.

Für die *Erkennung einer Ruderbewegung* sind drei Informationen erforderlich: die Position des aktiven Controllers, ein Threshold-Wert für eine Mindestgeschwindigkeit und der Abstand zwischen der Griffstange in der Ausgangslage und in durchgezogener Position. In festen Zeitschritten wird diese Funktion mehrmals pro Sekunde aufgerufen und speichert die letzte Position des Controllers, die mit der aktuellen Position verglichen werden kann, sodass man eine Bewegung erhält. Mit dieser Information kann ermittelt werden, ob es zu einem Zustandsübergang gemäß Abbildung 4.5 kommen kann, ausgehend vom Startzustand S0. Findet eine Bewegung in -z Richtung also rückseitig statt und ist die Distanz größer als der Threshold-Wert, wird der Ruderstatus auf true gesetzt und die aktuelle Position wird als Startpunkt der Ruderaktion gespeichert. Beim nächsten

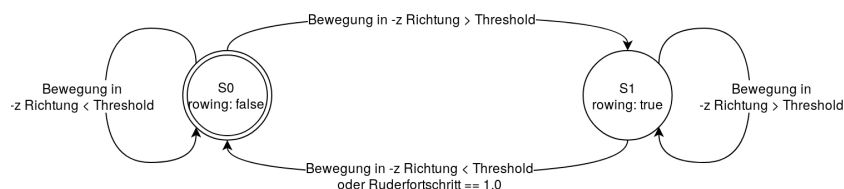


Abb. 4.5: Zustandsautomat über Zustände und Übergänge, zur Bestimmung, ob eine Ruderbewegung durchgeführt wird (Quelle: eigene Arbeit)

Durchlauf wird über das selbe Verfahren geprüft ob die Aktion immer noch durchgeführt wird, wenn nicht wechselt der Zustand und der Ruderstatus wird auf false gesetzt. Wird die Aktion hingegen weiterhin ausgeführt, wird der Ruderfortschritt ermittelt, basierend auf dem Startpunkt, der aktuellen Position des Controllers und der vorher festgelegten Gesamtdistanz einer Ruderbewegung. Daraus ergibt sich ein Wert zwischen 0.0 und 1.0, also ein Prozentwert in Dezimalschreibweise, der angibt wie weit eine Ruderbewegung fortgeschritten ist. Wird der Wert 1.0 erreicht, gilt die Bewegung als abgeschlossen und der Ruderstatus wird auf false gesetzt. Der Ruderstatus und der Ruderfortschritt werden in einem weiteren Schritt für die Berechnung des Antriebs benötigt.

Zur *Bereitstellung der Positions- und Rotationsdaten* wird nur die Information benötigt, welcher Controller momentan der aktive Controller ist, also welcher Controller gerade genutzt wird. Von diesem Controller werden dann die Daten ausgelesen und zur Verfügung gestellt, damit diese etwa für die Lenkung abgerufen werden können.

Für die *Auswertung von Trackpad-, Schultertasten- und Buttonevents* wird mithilfe von SteamVR überprüft, ob eine dieser Tasten gedrückt wurde, also ob ein entsprechend abgefragter Wert auf true gesetzt wurde. Wurde der Button zum Pausieren der Anwendung gedrückt, wird die Zeit angehalten und alle Bewegungen werden eingefroren. Bei einem weiteren Tastendruck läuft die Zeit in normaler Geschwindigkeit weiter. Sollte der Button zum Wechsel der Steuerungsart gedrückt werden, wird ein entsprechender boolean-Wert gesetzt, welcher von anderen Skripten eingesehen werden kann. Das Trackpad bietet gleich zwei Informationen, einmal die Information ob es gedrückt wurde und einen Vektor mit 2 Werten, welche die x und y Werte liefern wo das Trackpad berührt wird. Aus diesen Werten lässt sich ermitteln ob das Trackpad im oberen, unteren, linken oder rechten Viertel gedrückt wurde. Wurde das Trackpad nun gedrückt, wird zunächst geprüft ob der Controller auch der aktive Controller ist, also derjenige der die Informationen für den Ruderstatus liefert. Wenn dies nicht der Fall ist, wird er als aktiver Controller gesetzt, andernfalls wird eine Aktion ausgeführt abhängig davon in welchem Bereich das Trackpad gedrückt wurde. Wurde der linke oder rechte Bereich gedrückt, wird das vorherige bzw. nächste Level geladen. Hierfür wird eine Liste mit den Namen der Szenen benötigt um herauszufinden, welches das Folgelevel ist. Wird hingegen der obere oder untere Bereich gedrückt, wird das aktuelle Level neu geladen, jedoch wird beim oberen Bereich das Vorwärtsruderboot und im unteren Bereich das klassische rückwärtsgewandte Ruderboot jeweils als das aktive Ruderboot gesetzt.

### Interaktion - Schnittstelle Sensoren

Für die Integration der Sensoreinheit in die Anwendung, müssen sich diese beiden austauschen können. Seitens der Anwendung geschieht dies über ein Skript. Der Arduino, der in der Sensoreinheit verwendet wird, überträgt die Informationen seinerseits in einem Sketch (die Bezeichnung für ein Arduino Programm) an den Computer. Die Übertragung findet in Form einer seriellen Datenübertragung über USB statt.

Die Information welche übertragen wird, beschränkt sich auf einen Zahlenwert der zwischen -2046 und 2046 liegt und Auskunft darüber gibt, wie stark die Gewichtsverteilung auf den Sensoren nach links oder rechts ist. Das Skript zum Einlesen dieser Information besitzt drei Funktionen: Open, Close und Read und benötigt einen Port und eine Baudrate zum Aufbau der Verbindung. Über Open und Close kann die Schnittstelle geöffnet bzw. geschlossen werden. Mit Read wird immer der jeweils aktuellste ausgelesene Wert ausgegeben und ältere, sofern vorhanden, werden verworfen. Sollte kein Wert zum Auslesen vorhanden sein, wird „0“ zurückgegeben.

### Interaktion - Antrieb und Lenkung

Damit auf die vom Nutzer erbrachten Aktionen auch eine Reaktion in der virtuellen Welt folgt, muss der Input weiterverarbeitet werden. Dafür werden auf Basis der bereitgestellten Daten, Werte für die Lenkung und den Antrieb des Ruderbootes erstellt. Diese Daten werden als Kraft an die Physik-Komponente des Bootes weitergegeben.

Um die Informationen darüber, ob gerade gerudert wird und wie weit die Aktion fortgeschritten ist, in eine Beschleunigung umzusetzen, hilft ein Blick auf die Messdaten echter Trainingseinheiten. Es ist zu erkennen, dass sich die auf das Trainingsgerät wirkende Kraft wie eine Kurve verhält. Dabei sind die Phasen Catch und Finish der Ruderübung von Bedeutung, da sie den Beginn und das Ende der Zugbewegung darstellen<sup>4</sup>. Während die Kraftkurve vor dem Catch noch relativ nahe bei Null liegt, zeigt sich danach ein sehr steiler Anstieg. Der höchste Punkt der Kurve wird etwa bei der Hälfte der Zugbewegung erreicht, danach fällt diese gegen Null zurück. [9][12] Zwar fällt eine solche Kurve für jede Wiederholung etwas anders aus, jedoch lässt sich für den Zweck der Anwendung eine solche Kurve in einer stark vereinfachten Form modellieren um die Beschleunigung des Bootes zu realisieren. Zu diesem Zweck wird die Kurve in Abbildung 4.6 verwendet, die sich aus der Formel  $-4x^2 + 4x$  ergibt. Von Interesse ist dabei der Bereich auf der x-Achse von 0.0 bis 1.0.

