

MASTERARBEIT

Dynamische Impossible Spaces in VR - Mapping von virtuellen und realen Hindernissen

vorgelegt am 19. Juni 2025
Ben Meyer

Erstprüfer: Prof. Dr. Eike Langbehn
Zweitprüfer: Simon Dewert

**HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG**
Department Medientechnik
Finkenau 35
22081 Hamburg

Zusammenfassung

In dieser Arbeit wird ein Ansatz zum Mapping realer Hindernisse in virtuelle Umgebungen (VR) entwickelt und vorgestellt, bei dem physische Objekte mithilfe des Passthrough-Modus und des Raumsan-Features der Meta Quest 3 in Echtzeit erkannt und dynamisch auf virtuellen Hindernissen gemappt werden. Zur Verbesserung der Nutzererfahrung werden zusätzlich Impossible Spaces eingesetzt, die prozedural mithilfe einer graphenbasierter Tiefensuche generiert werden. Der Ansatz wurde in einer Nutzerstudie mit 16 Teilnehmenden untersucht, in der drei Szenarien verglichen wurden: keine Hindernisse, zufällige Platzierung und optimierte Hindernisplatzierung. Die Bewertung erfolgte anhand der Kriterien Präsenz, spatiale Präsenz, Immersion, Realismus, Involviertheit und Angstgefühl. Die Ergebnisse zeigen, dass die optimierte Hindernisplatzierung in allen Kategorien bessere Werte erzielt als die zufällige Platzierung. Im Vergleich zur Variante ohne Hindernisse schnitt die optimierte Platzierung in keiner Kategorie schlechter ab. Der Vergleich mit einer bestehenden Studie, die einen ähnlichen Ansatz ohne Impossible Spaces verfolgt, verdeutlicht zusätzlich die positiven Effekte der Impossible Spaces auf Präsenz, Immersion und spatiale Präsenz. Im Szenario mit zufällig platzierten Hindernissen fielen Realismus und Involviertheit hingegen deutlich ab. In allen anderen Fällen zeigen sich Impossible Spaces als Verbesserung.

Abstract

This Thesis presents and implements an approach for mapping real-world obstacles into virtual reality (VR) environments, in which physical objects are detected in real time using the Passthrough mode and room scanning feature of the Meta Quest 3 and dynamically mapped to virtual obstacles. To further enhance the user experience and enable large virtual worlds, Impossible Spaces are employed, which are procedurally generated using a graph-based depth-first search algorithm. The approach was evaluated in a user study with 16 participants, comparing three scenarios: optimized obstacle placement, random placement, and no obstacles. The evaluation was based on the categories of presence, spatial presence, immersion, realism, involvement, and fear. The results show that the optimized obstacle placement achieved better values than random placement in all categories. Compared to the scenario without obstacles, the optimized version did not perform worse in any category. A comparison with an existing study employing a similar approach without Impossible Spaces further highlights the positive effects of Impossible Spaces on presence, immersion, and spatial presence. In contrast, realism and involvement were rated noticeably lower in the scenario with randomly placed obstacles. In all other cases, Impossible Spaces proved to be a beneficial enhancement.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
1 Einleitung	1
1.1 Problemstellung	2
2 Definitionen	4
2.1 Begriffsdefinitionen im Kontext von VR	4
2.2 Graphentheorie	6
3 Hintergrund	8
3.1 Präsenz und Immersion	8
3.2 Locomotion in VR	9
3.3 Dynamische virtuelle Umgebungen	11
4 Implementierung	12
4.1 Prozess der Umsetzung	12
4.2 Implementierungsdetails	13
4.2.1 Scannen des Raumes	14
4.2.2 Erstellung des Spielfeldes	15
4.2.3 Erstellung des Graphen	18
4.2.4 Raumgenerierung	20
4.2.5 Generierung der Objekte	21
4.3 Spieldesign	25
5 Studie	29
5.1 Studiendesign	29
5.2 Durchführung	31
6 Ergebnisse und Diskussion	32
6.1 Statistische Auswertung	33
6.2 Diskussion	35
6.2.1 Interpretation	35
6.2.2 Vergleich	37

7 Limitationen	41
8 Ausblick & Fazit	43
Literatur	45

Abbildungsverzeichnis

2.1	Beispiel eines Impossible Spaces bei dem zwei Räume (rot,grün) sich gegenseitig überlappen und durch einen Flur miteinander verbunden sind.	5
2.2	Ein Graph mit 6 Knoten, der einen Kreis enthält. Die roten Kanten markieren den Kreis.	7
2.3	Beispiel einer Tiefensuche	7
4.1	Screenshot aus Unity des ersten Prototypen, in rot alle Kanten des Graphen und in grün und gelb zwei gefundene Kreise	13
4.2	Grafische Übersicht des Anwendungsablaufs	13
4.3	Screenshot aus der Anwendung mit instanziierten Würfeln in gelb in jeder Ecke des Raumes	14
4.4	Zeichnung eines möglichen Raumscannes mit Wandeinkerbung und Möbelstück in einer Ecke	16
4.5	Darstellung des Prozesses der automatischen Hinderniserfassung	17
4.6	Screenshot der erfassten Hindernisse in rot, <i>Border</i> Platten in weiß und freie Platten in blau	18
4.7	Screenshot des Graphens aus der Anwendung mit grünen Kanten und roten Knoten	19
4.8	Übersicht aller Wände, aus welchen die Umgebung generiert wird	22
4.9	Übersicht aller Hindernisse in steigender Größe von links nach rechts 1×1 , 2×2 und 2×3	23
4.10	Übersicht der Wände und Türen innerhalb der Räume	24
4.11	Beispiel eines Tutorial-Hinweises in der Anwendung	26
4.12	Screenshot aus der Sicht des Spielers während der Anwendung	28
5.1	Aufnahme des Raumes in welcher die Studie durchgeführt wurde. Einige Stühle sind als Hindernis platziert.	30
6.1	Boxplot der Ergebnisse des Fragebogens aller drei Testszenarien	34
6.3	Boxplot der Ergebnisse der Studie von Liu et al. [Liu et al., 2021]	37
6.2	Screenshot aus der Anwendung welche in der Studie von Liu et al. verwendet wurde [Liu et al., 2021]	38

1 Einleitung

Virtual Reality (VR) bezeichnet digitale künstliche Welten, welche durch die Nutzung einer Virtual Reality Brille (VR- Brille) ein immersives, interaktives und reales Erlebnis ermöglichen. VR- Brillen finden zunehmend Anwendung in verschiedensten Bereichen, sei es in Trainingsanwendungen, wie dem Überwinden der Angst vor dem Präsentieren [Bachmann et al., 2023, Barnard, 2024] oder für soziale Events beispielsweise im Metaverse [Design4Real, 2024]. Die durch VR geschaffene hohe Präsenz und Immersion verbessert die Effektivität von therapeutischen Trainingsanwendungen deutlich [Kiper et al., 2022]. Den Hauptteil der Anwendung von VR befindet sich weiterhin im Gaming-Sektor. Fast 70% aller VR- Nutzer verwenden VR- Brillen für Videospiele. [Elad, 2024] Aufgrund der immersiven Erfahrung, die eine VR-Brille ermöglicht, bietet sich dies an.

Ein großer Aspekt, der zur Immersion beiträgt, ist das Bewegen des virtuellen Avatars durch Fortbewegen in der realen Welt. Ein einschränkender Faktor ist hierbei jedoch der Platz, der in der realen Welt benötigt wird, um sich fortzubewegen. Selbst in großen Räumen/Wohnzimmern befinden sich meist viele Hindernisse, wie Sofa, Esstisch, Fernseher und Stühle. All diese Hindernisse beeinträchtigen den verfügbaren Raum für den Nutzer und können somit das immersive Erlebnis stark einschränken und die Erfahrung verschlechtern. In dieser Masterarbeit adressiere ich dieses Problem und widme mich der Fragestellung, ob sich dieses Problem verringern oder sogar beheben lässt. Dafür wird eine spielerische Anwendung entwickelt, welche eine virtuelle Welt erschafft, mit den gleichen Dimensionen, wie der Raum, in dem sich der Nutzer befindet. An den Hindernissen der realen Welt werden Hindernisse in der virtuellen Welt an denselben Positionen platziert. Ein Stuhl in der realen Welt wird beispielsweise zu einer Kiste in der virtuellen Welt. Durch das dynamische Anpassen der virtuellen Umgebung an die reale soll die immersive Erfahrung des Nutzers verbessert werden, ohne Angst zu haben, mit Hindernissen zu kollidieren. Der weiterhin begrenzte physische Raum wird darüber hinaus durch die Verwendung von sogenannten Impossible Spaces (2.1.1) vermeintlich vergrößert. Durch diese wird die virtuelle Umgebung für den Nutzer auf dem noch immer gleichem physisch verfügbaren Raum größer. Die Anwendung funktioniert automatisch und verwendet prozedurale Generierung (2.1.8) für mehr Abwechslung und einen hohen Wiederspielwert. Ziel dieser Arbeit ist es, zu beantworten, ob sich das Spielerlebnis aufgrund dieser Anpassungen verbessert. Zur Beantwortung dieser Frage wurde eine Studie mit 16 Personen durchgeführt und folgende Hypothese im Rahmen dessen beantwortet:

Die Nutzung dynamischer Impossible Spaces mit passendem Mapping realer und virtueller

Hindernisse erhöht Präsenz, Realismus, Involviertheit, spatiale Präsenz und reduziert Angst im Vergleich zu nicht abgestimmten oder fehlenden Hindernissen.

Im Folgenden werde ich zunächst die Problemstellung genauer definieren und auf einige bereits veröffentlichte Ergebnisse bezüglich desselben Problems eingehen. Anschließend werde ich die Implementierung der Anwendung näher beschreiben und ihre Funktionsweise erläutern und darauf folgend das Studiendesign und die Methodik näherbringen. Die Ergebnisse werden anschließend analysiert und ausgewertet. Schlussendlich werde ich die Limitationen der Anwendung näher beschreiben und einen Ausblick geben.

1.1 Problemstellung

Virtual Reality Anwendungen und vor allem Spiele nutzen meistens als Fortbewegungsmethode *continues movement* mittels des Controller-Steuerung oder Teleportation. Beide Methoden sind dabei nicht auf den limitierten Platz eines realen Raumes beschränkt, wodurch sich große virtuelle Umgebungen erkunden lassen. Jedoch entsteht dadurch bereits ein Bruch in der Immersion des Nutzers, da eine unnatürliche Fortbewegung die Illusion, sich in einer virtuellen Welt zu bewegen, abschwächt. [De et al., 2024, Nilsson et al., 2018] Das Fortbewegen in der realen Welt, als natürlichste Form der Bewegung, verstärkt folglich diese Illusion und steigert das Gefühl der Präsenz. Gleichzeitig wird durch das natürliche Fortbewegen und das Vermeiden von Fortbewegungsmethoden wie beispielsweise Teleportation Motion Sickness vermieden. Jedoch zerstört eine Uneinigkeit zwischen der realen und der virtuellen Umgebung das Gefühl der Präsenz ebenso stark [Slater, 2009].

Für den Entwickler oder Level-Designer eines Spiels ist es unmöglich zu wissen, wie die reale Umgebung aussieht, insbesondere die Platzierung von Möbeln etc., mit denen der Spieler möglicherweise kollidieren könnte. Individuell angepasste Level-Layouts sind dementsprechend unmöglich, für alle möglichen Konfigurationen zu erstellen. Jedoch ist selbst bei einem individuell angepassten Layout der Platz der virtuellen Umgebung durch den physisch verfügbaren Raum begrenzt.

Um dieses Problem zu beheben, haben Liu et al. in ihrem Paper *Virtual reality game level layout design for real environment constraints* [Liu et al., 2021], einen Ansatz vorgestellt bei dem die virtuelle Umgebung aufgrund der realen Umgebung und der in dieser vorhandenen Hindernisse erstellt wird. Dadurch sind die Hindernisse der realen Welt, wie beispielsweise ein Sofa oder Tisch, ebenfalls Hindernisse bzw. nicht begehbare Objekte in der virtuellen Welt mit gleichem Maß. Ein Tisch in der realen Welt kann beispielsweise durch einen Stein in der virtuellen Welt ersetzt werden. Die immersive Erfahrung des Spielers wird dadurch nicht durch zufällige Kollisionen mit der realen Welt unterbrochen. Außerdem wird diese verstärkt, indem sich der Spieler durch reales Bewegen in der virtuellen Umgebung fortbewegen kann, ohne Unstimmigkeiten der virtuellen Welt mit der realen Welt zu bemerken.

Aufgrund des Ansatzes von Liu et al. entsprang die Idee der im Rahmen dieser Arbeit entwickelten Anwendung, mit einem ähnlichen Ziel, virtuelle Hindernisse dynamisch auf die Position von realen Hindernissen zu mappen. Darüber hinaus werden Impossible Spaces verwendet, welche in diesem Szenario viele Vorteile bringen: Zum einen wird der begrenzte Platz, welcher zur Verfügung steht, möglichst effizient genutzt, durch eine größere und explorative virtuelle Welt, ohne einen physisch größeren Raum.

Zum anderen ergeben sich vielfältige und komplexe Level-Design-Möglichkeiten für die Entwicklung von Anwendungen mit Impossible Spaces, während die Immersion erhalten bleibt. Impossible Spaces ermöglichen es, durch reales Fortbewegen größere virtuelle Räume und Welten zu erkunden. Die virtuelle Welt kann organischer erkundet werden und Spieler erlangen ein stärkeres Präsenzgefühl.

2 Definitionen

In diesem Kapitel werden für das weitere Verständnis der Arbeit wichtige Begriffe definiert. Diese sind unterteilt in Definitionen im direkten Bezug zu VR und Definitionen aus der Graphentheorie, welche große Relevanz in der Implementierung besitzen.

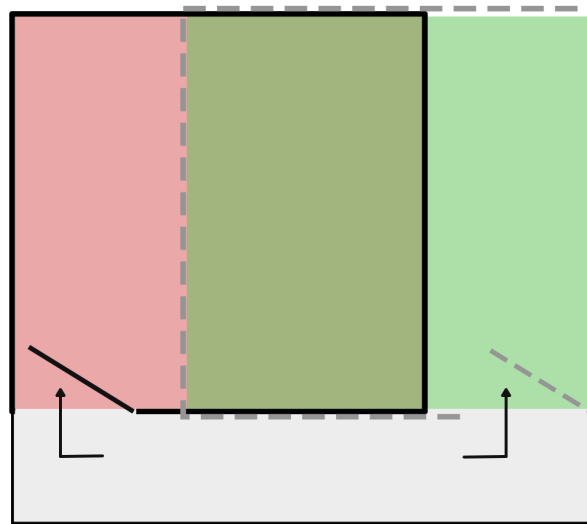
2.1 Begriffsdefinitionen im Kontext von VR

Definition 2.1.1 (Impossible Spaces). Impossible Spaces bezeichnet eine Designmechanik für virtuelle Umgebungen, bei welcher große virtuelle Räume und Umgebungen auf einem physisch kleineren Raum komprimiert werden. Diese Umgebungen werden realisiert durch selbst überlappende Architektur, wodurch zum Beispiel Räume, welche von außen kleiner als von innen sind, in der virtuellen Welt umsetzbar sind. Dies ist offensichtlich nicht real umsetzbar, jedoch in der virtuellen Welt möglich. In Abbildung 2.1 ist ein einfacher Fall eines Impossible Space dargestellt, bei dem zwei, sich zum Teil überlappende Räume, durch einen Flur verbunden sind.[Suma et al., 2012]

Definition 2.1.2 (Immersion). Immersion beschreibt den Grad, bis wohin es der Technologie möglich ist, dem Nutzer eine lebhaftere Illusion der Realität zu vermitteln. Limitiert ist die Immersion daher vor allem durch die technischen Möglichkeiten der VR-Brille. Dies betrifft das Tracking der Controller, die Auflösung des Bildschirms, die Grafikqualität und das Field of View. [Slater und Wilbur, 1997b]

Definition 2.1.3 (Präsenz). Das Gefühl der Präsenz im Kontext von VR wurde erstmals von Slater & Wilbur aufgegriffen [Slater und Wilbur, 1997a]. Im Bezug auf VR bezeichnet Präsenz das Konzept der „Illusion of nonmediation“. Mit anderen Worten, die VR-NutzerInnen neigen dazu, sich so zu verhalten, als wären sie in der realen Welt und nicht in der virtuellen [Bailenson, 2024]. Präsenz ist somit definiert als eine subjektive Annäherung, sich zu fühlen, als wäre der Nutzer wirklich an einem anderen Ort und nicht nur in der virtuellen Welt. Immersion kann das Gefühl der Präsenz hervorrufen.

Abbildung 2.1: Beispiel eines Impossible Spaces bei dem zwei Räume (rot,grün) sich gegenseitig überlappen und durch einen Flur miteinander verbunden sind.



Definition 2.1.4 (Spatial Presence). Spatial Presence (oder auch physical presence) bezeichnet die Dimension der Präsenz, welche das Gefühl des Nutzers beschreibt, in der virtuellen Umgebung physisch da zu sein und die Möglichkeit, mit der virtuellen Umgebung zu interagieren, als wäre diese real.

Definition 2.1.5 (Locomotion). Locomotion in Bezug auf VR bezeichnet die aktive Fortbewegung von einem Punkt in der virtuellen Umgebung zu einem anderen. Es gibt verschiedene Locomotion-Methoden. Die dabei gängigsten in VR und für diese Arbeit relevanten sind das *continues movement*, *real movement* oder auch *natural walking* und Teleportation. Beim *continues movement* bewegt sich der Nutzer kontinuierlich mithilfe des Controllers fort. *Natural walking* bezeichnet das Fortbewegen in der virtuellen Welt durch das natürliche Gehen in der realen Welt. Bei der Teleportation bewegt sich der Nutzer unmittelbar zu einem anderen virtuellen Ort und wird durch die Navigation des Controllers teleportiert. [Bozgeyikli, 2024]

Definition 2.1.6 (Room Scan). Room Scan bezeichnet in diesem Kontext das Erfassen eines Raumes sowie der darin enthaltenen Objekte wie Hindernisse, Fenster, Möbel und Wände. Die dabei gewonnenen Umgebungsdaten können anschließend algorithmisch weiterverarbeitet werden. In dieser Arbeit erfolgt der Room Scan mithilfe der Meta Quest 3, die dank eines integrierten Tiefensensors in Kombination mit zwei RGB-Kameras in der Lage ist, den Raum dreidimensional zu erfassen [Meta, 2015].

