



Barrierefreiheit in Virtual Reality – Konzeption,  
Implementierung und Evaluation von binären  
Interaktionsschnittstellen in PaneoVR für Nutzende  
mit motorischen Einschränkungen

Masterarbeit  
zur Erlangung des akademischen Grades M.Sc.  
im Studiengang Medieninformatik  
des Fachbereichs Elektrotechnik und Informatik  
an der Technischen Hochschule Lübeck

Eingereicht von:  
Finja Wegener  
Matrikelnummer: 311854

Erstprüfer:  
Prof. Dr. Thies Pfeiffer  
Hochschule Emden/Leer

Zweitprüfer:  
Dr. Daniel Sacristán  
Technische Hochschule Lübeck

Ausgabe: 08. August 2024  
Abgabe: 16. Januar 2025

# Eigenständigkeitserklärung

Declaration of Originality

Wegener, Finja

**Name, Vorname**

*Last name, first name*

311854

**Matrikelnummer**

*Matriculation number*

**Ich versichere hiermit, dass ich die vorliegende**

*I hereby declare that this*

☐ **Hausarbeit**  
*term paper*

☐ **Bachelorarbeit**  
*bachelor's thesis*

☒ **Masterarbeit**  
*master's thesis*

**mit dem Titel**

*with the title*

Barrierefreiheit in Virtual Reality – Konzeption, Implementierung und Evaluation

von binären Interaktionsschnittstellen in PaneoVR für Nutzende

mit motorischen Einschränkungen

**eigenständig und ohne unerlaubte fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und Entlehnungen aus anderen Arbeiten kenntlich gemacht. Für den Fall, dass die Arbeit zusätzlich elektronisch und/ oder digital eingereicht wird, erkläre ich, dass die schriftliche und die elektronische und/ oder digitale Form identisch sind. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.**

is my own original work and any assistance from third parties has been acknowledged. I have clearly indicated and acknowledged all sources and resources as well as any borrowings from other works. In case of an additional electronic and/or digital submission of this work, I declare that the written form and the electronic and/or digital form are identical. This work has not previously been submitted either in the same or in a similar form to another examination office.

**Ich bin damit einverstanden, dass die vorliegende Hausarbeit/ Bachelorarbeit/ Masterarbeit für Veröffentlichungen, Ausstellungen und Wettbewerbe des Fachbereiches verwendet und Dritten zur Einsichtnahme vorgelegt werden kann.**

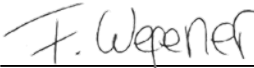
I agree that this work can be used for publishing, exhibition or competition purposes and can be inspected by third parties.

☒ **ja** ☐ **nein** ☐ **es liegt ein Sperrvermerk bis \_\_\_\_\_ vor**  
*Yes no there is an embargo period until*

Lübeck, 16.01.2025

**Ort, Datum**

*Place, Date*



**Unterschrift**

*Signature*

## **Themenbeschreibung**

Thema der Masterarbeit ist die Konzeption, die prototypische Implementierung sowie die Evaluation von zwei binären Interaktionsschnittstellen für das Tool PaneoVR. Damit soll es Menschen mit motorischen Einschränkungen ermöglicht werden, die mit PaneoVR erstellten Trainings zu erleben. Ziel ist es, alle erforderlichen Interaktionen über eine einfache, binäre Schalterbedienung durchführen zu können. Der wissenschaftliche Beitrag der Arbeit liegt insbesondere in der Gestaltung eines geeigneten Interfaces und der systematischen Konzeption der Interaktionen, um eine barrierearme Nutzung zu gewährleisten. Dazu werden zwei Konzeptentwürfe mit unterschiedlichem Schwerpunkt entwickelt und prototypisch umgesetzt. Anschließend erfolgt eine umfassende Evaluation der implementierten Interaktionsformen. Ziel ist es herauszufinden, welche Interaktionsform sich besser für den Einsatz in VR eignet und Empfehlungen für eine barrierearme Interaktion für Menschen mit motorischen Einschränkungen abzuleiten.

## Abstract

This thesis examines the development of binary interaction interfaces for virtual reality (VR) applications with the aim of improving accessibility for individuals with motor impairments. Based on a structured conceptualization, two approaches were developed that build on the established scanning methods Automatic Item Scanning and Continuous Cartesian Scanning, which were specifically adapted for use in VR. The implementation of both approaches was prototypically conducted on the basis of the tool PaneoVR.

The subsequent evaluation of the implemented interfaces analyzed differences in terms of efficiency, learnability, robustness, usability, user experience (UX), and the occurrence of motion sickness. The findings demonstrate that Item Scanning offers technical advantages such as higher interaction speed and robustness, while Cartesian Scanning is perceived more favorably in subjective assessments, particularly regarding usability and UX. Both approaches exhibit distinct strengths and weaknesses. Based on the evaluation, specific improvement strategies were derived, and recommendations for the development of binary interaction interfaces in VR applications were formulated. These recommendations include, among others, the implementation of diverse scanning methods, the integration of efficient error-correction mechanisms, and the ergonomic placement of UI and interaction elements. A key limitation of the results lies in the lack of a target group-specific evaluation, which restricts the generalizability of the findings to individuals with motor impairments. It is therefore recommended that future research implement the identified improvement strategies and validate them through evaluations conducted with the target population. The insights gained contribute significantly to the development of inclusive VR applications and advance the accessibility of virtual environments for individuals with motor impairments.

## Kurzfassung

Diese Arbeit untersucht die Entwicklung binärer Interaktionsschnittstellen für Virtual Reality (VR) Anwendungen mit dem Ziel, die Zugänglichkeit für Menschen mit motorischen Beeinträchtigungen zu verbessern. Auf Basis einer strukturierten Konzeption wurden zwei Ansätze entwickelt, die auf den etablierten Scanning-Verfahren Automatic Item Scanning und Continuous Cartesian Scanning basieren und gezielt für den Einsatz in VR adaptiert wurden. Beide Ansätze wurden prototypisch auf Basis des Tools PaneoVR implementiert.

Die anschließende Evaluation der implementierten Schnittstellen untersuchte die Unterschiede hinsichtlich Effizienz, Erlernbarkeit, Robustheit, Usability, User Experience (UX) und dem Auftreten von Motion Sickness. Die Ergebnisse zeigen, dass das Item Scanning durch technische Vorteile wie eine höhere Interaktionsgeschwindigkeit und Robustheit überzeugt, während das Cartesian Scanning in den subjektiven Bewertungen, insbesondere in Bezug auf Usability und UX, positiver wahrgenommen wird. Beide Ansätze zeigen spezifische Stärken und Schwächen. Auf Basis der Evaluation wurden konkrete Verbesserungsansätze abgeleitet und Empfehlungen für die Entwicklung binärer Interaktionsschnittstellen in VR-Anwendungen formuliert. Zu diesen Empfehlungen gehören u. a. die Implementierung unterschiedlicher Scanning-Verfahren, die Integration effizienter Mechanismen zur Fehlerkorrektur sowie die ergonomische Platzierung von UI- und Interaktionselementen. Eine zentrale Limitation der Ergebnisse besteht in der fehlenden zielgruppenspezifischen Evaluation, wodurch die Übertragbarkeit der Ergebnisse auf Menschen mit motorischen Beeinträchtigungen eingeschränkt ist. Zukünftige Arbeiten sollten die identifizierten Verbesserungsansätze umsetzen und durch zielgruppenspezifische Untersuchungen validieren. Die gewonnenen Erkenntnisse leisten einen Beitrag zur Entwicklung inklusiver VR-Anwendungen und tragen zur Verbesserung der Barrierefreiheit für Menschen mit motorischen Beeinträchtigungen bei.

# Inhaltsverzeichnis

<b>Themenbeschreibung</b>	<b>I</b>
<b>Abstract</b>	<b>II</b>
<b>Kurzfassung</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>X</b>
<b>Listings</b>	<b>XI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Zielsetzung . . . . .	2
1.3 Aufbau der Arbeit . . . . .	3
<b>2 Stand der Forschung</b>	<b>5</b>
2.1 Virtual Reality . . . . .	6
2.1.1 Motion Sickness . . . . .	9
2.2 Barrierefreiheit in VR . . . . .	12
2.2.1 Barrieren für Menschen mit motorischen Beeinträchtigungen bei der Nutzung von VR . . . . .	13
2.2.2 Lösungsansätze zur Überwindung von Barrieren . . . . .	15
2.3 Binäre Interaktionsschnittstellen . . . . .	16
2.3.1 Arten von Schaltern . . . . .	17
2.3.2 Scanning-Verfahren . . . . .	18
2.4 Usability und User Experience . . . . .	20
2.5 PaneoVR . . . . .	23
2.6 Zusammenfassung . . . . .	25
<b>3 Konzeption</b>	<b>27</b>
3.1 Definition des Design Space . . . . .	28

---

3.2	Ausprägungen der Interaktionsaufgaben und -komponenten . . . . .	30
3.3	Einordnung und Bewertung der Interaktionsaufgaben und -komponenten anhand der Parameter . . . . .	34
3.4	Ableitung der finalen Konzepte . . . . .	50
<b>4</b>	<b>Implementierung</b>	<b>53</b>
4.1	Architektur . . . . .	53
4.2	Gemeinsame Komponenten beider Konzepte . . . . .	56
4.3	Scanning-Verfahren . . . . .	58
4.3.1	Konzept 1 - Item Scanning . . . . .	58
4.3.2	Konzept 2 - Cartesian Scanning . . . . .	62
4.4	Strategien zur Fehlervermeidung und Debugging . . . . .	67
4.5	Limitationen und mögliche Erweiterungen . . . . .	67
4.6	Nutzung der Implementierung . . . . .	69
<b>5</b>	<b>Evaluation</b>	<b>70</b>
5.1	Konzeption der Evaluation . . . . .	70
5.1.1	Technische Evaluation . . . . .	70
5.1.2	Inhaltsbasierte Evaluation . . . . .	73
5.2	Durchführung . . . . .	76
5.3	Ergebnisse . . . . .	78
5.3.1	Usability (SUS) und User Experience (UEQ) . . . . .	78
5.3.2	Motion Sickness (SSQ) . . . . .	82
5.3.3	Ablenkung durch die Interaktionsschnittstellen . . . . .	86
5.3.4	Geschwindigkeit und Effizienz . . . . .	87
5.3.5	Robustheit . . . . .	90
5.3.6	Korrelationen und Zusammenhänge . . . . .	94
5.3.7	Qualitative Beobachtungen und Feedback der Teilnehmenden . . . .	96
<b>6</b>	<b>Diskussion</b>	<b>100</b>
6.1	Diskussion der Ergebnisse der Evaluation . . . . .	100
6.2	Zusammenfassung und Schlussfolgerungen . . . . .	113
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>118</b>
	<b>Literaturverzeichnis</b>	<b>121</b>
	<b>Anhang</b>	<b>129</b>
<b>A</b>	<b>Evaluationsprotokoll</b>	<b>130</b>

---

<b>B</b>	<b>Screenshots aus dem technischen Evaluationsabschnitt</b>	<b>133</b>
B.1	Item Scanning . . . . .	133
B.2	Cartesian Scanning . . . . .	135
<b>C</b>	<b>Screenshots aus dem inhaltsbasierten Evaluationsabschnitt</b>	<b>137</b>
C.1	Szenario Code . . . . .	137
C.2	Szenario Schlüssel . . . . .	140
<b>D</b>	<b>Ergebnisse UEQ</b>	<b>143</b>
<b>E</b>	<b>Digitaler Anhang</b>	<b>145</b>



# Abbildungsverzeichnis

2.1	Reality-virtuality continuum nach Milgram et al. [1995] in der erweiterten Version von Wohlgenannt et al. [2020] . . . . .	7
2.2	Einstufung der SUS-Scores nach Aaron Bangor and Miller [2008] . . . . .	22
3.1	Darstellung der möglichen Ausprägungen der Interaktionsaufgaben und -komponenten . . . . .	34
3.2	Einordnung der Gestaltungsoptionen der Interaktionsaufgabe Selektion anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort . . . . .	35
3.3	Einordnung der Gestaltungsoptionen der Interaktionsaufgabe Navigation anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort . . . . .	39
3.4	Einordnung der Gestaltungsoptionen der Interaktionskomponente Display anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort . . . . .	43
3.5	Einordnung der Gestaltungsoptionen der Interaktionskomponente Transition anhand der relevanten Parameter Erlernbarkeit, (Interaktions-) Geschwindigkeit und Komfort . . . . .	46
3.6	Einordnung der Gestaltungsoptionen der Interaktionskomponente Initialisierung anhand der relevanten Parameter Effizienz, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort . . . . .	48
3.7	Zusammenstellung des 1. Konzepts Item Scanning anhand der Gestaltungsoptionen für Interaktionsaufgaben und -komponenten . . . . .	50
3.8	Zusammenstellung des 2. Konzepts Cartesian Scanning anhand der Gestaltungsoptionen für Interaktionsaufgaben und -komponenten . . . . .	51
4.1	Grundlegender Informationsfluss der Softwarearchitektur . . . . .	55
5.1	Mittelwerte (inklusive SD) und Median der SUS-Scores beider Interaktionsschnittstellen . . . . .	79
5.2	Verteilung der SUS-Scores der beiden Interaktionsschnittstellen. Der SUS-Score liegt im Wertebereich zwischen 0 und 100 . . . . .	79
5.3	Visualisierung der UEQ-Ergebnisse des Item Scannings . . . . .	80

5.4	Visualisierung der UEQ-Ergebnisse des Cartesian Scannings . . . . .	82
5.5	Verteilung der SSQ-Scores beider Interaktionsschnittstellen . . . . .	84
5.6	Verteilung der Scores der SSQ Nausea Scale beider Interaktionsschnittstellen	84
5.7	Verteilung der Scores der SSQ Oculomotor Scale beider Interaktions- schnittstellen . . . . .	85
5.8	Verteilung der Scores der SSQ Disorientation Scale beider Interaktions- schnittstellen . . . . .	86
5.9	Ergebnisse (MW und SD) der Fragen zur Ablenkung durch die Interakti- onsschnittstellen. Die Bewertung erfolgte auf einer Skala zwischen 0 und 5 . . . . .	87
5.10	Benötigte Zeit zum Durchlaufen der Evaluationsabschnitte . . . . .	88
5.11	Darstellung der mittleren Interaktionsgeschwindigkeiten beider Interakti- onsschnittstellen nach Evaluationsabschnitt . . . . .	89
5.12	Häufigkeiten der Anzahl gemachter Fehler im Item Scanning . . . . .	91
5.13	Gründe und Häufigkeit von Fehlern beim Item Scanning . . . . .	92
5.14	Häufigkeiten der Anzahl gemachter Fehler beim Cartesian Scanning . . .	92
5.15	Gründe und Häufigkeit von Fehlern beim Cartesian Scanning . . . . .	93
5.16	Zusammenhang zwischen der Position des Elements im Sichtfeld und der Interaktionsgeschwindigkeit beim Cartesian Scanning . . . . .	94
5.17	Einsatz gezielter Kopfbewegungen der Nutzenden beim Cartesian Scanning	97
5.18	Einschätzungen der Teilnehmenden zur Scan Rate beider Interaktions- schnittstellen . . . . .	98
A.1	Evaluationsprotokoll Seite 1 . . . . .	131
A.2	Evaluationsprotokoll Seite 2 . . . . .	132
B.1	Ausschnitt aus der technischen Evaluation des Item Scannings. Aktives Scanning . . . . .	133
B.2	Ausschnitt aus der technischen Evaluation des Item Scannings. Geöffnetes Navigationsmenü . . . . .	134
B.3	Ausschnitt aus der technischen Evaluation des Item Scannings . . . . .	134
B.4	Ausschnitt aus der technischen Evaluation des Cartesian Scannings. Ak- tiver Navigationsmodus . . . . .	135
B.5	Ausschnitt aus der technischen Evaluation des Cartesian Scannings. Ak- tiver Selektionsmodus . . . . .	136
B.6	Ausschnitt aus der technischen Evaluation des Cartesian Scannings . . .	136
C.1	Ausschnitt aus dem inhaltsbasierten Szenario Code: Start des Szenarios .	137
C.2	Ausschnitt aus dem inhaltsbasierten Szenario Code: Raum Beispiel . . .	138
C.3	Ausschnitt aus dem inhaltsbasierten Szenario CodeGeöffnetes Bild mit versteckter Zahl . . . . .	138

---

C.4	Ausschnitt aus dem inhaltsbasierten Szenario Code: Finale Eingabe des Zahlencodes . . . . .	139
C.5	Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Start des Szenarios	140
C.6	Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Raum Beispiel . .	141
C.7	Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Geöffnetes Bild .	141
C.8	Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Finaler Schlüssel .	142
D.1	Genaue Verteilung der Bewertungen der Items des UEQ beim Item Scanning	143
D.2	Genaue Verteilung der Bewertungen der Items des UEQ beim Cartesian Scanning . . . . .	144

## Tabellenverzeichnis

5.1	Beschreibung der Testgruppen anhand der zuerst verwendeten Interaktionsschnittstelle und des zuerst durchgeführten inhaltsbasierten Szenarios .	78
5.2	UEQ-Ergebnisse des Item Scannings: Mittelwerte und Varianzen der Faktoren . . . . .	80
5.3	UEQ-Ergebnisse des Cartesian Scannings: Mittelwerte und Varianzen der Faktoren . . . . .	81
5.4	Häufigkeit der SSQ-Symptome beider Interaktionsschnittstellen. Die Symptome sind den Kategorien N: Nausea, O: Oculomotor und D: Disorientation zugeteilt . . . . .	83
5.5	Mittelwerte, Median und SD der SSQ-Ergebnisse nach Interaktionsschnittstelle und Kategorie . . . . .	83
5.6	Mittelwerte, Varianzen und Spannweiten der Interaktionsgeschwindigkeiten beider Interaktionsschnittstellen nach Evaluationsabschnitt in Sekunden . . . . .	89
5.7	Ergebnisse der linearen Regressionsanalyse (Mittelwert und Varianz der Regressionskoeffizienten b und Korrelationskoeffizienten r) . . . . .	90
5.8	Benötigte Scanning-Zyklen pro Interaktion beim Item Scanning . . . . .	90
5.9	Leere Eingaben beim Cartesian Scanning nach Evaluationsabschnitt . . .	93
5.10	Zusammenhang zwischen der Position des Elements im Sichtfeld und der Selektionsgeschwindigkeit beim Cartesian Scanning . . . . .	94
5.11	Zusammenhänge zwischen den Ergebnissen des SSQ und des SUS, des UEQ sowie der Interaktionsgeschwindigkeit . . . . .	95
5.12	Zusammenhang zwischen der Interaktionsgeschwindigkeit und der Bewertung der Usability und der UX . . . . .	96

## Listings

4.1 Vereinfachter Pseudocode der Coroutine IEnumerator ItemScanning des Item Scannings. Realisierung der schrittweisen Hervorhebung interaktiver Elemente . . . . .	59
4.2 Methode IsVisibleByCam: Umsetzung der Überprüfung, ob sich ein Element im Sichtfeld des Nutzens befindet . . . . .	60
4.3 Vereinfachter Pseudocode der Methode SwitchSelection des Item Scannings. Realisierung der Selektion des hervorgehobenen Elements . . . . .	61
4.4 Ausschnitt aus der Methode MoveLine zur Realisierung der Bewegung der Scan-Linien im Cartesian Scanning . . . . .	63
4.5 Methode Scanning im CartesianScanHandler: Überprüfung des aktuellen Zustands des Scannings . . . . .	64
4.6 Vereinfachter Pseudocode der Methode FindIntersection im CartesianScanHandler: Berechnung des Schnittpunktes der Scan-Linien . . . . .	65

# 1 Einleitung

Motorische Beeinträchtigungen umfassen den Verlust oder die Einschränkung der Fähigkeit, Muskeln oder Bewegungen zu kontrollieren, und betreffen zahlreiche Menschen weltweit. Die Ursachen sind vielfältig und reichen von Arthritis und Lähmungen bis hin zu neurologischen Erkrankungen wie Zerebralparese oder Verletzungen durch wiederholte Belastung [Yuan et al., 2011]. Solche Beeinträchtigungen wirken sich nicht nur auf die Mobilität der Personen aus, sondern können auch den Zugang zu digitalen Technologien erschweren. Die Gestaltung barrierefreier Systeme zielt entsprechend darauf ab, Produkte und Anwendungen so zu entwickeln, dass sie von einer möglichst breiten Nutzergruppe mit unterschiedlichen Anforderungen, Fähigkeiten und Fertigkeiten verwendet werden können. Dies umfasst auch die Berücksichtigung spezifischer Nutzungskontexte, einschließlich der Unterstützung assistiver Technologien [DIN EN ISO 9421-210:2020-03].

Virtual Reality (VR) bietet ein wachsendes Spektrum an Anwendungen, von Bildung und Medizin bis hin zu Entertainment und Training. Mit zunehmend erschwinglicher Hardware ist VR nicht mehr nur ein Nischenprodukt, sondern eine Technologie, die für den Consumer-Bereich zunehmend attraktiv wird. Diese Entwicklung eröffnet auch für Menschen mit gesundheitlichen Beeinträchtigungen neue Möglichkeiten. Insbesondere für Personen mit motorischen Beeinträchtigungen könnte VR eine Plattform bieten, die neue Formen der Interaktion und Teilhabe ermöglicht [Mott et al., 2020].

In Bezug auf die Barrierefreiheit weist VR jedoch eine gewisse Rückständigkeit im Vergleich zu anderen Technologien auf. Während gängige Betriebssysteme für Smartphones oder Computer bereits umfangreiche Tools zur Barrierefreiheit für Menschen mit unterschiedlichen Beeinträchtigungen integriert haben (vgl. z. B. [Apple Inc., 2024]), fehlen im Bereich VR weithin standardisierte Richtlinien und Werkzeuge [Ciccone et al., 2023]. Aus dieser Lücke in der Barrierefreiheit resultiert nicht nur eine technische Herausforderung, sondern auch die gesellschaftliche Notwendigkeit, sicherzustellen, dass technologische Fortschritte allen Menschen zugänglich gemacht werden.

## 1.1 Motivation

Die Entwicklung inklusiver VR-Anwendungen wird durch vier grundlegende Faktoren motiviert. (1) Besonders hervorzuheben ist das ethische und moralische Anliegen, technologische Fortschritte allen Menschen zugänglich zu machen. (2) Darüber hinaus hat sich gezeigt, dass VR-Technologie ein erhebliches Potenzial besitzt, Barrieren in den Bereichen Rehabilitation und Assistenztechnologien abzubauen und somit die Lebensqualität von Menschen zu verbessern. Dies bietet insbesondere für Menschen mit Beeinträchtigungen eine vielversprechende Perspektive. (3) Zusätzlich bringt die Ansprache einer größeren Nutzergruppe auch kommerzielle Vorteile mit sich, da sie den Zugang zu einem breiteren Markt und größeren Absatzchancen ermöglicht. (4) Schließlich führt ein barrierefreies Design häufig zu einer verbesserten Usability für alle Nutzenden. Dieser Aspekt ist besonders relevant im Hinblick auf situationale Einschränkungen, die auftreten, wenn äußere Umstände die Fähigkeit einer Person beeinträchtigen, bestimmte Aktivitäten auszuführen [Wobbrock et al., 2011]. Auch Personen ohne dauerhafte Beeinträchtigungen können von solchen situationalen Einschränkungen betroffen sein, etwa bei der Nutzung in beengten Räumen oder bei temporären Verletzungen. Ein inklusives Design, das die Bedürfnisse von Menschen mit dauerhaften Einschränkungen berücksichtigt, verbessert somit auch die Nutzungserfahrung für situativ eingeschränkte Personen [Dudley et al., 2023].

In der Forschung zur Barrierefreiheit in VR wird zunehmend betont, dass die individuellen Bedürfnisse der Endnutzenden im Zentrum der Entwicklung stehen müssen [Dombrowski et al., 2019]. Um für Menschen mit motorischen Beeinträchtigungen dieses Ziel zu erreichen, ist die Entwicklung binärer Interaktionsschnittstellen, die auf einfache Steuerungsmöglichkeiten reduziert sind, ein vielversprechender Ansatz. Diese erfordern minimale körperliche Anstrengung und ermöglichen damit einer breiten Nutzergruppe den Zugang zu VR-Anwendungen. Solche Schnittstellen bieten nicht nur eine Lösung für Menschen mit motorischen Beeinträchtigungen, sondern verbessern auch die Nutzbarkeit von Systemen in Situationen, in denen physische Bewegungen eingeschränkt oder unpraktisch sind, beispielsweise in beengten Räumen oder bei anderen situationalen Einschränkungen [Wobbrock et al., 2011].

## 1.2 Zielsetzung

Um Menschen mit motorischen Beeinträchtigungen die Nutzung der Anwendung Paneo-VR zu ermöglichen, besteht das Ziel dieser Arbeit in der Konzeption und prototypischen

Implementierung zweier binärer Interaktionsschnittstellen. Dadurch sollen mit PaneoVR erstellte VR-Trainings auch für diese Zielgruppe zugänglich und erlebbar werden.

Um eine möglichst barrierearme Nutzung zu gewährleisten, umfasst die Zielsetzung sowohl die Entwicklung eines geeigneten Interfaces als auch die systematische und detaillierte Konzeption der Interaktionen. Eine zentrale Anforderung dabei ist, dass alle erforderlichen Interaktionen mit dem System durch eine binäre Eingabe, bspw. über einen Schalter, ausgeführt werden können. Die implementierten Interaktionsformen sollen anschließend umfassend evaluiert werden und die gewonnenen Erkenntnisse dazu beitragen, Empfehlungen für die Entwicklung barrierearmer VR-Anwendungen abzuleiten.

Zur strukturierten Bearbeitung und Präzisierung dieser Zielsetzung wurden konkret folgende Forschungsfragen definiert, die im Rahmen dieser Arbeit beantwortet werden sollen:

- FF1: Wie können binäre Interaktionen in Virtual Reality gestaltet werden, um die Zugänglichkeit für Personen mit motorischen Beeinträchtigungen zu verbessern?
- FF2: Welche Herausforderungen treten bei der Implementierung binärer Interaktionsschnittstelle auf und wie können diese überwunden werden?
- FF3: Welche Unterschiede hinsichtlich Faktoren wie Effizienz, Robustheit, Usability und User Experience zeigen sich zwischen den entwickelten Ansätzen? Stimmen diese Ergebnisse mit den Erwartungen aus der Konzeption überein? Ist eine der implementierten Interaktionsformen geeigneter für die Verwendung in VR?
- FF4: Welche Empfehlungen können aus der Implementierung und Evaluation der binären Interaktionsschnittstellen in PaneoVR für die Entwicklung barrierearmer VR-Anwendungen abgeleitet werden?

## 1.3 Aufbau der Arbeit

Im Verlauf dieser Arbeit werden die aufgestellten Forschungsfragen systematisch in den einzelnen Kapiteln adressiert. Zunächst wird eine inhaltliche Grundlage geschaffen. Hierbei werden relevante theoretische und technologische Konzepte sowie der aktuelle Forschungsstand im Bereich barrierearmer Interaktionsschnittstellen und VR dargestellt (**Kapitel 2**). Dieser Überblick dient als Basis für die weitere Arbeit und ermöglicht ein umfassendes Verständnis der Thematik.



Darauffolgend wird im Rahmen der Konzeption die Forschungsfrage FF1 adressiert (**Kapitel 3**). Hier liegt der Fokus auf der Gestaltung der beiden binären Interaktionsschnittstellen. Zunächst wird ein Design Space definiert, der die Gestaltungsmöglichkeiten und Anforderungen beschreibt. Auf dieser Grundlage werden konkrete Ausprägungen für Interaktionsaufgaben und -komponenten erarbeitet. Diese Ausprägungen werden anschließend systematisch anhand relevanter Parameter bewertet, um die Designentscheidungen nachvollziehbar darzulegen und zu begründen. Abschließend werden die finalen Konzepte für die beiden Interaktionsschnittstellen abgeleitet.

Anschließend wird die technische Umsetzung der entwickelten Konzepte beschrieben (**Kapitel 4**). Neben der Vorstellung der Implementierung werden auch die Herausforderungen bei der Realisierung der binären Interaktionsschnittstellen thematisiert (FF2).

Im Anschluss werden zunächst Konzeption, Aufbau und Durchführung der Evaluation dargestellt, gefolgt von der Präsentation der Ergebnisse (**Kapitel 5**).