---

<sup>4</sup><http://mspfitness.com/a-concept2-rowing-tutorial-the-four-phases-of-rowing/> zuletzt aufgerufen am 11.08.2019

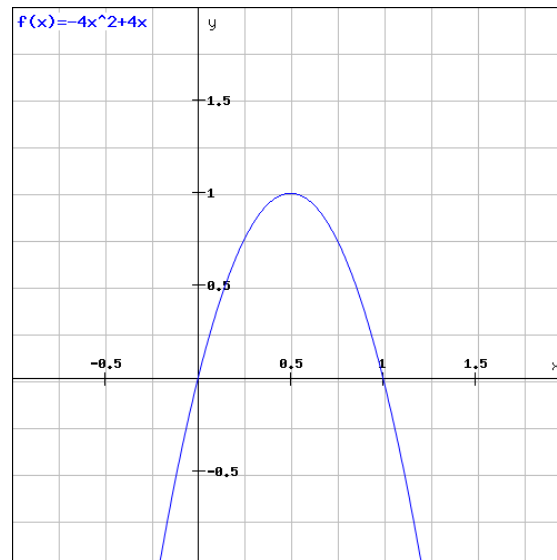


Abb. 4.6: Kurve einer quadratischen Funktion, welche die x-Achse an den Punkten Null und Eins schneidet. Die x-Achse steht für den Ruderfortschritt und die y-Achse für die Kraft. y ist normalisiert, damit es mit einem variablen Kraftwert multipliziert werden kann. (Quelle: eigene Arbeit)

Findet nun eine Ruderbewegung statt, lässt sich der Ruderfortschritt, welcher ebenfalls zwischen 0.0 und 1.0 liegt, für x in die Formel einsetzen, sodass man einen normalisierten y-Wert entsprechend dem Graphen erhält. Nun wird eine maximale Kraft in Newton benötigt, die auf das Ruderboot wirken kann. Diese Kraft wird mit dem y-Wert multipliziert, sodass eine Kraft für den aktuellen Ruderfortschritt resultiert, die auf der Kraftkurve basiert.

Bei einer zu langsamen oder ruckartigen Durchführung der Ruderbewegung wird der errechnete Kraftwert verringert. Hierfür wird eine Zeitvorgabe für die Dauer einer Ruderbewegung definiert. Der nun finale Kraftwert wird im letzten Schritt als vorwärts gerichtete Kraft an die Physik-Komponente des Ruderbootes weitergegeben, was zu einer Beschleunigung führt. Die Werte der maximalen Kraft und der optimalen Dauer der Ruderbewegung können vom Nutzer verändert werden, um sie an die Einstellung des Rudergerätes und an das eigene Trainingsverhalten anzupassen.

Die Lenkung des Bootes soll sowohl über die Sensoren als auch über den Controller funktionieren. Bei diesen Verfahren handelt es sich allerdings nicht um Abstraktionen realer Lenkmöglichkeiten, da dafür je eine Griffstange pro Ruder nötig wäre. Stattdessen stehen

die beiden Lenkmöglichkeiten zur Auswahl, damit der Nutzer die für sich angenehmere Lenkung wählen kann.

Bei Verwendung der Sensoren, werden die Informationen welche die Sensoreinheit sendet in festen Zeitschritten abgerufen. Der abgerufene Wert gibt aus, wie stark die Belastung der Sensoren links- oder rechtslastig ist. Um zu verhindern, dass jemand bei dem Versuch immer stärker einzulenken vom Rudergerät fällt, gibt es einen Maximalwert für die Messergebnisse. Alles was darüber liegt, wird auf den Maximalwert reduziert. Als nächster Schritt wird der Messwert durch den Maximalwert geteilt, sodass sich ein Wert zwischen -1.0 und 1.0 ergibt. Dieser Wert wird mit einem Multiplikator verrechnet, der dafür verantwortlich ist wie stark eine Drehung maximal sein darf. Zuletzt wird dieser Wert in einem Funktionsaufruf an die Physik-Komponenten des Ruderbootes weitergegeben, sodass eine Drehung erfolgt.

Die Lenkung über den Controller wird ähnlich ausgewertet, statt der Messergebnisse wird jedoch die Rotation des Controllers verwendet. Der Maximalwert sagt in diesem Fall aus, wie weit der Controller gedreht werden muss, um die maximale Lenkstärke zu erhalten. Das Teilen durch den Maximalwert, die Verrechnung mit dem Multiplikator, sowie die Beteiligung der Physik-Komponente verlaufen gleich wie bei den Sensoren.

### 4.4.2 Sensoreinheit

Die Sensoreinheit hat die Aufgabe mithilfe von Sensoren eine Gewichtsverteilung zu messen. Zu diesem Zweck kommen Drucksensoren zum Einsatz, welche auf dem Rudergerät verteilt und befestigt werden. Der Druck wird von einem Arduino gemessen und basierend darauf wird ein Wert an den Computer gesendet.

Die Abbildung 4.7 zeigt, wie die Verkabelung zwischen den Hardware-Elementen der Sensoreinheit realisiert ist. Die Sensoren werden mit einem Pin an eine 5V Stromversorgung des Arduino angeschlossen. Von dem zweiten Pin führt ein Kabel zu einer analogen Schnittstelle, weiter zu einem 100 k $\Omega$  Widerstand, der daraufhin an Masse anliegt. Die Drucksensoren agieren als Widerstände, die je nach Belastung einen höheren oder niedrigeren Widerstandswert aufweisen. Mit der analogen Schnittstelle kann die anliegende Spannung gemessen werden, die sich in einem Zahlenwert zwischen 0 und 1023 widerspiegelt. Wenn kein Druck auf die Sensoren ausgeübt wird, haben sie einen sehr hohen Widerstandswert, sodass fast keine Spannung an der analogen Schnittstelle anliegt, was

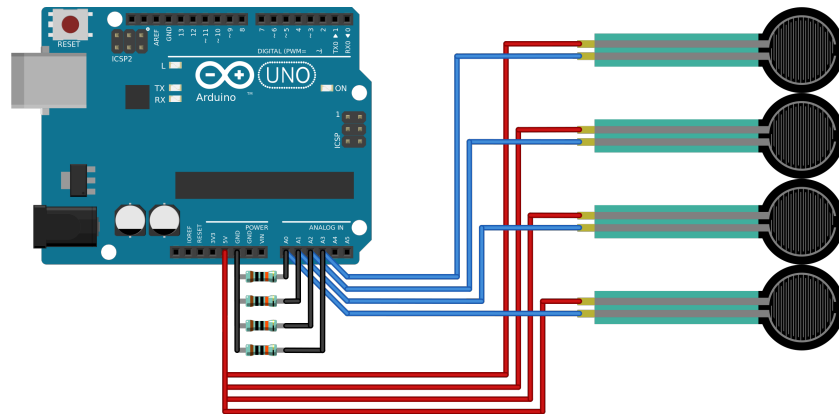


Abb. 4.7: Verkabelung der Drucksensoren und Widerstände mit dem Arduino Uno (Quelle: eigene Arbeit)

zum Messwert 0 führt. Mit zunehmender Last nimmt der Widerstand der Sensoren ab, was sich in einem höheren Messwert widerspiegelt.