Definition 2.1.7 (Passthrough). Passthrough bezeichnet im Kontext von VR die Technologie, bei der die reale Umgebung mithilfe von Kameras des VR-Headsets in Echtzeit erfasst und innerhalb der VR-Brille als Videobild dargestellt wird. Durch den Passthrough-Modus lässt sich die reale Welt durch die VR-Brille erkennen, zusätzlich zu virtuellen Objekten in dieser.

Definition 2.1.8 (Prozedurale Generierung). Prozedurale Generierung bezeichnet in virtuellen Umgebungen eine Technik, bei der Spielwelten, Levelstrukturen oder Objektanordnungen algorithmisch erzeugt werden, häufig im Kontext des Game Designs. Ziel ist es, mit vergleichsweise geringem Entwicklungsaufwand eine hohe Vielfalt, Wiederverwendbarkeit und einen gesteigerten Wiederspielwert durch Randomisierung zu erreichen. Short und Adams, 2017.

2.2 Graphentheorie

Definition 2.2.1 (Graph). Ein (ungerichteter, einfacher) *Graph* $G = (V, E)$ besteht aus:

- einer endlichen oder abzählbaren *Knotenmenge* V ,
- und einer *Kantenmenge* E , die eine Teilmenge der zwei-elementigen Teilmengen von V ist.

Eine Kante $e \in E$ ist also ein ungeordnetes Paar $\{u, v\}$, $u \neq v$. [Diestel, 2025]

Definition 2.2.2 (Weg). Ein *Weg* ist ein nicht leerer Graph $P = (V, E)$ der Form:

$$V = \{x_0, x_1, \dots, x_k\}, \quad E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\},$$

wobei die x_i paarweise verschieden sind. Die Ecken x_0 und x_k sind die *Endecken* von P ; sie sind durch P verbunden. Die Ecken $x_1, \dots, x_{k-1}$ sind die *inneren Ecken* von P . Die Anzahl der Kanten eines Weges ist seine *Länge*; den Weg der Länge k bezeichnen wir mit P_k . [Diestel, 2025]

Definition 2.2.3 (Kreis). Ist $P = x_0 \dots x_{k-1}$ ein Weg und $k \geq 3$, so ist der Graph $C := P + x_{k-1}x_0$ ein *Kreis*. Die *Länge* eines Kreises entspricht der Anzahl seiner Kanten. In Abb. 2.2 ist ein Graph mit 6 Knoten zu erkennen, mit einem rot markierten Kreis der Länge 4. [Diestel, 2025]

Definition 2.2.4 (Tiefensuche). Die Tiefensuche oder auch DFS (depth first search) ist ein Algorithmus, welcher das Ziel verfolgt, alle erreichbaren Knoten zu besuchen. In Abbildung 2.3 ist ein Ablauf einer Tiefensuche dargestellt mit den Zahlen neben den Knoten in der Reihenfolge, in der diese entdeckt werden. Die Tiefensuche erfolgt rekursiv und besucht den von ihm zuletzt gefundenen Knoten und noch nicht besuchte Nachbarknoten. Dies wird so

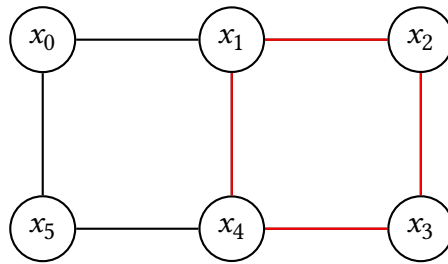


Abbildung 2.2: Ein Graph mit 6 Knoten, der einen Kreis enthält. Die roten Kanten markieren den Kreis.

oft wiederholt, bis jeder Knoten mindestens einmal besucht wurde. Die Tiefensuche besitzt eine Zeitkomplexität von $O(V + E)$, mit V der Anzahl der Knoten und E der Anzahl der Kanten. [Katja Mönius, 2021]

Für die Verwendung dieser Arbeit wird die Tiefensuche verwendet, um mögliche Kreise für die Implementierung der Impossible Spaces zu finden.

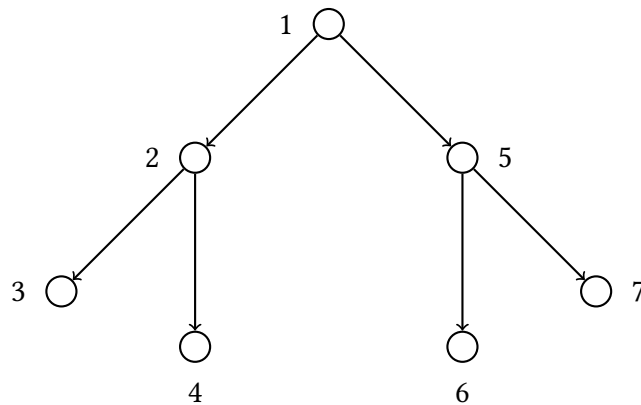


Abbildung 2.3: Beispiel einer Tiefensuche

3 Hintergrund

In diesem Kapitel werden Hintergründe zu den wichtigsten Bestandteilen der Arbeit erläutert und in den Kontext bisheriger Arbeiten gesetzt.

3.1 Präsenz und Immersion

Immersion und Präsenz sind herausstechende Merkmale in VR. Während auch in herkömmlichen Anwendungen oder Filmen und Spielen am Computer oder im Kino ein starkes Gefühl der Immersion und Präsenz geschaffen werden kann, geschieht dies in VR deutlich besser. In einer Übersichtsarbeit verglichen Hepperle et al. über 1000 Artikel im Bezug zur Immersion von Computerbildschirmen im Vergleich zu VR und der Realität [Hepperle und Wölfel, 2023]. Die Ergebnisse bestätigen die These, dass die Immersion und Präsenz in VR besser ist im Vergleich zu Computerbildschirmen. Besonders in Bezug auf die Interaktion und die Wahrnehmung der Umgebung schnitt VR deutlich besser ab. Einzige Nachteile, die sich zeigten, sind eine höhere kognitive Belastung und Motion Sickness, die in VR entstehen kann.

Die Motion Sickness ist häufig die Folge von schnellen Fortbewegungsmethoden, wie bereits im vorherigen Abschnitt beschrieben. Die erhöhte kognitive Belastung in VR ist vor allem Folge der oftmals komplizierten Menüs und User Interfaces. Eine möglichst simple und einfache Menüsteuerung kann die kognitive Belastung deutlich reduzieren [Sari et al., 2023].

Die Relevanz von Immersion zeigt sich in der direkten Verbindung zur Nutzungsqualität. Eine erhöhte Immersion verbessert die Erfahrung in vielen Aspekten, sei es beispielsweise das Erfahren einer Geschichte [Shin, 2018] oder das Besuchen eines Ortes in VR [Anaya-Sánchez et al., 2024]. Zur Immersion trägt insbesondere, wie bereits erwähnt, natürliches Gehen bei, wie es auch in dieser Arbeit verwendet wird [Lochner und Gain, 2021].

Um die Immersion und Präsenz quantifizieren zu können wurde der Igroup Presence Questionnaire (IPQ) [I-group.org-project, 2016] entwickelt. Dieser hat sich etabliert und wird in zahlreichen Studien zur Präsenz und Immersion in VR verwendet. Der von Liu et al. verwendete Fragebogen besteht zum größten Teils aus Fragen des IPQ Questionnaires [Liu et al., 2021]. Zur Vergleichbarkeit werden in der in dieser Arbeit durchgeführten Studie dieselben Fragen verwendet.

3.2 Locomotion in VR

Einen großen Beitrag zur Immersion und Präsenz leistet die Fortbewegungsmethode. Die Anwendung verwendet reales Fortbewegen als Locomotion-Methode, welche sich als große Schwierigkeit herausstellt, besonders in stark eingeschränkten realen Umgebungen. Für eine scheinbar unendlich große virtuelle Umgebung wurden bereits mehrere Methoden erprobt. Zum einen lässt sich dies mit externen Hardware-Lösungen, wie beispielsweise einem Omni One [Virtuix, 2025], welches als 360° Laufsystem funktioniert, lösen. Allerdings sind diese Methoden häufig sehr kostspielig und somit nicht sehr leicht verfügbar. Neben externen Hardware-Lösungen gibt es vor allem den Ansatz des Redirected Walking und impossible spaces oder beides in Kombination.

Redirected Walking (RDW) ist eine Technik, die es ermöglicht, die Limitationen von begrenztem Raum zu überwinden. Dafür bewegen sich die Nutzer real fort, jedoch werden die Nutzer dabei auf einen Pfad in der realen Welt geleitet, ohne dass diese dies bemerken. Unter anderem geschieht dies durch Manipulation der VR-Szene, welche der Nutzer sieht. Durch diese positioniert und orientiert sich der Nutzer unbewusst und bewegt sich beispielsweise entlang einer geraden Strecke in der Realität, während dieser in der virtuellen Welt einen Kreis entlangläuft. [Langbehn et al., 2018]

Diese Methode benötigt dennoch optimalerweise einen recht großen freien Bereich von ca. 6×6 m zum Bewegen für ein sehr gutes Ergebnis [Azmandian et al., 2015]. Eine freie Fläche in solcher Größe ist selten und den allerwenigsten VR-Nutzern verfügbar. Diese Methode werden wir daher in dieser Arbeit nicht näher betrachten. Stattdessen verwenden wir Impossible Spaces. Ein simpler Fall eines *Impossible Environment* mit zwei überlappenden Räumen wurde bereits in Abbildung 2.1 dargestellt. [Suma et al., 2012] Diese Möglichkeit der Überlappung von Räumen bietet die Option, den physischen realen Raum zu vergrößern, auch in Umgebungen mit weniger Platz, wie in einem Wohnzimmer oder ähnlichem 2.1.1. Damit die Überlappung dem Nutzer nicht auffällt, kann diese aber nicht beliebig groß sein. Evan A. Suma et al. führten eine Studie zum Überlappungsgrad durch und stellten fest, dass die Räume bis zu 56% miteinander bzw. mit der Umgebung überlappen können, bevor dies vom Nutzer bemerkt wird [Suma et al., 2012]. Für die Umsetzung der in dieser Arbeit verwendeten Methodik wird sich daher an dem von Evan A. Suma et al. ermittelten Überlappungsgrad orientiert. Die bereits in Abschnitt 1.1 erwähnten Vorteile von Impossible Spaces in Bezug auf Präsenz und Immersion wurden unter anderem von Lochner et al. in ihrem Paper *VR Natural Walking in Impossible Spaces* gezeigt. [Lochner und Gain, 2021] In einer Studie, welche verschiedene Fortbewegungstechniken in impossible environments verglich, konnte gezeigt werden, dass natürliches Fortbewegen in Impossible Spaces die Immersion steigert, motion sickness verringert und die allgemeine Nutzbarkeit der Anwendung verbessert. Selbst in kleineren Räumen, von nur etwa $7m^2$ konnten Impossible Spaces erschaffen werden, welche bis zu $100m^2$ große virtuelle Welten umfassen. [Rebelo et al., 2024] In dieser Arbeit wird der

reale Raum in der virtuellen Welt nicht um derart große Größenordnungen erweitert, jedoch zeigt dies die Möglichkeiten für Level-Designs selbst in kleinen Räumen.

Mithilfe der Verwendung von Impossible Spaces lässt sich folglich die virtuelle Umgebung künstlich vergrößern, ohne die Immersion des Nutzers zu beeinträchtigen. Jedoch bleibt weiterhin das Problem der Gefahr der Kollision mit Hindernissen in begehbaren Umgebung. Rebelo et al. stellten in ihrem Paper „Adaptive Virtual Environments in Small Physical Spaces“ [Rebelo et al., 2024] einen Ansatz vor, der versucht, dieses Problem einzugrenzen, durch ein dynamisch anpassbares Level-Layout aus verschiedenen Räumen, komprimiert in einen einzigen, durch die Verwendung von Impossible Spaces. Der dynamische Ansatz ist hier sehr interessant, da er eine einzigartige Anpassbarkeit ermöglicht. Dennoch ist auch hier weiterhin eine freie Grundfläche nötig. Valentini et al. stellten einen Ansatz zur Hindernisvermeidung vor, bei welchem zunächst ein 3D-Scan der Umgebung vorgenommen wurde und dieser anschließend verarbeitet und angepasst wurde, um die Hindernisse an die des 3D-Scans anzupassen. [Valentini et al., 2020] Obwohl diese Möglichkeit nicht auf einen leeren Raum eingeschränkt ist, erfordert diese jedoch immer einen Scan vorab als zusätzlichen Schritt. Eine Echtzeit-Hindernisvermeidung mithilfe eines Tiefensensors und Benachrichtigungen für den Nutzer zum Ausweichen, stellten Wozniak et al. vor [Wozniak et al., 2020]. Während eine Echtzeit-Hindernisvermeidung sehr interessant ist, funktioniert diese durch Benachrichtigungen für den Nutzer, welche die Immersion dennoch zu einem gewissen Grad stören und das Spielerlebnis beeinträchtigen. Einen dynamischen Ansatz, welcher außerdem viele Freiheiten zur Anpassung der virtuellen Welt bietet, stellten Liu et al. in ihrem Paper „Virtual reality game level layout design for real environment constraints“ [Liu et al., 2021] vor. Bei diesem wurden vor Beginn der Anwendung Hindernisse im Raum in VR markiert. Daraufhin wurde der Raum unterteilt und an den Positionen der Hindernisse in der realen Welt wurden auch Hindernisse in VR platziert. Dadurch sollte eine virtuelle Umgebung geschaffen werden, welche durch reales Fortbewegen als Locomotion-Methode zugänglich ist. Gleichzeitig braucht der Nutzer keine Angst zu haben, mit Hindernissen in der realen Welt zu kollidieren. Dafür wurde der reale verfügbare Raum in $50 \times 50\text{cm}$ große Kacheln unterteilt. Die Kacheln wurden entweder als begehbar oder als Hindernis markiert. Basierend darauf wurde dann das Layout für das Spiel erstellt. In einer Studie wurden 3 Anwendungen verglichen:

- Eine Anwendung ohne Hindernisse in VR und in der realen Welt
- Eine Anwendung bei der die virtuellen Hindernisse nicht mit denen der realen Welt übereinstimmen
- Eine Anwendung bei der die virtuellen Hindernisse mit der realen Welt übereinstimmen

Die Ergebnisse des optimierten Falls, bei dem die Hindernisse mit der realen Welt übereinstimmen, zeigten deutlich verbesserte Werte in allen gemessenen Bereichen (Presence,

Spatial Presence, Involvement, Experienced Realism und Fear). In dieser Arbeit wird ein sehr ähnlicher Ansatz verfolgt, mit einigen relevanten Unterschieden. Dennoch eignet sich die Ähnlichkeit zur Gegenüberstellung, weshalb die für diese Arbeit durchgeführte Studie mit den gleichen Fragebögen und denselben drei Fällen arbeitet.

3.3 Dynamische virtuelle Umgebungen

Für die dynamische Anpassung von virtuellen Umgebungen wird nahezu ausschließlich prozedurale Generierung verwendet 2.1.8. Außerhalb von VR findet sich prozedurale Generierung vor allem in Spielen wieder bei der Generierung von Straßen [Galín et al., 2010], Dungeons [Van der Linden et al., 2014], anderen repetitiven Strukturen oder ganzen Welten [Mojang Studios, 2011]. In Virtual Reality wird prozedurale Generierung für dieselben Zwecke genutzt, ob für die Generierung von Vegetation, Landschaften, Straßen oder ähnlichem. In dieser Arbeit wird prozedurale Generierung relevant durch die Anpassung der virtuellen Umgebung an die Realität. Bisherige Arbeiten in diesem Kontext sind unter anderem der dynamische Ansatz zur Levelgenerierung von Liu et al. [Liu et al., 2021] oder Ansätze, bei denen vorab der Raum mittels eines Handys oder eines anderen Gerätes gescannt wird. Dies setzten beispielsweise Sra et al. um, welche eine vielversprechende Anwendung präsentierten, bei der vorab der Raum mittels eines Handys gescannt wurde [Sra et al., 2016], sehr ähnlich zu dem Ansatz von Rebelo et al. [Rebelo et al., 2024]. Beide Ansätze verwenden jedoch einen benötigten vorab Scan, was mehr Zeit und Aufwand für den Nutzer bedeutet.

Diese Ansätze erfordern jedoch weiterhin entweder eine große freie Grundfläche, wie im Fall von Rebelo et al. oder aber es werden keine Impossible Spaces verwendet, wie im Fall von Liu et al. und Sra et al. [Rebelo et al., 2024, Liu et al., 2021, Short und Adams, 2017]. In dieser Arbeit wird die komplette Umgebung prozedural generiert, inklusive Impossible Spaces.

Ein großer Vorteil von prozeduraler Generierung liegt neben der Wiederverwendbarkeit im Bezug auf VR vor allem in der Skalierbarkeit. Eine Anwendung, bei der die Umgebung vollständig prozedural generiert wird, lässt sich in kleinen oder großen und möblierten oder vollkommen leerstehenden Räumen spielen. Jedoch bringt prozedurale Generierung auch Probleme mit sich. Aufgrund der automatischen algorithmischen Erstellung der Welt geht die Kontrolle über die Umgebung bei der Entwicklung verloren. Eine prozedurale Umgebung ohne Probleme erfordert viel Testen der Anwendung zum Erkennen der Probleme [Smelik et al., 2014, Silva et al., 2025].

