



**FH Salzburg**  
MultiMediaTechnology

# ***Endless Running - Systeme für Redirected Walking in Virtual Reality***

## **Bachelorarbeit**

Autor\*in: Daniel Zirngast

Betreuer\*in: Dr. Markus Tatzgern

Repository: [https://gitlab.mediacube.at/fhs45915/bachelorarbeit\\_endless\\_runner](https://gitlab.mediacube.at/fhs45915/bachelorarbeit_endless_runner)

Salzburg, Österreich, 03.07.2023

## Kurzfassung

Virtual Reality (VR) hat sich in den letzten Jahren enorm weiterentwickelt und gewinnt deshalb immer mehr an Popularität. Dadurch, dass VR nun immer leistbarer für die durchschnittlichen Anwender\*in geworden ist, können Menschen auf der ganzen Welt VR als Teil ihrer täglichen Technologie- und Unterhaltungserfahrung nutzen. Aus diesem Grund war es noch nie so wichtig wie heute dafür zu sorgen, den Anwender\*innen ein immersives und realistisches VR-Erlebnis zu bieten. Diese Arbeit untersucht die Anwendung von Redirected Walking als potenzielle Alternative zu traditionellen Fortbewegungsmethoden in VR. Dabei wird untersucht, ob Redirected Walking eine realistischere, immersivere und präsentere Erfahrung bietet, indem es den\*die Benutzer\*in durch subtile Techniken in die richtige Richtung lenkt. Es wird der Frage nachgegangen, welche Techniken und Algorithmen am besten geeignet sind, um Redirected Walking in VR-Spielen und Anwendungen erfolgreich umzusetzen und welche Auswirkungen es auf die Präsenz und das Spielerlebnis hat. Insbesondere wird der Einsatz von Steer-to-Orbit und Steer-to-Center untersucht und auf ihre Auswirkungen auf die Benutzer\*in eingegangen. Die Studie wird auch die Benutzererfahrung unter Verwendung von Redirected Walking mit unterschiedlich großen Spielflächen vergleichen, um die Wirksamkeit und Akzeptanz von Redirected Walking zu bewerten. Am Ende werden Empfehlungen für Entwickler\*innen und Designer\*innen gegeben, wie sie Redirected Walking erfolgreich in VR-Anwendungen implementieren können und ob es sinnvoller eingesetzt werden kann, um traditionelle Fortbewegungsmethoden von VR ablösen zu können.

## **Abstract**

Virtual Reality (VR) has evolved tremendously in recent years, leading to a surge in popularity. As VR becomes increasingly affordable to the average user, people around the world are incorporating it into their daily technology and entertainment experiences. Therefore, it has never been more important to provide users with an immersive and realistic VR experience. This thesis examines the application of Redirected Walking as a potential alternative to traditional locomotion methods in VR. It explores whether Redirected Walking provides a more realistic, immersive, and present experience by subtly guiding the user in the right direction. The study addresses the question of which techniques and algorithms are best suited to successfully implement Redirected Walking in VR games and applications and their effects on presence and the user experience. In particular, the use of Steer-to-Orbit and Steer-to-Center is examined, and their impact on the user is discussed. The study also compares the user experience using Redirected Walking with different sized play spaces to evaluate its effectiveness and acceptance. Finally, recommendations are given for developers and designers on how to successfully implement Redirected Walking in VR applications and whether it can be a viable replacement for traditional locomotion methods in VR.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>2</b>
2.1	Fortbewegungstechniken in VR . . . . .	4
2.2	Bewegen in Virtual Reality . . . . .	5
2.2.1	Teleportation . . . . .	5
2.2.2	Bewegung mittels Joystick . . . . .	7
2.2.3	Walking in Place . . . . .	7
2.2.4	Omni-Directional Treadmills . . . . .	8
<b>3</b>	<b>Redirected Walking als Lösungsansatz</b>	<b>10</b>
3.1	Scaling . . . . .	10
3.2	Translation Gain . . . . .	11
3.3	Rotation Gain . . . . .	11
3.4	Curvature Gain . . . . .	12
3.5	Bending Gain . . . . .	12
<b>4</b>	<b>Endless Runner in Virtual Reality mit Redirected Walking</b>	<b>13</b>
<b>5</b>	<b>Entwicklung eines Endless Runners in VR</b>	<b>15</b>
5.1	Hauptkomponenten . . . . .	15
5.2	Game Manager . . . . .	16
5.3	Unendliche Map Generation . . . . .	16
5.4	Redirecting Algorithmen . . . . .	17
5.4.1	Redirection Manager . . . . .	17
5.4.2	Steer-To-Center . . . . .	20
5.4.3	Steer-To-Orbit . . . . .	22
5.5	Resets kurz vor Verlassen des Spielbereichs . . . . .	23
<b>6</b>	<b>Evaluierung des Programms mittels Studie</b>	<b>24</b>
<b>7</b>	<b>Zusammenfassung</b>	<b>29</b>
	<b>Appendices</b>	<b>33</b>

<b>A</b>	<b>git-Repository</b>	<b>33</b>
<b>B</b>	<b>Vorlagen für Studienmaterial</b>	<b>33</b>
B.1	Fragebogen zur Studie . . . . .	33
B.2	Datenerhebungseinwilligung . . . . .	36

## Abbildungsverzeichnis

1	Zusammenspiel von Immersion und Präsenz (Quelle: <a href="https://relilab.org/vr/">https://relilab.org/vr/</a> ) . . .	3
2	Verschiedene VR Fortbewegungsarten und deren Interaktionsaspekte (Boletsis und Cedergren 2019) . . . . .	5
3	Teleportation im virtuellen Raum (Quelle: <a href="https://www.engadget.com/2016-10-07-why-teleportation-makes-sense-in-virtual-reality.html">https://www.engadget.com/2016-10-07-why-teleportation-makes-sense-in-virtual-reality.html</a> ) . . . . .	6
4	Walk in Place Bewegung (Lee, Ahn und Hwang 2018) . . . . .	8
5	Fortbewegen mittels Omni-Directional Treadmills (Quelle: <a href="https://www.uploadvr.com/kat-walk-c-2-kickstarter/">https://www.uploadvr.com/kat-walk-c-2-kickstarter/</a> ) . . . . .	9
6	The vier Arten für perspektivische Manipulation: (a) Translation Gain, (b) Rotation Gain, (c) Curvature Gain, and (d) Bending Gain. Violette und blaue Linien zeigen die realen bzw. virtuellen Transformationen an (Nilsson u. a. 2018) . . .	10
7	(a) Der*Die Benutzer*in geht in VR eine Kruve, welche einen anderen Radius hat als die Kurve der realen Welt auf welchem er umgeleitet wird, und (b) die Sicht des*der Benutzers*in in VR mit einem gekrümmten virtuellen Pfad (Langbehn u. a. 2017) . . . . .	13
8	Endless Runner Games Subway Surfer, Temple Run und Sonic Dash (Quelle: <a href="https://machinations.io/docs/infinite-runner-game-loop/">https://machinations.io/docs/infinite-runner-game-loop/</a> ) . . . . .	14
9	Platform Prefabs . . . . .	16
10	Vererbungshierarchie der Applikation . . . . .	18
11	Unity Game Hierarchie . . . . .	23
12	Physisch verfügbare Fläche in relation zum virtuellen Raum . . . . .	24
13	Boxplotauswertung der Fragen. Siehe Fragebogen zur Studie für Fragenzuordnung zu den einzelnen Nummern . . . . .	26
14	Weg (in Gelb) der in echt bei STC in der physisch verfügbaren Fläche zurückgelegt wurde . . . . .	28
15	Weg (in Gelb) der bei STO in echt in der physisch verfügbaren Fläche zurückgelegt wurde . . . . .	28
16	Unterschiedliche Wege, welche die Propanden*innen zurückgelegt haben . . .	29

## Listings

## Tabellenverzeichnis

# **Abkürzungsverzeichnis**

**VR** Virtual Reality

**WiP** Walking in Place

**ODTs** Omni-Directional Treadmills

**STO** Steer-To-Orbit

**STC** Steer-To-Center

# 1 Einleitung

Virtual Reality (VR) hat in den letzten Jahren ein enormes Wachstum erlebt und ist zu einem wichtigen Bereich der Technologie geworden, da es immer erschwinglicher und zugänglicher für eine breitere Bevölkerung geworden ist. Dank der Fortschritte in der Technologie und der immer größeren Verfügbarkeit von VR-Headsets können immer mehr Menschen die virtuelle Welt erleben. Unternehmen haben auch begonnen, VR für verschiedene Anwendungen zu nutzen, wie z.B. für Training, Simulationen, Architektur-Visualisierungen, Unterhaltung und mehr (Hamad und Jia 2022, 1). Auch die COVID-19-Pandemie hat dazu beigetragen, dass sich VR als wichtige Alternative zu persönlichen Interaktionen etabliert hat (Singh u. a. 2020, 1). VR bietet eine einzigartige Möglichkeit in eine andere Welt einzutauchen und eine unvergleichliche Benutzererfahrung zu erleben. Daher ist es kein Wunder, dass immer mehr Menschen auf der ganzen Welt von VR fasziniert sind und es als Teil ihrer täglichen Technologie- und Unterhaltungserfahrung nutzen (Singh u. a. 2020, 2). Obwohl VR in den letzten Jahren ein enormes Wachstum erlebt hat, ist die Technologie noch lange nicht perfekt und es gibt viele Herausforderungen zu bewältigen, um ein wirklich immersives VR-Erlebnis zu schaffen (Hamad und Jia 2022, 2). Eine dieser Herausforderungen ist die Fortbewegung des\*der Benutzers\*in in der virtuellen Welt. Bisher gibt es keine perfekte Fortbewegungsmethode und jede Implementierung hat ihre Vor- und Nachteile (Boletsis und Cedergren 2019, 12). Das kann dazu führen, dass Benutzer\*innen Schwierigkeiten haben, sich in der virtuellen Welt zu bewegen und ein wirklich immersives Erlebnis zu haben. Daher ist es wichtig, neue Fortbewegungstechniken und -systeme zu entwickeln und zu erforschen, die die Nachteile der bisherigen Systeme minimieren und ein realistischeres VR-Erlebnis schaffen können (Boletsis und Cedergren 2019, 12). Ein weiteres Problem ist die begrenzte Bewegungsfreiheit des\*der Benutzers\*in, da er\*sie durch das Headset und die Verkabelung eingeschränkt wird. Dies kann dazu führen, dass sich der\*die Spieler\*in unwohl fühlt oder sogar Stolperfallen im Raum hat, wenn er\*sie sich zu sehr in der virtuellen Welt bewegt. Abgesehen von den Herausforderungen bei der Fortbewegung in VR gibt es noch andere Probleme, die die breitere Akzeptanz von VR-Anwendungen behindern. Eines der Hauptprobleme ist die Hardware, da VR-Geräte immer noch teuer sind und eine leistungsstarke Grafikkarte benötigen, um eine reibungslose VR-Erfahrung zu ermöglichen. Auch wenn die Preise in den letzten Jahren gesunken sind, sind sie immer noch für viele Verbraucher unerschwinglich. In dieser Bachelorarbeit soll untersucht werden, welche Probleme bei der Fortbewegung in VR bestehen und welche Techniken und Methoden es gibt, um diese Herausforderungen zu überwinden. Dabei soll auch auf die Auswirkungen der Fortbewegung mittels Redirected Walking als neue Fortbewegungsmöglichkeit in VR und auf die Benutzererfahrung und das Wohlbefinden des Spielers eingegangen werden.

Ein Schwerpunkt der Arbeit wird auf der Untersuchung der Fortbewegungsmethode „Redirected Walking“ liegen, welche das traditionelle Bewegen durch Teleportation oder Joysticks ersetzt. Dieses System ist eine vielversprechende Technik, welches den\*die Benutzer\*in subtil abgelenkt, sodass er\*sie unbewusst von der tatsächlichen Bewegung abgelenkt wird, während er\*sie sich in der virtuellen Welt bewegt (Azmandian u. a. 2016, 2). Außerdem kann es dazu beitragen, die ein realistischeres VR-Erlebnis durch effiziente Navigation zu schaffen (Azmandian u. a. 2016, 1). Das Ziel des praktischen Teils wird es sein, einen sogenannten „Endless Runner“



in VR zu entwerfen, welcher sich auf das unendliche Laufen des Spielcharakters durch eine sich ständig verändernde Spielwelt konzentriert. Das Ziel solcher Spiele ist es, so weit wie möglich zu kommen und dabei Hindernissen auszuweichen oder diese zu überwinden, um einen höheren Punktestand zu erreichen. Der Spielcharakter bewegt sich automatisch fort und der\*die Spieler\*in hat keine Kontrolle über die Geschwindigkeit. Stattdessen muss er\*sie schnell reagieren, um Hindernissen wie Abgründen, Felsen oder anderen Gefahren auszuweichen. Da der\*die Benutzer\*in bei fehlender Kontrolle über die Bewegung des Charakters schnell an „Motion Sickness“ erkranken kann und das Spiel beenden muss, wird das Endless-Running-Prinzip in VR nicht vollständig umsetzbar sein. In dieser adaptierten Variante soll der\*die Spieler\*in jedoch durch „Redirected Walking“ selbstständig einem quasi endlosen Pfad folgen können, Punkte sammeln und durch ansteigender Schwierigkeit versuchen, solange am Leben zu bleiben wie möglich. Mittles der Applikation soll analysiert werden, ob das Redirected Walking System für gewisse Szenarien geeignet sein könnte, welche Auswirkungen es auf die Immersion und die Benutzererfahrung hat und ob es bereits so weit vorangeschritten ist, um die herkömmlichen Fortbewegungsmöglichkeiten vollständig zu ersetzen.

Die Arbeit wird sowohl eine theoretische Analyse als auch eine praktische Umsetzung des oben genannten Systems beinhalten. Im Rahmen der theoretischen Analyse werden relevante Literaturquellen zum Thema VR und Fortbewegung ausgewertet. Im praktischen Teil wird eine VR-Anwendung entwickelt, in der redirecten mittels verschiedenen Algorithmen implementiert wird. Die Benutzererfahrung und die Auswirkungen auf das Wohlbefinden der Benutzer\*innen werden mittels Umfragen und Interviews erfasst und ausgewertet.

Diese Arbeit trägt dazu bei, das Verständnis von Fortbewegung in VR zu vertiefen und bietet einen Beitrag zur Entwicklung von VR-Anwendungen, indem die Sinnhaftigkeit vom Einsatz von Redirected Walking erforscht wird.

## 2 Related Work

Virtual Reality, oft durch VR abgekürzt, beschreibt eine computergenerierte Umgebung, die es dem\*der Benutzer\*in ermöglicht, in eine virtuelle Welt einzutauchen und mit ihr zu interagieren (Hamad und Jia 2022, 1). Der\*Die Benutzer\*in trägt dabei oft eine spezielle VR-Brille (Head-Mounted Display), die ihm\*ihr ein dreidimensionales Erlebnis ermöglicht und ihm\*ihr das Gefühl gibt, Teil der virtuellen Welt zu sein (Hamad und Jia 2022, 1). Durch die Verwendung von VR-Controllern oder anderen Eingabegeräten kann der\*die Anwender\*in in der virtuellen Umgebung navigieren, Objekte manipulieren oder mit virtuellen Charakteren interagieren. Präsenz und Immersion sind zwei der zentralen Ziele bei der Entwicklung von VR-Technologie und virtuellen Anwendungen (Bowman und McMahan 2007, 4). Das Zusammenspiel von Präsenz und Immersion ist in Abb. 1 dargestellt. Ein realistisches Erscheinungsbild von Objekten und Bewegungen in VR kann dazu beitragen, dass der\*die Spieler\*in die virtuelle Welt als authentisch und glaubwürdig empfindet. Dies wird erreicht durch Technologien wie 3D-Modellierung, Texturierung und Animation, welche eine detailreiche und naturgetreue Darstellung von Objekten ermöglichen (Kuang und Bai 2018, 1).