Diese Ergebnisse werden daraufhin umfassend eingeordnet und interpretiert (**Kapitel 6**). Dabei liegt ein besonderer Fokus auf dem Vergleich der beiden Interaktionsschnittstellen hinsichtlich Faktoren wie Effizienz, Robustheit, Usability, User Experience und Motion Sickness. Es wird untersucht, ob die erzielten Ergebnisse mit den im Vorfeld formulierten Erwartungen übereinstimmen und ob eine der entwickelten Interaktionsschnittstellen besser für den Einsatz in VR geeignet ist (FF3). Zudem werden Stärken, Schwächen und mögliche Verbesserungsansätze der Interaktionsschnittstellen aufgezeigt und Empfehlungen für die Entwicklung binärer Interaktionsschnittstellen abgeleitet (FF4).

Abschließend werden die zentralen Ergebnisse der Arbeit zusammengefasst und mögliche Perspektiven für weiterführende Forschung aufgezeigt (**Kapitel 7**).

## 2 Stand der Forschung

Dieses Kapitel legt die Grundlage für ein umfassendes Verständnis der Arbeit, indem relevante theoretische und technologische Konzepte sowie der aktuelle Forschungsstand dargestellt werden. Zunächst werden die Grundlagen von Virtual Reality erläutert, einschließlich der Definition von Virtual Reality und zentraler Begriffe wie Immersion und Presence. Zudem erfolgt eine Abgrenzung zu verwandten Begriffen und Konzepten wie Augmented Reality und Mixed Reality. Ergänzend werden grundlegende Informationen zu VR-Ausgabegeräten bereitgestellt, die eine technische Basis für das Verständnis der nachfolgenden Inhalte schaffen. Ein besonderer Schwerpunkt liegt auf dem Phänomen der Motion Sickness, da das Auftreten körperlicher Symptome eine häufige Herausforderung bei der Nutzung von VR-Anwendungen darstellt.

Im weiteren Verlauf widmet sich das Kapitel dem aktuellen Forschungsstand zur Barrierefreiheit in VR. Zunächst werden bestehende Richtlinien vorgestellt. Anschließend werden Barrieren beleuchtet, die Menschen mit motorischen Beeinträchtigungen bei der Nutzung von VR-Anwendungen erfahren, und bestehende Lösungsansätze zur Überwindung dieser Barrieren präsentiert. Dabei wird auch auf Bereiche mit weiterem Forschungsbedarf hingewiesen.

Ein weiterer Abschnitt behandelt binäre Interaktionsschnittstellen als spezifische Lösung zur Verbesserung der Zugänglichkeit für Menschen mit motorischen Beeinträchtigungen. Hierbei werden zunächst verschiedene Arten von Schaltern beschrieben und anschließend gängige Scanning-Verfahren erläutert, die die Nutzung dieser Schalter ermöglichen.

Die Konzepte Usability und User Experience bilden eine wichtige Grundlage für die Bewertung von VR-Anwendungen. In diesem Zusammenhang werden etablierte Messmethoden wie der System Usability Scale (SUS) und der User Experience Questionnaire (UEQ) vorgestellt.

Anschließend wird das Tool PaneoVR eingeführt, das als Grundlage für die Implementierung der im Rahmen dieser Arbeit erarbeiteten Konzepte dient. Eine Zusammenfassung des Kapitels bietet einen abschließenden Überblick über den Stand der Forschung und leitet über zur Konzeption.

Die Literaturrecherche für diese Arbeit erfolgte iterativ und kombinierte verschiedene Ansätze, um einen umfassenden Überblick über den aktuellen Forschungsstand zu gewinnen. Als Ausgangspunkt wurden Literaturübersichtsarbeiten herangezogen, um eine

fundierte Grundlage für die weitere Recherche zu schaffen. Aufbauend darauf wurde die in diesen Arbeiten referenzierte Literatur analysiert und auf ihre Relevanz für den spezifischen Rahmen dieser Arbeit überprüft. Bei der Identifikation hilfreicher Arbeiten wurde zudem geprüft, ob die jeweiligen Autor:innen weitere relevante Publikationen in diesem Forschungsfeld veröffentlicht haben. Darüber hinaus wurde eine ergänzende Recherche in gängigen wissenschaftlichen Suchmaschinen wie bspw. Google Scholar und SpringerLink, durchgeführt, um zusätzliche Arbeiten zu identifizieren. Dabei wurden diverse Suchtermini verwendet, um das Thema weiter erfassen und abdecken zu können.

## 2.1 Virtual Reality

Der Begriff "Virtual Reality (VR)" wird in der Literatur nicht einheitlich definiert [Wohlgenannt et al., 2020]. Es existieren verschiedene Definitionen, die unterschiedliche Aspekte der Technologie betonen. So wird VR beispielsweise von Berg and Vance [2017] als eine Sammlung von Technologien beschrieben, die es Menschen ermöglichen, immersiv eine Welt jenseits der Realität zu erleben. Nach Bowman and McMahan [2007] simuliert VR eine virtuelle Umgebung, die den Nutzenden das Gefühl vermittelt, *"dort zu sein"*. Jerald [2016] definiert VR hingegen als eine computergenerierte digitale Umgebung, die erlebt und interagiert werden kann, als ob diese Umgebung real wäre. In ihrer Definition von VR führen Wohlgenannt et al. [2020] die verschiedenen Ansätze zur Beschreibung von VR zusammen und präzisieren, dass VR immersive Technologien nutzt, um interaktive virtuelle Umgebungen oder virtuelle Welten zu simulieren, in die sich die Nutzenden subjektiv einbringen und in denen sie sich physisch anwesend fühlen.

Des Weiteren identifizieren Walsh and Pawlowski [2002] Immersion, Interaktivität und Presence als drei zentrale Konzepte von VR. Der Begriff der Presence wird definiert als *"die subjektive Erfahrung, sich an einem Ort oder in einer Umgebung zu befinden, auch wenn man physisch an einem anderen Ort ist"* [Witmer and Singer, 1998]. In einer virtuellen Umgebung entsteht das Gefühl von Presence insbesondere durch die Verlagerung der Aufmerksamkeit von der physischen auf die virtuelle Umgebung, wobei jedoch keine vollständige Ablösung von der physischen Realität notwendig ist [Witmer and Singer, 1998].

Im Gegensatz zur Presence ist der Begriff der Immersion weniger eindeutig definiert. Witmer and Singer [1998] beispielsweise definieren Immersion als einen psychologischen Zustand, in dem eine Person sich als umhüllt und in eine Umgebung eingebunden empfindet, die kontinuierliche Reize und Erfahrungen liefert. Sanchez-Vives and Slater [2005] hingegen definieren Immersion als die technische Fähigkeit eines Systems, eine umfassende und überzeugende Umgebung zu erschaffen, mit der die Nutzenden interagieren

können. Die Frage, ob Immersion als psychologischer Zustand oder als technische Eigenschaft des Systems zu verstehen ist, wirkt sich maßgeblich auf die Identifikation von Faktoren aus, die einen Einfluss auf die Immersion haben. Zu den wesentlichen Einflussfaktoren auf die Immersion zählen gemäß [Sanchez-Vives and Slater \[2005\]](#) unter anderem das *Field of View*, die Anzahl der simulierten sensorischen Systeme, die Qualität der Wiedergabe in jeder Sinnesmodalität, die Genauigkeit des Trackings, die Bildfrequenz und die Latenzzeit sowie die Übereinstimmung der simulierten sensorischen Daten mit der eigenen Körperwahrnehmung (Propriozeption). [Witmer and Singer \[1998\]](#) hingegen definieren die Isolation von der physischen Umgebung, die Wahrnehmung der eigenen Einbindung in die virtuelle Umgebung, natürliche Interaktionsmodi sowie die Wahrnehmung der Eigenbewegung als relevante Einflussfaktoren.

Neben diesen zentralen Konzepten betonen viele Definitionen die Bedeutung einer virtuellen Umgebung oder einer virtuellen Welt im Kontext von VR. Diese beiden Begriffe werden häufig in der Literatur verwendet, unterscheiden sich jedoch in ihrer Bedeutung. [Barfield et al. \[1995\]](#) definieren eine virtuelle Umgebung als softwarebasierte Darstellungen realer (oder imaginierter) Agenten, Objekte und Prozesse sowie eine Mensch-Computer-Schnittstelle zur Darstellung und Interaktion mit diesen Modellen. Eine virtuelle Welt hingegen ist eine spezifische Form einer (multi-user) virtuellen Umgebung, die gemeinsame, simulierte Räume bietet, die von den Bewohnern, die als Avatare repräsentiert werden, bewohnt und gestaltet werden [[Girvan, 2018](#)].

Bei der Betrachtung des Themenbereichs Virtual Reality erfolgt eine häufige Bezugnahme auf verwandte Konzepte wie Mixed Reality oder Augmented Reality. Zur Klassifizierung und Abgrenzung aller Technologien, die Realität und Virtualität vereinen, entwarfen [Milgram et al. \[1995\]](#) das sogenannte Reality-Virtuality Continuum (RVC) (vgl. [Abbildung 2.1](#)).

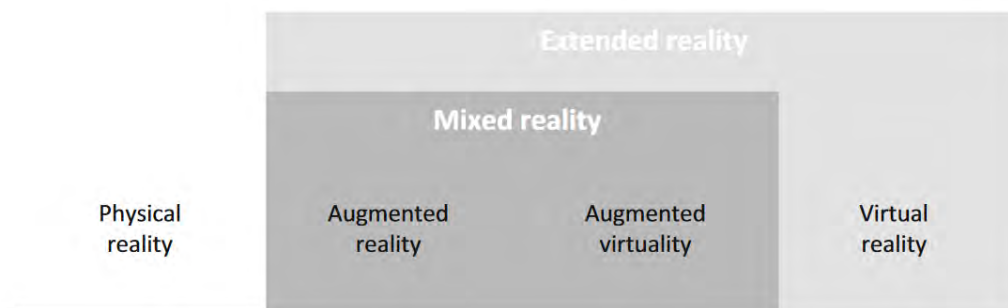


Abbildung 2.1: Reality-virtuality continuum nach [Milgram et al. \[1995\]](#) in der erweiterten Version von [Wohlgenannt et al. \[2020\]](#)

Auf dem RVC bilden die reale Umgebung und die virtuelle Umgebung die beiden Pole des Kontinuums. Die physikalische Realität setzt sich ausschließlich aus Elementen zusammen, welche durch eine Person direkt wahrgenommen werden können. Die virtuelle Umgebung, also die VR, besteht ausschließlich aus virtuellen Elementen. Der Nutzende wird dabei von der physikalischen Umgebung abgeschirmt, sodass er vollständig in die virtuelle Umgebung eintauchen kann. Der Bereich, der sich zwischen den beiden Polen des Kontinuums befindet, kombiniert virtuelle und physikalische Elemente und wird als Mixed Reality bezeichnet. Eine Umgebung, in der die physikalischen Elemente überwiegen, wird als Augmented Reality bezeichnet. Hierbei findet eine Erweiterung der physikalischen Realität durch virtuelle Elemente statt. Eine Erweiterung einer virtuellen Umgebung um physikalische Elemente wird demgegenüber als Augmented Virtuality bezeichnet. Ein weiterer Begriff, der in diesem Kontext häufig Verwendung findet, ist der Begriff der Extended Reality. Dieser wird häufig als Überbegriff verwendet und erfasst alle realen und virtuellen kombinierten Umgebungen und Mensch-Maschine-Interaktionen, die durch Computertechnologie und tragbare Geräte erzeugt werden [Fast-Berglund et al., 2018]. Wohlgenannt et al. [2020] haben daher das ursprüngliche RVC um diesen Begriff erweitert (vgl. [Abbildung 2.1](#)).

Um virtuelle Umgebungen für die Nutzenden erfahrbar zu machen, werden sowohl Ausgabegeräte benötigt, die die virtuelle Umgebung darstellen, als auch Eingabegeräte, die die Interaktion mit dieser Umgebung ermöglichen. Ziel der Ausgabegeräte ist es, die virtuelle Welt so darzustellen, dass sie von den Nutzenden möglichst ähnlich wie die reale Welt wahrgenommen werden kann. Zu den am weitesten verbreiteten Ausgabegeräten zählen insbesondere sogenannte *Head-Mounted Displays* (HMDs). Dabei handelt es sich um Displays, die direkt am Kopf getragen werden und sich direkt vor den Augen der Nutzenden befinden. HMDs verfügen in der Regel über integrierte Trackingsysteme oder werden mit externen Trackinglösungen kombiniert, um die Position und Blickrichtung des Kopfes zu erfassen. Dies ermöglicht eine kontinuierliche Anpassung der virtuellen Kamera an die aktuelle Position und Orientierung des HMDs [Dörner et al., 2019]. Darüber hinaus können aber z. B. auch Immersive Räume, auch CAVE (C-Automatic Virtual Enviroments) genannt, oder große Monitore Ausgabegeräte für VR-Anwendungen sein. Nach Somrak et al. [2019] können VR-Ausgabegeräte nach folgender Taxonomie beschrieben werden:

- **Tragbare Geräte:**

- Mit einem Smartphone als Anzeige- und Verarbeitungseinheit (z. B. Google Cardboard)
- Eigenständige tragbare VR-Geräte (z. B. Meta Quest 3)

- **Kabelgebundene Geräte:**

- Geräte mit einer Kabelverbindung zu einem leistungsstarken Computer (z. B. HTC Vive)
- Geräte, die an die Spielkonsole angeschlossen werden können (z. B. Sony PlayStation VR)

- **Immersive Räume - CAVE**

- **Große Monitore**

Die Interaktion mit der virtuellen Umgebung erfolgt in der Regel über spezielle Eingabegeräte. Hier kommen insbesondere VR-Controller zum Einsatz. Alternativ oder ergänzend kann die Eingabe durch direktes Tracking von Körperteilen der Nutzenden, wie z. B. Finger, Hände, Arme oder Augen, erfolgen. In diesem Fall werden bspw. Gesten erkannt und in der virtuellen Umgebung als Eingaben interpretiert [Dörner et al., 2019].

### 2.1.1 Motion Sickness

Das Erleben von VR kann bei Nutzenden Symptome auslösen, die denen von Motion Sickness aus anderen Bereichen ähneln [Somrak et al., 2019]. In der wissenschaftlichen Literatur werden neben dem Begriff *Motion Sickness* verschiedene Begriffe zur Beschreibung von unerwünschten Begleiterscheinungen virtueller Umgebungen verwendet. Dazu gehört der Begriff *Simulator Sickness*, der insbesondere in den frühen militärischen Flugsimulatoren geprägt wurde [Kennedy et al., 1993], sowie *Cybersickness*, der ursprünglich die Begleiterscheinungen virtueller Umgebungen allgemein beschrieb [McCauley and Sharkey, 1992]. In Studien mit HMDs findet zudem der Begriff *VR-Sickness* Verwendung (z. B. in [Kim et al., 2018]). In der Forschung zu virtuellen Umgebungen werden diese Begriffe oft synonym verwendet, wobei eine spezifische Abgrenzung nicht erfolgt [Saredakis et al., 2020]. Im weiteren Verlauf dieser Arbeit wird der Begriff *Motion Sickness* verwendet.

Das Spektrum der Symptome von Motion Sickness ist vielfältig. Zu den am häufigsten auftretenden Symptomen zählen unter anderem Unwohlsein, Apathie, Übelkeit, Schläfrigkeit, Desorientierung, Augenbelastung und Müdigkeit [Somrak et al., 2019]. In besonders schweren Fällen können darüber hinaus Symptome wie Erbrechen, Schweißausbrü-

che, übermäßiger Speichelfluss, Schwindel, Magenschmerzen und völlige Arbeitsunfähigkeit auftreten [Kennedy et al., 2010].

Die genauen Ursachen von Motion Sickness sind bislang nicht vollständig aufgeklärt. Eine der prominentesten Erklärungen liefert die Sensorische Konflikttheorie [Oman, 1990]. Diese besagt, dass die Symptome durch eine Diskrepanz zwischen visuellen, vestibulären und propriozeptiven Signalen entstehen. Diese Signale dienen in der Regel der Wahrnehmung der Ausrichtung und Bewegung des Körpers. In einer virtuellen Umgebung tritt jedoch häufig der Fall ein, dass der Körper visuell Bewegung wahrnimmt, während die vestibulären und propriozeptiven Systeme keine entsprechende Bewegung registrieren. Diese widersprüchlichen Signale führen zu sensorischen Diskrepanzen und können Motion Sickness auslösen.

Als weitere Erklärungsansätze werden die Vergiftungstheorie sowie die Posturale Instabilitätstheorie diskutiert. Die Vergiftungstheorie besagt, dass der menschliche Körper bei der Aufnahme von Gift eine Reaktion auslöst, die das visuelle und vestibuläre System beeinflusst. In virtuellen Umgebungen können Reize das visuelle und vestibuläre System irritieren, sodass der Körper fälschlicherweise glaubt, Gift aufgenommen zu haben. Als Konsequenz treten unangenehme Symptome und eine Übelkeitsreaktion auf [LaViola, 2000]. Die Posturale Instabilitätstheorie nach Riccio and Stoffregen [1991] basiert auf der Annahme, dass die Aufrechterhaltung der posturalen Stabilität in der Umgebung eines der Hauptziele menschlichen Verhaltens darstellt. Posturale Stabilität beschreibt den Zustand, in dem unkontrollierte Bewegungen der Wahrnehmungs- und Handlungssysteme minimiert werden. In virtuellen Umgebungen können abrupte oder signifikante Veränderungen zu einem Verlust der posturalen Kontrolle führen, insbesondere wenn die entsprechenden Kontrollstrategien noch nicht erlernt sind. Die Theorie besagt, dass anhaltende posturale Instabilität die Hauptursache für Motion Sickness ist. Je länger diese Instabilität andauert, desto schwerer sind die auftretenden Symptome.

Die Erfassung der Symptome von Motion Sickness erfolgt mittels subjektiver oder auch seltener objektiver Messmethoden [Somrak et al., 2019]. Zu diesen objektiven Verfahren zählt die Messung physiologischer Veränderungen im menschlichen Körper, beispielsweise von Herzfrequenz, Blinkrate, Hauttemperatur oder Hirnstromaktivität (EEG), um Motion Sickness quantitativ zu erfassen [Somrak et al., 2019]. Als Standard in der subjektiven Messung von Motion Sickness gilt der von Kennedy et al. [1993] entwickelte Simulator Sickness Questionnaire (SSQ) [Jerald, 2016].

Der SSQ basiert auf der Einschätzung der Teilnehmenden, die nach der Nutzung einer Anwendung den Schweregrad von 16 verschiedenen Symptomen bewerten. Die Bewertung erfolgt auf einer Skala von 0 (keine Wahrnehmung) bis 3 (starke Wahrnehmung). Die erfassten Symptome lassen sich in die drei übergeordnete Kategorien Nausea (Übel-

keit), Oculomotor disturbance (Okulomotorische Störungen) und Disorientation (Disorientierung) einteilen. Für jede dieser Kategorien wird ein separater Score berechnet, indem die Summen der Einzelbewertungen mit einem konstanten Gewichtungsfaktor multipliziert werden. Zusätzlich kann ein Gesamtscore berechnet werden, der alle drei Kategorien zusammenfasst. Höhere Gesamtscores und Scores in den einzelnen Kategorien deuten auf eine stärkere Wahrnehmung der entsprechenden Symptome hin und sind daher nicht erstrebenswert [Kennedy et al., 1993].

Zur Interpretation der Scores werden häufig die Grenzwerte nach Stanney et al. [1997] herangezogen. Diese beruhen auf Erkenntnissen, die aus einer umfangreichen Datenbasis von Militärpiloten gewonnen wurden. Die Grenzwerte der Gesamtscores liegen bei  $< 5$  (vernachlässigbar), 5-10 (minimal), 10-15 (signifikant) und 15-20 (besorgniserregend). Anwendungen, die Gesamtscores von über 20 erzeugen, werden demnach als problematisch eingestuft. Es ist jedoch zu beachten, dass diese Schwellenwerte aus Daten von Militärpiloten abgeleitet wurden, die möglicherweise weniger anfällig für Motion Sickness sind als die Allgemeinbevölkerung. In vielen Evaluationen von VR-Anwendungen werden daher häufig durchschnittliche Scores von über 20 beobachtet [Bimberg et al., 2020]. Hohe Scores können u. a. darauf zurückzuführen sein, dass aufgrund der Gewichtungsfaktoren bereits geringe Zunahmen einzelner Symptome ausreichen, um diesen Schwellenwert zu überschreiten. Daher sollte die Interpretation von VR-Anwendungen anhand der Grenzwerte von Stanney et al. [1997] mit kritischem Blick erfolgen.

Darüber hinaus ist eine Annahme des SSQ, dass die Teilnehmenden vor der Nutzung einer Anwendung völlig symptomfrei sind. In der Praxis ist dies jedoch schwer zu gewährleisten, da Symptome durch externe Faktoren wie Tageszeit, Anreise oder Stimmung beeinflusst werden können. Um den Einfluss der Anwendung auf das Auftreten von Motion Sickness genauer messen zu können, wird der SSQ häufig sowohl vor als auch nach der Nutzung durchgeführt, um anschließend die daraus resultierenden Unterschiede betrachten zu können [Bimberg et al., 2020]. Kennedy et al. [1993] kritisieren jedoch die geringe Zuverlässigkeit von Differenzwerten und schlagen vor, die Testpersonen stattdessen zu Beginn direkt zu fragen, ob sie sich in ihrem üblichen Gesundheitszustand befindet oder sich in irgendeiner Form krank fühlt. Personen, die sich nicht in ihrem üblichen Gesundheitszustand befinden sollten von der Analyse des SSQ folgend ausgeschlossen werden [Jerald, 2016].



## 2.2 Barrierefreiheit in VR

Die Forschung zur Barrierefreiheit in VR ist ein noch junges Forschungsgebiet, das besonders in den letzten Jahren zunehmend an Aufmerksamkeit gewonnen hat. Häufig konzentrieren sich die Studien auf eine bestimmte Form der Barriere oder Beeinträchtigung, wie Seh-, Hör-, motorische oder kognitive Beeinträchtigungen. Dabei werden die spezifischen Probleme und Herausforderungen identifiziert sowie entsprechende Konzepte und Lösungsansätze erarbeitet. Da sich diese Arbeit auf Menschen mit motorischen Beeinträchtigungen bezieht, liegt der Fokus im Folgenden auf dem aktuellen Forschungsstand in diesem Bereich.

Für die meisten Softwareprodukte wie Webseiten, Apps oder Spiele existieren Richtlinien zur Barrierefreiheit, die Entwickler:innen bei der Gestaltung neuer Produkte und Systeme als Orientierung dienen können, um Barrierefreiheit zu gewährleisten. Wie [Heilemann et al. \[2021\]](#) feststellen, berücksichtigen die meisten dieser Richtlinien jedoch kaum die spezifischen Anforderungen der Barrierefreiheit in virtuellen Umgebungen. Sie sind stattdessen häufig auf bestimmte Beeinträchtigungen oder Geräte ausgerichtet und basieren auf Interaktionsmodellen, die nicht direkt auf VR-Anwendungen übertragbar sind. Die Interaktionsweise mit VR-Systemen unterscheidet sich grundlegend von anderen Softwarearten, sodass bestehende Leitlinien nur begrenzt auf VR übertragen werden können. Ein Fortschritt in diesem Bereich ist der Bericht der XR Association (XRA) aus dem Oktober 2020, der explizite Empfehlungen für die Entwicklung barrierefreier VR- und AR-Anwendungen enthält. Darin wird betont, dass Barrierefreiheit von Beginn an in den Designprozess integriert werden sollte. Meta hat in den sogenannten Virtual Reality Check-Richtlinien [\[Meta, 2024\]](#) ebenfalls Abschnitte zur Barrierefreiheit aufgenommen. Die Umsetzung dieser Richtlinien ist jedoch nicht verpflichtend, um Produkte im Meta Quest Store anzubieten. Darüber hinaus umfassen sie derzeit lediglich neun Punkte. Zu den Empfehlungen zählen u. A. die Möglichkeit der Steuerung mit nur einer Hand, Anpassungsoptionen für Anzeigeeinstellungen wie Helligkeit und Kontrast sowie die Bereitstellung verschiedener Fortbewegungsstile. Ein weiterer Ansatz ist die Gründung von XR Access [\[XR Access - Virtual, Augmented & Mixed Reality for People with Disabilities, 2022\]](#), einer Initiative aus Partnern von Universitäten und der Industrie. Ziel von XR Access ist es, inklusives Design und Barrierefreiheit zu grundlegenden und selbstverständlichen Bestandteilen der XR-Entwicklung zu machen.

Darüber hinaus haben [Heilemann et al. \[2021\]](#) in einer umfassenden Übersichtsarbeit bestehende Richtlinien zur Barrierefreiheit von VR-Spielen analysiert und zusammengefasst. In Bezug auf motorische Einschränkungen werden mehrere Empfehlungen hinsichtlich der Eingabe gegeben. Es wird betont, dass eine Möglichkeit zur Neuzuweisung oder Neukonfiguration der Steuerungen vorhanden sein sollte. Außerdem sollten Handbewegungen wie Kneifen, Drehen oder starkes Greifen vermieden werden. Stattdessen

sollten einfache Steuerungsschemata angeboten werden. Des Weiteren wird empfohlen, die Kompatibilität mit assistiven Technologien wie Schaltern sicherzustellen und mehrere Eingabegeräte gleichzeitig zu unterstützen. Darüber hinaus sollte darauf geachtet werden, Auslöser für Motion Sickness zu vermeiden oder zumindest eine Option zur Deaktivierung solcher Auslöser bereitzustellen. Die Steuerung sollte nicht ausschließlich auf Bewegungserkennung, Kopfdotation oder bestimmte Körpertypen angewiesen sein.

### 2.2.1 Barrieren für Menschen mit motorischen Beeinträchtigungen bei der Nutzung von VR

Die Nutzung von VR-Anwendungen kann für Menschen mit motorischen Beeinträchtigungen eine Herausforderung darstellen. Bewegungsbasierte Interaktionen, die in VR häufig erforderlich sind, sind für viele Betroffene nur eingeschränkt oder gar nicht durchführbar. Diese Barrieren können nicht nur die Nutzbarkeit der Anwendungen einschränken, sondern den Zugang zu virtuellen Erlebnissen grundsätzlich erschweren. In der wissenschaftlichen Literatur wurden Probleme identifiziert, die im Zusammenhang mit motorischen Einschränkungen und der Interaktion mit virtuellen Umgebungen stehen. Im Folgenden wird der aktuelle Forschungsstand zu diesen Barrieren zusammengefasst. Insgesamt können drei Kategorien von Barrieren identifiziert werden.

#### 1. Einrichtung und Nutzung von VR-Ausgabegeräten

Ein grundlegendes Problem für Menschen mit motorischen Beeinträchtigungen bei der Nutzung von VR-Anwendungen stellt bereits die Einrichtung des Geräts dar, da dieser Prozess oft einen hohen körperlichen Einsatz oder auch feine motorische Fähigkeiten erfordert [Gerling and Spiel, 2021]. So kann bereits das Einlegen der Batterien in die VR-Controller, das Anschließen von Kabeln an den Computer oder das Festlegen des räumlichen Brennungsradius eine Barriere darstellen [Mott et al., 2020]. Auch das eigenständige Auf- und Absetzen eines VR-HMD sowie das Einstellen des Kopfbandes kann für Personen mit eingeschränkter Beweglichkeit der Arme oder des Oberkörpers eine erhebliche Herausforderung darstellen [Mott et al., 2020]. Eine zusätzliche Barriere stellen HMDs mit Kabelverbindung zum Computer dar. Dies gilt sowohl für die Einrichtung des Gerätes als auch für die Nutzung. So kann es z. B. vorkommen, dass Personen die einen Rollstuhl benutzen, versehentlich über die Kabel fahren oder sich diese in den Reifen verfangen [Mott et al., 2020, Wong et al., 2017].