4 Implementierung

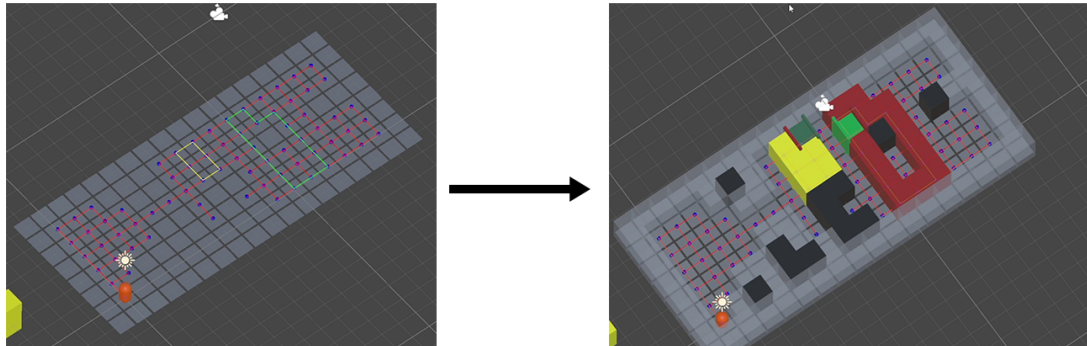
In diesem Kapitel wird die Implementation der Anwendung detailliert beschrieben und erläutert. Die Entwicklung geschah in Unity in der Version 6000.0.23f1 [Unity Technologies, 2024] für die Meta Quest 3.

4.1 Prozess der Umsetzung

Zur Analyse der Umsetzbarkeit und der weiteren Bestimmung der Anforderungen wurde zunächst ein Prototyp entwickelt, der zur Laufzeit den Unity Editor voraussetzte. Anschließend eine Standalone-Anwendung für die Quest. Ziel war es, das Grundkonzept zu erproben und auf mögliche Probleme aufmerksam zu werden. Dafür wurde ein Raum durch simple Rechtecke nachgebaut, welche als Knoten für den Graphen dienten. Von diesen wurden dann einige als Hindernisse markiert. Auf den übriggebliebenen Knoten wurde ein Graph gebildet, bei dem jeder Knoten zu jedem ihm direkt adjazenten Knoten eine Kante bildet, horizontal und vertikal. Mittels Tiefensuche wurden dann zwei Kreise gefunden, auf denen Quader zur Darstellung von Räumen platziert wurden, welche beim Betreten durch eine Tür in Richtung des Betretens größer wurden. Eine Abbildung dieses Prototypen ist in Abbildung 4.1 zu sehen. Mithilfe des Prototypes wurde die generelle Umsetzbarkeit der geplanten Methodik gezeigt. Im nächsten Schritt der Umsetzung fand die Implementierung der Anwendung in VR statt. Aufgrund mangelnder Verfügbarkeit der Quest 3 wurde zuerst mit der Quest 2 entwickelt. Jedoch verfügt diese noch über kein Room Scan Feature und lediglich einen schwarz-weiß Passthrough Modus über Infrarot. Daher wurde die Quest 2 zunächst nur für die korrekte Generierung der Umgebung verwendet. Später fand der Umstieg auf die Quest 3 statt. In diesem Stadium konnte mithilfe des Room Scan Features die dynamische und automatische Generierung der Welt und die Anwendung vollends umgesetzt werden.

Sobald eine Quest 3 verfügbar war, begann der Umbau bzw. die Anpassung an eine automatische Erkennung der Hindernisse und das Erfassen der Raumgröße. Anschließend wurde der Anwendung noch ein Sinn oder Spielziel gegeben, welches die Testpersonen beim späteren Testen der Anwendung verfolgen konnten.

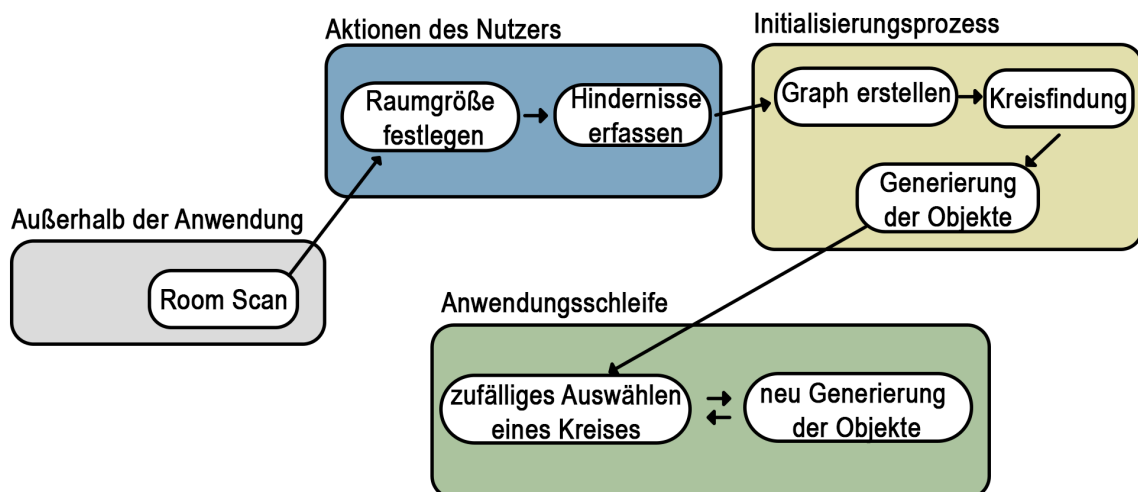
Abbildung 4.1: Screenshot aus Unity des ersten Prototypen, in rot alle Kanten des Graphen und in grün und gelb zwei gefundene Kreise



4.2 Implementierungsdetails

Nachdem im vorherigen Kapitel der Prozess der Umsetzung grob skizziert wurde, werden nun die einzelnen Schritte näher erläutert und darauf eingegangen, wie und mit welchen Methoden die Stufen der Entwicklung umgesetzt wurden. Eine grafische Übersicht der Schritte, auf die im Folgenden im Detail eingegangen wird, ist in Abbildung 4.2 zu sehen.

Abbildung 4.2: Grafische Übersicht des Anwendungsablaufs



4.2.1 Scannen des Raumes

Der erste Schritt der entwickelten Methodik erfordert die Erfassung der Raumgröße. Dafür wurde vorab der Raum über das Spatial Feature der Quest 3 außerhalb der Anwendung bereits gescannt. Der nächste Schritt besteht darin, die Ecken des Raumes für die spätere virtuelle Welt zu markieren. Diese markiert der Nutzer selbst dafür startet die Anwendung zunächst im Passthrough-Modus. Über einige Tutorial-hinweise, auf welche später in Kapitel 4.3 noch näher eingegangen wird, wird dem Nutzer erklärt, was dieser zu tun hat. Mit den vorhandenen Daten über den Raum werden alle Kanten des Raumes markiert und in den Ecken wird ein Würfel gespawnt. In Abbildung 4.3 ist ein Screenshot aus der Anwendung zu erkennen, bei dem Würfel in allen Ecken des Raumes instanziiert wurden.

Abbildung 4.3: Screenshot aus der Anwendung mit instanziierten Würfeln in gelb in jeder Ecke des Raumes



Die Funktionalität dessen findet in einem einzelnen Script statt, welches mehrere Funktionen hat. Zum einen ist es für die Erfassung der Raumgröße zuständig und zum anderen zum automatischen Erkennen der Hindernisse, wie Tische, Sofas und andere Möbel. Für die Umsetzung wurde das XR Interaction Toolkit Version 3.0.6 verwendet. Mit dessen Hilfe lassen sich die von der Quest erfassten Flächen und Kanten anzeigen lassen. Im nächsten Schritt werden die Würfel in den Ecken des Raumes platziert, weshalb zunächst nicht vertikal verlaufende Ebenen, wie Zimmerdecken oder Böden, ignoriert werden. Außerdem werden Flächen mit

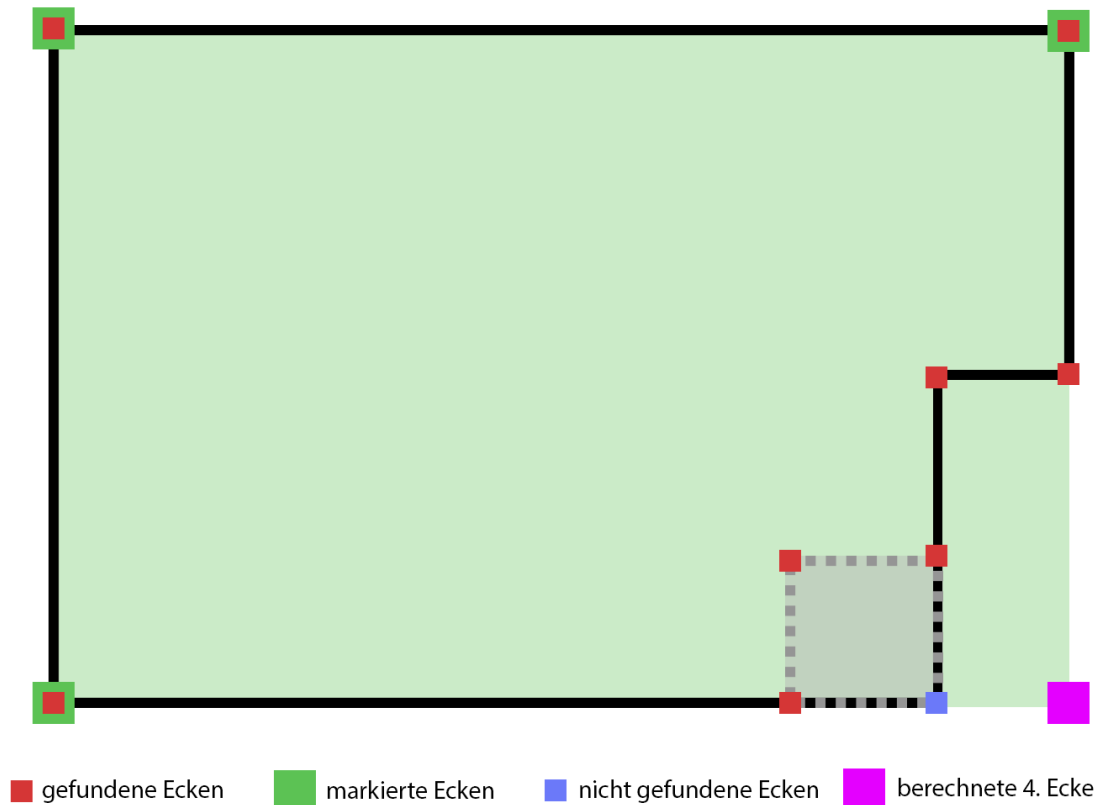
einem kleinen Umfang, in diesem Fall alle Flächen mit einem Umfang unter 3 Metern, ignoriert, da in diesem Schritt das gesamte Spielfeld als eine einzige Fläche erfasst werden soll. Anschließend wird in jeder Ecke ein Würfel gespawnt, welcher der Nutzer, wie zu Beginn beschrieben, entsprechend markiert. Um sicherzustellen, dass nur ein Würfel instanziiert wird, wurde ein Dictionary als Datenstruktur verwendet, um die bereits platzierten Würfel zu speichern.

Im Optimalfall besitzt der Raum 4 Ecken, welche für die Quest 3 gut erfassbar sind und somit erkannt werden können. Allerdings kann in einem möblierten Raum auch häufig ein Schrank oder Regal in einer Ecke stehen oder auch Einkerbungen in der Wand vorhanden sein, welche den Scan verunreinigen und folglich entweder eine zusätzliche Raumecke platziert oder diese nicht in der eigentlichen Ecke des Raumes platziert wird. Um dem entgegenzuwirken, ist der Nutzer dazu aufgefordert, 3 von 4 Raumecken auszuwählen. Die 4. Raumecke wird automatisch instanziiert, basierend auf der Position, wo diese für ein Rechteck sein müsste. In Abbildung 4.4 ist ein möglicher Raums캔, mit einem Möbelstück in einer Ecke und einer Einkerbung in der Wand dargestellt. Der Raum ist zum einen nicht rechteckig und zum anderen kann die eigentliche, in blau markierte Ecke, nicht gefunden werden. Der Nutzer kann in diesem Fall die in grün hinterlegten Ecken auswählen und markieren. Eine vierte, in der Zeichnung lilafarbene Ecke, wird an der berechneten Position für ein Rechteck instanziiert. Die Grundfläche des Raumes ragt somit über den eigentlichen Raum hinaus, die zusätzliche Fläche wird aber vom nachfolgenden Spielfeld ausgeschlossen, was näher in 4.2.2 erklärt wird. Vor dem Instanziiieren der 4. Raumecke und damit dem Festlegen der Begrenzungen des Raumes werden außerdem die Position jeder Ecke miteinander verglichen, um sicherzustellen, dass diese zu 100% auf denselben Koordinaten liegen. Aufgrund von Ungenauigkeiten im Scan oder auch leicht schrägen Wänden kann es zu leicht unterschiedlichen Positionen führen. Jedoch erfordert die hier entwickelte Anwendung einen strikten rechteckigen Raum, was in Abschnitt 4.2.3 zum Erstellen der Graphen deutlich wird. Liegt beispielsweise die erste Raumecke auf $(x = 1, y = 1)$ und die zweite Raumecke auf $(x = 1.1, y = 2)$, würden die für die weitere Verarbeitung platzierten Raumecken im Mittelwert auf $(x = 1.05, y = 1)$ und $(x = 1.05, y = 2)$ liegen. Ist dieser Prozess abgeschlossen, wird eine rechteckige Fläche des Raumes, in dem der Nutzer sich befindet, ermittelt. Dies ist die Grundfläche welche für die folgende Anwendung verwendet wird und auf der im nächsten Schritt ein Graph erstellt wird.

4.2.2 Erstellung des Spielfeldes

Bevor der Graph für die Platzierung der Hindernisse und der Räume erstellt werden kann, wird zunächst der Raum in Quadrate unterteilt, welche die Grundfläche des Spielfeldes darstellen werden. Die zuvor bestimmte Raumgröße dient hierbei als Maß. Der Raum wird in Quadrate von $30 \times 30\text{cm}$ Größe unterteilt. Eine kleinere Unterteilung hätte die Möglichkeit für

Abbildung 4.4: Zeichnung eines möglichen Raumscannes mit Wandeinkerbung und Möbelstück in einer Ecke

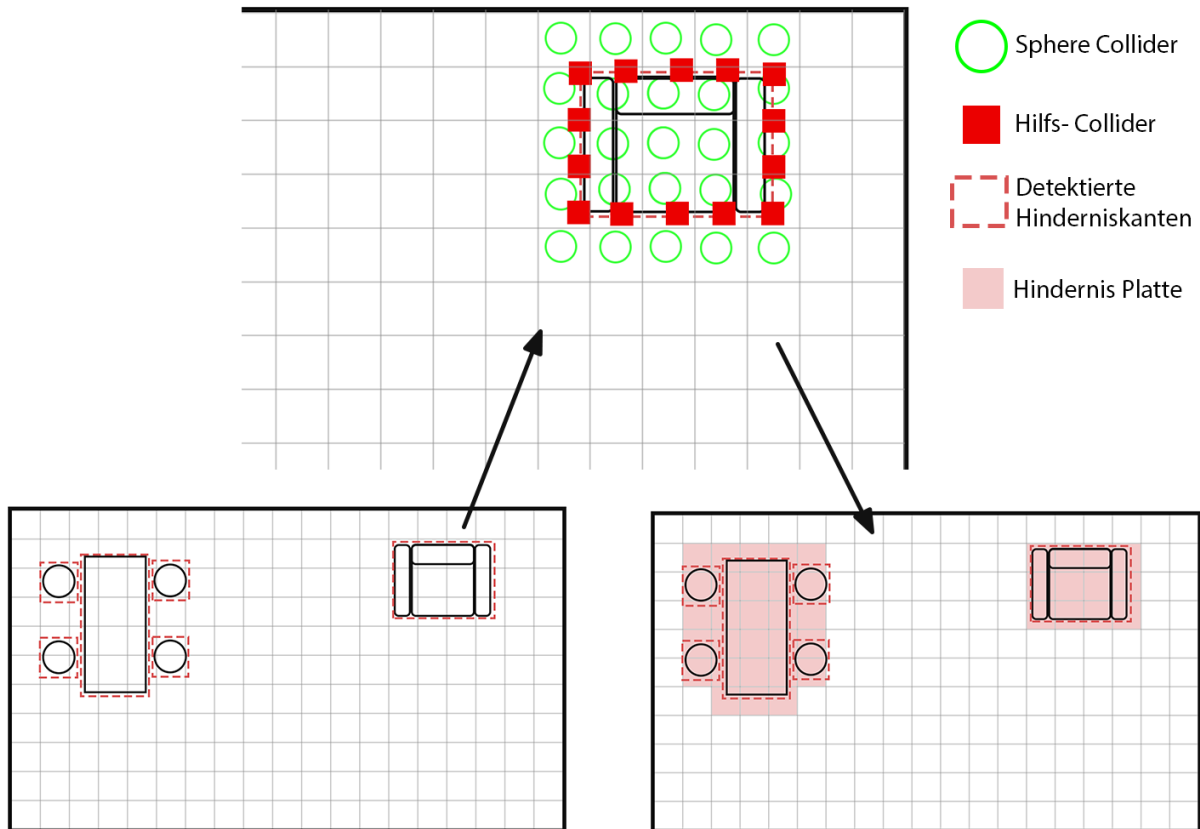


mehr Details gegeben, jedoch steigt bereits bei einer Größe von $30 \times 30\text{cm}$ die Rechenzeit bei größeren Räumen beträchtlich und kann zu Abstürzen führen. Die platzierten Platten bzw. Quadrate dienen gleichzeitig später als Boden des Spiels der Anwendung, weshalb jede Platte mit einer zufälligen Rotation instantiiert wird. Mittels *Mathf.FloorToInt* wird entschieden, wenn der Raum kein Vielfaches von 30cm ist, ob noch eine Platte platziert wird. Folglich gibt es eine Reihe an Platten, welche zum Teil in die Wand ragen. Diese werden im Verlauf als Hindernis markiert, sorgen jedoch für eine optisch etwas größere Umgebung.