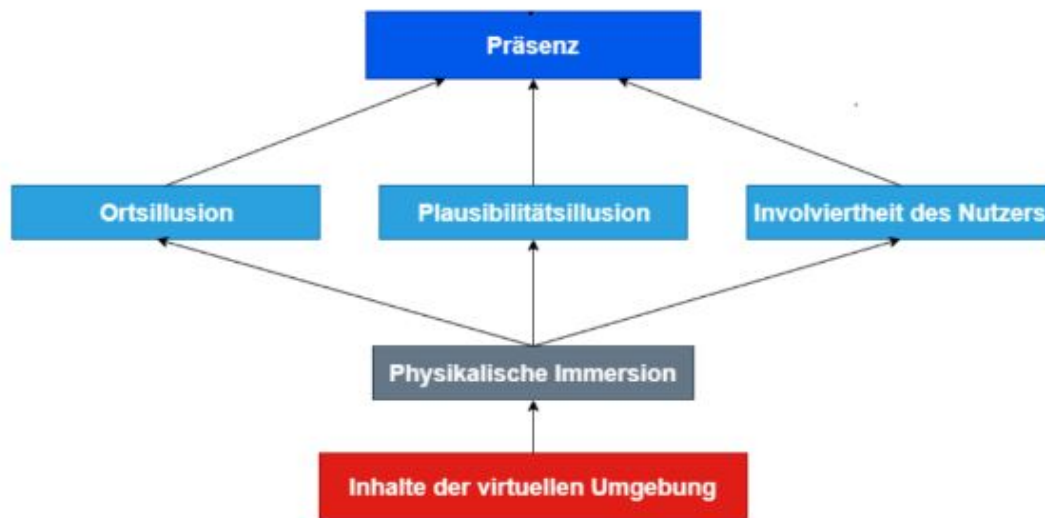


Abbildung 1: Zusammenspiel von Immersion und Präsenz (Quelle: <https://relilab.org/vr/>)

Die Möglichkeit zur Fortbewegung in VR ist ein wesentlicher Faktor, der das realistische Gefühl des Nutzers in der virtuellen Welt beeinflusst. Dabei spielt die Bewegungsfreiheit eine wichtige Rolle für die Präsenz in VR (Cardoso und Perrotta 2019, 1). Eine realistische Bewegungsfreiheit in VR bedeutet, dass die Bewegungen des virtuellen Avatars den natürlichen menschlichen Bewegungen ähnlich sind und dass sich der\*die Spieler\*in in einem größeren Bereich bewegen kann, ohne aus dem VR-Setup auszusteigen. Wenn der Nutzer sich nicht frei in der virtuellen Umgebung bewegen kann, kann dies zu einem Gefühl der Begrenzung und Unwirklichkeit führen (Cardoso und Perrotta 2019, 2). Eine zu hohe Bewegungsfreiheit kann jedoch auch problematisch sein. Wenn sich der\*die Benutzer\*in zu schnell bewegt oder sich in einer unnatürlichen Art und Weise bewegt, kann dies dazu führen, dass sich der\*die Anwender\*in desorientiert oder unwohl fühlt. Das kann sogar zu Übelkeit und Motion Sickness führen (Kim u. a. 2018, 3). Um eine möglichst realistische Bewegungsfreiheit zu erreichen, müssen verschiedene Aspekte berücksichtigt werden. Zum einen ist es wichtig, dass der virtuelle Avatar die Bewegungen des Spielers natürlich und flüssig ausführt. Das bedeutet, dass die Bewegungen des Avatars in Echtzeit berechnet werden müssen, um Verzögerungen oder Ruckeln zu vermeiden. Zum anderen muss das VR-Setup ausreichend Platz und Freiheit für den\*die Benutzer\*in bieten, um sich in der virtuellen Umgebung zu bewegen (Saker und Frith 2020, 9). Hierzu gehört auch, dass der\*die Anwender\*in Hindernisse in der realen Welt vermeiden kann und genügend Platz hat, um sich sicher und bequem zu bewegen. Eine natürliche, intuitive und immersive Fortbewegung in VR trägt dazu bei, dass der Nutzer das Gefühl hat, tatsächlich in der virtuellen Welt präsent zu sein. Es gibt viele Möglichkeiten zur Fortbewegung in VR, von der klassischen Laufbewegung bis hin zu neuartigen Technologien wie Redirected Walking. Die Wahl der Fortbewegungsmethode kann auch stark vom Anwendungsfall und der Art des VR-Erlebnisses abhängen. Insgesamt ist die Qualität der Fortbewegung in VR ein wesentlicher Faktor, um ei-

ne authentische und immersiven Erfahrung zu ermöglichen. Die Präsenz hingegen bezieht sich darauf, wie stark der\*die Spieler\*in in die virtuelle Welt eingebunden ist und wie sehr man sich als Teil dieser Welt fühlt (Bowman und McMahan 2007, 3). Eine hohe Immersion kann dazu führen, dass der\*die Benutzer\*in vergisst, dass er\*sie sich in einer virtuellen Umgebung befindet und eine realitätsnahe Erfahrung genießt. Dies kann durch verschiedene Technologien und Design-Prinzipien erreicht werden, wie beispielsweise durch eine intuitive Steuerung und Interaktion mit der virtuellen Welt, eine realistische Audio-Kulisse und die Nutzung von haptischem Feedback (Bowman und McMahan 2007, 1). Zusammen können Realismus und Immersion dazu beitragen, dass der\*die Spieler\*in eine bessere Erfahrung in VR erlebt und ein höheres Maß an Engagement und Unterhaltung erzielt wird. Darüber hinaus ergeben sich auch praktische Anwendungen in verschiedenen Branchen wie beispielsweise im Bereich der medizinischen Simulation, wo VR dazu beitragen kann, realistische Trainingsumgebungen zu schaffen und medizinische Fachkräfte besser auf ihre Arbeit vorzubereiten.

## 2.1 Fortbewegungstechniken in VR

In der Forschung zum Thema Fortbewegung in Virtual Reality (VR) wurden bereits verschiedene Ansätze untersucht. Das Redirected Walking Toolkit ermöglicht es Entwickler\*innen verschiedene Redirected Walking Algorithmen in Simulatoren und realen Anwendungen zu testen (Azmandian u. a. 2016, 1). Außerdem wird eine umfassende Lösung vorgestellt, die es Entwickler\*innen ermöglicht, Redirected Walking in ihre VR-Anwendungen zu integrieren. Es wird darauf eingegangen, dass das Konzept des „Redirected Walking“ ein vielversprechender Ansatz ist, um das Bewegungsproblem in VR zu lösen. Allerdings ist die Implementierung dieser Technologie noch sehr aufwendig und erfordert viel Fachwissen und Erfahrung in der Programmierung (Azmandian u. a. 2016, 6). Die Autoren der wissenschaftlichen Arbeit bieten eine detaillierte Beschreibung der zugrunde liegenden Technologien und Algorithmen, sowie eine Anleitung zur Verwendung des Redirected Walking Toolkit in Unity. Die Arbeit ist von besonderem Interesse, da sie Entwicklern eine umfassende Plattform bietet, um die Leistung von Redirected Walking in verschiedenen Umgebungen und Anwendungen zu untersuchen. Die Ergebnisse aus dieser Forschung können als wichtige Grundlage für die Implementierung von Redirected Walking in anderen VR-Anwendungen dienen. Die wissenschaftliche Arbeit von Costas Boletis et al. (2019) vergleicht verschiedene Fortbewegungstechniken in VR. Dabei wurde in einer Studie festgestellt, dass die Wahl der Fortbewegungsmethode einen erheblichen Einfluss auf das VR-Erlebnis und die Nutzer\*innenzufriedenheit hat (Boletis und Cedergren 2019, 12). Eine weitere Studie untersuchte drei verschiedene Methoden: Teleportation, Joystick und Redirected Walking. Unter den getesteten Techniken zeigte Redirected Walking die größte Potenzial zur Erzeugung von räumlicher Wahrnehmung in der virtuellen Welt (Langbehn, Lubos und Steinicke 2018, 7). Redirected Walking nutzt subtile Abweichungen von der realen Bewegung, um den\*die Benutzer\*in im Glauben zu lassen, dass er\*sie sich geradeaus bewegt, während er\*sie tatsächlich in Kurven oder Kreisen um die virtuelle Umgebung geführt wird (Azmandian u. a. 2016, 2). In einer weiteren Arbeit wird die Wirksamkeit von Redirected Walking bei der Navigation in virtuellen Umgebungen und wie effektiv es verhindert werden kann, dass der\*die Benutzer\*in den physisch verfügbaren Platz verlässt (Hodgson, Bachmann

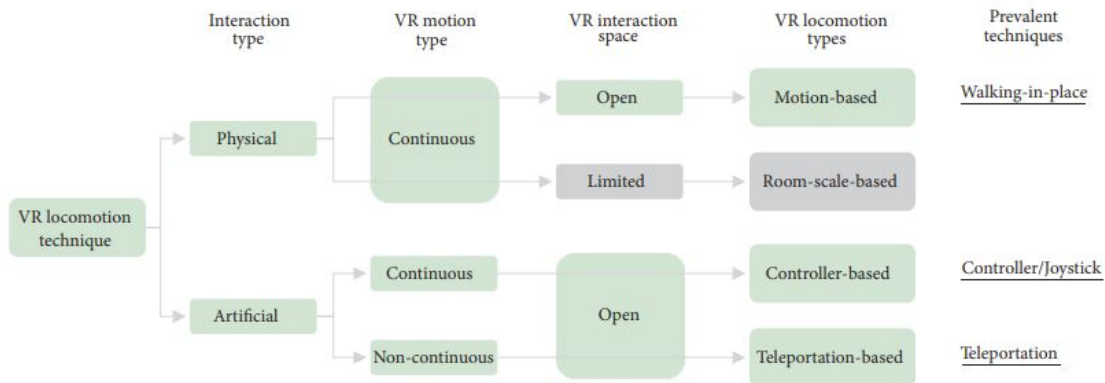


Abbildung 2: Verschiedene VR Fortbewegungsarten und deren Interaktionsaspekte (Boletsis und Cedergren 2019)

und Waller 2008, 1). Die Ergebnisse zeigen, dass Redirected Walking eine vielversprechende Methode ist, um virtuelle Umgebungen zu erkunden und den\*die Spieler\*in im physisch verfügbaren Platz zu halten (Hodgson, Bachmann und Waller 2008, 19). In der Studie wurden die Teilnehmer\*innen in virtuellen Umgebungen navigiert und die räumliche Kognition und Wahrnehmung wurden mittels verschiedener Tests bewertet, sowie die Anzahl der Kollisionen mit dem Guardian (Hodgson, Bachmann und Waller 2008, 17).

## 2.2 Bewegen in Virtual Reality

In den letzten Jahren hat die VR-Technologie eine schnelle Entwicklung erfahren, was zu einem wachsenden Interesse an der Fortbewegung in virtuellen Umgebungen geführt hat. In der wissenschaftlichen Literatur wurden zahlreiche Ansätze untersucht, um die Bewegung des Nutzers in VR zu ermöglichen und gleichzeitig unerwünschte Symptome wie Bewegungskrankheit oder Desorientierung zu vermeiden (Abb. 2). Im Folgenden werden einige der wichtigsten Fortbewegungsarten zu diesem Thema vorgestellt und es wird auf ihre Vor- und Nachteile eingegangen.

### 2.2.1 Teleportation

Teleportation ist eine der am häufigsten verwendeten Methoden zur Fortbewegung in VR. Es ermöglicht dem\*der Benutzer\*in, schnell und einfach von einem Ort zum anderen zu gelangen, ohne sich physisch zu bewegen. Der\*Die Spieler\*in wählt eine Position aus zu der er\*sie teleportieren möchte, z.B. durch Zeigen mit einem Controller oder durch Drücken eines Knopfes auf einem Controller wie abgebildet in Abbildung 3.

Dann wird das Bild in der VR-Brille verschwommen, während der\*die Benutzer\*in in die neue Position teleportiert wird. Sobald der\*die Anwender\*in in der neuen Position angekommen ist, wird das Bild wieder scharf (Boletsis und Cedergren 2019, 3). Nachdem diese Teleportation

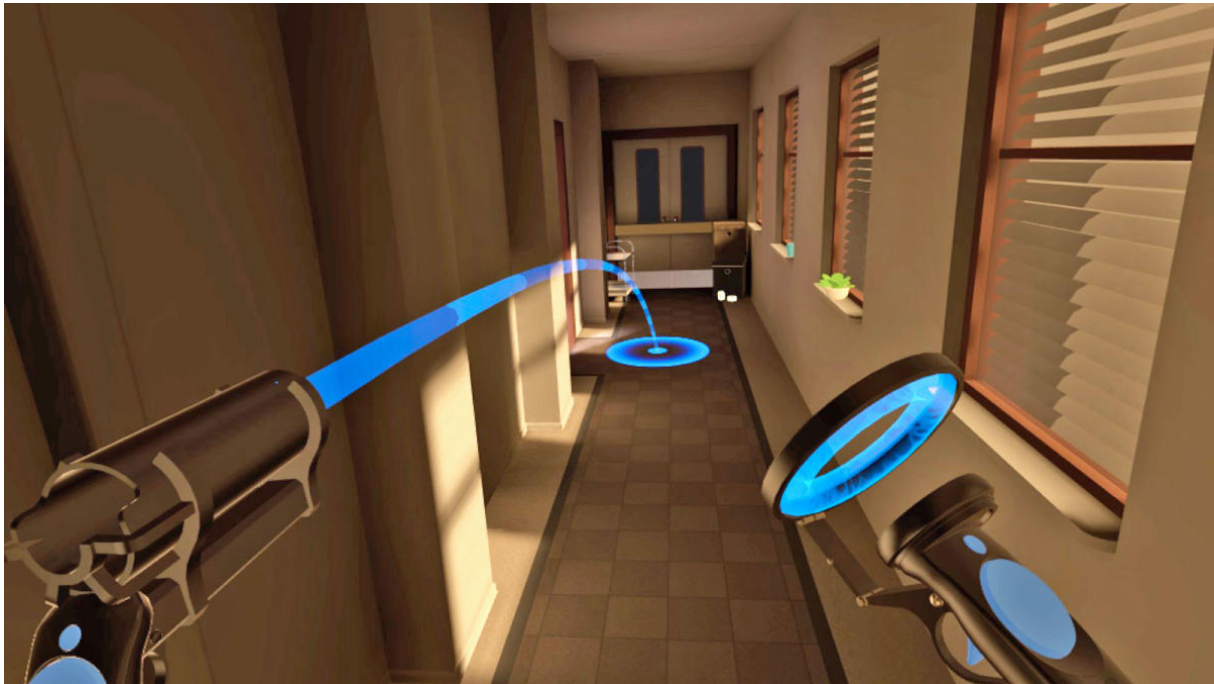


Abbildung 3: Teleportation im virtuellen Raum (Quelle: <https://www.engadget.com/2016-10-07-why-teleportation-makes-sense-in-virtual-reality.html>)

unbewusst passiert und der\*die Spieler\*in nicht physisch durch den Raum bewegt wird, ist der\*die Anwender\*in nicht von „Motion Sickness“ betroffen. Ein bekanntes Beispiel für die Verwendung von Teleportation in VR ist das Spiel „Budget Cuts“ von Neat Corporation (2018). Jedoch hat diese Technik einige Nachteile und Limitierungen. Die begrenzte Reichweite des Teleportations-Pointers kann dazu führen, dass der\*die Benutzer\*in nicht alle Bereiche der virtuellen Welt erreichen kann und man sich mehrmals infolge teleportieren muss um zu seinem Ziel zu gelangen (Langbehn, Lubos und Steinicke 2018, 7). Zudem kann es schwierig sein, sich in der virtuellen Welt zurechtzufinden, da es nicht einfach ist, eine räumliche Orientierung zu entwickeln. Ein weiterer Nachteil der Teleportation ist, dass sie das natürliche Bewegungsgefühl des Benutzers unterbricht und zu einer Art „Warpanimation“ führt wodurch man die räumliche Orientierung verlieren kann. Das plötzliche Erscheinen an einem anderen Ort kann zudem dazu führen, dass der\*die Anwender\*in Schwierigkeiten hat, seine\*ihre Position und Ausrichtung in der virtuellen Welt beizubehalten (Langbehn, Lubos und Steinicke 2018, 2). Ein weiteres Problem ist, dass die Teleportation das Gefühl von Präsenz beeinträchtigen kann. Da man sich nicht auf natürliche Weise in der virtuellen Welt bewegt, kann es schwieriger sein, das Gefühl zu entwickeln, tatsächlich physisch präsent zu sein (Langbehn, Lubos und Steinicke 2018, 2). Teleportation wird als Ergänzung benutzt für den Fall, dass dem\*der Anwender\*in in der realen Welt der Platz ausgeht. Dadurch ist der\*die Spieler\*in auf einen bestimmten physischen Bereich beschränkt, da er\*sie nicht über die Möglichkeit verfügt, frei in der virtuellen Welt herumzulaufen. Das kann insbesondere in „Open-World-Spielen“, wo eine große Welt erkundet werden soll, zu einem Problem werden. Insgesamt ist die Teleportation zwar eine nützliche Methode

der Fortbewegung in VR, hat jedoch einige Nachteile und Limitierungen, die die Immersion beeinträchtigen und die Bewegungsfreiheit des Benutzers einschränken können.