In den Abbildungen 4.8 und 4.9 ist zu sehen wie die Sensoren und der Arduino an dem Rudergerät verbaut wurden. 4.8 zeigt die Oberseite des Sitzes, auf dem die Sensoren an den äußeren Rändern gegenüberliegend befestigt wurden. Für das Training wird zusätzlich eine Schaumstoffmatte auf den Sitz gelegt, um eine bessere Verteilung des Gewichtes und besser verwertbare Messergebnisse zu erzielen. 4.9 zeigt die Verlegung der Kabel, welche von den Sensoren zum Arduino verlaufen. Dieser befindet sich in einem schwarzen Gehäuse und ist an der Unterseite der Sitzfläche befestigt, sodass der Schlitten bewegt



Abb. 4.8: Verteilung der Sensoren auf dem Sitz des Rudergerätes

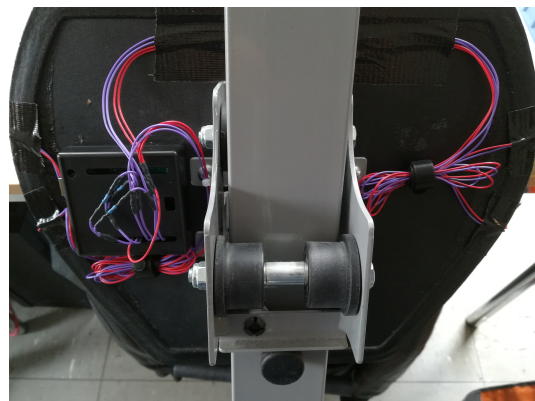


Abb. 4.9: Befestigung des Arduino und der Kabel auf der Rückseite des Sitzes vom Rudergerätes

werden kann ohne dass die Sensoreinheit im Weg ist. An den Arduino wird ein langes USB-Kabel angeschlossen, um die Verbindung mit dem Computer herzustellen.

Das Sketch welches auf dem Arduino läuft liest mehrmals pro Sekunde die analogen Schnittstellen aus, addiert jeweils die Werte der linken und der rechten Sensoren und subtrahiert die beiden Zwischenwerte. Heraus kommt ein Wert zwischen -2046 und 2046, der bei einer Gleichbelastung bei 0 liegt. Dieser Wert wird daraufhin über eine serielle Schnittstelle an den Computer übertragen.

### 4.4.3 Ruder

Die Ruder sollen der Immersion dienen und erfüllen keinen weiteren praktischen Zweck, da der Antrieb nicht auf Basis der Ruder umgesetzt wurde. Dementsprechend sollen diese nur simuliert werden und die Bewegung realer Ruder nachahmen. Die Ruder gibt es in zwei Varianten, jeweils eine für jede Ausführung des Ruderbootes nach 3.1.5.



Abb. 4.10: Klassisches Ruder



Abb. 4.11: Invertiertes Ruder

In den Bildern 4.10 und 4.11 sind die unterschiedlichen Ruder zu sehen. Während das erste Ruder ein einziges bewegliches Element ist, besteht das Ruder im zweiten Bild aus zwei Elementen. Diese beiden Elemente bewegen sich an ihrem Treffpunkt gespiegelt zueinander, da dies in etwa der Bewegung eines Ruders von einem Vorwärtsruderboot entspricht. Dieser Punkt ist bei beiden Varianten außerdem der Punkt, an dem die Ruder fest an das Ruderboot fixiert sind, wohingegen die anderen Enden bewegt werden können.

Um nun die Bewegung der Ruder zu simulieren, sollen die Griffenden zum Controller hin ausgerichtet werden und sich bei der Benutzung des Rudergerätes mitbewegen. Die

Bewegung auf dem Rudergerät ist jedoch linear und die Ruder würden nicht wie bei einem echten Ruderboot ins Wasser ein- und auftauchen. Aus diesem Grund wird ein unsichtbares GameObject verwendet, welches die Position des Controllers kopiert, aber dessen Höhenwert, also die y-Achse, verändert. Nach diesem Objekt richten sich die Griffenden der Ruder aus. Abhängig davon ob gerade gerudert wird oder nicht, wird das Objekt angehoben, bzw. abgesenkt, was dazu führt, dass die Ruder ins Wasser ein- oder auftauchen. Die Position des Controllers, sowie die Information ob gerade gerudert wird, erhält das Skript von der Controller Integration.

### 4.4.4 Ruderboot

Das Ruderboot wird durch ein GameObject repräsentiert und stellt einen Container für mehrere Elemente dar, wie etwa die Interaktionskomponenten und die Ruder. Darüber hinaus beinhaltet das Ruderboot eine eigene Pseudophysik, zusammengesetzt aus einer von Unity gegebenen Grundphysik und einer erweiterten Physik. Die Grundphysik bietet etwa Schwerkraft oder auch die Verarbeitung von Kollisionen an, wohingegen die erweiterte Physik für die Bewegung auf dem Wasser nötig ist. Das Ruderboot ist außerdem der Punkt an dem der Benutzer über die Spieler-Komponente in die virtuelle Welt eingebettet ist. Das Boot ist in zwei Ausführungen vorhanden, welche sich in der Art der Ruder und der Ausrichtung der VR-Elemente.

#### **Pseudophysik**

Die Pseudophysik, die dafür sorgt, dass sich das Boot über Wasser hält und annähernd realistisch verhält, hat drei wesentliche Aufgaben. Die erste ist die Berechnung des Auftriebs, als Gegenkraft zur Schwerkraft, die zweite ist die Berechnung des Strömungswiderstandes, der die Bewegungen des Bootes verlangsamt und schließlich eine Ausrichtung zur Anfangsrotation, sodass sich das Boot nach einer Kollision wieder begradigt. Einige dieser Funktionen benötigen Informationen über die Form des Ruderbootes, zu diesem Zweck werden zum Start der Anwendung die Höhe, das Volumen und einige Querschnittsflächen auf Basis des Hüllkörpers des Bootes geschätzt.

#### **Auftrieb**

Für den Auftrieb ist das Verdrängungsvolumen entscheidend, mit welchem das Ruderboot das Wasser verdrängt. Durch die höhere Dichte des Wassers ergibt sich für das Boot eine Auftriebskraft. Für die Berechnung wird zunächst ermittelt wie weit das Boot in das



Wasser eingetaucht ist. Diese Information wird mit dem Gesamtvolumen des Bootes verrechnet, sodass das Volumen resultiert, mit welchem das Wasser verdrängt wird. Mit diesem Verdrängungsvolumen ( $V$ ), der Wasserdichte ( $\rho$ ) und dem Ortsfaktor ( $g$ ) kann die Auftriebskraft ( $F_A$ ) nach der Formel  $F_A = \rho * V * g$  berechnet werden. [11] Diese ermittelte Kraft wird daraufhin als aufwärts gerichtete Kraft an den Rigidbody des Ruderbootes weitergegeben. Befindet sich das Boot nicht im Wasser wird diese Berechnung nicht durchgeführt.

### **Strömungswiderstand**

Der Strömungswiderstand ist eine Kraft die wirkt, wenn sich das Boot bewegt und durch das Wasser ausgebremst wird. Diese Kraft berechnet nach der Formel  $F_W = \frac{1}{2} * c_W * A * \rho * v^2$ . Die Variable  $c_W$  steht für den Widerstandsbeiwert und ist eine Zahl die von der Form des betreffenden Körpers, also des Bootes, abhängig ist. Eine angeströmte Halbkugel hat etwa einen Wert von 0,4 und ein Körper mit Stromlinienform 0,055 [20].  $A$  steht für die umströmte Querschnittsfläche des Körpers,  $\rho$  für die Dichte des Wassers und  $v$  für die Geschwindigkeit des Bootes, welche zum Quadrat genommen wird. [19] Diese Widerstandskraft wird für die Geschwindigkeit auf der x- und z-Achse des Bootes jeweils separat berechnet. Für die Höhenachse y wird dies ebenfalls berechnet, wenn sich das Boot in Richtung des Wassers bewegt. Die errechneten Widerstandskräfte werden zuletzt als Gegenkraft zur aktuellen Bewegung auf das Ruderboot gewirkt. Auch diese Berechnungen finden nur statt, wenn sich das Boot im Wasser befindet.

### **Ausrichtung zur Anfangsrotation**

Bei einer Veränderung der Rotation der x- oder z-Achse, wird eine Rotation in Richtung der Anfangsrotation mit einer voreingestellten Geschwindigkeit durchgeführt. Die y-Achse wird ignoriert, da diese nicht dafür ausschlaggebend ist, ob das Boot waagrecht auf dem Wasser liegt. Zusätzlich kann Schaukeln aktiviert werden, was dazu führt, dass kurz vor Erreichen der Ausgangsrotation ein geringer Kraftwert wirkt, sodass es zu einem seichten Schaukeln kommt.