## 2. Annahmen in Bezug auf den Körper

Die umfangreichen Anforderungen an die körperliche Beteiligung, die VR-Technologie stellt, können Barrieren für Menschen schaffen, die ihren Körper auf andere Weise in das System einbringen [Gerling and Spiel, 2021]. Die geforderten Körperbewegungen basieren häufig auf Fähigkeiten und Funktionen von Menschen ohne Behinderung. Dazu gehören bspw. die Nutzung im Stehen sowie die Verwendung von Gesten und beiden Händen zur Interaktion mit virtuellen Objekten [Wong et al., 2017]. VR-Anwendungen haben oft Probleme, Körper zu verfolgen und zu erfassen, die vom „Standard“ abweichen [Wong et al., 2017]. Darüber hinaus basieren auch anwendungsinterne Anforderungen an Energie und Ausdauer auf den Fähigkeiten von Menschen ohne Behinderungen [Wong et al., 2017]. Dies kann bei Menschen mit körperlichen Einschränkungen zu körperlicher, geistiger und zeitlicher Ermüdung führen. Als Folge können Schmerzen auftreten oder bereits bestehende körperliche Symptome sich verschlimmern [Creed et al., 2024a]. Personen, die unter Erschöpfung oder chronischen Schmerzen leiden, könnten dadurch unter Umständen gänzlich von der Nutzung ausgeschlossen werden [Wong et al., 2017]. Ein weiteres Problem, das aus diesen Annahmen resultiert, ist, dass VR-Anwendungen in der Regel die Integration von assistiven Technologien nicht unterstützen bzw. mit diesen nicht kompatibel sind. Dies kann dazu führen, dass Nutzende weniger (Selbst-)Sicherheit bei der Anwendung empfinden [Creed et al., 2024a].

## 3. Interaktion mit VR-Controllern

Die Nutzung gängiger VR-Controller stellt für Personen mit motorischen Beeinträchtigungen eine erhebliche Herausforderung dar. Aktuelle VR-Systeme sind in erster Linie für Personen konzipiert, die zwei Hände verwenden, lange stehen und virtuelle Objekte mit Hilfe der Hände manipulieren können. Diese Anforderungen schließen viele Menschen mit motorischen Beeinträchtigungen grundsätzlich von der Nutzung aus [Dombrowski et al., 2019]. Zusätzlich wird in VR-Anwendungen oftmals der gesamte Bewegungsradius der Nutzenden ausgenutzt und es werden Eingaben über Kopfhöhe sowie Bewegungen und Drehungen des Oberkörpers gefordert [Gerling and Spiel, 2021]. Insbesondere die Nutzung von VR-Controller setzt dabei voraus, dass Nutzende eine oder beide Hände mit vollständiger Finger-, Handgelenks- und Armbeweglichkeit zur Verfügung haben [Mott et al., 2019]. Daher haben viele Nutzende mit motorischen Beeinträchtigungen Schwierigkeiten, die Tasten auf den Controllern zu erreichen, zu drücken und gedrückt zu halten, insbesondere wenn gleichzeitig mehrere Tasten betätigt werden müssen [Mott et al., 2020]. Insgesamt werden VR-Controller in der Regel nach dem Prinzip „One size fits all“ konzipiert sind, wodurch es an Flexibilität und Anpassungsfähigkeit für Nutzende mangelt [Creed et al., 2024a]. Zusätzlich müssen bei vielen gängigen VR-Brillen die Controller stets im Sichtfeld der Headset-Kameras bleiben, damit ihre Position korrekt erfasst werden kann, was eine weitere Herausforderung darstellt [Mott et al., 2020]. Für

Personen, die nur wenig bis gar keine Beweglichkeit der Arme oder Hände haben, sind VR-Controller komplett unzugänglich [Mott et al., 2020]. Darüber hinaus setzt der Einsatz von VR-Controllern voraus, dass die Nutzenden ihre Körperbewegungen jederzeit kontrollieren und daher bspw. präzise Selektionen vornehmen können. Dies ist jedoch nicht bei allen Menschen der Fall. Daher können unwillkürliche Körperbewegungen eine Herausforderung bei der Interaktion mit VR-Controllern darstellen. Diese Barriere gilt auch für Nutzende mit unwillkürlichen Augenbewegungen und ist besonders problematisch für VR-Anwendungen, bei denen konstante und zielgerichtete Augenbewegungen für die Interaktion erforderlich sind [Creed et al., 2024a].

### 2.2.2 Lösungsansätze zur Überwindung von Barrieren

In der Literatur werden bereits einige Ansätze zur Verringerung der genannten Barrieren vorgestellt. So argumentieren Mott et al. [2020] bzgl. der Herausforderungen hinsichtlich gängiger HMDs, dass Drehknöpfe zum Anpassen des Kopfbandes näher an der Vorderseite des Headsets positioniert werden sollten, um die Erreichbarkeit zu verbessern. Eine automatische Anpassung des Kopfbandes anstelle einer manuellen Einstellung wird ebenfalls als hilfreich bewertet. Des Weiteren könnte die Verwendung eines kabellosen HMD eine einfache Lösung darstellen, um die Bewegungsfreiheit der Nutzenden zu erhöhen. Hinsichtlich der Interaktionsmöglichkeiten wird die Bereitstellung einer größeren Auswahl an Optionen und Anpassungsmöglichkeiten für Controller vorgeschlagen, um den unterschiedlichen Bedürfnissen der Nutzenden gerecht zu werden. Darüber hinaus könnten Eingabemethoden wie Sprachsteuerung und Blicksteuerung eine zugänglichere Alternative zu den herkömmlichen VR-Controllern darstellen. Des Weiteren wird vorgeschlagen, dass Nutzende die Möglichkeit haben sollten, Interaktionsmethoden oder Steuerungen neu zu konfigurieren, um eine individuellere Nutzung zu ermöglichen. Dombrowski et al. [2019] führen aus, dass weitere potenzielle Verbesserungen die Steuerung von VR-Anwendungen mit alternativen Eingabegeräten wie Schaltern und die Verwendung von sich automatisch anpassenden Einstellknöpfen an VR-HMDs umfassen könnten. Darüber hinaus sollten Anwendungen und Eingabegeräte in der Lage sein, ungleichmäßige Eingaben zu tolerieren und versuchen, die Absicht des Nutzenden zu interpretieren. Das Design der Anwendung und die Interaktionsformen sollten effizient und komfortabel sein und mit minimaler Ermüdung genutzt werden können.

Neben diesen theoretischen Lösungsansätzen wurden in der Forschung bisher nur wenige praktische Untersuchungen hinsichtlich alternativer Interaktionsmethoden durchgeführt. Bei diesen Untersuchungen liegt der Fokus oftmals auf Sprach- und Blicksteuerung. Beispielsweise untersuchten und verglichen Minakata et al. [2019] alternative Eingabeoptionen wie Kopf-, Blick- und Fußsteuerung hinsichtlich mentaler und physischer

Anstrengung sowie Präzision. Wang et al. [2018] entwickelten und evaluierten ein Eingabesystem, das auf Gesichtsausdrücken und Augenbewegungen basiert. L. Franz et al. [2021] entwickelten Nearmi, ein Interaktionssystem, das darauf abzielt, die Bewegungen des Oberkörpers und des Kopfes zu reduzieren und das Auffinden und Erreichen von Points of Interest in der virtuellen Umgebung zu erleichtern. Dieses Framework basiert jedoch weiterhin überwiegend auf Eingaben mittels VR-Controller. Valakou et al. [2024] entwickeln ein XR Accessibility Framework, das es Entwickler:innen erleichtern soll, verschiedene Barrierefreiheitsfunktionen in Anwendungen zu integrieren. Das Framework bietet unter anderem individuelle Texteneinstellungen, interaktive Objektbeschreibungen und dynamisches Hervorheben von Kanten bei 3D-Objekten. Für Menschen mit motorischen Beeinträchtigungen, die Schwierigkeiten mit Point-and-Select Interaktionen haben, bietet das Framework zusätzlich eine Interaktionsmethode mittels sequentiellen Scanning. Dabei werden alle interaktiven Elemente in der Szene in einer hierarchischen Reihenfolge durchlaufen und hervorgehoben und können entsprechend ausgewählt werden.

Obwohl diese wenigen Ansätze bereits existieren, hat sich die Forschung bisher hauptsächlich auf die Identifizierung von Barrieren bei der Nutzung von VR-Anwendungen konzentriert. Insgesamt gibt es nur wenige Arbeiten zur Gestaltung und Entwicklung inklusiver VR-Anwendungen. Aus diesem Grund haben Creed et al. [2024b] eine Forschungsagenda für diesen Bereich erarbeitet, die die Kernbereiche hervorhebt, in denen weitere Forschung dringend erforderlich ist, um inklusivere AR- und VR-Anwendungen zu ermöglichen. In Bezug auf Menschen mit motorischen Beeinträchtigungen wurden z. B. die Erforschung alternativer Eingabemöglichkeiten (z. B. durch Blickverfolgung und Sprachsteuerung), die Untersuchung von Methoden zum Herausfiltern ungewollter und seltener auftretender Bewegungen (z. B. Stürzen, Taumeln, subtile repetitive Bewegungen wie Tremor) sowie die Untersuchung alternativer Ansätze für das Headset-Design, weitere Forschung zu anpassbaren Geräteschnittstellen und die Entwicklung von Hardwaresystemen, die gängige assistive Technologien integrieren und unterstützen können, identifiziert.

## 2.3 Binäre Interaktionsschnittstellen

Binäre Interaktionsschnittstellen bieten eine grundlegende Möglichkeit der Systemsteuerung durch einfache Eingaben. Dabei wird zwischen zwei Zuständen unterschieden, z. B. „An“ und „Aus“ oder „Gedrückt“ und „Nicht gedrückt“. Derartige Schnittstellen basieren auf der Verwendung von Schaltern (eng. *Switches*), die als assistive Technologien für Menschen mit motorischen Beeinträchtigungen entwickelt wurden, um die Verwendung

von Mäusen, Tastaturen, Controllern oder Joysticks zu ersetzen, da diese oft schwer zugänglich sind. Schalter können prinzipiell mit jedem Körperteil bedient werden, das in der Lage ist, konsistente und willkürliche Bewegungen auszuführen. Die Art des verwendeten Schalters kann daher an die individuellen körperlichen Voraussetzungen einer Person angepasst werden. Während manche Menschen mit schweren motorischen Beeinträchtigungen nur einen Schalter bedienen können, sind andere in der Lage, mehrere Schalter parallel zu bedienen. Schalter können nach der Art der erforderlichen Aktion kategorisiert werden, wie z. B. Drücken, Ziehen, Kneifen oder durch Ein- und Ausatmen [Yuan et al., 2011, Weber and Demchak, 1996].

Im Folgenden werden verschiedene Arten von Schaltern sowie gängige Scanmethoden vorgestellt, die den Einsatz von Schaltern ermöglichen.

### 2.3.1 Arten von Schaltern

Schalter können auf verschiedene Weise aktiviert werden. Cook and Polgar [2015b] beschreiben drei grundlegende Möglichkeiten, wie Nutzende ein Signal an die Interaktionsschnittstelle senden kann: Bewegung, Atmung und Phonation.

#### Schalter basierend auf Bewegung

Der häufigste Mechanismus zur Steuerung von Schaltern ist die Erkennung von Körperbewegungen. Dies kann durch verschiedene Arten von Schaltern erreicht werden.

- *Mechanische Schalter:* Mechanische Schalter reagieren auf Kraft, die durch Körperbewegungen erzeugt wird. Beispiele sind klassische Druckknöpfe oder Wippschalter. Einige Schalter erfordern eine sichtbare Bewegung (mechanische Verschiebung), während andere nur auf die ausgeübte Kraft reagieren. Mechanische Schalter sind vielseitig einsetzbar, aber aufgrund des erforderlichen Kraftaufwands möglicherweise nicht für Personen mit Muskelschwäche geeignet.
- *Elektromagnetische Schalter:* Diese Schalter werden durch elektromagnetische Energie wie Licht oder Radiowellen aktiviert. Sie erfordern keinen direkten Körperkontakt und bieten somit eine berührungslose Steuerung. Ein Beispiel für elektromagnetische Schalter sind Lichtschranken.
- *Elektrische Schalter:* Elektrische Schalter nutzen elektrische Signale, die vom menschlichen Körper erzeugt werden. Elektromyographische Signale (EMG), die bei Muskelkontraktionen entstehen, können mit Hilfe von Elektroden auf der Haut gemes-

sen und zur Eingabe verwendet werden. Diese Schalter eignen sich besonders für Menschen mit Muskel- oder Bewegungsstörungen, da sie ohne mechanischen Aufwand bedient werden können. Auch Augenbewegungen und Hirnströme (z. B. in Brain-Computer-Interfaces, BCIs) können mit solchen Schaltern gemessen und zur Steuerung verwendet werden.

- *Proximity-Schalter*: Proximity-Schalter registrieren Bewegungen ohne physischen Kontakt zum Körper. Dazu gehören z. B. Systeme zur Gestenerkennung oder wärmeempfindliche Schalter.

### Schalter basierend auf Atmung

Eine weitere Möglichkeit zur Erzeugung von Eingangssignalen ist die Nutzung der Atmung. Pneumatische Schalter, wie das weit verbreitete Sip-and-Puff-System, basieren auf der Erfassung von Luftdruck. Nutzende steuern das System durch Saugen (*sip*) oder Blasen (*puff*) in ein Röhrchen. Diese Technologie ist besonders nützlich für Menschen mit stark eingeschränkter Beweglichkeit.

### Schalter basierend auf Phonation

Diese Kategorie umfasst Schalter, die auf Geräusche oder Sprache reagieren. Akustische Schalter können z. B. durch Klickgeräusche, regelmäßige Vokalisierungen oder konsistente Sprachlaute aktiviert werden. Diese Technologie eignet sich vor allem für Personen, die andere Eingabemethoden nicht nutzen können, aber über eine kontrollierte Lautproduktion verfügen.

## 2.3.2 Scanning-Verfahren

Scanning ist eine Eingabemethode, bei der dem Nutzenden eine Auswahl an Selektionsoptionen sequenziell auf einem Display präsentiert wird. Sobald das gewünschte Element angezeigt wird, erfolgt eine Interaktion durch den Nutzenden, um dieses auszuwählen. Typischerweise wird dafür ein einzelner Schalter oder ein Array aus zwei oder mehr Schaltern verwendet. Scanning ermöglicht eine interaktive Steuerung, ohne dass umfangreiche motorische Fähigkeiten erforderlich sind [Cook and Polgar, 2015a].

Es existieren diverse Scanning-Verfahren, die sich in der Art und Weise der Durchlaufung der Auswahlmöglichkeiten voneinander unterscheiden. Die gängigsten Verfahren sind dabei das *Item Scanning*, welches sich in den Unterarten *Automatic*, *Step* und *Inverse Scanning* differenziert, sowie das *Continuous Cartesian Scanning*.



Beim Item Scanning erfolgt eine sukzessive Hervorhebung einzelner Elemente. Die Selektion eines gewünschten Elements erfolgt durch Betätigung eines Schalters, sofern das entsprechende Zielelement hervorgehoben ist [Steriadis and Constantinou, 2003].

### **Automatic Item Scanning**

Beim Automatic Item Scanning erfolgt die Hervorhebung der Elemente automatisch. Die Hervorhebung bleibt bei jedem Element für ein vordefiniertes Zeitintervall stehen. Wird während der Hervorhebung eines Elementes eine Eingabe vorgenommen, wird das hervorgehobene Element ausgewählt. Andernfalls wandert die Hervorhebung zum nächsten Element [Steriadis and Constantinou, 2003]. Der wesentliche Vorteil des Automatic Item Scannings besteht in der Möglichkeit, die Interaktionen des Nutzenden auf ein Minimum zu reduzieren. Allerdings ist das Verfahren relativ langsam. Des Weiteren erfordert es ein hohes Maß an sensorischer und kognitiver Aufmerksamkeit, um die Reihenfolge der Hervorhebung zu beobachten und zu verfolgen [Cook and Polgar, 2015a].

### **Step Item Scanning**

Beim Step Item Scanning erfolgt die Hervorhebung der Elemente nicht automatisch, sondern durch wiederholte Aktivierung des Schalters durch den Nutzenden. Mit jeder Aktivierung erfolgt ein Sprung zur nächsten Auswahlmöglichkeit, bis das gewünschte Element erreicht ist. Die Selektion kann entweder mittels eines zusätzlichen Schalters getroffen werden, oder das System akzeptiert die Auswahl nach Ablauf einer festgelegten Zeitspanne (*Dwell Selection*). Ein wesentlicher Vorteil des Step Item Scannings besteht in der Möglichkeit, dass die Nutzenden die Geschwindigkeit der Hervorhebung selbst steuern können. Dadurch ist das Verfahren insgesamt schneller als das Automatic Item Scanning. Allerdings ist zu berücksichtigen, dass dieses Vorgehen eine wiederholte Aktivierung des Schalters erfordert, was mit einer motorischen Ermüdung einhergehen kann [Cook and Polgar, 2015a].

### **Inverse Item Scanning**

Die Initiierung des Inverse Scanning erfolgt durch Aktivierung und kontinuierliches Halten des Schalters. Solange die Betätigung des Schalters aufrechterhalten wird, erfolgt der Scan der Elemente. Sobald das gewünschte Element hervorgehoben wird, kann die Person den Schalter loslassen, um die Auswahl zu bestätigen. Inverse Scanning erfordert im Vergleich zum Step Scanning eine geringere Anzahl an Aktivierungen des Schalters, ist jedoch mit einer höheren sensorischen und kognitiven Aufmerksamkeit verbunden, da das Scanning kontinuierlich überwacht werden muss. Insbesondere für Personen, die einen erhöhten Zeitaufwand für die Initiierung und Ausführung von Bewegungen benötigen, kann dieses Verfahren von Vorteil sein [Cook and Polgar, 2015a].



### Continuous Cartesian Scanning

Im Rahmen des Continuous Cartesian Scannings erfolgt das Scannen entlang orthogonaler Richtungen, bis das jeweilige Ziel erreicht ist. Eine horizontale Scanlinie beginnt mit einem kontinuierlichen Scan vom oberen Rand des Sichtfelds nach unten. Sobald die Scanlinie das Zielobjekt kreuzt, aktiviert der Nutzende den Schalter, wodurch die horizontale Linie an ihrer Position sichtbar bleibt. In der Folge wird eine vertikale Scanlinie initiiert, welche kontinuierlich von der linken Seite des Sichtfelds nach rechts scannt. Wenn auch die vertikale Scanlinie das Zielobjekt kreuzt, aktiviert der Nutzende den Schalter, um dieses Objekt auszuwählen [Blackstien-Adler et al., 2004].

Der wesentliche Vorteil der Nutzung eines Scanning-Verfahrens liegt darin, dass eine Selektion von Elementen mit minimalem motorischen Aufwand realisierbar ist. Allerdings setzt die Nutzung dieser Methode gute visuelle Fähigkeiten, hohe Aufmerksamkeit sowie die Fähigkeit voraus, die Abfolge der Auswahlmöglichkeiten zu erfassen, insbesondere im Kontext des Item Scannings. Des Weiteren ist Scanning im Vergleich zu anderen Eingabemethoden relativ langsam. Um diesen Nachteil zu kompensieren, wurden verschiedene Ansätze zur Beschleunigung des Scannings entwickelt. Eine zentrale Methode ist das sogenannte Rate Enhancement beim Item Scanning, bei dem Elemente zu Gruppen zusammengefasst und selektiert werden können. Dadurch kann der Auswahlprozess signifikant beschleunigt werden [Cook and Polgar, 2015a]. Darüber hinaus stellt die Festlegung der optimalen Scangeschwindigkeit (Scan Rate) eine der größten Herausforderungen beim Scanning dar [Cook and Polgar, 2015a]. Beim Automatic und Inverse Item Scanning beschreibt die Scan Rate die Zeitspanne, während der ein Element für die Auswahl durch den Nutzenden hervorgehoben ist [Richard Simpson and LoPresti, 2007]. Beim Continuous Cartesian Scanning ist die Scan Rate definiert als die Strecke, die der Scan in einer bestimmten Zeit zurücklegt [Blackstien-Adler et al., 2004]. Eine zu hohe Scan Rate kann dazu führen, dass Nutzende Schwierigkeiten haben, im richtigen Moment eine Auswahl zu treffen und dadurch viele Fehler machen oder das System gar nicht nutzen können. Eine zu niedrige Scan Rate hingegen verzögert den Eingabeprozess unnötig [Richard Simpson and LoPresti, 2007].

## 2.4 Usability und User Experience

**Usability** Der Begriff *Usability*, auch im Deutschen auch als Gebrauchstauglichkeit übersetzt, wird in der [DIN EN ISO 9241-11:2018-11](#) wie folgt definiert:

*„Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ DIN EN ISO 9241-11:2018-11.*

In den Anmerkungen zu dieser Definition wird verdeutlicht, dass die Gebrauchstauglichkeit immer in Bezug auf die Kombination aus spezifischen Benutzern, spezifischen Zielen und dem spezifischen Nutzungskontext betrachtet werden muss. Dies bedeutet, dass die Gebrauchstauglichkeit eines Systems, eines Produkts oder einer Dienstleistung in Abhängigkeit von den Zielen, der Art der Nutzenden und weiteren Aspekten des Nutzungskontextes unterschiedlich sein kann. Darüber hinaus wird präzisiert, dass sich der Begriff *Gebrauchstauglichkeit* nicht nur auf die Qualität eines Systems bezieht, sondern auch als Qualifizierungsmerkmal verwendet wird, um Kompetenzen, Aktivitäten oder Methoden zur Verbesserung der Gebrauchstauglichkeit zu beschreiben.

Im weiteren Verlauf dieser Arbeit wird der Begriff *Usability* verwendet, da dieser Begriff auch im Deutschen überwiegend verwendet wird.

Die Messung der Usability basiert direkt auf den drei in der Definition verankerten Faktoren Effektivität, Effizienz und Zufriedenheit. Diese Merkmale können durch eine Vielzahl von Methoden erfasst werden, darunter Usability-Tests und speziell entwickelte Fragebögen. Ein bekanntes und weit verbreitetes Instrument zur Messung der Usability ist die System Usability Scale (SUS), die 1986 von John Brooke entwickelt und zehn Jahre später veröffentlicht wurde [Brooke, 1996].

Die SUS besteht aus lediglich zehn Items und ist daher schnell und einfach durchzuführen. Anstelle einer differenzierten Bewertung einzelner Usability-Faktoren liefert die SUS einen Gesamtwert im Bereich von 0 bis 100. Zur Interpretation des SUS Scores werden üblicherweise die Ergebnisse von Aaron Bangor and Miller [2008] herangezogen (vgl. Abbildung 2.2). Demnach liegt der durchschnittliche SUS Score bei 68. Scores unter 50 gelten als inakzeptabel, während Produkte mit einem Score von 70 oder höher als akzeptabel eingestuft werden. Bessere Produkte erreichen Werte zwischen 70 und 80, während herausragende Systeme Scores von über 90 erreichen. Produkte mit einem Score unter 70 sollten genauer überprüft und verbessert werden.

## User Experience

Neben der Usability ist auch die User Experience (UX) ein relevantes Konzept im Human Computer Interaction (HCI). Die DIN EN ISO 9421-210:2020-03 definiert UX als:

*„Wahrnehmungen und Reaktionen einer Person, die aus der tatsächlichen und/oder der erwarteten Benutzung eines Systems, eines Produkts oder einer Dienstleistung resultieren.“ DIN EN ISO 9421-210:2020-03.*

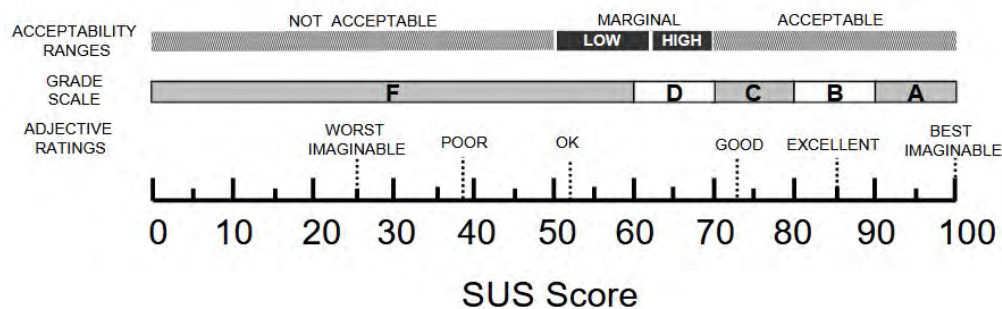


Abbildung 2.2: Einstufung der SUS-Scores nach Aaron Bangor and Miller [2008]

Die UX ist demnach essentiell, um nachvollziehen zu können, wie ein interaktives Produkt oder System von Nutzenden wahrgenommen wird. Damit hat sie gleichzeitig einen erheblichen Einfluss auf den Erfolg eines Produktes oder Systems. Fallen die Reaktionen und Wahrnehmungen der Nutzenden negativ aus, wird sich dies auf den Erfolg des Produkts auswirken. Die Definition der DIN-Norm bietet, im Gegensatz zur Usability, jedoch keine Kriterien zur Messung. Um die UX messbar zu machen, wird das Modell der pragmatischen und hedonischen Qualität nach Hassenzahl [2008] verwendet. Danach setzt sich die UX aus der pragmatischen und der hedonischen Qualität zusammen. Unter pragmatischer Qualität wird die aufgabenbezogene Qualität eines Produktes verstanden. Ein Produkt besitzt pragmatische Qualität, wenn es Nutzende dabei unterstützt, Aufgaben effizient und effektiv zu erledigen. Die hedonische Qualität hingegen bezieht sich auf nicht aufgabenbezogene Aspekte eines Produktes. Dazu zählen beispielsweise Aspekte wie Einzigartigkeit und Originalität. Viele etablierte Methoden zur Evaluation der UX basieren auf dieser Unterscheidung von Qualitäten.

Die meisten Messmethoden der UX wurden für 2D-Anwendungen, wie Webseiten, Apps o. Ä. erstellt und validiert worden. Für VR-Anwendungen gibt es derzeit noch keine etablierten oder standardisierten Methoden zur Messung der UX [Alexandrovsky et al., 2021]. Am häufigsten werden jedoch quantitative Methoden zur Messung bei VR-Anwendungen eingesetzt [Yong Min Kim and Yun, 2020]. Ein häufig eingesetzter Fragebogen ist der User Experience Questionnaire (UEQ) [Laugwitz et al., 2008]. Dieser besteht aus 26 bipolaren Items und verwendet eine 7er Likert-Skala. Gemessen werden die UX-Faktoren Attraktivität, Druckschaubarkeit, Effizienz, Steuerbarkeit, Stimulation und Originalität. Der UEQ steht in über 20 Sprachen zur Verfügung. Der Fragebogen sowie ein spezifisches Auswertungstool kann kostenlos genutzt werden.

## 2.5 PaneoVR

Das PaneoVR-Tool wurde entwickelt, um immersive 360°-Videotrainings bereitzustellen und die Erstellung virtueller Lehrinhalte für Bildungszwecke zu vereinfachen. Es handelt sich dabei um eine Web- und VR-Anwendung, die es Nutzenden ermöglicht, ohne tiefgehende Fachkenntnisse in Informatik oder Medientechnik ansprechende VR-Szenarien zu erstellen und zu nutzen [Meyer, 2025b]. Das Konzept wurde im Rahmen des vom Bundesministerium für Bildung und Forschung geförderten Projekts *ViRDiPA* [Hochschule Bielefeld, 2023] entwickelt. Dieses Projekt lief vom 01.03.2020 bis zum 31.08.2023 und hatte das Ziel, VR als innovative Lehrmethode in der Pflegeausbildung zu etablieren. Die Entwicklung erfolgt in Zusammenarbeit mit dem *Mixed Reality Lab* der Hochschule Emden/Leer.

PaneoVR richtet sich an Personen, Unternehmen oder Einrichtungen, die eigenständig virtuelle Trainings ohne den Aufwand und die Kosten aufwendiger 3D-Designs und Animationen erstellen möchten. Stattdessen nutzt das Tool 360°-Videos, die eine realitätsgetreue Darstellung auf einfache Weise ermöglichen. Diese Methode erlaubt es, Szenarien von einfachen Raumtouren bis hin zu komplexen Rollenspiel-Erlebnissen zu erstellen. Nutzende stehen innerhalb von Video- oder Fotosphären und können durch Interaktion mit digitalen Elementen Einfluss auf die Umgebung nehmen, was eine dynamische und immersive Erfahrung schafft [Meyer, 2025b].

Das Tool besteht aus drei wesentlichen Komponenten:

- **Diagrammeditor/Webeditor:** Der nodebasierte Diagrammeditor bildet die zentrale Plattform zur Erstellung der VR-Szenarien. Nutzende können 360°-Videos, Bilder und Audiodateien hochladen und daraus interaktive Szenen gestalten und zu Szenarien verknüpfen.
- **VR-Anwendung/PaneoVR App:** Die PaneoVR App läuft auf allen gängigen VR-Brillen und ermöglicht es, die erstellten Szenarien direkt in Virtual Reality zu durchlaufen. Zusätzlich bietet die App einen Editor-Modus, über den unter anderem die Positionierung interaktiver Elemente in der Szene angepasst werden kann.
- **Web Player:** Als Alternative zur VR-Anwendung erlaubt der Web Player, die Szenarien auf einem Rechner zu durchlaufen oder zu bearbeiten.