Im nächsten Schritt, ebenfalls noch vor der Erstellung des Graphen, werden die Daten des Raumscans genutzt, um Hindernisse wie Möbel automatisch zu erkennen und zu markieren. Umgesetzt wird dies, indem zunächst nach den verfügbaren Möbeloptionen des XR Interaction Toolkit gefiltert wird. Diese umfassen Tisch, Stuhl, Sofa und Sitze jeglicher Art. Für alle gefundenen Objekte dieser Art werden entlang der Kanten der Möbel für den Nutzer unsichtbare Hilfsobjekte platziert, welche einen Box-Collider besitzen. Jede der zu Anfang platzierten quadratischen Platten besitzt darüber hinaus ebenfalls einen Sphere-Collider. Dieser deckt damit zwar nicht die gesamte Platte ab, ist jedoch performanter. Durch die

platzierten Hilfsobjekte wird auf eine Überlappung der Collider getestet. Wenn es zu einer kommt, wird die Platte, deren Collider überlappt, als Hindernis markiert. Im Anschluss werden alle Hilfsobjekte zerstört, da diese für den weiteren Prozess nicht mehr benötigt werden. Eine exemplarische Zeichnung der Hindernismarkierung ist in Abbildung 4.5 dargestellt.

Abbildung 4.5: Darstellung des Prozesses der automatischen Hinderniserfassung

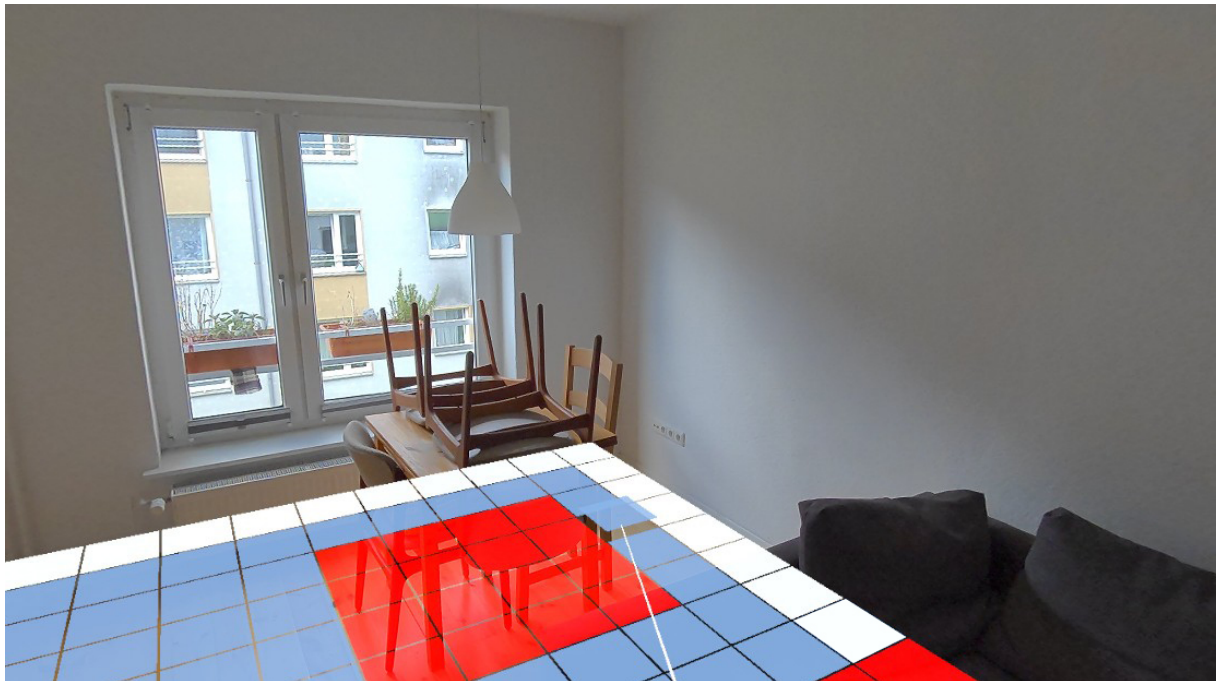


Die markierten Hindernisse sind für den Nutzer in einem rötlichen Ton sichtbar. Es kann jedoch häufig dazu kommen, dass einige Hindernisse vom Raumsan nicht erkannt wurden oder die Dimensionen nicht der Realität entsprechen. In diesem Fall kann der Nutzer manuell mittels des Controllers Platten als Hindernis markieren oder auch wieder als Bodenplatte markieren. Dafür wird von dem Controller ein Ray gecastet. Wenn dieser mit einer Platte kollidiert, erhebt sich diese leicht und es gibt eine geringe Vibration als haptisches Feedback. Mittels des rechten Triggers kann die Platte dann markiert werden.

Außerdem werden automatisch die äußersten Platten als *Border* in einer weißen Farbe markiert. Diese Platten werden für die spätere Generierung der Räume ausgeschlossen, aber es werden keine virtuellen Hindernisse auf diesen platziert und dienen damit als Abstandhalter, um zu garantieren, dass ein Minimum von mindestens 30cm bis zur nächsten Wand bzw.

Abstand zum Hindernis besteht und in den meisten Fällen 60cm oder mehr. Außerdem sorgen diese dafür, dass der Spieler garantiert jeden Punkt des Spielfeldes erreichen kann und dieser nicht, beispielsweise durch einen Raum durch die Mitte, zweigeteilt werden kann. In Abbildung 4.6 ist ein Screenshot aus der Anwendung zu sehen, bei der ein Tisch und ein Sofa mit der eben beschriebenen Methode als Hindernis erfasst wurden. Außerdem sind um das Spielfeld herum weiße *Border* Platten zu erkennen. Die von dem Spieler gerade ausgewählte Platte wird leicht angehoben als Hinweis für den Spieler, sollte dieser die ausgewählte Platte zu einer Hindernisplatte machen wollen.

Abbildung 4.6: Screenshot der erfassten Hindernisse in rot, *Border* Platten in weiß und freie Platten in blau



Als letztes wird manuell vom Spieler die *starting area* markiert. Diese muss irgendwo am Rande des Raumes platziert werden und nimmt 2×2 Kacheln in Anspruch. Durch diese wird der Spieler im Verlauf des Spiels laufen müssen, um einen neuen Raum bzw. eine neue Umgebung zu generieren.

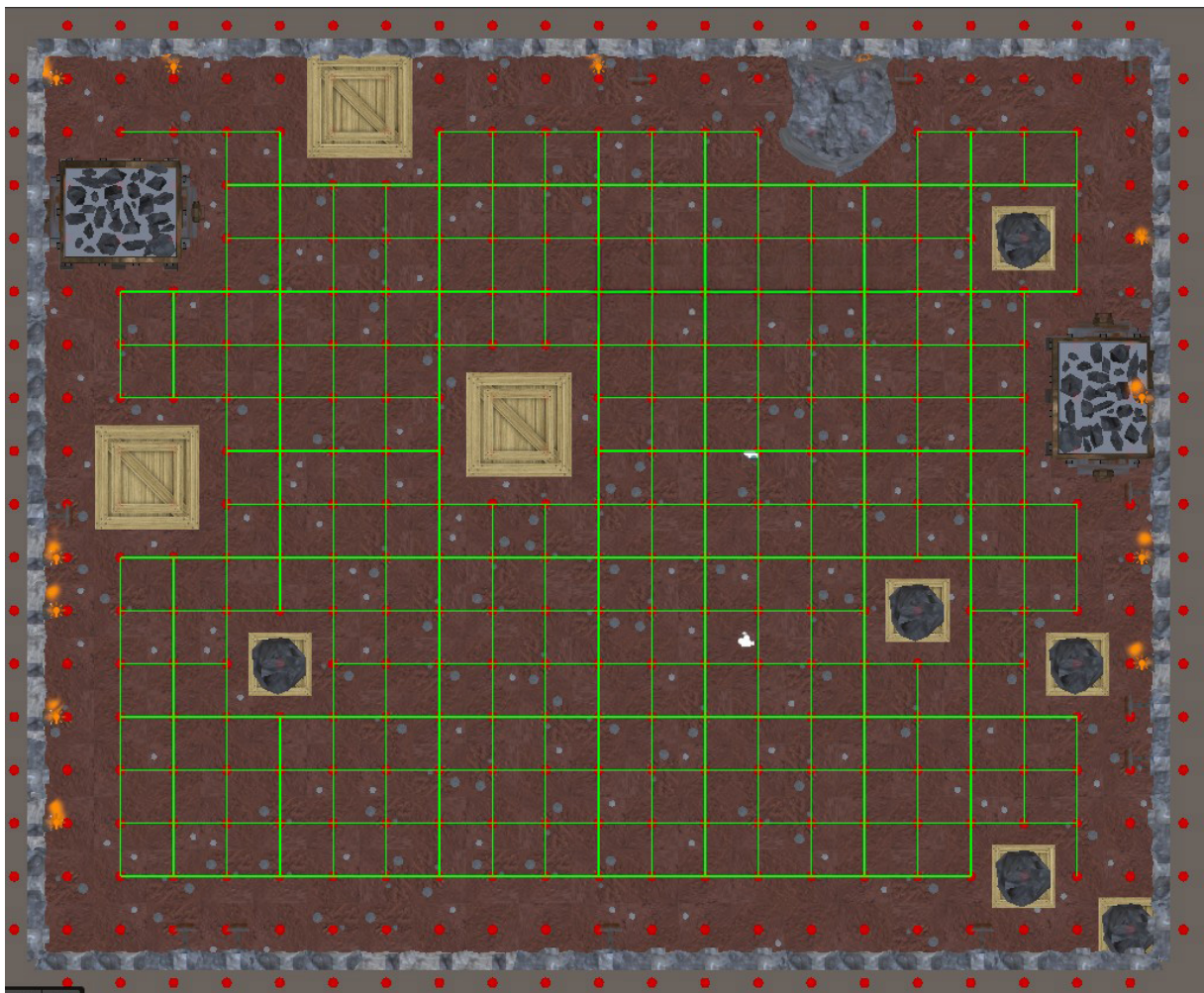
4.2.3 Erstellung des Graphen

Nachdem im vorherigen Abschnitt die Grenzen des Spielfeldes und alle Einschränkungen festgelegt wurden, wird im nächsten Schritt ein Graph erstellt. Für diesen Graphen werden die übriggebliebenen Bodenplatten, also alle Platten, welche nicht als Hindernis, *Border*

oder *starting area* markiert wurden, als Basis genutzt und werden als *Node* bzw. Knoten abgespeichert. Eine eigene Klasse speichert hierfür alle Informationen über die Platte. Dies beinhaltet den aktuellen Status der Platte, ob diese *Ground*, *Obstacle*, *CyclePart*, *StartPlate* ist, sowie die Position als *Vector2*, da alle auf $z = 0$ liegen und somit nur die *X* und *Y* Position relevant sind. Außerdem wird für jede Platte ihre Nachbarn und die Objekte, welche evtl. auf der Platte instanziiert wurden, gespeichert.

Von jedem Knoten des Graphen werden Kanten zu allen verfügbaren direkten Nachbarn gebildet, welche ebenfalls Bodenplatten sind. Ein Beispiel, wie ein solcher Graph aussehen kann, ist in Abb 4.7 zu erkennen.

Abbildung 4.7: Screenshot des Graphens aus der Anwendung mit grünen Kanten und roten Knoten



Dieser Graph wird anschließend verwendet, um Kreise zu finden. Diese Kreise werden schlussendlich die Räume werden, in welche der Spieler hineinlaufen kann. Für die Findung

der Kreise wird zunächst eine minimale Kreislänge festgelegt. Das Festlegen dieser erfolgt basierend auf der Anzahl der Bodenplatten. Dafür wird die Anzahl der Bodenplatten mit 0.11 multipliziert. Dieser Wert wurde empirisch durch Testen ermittelt, um keine zu kleinen Räume zu verhindern, bei denen sich der Spieler beengt fühlen würde und gleichzeitig viele Variation für mehr Abwechslung möglich ist. Dadurch werden die Räume in der Anwendung größer, umso größer der reale Raum ist und bleibt dennoch performant. Alle Kreise, welche gefunden werden, mit einer Länge kleiner als diese minimale Länge, werden verworfen und nicht gespeichert. Dies sorgt für eine bessere Performance. Mittels Tiefensuche werden dann Kreise gefunden. Hier wird die Anzahl der Kreise, welche gespeichert werden, auf 100 limitiert, ebenfalls aus Performance-Gründen. Die Tiefensuche funktioniert sehr ähnlich, wie in 2.2.4 beschrieben. Es wird über alle Knoten iteriert. Dabei wird jeweils von dem aktuellen Knoten ausgehend die Nachbarn besucht und als besucht markiert. Wenn der besuchte Knoten der Anfangsknoten ist, werden die besuchten Knoten als Kreis abgespeichert. Anschließend wird einer der gefundenen Kreise zufällig ausgewählt und ein Raum auf diesem instanziiert.

4.2.4 Raumgenerierung

Für die Raumgenerierung wird sich zunächst auf die reine Generierung des Raumes und noch nicht auf das tatsächliche Instanziiieren der visuellen Umgebungen beschränkt. Zu Beginn wird, wie bereits im vorherigen Abschnitt erwähnt, ein zufällig gefundener Kreis ausgewählt. Anschließend wird zur Sicherheit dieser auf doppelte Knoten überprüft, da es in einigen seltenen Fällen dazu kommen kann, dass Start- und Endknoten eines Kreises doppelt hinzugefügt wurden, da die Position eines Knotens als *float* gespeichert ist und es in Unity selten zu *floating point errors* kommen kann. Aufgrunddessen wird ein Knoten als zwei unterschiedliche Knoten gespeichert. Die Funktion *RemovDoubleNodes* iteriert daher über alle Knoten des ausgewählten Kreises und entfernt alle Knoten, deren Distanz zueinander kleiner als 0.02 ist.

Subkreisbildung

Im Anschluss daran wird der ausgewählte Kreis in zwei weitere Kreise aufgeteilt. Diese bilden die Basis für die Impossible Spaces. Wie bereits in 3.2 erwähnt, stellten Evan A. Suma et al. in ihrer Untersuchung zur Überlappung von Impossible Spaces [Suma et al., 2012] fest, dass eine Überlappung von bis zu 56% nicht bzw. nur geringfügig auffällt. Diese Grenzen maximal auszunutzen und einzuhalten ist aufgrund der zufälligen Generierung nicht immer möglich, dennoch wird diese auf einen Bereich von 40% bis maximal 56% Überlappung eingegrenzt. Dafür wird zuerst die Länge des ersten Subkreises bestimmt, zufällig in einem Bereich von 40% – 56% der Länge des originalen Kreises. Dabei ist es das Ziel, dass jeder Subkreis 50% des ursprünglichen Kreises beansprucht und zu 56% mit den übrigen 50% überlappt. Daher

beträgt die minimale Länge des Subgraphen $minSubcycleLength = CycleLength \times 0.7$ und die maximale Länge $maxSubcycleLength = CycleLength \times 0.78$. Die Werte 0.7 und 0.78 berechnen sich hier als $0.5 \times 0.56 + 0.5 = 0.78$ bzw. $0.5 \times 0.4 + 0.5 = 0.7$. Folglich ergibt sich mit 78% Abdeckung aller Knoten des original ausgewählten Kreises eine Überlappung beider Subkreise von maximal 56%. Für den zweiten Subkreis wird die Länge wie folgt berechnet: $SubCycle2Length = cycleLength - subcycle1Length + minOverlapNeeded$. Die Länge ergibt sich also als Differenz des ursprünglichen Kreises und dem zuerst gefundenen Subkreis, addiert mit einem Parameter *minOverlapNeeded*. Dieser garantiert, dass alle Knoten des ursprünglichen Kreises garantiert in einem der beiden Subgraphen enthalten sind. Eine Verletzung der 56% Grenze wird hierbei in Kauf genommen, da ein Knoten, welcher in keinem der beiden Subkreise vorhanden ist, aber im originalen Kreis, zu Problemen bei der späteren Instanziierung der Objekte auf den Knoten führen kann.

Im nächsten Schritt werden die Türen bzw. *Transition Areas* markiert und platziert. Es werden jeweils zwei benachbarte Knoten gesucht, welche sich außerhalb der *Transition Areas* befinden, so dass diese einen oder zwei Nachbarn besitzen, welche *Ground* Knoten sind. Dadurch wird garantiert, dass der Spieler Zugang zu diesen hat und die Türen betreten kann. Im optimalen Fall befindet sich mindestens ein Knoten zwischen beiden Türen. Falls eine solche Kombination innerhalb von 20 Versuchen nicht gefunden werden kann, wird ebenfalls eine Konfiguration akzeptiert, bei welcher beide Türen direkt nebeneinander liegen. In diesem Fall funktioniert alles wie geplant, jedoch kann die Immersion evtl. gestört werden, da deutlich offensichtlicher beide Türen von außen sichtbar sind und beim Betreten des Raumes eine Tür verschwindet. Mit dem Platzieren der *Transition Areas* sind alle Schritte für die Vorbereitung zum tatsächlichen Instanzieren der Objekte vollzogen.

4.2.5 Generierung der Objekte

Das Platzieren der Objekte der Umgebung erfolgt in zwei Schritten. Im ersten Schritt wird nur die *Starting Area*, der Boden, die Wände und die Hindernisse platziert. Den Raum bzw. die Räume, welche der Spieler begehen kann, werden erst instanziiert, sobald dieser durch die *Starting Area* hindurchläuft. Die Anwendung ist optisch an einer Höhle, einem Bergbau oder Ähnlichem orientiert. Die Idee und der Grund dieser Wahl werden später in Abschnitt 4.3 erklärt. Ein Screenshot aus der Anwendung ist in Abbildung 4.12 zu sehen.