#### **4.4.5 Level**

Das Level umfasst die meisten Elemente, die der Benutzer sehen kann, die aber nicht direkt mit dem Ruderboot in Verbindung stehen. Zu diesen Elementen gehören das Terrain, das Wasser, Bäume, Steine, die Piratenschiffe und die Stromschnellen. Jedes Level

hat einen anderen Aufbau um den Anforderungen der jeweiligen Trainingsmodi gerecht zu werden. Die Terrains wurden mit Tools erstellt, welche Unity für diesen Zweck selbst anbietet.

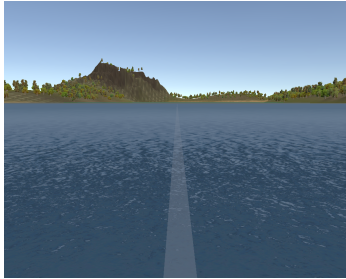


Abb. 4.12: Sprintlevel



Abb. 4.13: Parcourslevel



Abb. 4.14: Piratenlevel

### Gestaltung

Für das *Sprintlevel*, zu sehen in Abbildung 4.12, dient ein großer See, umgeben von Bäumen und Bergen, als Trainingsumgebung. Zusätzlich gibt es eine Linie auf dem Wasser, welche die Fahrtrichtung angibt. Dieser Aufbau soll dazu dienen, dass der Nutzer sich auf das Training konzentrieren kann und nicht von irgendwelchen Hindernissen abgelenkt oder aus der Spur gebracht wird. Das *Parcourslevel* in Abbildung 4.13 spiegelt einen Fluss wieder, der einen Start- und Endpunkt hat. Diese Punkte stellen den Start und das Ziel des Trainings dar. Dazwischen verläuft der Fluss sehr kurvenreich und ist mit vielen Steinen als Hindernis gespickt. In etwa der Mitte des Levels befindet sich eine aus dem Wasser hervorragende Insel, an welcher der Nutzer vorbei rudern muss. Neben den Steinen gibt es noch Stromschnellen auf dem Flussverlauf anzutreffen. Die Stromschnellen sind animiert und üben eine Kraft auf Objekte aus, die darin hinein fahren. Das Level ist so gestaltet, dass der Nutzer oft Hindernissen ausweichen muss, z.T. mehrere Möglichkeiten hat die Strecke zu befahren und sich überlegen muss ob er Stromschnellen für sich nutzt oder ihnen lieber ausweicht. Das *Piratenlevel*, zu sehen in Abbildung 4.14, besteht aus einem sehr großen See, in welchem sich mehrere Inseln befinden. Zusätzlich sind einige Piratenschiffe verteilt. Die Gestaltung des Levels soll dazu dienen, dass der Nutzer die Piratenschiffe gegen die Inseln lotsen kann um diese auszuschalten.

### Piratenschiff

Während die Gestaltung der Level überwiegend im Unity Editor umgesetzt wurde, benötigt das Piratenschiff Funktionalitäten, welche über ein Skript entwickelt werden müssen.

Dies umfasst die Steuerung des Piratenschiffes, das Verhalten der Kanonen und die Reaktion auf eine Kollision. Die Pseudophysik wird über die selben Komponenten wie beim Ruderboot realisiert.

Für die Steuerung des Schiffes ist vorgesehen, dass es sich in Bewegung setzt, wenn sich das Boot des Spielers in einer gewissen Reichweite befindet. Für die Beschleunigung wird regelmäßig eine Kraft auf das Schiff ausgeübt, welche über eine Zeitspanne bis zu einem maximalen Wert ansteigt, in einer Kurvenfahrt wird der Wert jedoch verringert. Damit es nicht zu einer Kollision mit dem Ruderboot kommt, soll dies nicht direkt vom Schiff angesteuert werden. Stattdessen werden Punkte die links und rechts vom Ruderboot liegen anvisiert, wohin sich das Schiff bewegen soll. Die Wahl welcher Punkt angepeilt wird ist abhängig davon wie das Schiff relativ gesehen zum Boot steht. Als Fahrmuster resultiert eine Kreisfahrt um das Ruderboot herum.



Abb. 4.15: Beispielhafte Darstellung der Bereiche, in denen das Schiff Kanonenkugeln abfeuert

Wenn das Schiff nun eine gewisse Nähe zum Ruderboot hat, soll aus den Kanonen gefeuert werden. Dazu werden die Kanonen in die Bereiche Vorne, Mitte, Hinten eingeteilt. Abhängig davon in welchem Grad-Bereich sich das Ruderboot nach Abbildung 4.15 befindet, wird einer dieser Bereiche gewählt um die Kugel abzuschießen. Anhand einer Feuerrate und dem Zeitpunkt an dem die letzte Kugel abgefeuert wurde, wird überprüft ob eine neue Kugel geschossen werden darf. Ist dies der Fall wird eine zufällige Kanone aus dem Bereich gewählt, in dem sich das Ruderboot befindet, an deren Vorderseite eine neue Instanz einer Kanonenkugel erstellt wird. Diese Kugel erhält eine Kraft, die von der Kanone weg gerichtet ist, sodass es aussieht als würde die Kugel aus der Kanone geschossen kommen. Diese Kraft setzt sich aus einem Grundwert und einem Zufallswert zusammen. Der Grundwert ergibt sich aus der Distanz vom Schiff zum Ruderboot, der Zufallswert liegt in einem gewissen Zahlenbereich und soll den Effekt haben, dass die Kugel nicht

immer gleich weit fliegt. Dadurch landet die Kugel oft nahe am Boot, trifft es aber nicht allzu oft. Verlässt die Kugel die Spielwelt, also fällt sie unten aus dem Wasser, wird sie entfernt. Durch eine Anpassung der Feuerrate lässt sich auch der Schwierigkeitsgrad variieren.

Wenn es zu einer Kollision mit einem Objekt kommt, überprüft ein Skript worum es sich bei dem anderen Objekt handelt. Kollidiert das Schiff mit einem anderen Piratenschiff oder mit dem Terrain, wird das Bewegungsskript deaktiviert und die schwarzen Segel des Schiffes werden durch weiße Segel ersetzt. Andere Objekte werden nicht beachtet und die weitere Kollisionsverarbeitung wird Unity überlassen.

### 4.4.6 Trainingsmodi

Die Trainingsmodi geben dem Benutzer ein zu erreichendes Ziel und Herausforderung in den verschiedenen Leveln. Die Trainingsmodi unterscheiden sich dabei von Level zu Level. Durch ständiges Messen des Fortschrittes kann ermittelt werden, wann der Nutzer einen Trainingsmodus abgeschlossen hat und es ermöglicht die Wiedergabe der Fortschrittsdaten während und nach dessen Beenden. Zuletzt werden diese Daten durch ein diegetic UI, welches an das Ruderboot geheftet ist, visualisiert.

#### Parcours-Trainingsmodus

Für das *Parcourslevel* gibt es eine Zeitmessung wie lange der Nutzer benötigt um ein Ziel zu erreichen. Das Ziel gilt als erreicht, wenn eine bestimmte Distanz zum Startpunkt hergestellt wurde. Die Zeitmessung wird gestartet, sobald sich das Ruderboot in Bewegung setzt. Das UI bleibt ausgeblendet und wird erst sichtbar sobald die Zieldistanz erreicht wurde. Darauf werden dann die benötigte Zeit und ein Zeitstempel angezeigt. Auf einem zweiten Element werden die besten fünf Einträge aus einer Bestenliste angezeigt. Dazu wird zunächst ein neuer Eintrag in der Bestenliste erstellt, woraufhin diese neu sortiert wird. Die Einträge auf dem UI zeigen dann die aktuellen Bestzeiten, dargestellt mit der benötigten Zeit und einem Zeitstempel.