PaneoVR bietet eine Vielzahl interaktiver Elemente, die es Nutzenden ermöglichen, Szenarien individuell und ansprechend zu gestalten. Die Basisfelder repräsentieren dabei die einfachsten Interaktionsmöglichkeiten und dienen der einfachen Navigation zwischen Szenen innerhalb eines Szenarios. Zu den Basisfeldern zählen die Elemente *Wegpunkt*, *Benutzen* und *Textpunkt*. Diese erlauben es, durch einen Klick zur jeweils verknüpften Szene zu wechseln. Die Elemente unterscheiden sich dabei nur in der Darstellung des Buttons und nicht in ihrer Funktionsweise. Auch *automatische Wechsel* können integriert werden. Hierbei erfolgt der Szenenwechsel automatisch nach Abschluss des laufenden Szenenvideos. Dieses Element wird in der Szene nicht dargestellt.

Für komplexere und nicht-lineare Szenenwechsel stehen Verzweigungselemente zur Verfügung. Das *Dialog-Element* bietet bis zu vier verschiedene Multiple-Choice-Optionen, die beispielsweise als Antworten auf zuvor im Video gesagte Inhalte verwendet werden können. Jede Option kann mit einer nachfolgenden Szene verknüpft und textlich dargestellt werden. Das *Quiz-Element* ähnelt dem Dialog, beinhaltet jedoch eine zentrale Frage, die in der Mitte der Multiple-Choice-Antworten dargestellt wird. Das *Zufällig-Element* wird in der Szene wie ein einfacher Wegpunkt dargestellt. Bei Auswahl dieses Elements wird jedoch zufällig eine der verknüpften Szenen ausgewählt, was eine dynamische Variation innerhalb eines Szenarios erlaubt. PaneoVR bietet außerdem die Möglichkeit, Bilder in die Szenen einzubinden und diese interaktiv zu nutzen. Ein *Bild-Element* bindet das gewählte Bild als Miniaturansicht in die Szene ein. Wird dieses ausgewählt, erscheint eine vergrößerte Ansicht des Bildes, die über einen Zurück-Button wieder geschlossen werden kann. Das Element *Untersuchen* verfolgt einen ähnlichen Ansatz, ersetzt jedoch die Miniaturansicht durch einen Button mit dem Icon einer Lupe, um den Eindruck eines detaillierten Betrachtens zu vermitteln. Der Bild-Wegpunkt verbindet die Anzeige eines Miniaturbilds mit der Funktionsweise eines Wegpunkts. Wird das Element ausgewählt, erfolgt ein Szenenwechsel.

Auditive Elemente spielen ebenfalls eine zentrale Rolle bei der Gestaltung immersiver Szenarien. Hintergrundsounds wie Musik, Umgebungsgeräusche oder nachträglich synchronisierte Inhalte können über das *Audio-Element* einer Szene hinzugefügt werden. Diese werden automatisch beim Start einer Szene abgespielt. Das Element *Audio-Anwählbar* erlaubt es den Nutzenden, Audiodateien gezielt durch Klicken eines Play-Buttons zu starten. Über einen *Audio-Wegpunkt* kann ein automatischer Szenenwechsel nach Beendigung der abgespielten Audiodatei integriert werden.

Darüber hinaus gibt es das Element *Hinweis*, welches nach einer definierten Zeitspanne einen eingeblendeten Text als Pop-up erscheinen lässt. Dies kann genutzt werden, um zusätzliche Informationen oder Hilfestellungen zu vermitteln. Schließlich können Szenarien durch *Szenario-Ende-Elemente* abgeschlossen werden. *Szenario Abgeschlossen* färbt die Szene grün und signalisiert den erfolgreichen Abschluss, während *Szenario Fehlgeschlagen* eine rote Markierung verwendet, um ein Scheitern anzuzeigen. Auch bei diesen Elementen öffnet sich ein Pop-Up, über das das Szenario beendet oder fortgesetzt werden kann [Meyer, 2025a].

Die vorliegende Arbeit basiert auf der PaneoVR-App für VR-Brillen. Der Schwerpunkt liegt auf der Integration binärer Interaktionsschnittstellen, um die Nutzung der erstellten Szenarien zu verbessern. Der Editor-Modus der App wird dabei bewusst vernachlässigt, da der Fokus auf dem Durchlaufen und der Interaktion mit den Szenarien liegt.

## 2.6 Zusammenfassung

In diesem Kapitel wurden die theoretischen und technologischen Grundlagen sowie der aktuelle Forschungsstand dargelegt, um die Relevanz dieser Arbeit zu unterstreichen und eine Grundlage für Entscheidungen zu schaffen, die im Rahmen der Konzeption sowie der Planung der Evaluation getroffen werden. Darüber hinaus baut die Einordnung der Evaluationsergebnisse auf den Erkenntnissen dieses Kapitels auf.

Es wurde dargelegt, dass der Forschungsbereich der Barrierefreiheit in VR ein relativ junges Forschungsfeld ist, das erst in den vergangenen Jahren vermehrt an Aufmerksamkeit gewonnen hat. Es existieren bisher nur wenige Richtlinien für die Entwicklung von barrierearmen VR-Anwendungen, an denen sich Entwickler:innen orientieren können. Aus der Übersicht und Zusammenfassung von Richtlinien von [Heilemann et al. \[2021\]](#) wird in Bezug auf die Kategorien Steuerung und Eingabe deutlich, dass eine Notwendigkeit vorliegt, einfache Steuerungsschemata bereitzustellen sowie assistive Technologien wie Schalter in VR-Anwendungen zu integrieren. Diese Notwendigkeit konnte im weiteren Verlauf des Kapitels durch die Analyse bestehender Barrieren für Menschen mit motorischen Beeinträchtigungen bestätigt werden. Es wurde gezeigt, dass aktuelle VR-Systeme primär für Menschen ohne Behinderung ausgelegt sind [\[Wong et al., 2017\]](#) und viele verbreitete Nutzungsanforderungen, wie die Bedienung von Controllern oder die Durchführung präziser Bewegungen zur Interaktion, Menschen mit motorischen Beeinträchtigungen von der Nutzung dieser Systeme ausschließen [\[Dombrowski et al., 2019\]](#). Obwohl bereits einige theoretische Lösungsansätze zur Überwindung dieser Barrieren existieren, gibt es bisher nur wenige Arbeiten, in denen praktische Implementierungen hinsichtlich alternativer Interaktionsmethoden durchgeführt wurden. Als Alternative zur klassischen Point-and-Select-Interaktion mittels VR-Controller haben bspw. [Valakou et al. \[2024\]](#) eine Interaktionsmethode mittels sequentiellen Scanning in ihr entwickeltes XR Accessibility Framework integriert. Eine ausführliche Testung und Evaluation dieses Frameworks steht jedoch noch aus. Auch die von [Creed et al. \[2024b\]](#) entwickelte Forschungsagenda zur Barrierefreiheit in VR hebt die Notwendigkeit hervor, alternative Eingabemethoden zu entwickeln und Systeme zu schaffen, die mit bestehenden assistiven Technologien kompatibel sind.

Diese Ergebnisse betonen die Relevanz der vorliegenden Arbeit und weisen auf einen deutlichen Bedarf in der Entwicklung und Untersuchung alternativer Interaktionsschnittstellen hin, um Menschen mit motorischen Beeinträchtigungen die Nutzung von VR-Anwendungen zu erleichtern bzw. zu ermöglichen. Binäre Interaktionsschnittstellen stellen in diesem Zusammenhang eine vielversprechende Lösung dar, da sie Interaktionen ohne präzise Bewegungen oder umfangreiche motorische Fähigkeiten ermöglichen und die Einbindung von Schaltern als assistive Technologie erlauben. Die große Vielfalt an verfügbaren Schaltern ermöglicht eine flexible Anpassung an die individuellen Bedürfnisse der Nutzenden.

Um im Rahmen der Konzeption fundierte Gestaltungsentscheidungen treffen zu können, wurden in diesem Kapitel gängige Scanning-Verfahren hinsichtlich ihrer Vor- und Nachteile präsentiert. Darüber hinaus wurde eine Einführung in den theoretischen Hintergrund von VR als Technologie gegeben, bei der Immersion, Interaktivität und Presence zentrale Konzepte dargestellt wurden. Des Weiteren wurde das Auftreten von Motion Sickness als Herausforderung bei der Nutzung von VR-Anwendungen genauer betrachtet. In ihrer Zusammenfassung von Richtlinien betonen [Heilemann et al. \[2021\]](#), dass VR-spezifische Trigger für Motion Sickness in der Entwicklung von Anwendungen möglichst zu vermeiden sind. Dies sollte demnach bei der Konzeption der Interaktionsschnittstellen berücksichtigt werden. Darüber hinaus wurde auf die Bedeutung von Usability und UX als Evaluierungsinstrumente für VR-Anwendungen eingegangen. Die etablierten Methoden, wie die SUS und der UEQ, ermöglichen eine einfache und schnelle Evaluierung dieser Konzepte. Der vorliegende theoretische Hintergrund verdeutlicht somit, dass im Rahmen der späteren Evaluation der konzipierten und implementierten Interaktionsschnittstellen insbesondere die Konzepte der Usability, UX sowie des Auftretens von Motion Sickness berücksichtigt und untersucht werden sollten. Die vorgestellten Informationen zur Interpretation der SUS von [Aaron Bangor and Miller \[2008\]](#), das ausführliche Auswertungstool für den UEQ und die Informationen zur Interpretation der Ergebnisse des SSQ nach [Stanney et al. \[1997\]](#) liefern dabei eine wertvolle Grundlage, um die im Rahmen der Evaluation erhobenen Daten einzuordnen.

Die gewonnenen Erkenntnisse aus diesem Kapitel bilden eine umfassende Grundlage für die im Folgenden vorgestellte Konzeption. Aufbauend auf den identifizierten Forschungslücken, bestehenden Barrieren und theoretischen Grundlagen werden im nächsten Kapitel die methodischen und technischen Überlegungen zur Gestaltung binärer Interaktionsschnittstellen für VR-Anwendungen dargelegt.



## 3 Konzeption

In diesem Kapitel wird die Konzeption der beiden binären Interaktionsschnittstellen dargelegt. Diese Konzeption bildet das Fundament dieser Arbeit, da sie die Grundlage für die spätere Implementierung und Evaluation legt. Der Prozess folgt einem systematischen Ablauf, der auf den Grundprinzipien der morphologischen Analyse von [Zwicky \[1967\]](#) basiert und für den spezifischen Kontext dieser Arbeit angepasst wurde. Diese Methode ermöglicht eine strukturierte und umfassende Betrachtung des Design Space. Der Ablauf besteht aus vier zentralen Schritten:

1. **Definition des Design Space:** In diesem Schritt werden die grundlegenden Gestaltungsoptionen und Anforderungen für die Interaktionen erarbeitet. Dazu werden die für die Zielsetzung dieser Arbeit relevanten Interaktionsaufgaben abgeleitet sowie Interaktionskomponenten und -Parameter identifiziert.
2. **Erarbeitung konkreter Ausprägungen:** Auf Grundlage des definierten Design Space werden spezifische Ausprägungen für die Interaktionsaufgaben und -komponenten entwickelt, die die Bandbreite möglicher Ansätze abbilden.
3. **Einordnung anhand der Parameter:** Die erarbeiteten Optionen werden systematisch anhand der zuvor (1.) identifizierten Parameter eingeordnet. Dieser Schritt dient dazu, die Designentscheidungen transparent darzulegen und zu begründen.
4. **Ableitung der finalen Konzepte:** Abschließend werden die finalen Konzepte für die beiden binären Interaktionsschnittstellen abgeleitet, die anschließend im Prototyp implementiert werden.

Durch diese strukturierte Vorgehensweise wird die erste Forschungsfrage (FF1) adressiert.



## 3.1 Definition des Design Space

Der Design Space für binäre Interaktionen in PaneoVR wird anhand von drei aufeinander aufbauenden Ebenen beschrieben. Die grundlegende Ebene wird durch die Interaktionsaufgaben definiert, die die für die Nutzung der Anwendung erforderlichen Interaktionsarten umfassen. Die zweite Ebene umfasst die Interaktionskomponenten, die spezifische Teilaspekte der Interaktion näher bestimmen. Die letzte Ebene bilden die Parameter, die in diesem Zusammenhang Designüberlegungen beschreiben, anhand derer die Interaktionstechniken und -ausprägungen bewertet werden. Die definierten Parameter beeinflussen dabei nicht nur den Zeitpunkt der Interaktion, sondern das gesamte VR-Erlebnis [L. Franz et al., 2021]. Die Kombination verschiedener Ausprägungen von Interaktionsaufgaben und -komponenten aus diesem Design Space kann anschließend als Grundlage für die Entwicklung neuartiger Interaktionstechniken verwendet werden.

### Interaktionsaufgaben

Die Nutzung der mit PaneoVR erstellten Trainings basiert auf zwei grundlegenden Interaktionsaufgaben. Die erste Interaktionsaufgabe besteht in der Selektion von Interaktionselementen innerhalb der Szene, d.h. von Elementen wie z. B. Wegpunkten oder Dialogfeldern, sowie von UI-Elementen z. B. innerhalb des Hauptmenüs. Die zweite Interaktionsaufgabe umfasst die Navigation innerhalb der Szene, einschließlich der Änderung der Blickrichtung. Da PaneoVR keinen dreidimensionalen Navigationsraum bereitstellt, sondern lediglich 360°-Videos präsentiert, ist eine klassische Navigation im 3D-Raum nicht notwendig. Stattdessen genügt eine Rotation der Szene, um die Blickrichtung zu ändern und damit innerhalb der virtuellen Welt zu navigieren. Da davon auszugehen ist, dass Nutzende mit motorischen Beeinträchtigungen Kopfbewegungen nicht frei und ohne Einschränkungen ausführen können, ist die Entwicklung einer alternativen Navigationsform unabdingbar.

### Interaktionskomponenten

In Bezug auf die Interaktionen in PaneoVR lassen sich die drei Komponenten Display, Transition und Initialisierung identifizieren. Deren konkrete Ausgestaltung nimmt einen direkten Einfluss auf die Erfahrung.

Die Komponente Display definiert Darstellung und Integration des Interaktionsverfahren in die virtuelle Umgebung. Im Rahmen der Festlegung dieser Interaktionskomponente ist insbesondere zu bestimmen, ob sich die Interaktion auf das Sichtfeld des Nutzers beschränkt. Die Relevanz dieser Komponente zeigt sich insbesondere in Kontexten, in denen Nutzende in der Lage sind, ihren Kopf zu bewegen und damit ihr Blickfeld

unabhängig von der binären Interaktionsschnittstelle zu verschieben.

Eine weitere zu berücksichtigende Komponente ist die Transition. Diese ist insbesondere bei der Betrachtung des Navigationsprozesses von Bedeutung. Die Transition beschreibt die Art und Weise, wie die Drehung der First-Person-Kamera erfolgt, wenn Nutzende eine navigationsbezogene Interaktion durchführen. Das heißt, Transition beschreibt den Übergang zwischen der aktuellen und der neuen Blickrichtung.

Die letzte relevante Komponente ist die Initialisierung. Dabei wird festgelegt, ob das gewählte Interaktionsverfahren automatisch und kontinuierlich aktiv ist oder zunächst durch eine initiale Interaktion initialisiert werden muss.

### Parameter

Die Entwicklung und Bewertung von Interaktionen basiert in der Regel auf der Berücksichtigung verschiedener Parameter, welche das Gesamterlebnis einer VR-Anwendung maßgeblich beeinflussen [L. Franz et al., 2021]. Die relevanten Parameter für die Interaktionen in PaneoVR werden auf Basis vorangegangener Forschungsarbeiten identifiziert. Da die Gewährleistung einer guten Usability das primäre Ziel der Gestaltung von Interaktionen in VR darstellt [Dörner et al., 2019], werden die Parameter Effizienz [DIN EN ISO 9241-11:2018-11], Effektivität [DIN EN ISO 9241-11:2018-11], Erlernbarkeit [DIN EN ISO 9241-110:2020-10] sowie Robustheit [DIN EN ISO 9241-110:2020-10] definiert. Darüber hinaus werden die Parameter Interaktionsgeschwindigkeit [Cook and Polgar, 2015a] und Komfort [Jerald, 2016] berücksichtigt.

*Effizienz* beschreibt das Verhältnis zwischen den eingesetzten Ressourcen und dem damit erreichten Ziel [DIN EN ISO 9241-11:2018-11]. Im Anwendungskontext ist insbesondere der benötigte Aufwand als relevante Ressource zu betrachten.

*Effektivität* bezeichnet die Genauigkeit und Vollständigkeit, mit der Benutzer ihre Ziele erreichen. Sie beschreibt, inwieweit die tatsächlichen Ergebnisse mit den angestrebten übereinstimmen. Je nach Kontext kann die Genauigkeit entweder anhand der Korrektheit eines Ergebnisses oder anhand des Erreichens eines akzeptablen Grads der Übereinstimmung mit dem Ziel bewertet werden [DIN EN ISO 9241-11:2018-11]. Im Anwendungskontext ist insbesondere die Präzision der Interaktion im Hinblick auf die Effektivität zu betrachten.

*Erlernbarkeit* beschreibt das Ausmaß, in dem eine Interaktion ohne vorherige Einarbeitung verstanden und genutzt werden kann. In diesem Kontext ist von Relevanz, dass insbesondere ein hoher Wiedererkennungswert der Interaktion eine einfache und intuitive Bedienbarkeit bedingt [Jerald, 2016].

*Robustheit* beschreibt die Fähigkeit eines Systems, auf Fehler von Nutzenden angemessen zu reagieren [DIN EN ISO 9241-110:2020-10]. Im Rahmen dieser Betrachtung wird untersucht, welche Maßnahmen ergriffen werden, um das Auftreten von Fehlern zu verhindern und inwiefern eine Korrektur von Fehlern möglich ist.

*Interaktionsgeschwindigkeit* beschreibt die Zeit, die benötigt wird, um ein Interaktionsziel zu erreichen. Es wird somit die Zeit betrachtet, die für die Ausführung einer Interaktionsaufgabe benötigt wird. Im Rahmen dieser Betrachtung wird insbesondere die Wartezeit während der Interaktion analysiert.

*Komfort* bezeichnet die Wahrnehmung der Nutzenden bezüglich der Angenehmheit der Interaktion [Jerald, 2016]. In diesem Zusammenhang ist insbesondere die Wahrscheinlichkeit des Auftretens von Motion Sickness von Bedeutung. Dabei werden Faktoren wie Ermüdung, erhöhte Konzentration und visuelle Komplexität in die Bewertung einbezogen.

## 3.2 Ausprägungen der Interaktionsaufgaben und -komponenten

Folgend werden mögliche Ausprägungen zur Implementierung der zuvor beschriebenen Interaktionsaufgaben und -komponenten erörtert.

### Interaktionsaufgaben

#### 1. Selektion

Die Realisierung einer binären Interaktion erfordert die Implementierung einer indirekten Selektionsmethode. In diesem Zusammenhang sind Scanning-Verfahren eine der am weitesten verbreiteten Methoden [Cook and Polgar, 2015a]. Zur Implementierung in PanoVR werden die gängigen Scanning-Verfahren Automatic Item Scanning, Step Item Scanning mit Dwell Selection, Inverse Item Scanning sowie Continuous Cartesian Scanning (vgl. [Unterabschnitt 2.3.2](#)) in Betracht gezogen.

#### 2. Navigation

Im Sinne einer konsistenten Gestaltung sollte die Art der Interaktion bei der Navigation einheitlich zu der bei der Selektion sein. Damit soll erreicht werden, dass sich das System für die Nutzenden möglichst erwartungskonform verhält. In dieser Konsequenz bedingt sich die Auswahl der Ausprägungen gegenseitig.

In der Literatur liegen bisher keine konkreten Umsetzungen für binäre Navigationschnittstellen in VR vor. Ein Ansatz, den bspw. [Wentzel \[2023\]](#) beschreibt, wäre die Übertragung von bereits existierenden Accessibility-Techniken für Smartphones auf den Einsatz in VR. Eine weit verbreitete Technik bei Smartphones besteht darin, dass sich ein Cursor von links nach rechts über den Bildschirm bewegt und der Nutzende durch das Drücken eines Schalters an die gewählte Position teleportiert wird [[Wentzel, 2023](#)]. Dieses Prinzip lässt sich auf das Continuous Cartesian Scanning übertragen. Anstelle eines Cursors wird hier der gesetzte Schnittpunkt der Scan-Linien als Zielpunkt der Navigation gewählt. Im Kontext dieser Arbeit, in dem die Navigation lediglich aus einer Rotation der First-Person-Kamera besteht, würde dies konkret bedeuten, dass zunächst der Schnittpunkt gesetzt wird und die Kamera daraufhin so rotiert wird, dass der Mittelpunkt des Blickfelds auf den definierten Schnittpunkt ausgerichtet wird.

Aus diesem Grundprinzip lassen sich zwei konkrete Optionen für die Navigation ableiten.

- *Direkte Navigation:* Hier werden Navigation und Selektion in einem Prozess kombiniert. Sobald der Schnittpunkt der Scan-Linien gesetzt ist, prüft das System, ob sich an diesem Punkt ein interaktives Element befindet. Ist dies der Fall, wird das entsprechende Element selektiert. Ist kein interaktives Element vorhanden, wird stattdessen die Kamera auf die Position des Schnittpunktes ausgerichtet und die Navigation durchgeführt.
- *Navigations- und Selektionsmodus:* Es stehen zwei getrennt nutzbare Modi zur Verfügung, ein Navigationsmodus und ein Selektionsmodus. Im Navigationsmodus führt das Setzen eines Schnittpunktes immer zu einer Navigation, unabhängig davon, ob sich hinter dem Schnittpunkt ein interaktives Element befindet oder nicht. Im Selektionsmodus hingegen wird der Schnittpunkt ausschließlich zur Selektion von Interaktionselementen verwendet. Befindet sich kein interaktives Element hinter dem Schnittpunkt, wird keine Aktion ausgeführt. Diese klare Trennung der Modi stellt sicher, dass im Navigationsmodus keine Selektion und im Selektionsmodus keine Navigation möglich ist.

Auch in Bezug auf das Item Scanning können bereits existierende Accessibility-Techniken als Inspiration für die Entwicklung einer Navigation in VR herangezogen werden. In Computerspielen ist es z. B. möglich, Steuerungen durch auf dem Bildschirm platzierte Buttons darzustellen und so der Auswahl durch Scanning zugänglich zu machen (vgl. z. B. [[Folmer et al., 2011](#), [Trewin et al., 2009](#)]). Dieses Prinzip könnte auch auf VR-Anwendungen übertragen werden. Hier könnten Buttons für die Rotationsrichtungen (links und rechts) in ein UI-Navigationsmenü integriert werden. Hinsichtlich der genauen

Funktionsweise der Buttons können wiederum zwei mögliche Ausprägungen unterschieden werden.

- *Direkte Rotation*: Die Selektion eines Buttons bewirkt eine Rotation um einen festen Winkel. Um eine größere Rotation zu erreichen, muss der entsprechende Button wiederholt selektiert werden.
- *Kontinuierliche Rotation*: Die Selektion eines Buttons löst eine kontinuierliche Rotation in die vom Nutzenden gewählte Richtung aus. Durch erneute Selektion des entsprechenden Buttons kann die Rotation jederzeit gestoppt werden. Ein vergleichbares Prinzip wurde z. B. von [Folmer et al. \[2011\]](#) zur Navigation von Spielavataren entwickelt und evaluiert.

## Interaktionskomponenten

### 1. Display

Die Interaktionskomponente *Display* bezieht sich auf die Darstellung und Integration des Scanning-Verfahrens in die virtuelle Umgebung. Dabei ist insbesondere entscheidend, ob sich das Scanning-Verfahren auf das Sichtfeld der Nutzenden beschränkt oder die gesamte 360°-Szene einbezieht. Da es sich somit um eine VR-spezifische Komponente handelt, können diesbezüglich keine Ansätze aus anderen Bereichen wie Computerspielen oder Smartphones als Grundlage herangezogen werden. Wie bereits in [Unterabschnitt 2.2.2](#) dargestellt, gibt es in der Literatur kaum Arbeiten, die sich speziell mit der Integration von binären Eingaben oder Scanning-Verfahren in VR auseinandersetzen. Eine Ausnahme bildet die Arbeit von [Valakou et al. \[2024\]](#), in der jedoch keine konkreten Aussagen zur Gestaltung der Interaktionskomponente Display getroffen werden. Aus diesen Gründen müssen eigenständige Konzepte entwickelt werden. Für die Gestaltung der Komponente *Display* wurden zwei grundlegende Optionen entwickelt.

- *Fixiert im Sichtfeld*: Das Scanning-Verfahren ist wie ein *Head-Up-Display* (HUD) im Sichtfeld des Nutzers fixiert. Das bedeutet, dass das Verfahren an die Blickrichtung des Nutzenden gekoppelt ist, sodass nur Elemente ausgewählt werden können, die sich innerhalb des Sichtfeldes befinden. Dies ermöglicht es, beim Cartesian Scanning die Position der Scan-Linien durch Kopfbewegungen direkt zu beeinflussen. Beim Item Scanning wirken sich Kopfbewegungen ebenfalls auf das Scanning aus, da hier das Scan-Set dynamisch angepasst wird. Elemente, die durch eine Kopfbewegung aus dem Sichtfeld verschwinden, werden aus dem Scan-Set entfernt, wodurch sich dessen Größe dynamisch verändert.

- *360°-Szene*: Der Scanning-Verfahren ist ein integraler Bestandteil der Szene. Das Scanning kann daher nicht durch Kopfbewegungen beeinflusst werden. Beim Cartesian Scanning bewegen sich die Scanlinien unabhängig von der Blickrichtung in einem Radius von 360° um den Nutzenden. Beim Item Scanning werden alle Interaktionselemente der Szene, auch außerhalb des Sichtfeldes, einbezogen und hervorgehoben.

## 2. Transition

Die Rotation der Kamera bei der Navigation kann auf verschiedene Arten realisiert werden. In der Literatur (vgl. z. B. [L. Franz et al. \[2021\]](#), [Farmani and Teather \[2020\]](#), [Onuki and Kumazawa \[2019\]](#)) werden insbesondere die beiden folgenden Varianten diskutiert.

- *Direkte Rotation*: Bei der direkten Rotation erfolgt eine unmittelbare Rotation der Kamera um einen vorgegebenen Winkel ohne intermediäre Schritte.
- *Kontinuierliche Rotation*: Die zweite Option umfasst eine kontinuierliche Bewegung der Kamera. In diesem Fall erfolgt eine fließende Drehung der Szene um den Nutzenden, wodurch eine natürliche Kopfbewegung simuliert wird.

## 3. Initialisierung

Die Komponente *Initialisierung* beschreibt, wie das Scanning Verfahren aktiviert wird und ob die Aktivierung durch die Nutzenden gesteuert werden kann. Dabei sind zwei Möglichkeiten denkbar.

- *Automatisch*: Das Scanning-Verfahren wird automatisch aktiviert und lässt sich auch nicht durch den Nutzenden deaktivieren. Das Verfahren läuft kontinuierlich durch die Szene sowie die gesamte Anwendung hindurch.
- *Manuell*: Das Scanning-Verfahren muss durch eine initiale Eingabe manuell aktiviert werden.

## Zusammenfassung

Die erarbeiteten Ausprägungen der Interaktionsaufgaben und -komponenten werden zusammenfassend in der [Abbildung 3.1](#) dargestellt.

Interaktionsaufgaben und -komponenten	Gestaltungsoptionen			
	Automatic Item Scanning	Step Item Scanning	Inverse Item Scanning	Continuous Cartesian Scanning
Selektion				
Navigation	Diskretes Navigationsmenü	Kontinuierliches Navigationsmenü	Direkte Navigation	Interaktionsmodi
Display	360°-Szene	Fixiert im Sichtfeld		
Transition	Direkte Rotation	Kontinuierliche Rotation		
Initialisierung	Manuell	Automatisch		

Abbildung 3.1: Darstellung der möglichen Ausprägungen der Interaktionsaufgaben und -komponenten

### 3.3 Einordnung und Bewertung der Interaktionsaufgaben und -komponenten anhand der Parameter

In diesem Abschnitt werden die zuvor beschriebenen Gestaltungsoptionen der Interaktionsaufgaben und -komponenten systematisch anhand der definierten Parameter (siehe [Abschnitt 3.1](#)) bewertet und eingeordnet. Dadurch soll eine fundierte Grundlage für die Auswahl der finalen Konzepte geschaffen werden. Diese Einschätzung basiert auf den Erkenntnissen der Literatur sowie sowie der in diesem Kapitel durchgeführten Analyse der vorgestellten Optionen. Es ist jedoch wichtig zu betonen, dass nicht alle Bewertungen durch empirische Daten belegt oder validiert sind, sondern auf einer fundierten, jedoch subjektiven Einschätzung beruhen.