### 2.2.2 Bewegung mittels Joystick

Joystick-basierte Bewegung ist eine weitere gängige Technik für Fortbewegung in VR. Benutzer\*innen verwenden einen Joystick, um sich innerhalb der virtuellen Umgebung zu bewegen, ähnlich wie bei der Navigation in einem Videospiel oder einer 3D-Anwendung bei denen man mit den Joystick Inputs sich nach vorne, nach hinten, nach links und rechts bewegen kann (Langbehn, Lubos und Steinicke 2018, 3). Diese Technik ermöglicht Spielern eine schnelle und kontinuierliche Bewegung innerhalb der Umgebung, was insbesondere in Anwendungen wie VR-Spielen oder Erkundungen von großen virtuellen Umgebungen nützlich sein kann. Diese Methode bietet im Vergleich zur Teleportation ein höheres Maß an Kontrolle und ermöglicht es dem\*der Anwender\*in, sich kontinuierlich durch die virtuelle Umgebung zu bewegen. Eine der Hauptlimitierungen der Joystick-Steuerung ist die sogenannte „Motion sickness“ (Langbehn, Lubos und Steinicke 2018, 1). Wenn der\*die Benutzer\*in eine Bewegung im Spiel ausführt, die nicht der Bewegung seines\*ihres Körpers entspricht, kann dies zu Übelkeit und Desorientierung führen. Da der\*die Spieler\*in in VR-Spielen typischerweise sitzt oder steht und nicht wirklich läuft, kann die Verwendung von Joysticks dazu führen, dass die Wahrnehmung des Spielenden von seiner\*ihrer tatsächlichen Position und Bewegung in der realen Welt abweicht (Langbehn, Lubos und Steinicke 2018, 1). Ein weiteres Problem bei der Verwendung von Joysticks zur Fortbewegung in VR ist die begrenzte Bewegungsfreiheit des Benutzers. Nachdem der\*die Spieler\*in von der Bewegung des Controllers abhängig ist, kann er\*sie nicht frei in alle Richtungen gehen oder sich in alle Richtungen drehen. Dies kann zu Einschränkungen bei der Erkundung der virtuellen Umgebung führen und die Präsenz in das Spiel beeinträchtigen (Langbehn, Lubos und Steinicke 2018, 1).

### 2.2.3 Walking in Place

Walking in Place (WiP) ist ein Ansatz, bei dem der\*die Benutzer\*in am Stand geht, während er\*sie sich in der virtuellen Umgebung bewegt. Es nutzt die natürliche Ganganimation des Spielenden und bietet ein hohes Maß an Immersion (siehe Abbildung 4).

Forschungen haben gezeigt, dass diese Methode dazu beitragen kann, die Bewegungskrankheit zu reduzieren (Usoh u. a. 1999, 4). Ein Beispiel für die Verwendung von Walking in Place in VR ist das Spiel „Sprint Vector“ von Survios (2018). Einer der Hauptnachteile von WiP ist die räumliche Begrenzung. Der\*Die Benutzer\*in kann sich nur so weit bewegen, wie der physische Raum es zulässt und muss möglicherweise oft umkehren oder Teleportation als zusätzliche Fortbewegungsmethode verwenden. Außerdem kann es schwierig sein, die Bewegungen des Benutzers genau zu verfolgen, insbesondere wenn der\*die Spieler\*in in einem begrenzten Raum eingeschränkt ist oder schnelle Bewegungen ausführt (Lee, Ahn und Hwang 2018, 1). Jedoch kann bei WiP auch wieder motion sickness auftreten da der\*die Benutzer\*in, der in der Realität auf der Stelle läuft, zwar in der virtuellen Welt vorwärts geht, aber keine physische Fortbewegung



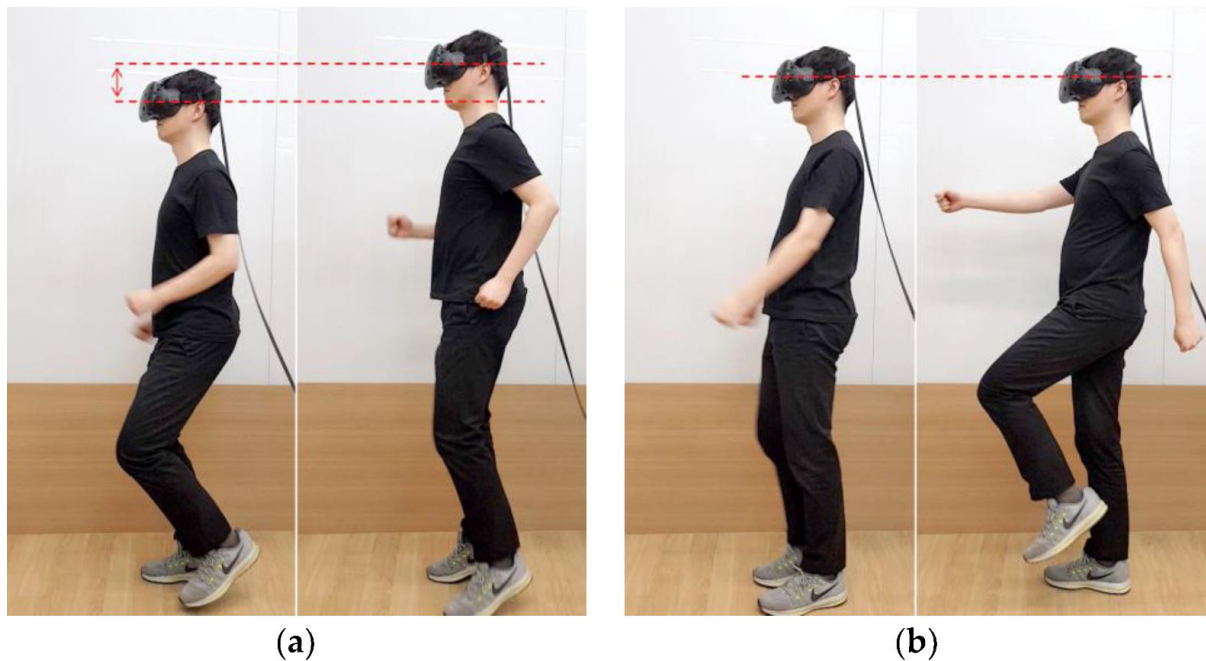


Abbildung 4: Walk in Place Bewegung (Lee, Ahn und Hwang 2018)

empfindet. Dies kann dazu führen, dass der Spielende das Gefühl hat, in der virtuellen Welt zu schweben, was das Gefühl der Präsenz verringert. Schließlich kann Walking in Place auch körperlich anstrengend sein, insbesondere wenn der\*die Anwender\*in längere Zeit in der virtuellen Welt unterwegs ist. Das ständige Gehen auf der Stelle kann Muskelermüdung verursachen und zu körperlichen Beschwerden führen.

#### 2.2.4 Omni-Directional Treadmills

Omni-Directional Treadmills (ODTs) sind Geräte, die es ermöglichen, sich in der virtuellen Umgebung zu bewegen, indem man auf einem sich bewegenden Laufband steht (Darken, Cockayne und Carmein 1997). Diese Geräte sind oft sehr teuer und benötigen viel Platz, sind aber in der Lage, eine sehr realistische Bewegungserfahrung zu ermöglichen (Darken, Cockayne und Carmein 1997) (Abbildung 5).

Einer der Hauptvorteile besteht darin, dass der\*die Nutzer\*in auf natürliche Weise läuft und somit ein extrem immersiveres Erlebnis hat. Im Vergleich zu Teleportation oder Joystick-Steuerung ermöglicht das Laufen auf einem ODT eine natürlichere Bewegung, die dem Laufen in der realen Welt näher kommt (Darken, Cockayne und Carmein 1997, 2). Ein weiterer Vorteil von ODTs ist, dass sie eine größere Bewegungsfreiheit bieten. Während bei anderen Methoden oft nur begrenzte physische Bereiche zur Verfügung stehen, kann ein ODT eine viel größere Bewegungsfläche bereitstellen. Dadurch kann der\*die Spieler\*in sich in einer größeren virtuellen Welt bewegen, ohne dabei reale Hindernisse zu berühren oder gegen Wände zu laufen. Zusätzlich kann ein ODT auch die Navigation in der virtuellen Umgebung erleichtern. Durch



Abbildung 5: Fortbewegen mittels Omni-Directional Treadmills (Quelle: <https://www.uploadvr.com/kat-walk-c-2-kickstarter/>)

die Möglichkeit, sich frei in alle Richtungen zu bewegen, können die Nutzer sich intuitiver in der Umgebung orientieren und schnell auf Veränderungen reagieren (Darken, Cockayne und Carmein 1997, 1). Dies erhöht auch die Nutzerfreundlichkeit der VR-Anwendung und verringert mögliche Bewegungsübelkeit. Ein weiterer Vorteil ist es, dass ODTs die körperliche Aktivität des Nutzers erhöhen. Im Gegensatz zu anderen Fortbewegungsmethoden wie Teleportation oder Joystick-Steuerung, bei der man sich körperlich kaum bewegt, erfordert das Laufen auf einem ODT körperliche Anstrengung (Darken, Cockayne und Carmein 1997, 8). Ein großer Nachteil von Omni-Directional Treadmills ist, dass sie teuer und sperrig sind, was sie für den durchschnittlichen VR-Nutzer\*in unerschwinglich machen kann. Die meisten dieser Geräte sind auch schwer und erfordern eine Menge Platz, was bedeutet, dass sie nur in großen Räumen oder speziellen VR-Zentren eingesetzt werden können. Obwohl diese Geräte es Benutzern erlauben, in alle Richtungen zu gehen, ist die Bewegung immer noch begrenzt auf die Größe des Laufbandes, was bedeutet, dass Benutzer\*innen sich immer noch in einem begrenzten Bereich bewegen müssen und teilweise kleine und langsame Bewegungen nicht von dem Laufband getrackt werden (Darken, Cockayne und Carmein 1997, 2). Darüber hinaus können Omni-Directional Treadmills auch körperlich anstrengend sein, da sie Spielern ermöglichen, lange Strecken zurückzulegen, was zu Erschöpfung und Verletzungen führen kann. Darüber hinaus müssen sich die Spieler\*innen erst an das Laufband gewöhnen und von Grund auf lernen sich darauf fortzubewegen (Darken, Cockayne und Carmein 1997, 8).



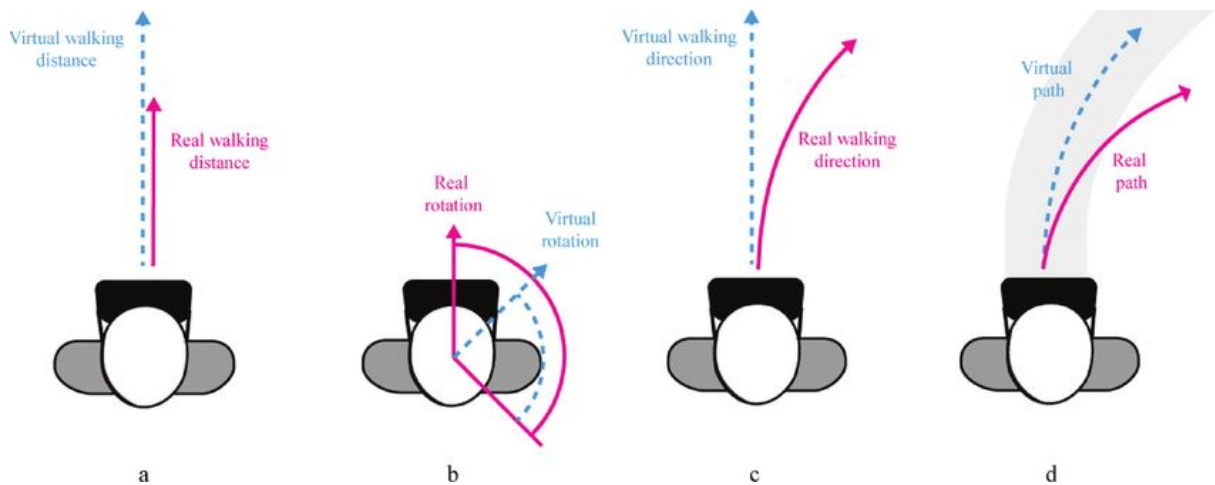


Abbildung 6: The vier Arten für perspektivische Manipulation: (a) Translation Gain, (b) Rotation Gain, (c) Curvature Gain, and (d) Bending Gain. Violette und blaue Linien zeigen die realen bzw. virtuellen Transformationen an (Nilsson u. a. 2018)

### 3 Redirected Walking als Lösungsansatz

Redirected Walking ist ein Ansatz, bei dem der\*die Nutzer\*in durch subtile Änderungen in der virtuellen Umgebung dazu gebracht wird, in eine bestimmte Richtung zu gehen, ohne es bewusst zu bemerken (Azmandian u. a. 2016, 2). Dieser Ansatz ermöglicht es dem\*der Benutzer\*in, sich in einer unbegrenzten virtuellen Welt zu bewegen, während er\*sie sich tatsächlich nur in einem begrenzten physischen Raum befindet. Forschungen haben gezeigt, dass Redirected Walking die Präsenz in der virtuellen Umgebung verbessern kann (Langbehn, Lubos und Steinicke 2018, 1). Die meisten Ansätze von Redirected Walking beinhalten die Verwendung von Algorithmen, die die Position und Ausrichtung des Benutzers im Raum erfassen und entsprechende virtuelle Veränderungen vornehmen, um ihn\*sie in eine bestimmte Richtung zu leiten (Azmandian u. a. 2016, 2). Ein Beispiel für die Verwendung von Redirected Walking in VR ist das Redirected Walking Toolkit, welches einem ermöglicht verschiedenste Algorithmen zu simulieren und selbst auf einem VR-Headset zu testen sowie eine Schnittstelle bietet um eigene Algorithmen dem Toolkit hinzuzufügen (Azmandian u. a. 2016, 1). Es gibt verschiedene Arten von Redirects die angewendet werden können, um Spieler\*innen zu manipulieren und seine\*ihre Bewegung in der virtuellen Welt zu kontrollieren. Jede dieser Techniken hat ihre eigenen Vorteile und Herausforderungen, die es zu verstehen gilt, um eine effektive VR-Erfahrung zu schaffen. In diesem Kapitel werden die verschiedenen Arten von Redirects untersucht und diskutiert, um ein besseres Verständnis für ihre Anwendungen und Grenzen zu schaffen.

#### 3.1 Scaling

Scaling ist ein bedeutendes Grundprinzip für Redirecting-Verfahren, das bei der Implementierung von Redirected Walking Algorithmen und Methoden in Virtual Reality Anwendung findet.

Das Ziel ist es, den\*die Anwender\*in unbemerkt zu führen und ihm\*ihr das Gefühl zu geben, sich in einer größeren Umgebung zu bewegen, als tatsächlich physisch vorhanden (Steinicke u. a. 2008, 7). Um das zu erreichen muss die VR-Umgebung kontinuierlich neu berechnet werden, um sicherzustellen, dass sie den Bewegungen der Testperson folgt, ohne den Eindruck zu erwecken, dass er\*sie sich in einer sich wiederholenden Umgebung bewegt. Scaling wird als Grundlage für die in den folgenden Kapiteln beschriebenen Arten der Umleitung (Redirection) verwendet. Translation Gain, Rotation Gain, Curvature Gain und Bending Gain nutzen alle Scaling, um die Bewegungen des Benutzers in bestimmte Richtungen zu vergrößern oder zu verkleinern (siehe Abbildung 4). Dadurch kann dem Benutzer das Gefühl vermittelt werden, in der virtuellen Realität eine größere oder kleinere Strecke zurückgelegt zu haben, als er tatsächlich in der physischen Welt zurückgelegt hat.