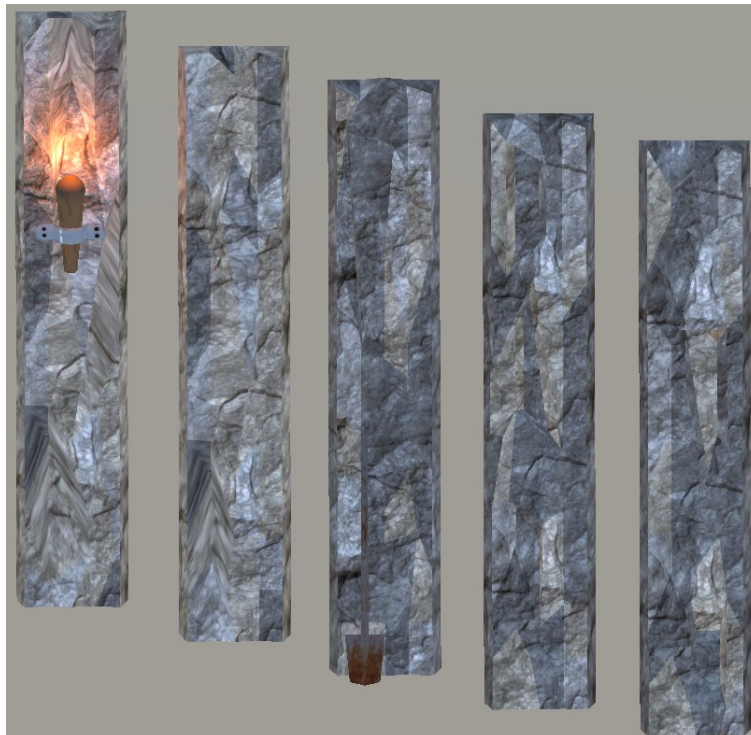
Schritt 1

Für den ersten Schritt werden Wände am Rande des Raumes bzw. des Spielfeldes platziert. Für diesen Schritt wurde bereits bei der Erstellung des Graphen Vorarbeit geleistet. Bei der Erstellung des Graphen wurden zu Beginn die Nachbarn von jedem Knoten gesucht, in positiver X und Y und in negativer X und Y Richtung. Sollte kein Nachbar in dieser Richtung

gefunden worden sein, wurde ein Knoten an dieser Stelle gespeichert und als *outside Obstacle* markiert. Diese Knoten liegen also außerhalb des physischen Raumes und dienen nur zur Platzierung der Wände.

Mithilfe dieser Knoten können alle Wände korrekt platziert werden. Dafür wird über alle Knoten iteriert und bei denen, welche einen Nachbarn besitzen, welcher ein *outside Obstacle* ist, auf diesen werden Wände platziert. Mithilfe von zwei Hilfsfunktionen wird die korrekte Rotation und Position bestimmt. Dafür wird der aktuelle Knoten mit dem gefundenen Nachbarknoten verglichen und anhand der Position zueinander die Position und Rotation der Wand bestimmt. Es gibt insgesamt 5 verschiedene Wände, welche zufällig ausgewählt werden. Eine Übersicht der Wände ist in Abbildung 4.8 zu erkennen.

Abbildung 4.8: Übersicht aller Wände, aus welchen die Umgebung generiert wird

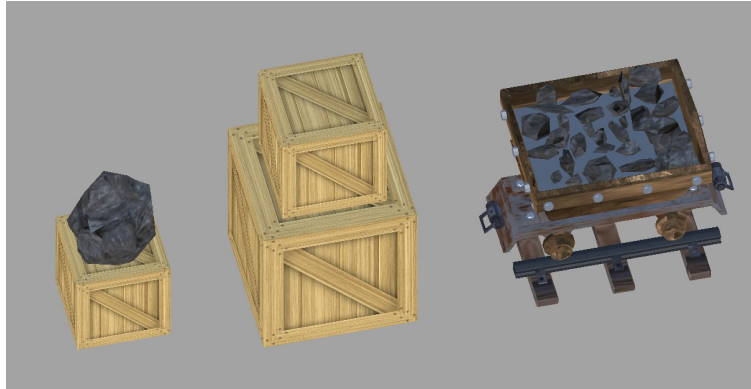


Diese, wie auch alle anderen Assets, wurden eigens mithilfe von Blender modelliert. Die Wände werden dabei aus einer Liste gezogen, in der alle verfügbaren Wände sind. Um eine natürliche Atmosphäre für den Spieler zu erschaffen, wurden die Häufigkeiten der Elemente angepasst, damit Wand 2, 4 und 5 häufiger vorkommen, da diese einer simplen Steinwand gleichen. Insgesamt waren diese Wände jeweils 4 Mal in der Liste vorhanden die Wände 1 und 3 nur ein einziges Mal. Folglich bestand eine Wahrscheinlichkeit von $\approx 85\%$ für eine normale Steinwand und $\approx 15\%$ für eine Wand mit einer Fackel (Wand 1) oder eine mit einer Schaufel (Wand 3).

Als nächstes werden die Hindernisse platziert, bevor als letztes die *Starting Area* platziert

wird. Für den Spieler passiert das Platzieren der Wände, der Hindernisse und der *Starting Area* alles in einem Schritt. Insgesamt wurden drei verschiedene Arten von Hindernissen modelliert. In Abbildung 4.9 sind alle Hindernisse in aufsteigender Größe zu erkennen.

Abbildung 4.9: Übersicht aller Hindernisse in steigender Größe von links nach rechts 1×1 , 2×2 und 2×3



Diese beanspruchen unterschiedlich viele Knoten. Hindernis 1 beansprucht nur einen einzigen Knoten. Hindernis 2 benötigt eine Fläche von 2×2 Knoten und Hindernis 3 einen von 2×3 Knoten. Durch die verschiedenen Arten von Hindernissen sollen für den Spieler Variation geschaffen werden. Das Platzieren erfolgt durch zwei Hilfsfunktionen: *CheckAndPlaceBigObstacle* und *CheckAndPlaceMediumObstacle*. Es wird zunächst über alle Hindernisknoten, also alle Knoten, welche in vorherigen Schritten als *Obstacle* markiert wurden, iteriert. Außerdem werden in drei HashSets alle Knoten gespeichert, welche Verwendung in den jeweiligen Hindernissen finden. Dadurch soll verhindert werden, dass mehrere Hindernisse ineinander platziert werden. Zuerst wird für jeden Knoten überprüft, ob dieser bereits in einem der drei HashSets vorhanden ist. Wenn dies nicht der Fall ist, wird überprüft, ob die Platzierung eines großen Hindernisses, also eines 2×3 Hindernisses mithilfe von *CheckAndPlaceBigObstacle* möglich ist. In dieser Funktion wird mit einer weiteren Hilfsfunktion überprüft, ob es ein mögliches 2×3 Feld an Knoten gibt. Diese Funktion bekommt unter anderem drei Knoten übergeben. Von diesen drei Knoten werden die Nachbarn, welche ebenfalls Hindernisse sind, als potentielle Knoten in einer temporären Liste gespeichert. Anschließend wird diese Liste in eine neue umgewandelt, welche die doppelten Knoten entfernt, wenn z.B. zwei Knoten denselben Nachbarn besitzen. Wenn auf diese Weise 6 einzigartige Knoten gefunden wurden, wird überprüft mit einer weiteren Funktion *IsRectangle*, ob sich diese als ein Rechteck orientiert in *X* oder *Z* Richtung anordnen lassen. Ist dies der Fall, wird ein 2×3 Hindernis instanziiert. Die Position wird dabei als Mittelwert aller 6 Knoten ermittelt. Zum Schluss werden die verwendeten Knoten im Hashset gespeichert.

Als nächster Schritt wird die Platzierung eines 2×2 Hindernisses überprüft, mit sehr ähnlicher Methodik, wobei hier die Überprüfung, aufgrund der geringeren Größe, deutlich einfacher

ist. Auf allen Hindernis-Knoten, welche auf diese Weise nicht abgedeckt wurden, werden 1×1 große Hindernisse platziert, um sicherzustellen, dass jedes Hindernis von einem Objekt belegt ist.

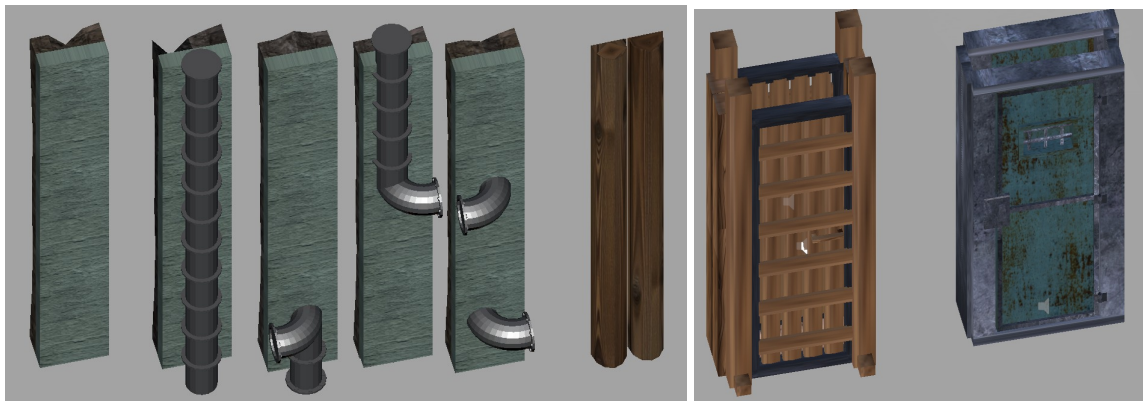
Als letzten Schritt wird, wie bereits erwähnt die *Starting Area* platziert. Diese wird immer an den Rand des Spielfeldes bzw. Raumes platziert, so dass diese möglichst wenig Platz vom Spielfeld beansprucht. Die *Starting Area* ist als simpler Felsdurchgang modelliert, mit einer leichten Krümmung, um eine neue Generierung der Spielwelt für den Spieler weniger offensichtlich zu machen. Für die korrekte Rotation wird ermittelt, welche der 4 *Starting Area* Knoten *outside Obstacles* als Nachbarn besitzen. Da dies immer nur 2 sein können, wird basierend darauf die Rotation festgelegt. Die Position erfolgt erneut als Mittelwert der 4 Knotenpositionen.

Schritt 2

In Schritt 2 wird der Raum generiert, den der Spieler betreten kann und es werden Goldklumpen instanziiert, die der Spieler aufsammeln kann. Eine weitere Erklärung der Goldklumpen wird in Abschnitt 4.3 gegeben. In diesem Schritt wird außerdem erklärt, wie die Platzierung der Objekte beim Betreten der Räume funktioniert.

Zunächst werden auf den in Abschnitt 4.2.4 gefundenen *Transition Areas* Prefabs von Türen platziert. Es gibt 2 Varianten von Türen bzw. von Räumen, die der Spieler betreten kann. Genau wie das Innere der Räume ist eine Tür einem industriellen Stil nachempfunden und die andere hölzernen. Der Kontrast soll einen klaren Unterschied zwischen den Räumen schaffen.

Abbildung 4.10: Übersicht der Wände und Türen innerhalb der Räume



Die Türen, wie in Abb 4.10 auf der rechten Seite zu sehen sind, bestehen jeweils aus einer Vorder- und Hintertür. Beim Betreten, in dem Moment, wo der Spieler einen Collider triggert, wird die Vorder- oder Hintertür deaktiviert. Ein gefundener Kreis kann Bodenplatten bzw. Knoten einschließen, welche nicht Teil des Kreises sind. In diesem Fall sind die vorderen und hinteren Nachbarn der Tür beides Bodenplatten, weshalb sich nicht direkt sagen lässt,

wo die Außenseite des Raumes ist, deshalb gibt es immer eine Tür auf beiden Seiten. Neben dem Deaktivieren der anderen Türhälfte wird die andere *Transition Area* bzw. die Tür zum 2. Raum deaktiviert und temporär als *extended CyclePart* markiert. Die Knoten der *Transition Area* werden außerdem gespeichert, damit beim Hinausgehen die Tür wieder an ein und derselben Stelle platziert werden kann.

Beim Betreten einer der beiden Türen wird der Raum instanziiert und gleichzeitig erweitert. Die Erweiterung ist dabei ebenfalls wieder auf das 56%-fache der aktuellen Größe limitiert. Jeder Nachbar des aktuellen Kreises wird überprüft. Wenn dieser als *Ground* oder *CyclePart* markiert ist, wird dieser als potentieller Knoten für die Erweiterung des Kreises gespeichert. Außerdem müssen die gefundenen Nachbarn in einem Kegel von 180° in Richtung des Betretens der Tür liegen, damit der Kreis nicht hinter der Tür beim Betreten erweitert wird, um zu verhindern, dass die Tür plötzlich inmitten des Raums steht.

Von den gespeicherten potentiellen Knoten werden anschließend so viel hinzugefügt, bis die maximale Anzahl, also 56% der ursprünglichen Größe erreicht ist. Anschließend werden die Wände und der Boden des Raumes instanziiert. Die Rotation und Position der Wände wird mithilfe der Informationen über die Nachbarn berechnet. Alle Knoten, welche einen oder mehrere Nachbarn besitzen, welche nicht Teil des Kreises sind, benötigen eine Wand. Die Rotation kann hierbei aus der Position des Nachbarn zu dem aktuellen Knoten berechnet werden. An den Ecken des Raumes werden zwei Wände instanziiert, da es auch 2 Nachbarn gibt, welche nicht Teil des Kreises sind. Außerdem wird unabhängig von den Nachbarn auf allen Knoten des Kreises ein Boden und eine Decke instanziiert. Zum Schluss werden noch 1 – 2 Goldklumpen für den Spieler im Raum an zufälligen Positionen platziert.

Beim Verlassen des Raumes wird der Ursprungszustand wiederhergestellt und alle instanziierten Objekte werden zerstört. Wenn der Spieler in beiden Räumen war und somit die Spielwelt vollständig erkundet hat, kann dieser durch die *Starting Area* gehen und es wird eine neue Welt instanziiert. In dem Moment, wo der Spieler den Trigger der *Starting Area* auslöst, werden bis auf die Hindernisse alle Objekte zerstört und ein neuer Kreis aus den zu Anfang in Abschnitt 4.2.3 gefundenen Kreisen ausgewählt. Dadurch lässt sich eine gewisse Variation der Spielwelt erleben und bietet die Möglichkeit, mehrere Räume zu erkunden. Nachdem jetzt in den vorherigen Abschnitten die technische Umsetzung ausführlich beschrieben wurde, wird der nächste Abschnitt näher auf das Ziel für den Spieler eingehen und die damit verbundene Zielsetzung der Studie.

4.3 Spieldesign

In diesem Abschnitt wird die Wahl des Höhlen-Settings für die virtuelle Umgebung begründet und das Ziel des Spiels für dne Spieler näher erläutert.

Ein Wohnzimmer besitzt auch mit den hier vorgenommenen Optimierungen zur Hinder-

nisvermeidung nur begrenzten Platz. Gleichzeitig sorgt eine Plattengröße von 30×30 cm für die Knoten des Graphen für potenziell schmale Wege. Deshalb fiel die Wahl auf ein Höhlen-Setting. Eine Höhle vermittelt das Gefühl, in einem eher begrenzten Raum zu sein, was den Platz angeht. Gleichzeitig kann die eher geringe Beleuchtung gut genutzt werden, als Anreiz für den Spieler, die Welt zu erkunden. Dem Spieler wurde eine feste Laterne in die Hand gegeben, fest an den Controller gebunden, welche ein schwaches Licht ausstrahlt. Ziel der Anwendung bzw. des Spiels ist es, fünf Goldklumpen zu sammeln. Über einige Tutorial-Hinweise wird dem Spieler zum einen erklärt, wie er die Anwendung zu bedienen hat, und zum anderen, was das Ziel des Spiels ist. Die Tutorial-Hinweise sind jederzeit über die Menü-Taste des linken Controllers aufrufbar und erscheinen direkt vor einem wie in Abbildung 4.11 zu erkennen.

Abbildung 4.11: Beispiel eines Tutorial-Hinweises in der Anwendung



Dafür werden zunächst alle fünf Goldklumpen in grau angezeigt und werden farblich ausgefüllt, sobald einer eingesammelt wurde. Als entsprechenden Namen wurde die Anwendung „Treasure Tunnels“ genannt.

Durch das Ziel, fünf Goldklumpen einzusammeln, wird der Spieler gezwungen bzw. dazu angeregt, sich durch die Welt fortzubewegen und ein Gefühl für die Umgebung zu bekommen. In Abbildung 4.12 ist ein Screenshot aus der Sicht eines Spielers zu sehen. In der unteren linken Ecke lässt sich ein Goldklumpen zum Einsammeln erkennen. Unter dem Controller ist außerdem eine Lampe fest angebracht, womit der Spieler leuchten kann, um mehr zu sehen.

Außerdem reicht es nicht nur, einen Raum zu besuchen, um alle fünf Goldklumpen zu finden, weshalb jeder Spieler mindestens einmal durch die *Starting Area* laufen muss, um alle fünf zu finden. Mit diesem Ziel wurde das Studiendesign am Paper von Liu et al. Liu et al., 2021

orientiert.

Abbildung 4.12: Screenshot aus der Sicht des Spielers während der Anwendung



5 Studie

Mit der in dieser Arbeit durchgeführten Studie soll ermittelt werden, ob sich die Implementierte Anwendung mit der Verwendung von dynamischen Impossible Spaces mit passendem Mapping realer und virtueller Hindernisse positiv auf die Nutzerfahrung auswirkt im Vergleich zu nicht abgestimmten oder fehlenden Hindernissen. Außerdem sollen die Ergebnisse der Studie genutzt werden, um einen direkten Vergleich zu ziehen, mit der von Liu et al. [Liu et al., 2021], um den Effekt von Impossible Spaces zu Messen. Aufgrund dessen, ist das Studiendesign sehr an dem von Liu et al. orientiert.

5.1 Studiendesign

Für das Testen wurden drei Szenarien für jeden Teilnehmer vorbereitet:

- Eine virtuelle Umgebung ohne Hindernisse
- Eine virtuelle Umgebung mit unstimmigen Positionen der realen und virtuellen Hindernisse
- Eine virtuelle Umgebung mit Hindernissen positioniert an den Positionen der Hindernisse in der realen Welt

Für das Testen wurde ein Raum von $\sim 7 \times 4.5m$ verwendet. Für den ersten Fall wurden keine Hindernisse im Raum platziert. Nur einige Stühle und Tische am Rande waren nicht zu entfernen, weshalb diese alle als Hindernisse markiert wurden. Diese waren jedoch zu keinem Zeitpunkt zugänglich und daher nicht Teil der begehbaren virtuellen Umgebung. In Abbildung 5.1 ist eine Aufnahme des Raumes zu sehen, in der die Studie durchgeführt wurde.

Für die zufällige Platzierung wurde ein manuell gewählter Wert von 12% für die Anzahl der Hindernisse in der virtuellen Welt genommen. In der realen Welt wurden anschließend Hindernisse in Form von Stühlen und Tischen platziert, welche folglich nicht mit den virtuellen übereinstimmten. Damit der Spieler nicht gegen diese stolpert, wurde das Quest Guardian System verwendet. Dieses aktiviert automatisch den Passthrough-Modus, sollte der Spieler einem Hindernis zu nahe kommen und schützt damit den Spieler vor einer Kollision.