#### Sprint-Trainingsmodus

Die Bestenliste und Zeitmessung für das *Sprintlevel* funktionieren nach dem selben Prinzip wie beim Parcourslevel, lediglich bei der Auswahl der Einträge für die Bestenliste wird zusätzlich noch nach der zurückzulegenden Distanz gefiltert, da diese einstellbar ist

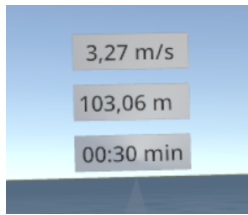


Abb. 4.16: UI während des Trainings im Sprintlevel

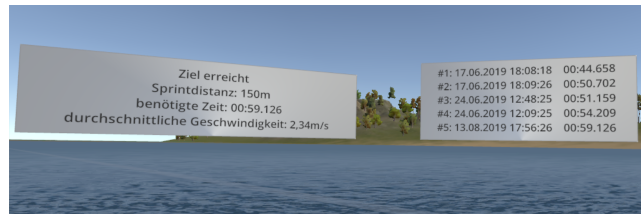


Abb. 4.17: UI nach Erreichen des Ziels im Sprintlevel

und die Bestzeiten nur für die aktuelle Sprintdistanz angezeigt werden sollen. Zusätzlich dazu werden während des Trainings unterschiedliche Informationen angezeigt. Dies umfasst die zurückgelegte Distanz, die aktuelle Geschwindigkeit und die vergangene Zeit, zu sehen in Abbildung 4.16. Die Distanz und Zeit lassen sich auf Basis der Startzeit und des Startpunktes ermitteln und die Geschwindigkeit wird vom RigidBody des Ruderbootes bereitgestellt. Nach Erreichen des Ziels werden neben der Bestenliste noch die zurückgelegte Distanz, die benötigte Zeit und die durchschnittliche Geschwindigkeit angezeigt, wie in Abbildung 4.17 zu sehen. Für die durchschnittliche Geschwindigkeit wird in festen Zeitschritten mehrmals pro Sekunde die aktuelle Geschwindigkeit einer Liste angefügt. Diese Werte werden zum Schluss alle aufaddiert und durch die Gesamtmenge an Einträgen dividiert.

### Piraten-Trainingsmodus

Im *Piratenlevel* hat der Spieler eine Anzahl an Leben, welche über ein UI angezeigt werden. Diese Leben werden bei einer Kollision mit einer Kanonenkugel um eins reduziert. Bei einer Kollision mit einem Piratenschiff oder dem Terrain werden diese Leben auf Null gesetzt. In diesem Fall werden alle Piratenschiffe deaktiviert und es wird eine Anzeige eingeblendet, auf welcher „Verloren“ steht. Schließt der Nutzer das Level hingegen erfolgreich ab, wird „Gewonnen“ angezeigt. Um dies zu erreichen müssen die Piratenschiffe mit dem Terrain oder sich gegenseitig kollidieren, damit diese deaktiviert werden. Der Status aller im Level vorhandenen Piratenschiffe wird dazu regelmäßig überwacht und sobald alle Piratenschiffe deaktiviert wurden, ist das Level erfolgreich abgeschlossen.

### Bestenliste

Das Laden und Speichern einer Bestenliste erfolgt über statische Funktionen. Eine Bestenliste ist eine Liste die eine Menge von Objekten beinhaltet. Diese Objekte sind Instanzen einer Score-Klasse, die Informationen wie einen Zeitstempel, eine Zeitspanne und

eine Distanz besitzen. Zum Speichern und Laden dieser Bestenliste wird diese serialisiert und in eine Datei geschrieben, bzw. geladen und deserialisiert. Die Angabe eines Dateinamens beim Funktionsaufruf erlaubt es für jeden Trainingsmodus eine eigene Bestenliste zu pflegen.

### 4.5 Zusammenfassung

In diesem Kapitel wurde der Entwurf der VR-Ruderanwendung vorgestellt. Dazu wurde im Abschnitt 4.1 die Verteilung der Software auf die Hardware und die Schnittstellen zwischen der Hardware erläutert, gestützt durch das Verteilungsdiagramm 4.1. In Abschnitt 4.2 dieses Kapitels wurden die Softwarekomponenten des Komponentendiagramms 4.2 erläutert. Dieses Diagramm zeigt die einzelnen Komponenten, die für die entwickelte Software von Bedeutung sind. Nacheinander wurden der Zweck, die Art und die Abhängigkeiten der Komponenten beschrieben. Im Abschnitt 4.3 ging es weiter mit einer Abbildung dieser Komponenten auf das Programmiermodell von Unity. Dazu wurde das Programmiermodell zunächst knapp erläutert und anschließend wurde beschrieben, welche Komponente mit welchen Elementen in Unity umgesetzt wurde.

In dem letzten und längsten Abschnitt 4.4 wurde die genaue Umsetzung der Komponenten vorgestellt. Dies begann mit einer Erläuterung der Interaktionskomponenten in Abschnitt 4.4.1, die für die Verwaltung der Controller und Sensoreinheit sowie die Lenkung und den Antrieb zuständig sind. Darauf folgte in Abschnitt 4.4.2 eine Beschreibung, wie die Sensoreinheit umgesetzt wurde. Die Umsetzung der Ruder sowie des Ruderbootes und seiner Pseudophysik sind in den Abschnitten 4.4.3 und 4.4.4 zu finden. Abschnitt 4.4.5 erläutert, wie die unterschiedlichen Level gestaltet wurden und wie das Piratenschiff umgesetzt wurde. Zuletzt wurden in Abschnitt 4.4.6 die verschiedenen Trainingsmodi beschrieben. Die Komponenten wurden auf der Grundlage der funktionalen Anforderungen im Kapitel 3.1 entwickelt und erfüllen die dort beschriebenen Anforderungen. Die nicht funktionalen Anforderungen aus Kapitel 3.2 wurden ebenfalls berücksichtigt und es wurden die Möglichkeiten zur Konfiguration integriert.

# 5 Experimentelle Überprüfung

Um eine erste Einschätzung zu erhalten, ob Leute bereit wären mit einem VR-Exergame zu trainieren, und um zu bewerten, wie die entwickelte Anwendung die Aspekte Bedienbarkeit, Risiko von Cybersickness und die Eignung der Level erfüllt, wurde die VR-Ruderanwendung von einigen Personen getestet. Im Abschnitt 5.1 wird der dazugehörige Versuchsaufbau beschrieben. Im Abschnitt 5.2 findet anschließend eine Auswertung von Fragen statt, zu denen die Testpersonen ihre Einschätzung geben sollten.

## 5.1 Versuchsaufbau

Vor den Tests wurde den Probanden kurz vorgestellt worum es sich bei der VR-Ruderanwendung handelt, wie diese kontrolliert wird und wie die Tastenbelegung ist. Des Weiteren wurden die Aufgaben in den verschiedenen Leveln erläutert. Bei der Erprobung hatten die Testpersonen die Möglichkeit nach eigenem Belieben die VR-Ruderanwendung auszuprobieren. Es gab keine Zeitlimits und keine Vorgaben, welche Level und Lenkungsarten getestet werden sollen. Dementsprechend hat nicht jeder alle Level getestet. Dies findet sich durch einen entsprechenden Eintrag in den Antworten wieder.

Im Anschluss an den Testlauf hat jede Testperson einen Fragebogen ausgefüllt, in der sie ihre Einschätzung zu Fragen über das Training abgeben konnte. Dies umfasste Fragen über die eigene Einstellung zum Training, die Bedienbarkeit der Anwendung, die Steuerung, sowie die Eignung der Level für den ihnen angedachten Zweck. Zusätzlich konnte weiteres Feedback über ein Freifeld gegeben werden. Der Fragebogen umfasst insgesamt zehn Fragen.