### 3.2 Translation Gain

Die erste Methode um redirecten zu erreichen nennt sich Translation Gain und ist ein Algorithmus, der bei Redirected Walking eingesetzt wird, um dem\*der Benutzer\*in das Gefühl zu geben, sich in einer geraden Linie in eine bestimmte Richtung zu bewegen. Jedoch werden für den\*die Anwender\*in die zurückgelegte Distanz entlang dieser Linie jeweils hoch- oder runter skaliert wodurch dieser in VR entweder eine größer oder kleiner Distanz als in echt zurücklegt (Steinicke u. a. 2009, 5), dargestellt in Abbildung 4 (a). Durch Translation Gain kann der\*die Benutzer\*in eine größere oder kleinere Entfernung zurücklegen, ohne dass er\*sie es bemerkt. Der Algorithmus funktioniert, indem er die Bewegungen des Spielers im virtuellen Raum manipuliert. Wenn der\*die Anwender\*in in eine bestimmte Richtung geht, wird der Algorithmus die virtuelle Welt in die entgegengesetzte Richtung drehen, um den\*die Spieler\*in auf einem scheinbar geraden Weg zu halten (Steinicke u. a. 2009, 5). Auf diese Weise kann man länger laufen, ohne das Gefühl zu bekommen, in Kreisen zu laufen oder sich zu drehen. Es gibt jedoch Grenzen für die Anwendung von Translation Gain. Wenn das hoch- oder runterskalieren zu viel wird kann der Algorithmus nicht mehr angewendet werden, da er zu offensichtlich wird und dieser „Trick“ zu auffällig wird (Steinicke u. a. 2008, 7). In der Forschung wurde gezeigt, dass Translation Gain eine vielversprechende Technik ist, um den Bewegungsspielraum in VR zu erweitern und dem\*der Anwender\*in das Gefühl zu geben, eine größere Entfernung zurückzulegen (Azmandian u. a. 2022, 6). Es wurde auch gezeigt, dass die Kombination von Translation Gain mit anderen Redirecting-Techniken, wie Rotation Gain, zu einem noch besseren VR-Erlebnis führen kann (Azmandian u. a. 2022, 6).

### 3.3 Rotation Gain

Die nächste Gängige Methode um Redirected Walking umzusetzen und zu erreichen ist „Rotation Gain“. Diese Methode wird verwendet, um dem\*der Benutzer\*in das Gefühl zu geben, sich in eine andere Richtung zu drehen, als er\*sie tatsächlich in der virtuellen Welt dreht (Steinicke u. a. 2009, 4). Die Rotation Gain-Methode wird erreicht, indem die Drehung des Benutzers in der virtuellen Welt erhöht oder verringert wird, während deren Bewegung in der realen

Welt unverändert bleibt (Steinicke u. a. 2009, 4), siehe Abbildung 4 (b). Einer der wichtigsten Faktoren ist die Geschwindigkeit, mit der sich Anwender\*innen bewegen. Die Rotation Gain-Methode ist effektiver bei langsamen Geschwindigkeiten, da bei höheren Geschwindigkeiten der\*die Benutzer\*in eher bemerkt, dass er\*sie sich nicht die gleiche Distanz dreht wie in der Realität. Außerdem ist der Threshold in denen die Rotationen hoch- und runterskaliert werden können limitiert, da es sonst zu auffällig für den\*die Anwender\*in ist (Steinicke u. a. 2008, 7). In diesen Fällen kann die Kombination mit anderen Redirecting-Methoden erforderlich sein, um eine realistische Erfahrung zu schaffen wie zum Beispiel Translation Gain. Wenn die induzierten Rotationen in Kombination mit Translation Gain klein genug sind, erhält der Benutzer den Eindruck, dass er in der virtuellen Welt in jede beliebige Richtung ohne Einschränkungen gehen kann (Steinicke u. a. 2008, 1). Eine weitere Herausforderung bei der Anwendung der Rotation Gain-Methode ist die Vermeidung von Motion Sickness. Wenn der\*die Benutzer\*in das Gefühl hat, sich tatsächlich oder zu schnell zu drehen, während er\*sie in der echten Welt sich nicht ähnlich dreht, kann dies dazu führen, dass man sich unwohl fühlt und das VR-Erlebnis nicht mehr genießen kann.

### 3.4 Curvature Gain

Curvature Gain ist eine weitere Technik des Redirected Walking, bei der der Bewegungspfad des Nutzers so angepasst wird, dass er\*sie in einer Kurve geht, anstatt in einer geraden Linie zu laufen. Diese Technik wird verwendet, um den verfügbaren Raum in der virtuellen Umgebung zu maximieren und gleichzeitig die Bewegung des Spielers unbemerkt zu manipulieren, wie zu sehen in Abbildung 4 (d). Bei der Verwendung von Curvature Gain wird der Pfad des Benutzers so geändert, dass er\*sie in einer Kurve läuft, während er\*sie den Eindruck hat, geradeaus zu laufen (Steinicke u. a. 2009, 4). Der Kurvenradius wird so berechnet, dass er klein genug ist, um den\*die Spieler\*in nicht zu stören, aber groß genug, um ihn\*sie in die gewünschte Richtung zu führen (Steinicke u. a. 2009, 9). Die Biegung des Pfades wird durch subtile Drehungen des virtuellen Raums erreicht, während der\*die Nutzer\*in sich fortbewegt (Azmandian u. a. 2022, 2). Die Verwendung von Curvature Gain ermöglicht es Nutzern, in einer virtuellen Umgebung zu navigieren, die viel größer ist als der physisch verfügbare Raum. Auf diese Weise kann der\*die Spieler\*in durch eine Stadt laufen oder eine virtuelle Landschaft erkunden, ohne den Raum wechseln zu müssen. Um Curvature Gain effektiv anzuwenden, müssen mehrere Faktoren berücksichtigt werden, wie zum Beispiel der Kurvenradius, die Geschwindigkeit des Nutzers und die Größe des verfügbaren Raums.

### 3.5 Bending Gain

Bending gain ist ein Konzept, das es ermöglicht, dem\*der Benutzer\*in das Gefühl zu geben, dass er\*sie sich auf einer geraden Strecke bewegt, während er\*sie in Wirklichkeit eine Kurve macht. Bending-Gains sind ähnlich wie Curvature gains, da sie Gehen und Rotationen kombinieren, jedoch auf gekrümmte Pfade anstelle von geraden Pfaden angewendet werden (Langbehn u. a. 2017, 3). Dies wird durch eine Kombination aus visueller Täuschung und haptischem

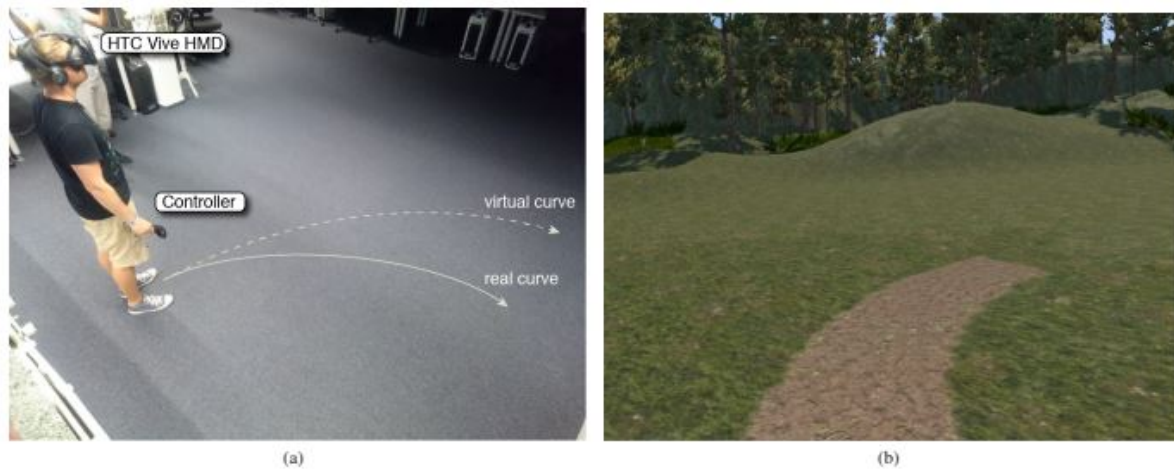


Abbildung 7: (a) Der\*Die Benutzer\*in geht in VR eine Krueve, welche einen anderen Radius hat als die Kurve der realen Welt auf welchem er umgeleitet wird, und (b) die Sicht des\*der Benutzers\*in in VR mit einem gekrümmten virtuellen Pfad (Langbehn u. a. 2017)

Feedback erreicht. Visuell wird die Szene in der virtuellen Welt so verändert, dass sie den Eindruck erweckt, als ob sich der\*die Spieler\*in auf einer geraden Strecke bewegt, obwohl er\*sie in Wirklichkeit eine Kurve macht (Langbehn u. a. 2017, 3). Zum Beispiel wird die Straße auf der virtuellen Welt so gebogen, dass sie der tatsächlichen Kurve entspricht, auf der man sich in der realen Welt befindet (Abb. 7). Haptisches Feedback wird verwendet, um dem\*der Anwender\*in das Gefühl zu geben, eine Kurve zu laufen (Langbehn u. a. 2017, 2). Zum Beispiel kann eine leichte Vibration oder ein Druck auf der Seite des Körpers erzeugt werden, die der Richtung der Biegung in der virtuellen Welt entspricht.

## 4 Endless Runner in Virtual Reality mit Redirected Walking

Das Prinzip von Endless Runner-Spielen ist vergleichsweise einfach. Der\*Die Spieler\*in steuert eine Figur oder ein Objekt, die sich auf einer endlos generierten Strecke fortbewegt. Ziel des Spiels ist es, möglichst weit zu kommen und dabei so viele Punkte wie möglich zu sammeln. Die Spielfigur bewegt sich dabei automatisch vorwärts und der\*die Spieler\*in muss Hindernissen ausweichen, Sprünge machen und gegebenenfalls Gegner bekämpfen (Abbildung 8.). Typischerweise ist das Gameplay von Endless Runner-Spielen sehr schnell und erfordert rasche Reaktionszeiten von dem\*der Spieler\*in. Die Strecke ist meist mit verschiedenen Hindernissen gespickt, die es zu überwinden gilt, wie zum Beispiel Lücken im Boden, Wänden oder Gegnern. Um diese Hindernisse zu überwinden, muss der\*die Spieler\*in meistens springen, ducken oder seitlich ausweichen. Die Strecke in Endless Runner-Spielen wird dabei automatisch generiert und ist unendlich lang, daher der Name „endless“. Die Geschwindigkeit der Spielfigur nimmt dabei mit der Zeit zu, was das Spiel immer schwieriger macht und für die Reaktionszeit des Spielers eine Herausforderung darstellt. Endless Runner-Spiele sind vor allem

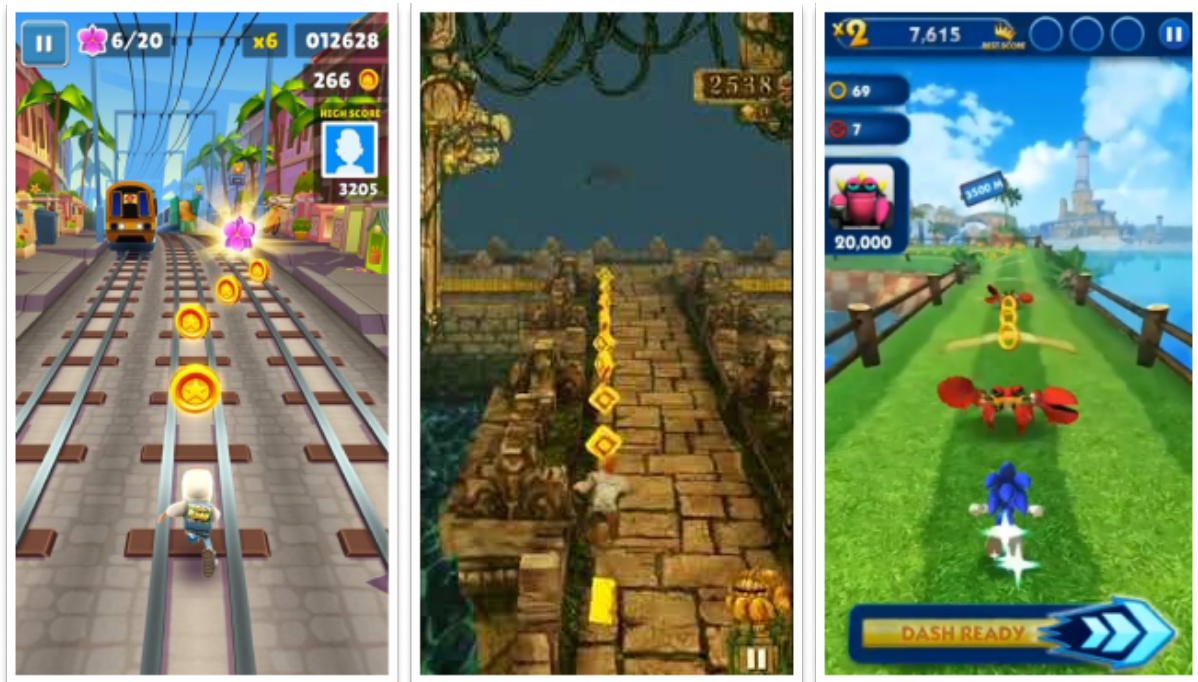


Abbildung 8: Endless Runner Games Subway Surfer, Temple Run und Sonic Dash (Quelle: <https://machinations.io/docs/infinite-runner-game-loop/>)

auf mobilen Geräten sehr beliebt, da sie schnell zu spielen sind und ein kurzweiliges Spielvergnügen bieten. Aber auch auf anderen Plattformen, wie zum Beispiel VR, können Endless Runner-Spiele zu einer Herausforderung werden. Wenn man einen Endless Runner in VR entwickeln möchte, müssen die Grenzen von VR berücksichtigt werden. Eine automatische Bewegung durch eine unendlich generierte Welt, wie es in traditionellen Endless Runner Spielen der Fall ist, würde in VR zu Motion Sickness führen und die Immersion beeinträchtigen. Im Gegensatz zu herkömmlichen Endless Runnern auf mobilen Geräten oder PCs, bei denen die Steuerung meist auf Taps oder Swipes beschränkt ist, bietet VR eine viel größere Bandbreite an Interaktionsmöglichkeiten. Daher wird Redirected Walking eingesetzt, um dem\*der Spieler\*in das Gefühl zu geben, selbst Teil des Spiels zu sein, indem er\*sie versucht, dem unabwendbaren Tod durchs Davonlaufen zu entkommen. Dabei versucht der\*die Spieler\*in auf einem unendlich generierten Weg von Plattformen so weit wie möglich voranzuschreiten. Jede Plattform, sei es eine Gerade oder eine Kurve, besitzt einen sogenannten „Waypoint“, den es als Checkpoint zu erreichen gibt und auf welchen die Applikation den\*die Spieler\*in hinleitet. Dieser Waypoint repräsentiert Münzen in traditionellen Endless Runner Games. Bei Erreichen dieses Checkpoints bekommt man Punkte, die den Highscore bestimmen. Des Weiteren werden bei Erreichen der Waypoints neue Plattformen generiert, um das endlose Level weitergenerieren zu können. Zusätzlich wird nach Erreichen eines Waypoints der nächste zu erreichende Checkpoint sichtbar, auf welchen der\*die Spieler\*in hingeleitet wird. Durch das ständige Aktualisieren der Checkpoints hat der\*die Spieler\*in dauerhaft das Gefühl, frei durch die virtuelle Welt laufen zu können. Hindernisse wie Schluchten, umgefallene Bäume und riesige Steine können nicht

durch künstliches Springen überwunden werden, da dies wiederum Motion Sickness verursachen kann und man in der Realität nicht hoch genug Springen kann, um große Hindernisse zu überwinden. Stattdessen wird das Spiel so gestaltet, dass sich der bereits vorhandene Weg immer schneller auflöst, je weiter der\*die Anwender\*in kommt. Wenn der\*die Spieler\*in nicht schnell genug zum nächsten Checkpoint gelangt, stürzt er\*sie in den Tod und das Spiel endet.

## 5 Entwicklung eines Endless Runners in VR

Um das Setup für die Applikation nicht allzu komplex zu gestalten, wurde das vorgefertigte VR-Template von Unity als Ausgangspunkt gewählt, welches die notwendigen Komponenten und Einstellungen für das VR-Setup bereitstellt. Das Template enthält ein paar grundlegende Komponenten, um das VR-Setup in Unity zu realisieren. Zum Beispiel gibt es eine Main Camera, die für die Darstellung der Szene in VR zuständig ist. Zusätzlich gibt es eine Game-Object-Komponente namens „XR Rig“, welche dafür sorgt, dass die Kameraposition und -orientierung an die des VR-Headsets angepasst wird. Das Template beinhaltet auch ein „Input System“, das Eingaben aus dem VR-Headset und den Controllern empfängt und an die entsprechenden Game Objects weiterleitet, welches Momentan für die aktuelle Versions des Spiels keine weitere Relevanz hat. Darüber hinaus ist das Template mit einer Vielzahl von Einstellungen vorkonfiguriert, um eine möglichst optimale Performance und visuelle Qualität in VR zu gewährleisten. Dazu gehören unter anderem die richtige Skalierung der Szene und die Anpassung der Render-Einstellungen für die VR-Brille.