Für den optimierten Fall wurden, wie im zweiten Fall, einige Hindernisse in der realen Welt

Abbildung 5.1: Aufnahme des Raumes in welcher die Studie durchgeführt wurde. Einige Stühle sind als Hindernis platziert.



platziert, jedoch diesmal auch mit virtuellen Hindernissen an den gleichen Positionen. In allen drei Fällen wurde vorab, bevor die Teilnehmer die Anwendung spielen konnten, zur Initialisierung die ersten Schritte durchgeführt. Dies beinhaltet neben dem Scannen des Raumes, auch das Auswählen der Raum Begrenzungen. Das Auswählen dieser wurde zwar im Tutorial für den Spieler erklärt, dennoch stellt der Raum in dem die Studie stattfand einige Schwierigkeiten dar. Der Raum war getrennt mit einigen Vorhängen, Tischen und ähnliches, wie in Abbildung 5.1 auf der rechten Seite zu erkennen. Dadurch besaß der Raum der Studie nur drei wirkliche Wände. Diese Problematik sorgte dafür, das der Raumscan der Quest 3 mehrere Raumecken identifizierte auf unterschiedlichen Höhen. Eine Wahl einer falschen Raumecke hätte die Anwendung zunichte gemacht, weshalb dieser Teil ebenfalls vorab übernommen wurde. Für den zweiten Fall der zufälligen nicht übereinstimmenden Platzierung wurde außerdem der Quest Guardian für den Spieler bereits vorab eingezeichnet, da dies keinen Einfluss auf die Anwendung und die Ergebnisse hat.

5.2 Durchführung

Jeder Teilnehmer testete alle 3 Szenarien, wobei die Reihenfolge zufällig ausgewählt wurde. Zu Beginn der Durchführung wurde der Teilnehmer grob über die ihm bevorstehende Anwendung aufgeklärt. Hierbei wurde darauf geachtet, nicht preiszugeben, welches der drei Szenarien der Teilnehmer zuerst spielen würde. Es wurde auf das Ziel des Spiels hingewiesen, die Funktion der Raumgenerierung durch Betreten der *Starting Area* und dem Hinweis, dass es zwei Räume in jeder Welt gibt. Außerdem unterschrieb jeder Teilnehmer vorab eine Erklärung, dass die von ihm angegebenen Daten im Rahmen dieser Masterarbeit verwendet werden dürfen und online gespeichert werden. Hätte es einem Teilnehmer unwohl werden sollen oder sonstige Beschwerden gegeben, war es jedem freigestellt, jederzeit abubrechen oder eine Pause einzulegen.

Nach Abschluss eines Durchlaufs füllten die Teilnehmer einen Fragebogen zu der gerade erlebten Erfahrung aus. Die Fragen dieses Fragebogens decken sich mit dem Fragebogen, welchen Liu et al. [Liu et al., 2021] verwendeten, ein Beispiel ist im Anhang zu finden. Der Fragebogen bestand aus einer Frage zum Präsenzgefühl, 5 Fragen zur spatial Präsenz, 4 Fragen zum Realismus, 4 Fragen zum Angstgefühl während der Anwendung, 4 Fragen, wie involviert sich die Teilnehmer gefühlt haben, sowie einer Freitextfrage. Die Fragen zur Präsenz, zur spatial Präsenz, zum Realismus und zur Involviertheit stammen aus dem 14-teiligen Igroup Presence Questionnaire (IPQ) [I-group.org-project, 2016]. Die Fragen wurden aufgeteilt in die soeben beschriebenen Kategorien. Die Frage zur Präsenz richtet sich nach der Definition zur Präsenz nach Slater und Usoh [Slater et al., 1994]. Die 4 Fragen zum Angstgefühl wurden aus dem Fragebogen zur Emotion von Tcha-Tokey et al. [Tcha-Tokey et al., 2016] entnommen. Die Reihenfolge der Fragen wurde dabei für jeden Teilnehmer und jedes Szenario zufällig ausgewählt. Dafür wurde SoSci Survey genutzt [Leiner, 2021]. Zu Beginn des Fragebogens wurden noch einige generelle Fragen über jeden Teilnehmer gestellt. Hier wurde nach Alter, Geschlecht, VR-Erfahrung, Gaming-Erfahrung und ob der Teilnehmer ein eigenes VR-Headset besitzt, gefragt. Die Fragen wurden alle auf Englisch gestellt, um den Sinn und die Intention jeder ursprünglichen Frage beizubehalten. Wenn Fragen während der Beantwortung aufkamen, war ich jederzeit anwesend, um diese zu beantworten.

Die Fragen bzw. Aussagen des Fragebogens waren jeweils bewertbar auf einer Skala von 1 bis 7 mit unterschiedlichen Minima und Maxima. Beispielsweise konnte die Aussage „*Somehow I felt that the virtual world surrounded me.*“ mit „*Fully Disagree*“ bis hin zu „*Fully Agree*“ bewertet werden. Die Durchführung aller 3 Szenarien, inklusive des Beantwortens der Fragebögen, beanspruchte etwa 25-30 min pro Teilnehmer.

6 Ergebnisse und Diskussion

In diesem Kapitel werden die Ergebnisse der Studie statistisch ausgewertet und anschließend interpretiert. Außerdem wird ein direkter Vergleich zur Studie von Liu et al. [Liu et al., 2021] gezogen. Zunächst werden die Hypothese und Analysearten genannt und begründet, bevor die Ergebnisse dieser gezeigt werden.

Für diese Arbeit wurde eine Studie mit 16 Personen durchgeführt. Davon waren 6 weiblich und 10 männlich im Alter von 23 bis 29 Jahren. Alle befragten Personen waren Studenten im Bachelor oder Master. Mithilfe dieser Studie wurde folgende Hypothese überprüft:

Die Nutzung dynamischer Impossible Spaces mit passendem Mapping realer und virtueller Hindernisse erhöht Präsenz, Realismus, Involviertheit, spatiale Präsenz und reduziert Angst im Vergleich zu nicht abgestimmten oder fehlenden Hindernissen

Mithilfe von SoSci [Leiner, 2021] konnten die Daten aller Teilnehmer als CSV-Datei heruntergeladen werden. Von diesen wurden dann für jeden Testfall für jede der fünf zu testenden Kategorien (Präsenz, Spatial Präsenz, Realismus, Angst, Involviertheit) für jeden Teilnehmer der Durchschnittswert ermittelt. Mit diesen Ergebnissen wurde eine einfaktorielle Varianzanalyse mit Messwiederholungen (ANOVA) durchgeführt und auf Signifikanz überprüft. Die Datenanalyse der Uni Zürich [Zürich, 2023] diente hierbei für die Entscheidungsfindung der Analyseart.

Die in diesem Fall unabhängige Variable sind die drei verschiedenen Anwendungsvarianten. Im Folgenden wird der Anwendungsfall, dass die Hindernisse der realen Welt mit denen der virtuellen Welt übereinstimmen, als **Matching** bezeichnet, der Fall, in dem die Hindernisse nicht übereinstimmen und zufällig platziert werden, als **Random** und der Fall komplett ohne Hindernisse, real und virtuell, als **No Obstacles**. Die abhängige Variable ist das Ergebnis des Durchschnitts, der zu jeder Kategorie zugehörigen Fragen. Als Voraussetzung für die ANOVA Varianzanalyse wurde die Sphärizität der Datensätze mithilfe des Mauchly Tests

[Rasch et al., 2022] überprüft. Die Normalverteilung wurde mithilfe des Shapiro-Wilk Tests [Duller, 2022] überprüft. Die Durchführung der Tests und auch die Varianzanalyse wurden in Python mithilfe der Bibliotheken von Pandas [pandas development team, 2020] und Scipy [Virtanen et al., 2020] durchgeführt.

Eine A Priori Analyse wurde mithilfe von G*Power [Buchner, 1996] durchgeführt und ein paarweiser mit Bonferroni angepasster t-Test als Post hoc in Python.

6.1 Statistische Auswertung

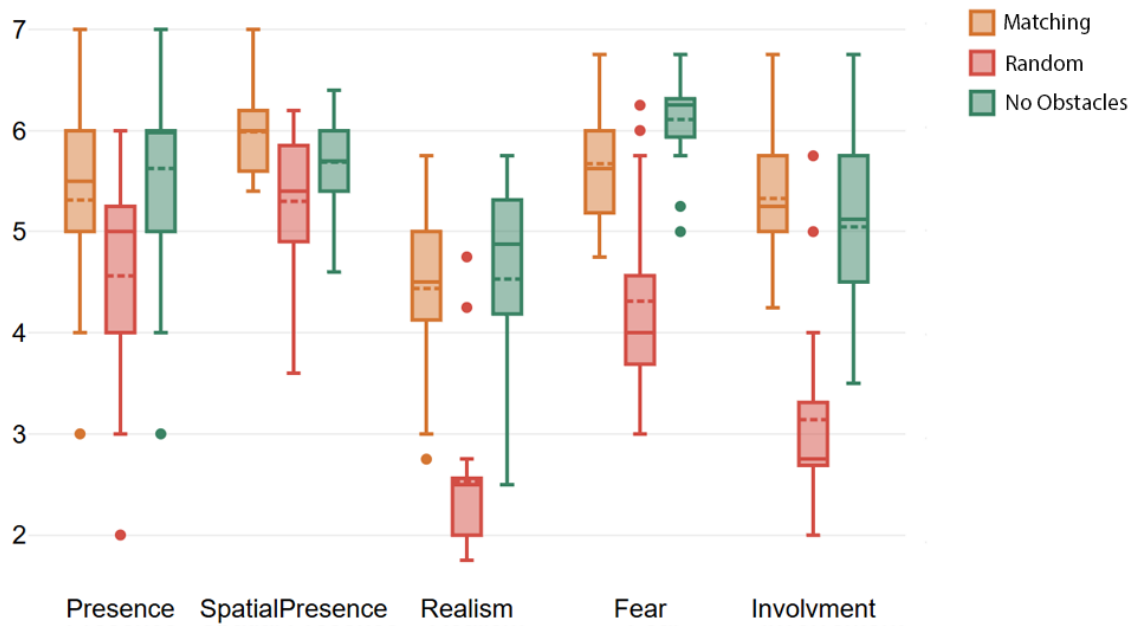
In diesem Abschnitt werden zunächst die Testergebnisse genannt, bevor diese dann im darauffolgenden Kapitel interpretiert werden. Vor Beginn der Studie wurde die A-Priori-Analyse durchgeführt, mit einer Effektgröße von 0.25 einem $\alpha_{err prob}$ von 0.05, einer Gruppe mit 3 Messungen für die 3 verschiedenen Testszenarien und einer Korrelation von 0.8 zwischen den wiederholten Messungen.

Die Ergebnisse ergaben eine benötigte Anzahl von 19 Testpersonen mit einer Power von 0.961. Wie bereits zu Beginn erwähnt, wurde diese Studie nur mit 16 Personen durchgeführt. Die mit dieser Anzahl durchgeführte Post-hoc-Analyse mit denselben Parametern resultierte in einer Stärke ($1 - \beta_{err prob}$) von ≈ 0.92 . Die Wahrscheinlichkeit für einen Fehler zweiter Art liegt folglich nur bei etwa 8% bzw. die Wahrscheinlichkeit, einen tatsächlichen Effekt zu finden, sollte es einen geben, bei 92%. Die statistische Aussagekraft ist dementsprechend weiterhin sehr hoch.

Die Ergebnisse des Mauchly-Tests zeigen eine Verletzung der Sphärizität für den Realismus mit $p = 0.018$. Aufgrund dessen wurde hier die Greenhouse-Geisser-Korrektur [Duller, 2022] verwendet. Der korrigierte p-Wert liegt bei $p = 0.0014$. Die Normalität der Residuen nach dem Shapiro-Wilk-Test ist für fast alle Kategorien gegeben (Präsenz $p = 0.2048$, Spatial Präsenz $p = 0.2788$, Angst $p = 0.5669$, Involviertheit $p = 0.6223$). Lediglich für den Realismus liegt mit $p = 0.0403$ eine leichte Verletzung der Normalverteilungsannahme vor, da der Wert knapp unter dem Grenzwert von 0.05 liegt. Da die ANOVA bei leichten Verletzungen der Normalverteilungsannahme jedoch als robust gilt [Zürich, 2023], insbesondere bei ausgeglichenen Gruppengrößen wie in dieser Studie, wurde das Verfahren dennoch angewendet.

In Abbildung 6.1 sind die Ergebnisse des Fragebogens aller Teilnehmer als Boxplot dargestellt. Die verschiedenen Kategorien wurden gegenüber der Skala des semantischen Differential der Fragen des Fragebogens aufgetragen.

Abbildung 6.1: Boxplot der Ergebnisse des Fragebogens aller drei Testszenarien



Zusammen mit den Ergebnissen der Varianzanalyse mit Messwiederholungen lassen sich mehrere Erkenntnisse ziehen. Es konnten signifikante Unterschiede für das **Präsenzgefühl** über alle 3 Konditionen hinweg festgestellt werden ($F(2, 30) = 5.15, p = 0.012, np^2 = 0.256$). Eine Post-hoc-Analyse zeigt darüber hinaus signifikante Unterschiede zwischen *Random* und *No Obstacles* mit $p = 0.024$, mit Durchschnittswerten von $M = 5.63$, im Fall *No Obstacles* und $M = 4.56$ für den Fall *Random*. Zwischen *Matching* und *Random* lässt sich kein signifikanter Unterschied zeigen ($p = 0.184$). Dennoch lässt sich sowohl anhand des Boxplots als auch beim Vergleich des Durchschnittswerts mit $M = 5.31$ eine Verbesserung erkennen.

Für die **Spatial Präsenz** konnte ebenfalls ein signifikanter Unterschied festgestellt werden ($F(2, 30) = 12.33, p < 0.001, np^2 = 0.451$). Aus der Post-hoc-Analyse lassen sich signifikante Unterschiede zwischen *Matching* und *Random*, sowie *No Obstacles* mit $p = 0.001$ und $p = 0.019$ zeigen. Der Durchschnittswert liegt für den *Matching* Fall bei $M = 5.99$ im Vergleich zu *Random* ($M = 5.30$) und *No Obstacles* ($M = 5.69$) sehr hoch.

Der **erfahrene Realismus** weist signifikante Unterschiede zwischen den Konditionen *No Obstacles* und *Matching* im Bezug auf die *Random* Kondition auf. Dies lässt sich sehr leicht ebenfalls anhand des Boxplots 6.1 erkennen. Der *Matching* Fall hat einen Durchschnittswert von $M = 4.44$, die *No Obstacles* Kondition $M = 4.53$, dagegen hat der Fall *Random* nur einen Durchschnittswert von $M = 2.53$ und $p < 0.001$. Über alle Kategorien hinweg konnte insgesamt somit ein signifikanter Unterschied festgestellt werden, wobei hier erneut anzumerken ist, dass die Greenhouse-Geisser Korrektur genutzt wurde, aufgrund der verletzten Sphärizität ($F(2, 30) = 25.31, p < 0.001, np^2 = 0.628$).

Das **Angstgefühl** während der Anwendung zeigt signifikante Unterschiede mit einer ebenfalls sehr großen Effektgröße auf ($F(2, 30) = 25.69, p < 0.001, np^2 = 0.631$). Aus der post-hoc-Analyse und aus dem Boxplot lässt sich ein signifikanter Unterschied von *Matching* ($p = 0.001$) und *No Obstacles* ($p < 0.001$) gegenüber *Random* erkennen. Die Durchschnittswerte von *Matching* ($M = 5.67$), *Random* ($M = 4.31$) und *No Obstacles* ($M = 6.11$) verdeutlichen dies noch einmal.

Die **Involviertheit** der Teilnehmer mit der Anwendung weist signifikante Unterschiede auf ($F(2, 30) = 32.10, p < 0.001, np^2 = 0.682$). Erneut lässt sich aus der post-hoc Analyse ein signifikanter Unterschied zwischen den Konditionen *Matching* ($M = 5.33, p < 0.001$) und *No Obstacles* ($M = 5.05, p < 0.001$) gegenüber *Random* ($M = 3.14, p < 0.001$) erkennen.

6.2 Diskussion

Die im vorherigen Abschnitt ausgewerteten Ergebnisse werden in diesem Kapitel interpretiert und anschließend mit denen von Liu et al. verglichen.

6.2.1 Interpretation

Die Ergebnisse bestätigen die aufgestellte Hypothese und zeigen, dass die Nutzung dynamischer Impossible Spaces mit passendem Mapping realer und virtueller Hindernisse die Präsenz, Involviertheit, spatiale Präsenz, den Realismus und die Angst reduziert im Vergleich zu nicht abgestimmten oder fehlenden Hindernissen. Es ließen sich signifikante Unterschiede in jeder der fünf Kategorien feststellen. Die post-hoc-Analysen sowie der Boxplot geben uns jedoch noch weitere Informationen, die wir im Bezug auf die Anwendung interpretieren können.

Das Präsenzgefühl weist nur signifikante Unterschiede zwischen *No Obstacles* und *Random* auf. Dennoch zeigt der Boxplot sowie der Durchschnittswert vom *Matching* Fall mit $M = 5.31$ im Vergleich zum *Random* Fall mit $M = 4.56$ einen Trend der Verbesserung auf. Dass hier keine signifikante Verbesserung festgestellt werden konnte, kann eventuell an der teils sehr engen Umgebung mit mehreren Hindernissen liegen, wodurch sich die Teilnehmer wohler fühlten, als in einer freieren ohne Hindernisse. Außerdem wurde die Präsenz nur mit einer einzigen Frage ermittelt, weshalb sich hier der Wert entsprechend schnell festlegt. Die gewählte Frage war hier, wie bereits erwähnt, aus dem IPQ Questionnaire [I-group.org-project, 2016] und lautete: *In the computer generated world I had a sense of "being there"*. Des Weiteren war wohl möglich für ein signifikantes Ergebnis die Fallzahl zu gering.