#### Interaktionsaufgaben

##### 1. Selektion

Im Folgenden erfolgt die Einordnung der für die Selektion erarbeiteten Gestaltungsoptionen anhand der definierten Parameter. Dabei wird jeder Parameter einzeln betrachtet, um die spezifischen Stärken und Schwächen der Optionen detailliert zu analysieren. Die Ergebnisse werden in [Abbildung 3.2](#) visualisiert und bieten eine zusammenfassende Darstellung der Parameterbewertungen.

*Effizienz:* Ein effizientes binäres Selektionsverfahren sollte eine geringe Anzahl von Schalterbetätigungen und einen möglichst geringen physischen Aufwand für die Interaktion erfordern. Die verschiedenen Auswahlverfahren unterscheiden sich in dieser Hinsicht

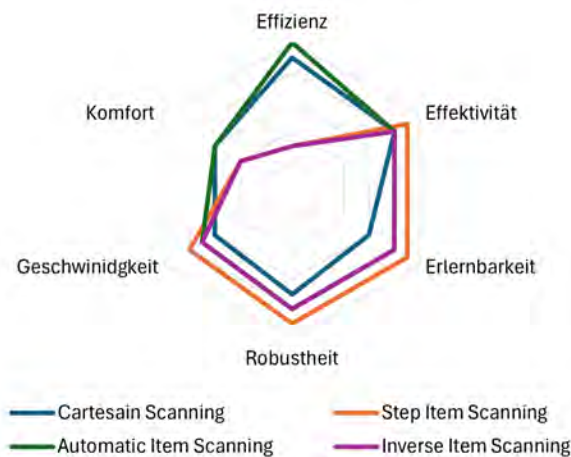


Abbildung 3.2: Einordnung der Gestaltungsoptionen der Interaktionsaufgabe Selektion anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort

deutlich. Cartesian Scanning erfordert zwei Schalteraktivierungen, um eine Interaktion durchzuführen. Die klare Struktur und die relativ geringe Anzahl an Aktivierungen ermöglichen eine gute Effizienz. Automatic Item Scanning zeichnet sich durch eine besonders geringe Anzahl von Aktivierungen aus, da die Selektion eines Elements in der Regel durch eine einzige Schalteraktivierung erfolgt. Dies macht die Methode besonders effizient. Wird allerdings eine Gruppierung von Elementen zur Beschleunigung der Interaktionsgeschwindigkeit eingesetzt, wirkt sich dies leicht negativ auf die Effizienz aus, da die Anzahl der erforderlichen Aktivierungen steigt. Step Item Scanning zeigt eine Effizienz, die stark von der Anzahl der Elemente in der Szene abhängt. Bei einer großen Anzahl von Elementen benötigt das Verfahren entsprechend viele Aktivierungen, um ein Ziel zu erreichen, was die Effizienz deutlich verringert. Eine Gruppierung der Elemente kann jedoch die Effizienz in Szenen mit vielen Elementen erhöhen, da weniger Schritte notwendig sind, um eine Auswahl zu treffen. Auch in Szenarien mit wenigen Elementen ist der Aufwand höher als bei Cartesian Scanning oder Automatic Item Scanning. Inverse Item Scanning ist ebenfalls stark von der Anzahl der Elemente in der Szene abhängig. Je mehr Elemente vorhanden sind, desto länger muss der Schalter gehalten werden, um das gewünschte Zielelement zu erreichen. Dies führt zu einem proportionalen Anstieg der körperlichen Anstrengung und verringert die Effizienz in Szenen mit vielen Elementen. Bei wenigen Elementen ist der Aufwand entsprechend geringer. Dennoch bleibt das kontinuierliche Halten des Schalters eine körperlich anstrengendere Interaktionsform.

*Effektivität:* Die Effektivität beschreibt, wie zuverlässig und präzise ein gewünschtes In-



teraktionselement ausgewählt werden kann. Das Cartesian Scanning erfordert präzises Timing, da der Nutzende den Schnittpunkt der Linien im richtigen Moment setzen muss. Ein Verfehlen des Zielelements ist möglich, insbesondere bei geringer Übung oder hohen Geschwindigkeiten des Scannings. Das Cartesian Scanning ermöglicht jedoch eine sehr präzise Selektion, insbesondere mit etwas Erfahrung. Das Automatic Item Scanning erzielt eine leicht höhere Effektivität. Da nur interaktive Elemente selektiert werden können, sind Fehlselektionen weniger wahrscheinlich. Dennoch spielt auch hier das Timing eine wesentliche Rolle, da das gewünschte Element verpasst werden kann, wenn die Aktivierung nicht rechtzeitig erfolgt. Das Step Item Scanning wird als das effektivste Verfahren eingeschätzt. Die Nutzenden haben die volle Kontrolle über die Auswahl und können in ihrem eigenen Tempo navigieren. Dies minimiert die Wahrscheinlichkeit von Fehlselektionen und gibt ihnen mehr Sicherheit, insbesondere in komplexen Szenarien mit vielen Elementen. Das Inverse Item Scanning zeigt eine Effektivität, die mit der des Automatic Item Scannings vergleichbar ist. Auch hier ist das Timing entscheidend, da der Schalter zum richtigen Zeitpunkt losgelassen werden muss, um das gewünschte Ziel zu erreichen. Die Möglichkeit von Fehlselektionen durch ungenaues Timing ist ein potenzieller Nachteil [Cook and Polgar, 2015a].

*Erlernbarkeit:* Die Erlernbarkeit beschreibt, wie schnell und intuitiv ein Scanning-Verfahren von den Nutzenden verstanden und angewendet werden kann. Das Step Item Scanning weist die höchste Erlernbarkeit auf. Es ermöglicht den Nutzenden die größte Kontrolle über das Scanning, da sie in ihrem eigenen Tempo navigieren können. Diese direkte Steuerbarkeit reduziert die kognitiven und sensorischen Anforderungen [Cook and Polgar, 2015a], was sich positiv auf das Verständnis der Funktionsweise auswirken könnte. Automatic Item Scanning und Inverse Item Scanning sind in ihrer Erlernbarkeit vergleichbar. Beide Verfahren erfordern die Fähigkeit, das richtige Timing zu finden, um das gewünschte Element auszuwählen. Cartesian Scanning wird als das herausforderndste Verfahren eingestuft. Hier müssen die Nutzenden sowohl der horizontalen als auch der vertikalen Bewegung der Scan-Linien folgen, was ein höheres Maß an Aufmerksamkeit erfordert. Darüber hinaus sind genauere Eingaben erforderlich. Insbesondere für Personen, die mit Scanning-Verfahren weniger vertraut sind, könnte dies anfangs zu Schwierigkeiten führen. Insgesamt dürften Personen mit Vorerfahrung im Umgang mit Schaltersteuerungen alle beschriebenen Verfahren als vertraut wahrnehmen. Diesen Personen sind Scanning-Verfahren vermutlich aus gängigen Betriebssystemen bekannt (z. B. [Apple Inc., 2024, Google, 2024]), was die Erlernbarkeit für diese Zielgruppe deutlich erleichtern kann.

*Robustheit:* Die Robustheit beschreibt, wie zuverlässig ein Scanning-Verfahren funktioniert und das Auftreten von Fehlern vermeidet. Die Festlegung einer geeigneten Scan-Rate ist ein wesentlicher Faktor für die Robustheit von Scanning-Verfahren. Eine zu hohe Scan-Rate kann dazu führen, dass den Nutzenden nicht ausreichend Zeit zur Ver-

fügung steht, um eine präzise Selektion durchzuführen [Cook and Polgar, 2015a]. Dieses Problem betrifft insbesondere das Cartesian Scanning sowie das Automatic und Inverse Item Scanning. Das Step Item Scanning hingegen wird als das robusteste Verfahren eingestuft, da die Nutzenden die Scan-Rate jederzeit selbst steuern können. Fehler, die aus einer zu schnellen oder zu langsamen Scan-Rate resultieren, können so vermieden werden. Diese Steuerbarkeit stellt sicher, dass das Verfahren flexibel auf unterschiedliche Anforderungen der Nutzenden ausgerichtet werden kann [Cook and Polgar, 2015a].

Das Automatic Item Scanning und das Inverse Item Scanning weisen eine etwas geringere Robustheit auf. Beide Verfahren sind stark von einer geeigneten Einstellung der Scan-Rate abhängig, da die Nutzenden das Timing der Selektion genau treffen müssen. Zudem können bei beiden Verfahren Fehler auftreten, wenn die Reihenfolge der Elemente nicht intuitiv gestaltet ist. Um dem entgegenzuwirken, sollte die Reihenfolge der Elemente möglichst nachvollziehbar gestaltet werden und eine klare Visualisierung gewählt werden, die bereits Hinweise auf das jeweils nächste Element in der Scan-Reihenfolge gibt. Das Cartesian Scanning wird mit einer mittleren Robustheit bewertet. Auch hier kann eine unpassende Scan-Rate die Präzision der Selektion beeinträchtigen. Außerdem besteht das Risiko, dass kleine Ungenauigkeiten beim Setzen der Scan-Linien zu unerwünschten Ergebnissen führen. Um die Robustheit zu erhöhen, könnte ein Toleranzbereich implementiert werden, der kleinere Ungenauigkeiten toleriert und damit die Intentionen der Nutzenden besser interpretiert. Die Interpretation der Intention von Eingaben stellt gemäß Dombrowski et al. [2019] generell eine Maßnahme dar, um Barrieren bei der Nutzung von VR abzubauen.

*Interaktionsgeschwindigkeit:* Im Vergleich zu direkten Interaktionsmethoden weisen Scanning-Verfahren generell eine langsamere Interaktionsgeschwindigkeit auf [Cook and Polgar, 2015a]. Die folgende Bewertung bezieht sich daher ausschließlich auf den relativen Vergleich der verschiedenen Scanning-Methoden untereinander. Ein wesentlicher Einflussfaktor auf die Interaktionsgeschwindigkeit beim Item Scanning ist die Anzahl der Interaktionselemente in der Szene, da ein sequentieller Durchlauf aller Interaktionselemente erfolgt Folmer et al. [2011]. Mit zunehmender Anzahl der Elemente steigt daher die Wahrscheinlichkeit, dass sich das Zielelement weiter hinten in der Scan-Reihenfolge befindet, was die Interaktionsgeschwindigkeit deutlich reduziert. Eine Maßnahme, um diesem Umstand entgegenzuwirken, besteht in der Gruppierung räumlich oder semantisch zusammenliegender Elemente. Die Geschwindigkeit des Item Scannings lässt sich dadurch deutlich erhöhen [Cook and Polgar, 2015a]. In Szenen mit wenigen Elementen oder wenn sich das Zielelement weiter vorne in der Sequenz befindet, kann hingegen eine höhere Interaktionsgeschwindigkeit erreicht werden. Das Step Item Scanning bietet gegenüber dem Automatic und Inverse Item Scanning den Vorteil, dass die Nutzenden den Scan-Vorgang aktiv steuern und direkt zum gewünschten Zielelement navigieren können, wodurch längere Wartezeiten vermieden werden. Dementsprechend wird die Interaktionsgeschwindigkeit des Automatic und Inverse Item Scanning als moderat bewertet,

während das Step Item Scanning eine höhere Interaktionsgeschwindigkeit aufweist. Das Cartesian Scanning hingegen weist eine konstantere Interaktionsgeschwindigkeit auf, da die Anzahl der Elemente in der Szene keinen direkten Einfluss auf die Dauer des Scan-Vorgangs hat. Es besteht jedoch ein Zusammenhang zwischen der Interaktionsgeschwindigkeit und der Position des Zielelements in der Szene. Befindet sich das Element weiter entfernt vom Startpunkt der Scan-Linien, kann es zu Verzögerungen kommen, bis die Linien das gewünschte Element erreichen. Die Stärke dieses Effekts ist jedoch noch unklar und bedarf weiterer Untersuchungen. Insgesamt wird die Interaktionsgeschwindigkeit des Cartesian Scanning als moderat eingestuft.

*Komfort:* Beim Komfort von Scanning-Verfahren ist insbesondere die Wahrscheinlichkeit des Auftretens motorischer Ermüdung sowie der kognitiven und visuellen Anstrengung zu berücksichtigen. Cartesian Scanning erfordert nur wenige Aktivierungen des Schalters, was das Risiko motorischer Ermüdung vergleichsweise gering hält. Allerdings erfordert das Verfahren ein präzises Timing sowie eine erhöhte visuelle Aufmerksamkeit, da die Bewegung der Scan-Linien kontinuierlich verfolgt werden muss. Dies führt zu einer erhöhten kognitiven Beanspruchung. Das Automatic Item Scanning zeichnet sich ebenfalls durch eine geringe Anzahl erforderlicher Schalteraktivierungen aus, wodurch die motorische Belastung begrenzt bleibt. Allerdings erfordert das Verfahren eine hohe visuelle Aufmerksamkeit und ein exaktes Timing, was die Anforderungen an die Konzentration erhöht. Das Step Item Scanning erfordert eine deutlich höhere Anzahl an Aktivierungen, was zu einer höheren motorischen Belastung und einem erhöhten Risiko der motorischen Ermüdung führt. Gleichzeitig ist die Anforderung an die visuelle Aufmerksamkeit geringer, da weder ein genaues Timing noch ein kontinuierliches Verfolgen des Scans erforderlich ist und die Nutzenden mehr Kontrolle über den Scan-Vorgang haben. Dadurch wird die kognitive Belastung reduziert. Dennoch wird der Komfort des Step Item Scanning aufgrund der hohen motorischen Anforderungen als gering bewertet. Inverse Item Scanning liegt hinsichtlich der motorischen Belastung zwischen Automatic und Step Item Scanning. Das Halten des Schalters erfordert eine moderate körperliche Anstrengung, die höher als beim Automatic Item Scanning, aber geringer als beim Step Item Scanning ist. Außerdem ist eine erhöhte visuelle Aufmerksamkeit erforderlich, um das Timing präzise zu steuern. Der Komfort des Inverse Item Scanning wird daher ebenfalls als gering bewertet [Cook and Polgar, 2015a].

## 2. Navigation

Im Folgenden erfolgt die Einordnung der für die Navigation erarbeiteten Gestaltungsoptionen anhand der definierten Parameter. Die Ergebnisse aus dieser Einordnung werden in **Abbildung 3.3** zusammenfassend visualisiert.

*Effizienz:* Die Effizienz der Gestaltungsoptionen der Navigation wird durch den erforder-



Abbildung 3.3: Einordnung der Gestaltungsoptionen der Interaktionsaufgabe Navigation anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort

derlichen Interaktionsaufwand bestimmt. Der Schwerpunkt liegt hier auf der Anzahl der Eingaben, die zur Durchführung der Navigation erforderlich sind. Die beiden auf dem Cartesian Scanning basierenden Optionen weisen unterschiedliche Effizienzniveaus auf. Die direkte Navigation erfordert lediglich die Festlegung des Schnittpunktes, was eine relativ hohe Effizienz gewährleistet. Allerdings steigt der Interaktionsaufwand bei größeren Rotationswinkeln, da in diesen Fällen mehrere Navigationsschritte erforderlich sein können, um den gewünschten Endpunkt zu erreichen. Die Navigation mit separaten Interaktionsmodi erfordert zusätzlich eine Interaktion, um den Moduswechsel zwischen Navigation und Selektion durchzuführen, was den Aufwand geringfügig erhöht. Daher wird die Effizienz der direkten Navigation etwas höher eingeordnet.

Bei den Optionen des Item-Scannings hat der gewünschte Rotationswinkel einen erheblichen Einfluss auf die Effizienz. Bei der diskreten Navigation führt ein größerer Winkel zu einer höheren Anzahl von Eingaben, da die Rotation in festen Schritten erfolgt. Dies macht die Methode bei großen Rotationswinkeln weniger effizient. Dagegen erfordert die kontinuierliche Navigation bei kleinen Rotationen einen geringfügig höheren Aufwand, da zwei Eingaben erforderlich sind (eine zum Aktivieren und eine zum Beenden der Navigation). Dieser Mehraufwand wird jedoch bei großen Rotationswinkeln durch die Möglichkeit der kontinuierlichen Navigation kompensiert. Die Effizienz der diskreten Navigation wird als etwas geringer eingestuft.

*Effektivität:* Die Effektivität der Gestaltungsoptionen der Navigation wird durch die Genauigkeit bestimmt, mit der ein gewünschter Rotationswinkel erreicht werden kann. Dabei wird insbesondere berücksichtigt, inwieweit die Nutzenden die Navigation kon-

trollieren und den Rotationswinkel exakt bestimmen können.

Ein deutlicher Unterschied in der Effektivität zeigt sich bei den Optionen, die auf dem Cartesian Scanning basieren. Die direkte Navigation wird durch die Position und Anzahl der Interaktionselemente in der Szene eingeschränkt. Je mehr Elemente vorhanden sind, desto mehr leidet die Effektivität, da nur Punkte im Raum ausgewählt werden können, an denen sich kein Interaktionselement befindet. Dies schränkt die Kontrolle über den Rotationswinkel erheblich ein. Bei Szenen mit wenigen Interaktionselementen ist dieser Effekt jedoch praktisch vernachlässigbar. Die Navigation mit Interaktionsmodi bietet hingegen eine deutlich höhere Flexibilität, da beliebige Rotationswinkel erreicht werden können.

Auch die Optionen des Item-Scannings unterscheiden sich in ihrer Effektivität. Das Navigationsmenü mit diskreter Rotation erlaubt nur eine begrenzte Anzahl von vorgegebenen Rotationswinkeln. Dies führt dazu, dass die Zielposition nicht immer exakt erreicht werden kann, was wiederum die Effektivität einschränkt. Im Gegensatz dazu bietet die kontinuierliche Rotation durch ihre Flexibilität die Möglichkeit, jeden beliebigen Winkel zu erreichen. Diese Methode gewährleistet somit eine hohe Präzision [Folmer et al., 2011].

*Erlernbarkeit:* Bei der direkten Navigation beim Cartesian Scanning profitieren die Nutzenden von der Vertrautheit mit dem Prinzip aus der Selektion, sodass keine neuen Mechanismen erlernt werden müssen. Allerdings kann die Durchführung der Navigation aufgrund der notwendigen Koordination zwischen Selektion und Navigation anfangs eine gewisse Herausforderung darstellen. Insgesamt wird die Erlernbarkeit daher als hoch eingeschätzt. Die Interaktionsmodi erfordern zusätzlich das Verständnis, wann welcher Modus aktiv ist und wie zwischen den Modi gewechselt wird. Dies könnte eine leichte kognitive Hürde darstellen, insbesondere wenn der Moduswechsel eine andere Art der Eingabe als die Selektion erfordert (z. B. Halten des Schalters oder zwei schnelle Aktivierungen hintereinander). Während das zugrundeliegende Konzept einfach ist, kann die anfängliche Anwendung der Modi für neue Nutzende anspruchsvoller sein. Die Erlernbarkeit wird daher als moderat bewertet. Die diskrete Navigation im Navigationsmenü basiert auf konsistenten Schritten. Die Nutzenden wählen Buttons aus, die festgelegte Rotationswinkel repräsentieren. Dies macht die Bedienung besonders einfach und einsteigerfreundlich, da keine zusätzlichen Mechanismen erlernt werden müssen. Daher wird die Erlernbarkeit als sehr hoch bewertet.

Die kontinuierliche Navigation ermöglicht es, die Rotation durch einmalige Selektion eines Buttons zu starten und durch erneute Selektion desselben Buttons zu stoppen. Dieser Mechanismus ist prinzipiell leicht verständlich, kann aber anfangs eine Herausforderung darstellen, da den Nutzenden möglicherweise nicht klar ist, wie die Rotation wieder gestoppt werden kann. Aufgrund dieser geringen Einstiegshürde wird die Erlernbarkeit als hoch eingeschätzt.

*Robustheit:* Bei der direkten Navigation (basierend auf dem Cartesian Scanning) können Fehler durch fehlende Abgrenzung zwischen Selektion und Navigation entstehen. Insbesondere in Szenen mit vielen Interaktionselementen oder bei ungenauem Timing kann es zu Verwechslungen kommen, etwa wenn Nutzende ein Element selektieren möchten, die Eingabe jedoch knapp verfehlen. Dies kann dazu führen, dass stattdessen eine Navigation ausgelöst wird, was die Robustheit maßgeblich beeinträchtigt. Die Einführung eines separaten Modus für Navigation und Selektion hingegen minimiert die Wahrscheinlichkeit solcher Fehler und erhöht die Robustheit deutlich. Dennoch können Fehler auftreten, wenn versehentlich der falsche Modus aktiv ist. Bspw. könnte eine Selektion gewünscht sein, während der Navigationsmodus noch aktiv ist. Solche Situationen erfordern eine bewusste Kontrolle durch die Nutzenden, um zwischen den Modi zu wechseln. Insgesamt wird die Robustheit der Interaktionsmodi jedoch als deutlich höher eingeschätzt als die der direkten Navigation.

Ein Navigationsmenü bietet in beiden Varianten eine robuste Kontrolle über die Navigation. Jedoch können Timing-Probleme dazu führen, dass versehentlich in die falsche Richtung navigiert wird. Solche Fehler lassen sich in beiden Varianten korrigieren, jedoch unterscheidet sich der Aufwand je nach Mechanismus. Bei der diskreten Navigation führt eine Fehleingabe lediglich zu einem einzelnen Schritt in die falsche Richtung. Der Fehler kann leicht korrigiert werden, indem im Anschluss die entgegengesetzte Richtung gewählt wird. Die kontinuierliche Navigation erfordert hingegen mehr Aufwand für die Fehlerkorrektur. Nutzende müssen zunächst die aktive Eingabe abbrechen, anschließend die Eingabe für die eigentlich gewünschte Richtung aktivieren und diese wiederum abbrechen, sobald die Korrektur abgeschlossen ist. Dies führt zu einem erhöhten Zeit- und Arbeitsaufwand, was die Robustheit des Verfahrens im Vergleich zur diskreten Navigation verringert [Folmer et al., 2011].

*Interaktionsgeschwindigkeit:* Die direkte Navigation wird als die schnellste Option angesehen. Da keine zusätzlichen Schritte für den Wechsel zwischen Selektion und Navigation notwendig sind, kann die Navigation unmittelbar durchgeführt werden. Darüber hinaus ermöglicht die direkte Navigation, einen großen Rotationswinkel mit nur einer Navigation zu erreichen, indem der Schnittpunkt bswp. weit an den Rand des aktuellen Sichtfeldes gesetzt wird. Dies reduziert die Anzahl der notwendigen Navigationen und erhöht damit die Interaktionsgeschwindigkeit. Einschränkungen können jedoch auftreten, wenn die Anzahl oder Position der Interaktionselemente in der Szene die Wahl eines großen Winkels verhindert. Die Option mit getrennten Interaktionsmodi erfordert einen Zwischenschritt, da zwischen Selektion und Navigation gewechselt werden muss, was zusätzliche Zeit in Anspruch nimmt. Nach dem Moduswechsel bietet die Navigation jedoch die gleichen Vorteile wie die direkte Navigation. Darüber hinaus bestehen hier keine Einschränkungen hinsichtlich der Anzahl oder Position der Elemente.

Die Navigation über ein Navigationsmenü erfordert zunächst das Öffnen des Menüs, was einen Zwischenschritt darstellt, der sich negativ auf die Geschwindigkeit auswirkt. Bei

der diskreten Navigation wird die Geschwindigkeit zusätzlich durch die Anzahl der erforderlichen Schritte beeinflusst. Um eine größere Rotation zu erreichen, müssen mehrere Interaktionen nacheinander ausgeführt werden. Dies führt zu einer insgesamt geringeren Geschwindigkeit, insbesondere bei größeren Rotationswinkeln. Im Vergleich zur diskreten Variante ist die kontinuierliche Navigation schneller [Folmer et al., 2011], da eine kontinuierliche Rotation gestartet werden kann, die nur durch eine weitere Interaktion gestoppt werden muss. Dies reduziert die Anzahl der notwendigen Interaktionen und beschleunigt die Navigation insbesondere bei größeren Rotationswinkeln. Jedoch wirkt sich auch hier der Zwischenschritt des Öffnens des Menüs negativ auf die Geschwindigkeit aus.

*Komfort:* Der Komfort der Navigation wird maßgeblich durch die erforderliche Konzentration, den allgemeinen kognitiven Aufwand sowie den visuellen Anspruch beeinflusst. Die Optionen, die auf dem Cartesian Scanning basieren, erfordern insgesamt eine präzise Auswahl, um den gewünschten Rotationspunkt zu erreichen. Diese Präzision geht mit einer leicht erhöhten Konzentration einher. Bei der direkten Navigation ist zusätzlich zu berücksichtigen, dass eine unbeabsichtigte Auswahl eines Interaktionselements möglich ist, was den kognitiven Aufwand weiter erhöht. Die Interaktionsmodi-Option reduziert das Risiko von Fehleingaben durch die Trennung von Selektion und Navigation. Dies führt zu einer leicht reduzierten kognitiven Belastung im Vergleich zur direkten Navigation. Die visuelle Komplexität der Szene wird durch das Hinzufügen zusätzlicher Interaktionselemente für das Navigationsmenü leicht erhöht. Die diskrete Navigation mit vorgegebenen Rotationswinkeln erfordert insgesamt die geringste Konzentration, da die Interaktion sehr konsistent ist. Dadurch wird die kognitive Belastung minimiert. Die kontinuierliche Navigation benötigt ein gewisses Timing, um die Rotation im gewünschten Moment zu stoppen. Dies erfordert eine erhöhte Konzentration im Vergleich zur diskreten Option. Darüber hinaus besteht bei einer durchgängigen Rotation der First-Person-Kamera ein erhöhtes Risiko für Motion Sickness, was den Komfort weiter beeinträchtigen kann [Farmani and Teather, 2020, Onuki and Kumazawa, 2019].

## Interaktionskomponenten:

### 1. Display

Folgend werden die erarbeiteten Gestaltungsoptionen für die Komponente Display anhand der definierten Parameter eingeordnet. Die Ergebnisse dieser Einordnung sind in [Abbildung 3.4](#) zusammenfassend visualisiert.

*Effizienz:* Die Effizienz der verschiedenen Optionen zur Darstellung des Scanning-Verfahrens in der virtuellen Umgebung wird hinsichtlich des erforderlichen Aufwands, insbesondere der Anzahl der erforderlichen Schalteraktivierungen, analysiert. Es zeigt sich, dass der



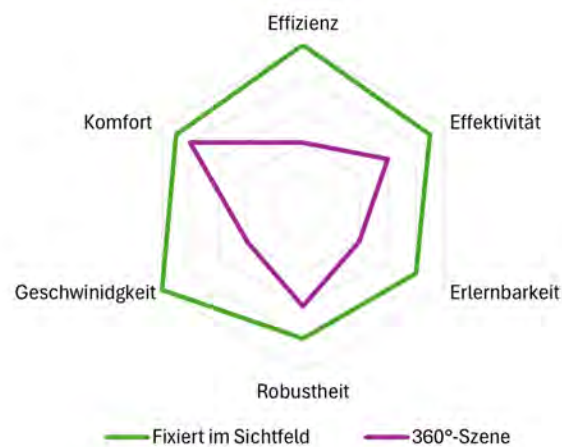


Abbildung 3.4: Einordnung der Gestaltungsoptionen der Interaktionskomponente Display anhand der Parameter Effizienz, Effektivität, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort

Einfluss der Display-Optionen von der Art des Scanning-Verfahrens abhängt. Sowohl beim Automatic Item Scanning als auch beim Cartesian Scanning ist der Aufwand unabhängig davon, ob das Scanning-Verfahren auf das Sichtfeld beschränkt oder auf die gesamte Szene ausgeweitet wird. Bei beiden Verfahren erfolgt das Scanning automatisiert und die präsentierte Auswahl (Item: hervorgehobenes Element, Cartesian: Position der Linien) wird durch Aktivierung des Schalters ausgewählt. Im Gegensatz dazu zeigt sich bei Step und Inverse Item Scanning ein deutlicher Einfluss der Display-Optionen. Wird das Scanning auf die gesamte Szene ausgeweitet, werden alle Interaktionselemente unabhängig von ihrer Position in das Scan-Set einbezogen. Dadurch erhöht sich die Anzahl der zu durchlaufenden Elemente. Beim Step Item Scanning erhöht sich dadurch die Anzahl der erforderlichen Schalterbetätigungen und beim Inverse Item Scanning wird ein längeres Halten des Schalters erforderlich. Dies führt zu einem deutlich erhöhten Aufwand und damit zu einer geringen Effizienz. Eine Beschränkung des Scan-Sets auf Elemente im Sichtfeld hingegen reduziert die Anzahl der zu durchlaufenden Interaktionselemente und damit den erforderlichen Aufwand bei beiden Verfahren deutlich. Dies ist insbesondere bei Szenen mit vielen Interaktionselementen entscheidend.