### 5.1 Hauptkomponenten

Das ganze Spiel baut auf das "Redirected Walking Toolkit" von Azmandian u. a. 2016 auf. Dieses stellt die Hauptkomponenten für Redirected Walking zur Verfügung. Dieses besteht aus einem Prefab Namens „Redirected User“ welcher den\*die Spieler\*in in der virtuellen Welt repräsentiert. Wenn der User redirected wird, verändert diese Game Objekt Position und Rotation (Azmandian u. a. 2016, 2). Der „Redirected User“ ist im Wesentlichen der Ursprung der realen Welt in Bezug auf die virtuelle Welt. Die Anwendung von „redirection“ bedeutet daher im Wesentlichen die Referenz der realen Welt in Bezug auf die virtuelle Welt zu bewegen (Azmandian u. a. 2016, 2). Das tatsächliche „redirecten“ übernimmt das Redirector Skript und das Zurücksetzen für den Fall, dass der\*die Benutzer\*in ans Ende der physisch verfügbaren Fläche gelangt, ein Resetter Skript. Die beiden Skripts werden zur Laufzeit durch den **RedirectionManager** dem Redirected User Prefab hinzugefügt. Des Weiteren bietet das Toolkit noch Skripts zur Analyse des zurückgelegten Wegs des Spielers. Das **TrailDrawer** Skript erzeugt einen sichtbaren Pfad unter Berücksichtigung des physisch verfügbaren Platzes und der virtuellen Welt (Azmandian u. a. 2016, 4).



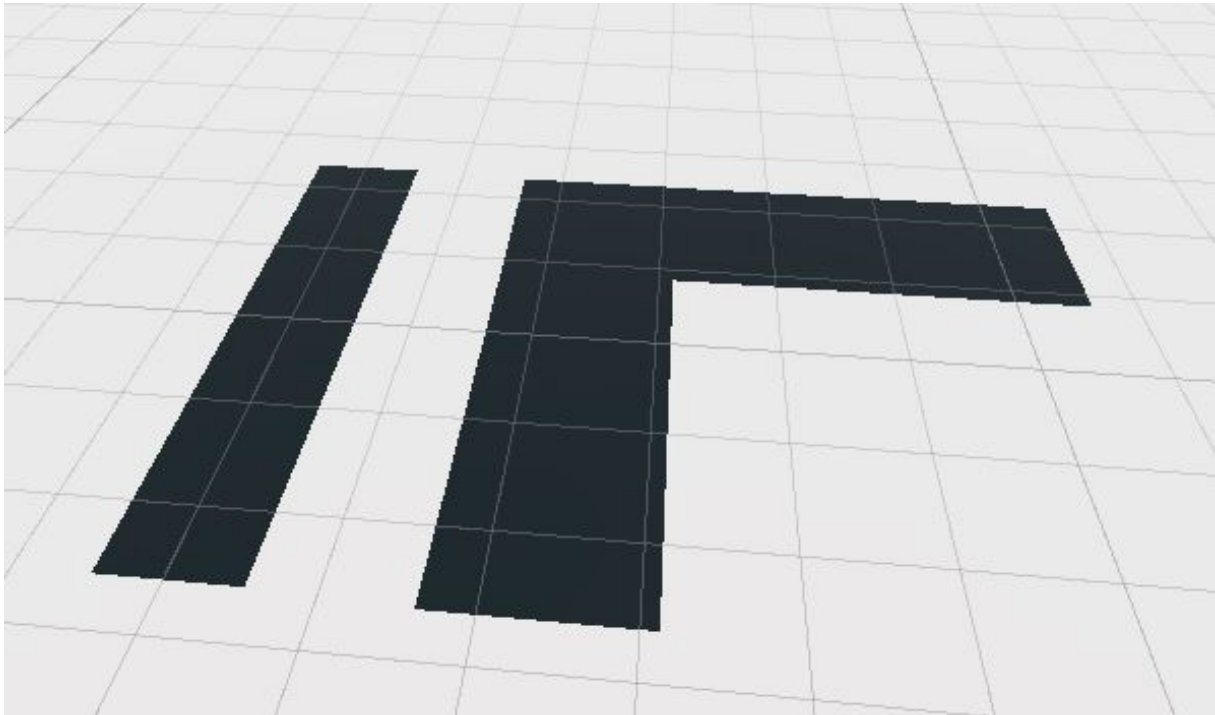


Abbildung 9: Plattform Prefabs

## 5.2 Game Manager

Um das Spiel mit dem unterliegenden Toolkit zu verbinden, wurde ein `SettingsManager` geschrieben. Dieser wird über den `RedirectionManager` initialisiert und aufgesetzt. Bei der Initialisierung werden sowohl Redirection Algorithmus als auch Reset Condition, welche der\*die Benutzer\*in vor starten der App angibt, ausgelesen und initialisiert. Des Weiteren ist der `SettingsManager` dafür verantwortlich, das Redirecten zu starten. Beim Start wird auch gleichzeitig der erste zu erreichende Waypoint gespawnt. Bei Erreichen des Waypoints wird dieser durch den nächsten ersetzt. Zu guter Letzt bestimmt diese Klasse noch die Geschwindigkeit in der die Plattformen wieder verschwinden.

## 5.3 Unendliche Map Generation

Für die Ezeugung der Plattformen zur unendlichen Mapgenerierung ist der **EndlessRunner-Manager** verantwortlich. Dieser kennt zwei Prefabs (Abbildung 9.): die **Straight**, welche eine Gerade in der Welt darstellt, und ein **Turn** Prefab, welches eine Kurve beschreibt. Außerdem ist die Länge der einzelnen Teile in Meter unter der Variable **ROADLENGTH** bekannt, um das unendliche Generieren und Anordnen der verschiedenen Plattformen in zufälliger Reihenfolge zu ermöglichen. Zusätzlich speichert die Applikation die momentane Richtung des Weges (rechts, links, vorwärts, rückwärts) um die Rotation der Geraden und Kurven beim Spawnen richtig anpassen zu können. Die nächste Plattform wird durch eine Zufallszahl generiert, wobei

hier die Wahrscheinlichkeit bei 60 Prozent liegt eine Gerade zu sein und zu 40 Prozent eine Kurve gespawnt wird (20 Prozent Rechts- und 20 Prozent Wahrscheinlichkeit für Linkskurve). Um zu gewährleisten, dass die neuen Teile immer an der zuletzt erzeugten Plattform angrenzen, wird die Position des zuletzt generierten Wegstücks gespeichert und als Ausgangspunkt für die neuen Wegteile verwendet. Je nachdem in welche Richtung sich der Weg gerade neigt, werden zu der gespeicherten Position die ROADLENGTH zu den X und Z Koordinaten addiert oder subtrahiert.

## 5.4 Redirecting Algorithmen

Redirecting wird durch geschickte Anwendung von Algorithmen erreicht, die dem\*der Nutzer\*in das Gefühl vermitteln, sich in einer natürlicheren Umgebung zu bewegen. Diese verwenden Techniken wie Rotation Gain, Translation Gain und Curvature Gain, um Bewegungen des Nutzers zu manipulieren und den Eindruck einer natürlicheren Umgebung zu erzeugen. Eine der größten Herausforderungen bei der Implementierung besteht darin, dass sie in Echtzeit ausgeführt werden müssen, um den\*die Spieler\*in ein reibungsloses Erlebnis zu bieten. Außerdem müssen sie nahtlos in die virtuelle Umgebung integriert werden, um den\*die Nutzer\*in nicht aus dem Erlebnis zu reißen. Das Ziel ist es, eine so realistische Erfahrung wie möglich zu bieten, ohne dabei auf die Vorteile von Virtual Reality verzichten zu müssen. In diesem Kapitel werden auf die im praktischen Teil verwendeten Redirected Walking-Algorithmen eingegangen und ihre Funktionsweise im Detail erläutert.

### 5.4.1 Redirection Manager

Diese Klasse ist die Hauptkomponente der Applikation und steuert das Redirecten und Resetten. Dieser Manager initialisiert alle Komponenten und andere im Spiel verwendete Manager. Darüber hinaus speichert sie die Position des Spielers und der Kameraposition um Redirecten anwenden zu können. Dabei werden im **LateUpdate** die Position und Ausrichtung des Nutzers aktualisiert um sie für die Berechnungen des Rotation- und Translationsanpassung als Ausgangspunkt verwenden zu können indem aus ihnen später die Deltawerte berechnet werden. Des Weiteren wird in dieser Methode unterschieden, ob nun **Redirecting** oder **Resetting** anzuwenden ist. Dabei wird der gespeicherte Tracked Space (physisch verfügbare Spielfläche) mit der Position des Spielers abgeglichen und getestet, ob der\*die Benutzer\*in Gefahr läuft, die Spielfläche zu verlassen. Wenn kein Resetten erforderlich ist, spricht der **RedirectionManager** den **Redirector** an, um den\*die Spieler\*in auf Basis der zwei zu Verfügung stehenden Algorithmen umzuleiten. Die Klasse **SteerToRedirector** ist hierbei ein wichtiger Schlüsselteil der Applikation. Sie dient als Schnittstelle zwischen **RedirectionManager** und dem unterliegenden Algorithmus. Die Klasse ist durch Vererbung so gebaut, dass man jederzeit den Redirection Algorithmus austauschen kann und auch selbst neue hinzufügen kann. Sie leitet von der Parent-Klasse **Redirector** ab (Abbildung 10), welche letztendlich auf den\*die Spieler\*in eine Rotation, Kurvenanpassung oder eine Translation anwendet, um ihn unschwellig in die gewünschte Richtung zu lenken. Die Klasse **SteerToRedirector** selbst ist auch eine Parent Klasse, aus



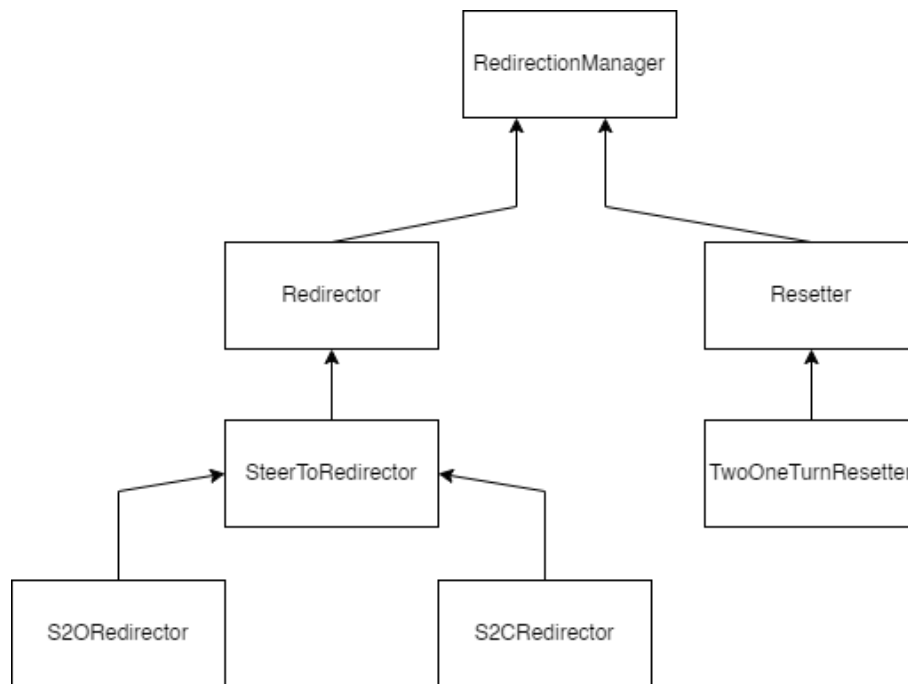


Abbildung 10: Vererbungshierarchie der Applikation

der die tatsächlichen Redirecting Algorithmen ableiten. Durch diesen Aufbau ist es jederzeit möglich, neue Algorithmen der Applikation hinzuzufügen.

Die Funktion **ApplyRedirection** wird vom **RedirectionManager** aufgerufen und wählt anfangs das temporäre Redirection Target aus, welches durch Anwendung eines Redirecting Algorithmus berechnet wird.

```

1 Vector3 deltaPos = redirectionManager.deltaPos;
2 float deltaDir = redirectionManager.deltaDir;
3
4 rotationFromCurvatureGain = 0;
5
6 if (deltaPos.magnitude / redirectionManager.GetDeltaTime() >
    MOVEMENT_THRESHOLD)
7 {
8     rotationFromCurvatureGain = Mathf.Rad2Deg * (deltaPos.magnitude /
        redirectionManager.CURVATURE_RADIUS);
9     rotationFromCurvatureGain = Mathf.Min(rotationFromCurvatureGain,
        CURVATURE_GAIN_CAP_DEGREES_PER_SECOND * redirectionManager.
        GetDeltaTime());
10 }
  
```

Anschließend wird die Delta-Position und Delta-Rotation des Benutzers im Vergleich zum vorherigen Frame abgerufen. Die Variable **rotationFromCurvatureGain** wird auf Null gesetzt, bevor berechnet wird, ob der\*die Benutzer\*in sich bewegt und ob die Krümmung der Umlenkung einen Beitrag zur Rotation leisten soll. Wenn die Bewegungsschwelle überschritten wird,

wird die Rotation aufgrund der Krümmung berechnet. Die Rotation wird jedoch durch eine Obergrenze (**CURVATURE\_GAIN\_CAP\_DEGREES\_PER\_SECOND**) begrenzt.

```

1 Vector3 desiredFacingDirection = Utilities.FlattenedPos3D(currentTarget.
    position) - redirectionManager.currPos;
2 int desiredSteeringDirection = (-1) * (int)Mathf.Sign(Utilities.
    GetSignedAngle(redirectionManager.currDir, desiredFacingDirection));
3
4 if (Mathf.Abs(deltaDir) / redirectionManager.GetDeltaTime() >=
    ROTATION_THRESHOLD)
5 {
6
7     //Determine if we need to rotate with or against the user
8     if (deltaDir * desiredSteeringDirection < 0)
9     {
10         //Rotating against the user
11         rotationFromRotationGain = Mathf.Min(Mathf.Abs(deltaDir *
            redirectionManager.MIN_ROT_GAIN),
            ROTATION_GAIN_CAP_DEGREES_PER_SECOND * redirectionManager.
            GetDeltaTime());
12     }
13     else
14     {
15         //Rotating with the user
16         rotationFromRotationGain = Mathf.Min(Mathf.Abs(deltaDir *
            redirectionManager.MAX_ROT_GAIN),
            ROTATION_GAIN_CAP_DEGREES_PER_SECOND * redirectionManager.
            GetDeltaTime());
17     }
18 }

```

Durch das zuvor berechnete Redirection Target kann nun die Richtung berechnet werden, in welche der\*die Benutzer\*in umgeleitet wird. Wenn der\*die Spieler\*in sich dreht, wird berechnet, ob die Umlenkung mit oder gegen die Drehrichtung des Anwenders erfolgen soll. Dann wird die Rotation aus der Drehung berechnet. Auch hier wird die Rotation durch eine Obergrenze (**ROTATION\_GAIN\_CAP\_DEGREES\_PER\_SECOND**) begrenzt.

```

1 float rotationProposed = desiredSteeringDirection * Mathf.Max(
    rotationFromRotationGain, rotationFromCurvatureGain);
2 bool curvatureGainUsed = rotationFromCurvatureGain >
    rotationFromRotationGain;
3
4 if (!dontUseDampening)
5 {
6     float bearingToTarget = Vector3.Angle(redirectionManager.currDir,
        desiredFacingDirection);
7     if (useBearingThresholdBasedRotationDampeningTimofey)
8     {
9         if (bearingToTarget <= BEARING_THRESHOLD_FOR_DAMPENING)
10             rotationProposed *= Mathf.Sin(Mathf.Deg2Rad * 90 *
                bearingToTarget / BEARING_THRESHOLD_FOR_DAMPENING);
11     }
12     else

```

```

13     {
14         rotationProposed *= Mathf.Sin(Mathf.Deg2Rad * bearingToTarget);
15     }
16     if (desiredFacingDirection.magnitude <=
17         DISTANCE_THRESHOLD_FOR_DAMPENING)
18     {
19         rotationProposed *= desiredFacingDirection.magnitude /
20         DISTANCE_THRESHOLD_FOR_DAMPENING;
21     }
22 }
23 float finalRotation = (1.0f - SMOOTHING_FACTOR) * lastRotationApplied +
24     SMOOTHING_FACTOR * rotationProposed;
25 if (!curvatureGainUsed)
26     InjectRotation(finalRotation);
27 else
28     InjectCurvature(finalRotation);

```

Die vorgeschlagene Rotation (**rotationProposed**) wird dann unter Berücksichtigung des Dämpfungsfaktors berechnet. Hierbei werden die Drehung und die Drehgeschwindigkeit des Benutzers gedämpft, wenn der\*die Spieler\*in sich in der Nähe des Ziels befindet. Die berechnete Rotation wird dann mit einem Glättungsfaktor (**SMOOTHING\_FACTOR**) kombiniert, um eine flüssigere Bewegung zu erzeugen. Schließlich wird die berechnete Rotation injiziert, um die Umlenkung auf den\*die Bediener\*in anzuwenden.