Die spatiale Präsenz dagegen zeigte einen signifikanten Unterschied nicht nur zwischen dem Fall *Random* und *No Obstacles* sondern auch zwischen *Random* und *Matching*. Insgesamt fiel hier die spatiale Präsenz am besten aus. Die Werte sind für alle drei Fälle ziemlich hoch,

was vermutlich auch unter anderem an einer der vier Fragen bzw. Aussagen zur spatialen Präsenz liegt, welche lautete: *I felt like I was just perceiving pictures*. Aufgrund der mittlerweile deutlich fortgeschrittenen Technologie im Vergleich zu dem, wann diese Frage entstanden ist, lagen die Frames beim Verwenden der Anwendung immer über 72 fps, um für ein angenehmes Erlebnis für die Teilnehmer zu sorgen. Einzig und allein beim ersten Instanzieren der Umgebung konnten die Frames kurz fallen, worauf aber die Teilnehmer hingewiesen wurden. Dies liefert eine potentielle Erklärung weshalb diese Frage über allen Konditionen hinweg sehr gut bewertet wurde, was den ins gesamten Durchschnitt entsprechend nach oben zieht.

Die Werte des Realismus der Anwendung fallen insgesamt von allen Kategorien am schlechtesten aus. Dennoch lässt sich ein deutlicher signifikanter Unterschied vom *Random* Fall gegenüber *Matching* und *No Obstacle* erkennen. Die insgesamt schlechten Ergebnisse lassen sich vermutlich auf die optische Umsetzung der Anwendung zurückführen. Aussagen wie *The virtual world seemed more realistic than the real world* wurden mit eher schlechteren Werten beantwortet, da gerade für Teilnehmer, die bereits VR-Erfahrung besitzen, wie es bei allen in dieser Studie der Fall war, die virtuelle Welt klar unterscheidbar ist von der realen. Darüber hinaus kam es weiterhin an einigen Stellen bei den Wänden oder auch Hindernissen zu clipping Problemen, bei denen einige Objekte in andere hineinragen, was den Realismus stört. Außerdem hatten von den befragten Personen 8 etwas bis sehr viel Erfahrung mit Videospielen generell und nur eine einzige Person gar keine. Dementsprechend waren die Testpersonen an virtuelle Welten, ob in VR oder in anderer Form, gewöhnt. Jedoch sind dies keine Limitationen der Methode und können durch eine verbesserte Optik in Form von Texturen und besseren Modellen behoben werden.

Der Bruch der Immersion durch das „Durchscheinen“ der realen Welt, sobald ein Teilnehmer mit dem Quest Guardian in Berührung kommt, wurde hier deutlich sichtbar im Fall der zufällig platzierten Hindernisse. In diesem Fall wird der Passthrough-modus der Quest aktiviert und die reale Welt wird sichtbar, bis der Teilnehmer wieder im markierten Bereich ist. Das Gefühl einer realen Welt wurde vermutlich hierdurch stark beeinträchtigt.

Das Angstgefühl war sowohl in der *No Obstacles* als auch in der *Matching* Variante deutlich geringer als im Vergleich zur *Random* Variante. Hier zeigt sich entsprechend deutlich, dass das Platzieren der virtuellen Hindernisse an Positionen der realen das Angstgefühl stark reduziert hat. Die Variante komplett ohne Hindernisse schnitt noch besser ab, was jedoch auch zu erwarten war, da komplett fehlende Hindernisse in realer und virtueller Welt eine Angst vor Kollision nahezu komplett entfernen.

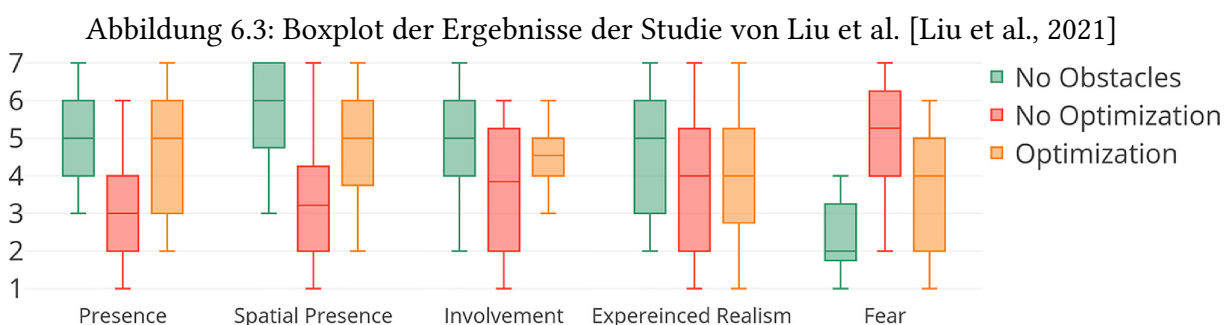
Die Involviertheit mit der Anwendung viel sowohl beim Fall *Matching* als auch bei *No Obstacles* sehr gut aus. Der Fall *Matching* zeigte sogar minimal bessere Ergebnisse im Durchschnitt, was vermutlich an der ausgeschmückteren Umgebung liegt, da diese mit mehr Hindernissen noch etwas voller wirken könnte. Ein deutlicherer Unterschied zwischen *Matching* und *No Obstacles* ließe sich eventuell in einer anderen Testumgebung ermitteln, da auch, wenn keine Hindernisse im *No Obstacles* Fall im Spielbereich des Spielers waren, dennoch welche sichtbar

waren, was die Welt fertiger aussehen ließ. Dies war, wie in Abbildung 5.1 zu erkennen ist, nicht anders möglich, aufgrund des künstlich zweigeteilten Raumes.

Insgesamt fiel der Anwendungsfall *Matching* in drei Kategorien besser aus als *No Obstacles* und in allen besser als *Random*. Ein signifikanter Unterschied gegenüber *No Obstacles* konnte nur in der spatialen Präsenz gezeigt werden. In keinem Fall fiel *Matching* signifikant schlechter als *No Obstacles*. Gegenüber dem *Random* Fall konnte in allen Fällen bis auf die Präsenz ein signifikanter Unterschied im Vergleich zu *Matching* gezeigt werden. Insofern lässt sich die Optimierung der Hindernisse als Erfolg bezeichnen.

6.2.2 Vergleich

In diesem Abschnitt werden die eben präsentierten Ergebnisse im Vergleich mit der Studie von Liu et al. [Liu et al., 2021] gesetzt, um den Einfluss der Impossible Spaces bewerten zu können. Wie bereits erwähnt, untersuchten Liu et al. in ihrer Studie ebenfalls den Effekt von optimierten Hindernissen in den drei verschiedenen Anwendungsfällen, da diese zufällig platziert wurden, ohne Hindernisse und dem *Matching* bzw. optimierten Fall. In ihrer Studie wurden keine Impossible Spaces verwendet, darüber hinaus wurde die Umgebung nicht zur Laufzeit, sondern im Voraus gescannt, um diese verarbeiten zu können. Das Ziel der Anwendung der Studie war ein ähnliches für den Spieler. Dieser musste umherlaufen und Puzzleteile einsammeln, um eine Schatzkiste zu öffnen. Die Umgebung war jedoch deutlich anders gestaltet und ähnelte eher einem Garten oder Park. Ein Screenshot der Anwendung, entnommen aus dem Paper von Liu et al. ist in Abbildung 6.2 zu sehen. Ein Boxplot der Ergebnisse ist in Abbildung 6.3 zu sehen. Die farbliche Kodierung der verschiedenen Konditionen wurde in dieser Arbeit zur besseren Vergleichbarkeit übernommen. Dennoch ist anzumerken, dass bei dem von Liu et al. gemessenen Angstgefühl ein niedriger Wert weniger Angst bedeutet, während in dieser Arbeit alle Fragen so angepasst wurden, dass 7 das optimale Ergebnis, also in diesem Fall so wenig Angst wie möglich, repräsentiert.



Auf einen ersten Blick beim Vergleich beider Boxplots fällt auf, dass die Ergebnisse für die Involviertheit und den erfahrenen Realismus im *Random* Fall in der eigenen Studie deutlich

Abbildung 6.2: Screenshot aus der Anwendung welche in der Studie von Liu et al. verwendet wurde [Liu et al., 2021]



schlechter ausfallen, als in der von Liu et al.. Ein möglicher Grund könnte hier die größere Vielfalt der Hindernisse im Fall von Liu et al. sein. In der eigenen Untersuchung wurden im Fall der zufällig platzierten Hindernisse nahezu immer nur Hindernisse der Größe 1×1 platziert, aufgrund des zufälligen Auswählens der Hindernisplatten. In der Anwendung von Liu et al. dagegen wurden fünf verschiedene Hindernisse der Größe 1×1 verwendet, was auch im Fall der zufällig platzierten Hindernisse für mehr Abwechslung und damit einen höheren Realismus sorgt. Das deutlich schlechtere Gefühl der Involviertheit in der eigenen Untersuchung, im Fall der zufällig platzierten Hindernisse, lässt sich nur durch Vermutungen erklären. Zum einen könnte die dynamische Platzierung der Räume dafür gesorgt haben, dass diese sehr unnatürlich wirken, da diese im zufälligen Fall bzw. *Random* Fall häufig unnatürlich wirken, um den Hindernissen auszuweichen, welche gleichmäßig über den Raum verteilt sind. Die einzusammelnden Gold Nuggets konnten dementsprechend an merkwürdigen Stellen platziert sein und teils in andere Objekte hineinragen. Der Spaßfaktor kann dadurch sinken und die Anwender fühlen sich weniger involviert. Die Oculus Quest verfügte außerdem noch über keinen Passthrough-modus, wodurch bei der Aktivierung des Quest-Guardians bei der bevorstehenden Kollision mit einem Hindernis nicht die reale Welt durchschien. Dadurch wurde die Immersion weniger stark gebrochen. Die genaueren Ursachen könnten in einer zukünftigen Studie und weiteren Tests untersucht werden.

Beim genaueren Blick auf die einzelnen Kategorien lassen sich mehrere Dinge feststellen und

im Bezug auf die verwendeten Impossible Spaces setzen.

Für die Präsenz fallen die Durchschnittswerte insgesamt schlechter aus in der Studie von Liu et al. jedoch im Verhältnis etwa gleich. Liu et al. erhielten Durchschnittswerte von $M = 4.64$ für den *Matching* bzw. optimierten Fall, einen Durchschnittswert $M = 4.88$ für den Fall *No Obstacles* und $M = 3.24$ für den Fall der zufällig platzierten Hindernisse. Im Vergleich dazu erzielte die eigene Untersuchung Durchschnittswerte von $M = 5.31$ für den *Matching* Fall, $M = 5.63$ für den *No Obstacles* Fall und $M = 4.56$ für den *Random* Fall. Die Verwendung von Impossible Spaces deutet also auf eine Verbesserung des Präsenzgefühls über alle 3 verschiedenen Konditionen hinweg hin. Während die Kategorien im Verhältnis zueinander ähnlich bleiben.

Die Ergebnisse für die spatiale Präsenz zeigen in beiden Studien einen signifikanten Unterschied auf. Anders als in der Studie von Liu et al., fallen in eigener Untersuchung die Ergebnisse des *Matching* Falls am besten aus. In der Studie von Liu et al. erzielt der *No Obstacles* Fall mit einem Durchschnittswert von $M = 5.56$ den besten Wert gegenüber dem *Matching* Fall mit $M = 4.92$. Der *Random* Fall fällt in beiden Studien am schlechtesten aus, bei Liu et al. mit $M = 3.52$, in dieser Arbeit mit $M = 5.3$. Auch hier sind erneut alle Werte in eigener Untersuchung insgesamt besser. Die besten Werte werden hier mit dem Fall *Matching* erzielt, jedoch ohne signifikanten Unterschied im direkten Vergleich zum Fall *No Obstacles*. Daher kann auch für die spatiale Präsenz eine Verbesserung über alle drei Konditionen hinweg erkannt werden.

Bei Betrachtung des Realismus der Anwendung fällt zum einen, wie bereits erwähnt, auf, dass die Werte im *Random* Fall deutlich schlechter ausfallen als bei der Studie von Liu et al., zum anderen wurde in dieser Kategorie in der Studie von Liu et al. kein signifikanter Unterschied festgestellt. Für den Mittelwert wurden keine genauen Zahlen genannt, jedoch lässt der Boxplot vermuten, dass diese für alle Kategorien zwischen 4 und 5 liegen. Damit ähneln die Werte den in eigener Studie erfassten Werten. Dementsprechend lässt sich vermuten, dass die Verwendung von Impossible Spaces in den Fällen *No Obstacles* und *Matching* keinen Einfluss auf den Realismus haben. Im Fall der zufälligen Platzierung könnte dies jedoch aufgrund der damit zusammenhängenden dynamischen Generierung für sehr unnatürliche Raumaufbauten gesorgt haben. Gleichzeitig sorgt die geringere Vielfalt an Hindernissen für ein repetitives Erlebnis, wodurch die Erfahrung weniger realistisch wird.

Für die Betrachtung der Angst werden die Werte von Liu et al., so umgedreht, dass ein hoher Wert ein geringeres Angstgefühl bedeutet. Ein Durchschnittswert von $M = 2.32$ im Fall *No Obstacles* entspricht also einem Wert von $M = 5.68$ mit 7 als optimalem Wert. Verglichen mit den Werten aus eigener Studie für *No Obstacles* $M = 6.11$ und *Matching* $M = 5.67$ im Vergleich zum Durchschnittswert im *Matching* Fall von $M = 4.52$ von Liu et al. lässt sich eine sehr geringfügige Verbesserung erkennen. Dennoch bleibt in beiden Studien das Resultat dasselbe, dass die Teilnehmer eine Umgebung mit gar keinen Hindernissen bevorzugen, um keine Angst vor einer Kollision zu haben. In beiden Fällen fällt der *Random* Fall deutlich am schlechtesten aus. Die insgesamt verbesserten Werte in eigener Untersuchung könnten

auch am Fortschritt der Technologie durch einen verbesserten Quest-Guardian und präzises Tracking liegen. Alle Teilnehmer hatten bereits VR-Vorerfahrung und auch wenn sie nicht wussten, welches der 3 Szenarien sie spielen würden, hatten diese entsprechendes Vertrauen in die Technologie.

Die Involviertheit mit der Anwendung zeigt erneut ähnliche Ergebnisse. Die stark reduzierte Involviertheit in der eigenen Anwendung für den Fall *Random* wurde bereits zu Anfang begründet. Die Ergebnisse von *Matching* und *No Obstacles* zeigen mit $M = 5.33$ und $M = 5.05$ ähnliche Resultate zu Liu et al. mit $M = 4.56$ im *Matching* Fall und $M = 4.96$ im Fall der *No Obstacles*. Eine leichte Verbesserung aller Werte mit Ausnahme des *Matching* Falls ist dennoch zu erkennen, was auf ein verstärktes Gefühl der Involviertheit aufgrund der Impossible Spaces hindeutet.

Dieser Vergleich zeigte, dass die Verwendung von Impossible Spaces definitiv keinen negativen Effekt auf die Anwendung hatte. Lediglich im Fall des Realismus und der Involviertheit schnitt der Fall der zufällig platzierten Hindernisse deutlich schlechter ab. Für alle anderen Fälle zeigt sich die Verwendung von Impossible Spaces als Bereicherung. Durch die Verwendung dieser wird eine gleichbleibende bis sogar bessere Immersion geschaffen bei größerer virtueller Fläche. Außerdem lassen sich vielfältigere Anwendungen umsetzen und explorative Möglichkeiten für den Spieler schaffen. Das Verhältnis aller Werte zueinander ist sehr ähnlich in der im Rahmen dieser Arbeit durchgeführten Studie als auch in der von Liu et al.. Der Durchschnitt der Werte ist in der Studie dieser Arbeit jedoch insgesamt höher.

7 Limitationen

Im vorherigen Kapitel wurden die erfolgreichen Ergebnisse präsentiert und gezeigt, welchen positiven Einfluss impossible Spaces haben.

Jedoch hat diese Anwendung auch Einschränkungen und Probleme, welche in diesem Kapitel erläutert werden. Die meisten dieser Probleme oder Limitationen wurden im Verlauf der Arbeit schon sehr kurz angerissen oder genannt. In diesem Kapitel werden diese noch einmal dargelegt.

Bereits bei der Konzeption der Anwendung wurden bewusst Limitationen in Kauf genommen. Die Anwendung wurde für rechteckige Räume entwickelt. Kleine Ecken in einem Raum oder Schränke, die Teil der freien Bodenfläche blockieren, lassen sich als Hindernisse markieren. Jedoch bei einem Raum, welcher eher einer Kurve entspricht, wäre ein sehr großer Teil als Hindernis markiert, was die Immersion etwas zerstören würde.

Jedoch ist dies kein generelles Problem, da eine Lösung durchaus möglich ist. Diese würde den Nutzer der Anwendung mehr fordern, da dieser mehrere Raumecken markieren müsste und außerdem wäre die Erstellung des Graphen komplexer. Für ein Testen dieser Anpassungen wären zusätzlich reale Räume mit entsprechender Architektur nötig gewesen. All dies wäre mit mehr Zeitaufwand definitiv umsetzbar, allerdings spielen diese Optimierungen und Anpassungen keine Rolle für das Ergebnis der im Rahmen dieser Arbeit durchgeführten Studie.

Eine weitere Limitation der Raumerstellung liegt bei der Begrenzung der Anwendung auf einen Raum. Die Idee liegt nahe, die Anwendung auf zwei oder mehrere Räume auszuweiten, für eine größere virtuelle Umgebung und damit eine immersivere Erfahrung. Jedoch treten hier zum einen die selben Probleme der Eckenfindung auf und zum anderen wird in der Anwendung das Scan-Feature der Quest 3 genutzt, welches nur einen Raum scannt. Dieses Feature erkennt entsprechend auch das Verlassen eines Raumes, wodurch die Anwendung terminiert bzw. in den Passthrough-Modus gezwungen wird, als würde man mit einem Hindernis kollidieren. Interne Funktionen der Quest lassen sich meistens nur sehr schwer anpassen, was eine eigene Implementierung des Scan-Features der Quest zufolge hätte, was deutlich den Rahmen dieser Arbeit sprengen würde. Entsprechend ist die Anwendung nur auf einen einzigen rechteckigen Raum begrenzt.