*Effektivität:* Die Fixierung des Scanning-Verfahrens an das Sichtfeld des Nutzenden ermöglicht eine feinere Kontrolle des Scannings, da der Scan-Bereich immer visuell verfolgt werden kann, was sich wiederum positiv auf die Effektivität auswirkt. Ist das Scanning-Verfahren auf die gesamte Szene ausgeweitet, tritt der gegenteilige Effekt ein. In diesem Fall wird eine präzise Steuerung erschwert, da der Scan-Bereich nicht vollständig visuell erfasst werden kann, was insbesondere bei komplexen Szenen zu Verwirrung und damit zu unpräzisen Interaktionen führen kann. Die Effektivität wird daher für die Fixierung als hoch bewertet und für die Ausweitung auf den gesamten 360°-Bereich der Szene als moderat.

*Erlernbarkeit:* Bei der Fixierung im Sichtfeld bleibt der gesamte Scan-Vorgang im sichtbaren Bereich, was das visuelle Feedback erleichtert. Die Nutzenden können jederzeit erkennen, ob der Scan-Vorgang aktiv ist und diesen verfolgen. Dies erleichtert nicht nur den Einstieg in die Nutzung, sondern reduziert auch das Risiko von Missverständnissen oder Fehleingaben. Im Falle einer Fehleingabe kann diese direkt visuell nachvollzogen und ggf. korrigiert werden. Aufgrund dieser Transparenz wird die Erlernbarkeit dieser Option als hoch bewertet.

Wird das Scanning-Verfahren auf die gesamte Szene ausgeweitet, erfordert dies ein höheres Maß an Verständnis für den Systemzustand, da der Scan-Vorgang auch außerhalb des Sichtfeldes stattfinden kann. Fehlerhafte Schalterbetätigungen können in diesem Fall Systemreaktionen auslösen, die für die Nutzenden nicht unmittelbar nachvollziehbar sind. Dies erschwert das Erlernen und erfordert zusätzliche Rückmeldesysteme, um den Status des Scans verständlich zu machen. Die Erlernbarkeit wird daher als geringer eingestuft.

*Robustheit:* Die Fixierung des Scanning-Verfahrens auf das Sichtfeld ermöglicht es den Nutzenden, den Scan-Vorgang durch gezielte Kopfbewegungen zu beeinflussen. Diese Möglichkeit kann insbesondere beim Cartesian Scanning die Robustheit erhöhen, da kleinere Ungenauigkeiten durch bewusste Kopfbewegungen korrigiert werden können. Dies setzt allerdings voraus, dass die Nutzenden dazu in der Lage sind, Kopfbewegungen präzise und kontrolliert vorzunehmen. Für Personen mit eingeschränkter motorischer Kontrolle könnte diese Abhängigkeit von der Kopfbewegung unter Umständen problematisch sein und die Robustheit der Interaktion verringern. Unbeabsichtigte Verschiebungen des Scan-Vorgangs durch unkontrollierte Kopfbewegungen oder ungenaue Steuerung können dazu führen, dass Interaktionselemente verfehlt werden. Dies erhöht das Risiko von Navigations- oder Selektionsfehlern. Trotz dieser möglichen Nachteile bleibt die visuelle Rückmeldung innerhalb des Sichtfeldes ein stabilisierender Faktor, der die Nachvollziehbarkeit von Fehlern und deren Korrektur erleichtert.

Die Integration des Scanning-Verfahrens in die gesamte Szene eliminiert die Abhängigkeit von Kopfbewegungen und verhindert somit unbeabsichtigte Verschiebungen des Scannings durch unkontrollierte Bewegungen und erhöht damit die Robustheit des Systems. Diese Option birgt jedoch ein erhöhtes Risiko unbeabsichtigter Selektionen, da

der Scan außerhalb des Sichtfeldes stattfinden kann und für die Nutzenden visuell nicht jederzeit nachvollziehbar ist. Bei den Scanning-Verfahren Automatic und Inverse Item Scanning sowie beim Cartesian Scanning können außerdem schneller Timing-Fehler entstehen, wenn der Scan-Vorgang von außerhalb des Sichtfeldes in den Sichtbereich übergeht. In diesem Fall könnte die Reaktionszeit des Nutzenden nicht ausreichen, um das Zielelement rechtzeitig zu selektieren. Aus diesen Gründen wird die Robustheit bei einer Integration in die gesamte Szene als etwas geringer angenommen als bei einer Fixierung im Sichtfeld.

*Interaktionsgeschwindigkeit:* Die Interaktionsgeschwindigkeit der beiden Display-Optionen unterscheidet sich deutlich. Die Option mit fixiertem Scanning im Sichtfeld bietet deutliche Vorteile in der Interaktionsgeschwindigkeit. Da hier beim Item Scanning nur die Elemente im Sichtfeld gescannt werden, ist die Größe des Scan-Sets begrenzt, was zu einem schnelleren Durchlauf der Sequenz und damit zu einer schnelleren Interaktionsgeschwindigkeit führt. Beim Cartesian Scanning ist die Bewegung der Linien auf einen kleinen Bereich beschränkt, sodass Wartezeiten reduziert werden können. Darüber hinaus können beim Cartesian Scanning Kopfbewegungen (sofern möglich) dazu genutzt werden, die Scan-Linien näher an das gewünschte Zielelement zu bringen, was die Interaktionsgeschwindigkeit beschleunigt. Insgesamt führt die Kombination aus kleiner Größe des Scan-Sets und aktiver Beeinflussbarkeit durch Kopfbewegungen zu einer hohen Interaktionsgeschwindigkeit.

Die auf die gesamte Szene erweiterte Option zeigt eine geringere Interaktionsgeschwindigkeit, insbesondere beim Item Scanning. Da hier alle Elemente der Szene unabhängig von ihrer Position im Sichtfeld berücksichtigt werden, nimmt die Größe des Scan-Sets deutlich zu. Dies führt zu einer Verzögerung, da der Scan-Vorgang mehr Zeit benötigt, um alle Elemente sequentiell zu durchlaufen. Beim Cartesian Scanning ist die Interaktionsgeschwindigkeit ebenfalls niedriger, da die Bewegung der Scan-Linien unabhängig von der Blickrichtung stattfindet und somit mehr Zeit in Anspruch nimmt. Außerdem ist es nicht möglich, den Scan-Vorgang durch Kopfbewegungen gezielt zu beschleunigen.

*Komfort:* Der Komfort der beiden Display-Optionen unterscheidet sich deutlich hinsichtlich der erforderlichen Konzentration, der visuellen Komplexität und der motorischen Ermüdung. Die Fixierung des Scan-Vorgangs im Sichtfeld erfordert ein höheres Maß an Konzentration, da selbst minimale Kopfbewegungen einen direkten Einfluss auf den Scan-Vorgang haben. Dies kann zu ungewollten Verschiebungen führen. Darüber hinaus kann die Kombination von Kopfbewegung und Scanning zu visuellen Irritationen führen, insbesondere wenn die Bewegung der Scan-Linien aufgrund der Kopfbewegung schneller oder unregelmäßiger erscheint. Ein Vorteil dieser Option liegt jedoch in der potenziell geringeren motorischen Ermüdung, da durch die Beschränkung auf das Sichtfeld das Scan-Set verkleinert werden kann, wodurch die Anzahl der erforderlichen Interaktionen reduziert wird, insbesondere bei Verfahren wie Step und Inverse Item Scanning.

Die Option, das Scanning auf die gesamte 360°-Szene zu erstrecken, bietet hinsichtlich der Konzentration einen deutlichen Vorteil, da Kopfbewegungen keinen Einfluss auf den Scan-Vorgang haben. Dies minimiert die Anforderungen an die Aufmerksamkeit und sorgt für einen gleichmäßigen Ablauf des Scannings, was den Komfort erhöht. Auch die visuelle Komplexität ist vergleichsweise gering, da die Bewegung des Scannings unabhängig von äußeren Einflüssen erfolgt und somit vorhersehbar bleibt. Allerdings zeigt diese Option Schwächen bei der motorischen Ermüdung, insbesondere im Zusammenhang mit Step und Inverse Item Scanning. Die größere Größe des Scan-Sets erfordert eine höhere Anzahl von Interaktionen oder ein längeres Halten des Schalters, was schnell zu Ermüdung führen kann.

## 2. Transition

Im Folgenden erfolgt die Einordnung der erarbeiteten Gestaltungsoptionen für die Komponente Transition anhand der definierten Parameter. Die Ergebnisse dieser Einordnung sind in [Abbildung 3.5](#) zusammenfassend dargestellt.



Abbildung 3.5: Einordnung der Gestaltungsoptionen der Interaktionskomponente Transition anhand der relevanten Parameter Erlernbarkeit, (Interaktions-) Geschwindigkeit und Komfort

*Effizienz:* Die Effizienz der Interaktion, gemessen am notwendigen Aufwand, wird durch die Gestaltung der Transition kaum beeinflusst. Beide Optionen führen die Rotation mit einem vergleichbaren Aufwand durch, da die grundlegende Interaktion zur Auslösung der Rotation identisch bleibt.

*Effektivität:* Die Effektivität der Interaktion, gemessen an der Präzision des Rotationswinkel, wird durch die Gestaltung der Transition nicht beeinflusst. Der exakte Rotationswinkel wird durch die Navigation bestimmt, während die Transition lediglich den visuellen Übergang zwischen Ausgangs- und Zielposition gestaltet.

*Erlernbarkeit:* Die Erlernbarkeit der Optionen ist als hoch einzuschätzen, da beide Ansätze bereits in etablierten VR-Anwendungen implementiert werden und somit den Nutzenden einen Wiedererkennungswert bieten.

*Robustheit:* Die Optionen zur Gestaltung der Transition haben keinen bedeutenden Einfluss auf das Auftreten oder die Vermeidung von Fehlern und somit keinen direkten Einfluss auf die Robustheit. Es kann davon ausgegangen werden, dass beide Optionen stabile und zuverlässige Transitionen darstellen, da sie bereits häufig in VR-Anwendungen implementiert sind. Einflüsse auf die Robustheit können sich ggf. ergeben, wenn durch die Art der Transition Motion Sickness auftritt und die Nutzenden in Folge dessen dazu neigen, eher Fehler zu machen.

*Interaktionsgeschwindigkeit:* Die Optionen für die Transition unterscheiden sich in der Interaktionsgeschwindigkeit. Direkte Rotation bietet eine schnellere Interaktionsgeschwindigkeit, da die Kamera ohne Übergangszeit unmittelbar in die gewünschte Position gebracht wird. Die kontinuierliche Rotation hingegen ist aufgrund der Übergangszeit langsamer. Hier findet eine Bewegung der Kamera über die Zeit statt. Daher benötigt diese Transition länger, um die Zielposition zu erreichen [Onuki and Kumazawa, 2019]. Der Unterschied zwischen den Interaktionszeiten wird durch die Geschwindigkeit der Kamerabewegung und die Größe des gewählten Rotationswinkels beeinflusst. Eine Rotation um einen großen Rotationswinkel erfordert mehr Zeit. Bei einer Direkten Rotation ist die Geschwindigkeit hingegen konstant und nicht von anderen Faktoren abhängig.

*Komfort:* Die Optionen für die Transition unterscheiden sich deutlich in Bezug auf den Komfort, insbesondere im Hinblick auf die Auswirkungen auf die Wahrscheinlichkeit für das Auftreten von Motion Sickness und die räumliche Wahrnehmung.

Die kontinuierliche Rotation ahmt die natürliche Bewegung des Kopfes nach und wird daher als realistischer empfunden [Onuki and Kumazawa, 2019]. Dies kann das räumliche Bewusstsein der Nutzenden verbessern [L. Franz et al., 2021]. Allerdings birgt die Kontinuierliche Transition ein sehr hohes Risiko für das Auftreten von Motion Sickness [Farmani and Teather, 2020, Onuki and Kumazawa, 2019], was den Komfort deutlich mindert. Im Gegensatz dazu ist das Risiko von Motion Sickness bei der direkten Rotation vergleichsweise gering. Allerdings kann die plötzliche Änderung des Sichtfeldes dazu führen, dass die Nutzenden kurzzeitig die räumliche Orientierung verlieren und sich in ihrer Umgebung neu orientieren müssen [L. Franz et al., 2021].

### 3. Initialisierung

Im Folgenden werden die erarbeiteten Gestaltungsoptionen für die Komponente Initialisierung anhand der definierten Parameter eingeordnet. Die Ergebnisse dieser Einordnung sind in **Abbildung 3.6** zusammenfassend dargestellt.

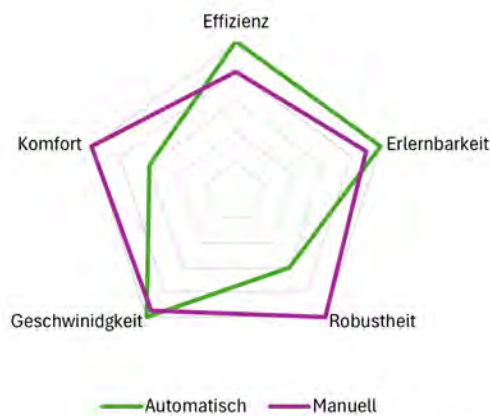


Abbildung 3.6: Einordnung der Gestaltungsoptionen der Interaktionskomponente Initialisierung anhand der relevanten Parameter Effizienz, Erlernbarkeit, Robustheit, (Interaktions-)Geschwindigkeit und Komfort

*Effizienz:* Die Effizienz der beiden Initialisierungsoptionen unterscheidet sich nur geringfügig. Die automatische Aktivierung ermöglicht eine kontinuierliche Verfügbarkeit des Scanning-Verfahrens, sodass Interaktionen jederzeit ohne vorherige Aktivierung durchgeführt werden können. Dies reduziert den Aufwand und erhöht die sofortige Bereitschaft für Interaktionen. Die manuelle Initialisierung erfordert hingegen einen zusätzlichen Zwischenschritt, bei dem die Nutzenden das Scanning-Verfahren aktivieren. Obwohl dieser Schritt allein nur minimalen Aufwand erfordert, kann er in Szenarien, in denen häufige Aktivierungen erforderlich sind, den erforderlichen Gesamtaufwand erhöhen.

*Effektivität:* Die Effektivität wird durch die Wahl der Initialisierungsoption kaum beeinflusst. Der Initialisierungsschritt bei der manuellen Aktivierung hat keinen Einfluss auf die Genauigkeit oder den Erfolg der Interaktion selbst, sondern nur auf den Beginn des Prozesses.

*Erlernbarkeit:* Die Erlernbarkeit der automatischen Aktivierung ist besonders hoch, da keine zusätzlichen Schritte oder Erklärungen erforderlich sind. Das Scanning-Verfahren wird automatisch gestartet und läuft kontinuierlich ab, was für die Nutzenden direkt ersichtlich ist. Im Gegensatz dazu erfordert die manuelle Initialisierung eine kurze Einführung, um den Aktivierungsmechanismus zu erklären. Da dies im Allgemeinen durch eine einfache Eingabe erfolgt, bleibt die zusätzliche kognitive Belastung gering und die Erlernbarkeit wird nur minimal beeinflusst.

*Robustheit:* Die Robustheit der manuellen Initialisierung wird als höher eingestuft, da das Scanning-Verfahren erst nach einer bewussten Aktivierung durch den Nutzenden startet. Dies minimiert das Risiko unbeabsichtigter Eingaben oder Fehlselektionen, die bei einer kontinuierlichen Aktivierung häufiger auftreten könnten. Die automatische Aktivierung hingegen erhöht die Wahrscheinlichkeit von Fehlselektionen, da das Scanning dauerhaft aktiv ist und unbeabsichtigte oder doppelte Eingaben potenziell direkt eine Interaktion auslösen.

*Interaktionsgeschwindigkeit:* Die Interaktionsgeschwindigkeit wird bei einem durchgehend aktivierten Scanning-Verfahren leicht positiv beeinflusst, da keine zusätzliche Initialisierung erforderlich ist. Die Nutzenden können direkt mit der Interaktion beginnen, was eine schnelle Ausführung ermöglicht. Im Vergleich dazu führt die manuelle Initialisierung zu einem minimalen Zeitverlust durch den zusätzlichen Zwischenschritt, der in der praktischen Anwendung jedoch vermutlich unerheblich ist.

*Komfort:* Der Komfort ist je nach Initialisierungsoption des Scanning-Verfahrens unterschiedlich. Eine automatische Aktivierung führt zu einer permanenten Überlagerung der Szene mit dem Scanning-Verfahren, was von den Nutzenden als störend empfunden werden könnte. Dies erhöht die visuelle Komplexität und erfordert eine stärkere Konzentration auf den eigentlichen Inhalt der Szene. Darüber hinaus könnte die kontinuierliche Bewegung des Scanning-Verfahrens bei längerer Nutzung ermüdend wirken und die immersive Wahrnehmung der Szene beeinträchtigen. Im Gegensatz dazu bietet die manuelle Initialisierung den Nutzenden die Möglichkeit, das Scanning-Verfahren nur bei Bedarf zu aktivieren. Dadurch wird die visuelle Komplexität der Szene reduziert, was zu einer entspannteren Nutzung und einer besseren Fokussierung auf wesentliche Inhalte beitragen könnte. Insbesondere bei längerer Nutzung oder bei Szenen mit komplexen visuellen Informationen kann dadurch der Komfort gesteigert werden. Darüber hinaus könnte die zeitweise Deaktivierung des Scanning-Verfahrens ein tieferes Eintauchen in die virtuelle Szene ermöglichen, da weniger Ablenkung durch visuelle Elemente besteht.

### 3.4 Ableitung der finalen Konzepte

Die Festlegung der zwei finalen Konzepte für die zu entwickelnden Interaktionsschnittstellen erfolgt auf Basis der detaillierten Bewertungen der Gestaltungsoptionen (vgl. [Abschnitt 3.3](#)). Dabei wird diejenige Option bevorzugt, die in der Gesamtbetrachtung die positivsten Bewertungen aufweist und ein ausgewogenes Verhältnis zwischen Effektivität, Effizienz, Robustheit, Interaktionsgeschwindigkeit, Erlernbarkeit und Komfort bietet. Der Komfort steht dabei besonders im Vordergrund, da das Auftreten von Motion Sickness einen erheblichen Einfluss auf die generelle Nutzbarkeit und Akzeptanz der Interaktionsschnittstellen hat. In den Bereichen, in denen keine der Optionen hinsichtlich der Parameter eindeutig überlegen ist, wird dieser Aspekt entsprechend stärker gewichtet. Damit soll sichergestellt werden, dass die ausgewählten Konzepte nicht nur funktional, sondern auch für die Nutzenden angenehm und zugänglich sind.

#### 1. Konzept: (Automatic) Item Scanning

Interaktionsaufgaben und -komponenten	Gestaltungsoptionen			
	Automatic Item Scanning	Step Item Scanning	Inverse Item Scanning	Continuous Cartesian Scanning
Selektion				
Navigation	Diskretes Navigationsmenü	Kontinuierliches Navigationsmenü	Direkte Navigation	Interaktionsmodi
Display	360°-Szene	Fixiert im Sichtfeld		
Transition	Direkte Rotation	Kontinuierliche Rotation		
Initialisierung	Manuell	Automatisch		

Abbildung 3.7: Zusammenstellung des 1. Konzepts Item Scanning anhand der Gestaltungsoptionen für Interaktionsaufgaben und -komponenten

Das erste Konzept (vgl. [Abbildung 3.7](#)) basiert auf dem Automatic Item Scanning, das sich durch eine vergleichsweise ausgewogene Bewertung in allen Parametern auszeichnet. Es erzielt zwar nicht in allen Kriterien die besten Ergebnisse, weist aber insgesamt die geringsten Defizite auf und bietet eine solide Grundlage für eine effektive und effiziente Interaktion. Die Navigation erfolgt über ein diskretes Navigationsmenü. Hier war die Bewertung der Optionen nicht eindeutig, jedoch überzeugt die diskrete Navigation insbesondere durch ihre deutlich geringeren Defizite bezüglich des Komforts. Dies ist insbesondere auf das geringere Risiko von Motion Sickness im Vergleich zur kontinuierlichen Navigation zurückzuführen.

Das Scanning-Verfahren wird im Sichtfeld der Nutzenden fixiert, da diese Gestaltungsoption der Komponente Display hinsichtlich aller Parameter durch klare Vorteile überzeugt. Ebenso wird eine direkte Rotation als Transitionsmethode verwendet, da diese sowohl hinsichtlich des Komforts als auch der Interaktionsgeschwindigkeit überzeugt. Die Initialisierung des Scanning-Verfahrens erfolgt manuell, da diese Variante im Vergleich zur automatischen Aktivierung zwar eine geringere Effizienz aufweist, aber insbesondere hinsichtlich Komfort und Robustheit Vorteile bietet.

Um mögliche Schwächen der gewählten Gestaltungsoptionen zu minimieren, wird das Scanning-Verfahren so gestaltet, dass es nur die Elemente einer aktiven Ebene durchläuft, z. B. nur die Antwortoptionen eines geöffneten Dialog-Elements. Dies reduziert die Größe des Scan-Sets und verkürzt die Interaktionszeit. Zusätzlich wird eine Zurück-Option integriert, die es erlaubt, den Dialog durch erneutes Auswählen wieder zu schließen. Um die Orientierung zu verbessern und eine klare Visualisierung der Scan-Reihenfolge zu gewährleisten, wird das Scanning-Verfahren durch eine Animation ergänzt, die die Scan-Reihenfolge verdeutlicht und dadurch die Robustheit erhöht.

## 2. Konzept: Cartesian Scanning

Interaktionsaufgaben und -komponenten	Gestaltungsoptionen			
	Automatic Item Scanning	Step Item Scanning	Inverse Item Scanning	Continuous Cartesian Scanning
Selektion				Interaktionsmodi
Navigation	Diskretes Navigationsmenü	Kontinuierliches Navigationsmenü	Direkte Navigation	
Display	360°-Szene	Fixiert im Sichtfeld		
Transition	Direkte Rotation	Kontinuierliche Rotation		
Initialisierung	Manuell	Automatisch		

Abbildung 3.8: Zusammenstellung des 2. Konzepts Cartesian Scanning anhand der Gestaltungsoptionen für Interaktionsaufgaben und -komponenten

Das zweite Konzept basiert auf dem Continuous Cartesian Scanning (vgl. [Abbildung 3.8](#)). Die Bewertungen der Selektionsoptionen unterscheiden sich nicht wesentlich. Das Cartesian Scanning wurde bevorzugt vor dem Step und Inverse Item Scanning ausgewählt, da es insbesondere hinsichtlich Komfort und Effizienz Vorteile aufweist. Für die Navigation werden Interaktionsmodi eingesetzt, da diese Option in den Bewertungen hinsichtlich Robustheit und auch Komfort leicht überlegen ist und eine klar strukturierte Interaktionsführung ermöglicht. Der Wechsel zwischen den Modi erfolgt durch kurzes Halten des Schalters. Diese Methode hat den Vorteil, dass unbeabsichtigte Wechsel minimiert werden und gleichzeitig eine einfache Bedienung ermöglicht wird. Alternative Wechselmecha-



nismen, wie ein doppeltes Drücken des Schalters oder die Integration eines zusätzlichen UI-Elements in die Szene, wurden verworfen, da sie entweder das Risiko unbeabsichtigter Eingaben erhöhen oder die visuelle Komplexität der Szene unnötig steigern würden.

Auch bei diesem Konzept wird das Scanning-Verfahren im Sichtfeld fixiert und die Transition erfolgt als direkte Rotation, da diese Gestaltungsoptionen deutliche Vorteile gegenüber den anderen Optionen aufweisen. Die Initialisierung erfolgt auch hier manuell, um unbeabsichtigte Aktivierungen zu vermeiden und den Komfort zu erhöhen.

Zur Optimierung des Cartesian Scanning werden unterschiedliche Farben für die Scan-Linien verwendet, um den aktiven Interaktionsmodus klar zu visualisieren. Zusätzlich ist das Scanning-Verfahren so gestaltet, dass die Linien nach ihrem ersten Durchlauf automatisch wieder von vorne beginnen, falls keine Eingabe erfolgt ist. Für den Fall, dass die erste Scan-Linie inkorrekt positioniert wurde, bietet das Verfahren die Möglichkeit, durch bewusstes mehrmaliges Durchlaufen der zweiten Scan-Linie den Scan abubrechen und die Auswahl zurückzusetzen. Diese Maßnahmen dienen insbesondere der Steigerung der Robustheit.

### **Zusammenfassung**

In diesem Kapitel wurde die Konzeption der Interaktionsverfahren durch eine systematische Analyse und Auswahl von Gestaltungsoptionen entlang definierter Parameter konkretisiert. Dabei wurden zwei in den Ausprägungen der Interaktionsaufgaben unterschiedliche Konzepte entwickelt, die auf den jeweiligen Stärken der analysierten Optionen aufbauen und durch gezielte Maßnahmen zur Kompensation möglicher Schwächen ergänzt werden. Während das erste Konzept auf dem Scanning-Verfahren Automatic Item Scanning basiert, wird für das zweite Konzept Continuous Cartesian Scanning als grundlegendes Interaktionsverfahren zur Selektion und Navigation gewählt. Im weiteren Verlauf der Arbeit werden die Konzepte als Vereinfachung über die verwendeten Scanning-Verfahren referenziert.

Im folgenden **Kapitel 4** wird die konkrete Umsetzung dieser Konzepte einschließlich der technischen Umsetzung der gewählten Gestaltungsoptionen sowie der definierten Maßnahmen zur Kompensation möglicher Schwächen vorgestellt.

## 4 Implementierung

In diesem Kapitel wird die technische Umsetzung der erarbeiteten Konzepte beschrieben. Dabei wird zunächst die grundlegende Architektur beschrieben, gefolgt von Grundlagen, die für beide Konzepte gleichermaßen entscheidend sind. Anschließend wird die spezifische Implementierung der beiden Konzepte tiefergehend vorgestellt. Hierbei wird zunächst das Item Scanning beschrieben und anschließend das Cartesian Scanning. Danach werden allgemeine Strategien zur Fehlerbehandlung sowie in der Entwicklung genutzte Debugging-Methoden vorgestellt. Abschließend werden Limitationen aufgezeigt und es wird ein kurzer Ausblick auf mögliche zukünftige Erweiterungen der Implementierung eröffnet.

### 4.1 Architektur

Die Implementierung der konzipierten Interaktionsschnittstellen erfolgt innerhalb des Tools PaneoVR. Die Entwicklung dieses Tools erfolgte in der *Game Engine Unity*, dementsprechend werden auch die Implementierungen im Rahmen dieser Arbeit in Unity durchgeführt. Die für die Implementierung relevanten Erweiterungen im Rahmen dieser Arbeit wurden in der *IngameScene* vorgenommen. Nachfolgend sind die zentralen Komponenten dieser Szene aufgeführt, die für die Weiterentwicklung im Rahmen dieser Arbeit relevant waren.

#### Zentrale Komponenten der IngameScene

- *Interaction Sphere*: Ein *GameObject*, an das beim Laden einer neuen Szene alle benötigten interaktiven Paneo-Elemente als Child-Objekte angehängt werden.
- *Scene Manager*: Enthält u. a. das Script *Scene Loader*, das für das dynamische Laden der interaktiven Elemente verantwortlich ist. Dieses Script wurde zur Implementierung des Item Scanning erweitert.

- *XR Rig Multiplayer*: Wird beim Start der Szene gespawnt und enthält u. a. die *Main Camera* sowie den *UI-Anchor*. Dem UI-Anchor sind zentrale UI-Komponenten wie das Ingame-Menü (Hauptmenü) und die Elemente für Nachrichten (Hint-Panel und Verify-Panel) untergeordnet.

### Erweiterungen der ursprünglichen Szene im Rahmen dieser Arbeit

Zur Implementierung der Scanning-Methoden wurden mehrere neue GameObjects und zugehörige Skripte in die IngameScene integriert (vgl. [Abbildung 4.1](#)). Diese werden folgend kurz aufgelistet und die wichtigsten Funktionalitäten vorgestellt.