## 5.2 Auswertung

Insgesamt haben 15 Leute die Anwendung getestet und einen Fragebogen ausgefüllt. Die Erstellung der Fragebögen und die Überführung der Antworten in Diagramme wurde mit Google Forms<sup>1</sup> umgesetzt. Die einzelnen Fragen und ihre Antworten werden im Folgenden aufgeführt.

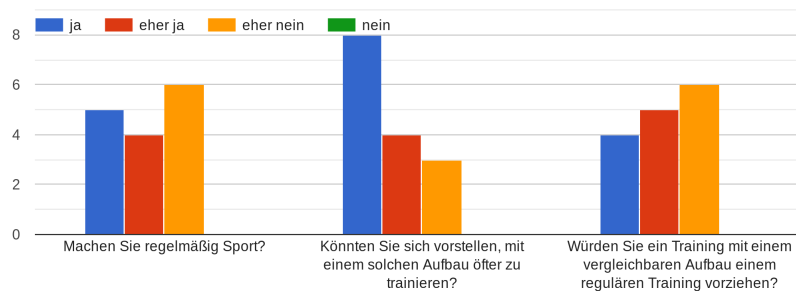


Abb. 5.1: Balkendiagramm über Fragen zur allgemeinen Einstellung zum Training

### Fragen 1-3

Die ersten drei Fragen sollten feststellen, ob die Testpersonen bereits sportlich aktiv sind und ob sie sich ein Training mit einem vergleichbaren Aufbau vorstellen könnten, bzw. es sogar bevorzugen würden. Es wurde explizit nicht danach gefragt, ob sie mit dem vorgestellten System, sondern mit einem vergleichbaren Aufbau trainieren würden. Dies soll eine erste Einschätzung liefern, ob die Testpersonen sich ein richtiges Training in einer virtuellen Umgebung vorstellen könnten, also ob das Potenzial für solche Anwendungen im allgemeinen besteht, unabhängig von ihrer Einstellung dieser expliziten Anwendung gegenüber.

Die Antworten (siehe Abbildung 5.1) ergaben, dass bereits der etwas größere Teil der Testpersonen zumindest gelegentlich Sport macht. Außerdem kann sich der Großteil vorstellen mit einer solchen Anwendung zumindest öfter zu trainieren, einem regulären Training würden es aber deutlich weniger Probanden vorziehen. Dennoch ist auch hier der Anteil der *ja* und *eher ja* Stimmen recht hoch. Unter Betrachtung der Einzelantworten lässt sich außerdem feststellen, dass Leute, die eher nicht regelmäßig Sport machen, ein Training mit einem vergleichbaren Aufbau einem regulären Training vorziehen würden.

<sup>1</sup><https://www.google.com/forms/about/>



So gaben drei Leute, die die erste Frage mit *eher nein* beantwortet haben, bei der dritten Frage ein *ja* und zwei weitere ein *eher ja* als Antwort an. Im Vergleich dazu haben drei Leute, die die erste Frage mit *ja* beantwortet haben, die letzte mit *eher nein* beantwortet. In einer der Anmerkungen wurde darauf hingewiesen, dass das VR-Headset beim Training evtl. zu warm ist, was durchaus einen Einfluss die Bereitschaft zum Training mit einem VR-System haben könnte.

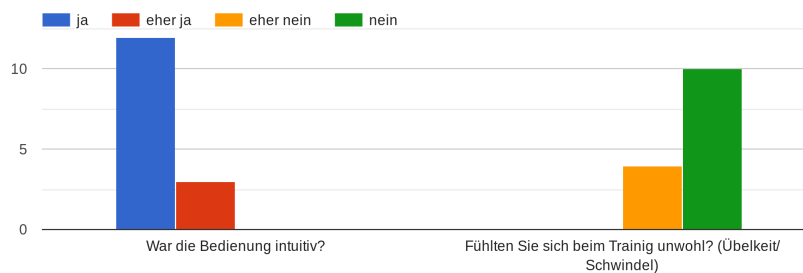


Abb. 5.2: Balkendiagramm über Fragen zur Nutzbarkeit der Anwendung

### Fragen 4-5

Die nächsten beiden Fragen sollten ermitteln, wie nutzbar die Anwendung ist. Dazu wurde gefragt, ob die Bedienung intuitiv war und ob den Testpersonen beim Training unwohl wurde. Dies sollte zu einer Einschätzung dienen, ob die nicht funktionale Anforderung der Bedienbarkeit nach Kapitel 3.2 erfüllt ist oder ob noch Überarbeitungsbedarf besteht. Auch sollte ermittelt werden, wie hoch das Potenzial für das Auftreten von Cybersickness und Unwohlsein ist und ob dagegen noch weiter vorgegangen werden muss.

Die Befragung in Abbildung 5.2 hat ergeben, dass die meisten Testpersonen die Bedienung intuitiv fanden, was darauf schließen lässt, dass das Bedienkonzept, inklusive der kurzen Einweisung, seinen Zweck im wesentlichen erfüllt. Auch gab es unter den Testpersonen niemanden dem richtig unwohl wurde, jedoch kann sich dies von Person zu Person unterscheiden und bei einem längerem Training könnten die Ergebnisse möglicherweise etwas anders ausfallen. Zumindest kürzere Trainings scheinen aber bei den wenigsten Probleme hervorzurufen.

### Fragen 6-7

Über die nächsten beiden Fragen sollte festgestellt werden, wie den Testpersonen die

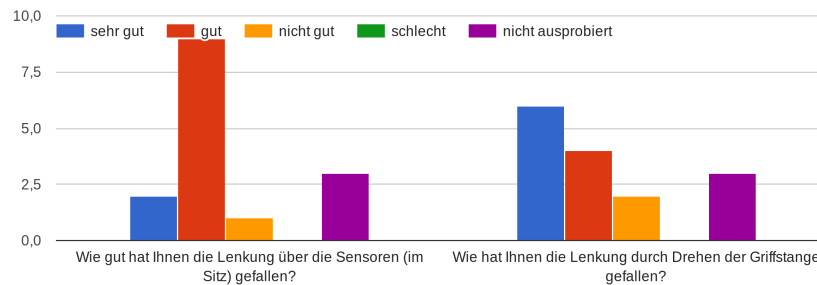


Abb. 5.3: Balkendiagramm über Fragen zur Lenkung

Lenkung gefallen hat. Dazu sollten sie für beide Lenkungsarten jeweils eine Bewertung abgeben. Da nicht jeder Tester alles ausprobieren musste, gab es hier die Möglichkeit *nicht ausprobiert* auszuwählen. Durch die Bewertungen soll ermittelt werden, welche der Lenkungsarten besser angenommen wurde und wo noch Überarbeitungsbedarf besteht.

Die Bewertungen in Abbildung 5.3 ergaben, dass die Lenkung über die Griffstange insgesamt etwas besser angenommen wurde als die Lenkung über die Sensoren. Die Sensorsteuerung wurde mit neun der zwölf Bewertungen mit *gut* bewertet. Daraus lässt sich schließen, dass es hierbei noch Verbesserungspotenzial gibt. In einer der Anmerkungen wurde darauf hingewiesen, dass die Sensorsteuerung besser wäre, wenn man sicherer im Sitz sein würde. Um diese Situation zu verbessern wäre es möglich die Lenkung feiner einzustellen, sodass weniger Gewichtsverteilung zum Lenken notwendig wäre. Zwei von zwölf Testern gefiel die Lenkung über die Griffstange nicht, was auch hier aufzeigt, dass es noch Bedarf für Überarbeitung gibt. Beispielsweise könnte die Lenkung direkter eingestellt werden.