#### 5.4.2 Steer-To-Center

Die Grundidee hinter dem Steer-to-Center-Algorithmus besteht darin, den\*die Spieler\*in durch Verwendung von Rotation- und Curvature Gains in der Mitte des physischen Raums zu halten, während er\*sie durch den virtuellen Raum geht. Der Algorithmus überwacht die Position des Benutzers innerhalb des physischen Raums und passt die virtuelle Umgebung an, um ihn\*sie in die Mitte des Raums zurückzubringen, wenn er\*sie sich zu weit von ihr entfernt (Azmandian u. a. 2016, 2). Die Umsetzung des Steer-to-Center-Algorithmus erfolgt durch die kontinuierliche Überwachung der Bewegungen des Spielenden und die entsprechende Anpassung der virtuellen Umgebung (Azmandian u. a. 2016, 2). Wenn der\*die Anwender\*in in eine Richtung geht, die ihn zu weit vom Zentrum entfernt, dreht die virtuelle Umgebung sich in die entgegengesetzte Richtung, um ihn\*sie zurück zum Zentrum zu führen. Dieser Prozess geschieht auf eine subtile Weise, so dass der\*die Spieler\*in es nicht bemerkt und weiterhin glaubt, dass man sich in einer größeren virtuellen Welt bewegt.

Die Funktion **PickRedirectionTarget** der Klasse **S2CRedirector** wird hierbei jedes mal im **LateUpdate** aufgerufen und aktualisiert die Richtung in welche der\*die Spieler\*in gelenkt werden muss, um ins Zentrum zu gelangen.

```

1 Vector3 trackingAreaPosition = Utilities.FlattenedPos3D(redirectionManager.
2   trackedSpace.position);
3 Vector3 userToCenter = trackingAreaPosition - redirectionManager.currPos;

```

Zunächst wird die **trackingAreaPosition** vom **RedirectionManager** ausgelesen, die den physisch verfügbaren Raum in der virtuellen Welt abbildet. Anschließend wird ein Richtungsvektor von der aktuellen Position des Spielers zur Mitte der verfolgten Fläche berechnet, da der\*die Spieler\*in in Richtung des Zentrums der verfügbaren Spielfläche gelenkt werden soll.

```
1 float bearingToCenter = Vector3.Angle(userToCenter, redirectionManager.
    currDir);
2 float directionToCenter = Utilities.GetSignedAngle(redirectionManager.
    currDir, userToCenter);
```

Anschließend wird der Winkel berechnet, um den sich der\*die Nutzer\*in drehen muss, um zum Zentrum zu gelangen, sowie die Richtung, welche eingeschlagen werden muss. Diese Berechnungen dienen als Grundlage für das sogenannte Redirection Target.

```
1 if (bearingToCenter >= S2C_BEARING_ANGLE_THRESHOLD_IN_DEGREE && !
    dontUseTempTargetInS2C)
2     {
3         //Generate temporary target
4         if (noTmpTarget)
5         {
6             tmpTarget = new GameObject("S2C Temp Target");
7             tmpTarget.transform.position = redirectionManager.currPos +
                S2C_TEMP_TARGET_DISTANCE * (Quaternion.Euler(0,
                directionToCenter * 90, 0) * redirectionManager.currDir)
                ;
8             tmpTarget.transform.parent = transform;
9             noTmpTarget = false;
10        }
11        currentTarget = tmpTarget.transform;
12    }
13    else
14    {
15        currentTarget = redirectionManager.trackedSpace;
16        if (!noTmpTarget)
17        {
18            GameObject.Destroy(tmpTarget);
19            noTmpTarget = true;
20        }
21    }
```

Wenn dieser Winkel größer als eine bestimmte Schwelle ist und die temporäre Zielposition noch nicht festgelegt wurde, wird eine temporäre Zielposition erstellt und als aktuelles Ziel verwendet. Die temporäre Zielposition wird berechnet, indem die Blickrichtung des Benutzers um einen bestimmten Winkel gedreht wird und dann in eine bestimmte Entfernung von der aktuellen Position des Benutzers gesetzt wird. Wenn der Winkel kleiner als die Schwelle ist oder die temporäre Zielposition bereits festgelegt wurde, wird die Mitte des Trackingbereichs als Ziel verwendet.

### 5.4.3 Steer-To-Orbit

Steer-to-Orbit ist ein weiterer Algorithmus, der in Redirected Walking in VR verwendet wird. Im Gegensatz zum Steer-to-Center-Algorithmus, der den\*die Benutzer in die Mitte des virtuellen Raums lenkt, lenkt der Steer-to-Orbit-Algorithmus der\*die Anwender\*in in eine kreisförmige Bahn um das Zentrum des virtuellen Raums. Die Idee hinter dem Steer-to-Orbit ist, dass man einen Kreis um das Zentrum des virtuellen Raums beschreibt, während die Testperson glaubt, in einer geraden Linie zu laufen. Dies wird erreicht, indem der\*die Spieler\*in durch die Verwendung von Rotation Gain langsam um das Zentrum des virtuellen Raums herumgedreht wird, während er\*sie in eine bestimmte Richtung geht (Azmandian u. a. 2016, 2). Der Algorithmus funktioniert, indem die Bewegungen des Benutzers im realen Raum nachverfolgt wird und dann die Bewegungen im virtuellen Raum so anpasst werden, dass der\*die Spieler\*in eine Bahn um das Zentrum des virtuellen Raums läuft (Azmandian u. a. 2016, 2). Wenn der\*die Benutzer\*in sich beispielsweise nach rechts dreht, wird er\*sie im virtuellen Raum nach links gedreht, um seine Bewegungen in eine kreisförmige Bahn um das Zentrum des virtuellen Raums zu lenken. Dies kann zu einem realistischeren und immersiveren VR-Erlebnis führen, da der\*die Anwender\*in das Gefühl hat, sich in einer größeren und offeneren Welt zu bewegen.

Die Funktion **PickRedirectionTarget** der Klasse **S2ORedirector** wird hierbei jedes mal im **LateUpdate** aufgerufen und aktualisiert die Richtung in welche der\*die Spieler\*in gelenkt werden muss, um sich im Kreis ums Zentrum zu bewegen.

```
1 Vector3 trackingAreaPosition = Utilities.FlattenedPos3D(redirectionManager.  
    trackedSpace.position);  
2 Vector3 userToCenter = trackingAreaPosition - redirectionManager.currPos;
```

Wie auch schon bei Steer-to-Center wird hierbei als erstes die Position des Tracking-Bereichs des Redirection Managers ausgelesen und ein Richtungsvektor vom aktuellen Standort des Spielers zum Zentrum des Tracking-Bereichs berechnet.

```
1 if (noTmpTarget)  
2 {  
3     tmpTarget = new GameObject("S2O Target");  
4     currentTarget = tmpTarget.transform;  
5     noTmpTarget = false;  
6 }
```

Als Nächstes wird überprüft, ob bereits ein temporäres Ziel erstellt wurde. Wenn nicht, wird ein neues Game Object erstellt und als temporäres Ziel verwendet.

```
1 float alpha;  
2  
3 if (userToCenter.magnitude < S2O_TARGET_RADIUS) //Inside the orbit  
4 {  
5     alpha = S2O_TARGET_GENERATION_ANGLE_IN_DEGREES;  
6 }  
7 else  
8 {  
9     alpha = Mathf.Acos(S2O_TARGET_RADIUS / userToCenter.magnitude) * Mathf.  
    Rad2Deg;  
10 }
```

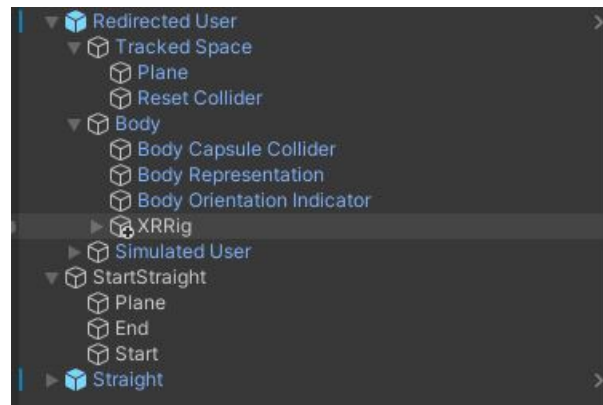


Abbildung 11: Unity Game Hierarchie

Daraufhin wird die Umleitung zur Umlaufbahn in 3 Schritten berechnet. Zunächst werden Winkel berechnet, die vom Zentrum zu potenziellen Zielen auf der Umlaufbahn führen. Hierbei wird berücksichtigt, wo sich der\*die Benutzer\*in aktuell befindet. Falls er\*sie sich bereits auf der Umlaufbahn befindet, wird der vordefinierte Winkel von  $60^\circ$  zur Umleitung verwendet. Falls sich der\*die Spieler\*in nicht auf der Umlaufbahn befindet, wird der Winkel durch die Verwendung von Tangenten von der Richtung zur Mitte der Spielfläche und dem Radius der Umlaufbahn berechnet.

```
1 Vector3 dir1 = Quaternion.Euler(0, alpha, 0) * -userToCenter.normalized;
2 Vector3 targetPosition1 = trackingAreaPosition + S2O_TARGET_RADIUS * dir1;
3 Vector3 dir2 = Quaternion.Euler(0, -alpha, 0) * -userToCenter.normalized;
4 Vector3 targetPosition2 = trackingAreaPosition + S2O_TARGET_RADIUS * dir2;
```

Der zweite Schritt besteht darin, aus dem berechneten Winkel die Richtung zu zwei potentiellen Zielen in der Umlaufbahn zu finden.

```
1 float angle1 = Vector3.Angle(redirectionManager.currDir, targetPosition1 -
    redirectionManager.currPos);
2 float angle2 = Vector3.Angle(redirectionManager.currDir, targetPosition2 -
    redirectionManager.currPos);
3 currentTarget.transform.position = (angle1 <= angle2) ? targetPosition1 :
    targetPosition2;
```

Abschließend werden die berechneten Richtungen mit der aktuellen Blickrichtung des Spielers verglichen und der Winkel zwischen der berechneten Richtung und der aktuellen Blickrichtung berechnet. Daraufhin werden diese Winkel erneut verglichen und der kleinere der beiden Winkel als Ergebnis ausgewählt.

## 5.5 Resets kurz vor Verlassen des Spielbereichs

Natürlich müssen auch die Szenarien bedacht werden, in denen das Redirecten nicht effektiv genug arbeiten kann, um den\*die Spieler\*in permanent in der verfügbaren Spielfläche hin- und

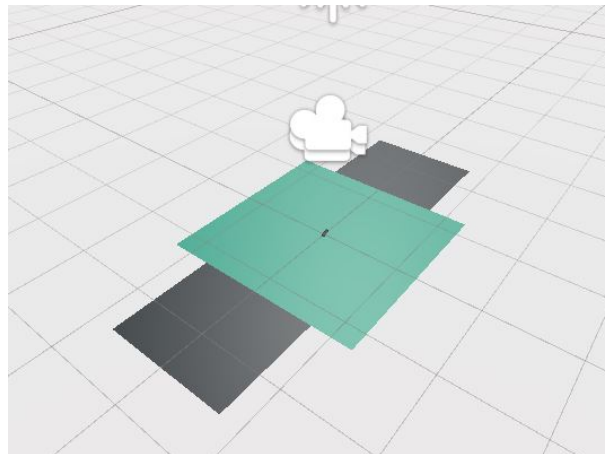


Abbildung 12: Physisch verfügbare Fläche in relation zum virtuellen Raum

herzuleiten. Dies kann unterschiedliche Gründe haben, wie die Spielefläche ist zu klein oder der\*die Anwender\*in bewegt sich bewusst/unbewusst ans Ende. Für diese Situationen ist der sogenannte Resetter verantwortlich. Die **trackedArea** in der Unity Hierarchie (Abbildung 11) ist die Repräsentation der physisch verfügbaren Fläche in der virtuellen Welt. Diese kann skaliert werden, indem man das Scaling des GameObjects anpasst. Dabei repräsentieren die x- und z-Werte die verfügbare Fläche in Meter. Die ganze Fläche besitzt in Unity einen Box-Collider, welcher als Trigger fungiert (Abbildung 12). Läuft der\*die Spieler\*in Gefahr aus diesem Trigger hinauszulaufen, feuert der Resetter ein Event, welches das Redirecten zum nächsten Waypoint stoppt und erst wieder fortführt, wenn der\*die Benutzer\*in die Reset-Condition erfüllt hat. Dabei gibt es unterschiedliche Arten des Resetten. In dieser Arbeit wurde das Konzept des Two-One-Turn benutzt. Der Algorithmus von Two-One-Turn basiert auf der Idee, dass der\*die Anwender\*in zuerst in eine neue Richtung gesteuert wird, bevor er\*sie in die gewünschte Richtung gedreht wird. Konkret wird dem Spielenden zuerst angeordnet sich am Stand um 360 Grad zu drehen, während die virtuelle Drehung mit dem Faktor 2 multipliziert wird (Azmandian u. a. 2016, 3). Das resultiert in einer 180° Drehung in Echt, während sich die virtuelle Welt für die Testperson gar nicht gedreht hat. Diese Methode kann auch als Rotation Gain angesehen werden, die an der Grenze des physischen Platzes angewendet wird und eine 180° Drehung in Echt auf eine 360° Drehung in VR abbildet (Azmandian u. a. 2016, 3). Dadurch wird der\*die Anwender\*in auf natürliche Weise in die neue Richtung gelenkt, ohne dass er\*sie dies bewusst wahrnimmt. Nachdem die 360° Rotation in echt durchgeführt wurde, wird der Resetter wieder auf Standby gesetzt und das Redirecten wird fortgesetzt.

## 6 Evaluierung des Programms mittels Studie

Die Sinnhaftigkeit von Redirected Walking in VR-Applikationen wurde mittels einer eigenen Studie ermittelt und die Ergebnisse im folgenden Kapitel offengelegt. Hierbei wurde insbesondere Wert auf die User Experience gelegt, der körperlichen Anstrengung und darauf, wie

auffällig das Umleiten für die Probanden\*innen war.

*Studien Design.* Es wurde ein Within-Subject Design benutzt, wobei Probanden\*innen in VR einem endlos generierten Pfad folgen mussten. Während dem Test veränderten sich pro Applikationsstart der Redirection Algorithmus und die physisch verfügbare Spielfläche. Dadurch ergaben sich folgende Testfälle mit den unabhängigen Variablen des *Redirection Algorithmus* und der *physisch verfügbaren Spielfläche*: (1) Steer-To-Center mit 10\*10 Metern Spielfläche, (2) Steer-To-Center mit 5\*5 Metern Spielfläche, (3) Steer-To-Center mit 2,5\*2,5 Metern Spielfläche, (4) Steer-To-Orbit mit 10\*10 Metern Spielfläche, (5) Steer-To-Orbit mit 5\*5 Metern Spielfläche und (6) Steer-To-Orbit mit 2,5\*2,5 Metern Spielfläche.

*Teilnehmer\*innen.* Für die Benutzerstudie haben sich 5 Probanden\*innen (1 weibliche Person, 4 männlichen Personen,  $\bar{x}$  = 30.4 Jahre,  $s$  = 12.5 Jahre) zur Verfügung gestellt, welche alle physisch in einer Verfassung waren, sich ohne Beeinträchtigung normal fortzubewegen bzw. sich laufend fortzubewegen. Dabei wurde darauf geachtet, dass bei den Probanden\*innen ein Unterschied im Alter, der Vorerfahrung mit VR und dem Berufsfeld gegeben war, um ein breit gelegtes Ergebnis zu bekommen.