- *SwitchHandler*: Dieses Script verarbeitet die Eingaben des Nutzens, die über das *Unity Input System* erfasst werden. Je nach gewählter Scanning-Methode (0 = Item Scanning, 1 = Cartesian Scanning) leitet der SwitchHandler die Eingaben an das entsprechende Script (*FindAndHighlightButtons* oder *Cartesian Handler*) weiter. Beim Cartesian Scanning überprüft der SwitchHandler, ob der Schalter länger als zwei Sekunden gehalten wurde, um den Interaktionsmodus zu wechseln.
- *FindAndHighlightButtons*: Hierbei handelt es sich um das zentrale Script für das Item Scanning. Es führt das schrittweise Hervorheben der interaktiven Elemente in der Szene aus und ermöglicht die Auswahl des aktuell hervorgehobenen Elements.
- *CartesianScanHandler*: Dies ist das zentrale Script für das Cartesian Scanning. Es steuert, zu welchem Zeitpunkt die Bewegungen der Scan-Linien aufgerufen werden, berechnet den Schnittpunkt der Scan-Linien, überprüft, ob sich hinter dem Schnittpunkt ein interaktives Element befindet, und führt die Selektion oder einen Moduswechsel aus.
- *VisibleByCamera*: Dieses Script ist insbesondere für das Item Scanning relevant und prüft, ob sich ein interaktives Element im Sichtfeld der Kamera befindet.
- *NextElementLine*: Dieses Script dient der Visualisierung der Scanning-Reihenfolge im Item Scanning. Durch die hier gesteuerten Animationen wird verdeutlicht, welches Element als nächstes hervorgehoben wird, um die Reihenfolge für Nutzende nachvollziehbar zu gestalten.
- *SoundHandler*: Dieses Script ist für die Steuerung von Feedback-Audioausgaben bei Interaktionen zuständig.

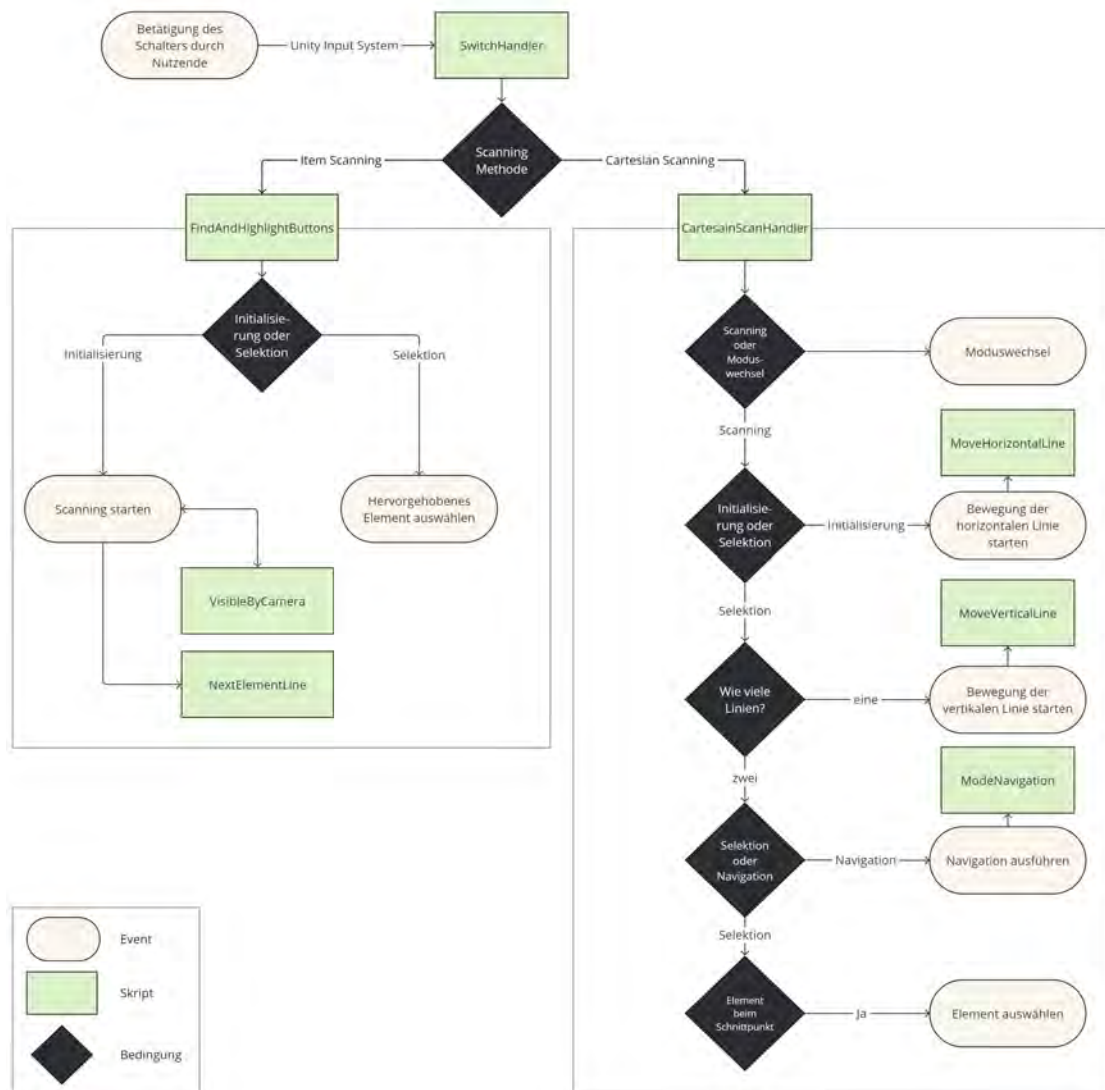


Abbildung 4.1: Grundlegender Informationsfluss der Softwarearchitektur

Die aufgeführten Skripte kommunizierten größtenteils über direkte Methodenaufrufe miteinander, was eine schnelle und direkte Weiterleitung von Eingaben und Verarbeitungsergebnissen ermöglicht. Um die Übersichtlichkeit und Wartbarkeit des Codes sicherzustellen, wurden die Scanning-Verfahren so gestaltet, dass für jedes ein eigenes Hauptskript mit der wichtigsten Funktionalität erstellt wurde. Dadurch befindet sich die gesamte Logik für jedes Verfahren an einem zentralen Ort. Es ist somit nicht erforderlich, zwischen mehreren Skripten zu wechseln, um spezifische Funktionen zu finden und anzupassen oder zu erweitern. Dennoch wurden einige Funktionen bewusst ausgelagert. Die Prüfung, ob ein Element sichtbar im Blickfeld der Kamera ist, wurde in ein separates Skript überführt. So kann diese Prüfung gegebenenfalls unabhängig vom Scanning-Verfahren wiederverwendet werden. Die Methoden zur Bewegung der Scan-Linien im Cartesian Scanning wurden direkt an die GameObjekte gebunden. Dies erleichtert die Konfiguration der Bewegung und fördert eine klare Trennung der Verantwortlichkeiten. Für die Ausführung der Navigation wurden ebenfalls zusätzliche Skripte erstellt. Das Skript *MenuNavigation* realisiert die Rotation des *XRRigs* beim Item Scanning, während das Skript *ModeNavigation* diese Funktionalität für das Cartesian Scanning übernimmt. Die beiden Funktionalitäten wurden dabei bewusst auf zwei Skripte geteilt, um die Unabhängigkeit der beiden Scanning-Verfahren beizubehalten.

Der grundlegende Informationsfluss, der zwischen diesen vorgestellten Skripten besteht, wird in [Abbildung 4.1](#) veranschaulicht. Die Eingaben der Nutzenden werden über das Unity Input System erfasst und zunächst im *SwitchHandler* verarbeitet. Abhängig von der gewählten Scanning-Methode wird die Eingabe an den entsprechenden *ScanningHandler* weitergeleitet. Beim Item Scanning übernimmt das Skript *FindAndHighlightButtons* die schrittweise Hervorhebung der Elemente sowie die Selektion des aktuell hervorgehobenen Elements. Beim Cartesian Scanning steuert der *CartesianScanHandler* die Animation der Linien, die Schnittpunktberechnung und die Selektion der interaktiven Elemente.

## 4.2 Gemeinsame Komponenten beider Konzepte

Die Implementierung der beiden Scanning-Methoden basiert u. a. auf gemeinsam genutzten Komponenten, die Funktionalitäten beider Interaktionsschnittstellen unterstützen. Dazu gehören die Eingabeverarbeitung, die Audioausgabe sowie ein zusätzliches Highlight-Game-Object, das allen interaktiven Elementen als Kindsobjekt hinzugefügt wird.

## Eingabeverarbeitung

Als Eingabe wurde zunächst eine neue Input Action erstellt, die die Leertaste der Tastatur als Simulation eines Schalters verwendet. Diese Lösung ermöglicht eine einfache Simulation während der Entwicklung und Evaluation, wenn kein Schalter zur Verfügung steht. Für verschiedene Arten von Schaltern kann diese Konfiguration zukünftig erweitert werden.

Für die Verarbeitung von Eingaben wurde das Skript *SwitchHandler* als zentrale Komponente implementiert. Dieses nutzt die Events *started* und *canceled* des Unity Input System, um die Aktionen des Nutzens zu erkennen und entsprechend zu verarbeiten. Bei der *Started-Action* wird die Methode *OnPressStarted* aufgerufen. In dieser wird der Zeitpunkt des Drückens zwischengespeichert und die Variable *isPressing* wird auf *true* gesetzt. Dies geschieht jedoch nur unter der Voraussetzung, dass das Cartesian Scanning als Scanning-Methode eingestellt ist. In der *Update-Methode* wird daraufhin kontinuierlich überprüft, ob *isPressing* auf *true* steht und ob seit dem Beginn des Drückens bereits zwei Sekunden vergangen sind. Wenn dies der Fall ist, wird ein akustisches Feedback ausgelöst, das dem Nutzenden signalisiert, dass ein Wechsel des Interaktionsmodus bei Loslassen des Schalters durchgeführt wird. Bei der *Canceled-Action* wird die Methode *OnPressEnded* aufgerufen. Hier wird zunächst geprüft, welche Scanning-Methode eingestellt ist. Ist das Item Scanning aktiv, wird die Eingabe an das *FindAndHighlightButtons*-Skript weitergeleitet. Ist das Cartesian Scanning aktiv, wird zunächst die Dauer des Tastendrucks geprüft. Wurde die Taste länger als zwei Sekunden gedrückt, wird die Methode *changeMode* im *CartesianScanHandler* aufgerufen, um den Interaktionsmodus zu wechseln. Andernfalls wird die Methode *Scanning* aufgerufen, um den Scanning-Vorgang zu starten. Die gewählte Scanning-Methode wird in einer Integer-Variable gespeichert, wobei 0 für das Item Scanning und 1 für das Cartesian Scanning steht.

## Audioausgabe

Die Anwendung bietet Nutzenden akustisches Feedback, das über den *AudioHandler* realisiert wird. Dieser ermöglicht das Erzeugen einer *AudioSource* zur Laufzeit, die den übergebenen *AudioClip* abspielt und anschließend automatisch wieder entfernt wird. Diese Implementierung ist ressourcenschonend und verhindert Probleme, die auftreten könnten, wenn eine *AudioSource* an ein *GameObject* gebunden ist und dieses während der Laufzeit entfernt wird. Durch diese Lösung wird sichergestellt, dass alle Sounds vollständig abgespielt werden, was insbesondere für Feedback-Sounds von zentraler Bedeutung ist.

## Highlight Objekt

Allen interaktiven Elementen wurde ein Highlight-Objekt als Child-Objekt hinzugefügt. Dieses Highlight besteht aus einer einfachen geometrischen Form (Sphere oder Cube), die mit einem transparenten Material versehen ist. Die Geometrie dient dazu, die Darstellung von Outlines zu ermöglichen, da diese nur an sichtbaren Meshes generiert werden können. Das Highlight-Objekt enthält zudem einen Mesh Renderer (initial deaktiviert), einen Collider und eine Outline, die mithilfe des kostenlosen Plugins Quick Outline [Chris Nolet, 2022] aus dem *Unity Asset Store* realisiert wurde. Das Highlight-Objekt wird von beiden Scanning-Methoden unterschiedlich genutzt. Beim Item Scanning wird die Outline durch Aktivieren des Mesh Renderers sichtbar gemacht, um zu visualisieren, dass dieses interaktive Element ausgewählt werden kann. Zusätzlich wird der Collider verwendet, um zu prüfen, ob das Element im Sichtfeld der Kamera liegt. Beim Cartesian Scanning wird der Collider ebenfalls verwendet. In diesem Zusammenhang jedoch um zu prüfen, ob hinter dem Schnittpunkt der Scan-Linien ein interaktives Element liegt. Auf die genaue Funktionsweise des Highlight-Objekts in den spezifischen Kontexten wird folgend in der Vorstellung der beiden Konzepte genauer eingegangen.

## 4.3 Scanning-Verfahren

### 4.3.1 Konzept 1 - Item Scanning

Das Item Scanning basiert auf einer Liste, die alle interaktiven Elemente einer Szene enthält. Diese interaktiven Elemente (Interactables) werden beim Laden einer Szene durch das Skript *SceneLoader* erzeugt und automatisch der Liste im Skript *FindAndHighlightButtons* hinzugefügt. Zusätzlich werden zwei UI-Buttons hinzugefügt, die immer als HUD sichtbar sind. Dabei handelt es sich um einen Button zum Öffnen des Hauptmenüs und einen Button zum Öffnen des Navigationsmenüs. Sobald der *SwitchHandler* eine Eingabe registriert, erfolgt eine Prüfung, ob das Scanning bereits aktiv ist oder initialisiert werden muss. Ist das Scanning noch nicht aktiv, wird die Methode *StartItemScanning* im Skript *FindAndHighlightButtons* aufgerufen, andernfalls die Methode *SwitchSelection*. Diese beiden Methoden stellen die zentrale Funktionalität des Scanning-Verfahrens dar. Im Folgenden werden diese und weitere relevante Aspekte näher beschrieben.

### Start und Verlauf des Scannings

Die Methode *StartItemScanning* startet eine Coroutine namens *ItemScanning*, in der der eigentliche Scan-Vorgang abläuft. Diese Coroutine iteriert über die Liste der interaktiven

Elemente und hebt sie nacheinander hervor. Der Ablauf ist im folgenden Pseudocode Listing 4.1 vereinfacht dargestellt.

```

1
2 IEnumerator ItemScanning()
3 {
4     while (isScanActive)
5     {
6         foreach (GameObject element in allElements)
7         {
8             // Versuche, das Highlight-Objekt zu finden
9             GameObject highlight = element.transform.Find("Highlight").
gameObject;
10
11             if (highlight != null)
12             {
13                 // Pruefe, ob das Highlight-Objekt sichtbar ist
14                 if (visibleByCamCheck.IsVisibleByCam(cam, highlight.
GetComponent<Collider>()))
15                 {
16                     // Aktiviere das Highlight-Objekt
17                     highlight.GetComponent<MeshRenderer>().enabled = true
;
18                     highlightedObject = element;
19
20                     // Starte die Animation zum naechsten Objekt
21                     GameObject nextObject = GetNextHighlightableObject();
22                     scriptNextElement.DrawLineBetweenElements(
highlightedObject, nextObject);
23
24                     // Warte fuer die Scan-Rate
25                     yield return new WaitForSeconds(ScanRate);
26
27                     // Deaktiviere das Highlight-Objekt, falls es noch
existiert
28                     if (highlight != null)
29                     {
30                         highlight.GetComponent<MeshRenderer>().enabled =
false;
31                     }
32                 }
33             }
34         }
35     }
36 }

```

Listing 4.1: Vereinfachter Pseudocode der Coroutine IEnumerator ItemScanning des Item Scannings. Realisierung der schrittweisen Hervorhebung interaktiver Elemente



Solange das Scanning aktiv ist, läuft diese Iteration. Für das aktuelle Element wird zunächst geprüft, ob es ein Highlight-GameObject als Kindsobjekt enthält. Diese Prüfung stellt sicher, dass wirklich nur Elemente in den Scan-Vorgang einbezogen werden, die auch visuell hervorgehoben werden sollen. Ist ein Highlight-Objekt vorhanden, wird über den Collider des Objekts ermittelt, ob es sich aktuell im Sichtfeld der Kamera befindet. Dies geschieht über das Skript *VisibleByCamera* in der Methode *IsVisibleByCam* (vgl. Listing 4.2). Ist diese Prüfung positiv, wird das Highlight durch Einschalten des Mesh Renderers aktiviert. Gleichzeitig wird das hervorgehobene Element in einer entsprechenden Variable gespeichert, um bei der Selektion darauf zugreifen zu können. Anschließend wird das nächste hervorzuhebende Element bestimmt und eine Animation gestartet, die eine Linie zwischen dem aktuellen und dem nächsten Element visualisiert. Diese Animation wird weiter unten ausführlicher behandelt. Nach Ablauf der Wartezeit, die durch die Scan Rate bestimmt wird, wird der Mesh Renderer des Highlight-Objektes wieder deaktiviert, sofern das Element noch existiert. Dies ist notwendig, da das Element unter Umständen während der Wartezeit gelöscht werden kann. Würde an dieser Stelle keine erneute Überprüfung erfolgen, käme es zu einer *NullPointerException*.

```
1
2     public bool IsVisibleByCam(Camera cam, Collider targetObjCollider)
3     {
4         Plane[] planes = GeometryUtility.CalculateFrustumPlanes(cam);
5         if(GeometryUtility.TestPlanesAABB(planes, targetObjCollider.
6         bounds))
7         {
8             return true;
9         }
10        else
11        {
12            return false;
13        }
14    }
```

Listing 4.2: Methode IsVisibleByCam: Umsetzung der Überprüfung, ob sich ein Element im Sichtfeld des Nutzens befindet

### Auswahl eines interaktiven Elements

Die Auswahl eines hervorgehobenen Elements erfolgt über die Methode *SwitchSelection*. Hier wird geprüft, ob ein Element aktuell hervorgehoben ist. Ist dies der Fall, wird die Button-Komponente des Elements gesucht und deren *OnClick-Event* ausgelöst. Der Ablauf ist im folgenden Pseudocode Listing 4.3 veranschaulicht.

```

1
2 public void SwitchSelection()
3 {
4     if (highlightedObject != null)
5     {
6         // Suche nach der Button-Komponente
7         GameObject buttonTrigger = highlightedObject.transform.Find("
Btn_Trigger").gameObject;
8         Button buttonToClick = buttonTrigger.GetComponent<Button>();
9
10        // Loese das OnClick-Event aus
11        if (buttonToClick != null)
12        {
13            ExecuteEvents.Execute(
14                buttonToClick.gameObject,
15                new BaseEventData(EventSystem.current),
16                ExecuteEvents.submitHandler
17            );
18        }
19    }
20 }

```

Listing 4.3: Vereinfachter Pseudocode der Methode SwitchSelection des Item Scannings.  
Realisierung der Selektion des hervorgehobenen Elements

Die gleiche Logik wird für die Navigation verwendet. Wenn ein Pfeilbutton im Navigationsmenü ausgewählt wird, wird das OnClick-Event des Buttons ausgelöst, welches wiederum die Methode *turnLeft* bzw. *turnRight* im Skript *MenuNavigation* aufruft. Diese Methoden drehen das XRRig um 40° (rechts) bzw. -40° (links).

### Animation zwischen dem aktuellen und folgenden Element

Um die Reihenfolge zu verdeutlichen, in der die interaktiven Elemente hervorgehoben werden, wird eine Linie zwischen dem aktuellen und dem nächsten Element animiert (siehe [Abbildung B.1](#)). Dazu wurde der *IngameScene* das GameObject *NextElementLine* hinzugefügt. Dieses enthält einen Line Renderer zur Darstellung der Linie sowie das Skript *NextElementLine*, in dem die Steuerung der Animation durchgeführt wird. Die Animation wird als Coroutine umgesetzt. Der Startpunkt der Linie ist die Position des aktuell hervorgehobenen Elements, der Endpunkt die Position des nächsten Elements. Die Sichtbarkeit der Linie wird über die *AlphaKeys* des Line Renderers gesteuert, die von 0 auf 1 gesetzt werden. Dadurch wird die Linie schrittweise sichtbar. Die Animationsdauer entspricht dabei der Scan Rate. Damit die Linie die Position dynamischer UI-Elemente korrekt darstellt, werden Start- und Endpunkte in der Update-Methode jeden Frame aktualisiert.

## Herausforderungen

Die Herausforderungen bei der Implementierung des Item Scanning lagen insbesondere in der Unterscheidung der verschiedenen Ebenen. Wird z. B. ein Menü oder ein Dialogfeld geöffnet, so muss das Scanning auf diese Teilauswahl von Items beschränkt werden. Die Lösung dieses Problems wird im Folgenden exemplarisch an einem Dialog-Element beschrieben. Wird das Dialog-Element innerhalb des Item Scanning ausgewählt, wird das OnClick-Ereignis ausgelöst. Darüber wird die Methode *DialogScan* aufgerufen. Dieser Methode werden beim Aufruf die Buttons der Antwortoptionen übergeben. Aus diesen Optionen wird ein Array erstellt, das dann mit der oben beschriebenen Logik durchlaufen werden kann. Die Auswahloptionen müssen also dynamisch angepasst werden, je nachdem welches Element aktiv ist bzw. sich auf der vordersten Interaktionsebene befindet. Dies erfordert besondere Aufmerksamkeit bei der Implementierung, da verschiedene Fälle berücksichtigt werden müssen. Für jedes Element, das eine weitere Ebene öffnet, muss geprüft werden, wie sich die Auswahl der Scan-Optionen dadurch ändert. Eine weitere Herausforderung bei der Implementierung des Item Scanning bestand darin, das Scanning-Set dynamisch auf die Elemente zu beschränken, die sich im aktuellen Sichtfeld des Nutzers befinden. Die Herausforderung bestand darin, eine Lösung zu finden, die die Performance der Anwendung nicht wesentlich beeinträchtigt. Dies ist insbesondere bei größeren Kopfbewegungen relevant, bei denen sich das Sichtfeld abrupt und deutlich ändert und das Scanning-Set entsprechend schnell und ressourcenschonend angepasst werden muss. Die implementierte Lösung dieser Überprüfung ist in [Listing 4.2](#) dargestellt.

### 4.3.2 Konzept 2 - Cartesian Scanning

Um das Cartesian Scanning zu implementieren, wurden zwei neue GameObjects erstellt und der Kamera im XRRig untergeordnet. Diese GameObjects enthalten jeweils ein Canvas, das annähernd auf die Größe des Sichtfeldes skaliert ist. Auf diesem Canvas befindet sich wiederum ein GameObject mit einer Rect Transform-Komponente, einem Line Renderer zur Darstellung der Scan-Linien sowie dem Skript *MoveHorizontalLine* bzw. *MoveVerticalLine*. Die GameObjects sind zu Beginn der Anwendung deaktiviert und werden erst während des Scan-Vorgangs aktiviert. Die horizontale Scan-Linie bewegt sich von oben nach unten vor dem Nutzenden, während die vertikale Scan-Linie sich von links nach rechts bewegt. Die Bewegung der Scan-Linien wird über die genannten Skripte gesteuert. Der Aufbau dieser Skripte wird im Folgenden näher beschrieben.

## Bewegung der Scan-Linien

Die Methode *StartAnimation* startet die Coroutine *MoveLine*, in der die Bewegung der Scan-Linie gesteuert wird. Hier wird zunächst die Start- und Endposition der Linie definiert. Bei der horizontalen Scan-Linie sind dies der obere und der untere Rand des Canvas. Bei der vertikalen Scan-Linie entspricht dies dem linken und rechten Rand des Canvas. Nachfolgend ist in Listing 4.4 ein Ausschnitt aus dieser Coroutine zur Veranschaulichung der Bewegung dargestellt.

```
1
2 while (currentTime <= scanRate)
3 {
4     currentTime += Time.deltaTime;
5     normalizedValue = currentTime / scanRate;
6     rectTransform.anchoredPosition = Vector3.Lerp(startPosition,
7     endPosition, normalizedValue);
8     yield return null;
9 }
```

Listing 4.4: Ausschnitt aus der Methode *MoveLine* zur Realisierung der Bewegung der Scan-Linien im Cartesian Scanning

Sobald die Scan-Linie ihre Endposition erreicht hat, wird über eine Zählervariable geprüft, ob die gesamte Bewegung bereits dreimal hintereinander ausgeführt wurde. Ist die Scan-Linie noch nicht dreimal durchgelaufen, wird die Bewegung zurückgesetzt und erneut gestartet. Erst nach dem dritten Durchlauf wird die *Reset*-Methode im *CartesianScanHandler* aufgerufen, wo der gesamte Fortschritt des Scannings zurückgesetzt wird. Dieses wiederholte Durchlaufen wurde zur Fehlerkorrektur und somit zur Erhöhung der Robustheit implementiert. Wird die gewünschte Position der Scan-Linie beim ersten Durchlauf verfehlt, kann abgewartet werden, bis die Scan-Linie wieder die gewünschte Position erreicht hat. Durch den Abbruch nach drei Durchläufen werden zudem vermeidbare Fehlauswahlen vermieden.

Eine weitere wesentliche Methode innerhalb des Skripts ist *StopAnimation*. Diese bewirkt, dass die aktuell laufende Coroutine gestoppt wird, sodass die Scan-Linie in ihrer aktuellen Position verbleibt. Die Methode wird aufgerufen, sobald Nutzende die Position der Scan-Linie im Zuge des Scannings setzen möchten.

## Steuerung des Scanning-Prozesses

Die eigentliche Steuerung des Cartesian Scanning erfolgt im Skript *CartesianScanHandler*. Wird im *SwitchHandler* eine Eingabe registriert, wird die Methode *Scanning* im *CartesianScanHandler* aufgerufen. Diese prüft, in welchem Zustand sich der Scanning-Prozess befindet, und ruft entsprechend die nächste erforderliche Methode auf (vgl. [Listing 4.5](#)).

```
1 public void Scanning()
2 {
3     if (!firstStarted)
4     {
5         ScanningFirstLine();
6     }
7     else
8     {
9         if (!secondStarted)
10        {
11            StopFirstLine();
12        }
13        else
14        {
15            Selection();
16        }
17    }
18 }
19 }
```

Listing 4.5: Methode *Scanning* im *CartesianScanHandler*: Überprüfung des aktuellen Zustands des Scannings

Die Methode *ScanningFirstLine* aktiviert das *GameObject* der horizontalen Scan-Linie und ruft die Methode *StartAnimation* im Skript *MoveHorizontalLine* auf. In der Methode *StopFirstLine* wird diese durch den Methodenaufruf *StopAnimation* gestoppt, das *GameObject* der vertikalen Scan-Linie aktiviert und die Bewegung dieser gestartet. In der Methode *Selection* wird anschließend die Bewegung der zweiten Scan-Linie gestoppt, die beiden *GameObjects* wieder deaktiviert, die Bewegung zurückgesetzt und die Variablen *firstStarted* und *secondStarted* wieder auf *false* gesetzt. Um die Selektion durchzuführen, wird die Methode *FindIntersection* aufgerufen. Hier wird der Schnittpunkt der Scan-Linien berechnet. Diese Berechnung wird im Folgenden anhand eines vereinfachten Pseudocodes veranschaulicht (vgl. [Listing 4.6](#)).

```
1
2 public void FindIntersection()
3 {
4     // Positionen der Linien im Canvas Space
5     Vector2 horizontalLinePosition = horizontalLine.anchoredPosition;
6     Vector2 verticalLinePosition = verticalLine.anchoredPosition;
7
8     // Berechnung Schnittpunkts Canvas Space
9     Vector2 intersectionInCanvasSpace = new Vector2(
10     verticalLinePosition.x, horizontalLinePosition.y );
11
12     // Umwandlung in Weltkoordinaten
13     Vector3 intersectionWorldPosition = horizontalLine.transform.
14     parent.TransformPoint(intersectionInCanvasSpace);
15
16     // Selektionsmodus ist aktiv
17     if (!navModeActive)
18     {
19         CheckForInteractiveElement(intersectionWorldPosition);
20     }
21     // Navigationsmodus ist aktiviert
22     else
23     {
24         modeNav.turn(intersectionWorldPosition);
25     }
26 }
```

Listing 4.6: Vereinfachter Pseudocode der Methode FindIntersection im CartesianScanHandler: Berechnung des Schnittpunktes der Scan-Linien

Je nachdem, welcher Interaktionsmodus aktiv ist, wird unterschiedlich verfahren. Ist der Selektionsmodus aktiviert, wird mit Hilfe der Methode *CheckForInteractiveElement* überprüft, ob sich hinter dem Schnittpunkt der Scan-Linien ein interaktives Element befindet. Dazu wird mittels eines *SphereCasts* ein Ray von der Kameraposition durch den berechneten Schnittpunkt erzeugt. An dieser Stelle wurde ein SphereCast anstelle eines einfachen RayCasts gewählt, um einen Toleranzbereich in Form eines Radius einführen zu können. Dadurch wird ein Element auch dann ausgewählt, wenn der Schnittpunkt das Element knapp verfehlt. Es wird geprüft, ob dieser Ray den Collider eines Highlight-Objektes getroffen hat. Ist dies der Fall, wird wie beim Item Scanning das OnClick-Event der Button-Komponente des entsprechenden Elements ausgelöst. Trifft der Ray auf mehrere Objekte, z. B. wenn zwei Ebenen mit interaktiven Elementen übereinander liegen, wird stets das Element ausgewählt, auf das der Ray zuerst trifft. Das bedeutet konkret, es wird das Element ausgewählt, das näher an der Kamera positioniert ist. Dadurch wird sichergestellt, dass UI-Elemente wie das Hauptmenü oder das Hint-Panel bevorzugt ausgewählt werden und keine unbeabsichtigten Selektionen bei Verdeckungen auftreten. Wurde kein Element getroffen, wird lediglich ein Audio-Feedback ausgelöst.