### Fragen 8-10

Die letzten drei Fragen sollten ermitteln, inwieweit die drei Level ihren Trainingszweck erfüllen. Die Ergebnisse sind in Abbildung 5.4 zu sehen, hier wurden für jede Frage insgesamt nur 14 Antworten abgegeben. Das Sprintlevel wurde seinem leistungsorientierten Zweck am stärksten gerecht. Das Parcourslevel ist mit drei *eher nein* Stimmen, seinem Zweck am wenigsten gerecht geworden. Zwar gibt es auch je vier *ja* und *eher ja* Stimmen, es besteht aber durchaus noch Verbesserungsbedarf. Beispielsweise könnte der Schwierigkeitsgrad erhöht werden. Das Piratenlevel bekam im allgemeinen Zustimmung, hat mit

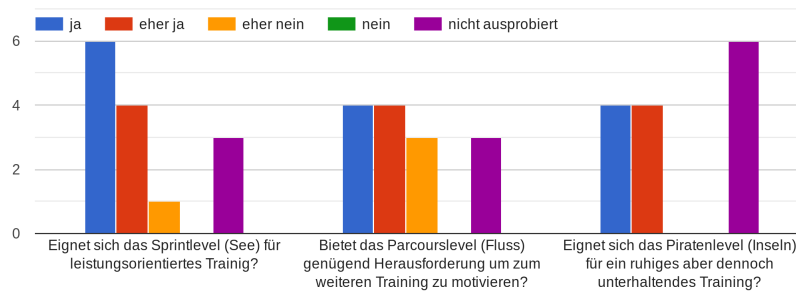


Abb. 5.4: Balkendiagramm über Fragen zur Eignung der Level

acht Testern aber auch nur eine geringe Testabdeckung. Verbesserungs- und Ausbaumöglichkeiten gibt es für alle drei der Level, so können diese optisch noch besser ausgearbeitet werden oder weitere funktionale Inhalte bekommen und gerade das Parcourslevel sollte einen stärkeren Fokus darauf setzen genug Herausforderung zu bieten.

### 5.3 Zusammenfassende Einschätzung

Insgesamt fallen die Rückmeldungen recht positiv aus, zeigen aber auch, an welchen Stellen noch Verbesserungspotential besteht. Unter anderem ist zu erkennen, dass einige Leute durchaus bereit wären mit einem VR-Exergame zu trainieren, insbesondere Leute, die wenig Sport treiben. Das allgemeine Steuerungskonzept scheint für diese Anwendung gut angenommen zu werden; die beiden Lenkungsarten könnten aber durch eine sensiblere und direktere Einstellung noch verbessert werden. Probleme mit Cybersickness scheint es bei kürzeren Trainings nicht zu geben. Die beiden nicht funktionalen Anforderungen der Bedienbarkeit und Performance sind also erfüllt. Die unterschiedlichen Level scheinen ihren Trainingszweck grundsätzlich zu erfüllen, jedoch ist das Parcourslevel nach der Einschätzung einiger Testpersonen aktuell nicht herausfordernd genug, um über eine längere Zeit hinweg zu motivieren. Eine Anpassung wäre z.B. durch höhere Schwierigkeitsgrade möglich. Die gegebenen Antworten und gefolgerten Einschätzungen zeigen eine Tendenz auf, die in einem weiteren Schritt mit einem statistisch belastbaren Versuchsaufbau, einer höheren Anzahl an Versuchsteilnehmern und einer repräsentativen Auswahl der Versuchsgruppe zu validieren wäre.

## 6 Zusammenfassung

Im Rahmen dieser Arbeit wurde ein VR-Exergame entwickelt, welches ein Rudergerät mithilfe eines VR-Systems in eine virtuelle Umgebung integriert. Das Rudergerät dient dabei zur Interaktion mit der virtuellen Realität. In einem Ruderboot, das virtuelle Gegenstück zum Rudergerät, befährt der Benutzer die virtuelle Welt. Für die Steuerung wurden zwei Lenkungsvarianten realisiert, eine durch Gewichtsverlagerung und eine durch Drehen der Griffstange. Mehrere Level bieten jeweils unterschiedliche Aufgaben und Herausforderungen für den Anwender. In einem abschließenden Experiment wurde das Exergame auf seine motivationssteigernde Wirkung und auf die Bedienbarkeit getestet.

In Kapitel 2 wurden zunächst einige Grundlagen vorgestellt und die Ruderanwendung in die Bereiche Exergames und Serious Games eingeordnet. Anschließend wurde auf einige vergleichbare Arbeiten eingegangen. Diese wurden zunächst vorgestellt und anschließend zu den Plänen dieser Arbeit abgegrenzt, mit dem Zweck einen besseren Überblick über das Thema der VR-Exergames zu erhalten und die Arbeit einordnen zu können. In Kapitel 3 wurden die funktionalen, sowie nicht funktionalen Anforderungen an die Anwendung formuliert. In den funktionalen Anforderungen wurden die drei Level und Trainingsmodi, die Arten mit der Anwendung zu interagieren und die Simulation des Ruderbootes spezifiziert. Die nicht funktionalen Anforderungen umfassen die Aspekte Bedienbarkeit, Performance und Konfigurierbarkeit. Kapitel 4 behandelt den Entwurf der VR-Ruderanwendung. Mithilfe eines Komponentendiagramms wurden die Aufgaben und Schnittstellen der unterschiedlichen Komponenten erläutert. In einem weiteren Schritt wurden diese auf das Unity Programmiermodell abgebildet. Abschließend findet sich eine detaillierte Beschreibung der Funktionsweisen der Komponenten. Kapitel 5 beschreibt den Aufbau und die Ergebnisse einer experimentellen Überprüfung mit 15 Testpersonen. Die Auswertung eines Fragebogens ergab, dass die Anwendung grundsätzlich gut angenommen wurde und gerade Leute, die weniger Sport treiben, durchaus bereit wären, mit einem solchen VR-Sportsystem zu trainieren. Allerdings zeigte sich auch, dass in den Bereichen Lenkung und Levelgestaltung noch Potential für weitere Entwicklung liegt.

### **Ausblick**

Während die experimentelle Überprüfung der VR-Ruderanwendung erste Eindrücke über einige Bereiche der Anwendung gesammelt hat, wäre der nächste Schritt eine statistisch belastbare Evaluierung durchzuführen. Dadurch könnten Resultate validiert und neue Erkenntnisse gewonnen werden, die wiederum dazu genutzt werden könnten, die Anwendung zu verbessern und ein klareres Bild von der Bereitschaft zum Sport in einer VR-Umgebung zu vermitteln. Auch eine Langzeitstudie wäre möglich, um Erkenntnis darüber zu erlangen, ob ein solches VR-Training langfristig motivierend sein kann.

Weiterhin könnten die bereits ermittelten Erkenntnisse für eine Überarbeitung genutzt werden. So wären die Lenkarten zu überarbeiten, z.B. indem weniger Bewegungsaufwand für den selben Lenkungsgrad erforderlich wäre. Dann müsste sich der Benutzer bei Verwendung der Sensorsteuerung weniger weit zur Seite lehnen und bei der Lenkung über den Controller die Griffstange weniger weit drehen. Dadurch würde der Bewegungsablauf des Trainings weniger beeinträchtigt.

Um die Trainingsmotivation zu erhöhen ließen sich die Level weiter ausbauen. So könnten Kulissen wie befahrbare Höhlen, detailreiche Schauplätze oder auch weitere Klimazonen und ein Tag-/Nachtzyklus die Welt immersiver wirken lassen. Zufallsgenerierte Umgebungen könnten darüber hinaus für mehr Abwechslung sorgen. Weiterhin ließe sich der Anspruch der Trainingsmodi erhöhen, indem diese um mehr Herausforderungen erweitert werden, etwa durch die Einführung unterschiedlicher Schwierigkeitsgrade, mehr unterschiedlicher Aufgaben oder Sammelobjekte, die einen vorübergehenden Vorteil bringen.