*Apparatus.* Jede Testperson wurde mit einer unmodifizierten Oculus Quest 2 ausgestattet. Je nach Bedingung wurde die Spielfläche, welches das Headset tracked angepasst.

*Aufgabe.* Aufgabe war es, im Endless jeweils soweit wie möglich fortzuschreiten (Fortschritt wurde mittels erreichten Waypoints ermittelt), bevor man in den Abgrund stürzt. Die Fortbewegung im Spiel basiert hierbei ausschließlich auf tatsächliches Bewegen in der Realität.

*Datenerhebung.* Die Datenerhebung fand mittels Fragebogen auf Basis der likert Skala statt. Es wurden dabei sowohl quantifizierbare Fragen als auch qualitatives Feedback der Probanden eingeholt. Bei den quantifizierbaren Fragen galt es, herauszufinden wie sehr die Testperson durch die begrenzte physisch verfügbare Fläche beeinträchtigt war und wie störend/auffällig das Redirecten zum jeweilig nächsten Waypoint wahrgenommen wurde. Beim qualitativen Feedback wurde Wert auf die Benutzererfahrung und den Spielspaß gelegt, sowie auf die möglichen körperlichen Grenzen an denen man als Spieler\*in stoßen kann, im Bezug auf das immer schnellere despawnen der Plattformen.

*Ablauf.* Teilnehmer\*innen testeten jeweils zuerst den einen, und dann den anderen Redirection Algorithmus mit den unterschiedlich verfügbaren physischen Platz. In der ersten Iteration wurde jeweils einer der beiden Algorithmen mit den unterschiedlich großen Spielflächen getestet und bewertet. Dabei wurde den Probanden\*innen nach dem dreimaligen durchlaufen des Spiels einzelne Fragen bezüglich Spielerlebnis gestellt. Nach der zweiten Iteration, hier mit dem zweiten Algorithmus, wurden erneut die gleichen Fragen beantwortet wie bei der ersten Iteration des Tests. Zusätzlich wurden noch vergleichende Fragen gestellt, welche von den beiden Algorithmen die Probanden\*innen als besser bzw. weniger störend oder weniger wahrnehmbar verspürten. Zu guter Letzt wurden noch die qualitativen Fragen bezüglich Spielerlebnis und Spielspaß beantwortet. Für die Analyses wurde von allen Probanden\*innen der Durchschnitt der pro Testfall einzeln beantworteten Fragen hergenommen und ausgewertet.

*Ergebnisse.* Basierend auf dem beigefügten Fragebogen wurden die Antwortmöglichkeiten wie folgt bewertet (Auswertung in Abbildung 13 dargestellt): Trifft nicht zu (1 Punkt), Trifft



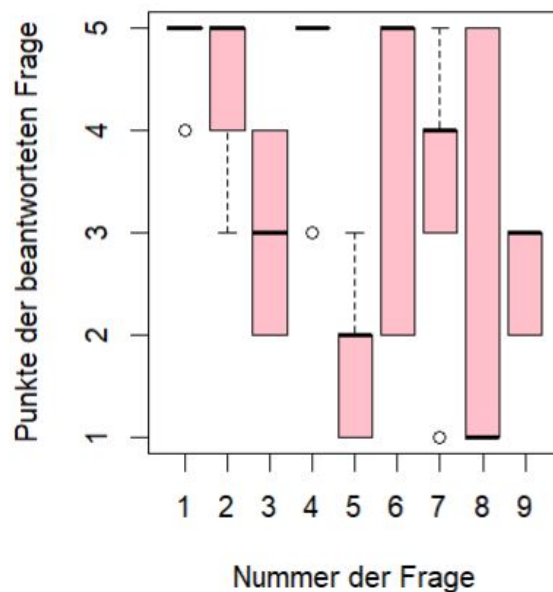


Abbildung 13: Boxplotauswertung der Fragen. Siehe Fragebogen zur Studie für Fragenzuordnung zu den einzelnen Nummern

eher nicht zu (2 Punkte), Teils/Teils (3 Punkte), Trifft eher zu (4 Punkte) und Trifft voll und ganz zu (5 Punkte). Die Punktzahlen aller Teilnehmer\*innen wurden addiert und daraus der Durchschnitt berechnet. Dies ergab folgende Ergebnisse für die jeweiligen Fragen: Fast alle Teilnehmer\*innen ( $\bar{x} = 4,8$ ) waren der Meinung, dass eine sich verkleinernde Spielfläche das Spielerlebnis mindert. Der Großteil der Probanden\*innen ( $\bar{x} = 4,4$ ) fühlte sich durch den STO-Algorithmus besser umgeleitet. Allerdings waren die Meinungen bezüglich der Orientierungslosigkeit sehr unterschiedlich ( $\bar{x} = 3$ ). Die Testpersonen waren jedoch nahezu einstimmig der Meinung, dass das Zurücksetzen störend und nervig war ( $\bar{x} = 4,6$ ). Ein positiver Aspekt von Redirected Walking ist, dass die Umleitung kaum spürbar war ( $\bar{x} = 1,8$ ). Bei der Frage, ob Redirected Walking spannender als traditionelle Fortbewegungsmethoden ist, waren die Meinungen wiederum unterschiedlich ( $\bar{x} = 3,8$ ). Dies lag daran, dass nicht alle Testpersonen Erfahrung mit Virtual Reality hatten und daher noch wenig bis gar keine Erfahrung mit anderen Fortbewegungsmethoden hatten. Auch beim Spielspaß waren die Meinungen geteilt ( $\bar{x} = 3,4$ ), da es schwierig war, das Spiel hinsichtlich der unterschiedlichen Spielflächen auszugleichen. Ein großer Minuspunkt von Redirected Walking war jedoch, dass die Probanden\*innen wenig Vertrauen in das System hatten ( $\bar{x} = 2,6$ ) und insbesondere zu Beginn große Bedenken hatten, in ein Hindernis zu laufen.

*Diskussion.* Zu Beginn sollte betont werden, dass Redirected Walking sich besser bei Personen bewährt hat, die bereits Erfahrung mit VR haben. Da sie bereits mit der virtuellen Umgebung vertraut sind, gibt es beim ersten Eintauchen ins Spiel weniger Verwirrung und Orientierungslosigkeit. Probanden\*innen mit wenig bis keiner VR-Erfahrung mussten sich zunächst an das Headset gewöhnen und benötigten mehr Zeit um zu verarbeiten, was gerade passiert. Durch das

Redirecting kam es zusätzlich zu Verwirrung, da die virtuelle Welt nicht mit der realen Welt übereinstimmte und die natürlichen Bewegungen der Benutzer\*innen beeinflusste. Zudem hatten vier von fünf Testpersonen wenig Vertrauen in den „Guardian“, der in der virtuellen Welt dem\*der Nutzer\*in signalisiert, wenn er\*sie sich dem Ende der Spielfläche nähert. Es ist wichtig zu betonen, dass Redirected Walking besser bei Probanden\*innen funktioniert hat, die bereits Erfahrung mit VR hatten. Diese Personen waren vertraut mit der virtuellen Umgebung und hatten daher weniger Schwierigkeiten bei ihrem ersten Eintauchen ins Spiel. Hingegen benötigten Tester\*innen mit wenig bis gar keiner VR-Erfahrung mehr Zeit, um zu verstehen, was gerade passiert. Durch das Redirecting entstand zusätzliche Verwirrung, da die virtuelle Welt nicht mit der realen Welt übereinstimmte und die natürlichen Bewegungen der\*die Benutzer\*in beeinflusst wurden. Zudem hatten 4 von 5 Probanden\*innen wenig Vertrauen in den „Guardian“, der in der virtuellen Welt visuell anzeigt, wenn man sich dem Ende der Spielfläche nähert. Da die gewohnten Fortbewegungsmöglichkeiten wie das Teleportieren wegfielen, war die Angst, gegen eine Wand oder ein Hindernis zu laufen, größer. Dies ist darauf zurückzuführen, dass die Testpersonen zuvor keine Erfahrung mit Redirected Walking hatten und daher wenig Vertrauen in die Algorithmen hatten. Alle Probanden\*innen fühlten sich erst nach einer gewissen Zeit sicher und begannen dann, sich mit mehr Selbstvertrauen und schneller zu bewegen. Die Geschwindigkeit des Fortbewegens hing jedoch von der Größe des physisch verfügbaren Platzes ab. Die Tester\*innen empfanden das Spiel flüssiger und besser, wenn mehr Platz zur Verfügung stand. Der Resetter sorgte jedoch anfangs bei einigen Probanden\*innen für Verwirrung, da man stehen bleiben und sich im Kreis drehen musste, um nicht aus der Spielfläche herauszulaufen. Außerdem empfanden die Testpersonen das Resetten als störend und immersionsbrechend, insbesondere bei einer Fläche von 5x5 Metern und 2,5x2,5 Metern. Nur ein\*eine Proband\*in litt unter Übelkeit aufgrund des ständigen Drehens und Zurücksetzens. In Bezug auf die Redirecting-Algorithmen empfanden alle Testpersonen den Steer-To-Orbit-Algorithmus besser geeignet als den Steer-To-Center-Algorithmus. Dies lag hauptsächlich daran, dass man durch STC immer von einer Ecke der Spielfläche über die Mitte zum nächsten Eck gelaufen ist und man auch tatsächlich immer wusste, wo man sich befand. Ein weiterer Nachteil von STC war, dass der verfügbare Platz nicht optimal ausgenutzt wurde, da die Testpersonen immer zur Mitte geleitet wurden (siehe Abbildung 14). Im Gegensatz dazu verloren die Probanden\*innen bei STO die Orientierung, wo sie sich tatsächlich im Raum befanden, was die Immersion jedoch erhöhte. Darüber hinaus konnte der Platz in der Spielfläche besser genutzt werden, da sich die Tester\*innen am Rand des verfügbaren Platzes aufhielten und nicht immer in die Mitte zurückgeführt wurden (siehe Abbildung 15). Beide Algorithmen nutzten jedoch unterschwellige Kameranpassungen, um die Probanden\*innen umzuleiten, was den meisten nicht bewusst war. Insbesondere bei STO bemerkte keiner der Teilnehmer, dass er\*sie sich in einer Kurve um das Zentrum der Spielfläche bewegte. Ein großer Nachteil der Anwendung war, dass der Spieler bei Anwendungsbeginn nicht immer in der Mitte des Spielfelds gespawnt wurde, obwohl er\*sie in der realen Welt mittig stand. Dadurch stimmten die virtuelle und reale Spielfläche nicht überein, was zu Fehlern beim Redirecten führte. Beispielsweise befanden sich die Probanden\*innen in der Realität bereits am Ende des Spielfelds, während sie im Spiel noch einige Meter davon entfernt waren und daher nicht umgeleitet werden konnten. Die Probanden empfanden das Spiel als erfrischend, da sie dank Redirected Walking VR in eine völlig neue Umgebung eintauchen konnten. Dadurch war der Spielspaß hoch, da sie tatsächlich selbst zu den Wegpunk-

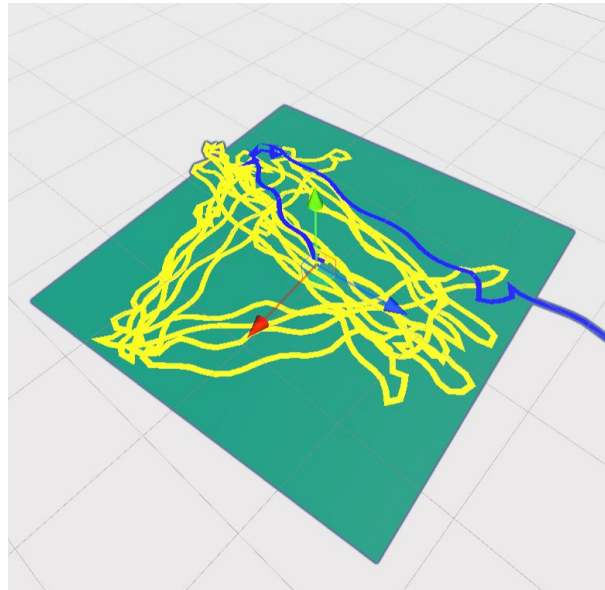


Abbildung 14: Weg (in Gelb) der in echt bei STC in der physisch verfügbaren Fläche zurückgelegt wurde

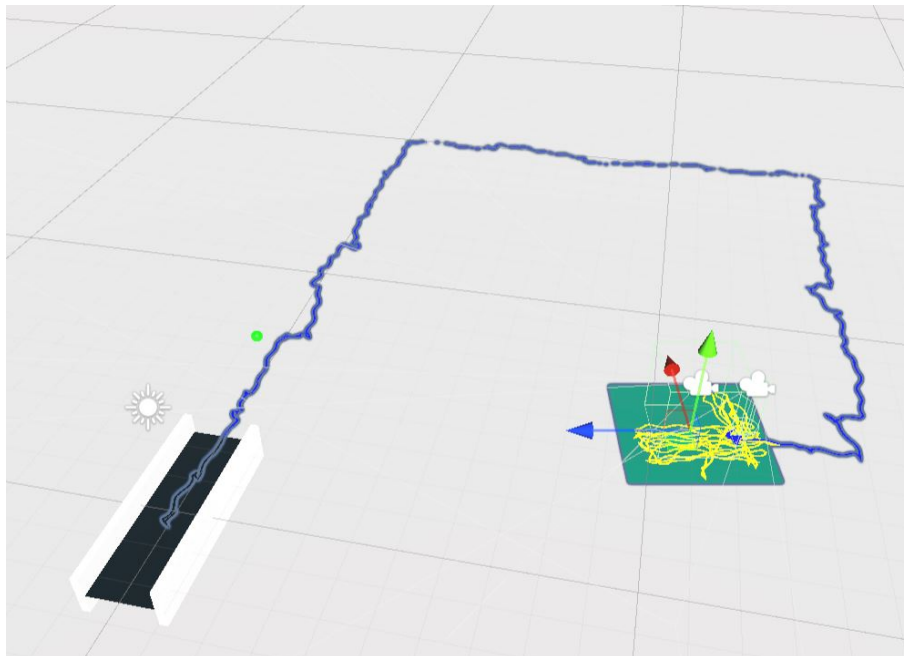


Abbildung 15: Weg (in Gelb) der bei STO in echt in der physisch verfügbaren Fläche zurückgelegt wurde

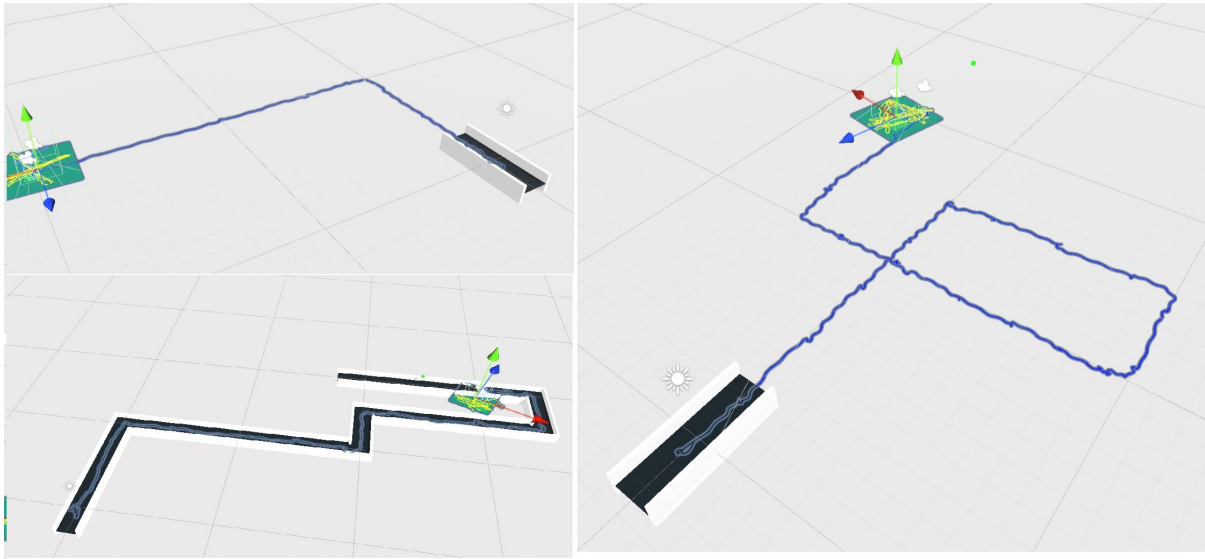


Abbildung 16: Unterschiedliche Wege, welche die Probanden\*innen zurückgelegt haben

ten laufen konnten. Zudem sorgte die zufällig generierte Karte für Abwechslung, sodass auch bei wiederholtem Testen keine Langeweile aufkam (siehe Abbildung 16). Zusammenfassend gaben die Probanden\*innen an, dass Redirected Walking eine faszinierende Technologie sei, um sich in der virtuellen Welt zu bewegen. Die Wirksamkeit dieser Technologie hängt jedoch stark von der Art der Anwendung oder des Spiels ab und ist nicht in jedem Fall besser als die Bewegung mittels Teleportation oder Joystick. Einige Probanden\*innen fanden Redirected Walking jedoch spannender und immersiver als andere Fortbewegungsmöglichkeiten und schlugen vor, dass visuelle Effekte verwendet werden könnten, um das Redirecten noch immersiver zu gestalten.