Bei der Erstellung des Graphen wurde eine Größe der Platten von 30×30 cm gewählt. Diese fest gewählte Größe bringt entsprechende Limitationen mit sich. Eine kleinere Plattengröße würde für eine bessere Anpassung und mehr Kontrolle über die genaue Position von Hinder-

nissen sorgen. Sollte beispielsweise ein Hindernis mit nur 10 cm mit einer Platte überlappen, so wird die gesamte Platte als Hindernis markiert. Kleinere Bodenplatten bringen allerdings viele weitere Schwierigkeiten mit sich. Zum einen wird durch kleinere Bodenplatten mehr Leistung beansprucht, die Anzahl der Knoten in dem Graphen steigt sehr schnell, wodurch mehr Leistung für das Finden der Kreise im Graphen gebraucht wird. Bereits bei der aktuell gewählten Größe führte dies bereits in größeren Räumen zu Problemen, weshalb Einschränkungen vorgenommen wurden, wie das Abbrechen nach 100 gefundenen Kreisen und der Wahl einer Mindestgröße für Kreise. Diese Einschränkungen haben zur Folge, dass in großen Räumen die Kreise häufig sehr ähnlich aufgebaut sind, mit einer ähnlichen Größe, da weitere komplexere Kreise mit noch mehr Kreisen erst später gefunden werden.

Aber auch diese Limitation könnte behoben werden, indem der Algorithmus für die Kreisfindung optimiert wird, um weniger Rechenleistung zu beanspruchen und zum anderen noch bessere dynamische Schranken für die Einschränkungen in der Kreisfindung zu schaffen.

Neben Problemen bei der Kreisfindung bringen jedoch kleinere oder auch größere Platten ein weiteres Problem bei der Erstellung der virtuellen Umgebung mit sich. In der Anwendung sind alle Objekte in der Welt, die Wände, Hindernisse, Türen usw., auf die Größe der Bodenplatten abgestimmt. Eine Tür beansprucht beispielsweise immer 2 Platten für eine Größe von 60 cm. Eine Verkleinerung der Platten würde entsprechende Anpassungen für die Welterstellung bedeuten.

Eine weitere Limitation sind die begrenzten unterschiedlichen Objekte in der Welt. Es gibt 3 verschiedene Hindernisgrößen, jedoch sind die Hindernisse für diese Größen immer dieselben, was schnell repetitiv werden kann. Auch für die Wände, sowohl innerhalb der Räume als auch außerhalb, gibt es nur eine sehr begrenzte Anzahl. Eine Erweiterung mit mehr Objekten wäre einfach umzusetzen, jedoch beansprucht das neue Modellieren von Objekten Zeit.

Neben den eben genannten Möglichkeiten zur Verbesserung der Anwendung und Limitationen dieser, gibt es auch einige Probleme in der Anwendung. Aufgrund der dynamischen Erstellung der Welt und daher einer Umgebung, die nie dieselbe ist, kommt es an einigen Wänden und auch Hindernissen zu Clipping-Problemen, bei denen Objekte in andere hineinragen. Diese Probleme sind häufig schwierig vollständig zu eliminieren, da diese zum einen nicht immer auftreten und erst durch intensiveres Testen gefunden werden, und zum anderen ist es oft nicht klar, ob die Clipping-Probleme aufgrund von fehlerhafter Logik zum Platzieren der Objekte entstehen oder ob die Modellierung der Objekte bei zu naher Platzierung aneinander dies verursacht. Zusammen mit der zufälligen Platzierung der Objekte lassen sich diese Probleme nur Stück für Stück über die Zeit beheben, weshalb es auch innerhalb der Studie für einige Teilnehmer zu Clipping-Problemen kam.

Zusammenfassend lassen sich alle Limitationen in zukünftigen Entwicklungen beheben, jedoch konnte die Anwendung bereits jetzt eindeutig demonstrieren, dass VR-Anwendungen selbst im eigenen Wohnzimmer gespielt werden können und dabei eine immersive, ununterbrochene Erfahrung ermöglichen, bei der Verwendung von natürlichem Gehen als Fortbewegung.

8 Ausblick & Fazit

In dieser Arbeit wurde eine vielversprechende VR-Anwendung präsentiert, welche Hindernisse in der realen Welt auf virtuelle Hindernisse mappt. Zusammen mit der Verwendung von impossible spaces, wurde eine Studie mit 16 Personen durchgeführt, um folgende Hypothese zu untersuchen:

Die Nutzung dynamischer Impossible Spaces mit passendem Mapping realer und virtueller Hindernisse erhöht Präsenz, Realismus, Involviertheit, spatiale Präsenz und reduziert Angst im Vergleich zu nicht abgestimmten oder fehlenden Hindernissen

Mithilfe der Studie konnte die Hypothese bestätigt werden und es wurde gezeigt, dass es signifikante Unterschiede gibt im Bezug auf die Präsenz, spatiale Präsenz, Realismus, Involviertheit und dem Angstgefühl während der Anwendung, im Vergleich zu einer Anwendung, bei der die Hindernisse nicht mit der realen Welt übereinstimmen und einer, bei der es gar keine Hindernisse gibt.

Die Ergebnisse zeigten signifikant bessere Ergebnisse des Matching-Falls, bei dem die Hindernisse der realen und virtuellen Welt übereinstimmen, in allen Kategorien außer der Präsenz. In der Kategorie der Präsenz war dennoch die Matching-Anwendung besser, jedoch ohne einen signifikanten Unterschied.

Aus dem Vergleich zu der Studie von Liu et al. [Liu et al., 2021] ließ sich außerdem eine Verbesserung erkennen. Die Verhältnisse der Kategorien untereinander sahen ähnlich aus, jedoch fiel die Bewertung der Kategorien insgesamt höher aus. Mit Ausnahme des Randoms Falls in der Kategorie Realismus und Involviertheit.

Die implementierte Anwendung demonstriert vielversprechend als Proof-of-Concept eine Anwendung, die die Möglichkeiten für derartige VR-Anwendungen zeigt. Es wurde gezeigt, dass die Methode für ein deutlich besseres immersives Erlebnis sorgt, besonders für ein besseres räumliches Präsenzgefühl und geringe Angst beim Fortbewegen in der virtuellen Welt. Limitationen im Bezug auf die Performance der Kreisfindung oder prozeduralen Generierung könnten in zukünftigen Verbesserungen der Anwendung behoben werden.

Die Konzeption und Entwicklung dieser Anwendung stellte die mit Abstand größte Schwierigkeit dar, aufgrund der dynamisch entstehenden Impossible Spaces. Vor allem das Finden der Kreise und Generieren der Objekte in der Welt bereitete große Schwierigkeiten. In Zukunft könnten die Probleme dieser Anwendung für eine für den Nutzer leicht zu bedienende Anwendung, ohne jegliche Clipping-Probleme oder Limitationen des Raumlayouts, behoben werden. Möglicherweise ließe sich dafür das Scan-Feature der Quest noch besser nutzen,

für eine vollständig automatische Erstellung des Graphen ohne Auswählen der Raumecken. Ein verbesserter Algorithmus oder mehr Rechenkapazität einer eventuellen Quest 4 oder ähnlichem könnten darüber hinaus dazu beitragen, mehr unterschiedliche Kreise zu finden und gleichzeitig komplexere Modelle für ein stärkeres Gefühl des Realismus zu ermöglichen. Insgesamt war die in dieser Arbeit entwickelte Anwendung ein Erfolg. Die Konzeption dieser Anwendung lässt sich für zukünftige Anwendungen mit unterschiedlichem Level-Design und Spielmechaniken, welche reales Gehen in der virtuellen Welt als Locomotion-Methode nutzen und dennoch die Angst vor Kollisionen verhindern, verwenden. Die Ergebnisse der Studie zeigten außerdem eine Verbesserung in der angepassten Kondition gegenüber den anderen beiden getesteten und außerdem zeigte der direkte Vergleich zur Studie von Liu et al. [Liu et al., 2021] bessere oder genauso gute Werte. Dementsprechend lässt sich die Verwendung von Impossible Spaces ebenfalls als erfolgreich bezeichnen. Durch diese wird dem Nutzer eine virtuelle Welt geschaffen, die größer als die reale physische ist. Gleichzeitig sorgen diese für explorative Möglichkeiten und Leveldesign-Optionen die sonst nicht möglich wären.

Literatur

- Anaya-Sánchez, R., Rejón-Guardia, F., & Molinillo, S. (2024). Impact of virtual reality experiences on destination image and visit intentions: the moderating effects of immersion, destination familiarity and sickness [last accessed 20.05.2025]. *International Journal of Contemporary Hospitality Management*. <https://doi.org/10.1108/ijchm-09-2023-1488>
- Azmandian, M., Grechkin, T., Bolas, M., & Suma, E. (2015). Physical space requirements for redirected walking: how size and shape affect performance. *Proceedings of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments*, 93–100.
- Bachmann, M., Subramaniam, A., Born, J., & Weibel, D. (2023). Virtual reality public speaking training: effectiveness and user technology acceptance [last accessed: 22.05.2025]. *Frontiers in Virtual Reality*, 4. <https://doi.org/10.3389/frvir.2023.1242544>
- Bailenson, J. N. (2024). Virtual Reality [<https://oecs.mit.edu/pub/2vci5sg1>]. In M. C. Frank & A. Majid (Hrsg.), *Open Encyclopedia of Cognitive Science*. MIT Press.
- Barnard, D. (2024). Examples of VR used for Training – Industry Case Studies [last accessed: 22.05.2025]. <https://virtualspeech.com/blog/vr-training-case-studies>
- Bozgeyikli, E. (2024). Locomotion in Virtual Reality Video Games [last accessed: 22.05.2025]. *Encyclopedia of Computer Graphics and Games*, 1070–1075.
- Buchner. (1996). GPOWER: A general power analysis program [last accessed: 22.05.2025]. <https://www.psychologie.hhu.de/arbeitsgruppen/allgemeine-psychologie-und-arbeitspsychologie/gpower>
- De, T., Tinga, A. M., & Louwerse, M. M. (2024). Natural- and redirected walking in virtual reality: Spatial performance and user experience [last accessed: 22.05.2025]. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-024-19879-1>
- Design4Real. (2024). <https://design4real.de/en/social-vr-events/>
- Diestel, R. (2025). *Graph Theory*. Springer-Verlag.
- Duller, C. (2022). *Einführung in die nichtparametrische Statistik mit SAS und R*. Physica-Verlag HD.
- Elad, B. (2024). Virtual Reality Statistics and Facts [last accessed: 22.05.2025]. <https://www.enterpriseappstoday.com/stats/virtual-reality-statistics.html>
- Galin, E., Peytavie, A., Maréchal, N., & Guérin, E. (2010). Procedural Generation of Roads [last accessed: 22.05.2025]. *Computer Graphics Forum*, 29(2), 429–438. <https://doi.org/10.1111/j.1467-8659.2009.01612.x>

- Hepperle, D., & Wölfel, M. (2023). Similarities and Differences between Immersive Virtual Reality, Real World, and Computer Screens: A Systematic Scoping Review in Human Behavior Studies [last accessed: 22.05.2025]. *Multimodal Technologies and Interaction*, 7(6), 56. <https://doi.org/https://doi.org/10.3390/mti7060056>
- I-group.org-project. (2016). <https://www.igroup.org/pq/ipq/download.php>
- Katja Mönius, P. S., Jörn Steuding. (2021). *Algorithmen in der Graphentheorie*. Springer-Verlag.
- Kiper, P., Przysiężna, E., Cieślik, B., Broniec-Siekaniec, K., Kucińska, A., Szczygieł, J., Turek, K., Gajda, R., & Szczepańska-Gieracha, J. (2022). Effects of Immersive Virtual Therapy as a Method Supporting Recovery of Depressive Symptoms in Post-Stroke Rehabilitation: Randomized Controlled Trial [last accessed: 22.05.2025]. *Clinical Interventions in Aging*, 17, 1673–1685. <https://doi.org/10.2147/CIA.S375754>
- Langbehn, E., Lubos, P., & Steinicke, F. (2018). Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking [last accessed: 22.05.2025]. *Proceedings of the Virtual Reality International Conference - Laval Virtual*. <https://doi.org/10.1145/3234253.3234291>
- Leiner, D. J. (2021). *SoSci Survey (Version 3.2.31)* [[Computer software]]. <https://www.soscisurvey.de>
- Liu, H., Wang, Z., Mazumdar, A., & Mousas, C. (2021). Virtual reality game level layout design for real environment constraints [last accessed: 22.05.2025]. *Graphics and Visual Computing*, 4, 200020. <https://doi.org/https://doi.org/10.1016/j.gvc.2021.200020>
- Lochner, D., & Gain, J. (2021). VR Natural Walking in Impossible Spaces [last accessed: 22.05.2025]. *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games*. <https://doi.org/10.1145/3487983.3488305>
- Meta. (2015). <https://www.meta.com/de/en/quest/quest-3/>
- Mojang Studios. (2011). Minecraft [last accessed: 08.06.2025].
- Nilsson, N. C., Serafin, S., Steinicke, F., & Nordahl, R. (2018). Natural Walking in Virtual Reality: A Review [last accessed: 22.05.2025]. *Comput. Entertain.*, 16(2). <https://doi.org/10.1145/3180658>
- pandas development team, T. (2020). *pandas-dev/pandas: Pandas (Version latest)* [last accessed: 22.05.2025]. Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Rasch, B., Frieze, M., Hofmann, W., & Naumann, E. (2022). *Quantitative Methoden 2* [last accessed: 22.05.2025]. Springer Berlin Heidelberg. <https://link.springer.com/book/10.1007/978-3-662-43548-9>
- Rebelo, A. R., Ferreira, P. A., & Nóbrega, R. (2024). Adaptive Virtual Environments in Small Physical Spaces: Navigation Design for Customized User Experiences in Virtual Reality [last accessed: 22.05.2025]. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3613905.3650942>
- Sari, R. C., Pranesti, A., Solikhatus, I., Nurbaiti, N., & Yuniarti, N. (2023). Cognitive overload in immersive virtual reality in education: More presence but less learnt? [last accessed:

- 22.05.2025]. *Education and Information Technologies*, 29(10), 12887–12909. <https://doi.org/https://doi.org/10.1007/s10639-023-12379-z>
- Shin, D. D. (2018). Empathy and embodied experience in virtual environment: To what extent can virtual reality stimulate empathy and embodied experience? [last accessed: 22.05.2025]. *Comput. Hum. Behav.*, 78. <https://doi.org/10.1016/j.chb.2017.09.012>
- Short, T., & Adams, T. (2017). *Procedural Generation in Game Design*. CRC Press.
- Silva, D. F., Torchelsen, R., & Aguiar, M. S. (2025). Procedural game level generation with GANs: potential, weaknesses, and unresolved challenges in the literature [last accessed: 22.05.2025]. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-025-20612-9>
- Slater, M. (2009). Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments [last accessed: 22.05.2025]. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535), 3549–3557. <https://doi.org/10.1098/rstb.2009.0138>
- Slater, M., Usoh, M., & Steed, A. (1994). Depth of Presence in Virtual Environments. *Presence*, 3, 130–144. <https://doi.org/10.1162/pres.1994.3.2.130>
- Slater, M., & Wilbur, S. (1997a). A Framework for Immersive Virtual Environments (FI-VE): Speculations on the Role of Presence in Virtual Environments [last accessed: 22.05.2025]. *Presence: Teleoperators and Virtual Environments*, 6(6), 603–616. <https://doi.org/10.1162/pres.1997.6.6.603>
- Slater, M., & Wilbur, S. (1997b). A Framework for Immersive Virtual Environments (FI-VE): Speculations on the Role of Presence in Virtual Environments [last accessed: 08.06.2025]. *Presence: Teleoperators and Virtual Environments*, 6(6), 603–616. <https://doi.org/10.1162/pres.1997.6.6.603>
- Smelik, R., Tutenel, T., Bidarra, R., & Benes, B. (2014). A Survey on Procedural Modelling for Virtual Worlds [last accessed: 22.05.2025]. *Computer Graphics Forum*, 33. <https://doi.org/10.1111/cgf.12276>
- Sra, M., Garrido-Jurado, S., & Schmandt, C. (2016). Procedurally generated virtual reality from 3D reconstructed physical space [last accessed: 22.05.2025]. *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. <https://doi.org/10.1145/2993369.2993372>
- Suma, E. A., Lipps, Z., Finkelstein, S., Krum, D. M., & Bolas, M. (2012). Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture [last accessed: 22.05.2025]. *IEEE Transactions on Visualization and Computer Graphics*, 18(4), 555–564. <https://doi.org/10.1109/TVCG.2012.47>
- Tcha-Tokey, K., Christmann, O., Loup-Escande, E., & Richir, S. (2016). Proposition and Validation of a Questionnaire to Measure the User Experience in Immersive Virtual Environments [last accessed: 22.05.2025]. *The International Journal of Virtual Reality*, 16, 33–48. <https://doi.org/10.20870/IJVR.2016.16.1.2880>

- Unity Technologies. (2024). *Unity* (Version 6000.0.23f1) [Game development platform]. <https://unity.com/>
- Valentini, I., Ballestin, G., Bassano, C., Solari, F., & Chessa, M. (2020). Improving Obstacle Awareness to Enhance Interaction in Virtual Reality [last accessed: 22.05.2025]. *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 44–52. <https://doi.org/10.1109/VR46266.2020.1581503942658>
- Van der Linden, R., Lopes, R., & Bidarra, R. (2014). Procedural Generation of Dungeons [last accessed: 22.05.2025]. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1), 78–89. <https://doi.org/10.1109/TCIAIG.2013.2290371>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python [last accessed: 22.05.2025]. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Virtuix. (2025). <https://virtuix.com/>
- Wozniak, P., Capobianco, A., Javahiraly, N., & Curticapean, D. (2020). Depth Sensor Based Detection of Obstacles and Notification for Virtual Reality Systems. In T. Ahram (Hrsg.), *Advances in Human Factors in Wearable Technologies and Game Design* (S. 271–282). Springer International Publishing.
- Zürich, U. (2023). Datenanalyse mit SPSS [last accessed: 22.05.2025]. https://www.methodenberatung.uzh.ch/de/datenanalyse_spss.html

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorstehende Masterthesis mit dem Titel

Dynamische Impossible Spaces in VR - Mapping von virtuellen und realen Hindernissen

bzw. im Falle einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit selbstständig ohne fremde Hilfe gefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum: _____

Unterschrift: _____