Als weiteres Mittel zur Motivationssteigerung würde sich ein Mehrspielermodus anbieten. Dabei wären viele neue Trainingsmodi möglich, wie Wettrennen, Wettkämpfe oder kooperative Modi. Die Möglichkeit mit einem Freund zu trainieren, würde sich möglicherweise positiv auf die Trainingsmotivation einiger Menschen auswirken und ein stärkeres Wettbewerbsgefühl erzeugen als etwa eine Bestenliste.

Um den Grad der Immersion weiter zu steigern, könnte Bodytracking eingesetzt werden um den Körper oder zumindest die Arme und Hände in die virtuelle Umgebung abzubilden. Dadurch ließen sich die eigenen Bewegungen durch einen virtuellen Avatar sichtbar machen.

# Literaturverzeichnis

- [1] BOLTON, John ; LAMBERT, Mike ; LIRETTE, Denis ; UNSWORTH, Ben: PaperDude: A Virtual Reality Cycling Exergame. In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2014 (CHI EA '14), S. 475–478. – URL <http://doi.acm.org/10.1145/2559206.2574827>. – ISBN 978-1-4503-2474-8
- [2] DAVIS, Simon ; NESBITT, Keith ; NALIVAICO, Eugene: A Systematic Review of Cybersickness. In: *Proceedings of the 2014 Conference on Interactive Entertainment*. New York, NY, USA : ACM, 2014 (IE2014), S. 8:1–8:9. – URL <http://doi.acm.org/10.1145/2677758.2677780>. – ISBN 978-1-4503-2790-9
- [3] DENK, Natalie: *Von der Immersion zur Illusion – Grundlagen und Definition von Virtual Reality*, Dissertation, 12 2016
- [4] DETERDING, Sebastian ; DIXON, Dan ; KHALED, Rilla ; NACKE, Lennart: From Game Design Elements to Gamefulness: Defining "Gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. New York, NY, USA : ACM, 2011 (MindTrek '11), S. 9–15. – URL <http://doi.acm.org/10.1145/2181037.2181040>. – ISBN 978-1-4503-0816-8
- [5] DETERDING, Sebastian ; KHALED, Rilla ; NACKE, Lennart E. ; DIXON, Dan: Gamification: Toward a definition. In: *CHI 2011 gamification workshop proceedings* Bd. 12 Vancouver BC, Canada (Veranst.), 2011
- [6] DÖRNER, R. ; BROLL, W. ; GRIMM, P. ; JUNG, B.: *Virtual und Augmented Reality (VR / AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer Berlin Heidelberg, 2014 (eXamen.press). – URL <https://books.google.de/books?id=aXckBAAQBAJ>. – ISBN 9783642289033

- [7] GEIGER, Christian ; POGSCHEBA, Patrick: *Unity-Tutorial I: Erste Schritte mit der Game Engine*. – URL <https://www.heise.de/-2615262>. – Zugriffsdatum: 2019-07-06
- [8] HAMARI, J. ; KOIVISTO, J. ; SARSA, H.: Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In: *2014 47th Hawaii International Conference on System Sciences(HICSS)* Bd. 00, URL [doi.ieeecomputersociety.org/10.1109/HICSS.2014.377](https://doi.ieeecomputersociety.org/10.1109/HICSS.2014.377), Jan. 2014, S. 3025–3034
- [9] KLESHNEV, Valery: *Rowing Biomechanics Newsletter*. – URL [http://www.biorow.com/RBN\\_en\\_2014\\_files/2014RowBiomNews03.pdf](http://www.biorow.com/RBN_en_2014_files/2014RowBiomNews03.pdf). – Zugriffsdatum: 2019-07-16
- [10] LAVIOLA, Joseph J.: A Discussion of Cybersickness in Virtual Environments. In: *SIGCHI Bull.* 32 (2000), Januar, Nr. 1, S. 47–56. – URL <http://doi.acm.org/10.1145/333329.333344>. – ISSN 0736-6906
- [11] LEIFIPHYSIK: *Auftriebskraft*. – URL <https://www.leifiphysik.de/mechanik/auftrieb-und-luftdruck/grundwissen/auftriebskraft>. – Zugriffsdatum: 2019-08-05
- [12] MCGREGOR, AH ; BULL, AMJ ; BYNG-MADDICK, R: A comparison of rowing technique at different stroke rates: a description of sequencing, force production and kinematics. In: *International journal of sports medicine* 25 (2004), Nr. 06, S. 465–470
- [13] NURKKALA, Veli-Matti ; KALERMO, Jonna ; JARVILEHTO, Timo: Development of Exergaming Simulator for Gym Training, Exercise Testing and Rehabilitation. In: *Journal of Communication and Computer* 11 (2014), S. 403–411
- [14] OH, Yoonsin ; YANG, Stephen: Defining exergames & exergaming. In: *Proceedings of Meaningful Play* (2010), S. 1–17
- [15] SCHMIDT, S. ; EHRENBRINK, P. ; WEISS, B. ; VOIGT-ANTONS, J. ; KOJIC, T. ; JOHNSTON, A. ; MÖLLER, S.: Impact of Virtual Environments on Motivation and Engagement During Exergames. In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2018, S. 1–6. – ISSN 2472-7814
- [16] SHERMAN, W.R. ; CRAIG, A.B.: *Understanding Virtual Reality: Interface, Application, and Design*. Elsevier Science, 2018 (The Morgan Kaufmann Series in Computer Graphics). – URL <https://books.google.de/books?id=D-OcBAAAQBAJ>. – ISBN 9780128010389

- [17] SINCLAIR, Jeff ; HINGSTON, Philip ; MASEK, Martin: Considerations for the Design of Exergames. In: *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*. New York, NY, USA : ACM, 2007 (GRAPHITE '07), S. 289–295. – URL <http://doi.acm.org/10.1145/1321261.1321313>. – ISBN 978-1-59593-912-8
- [18] SLATER, Mel ; WILBUR, Sylvia: A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments. In: *Presence: Teleoperators and Virtual Environments* 6 (1997), Nr. 6, S. 603–616
- [19] SPEKTRUM: *Strömungswiderstand*. – URL <https://www.spektrum.de/lexikon/physik/stroemungswiderstand/14101>. – Zugriffsdatum: 2019-08-05
- [20] SPEKTRUM: *Widerstandsbeiwert*. – URL <https://www.spektrum.de/lexikon/physik/widerstandsbeiwert/15587>. – Zugriffsdatum: 2019-08-05
- [21] TUVERI, Elena ; MACIS, Luca ; SORRENTINO, Fabio ; SPANO, Lucio D. ; SCATENI, Riccardo: Fitmersive Games: Fitness Gamification Through Immersive VR. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. New York, NY, USA : ACM, 2016 (AVI '16), S. 212–215. – URL <http://doi.acm.org/10.1145/2909132.2909287>. – ISBN 978-1-4503-4131-8
- [22] VALVESOFTWARE: *API Documentation*. – URL <https://github.com/ValveSoftware/openvr/wiki/API-Documentation>. – Zugriffsdatum: 2019-06-29
- [23] WITMER, Bob G. ; SINGER, Michael J.: Measuring Presence in Virtual Environments: A Presence Questionnaire. In: *Presence: Teleoperators and Virtual Environments* 7 (1998), Nr. 3, S. 225–240. – URL <https://doi.org/10.1162/105474698565686>
- [24] WRIGHT, W. ; BOGOST, I.: *Persuasive Games: The Expressive Power of Videogames*. MIT Press, 2007 (The MIT Press). – URL <https://books.google.de/books?id=vjbOnZw1wfUC>. – ISBN 9780262026147



## Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

*Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI*

## Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: \_\_\_\_\_

Vorname: \_\_\_\_\_

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

### **Entwicklung eines Exergames für Trainingseinheiten in Virtual Reality mit einem Rudergerät als Interaktionsmedium**

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

\_\_\_\_\_  
Ort                      Datum                      Unterschrift im Original