## 7 Zusammenfassung

Redirected Walking ist eine vielversprechende Technologie, die es ermöglicht, größere virtuelle Umgebungen zu erkunden, als dies bisher mit traditionellen Fortbewegungsmöglichkeiten in der Virtual Reality (VR) möglich war. Es gibt jedoch auch einige Limitierungen und Herausforderungen, die es derzeit schwierig machen, Redirected Walking als primäre Fortbewegungsmethode in VR zu verwenden. Ein Hauptproblem von diesem System ist, dass es eine begrenzte Bewegungsfläche gibt, innerhalb der sich der\*die Benutzer\*in bewegen kann, ohne dass er\*sie den Effekt bemerkt. Dies ist aufgrund der begrenzten Größe der meisten physischen Räume, in denen VR-Systeme normalerweise eingesetzt werden, ein großes Problem. Die Möglichkeit, diese Bewegungsfläche zu erweitern, indem man sie künstlich vergrößert, ist schwierig, da dies den Effekt von Redirected Walking beeinträchtigen kann und die Testperson anfälliger für Simulationserkennung machen kann. Zusätzlich mindert das dauerhafte Resetten auf kleinen Flächen die Immersion und hat negative Auswirkungen auf den Spielfluss. Ein weiteres Problem ist,

dass nicht alle Spielenden die gleiche Empfindlichkeit gegenüber dem Umleiten haben. Einige Benutzer\*innen können den Effekt leichter bemerken als andere, was dazu führen kann, dass der Effekt nicht immer zuverlässig ist und nicht bei allen Spielern gleich funktioniert. Dies kann zu einer inkonsistenten Erfahrung führen, was ein Problem bei der Verwendung von Redirected Walking als primäre Fortbewegungsmethode in VR darstellt. Derzeit ist es wahrscheinlicher, dass Redirected Walking in Kombination mit anderen Fortbewegungsmöglichkeiten wie Teleportation oder Joy-Stick Movement eingesetzt wird, um ein möglichst umfassendes VR-Erlebnis zu bieten. Weitere Herausforderungen, die dieses System noch überwinden muss, sind insbesondere der verfügbare Platz und die möglichen Gefahren, die mit der Bewegung in der realen Welt verbunden sind. Es gibt auch Fragen hinsichtlich der Akzeptanz und des Komforts der Benutzer bei der Verwendung von Redirected Walking-Technologie, insbesondere bei Personen mit Gleichgewichtsstörungen oder Kinetose (Reisekrankheit). Redirected Walking hat das Potenzial, die Art und Weise, wie wir uns in der virtuellen Welt bewegen, zu verändern und zu verbessern. Es kann dazu beitragen, dass VR-Erfahrungen noch immersiver und realistischer werden, indem es die Grenzen des physisch verfügbaren Platzes erweitert und es den Benutzern ermöglicht, sich frei und natürlich in der virtuellen Welt zu bewegen. Trotz dieser Herausforderungen gibt es viele Möglichkeiten, die Technologie zu verbessern und weiterzuentwickeln. Eine Möglichkeit besteht darin, die Algorithmen für das Redirecting zu verbessern, um eine noch nahtlosere Erfahrung zu schaffen. Eine andere Möglichkeit besteht darin, neue Technologien wie 360-Grad-Laufbänder oder haptisches Feedback zu integrieren, um die Immersion zu erhöhen und das Erlebnis noch realistischer zu gestalten. Eine mögliche zukünftige Anwendung von Redirected Walking könnte in der Unterhaltungsindustrie liegen, insbesondere in Spielen oder Freizeitparks. Es könnte auch in Bereichen wie Training und Bildung eingesetzt werden, um eine realistischere Umgebung für Lernende zu schaffen. Insgesamt ist Redirected Walking ein aufregender Bereich der VR-Technologie, der in Zukunft weiter erforscht werden wird. Es gibt jedoch noch viel Forschungsbedarf, um die Herausforderungen zu lösen, die dieses neue System mit sich bringt. Bis dahin ist es unwahrscheinlich, dass es die traditionellen Fortbewegungsmethoden in VR vollständig ablösen wird.

## Literaturverzeichnis

- Azmandian, Mahdi, Timofey Grechkin, Mark Bolas und Evan Suma. 2016. »The redirected walking toolkit: a unified development platform for exploring large virtual environments«. In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, 9–14. IEEE.
- Azmandian, Mahdi, Rhys Yahata, Timofey Grechkin, Jerald Thomas und Evan Suma Rosenberg. 2022. »Validating simulation-based evaluation of redirected walking systems«. *IEEE Transactions on Visualization and Computer Graphics* 28 (5): 2288–2298.
- Boletsis, Costas, und Jarl Erik Cedergren. 2019. »VR locomotion in the new era of virtual reality: an empirical comparison of prevalent techniques«. *Advances in Human-Computer Interaction* 2019.
- Bowman, Doug A, und Ryan P McMahan. 2007. »Virtual reality: how much immersion is enough?«. *Computer* 40 (7): 36–43.
- Cardoso, Jorge CS, und André Perrotta. 2019. »A survey of real locomotion techniques for immersive virtual reality applications on head-mounted displays«. *Computers & Graphics* 85:55–73.
- Darken, Rudolph P, William R Cockayne und David Carmein. 1997. »The omni-directional treadmill: a locomotion device for virtual worlds«. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, 213–221.
- Hamad, Ayah, und Bochen Jia. 2022. »How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations«. *International Journal of Environmental Research and Public Health* 19 (18): 11278.
- Hodgson, Eric, Eric Bachmann und David Waller. 2008. »Redirected walking to explore virtual environments: Assessing the potential for spatial interference«. *ACM Transactions on Applied Perception (TAP)* 8 (4): 1–22.
- Kim, Hyun K, Jaehyun Park, Yeongcheol Choi und Mungyeong Choe. 2018. »Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment«. *Applied ergonomics* 69:66–73.
- Kuang, Yang, und XiaoMei Bai. 2018. »The research of virtual reality scene modeling based on unity 3D«. In *2018 13th International Conference on Computer Science & Education (ICCSE)*, 1–3. IEEE.
- Langbehn, Eike, Paul Lubos, Gerd Bruder und Frank Steinicke. 2017. »Bending the curve: Sensitivity to bending of curved paths and application in room-scale vr«. *IEEE transactions on visualization and computer graphics* 23 (4): 1389–1398.
- Langbehn, Eike, Paul Lubos und Frank Steinicke. 2018. »Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking«. In *Proceedings of the Virtual Reality International Conference-Laval Virtual*, 1–9.

- Lee, Juyoung, Sang Chul Ahn und Jae-In Hwang. 2018. »A walking-in-place method for virtual reality using position and orientation tracking«. *Sensors* 18 (9): 2832.
- Nilsson, Niels Christian, Stefania Serafin, Frank Steinicke und Rolf Nordahl. 2018. »Natural walking in virtual reality: A review«. *Computers in Entertainment (CIE)* 16 (2): 1–22.
- Saker, Michael, und Jordan Frith. 2020. »Coextensive space: virtual reality and the developing relationship between the body, the digital and physical space«. *Media, Culture & Society* 42 (7-8): 1427–1442.
- Singh, Ravi Pratap, Mohd Javaid, Ravinder Kataria, Mohit Tyagi, Abid Haleem und Rajiv Suman. 2020. »Significant applications of virtual reality for COVID-19 pandemic«. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews* 14 (4): 661–664.
- Steinicke, Frank, Gerd Bruder, Jason Jerald, Harald Frenz und Markus Lappe. 2009. »Estimation of detection thresholds for redirected walking techniques«. *IEEE transactions on visualization and computer graphics* 16 (1): 17–27.
- Steinicke, Frank, Gerd Bruder, Timo Ropinski und Klaus Hinrichs. 2008. »Moving towards generally applicable redirected walking«. In *Proceedings of the Virtual Reality International Conference (VRIC)*, 15–24.
- Usoh, Martin, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater und Frederick P Brooks Jr. 1999. »Walking, walking-in-place, flying, in virtual environments«. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 359–364.

# Appendices

## A git-Repository

[https://gitlab.mediacube.at/fhs45915/bachelorarbeit\\_endless\\_runner](https://gitlab.mediacube.at/fhs45915/bachelorarbeit_endless_runner)

## B Vorlagen für Studienmaterial

### B.1 Fragebogen zur Studie





# FH Salzburg

## Bachelorarbeit zu Redirected Walking in VR

### Überblick:

Fragebogen für Propand\*innen der verschiedenen Redirected Walking Algorithmen basierent auf variierender Spielfläche

	Trifft nicht zu	Trifft eher nicht zu	Teils/Teils	Tifft eher zu	Trifft voll und ganz zu
<i>Umso kleiner die verfügbare Spielfläche wurde, um schlechter war die Erfahrung</i>					
<i>Der zweite Redirected Walking Algorithmus fühlte sich besser an</i>					
<i>Durch das dauerhafte Resetten trat Orientierungslosigkeit ein</i>					
<i>Ich empfand das Resetten als störend und nervig</i>					
<i>Ich habe deutlich gespürt, dass ich redirected worden bin</i>					

*Ich empfand Redirected  
Walking  
angenehmer/aufregender  
als sich mittels  
Teleportation oder  
andersweitig  
fortzubewegen*

*Ich hatte Spaß während  
des Spiels*

*Ich war auf Grund meiner  
VR-Vorerkenntnisse bei  
erstmaligem Betreten der  
VR-Welt überfordert*

*Ich habe Vertrauen in das  
System gehabt, und  
fühlte mich sicher zu  
keiner Zeit in eine Wand  
oder ein Hindernis zu  
laufen*


## **B.2 Datenerhebungseinwilligung**



FH Salzburg

# Datenschutzinformation und Einwilligungserklärung

## zur Erhebung personenbezogener Daten in Interviews

Sehr geehrte Teilnehmerin, sehr geehrter Teilnehmer!

Im Rahmen meines *Bachelor- bzw. Masterstudiums* Endless Runner: Redirected Walking Systeme in Virtual Reality

an der Fachhochschule Salzburg GmbH verfasse ich gerade meine *Bachelor- bzw. Masterarbeit*.

Im Rahmen dieser wissenschaftlichen Arbeit werden personenbezogene Daten erhoben und verarbeitet, deshalb finden Sie im Folgenden alle Informationen, zu deren Mitteilung ich als Verantwortliche/r gemäß der Datenschutz-Grundverordnung (DSGVO) verpflichtet bin.

Ihre Teilnahme am Interview erfolgt freiwillig und unentgeltlich. Sie können jederzeit ohne Angaben von Gründen ausscheiden. Ist die Datenverarbeitung nicht gewünscht, kann jedoch keine Teilnahme am Interview erfolgen.

### 1. Verantwortliche/r für die Datenverarbeitung

Interviewer/in, Verfasser/in der *Bachelor- oder Masterarbeit* und Verantwortliche/r für die Datenverarbeitung iSd Art 4 Z 7 DSGVO ist:

Name des/der Studierenden

Daniel Zirngast

Kontaktdaten

Email: fhs45915@fh-salzburg.ac.at

### Thema der Bachelor- oder Masterarbeit

Titel der wissenschaftlichen Arbeit:

Endless Runner: Redirected Walking System in Virtual Reality

Beschreibung (Kurzabstract):

Untersuchung des Spielekonzepts „Endless Runner“ in Virtual Reality mittels Redirected Walking als Fortbewegungsmittel.

## **2. Darstellung und Zweck der Datenverarbeitung**

Im Rahmen dieser wissenschaftlichen Arbeit werden personenbezogene Daten erhoben und verarbeitet, um Statistiken für die Arbeit zu generieren.

Die Daten werden Anonym und als Durchschnittsdaten aller Propanden\*innen in Tabellen ausgewertet und in der Arbeit veröffentlicht

## **3. Welche Daten werden erhoben bzw. verarbeitet?**

Im Rahmen dieser wissenschaftlichen Arbeit werden die im Interview mit Ihnen erhobenen Daten verarbeitet

- Alter
- Beruf
- VR Vorerfahrung

## **Rechtsgrundlage**

Die Rechtsgrundlage zur Verarbeitung dieser personenbezogenen Daten stellt Ihre Einwilligung nach Art. 6 Abs 1 lit a bzw. Art. 9 Abs 2 lit a DSGVO bei besonderen Kategorien personenbezogener Daten dar. Sie haben das Recht Ihre Einwilligung jederzeit zu widerrufen, was jedoch nicht die Rechtmäßigkeit der bis dahin erfolgten Verarbeitung berührt.

## **4. Übermittlungsempfänger und/oder Übermittlung an ein Drittland/an eine internationale Organisation**

Die personenbezogenen Daten werden nicht an Dritte weitergegeben.

## **5. Veröffentlichung und etwaige Publikationen**

Gemäß § 19 Abs 3 FHG sind positiv beurteilte Master- und Diplomarbeiten durch Übergabe an die Bibliothek der Fachhochschule zu veröffentlichen. Auch Bachelorarbeiten können veröffentlicht werden oder deren Inhalt für wissenschaftliche Publikationen genutzt werden.

Die Abschlussarbeit enthält lediglich anonymisierte Daten, wodurch bei einer Veröffentlichung oder Publikation jedenfalls keine Rückschlüsse auf einzelne Personen möglich sind.

## **Speicherdauer**

Sofern Ihre personenbezogenen Daten nicht nach der Erhebung anonymisiert werden, werden sie so lange gespeichert, wie sie zur Erfüllung des vorgesehenen Zwecks notwendig sind oder gesetzlich vorgesehen Aufbewahrungsfristen bestehen. Zum Nachweis der Einhaltung guter wissenschaftlicher Praxis können Forschungsdaten 10 Jahre aufbewahrt werden.

## **6. Betroffenenrechte**

Ihnen stehen in Bezug auf ihre personenbezogenen Daten folgende Rechte zu:

- Sie können Ihre **Einwilligung** jederzeit bei der oben angeführten Kontaktperson **widerrufen**.
- Recht auf **Auskunft, Berechtigung oder Löschung, Einschränkung** der Verarbeitung, **Datenübertragbarkeit**, welche Sie durch Kontaktaufnahme mit der oben angeführten Kontaktperson geltend machen können.
- Wenn Sie der Ansicht sind, dass Ihre personenbezogenen Daten nicht im Einklang mit dem bestehenden Datenschutzrecht verarbeitet werden, steht Ihnen weiters das **Recht auf Beschwerde** bei der österreichischen Datenschutzbehörde zu (Barichgasse 40-42, 1030 Wien, Telefon: +43 1 52 152-0, E-Mail: dsb@dsb.gv.at).

## 7. Einwilligungserklärung

Hiermit willige ich ein, dass die im Rahmen der Bachelor- oder Masterarbeit erhobenen personenbezogenen Daten durch die verantwortliche Person für die genannten wissenschaftlichen Zwecke in der beschriebenen Art und Weise verarbeitet werden dürfen. Sofern ich besondere Kategorien von personenbezogenen Daten angegeben habe, sind diese von der Einwilligungserklärung ausdrücklich umfasst.

**Hinweis:** Ihre Einwilligung ist freiwillig. Sie können die Einwilligung ablehnen, ohne dass Ihnen dadurch Nachteile entstehen. Sie haben das Recht Ihre Einwilligung jederzeit zu widerrufen, was jedoch nicht die Rechtmäßigkeit der bis dahin erfolgten Verarbeitung berührt.

---

Name der interviewten Person

---

Ort, Datum

---

Unterschrift