Im Navigationsmodus hingegen wird die Methode *turn* im Skript *ModeNavigation* aufgerufen. Hier wird die y-Koordinate des Schnittpunkts genutzt, um die Rotation des XRRigs entsprechend anzupassen.

### Wechsel zwischen Selektions- und Navigationsmodus

Wird im *SwitchHandler* registriert, dass der Schalter länger als zwei Sekunden gehalten wurde, wird die Methode *changeMode* im *CartesianScanHandler* aufgerufen. In dieser Methode wird die boolesche Variable *navModeActive* entsprechend geändert. War diese zuvor auf *true*, so wird sie nun auf *false* gesetzt und umgekehrt. Außerdem wird die Farbe der Linie *Renderer* angepasst. Ist der Navigationsmodus aktiv, werden die Linien blau dargestellt, ist der Selektionsmodus aktiv, werden die Linien pink dargestellt. Dies dient dazu, den Nutzenden eine visuelle Rückmeldung darüber zu geben, welcher Modus gerade aktiv ist.

### Herausforderungen

Die Implementierung des Cartesian Scanning war insgesamt aufwendiger als die Implementierung des Item Scanning. Es werden mehr Objekte und Skripte benötigt, um das Scanning zu realisieren. Ist das Grundprinzip jedoch einmal implementiert, ist es vergleichsweise flexibel. Ein Vorteil dieses Scanning-Verfahrens liegt insbesondere darin, dass Elemente auf unterschiedlichen Ebenen nicht separat betrachtet werden müssen. Dadurch kann die Implementierung insgesamt einfacher um neue Interaktionselemente erweitert werden. Eine Herausforderung bei der Implementierung war die Berechnung des korrekten Schnittpunktes sowie die Festlegung eines geeigneten Radius für den *SphereCast*. Bei der Berechnung des Schnittpunktes musste berücksichtigt werden, dass sich die Scan-Linien auf einem Canvas befinden und somit ihre Position in Abhängigkeit von diesem bestimmt wird. Um die tatsächliche Position in Weltkoordinaten zu erhalten, muss zunächst eine Umrechnung erfolgen (vgl. [Listing 4.6](#)). Die Festlegung des Radius des *SphereCasts* stellte eine Herausforderung dar, da hier ein geeigneter Kompromiss gefunden werden musste. Der Radius musste groß genug sein, um kleine Verfehlungen zu tolerieren, aber klein genug, um bei dicht beieinanderliegenden Elementen eine korrekte Auswahl zu ermöglichen. Es war daher notwendig, mit verschiedenen Werten zu experimentieren und zu prüfen, ob sich ein zufriedenstellendes Ergebnis einstellt.

## 4.4 Strategien zur Fehlervermeidung und Debugging

Um Fehlermeldungen vorzubeugen und die Anwendung robuster zu gestalten, wurde an vielen Stellen im Code mit *try and catch*-Abfragen gearbeitet. Insbesondere *NullPointerException* werden so abgefangen. Dies wird z. B. beim Item Scanning verwendet. Wenn bspw. ein Element kein Highlight-Objekt hat oder das Element zur Laufzeit der Coroutine gelöscht wird, wird dies abgefangen und das Element kann problemlos übersprungen werden. Des Weiteren wurde viel mit boolschen Variablen gearbeitet, um die Abläufe besser verfolgen und steuern zu können. So wird z. B. verhindert, dass zwei Item Scanning Prozesse gleichzeitig gestartet werden. Beim Cartesian Scanning geben die Funktionen bei der Selektion (*CheckForInteractiveElement* und *modeNav.turn*) boolsche Werte zurück, die eine Rückmeldung darüber liefern, ob die Ausführung erfolgreich war. So können die verschiedenen Ausgänge mit unterschiedlichem Audio-Feedback kommuniziert werden.

Für das Debugging wurde insbesondere die Debug-Klasse von Unity verwendet. An mehreren Stellen wurden *Debug.Logs* eingefügt, um den Ablauf besser nachvollziehen zu können und schneller zu erkennen, an welcher Stelle im Code Fehler auftreten. Für das Cartesian Scanning wurden zusätzlich einzelne Schritte visuell veranschaulicht. So wurde bspw. während der Entwicklung ein Testobjekt in Form eines einfachen Würfels an die Position des berechneten Schnittpunktes gelegt, um leichter überprüfen zu können, ob dieser korrekt berechnet wird. Ebenso wurde der *SphereCast* mittels *Debug.DrawRay* visualisiert, um in der Scene View überprüfen zu können, ob dieser korrekt gezeichnet wird.

## 4.5 Limitationen und mögliche Erweiterungen

Da die vorliegende Arbeit eine prototypische Implementierung umfasst, bestehen noch mehrere Limitationen, die zukünftige Weiterentwicklungen erforderlich machen. Im Rahmen der Implementierung wurde die Leertaste einer Tastatur als Ersatz für einen physischen Schalter verwendet, um die Interaktion zu simulieren. Für den praktischen Einsatz mit einem physischen Schalter müssten Anpassungen vorgenommen werden, um sicherzustellen, dass der Input eines Schalters korrekt erkannt und verarbeitet wird.

Eine zentrale Einschränkung beim Item Scanning liegt in der fehlenden Konfigurierbarkeit der Reihenfolge, in der interaktive Elemente hervorgehoben werden. Gegenwärtig basiert die Scan-Reihenfolge auf der Reihenfolge, in der die Elemente beim Laden der



Szene erzeugt werden. Dies kann dazu führen, dass in der Szene räumlich nahe beieinander liegende Elemente nicht nacheinander hervorgehoben werden, was von Nutzenden als nicht erwartungskonform wahrgenommen werden könnte. Um dieses Problem zu beheben, könnte ein Parameter „Position im Scanning“ in die Prefabs der interaktiven Elemente integriert werden. Durch eine Sortierung der Liste der interaktiven Elemente anhand dieses Parameters könnte die Reihenfolge an die räumliche Anordnung angepasst werden. Dieser Parameter könnte anschließend entweder in Webeditor von PaneoVR oder im Editor-Modus eingestellt werden.

Eine weitere Limitation betrifft die Konfigurierbarkeit der Scan Rate und des Scanning-Verfahrens. Diese Einstellungen können derzeit nur über den Unity-Editor angepasst werden, sodass die Nutzenden keine Möglichkeit haben, diese direkt innerhalb der Anwendung zu ändern. Diese Einschränkung wurde bewusst in Kauf genommen, um die Durchführung der Evaluation zu vereinfachen und vergleichbare Ergebnisse zu gewährleisten. Dies wird in **Kapitel 5** ausführlicher erläutert. Für den späteren produktiven Einsatz der Scanning-Verfahren wäre jedoch eine entsprechende Anpassungsoption notwendig, um die individuellen Präferenzen und Bedürfnisse der Nutzenden berücksichtigen zu können.

Beim Catesian Scanning stellt die Platzierung der Scan-Linien auf einem an das Sichtfeld der Nutzenden angepassten Canvas eine mögliche Einschränkung dar. Für einige Nutzende könnte dies möglicherweise nicht optimal sein. Zukünftige Entwicklungen könnten hier Anpassungsoptionen für den Scan-Bereich vorsehen, um die Usability zu erhöhen. Eine weitere Einschränkung liegt in der planaren Darstellung der Scan-Linien auf einer Ebene. Es wäre sinnvoll zu untersuchen, ob eine sphärische Darstellung, die sich an der Sphäre des 360-Grad-Videos orientiert, die Interaktion noch natürlicher gestalten und die Präzision erhöhen könnte.

Außerdem könnte getestet werden, ob ein multimodales Feedback das Cartesian Scanning noch unterstützen könnte. Beim Item Scanning wurde bspw. ein Audiosignal hinzugefügt, das jedes Mal ertönt, wenn ein neues Element hervorgehoben wird. Ein ähnliches Prinzip könnte beim Cartesian Scanning eingeführt werden. Hier könnte bspw. ein Audiosignal ertönen, wenn eine Scan-Linie während der Bewegung mit einem interaktiven Element kollidiert.

## 4.6 Nutzung der Implementierung

Der im Rahmen dieser Arbeit entwickelte Prototyp steht sowohl als ausführbare Build-Datei (.apk) als auch als vollständiges Unity-Projekt zur Verfügung (siehe [Anhang E](#)). Die Build-Dateien, die sich im Anhang befinden, umfassen separate Versionen für das Item und das Cartesian Scanning. Diese können beispielsweise über den *Meta Quest Developer Hub* auf einer Meta Quest 3 installiert werden. Alternativ kann das bereitgestellte Unity-Projekt genutzt werden, um die Anwendung direkt über den Unity-Editor zu starten. Hierfür ist eine Verbindung der VR-Brille mit dem PC über *QuestLink* erforderlich. Es ist wichtig, die Anwendung stets in der *StartupScene* zu starten. Die Anpassung des verwendeten Scanning-Verfahrens kann in der *IngameScene* vorgenommen werden. Dazu kann das GameObject *SwitchHandler* ausgewählt und in der zugehörigen *SwitchHandler*-Component im Inspector das gewünschte Scanning-Verfahren eingestellt werden.

## 5 Evaluation

Die Evaluation der entwickelten Interaktionsschnittstellen erfolgt in zwei Hauptabschnitten: einer technischen und einer inhaltsbasierten Evaluation. Ziel ist es, anhand spezifisch entwickelter Szenarien sowohl die technische Leistungsfähigkeit der Schnittstellen als auch deren Wechselwirkung mit den Inhalten zu analysieren. Im folgenden wird zunächst die technische Evaluation gefolgt von der inhaltsbasierten Evaluation beschrieben. Anschließend wird die Durchführung der Evaluation geschildert und die Ergebnisse werden vorgestellt.

### 5.1 Konzeption der Evaluation

#### 5.1.1 Technische Evaluation

Der technische Abschnitt der Evaluation dient insbesondere der Überprüfung der Annahmen zu technischen Parametern wie Geschwindigkeit, Effizienz, Robustheit und Erlernbarkeit, die im Rahmen der Konzeption getroffen wurden. Darüber hinaus soll die grundsätzliche Usability der Schnittstellen evaluiert werden. Für jede Schnittstelle wird ein spezifisches Szenario entwickelt, in dem die technischen Fähigkeiten der Schnittstellen in möglichst anspruchsvollen Situationen getestet werden. Die Szenarien sind in Bezug auf die verwendeten Elemente und die Anzahl der durchzuführenden Eingaben vergleichbar. Beide Szenarien bestehen aus zwei Abschnitten. Im ersten Abschnitt steht die einfache Selektion von Elementen im Vordergrund. Im zweiten Abschnitt hingegen liegt der Schwerpunkt auf der Interaktion mit Dialogfeldern und damit auf der Interaktion auf verschiedenen Ebenen sowie auf der Selektion von Elementen, die räumlich nahe beieinander liegen. Ziel dieses Aufbaus ist es, möglichst viele Interaktionen zu generieren, um eine breite Datengrundlage zu erhalten. Darüber hinaus ermöglicht das Design die Beobachtung möglicher Lerneffekte bei den Teilnehmenden. Die Anordnung der Elemente innerhalb der Szenen ist absichtlich herausfordernd gestaltet, um insbesondere die Robustheit und Effektivität der Schnittstellen zu erproben. Darüber hinaus sind die Elemente über die gesamte 360°-Szene verteilt, um die Nutzung der Navigation

zu forcieren. Im Folgenden werden die Herausforderungen, die bei der Anordnung der Elemente innerhalb der Szene berücksichtigt wurden, im Detail dargestellt.

### **Herausforderungen der implementierten Interaktionsschnittstellen**

Als Herausforderung bei der Verwendung von Item Scanning wird zunächst eine hohe Anzahl von Elementen in der Szene betrachtet, deren Reihenfolge im Scanning nicht unmittelbar ersichtlich ist. Dies erfordert ein aktives und konzentriertes Verfolgen der Reihenfolge, um das gewünschte Element auswählen zu können. Darüber hinaus wird die Positionierung von Elementen am Rand des Sichtfeldes als Herausforderung angesehen. Diese Elemente könnten von den Nutzenden leicht übersehen oder in der Scan-Reihenfolge nicht erwartet werden, was zu fehlerhaften oder unbeabsichtigten Selektionen führen könnte. Darüber hinaus stellen längere Wartezeiten für Elemente, die erst spät in der Scanning-Reihenfolge erscheinen, eine weitere potenzielle Herausforderung dar. Diese könnten zu Ungeduld und damit zu fehlerhaften Eingaben führen. Auch die Interaktion auf mehreren Ebenen, z. B. das Schließen von Pop-Ups oder das Navigieren in Dialogfeldern, wird als weitere Herausforderung angesehen.

Im Gegensatz dazu ist anzunehmen, dass die größte Herausforderung beim Cartesian Scanning darin besteht, das gewünschte Element aus nahe beieinander liegenden Elementen auszuwählen. Je näher die Elemente beieinander liegen, desto präziser muss die Selektion erfolgen. Dies erfordert dementsprechend ein genaues Timing und eine hohe Konzentration. Darüber hinaus könnte auch bei diesem Scanning-Verfahren die Positionierung von Elementen am Rand des Sichtfeldes als Herausforderung darstellen. Insbesondere wenn sich Elemente weit oben oder weit links befinden, kann es schnell passieren, dass Nutzende die Selektion verpassen und auf einen weiteren Durchlauf der Scan-Linie warten müssen. Als letzte Herausforderung wird die Interaktion auf mehreren Ebenen gesehen. Insbesondere bei beweglichen Pop-Ups oder Menüs, die sich mit der Kopfposition im Sichtfeld bewegen, könnte es schnell zu Verschiebungen und damit zu Fehleingaben kommen. Auch hier ist somit eine erhöhte Konzentration erforderlich.

Die Szenarien für den technischen Abschnitt der Evaluation werden so gestaltet, dass diese Herausforderungen gezielt getestet werden. Während im ersten Abschnitt die Positionierung der Elemente im Vordergrund steht, werden im zweiten Abschnitt die Nähe und die Ebenenstruktur der Elemente aufgegriffen. Zur Minimierung von Zufallseffekten wurde eine Mindestanzahl von fünf gleichartigen Elementen pro Szenario integriert. **Anhang B** beinhaltet Abbildungen der erstellten Szenarien.

## Fragestellungen und Datenerhebung

Durch die Datenerhebung im technischen Abschnitt sollen insbesondere folgende Fragestellungen zu den genannten Parametern beantwortet werden können:

### *Geschwindigkeit*

- Wie lange dauert es, das Szenario zu durchlaufen? Begünstigt eine Schnittstelle einen schnelleren Durchlauf?
- Wie lang ist die Interaktionsgeschwindigkeit bei den Schnittstellen? Bietet eine Schnittstelle eine deutlich schnellere Interaktionsgeschwindigkeit?
- Gibt es Faktoren, die die Interaktionsgeschwindigkeit beeinflussen, etwa die Position der Elemente im Sichtfeld?
- Gibt es beim Cartesian Scanning deutliche Unterschiede in der Interaktionsgeschwindigkeit abhängig von der Position der Elemente?
- Führt eine schnellere Interaktionsgeschwindigkeit zu einer besseren Bewertung der Usability?

### *Robustheit*

- Wie häufig treten Fehler bzw. unbeabsichtigte Eingaben bei den jeweiligen Schnittstellen auf?
- Was sind die Gründe für auftretende Fehler?
- Wie häufig werden zwei oder mehr Durchgänge im Scanning benötigt?
- Wie häufig werden beim Cartesian Scanning leere Eingaben zur Korrektur von Fehlern verwendet?

### *Erlernbarkeit*

- Zeigt sich im Verlauf des Szenarios ein deutlicher Lerneffekt?
- Wie entwickelt sich die Interaktionsgeschwindigkeit im Verlauf des Szenarios?
- Reduziert sich die Anzahl der Fehler mit zunehmender Erfahrung?

### Usability

- Wie wird die Usability der Schnittstellen bewertet?
- Gibt es deutliche Unterschiede in der Bewertung der Usability zwischen den beiden Schnittstellen?

Die Erfassung der technischen Daten erfolgt durch die Kombination von automatischem Logging direkt aus Unity und einem Evaluationsprotokoll (siehe [Anhang A](#)). Es wird für jede durchgeführte Selektion die Interaktionsgeschwindigkeit ausgegeben. Außerdem wird die benötigte Zeit von Start bis Abschluss des Szenarios gespeichert. Fehler werden manuell gezählt und kategorisiert und zusätzliche Beobachtungen dokumentiert. Zur Evaluation der Usability wird die SUS [Brooke, 1996] in digitaler Form verwendet.

### Erwartete Ergebnisse der technischen Evaluation

Die Erwartungen an die Ergebnisse der technischen Evaluation der Schnittstellen basieren auf den theoretischen Überlegungen, die im Rahmen der Konzeption aufgestellt wurden (vgl. [Abschnitt 3.3](#)). Es wird erwartet, dass das Item Scanning intuitiver und schneller erlernbar ist als das Cartesian Scanning. Außerdem wird erwartet, dass Item Scanning robuster ist, was sich in einer geringeren Anzahl von Fehlern in der Anwendung widerspiegelt. Beim Cartesian Scanning können insbesondere nah beieinander liegende Elemente oder Elemente am Rand des Sichtfeldes eine höhere Fehlerquote begünstigen. Auch hinsichtlich der Interaktionsgeschwindigkeit wird eine leichte Überlegenheit des Item Scanning erwartet. Gleichzeitig wird angenommen, dass das Item Scanning bei der Navigation insgesamt etwas langsamer ist, da mehrere Eingaben zum Erreichen eines größeren Rotationswinkels erforderlich sind. Das Cartesian Scanning hingegen ermöglicht eine effizientere Navigation und deckt größere Rotationswinkel schneller ab. Generell wird erwartet, dass das Item Scanning zu besseren technischen Ergebnissen führt.

#### 5.1.2 Inhaltsbasierte Evaluation

Der zweite Abschnitt der Evaluation widmet sich der Wechselwirkung zwischen Interaktionsschnittstelle und Inhalt. Hier stehen die Parameter Komfort und Effizienz im Vordergrund. Darüber hinaus wird die UX der Schnittstellen betrachtet. Die inhaltsbasierte Evaluation ergänzt den technischen Abschnitt, indem die Schnittstellen in realistischeren Anwendungsszenarien getestet werden. Es werden zwei Szenarien in Form einfacher Rätsel nach dem Prinzip von *Escape Rooms* erstellt. Die Aufgabe der Teilnehmenden besteht darin, einen hinter interaktiven Elementen wie Bildern oder Audiodateien versteckten

Zahlencode zu ermitteln (Szenario Code) bzw. einen Schlüssel zur Lösung des Szenarios zu finden (Szenario Schlüssel). Die Rätsel sind bewusst einfach gehalten, um die kognitiven Fähigkeiten der Teilnehmenden außerhalb der Verwendung der Schnittstellen nicht zu sehr fordern, da dies nicht im Fokus der Evaluation steht. Die Szenarien werden so gestaltet, dass sie realistische Bedingungen für die Nutzung der Interaktionsschnittstellen simulieren. Anzahl und Art der interaktiven Elemente sind in beiden Szenarien gleich, um die Vergleichbarkeit zu gewährleisten. Die Platzierung der Elemente erfolgt in Abstimmung mit den 360°-Videos der Szene. **Anhang C** beinhaltet Abbildungen der erstellten Szenarien.

### **Fragestellungen und Datenerhebung**

Ziel der inhaltsbasierten Evaluation ist es, die folgenden Aspekte zu untersuchen:

#### *Ablenkung durch die Interaktionsschnittstellen*

- Werden die Schnittstellen als störend empfunden?
- Lenken die Schnittstellen die Aufmerksamkeit der Teilnehmenden vom Inhalt ab?

#### *User Experience*

- Wie wird die User Experience der Schnittstellen bewertet?
- In welchen Faktoren treten (deutliche) Unterschiede in der Bewertung zwischen den Schnittstellen auf?
- Decken sich die subjektiven Angaben hinsichtlich der Effizienz mit den gemessenen technischen Daten?

#### *Motion Sickness*

- Tritt Motion Sickness auf? Wenn ja, wie stark sind die Symptome ausgeprägt?
- Welche Symptome treten auf vermehrt auf?
- Gibt es Unterschiede zwischen den Schnittstellen in Bezug auf die Häufigkeit und Intensität der Symptome?
- Bestehen Korrelationen zwischen dem Auftreten von Motion Sickness Symptomen und der Bewertung der Usability/UX sowie der Interaktionsgeschwindigkeit?

Die Erhebung der Daten erfolgt über standardisierte Fragebögen. Zur Messung der UX wird der UEQ [Laugwitz et al., 2008] verwendet. Der SSQ [Kennedy et al., 1993] erfasst Symptome von Motion Sickness. Um Aussagen hinsichtlich der Ablenkung und Presence tätigen zu können, werden zusätzlich spezifische Fragen zur Wechselwirkung zwischen Interaktion und Inhalt gestellt. Diese werden in Anlehnung an Fragen aus dem Presence Questionnaire von Witmer and Singer [1998] formuliert. Die gestellten Fragen lauten dabei:

- Wie sehr warst Du in die Erfahrung der virtuellen Umgebung involviert?
- Wie gut konntest Du Dich auf die zugewiesenen Aufgaben oder erforderlichen Tätigkeiten konzentrieren und nicht auf die Mechanismen, die zur Ausführung dieser Aufgaben oder Tätigkeiten genutzt werden?
- Wie gut konntest Du Dich auf den Inhalt und die visuellen Darstellungen in der Szene konzentrieren?

Der Bewertung der Fragen erfolgte auf einer Skala von 0 (Gar nicht) bis 5 (Vollständig).

Zusätzlich wird auch für den inhaltsbasierten Abschnitt ein automatisierter Log erstellt. Auch hier werden, wie zuvor im technischen Abschnitt, Daten zur Interaktionsgeschwindigkeit erfasst. Auch Fehler werden weiterhin im Evaluationsprotokoll vermerkt. Dadurch werden zusätzliche Daten generiert, die insbesondere für eine genauere Untersuchung der Erlernbarkeit der Interaktionsschnittstellen herangezogen werden können.

### **Erwartete Ergebnisse der inhaltsbasierten Evaluation**

Im Rahmen der inhaltsbasierten Evaluation wird erwartet, dass das Item Scanning aufgrund der begrenzten Anzahl von Elementen in den jeweiligen Szenen eine höhere Interaktionsgeschwindigkeit ermöglicht und damit eine bessere Bewertung der Effizienz im UEQ erzielt. Hinsichtlich des Komforts, insbesondere in Bezug auf das Auftreten von Motion Sickness, werden keine deutlichen Unterschiede zwischen den Schnittstellen erwartet. Sollten Symptome von Motion Sickness auftreten, dann vermutlich im Zusammenhang mit kognitiver oder visueller Anstrengung. Insgesamt wird jedoch erwartet, dass keine stark ausgeprägten Symptome auftreten. Darüber hinaus sollten beide Schnittstellen nur eine minimale Ablenkung vom Inhalt der Szene zeigen, sodass die Wahrnehmung des Inhalts und die Interaktion mit der Szene nicht wesentlich beeinträchtigt werden.



## 5.2 Durchführung

Die Evaluation der entwickelten Interaktionsschnittstellen wurde im Zeitraum vom 11. bis zum 15. Dezember 2024 an der Technischen Hochschule Lübeck durchgeführt. Insgesamt nahmen 16 Personen (10 männlich, 5 weiblich, 1 nicht-binär) ohne motorische Beeinträchtigungen an der Evaluation teil. Obwohl die Teilnehmenden nicht aus der primären Zielgruppe rekrutiert wurden, liefert ihre Teilnahme wertvolle Informationen über die Funktionalität und Usability der Implementierungen. Die Auswahl einer heterogenen Gruppe ohne motorische Beeinträchtigungen ermöglicht eine Bewertung der allgemeinen Nutzbarkeit der Interaktionsschnittstellen. Darüber hinaus können potenzielle Schwachstellen in der Interaktion und technische Herausforderungen unabhängig von spezifischen Beeinträchtigungen identifiziert werden. Dies schafft eine solide Grundlage für weitere Optimierungen, die auf eine verbesserte allgemeine Usability der Schnittstellen abzielen. Dennoch bleibt eine zielgruppenspezifische Evaluation unverzichtbar, um sicherzustellen, dass die entwickelten Schnittstellen den spezifischen Anforderungen von Personen mit motorischen Beeinträchtigungen gerecht werden.

Jede Evaluationssitzung folgte einem standardisierten Ablauf und dauerte zwischen 45 und 60 Minuten pro Person. Der genaue Ablauf gliederte sich wie folgt:

### 1. Einführung:

Die Teilnehmenden wurden begrüßt und in das Projekt, das Ziel der Evaluation und den Ablauf eingeführt. Die VR-Brille wurde auf die Person eingestellt (Größe und Abstand der Linsen).

### 2. Abfrage des gesundheitlichen Zustands:

Vor Beginn der Evaluationsabschnitte wurden die Teilnehmenden befragt, ob sie Symptome wie z. B. Kopfschmerzen, Schwindel oder Übelkeit verspürten. Damit sollte sichergestellt werden, dass mögliche Symptome von Motion Sickness nicht bereits vor der Nutzung der Anwendung vorhanden waren und die Ergebnisse des Fragebogens zuverlässiger ausgewertet werden können.

### 3. Erklärung der ersten Interaktionsschnittstelle:

Die Funktionsweise der ersten zu testenden Schnittstelle wurde erläutert. Es wurde erklärt, wie das Scanning gestartet werden kann und wie Selektion und Navigation funktionieren.

#### 4. Durchführung der technischen Evaluation:

Nach der Erklärung des Scanning-Verfahrens folgte eine kurze Erläuterung der Aufgabe im technischen Szenario, die dann von den Teilnehmenden durchgeführt wurde. Anschließend nahmen die Teilnehmenden die VR-Brille ab und füllten den ersten Fragebogen (SUS) in digitaler Form aus.

#### 5. Durchführung der inhaltsbasierten Evaluation:

Nach dem Ausfüllen des Fragebogens wurde den Teilnehmenden die Aufgabe des inhaltsbasierten Szenarios erklärt und dieses durchlaufen. Anschließend wurden die restlichen Fragebögen (UEQ, SSQ) sowie die formulierten Fragen bzgl. der Ablenkung, ebenfalls in digitaler Form, ausgefüllt.

#### 6. Wechsel zur zweiten Interaktionsschnittstelle:

Die gleiche Prozedur (Schritte 2-5) wurde danach für die zweite Schnittstelle wiederholt.

#### 7. Abschließendes Feedback und Danksagung:

Am Ende der Evaluation hatten die Teilnehmenden die Möglichkeit, sowohl anonym als auch in einem persönlichen Gespräch Feedback zu geben. Abschließend wurde den Teilnehmenden für ihre Unterstützung und Zeit gedankt.

Die Evaluation wurde stationär an einem Tisch durchgeführt, an dem die Teilnehmenden saßen. Vor ihnen stand ein Laptop, auf dem die Anwendung direkt über den Unity-Editor gestartet wurde. Die VR-Brille (Meta Quest 3) war über *LinkCable* mit dem Laptop verbunden. Aufgrund der begrenzten Kabellänge wurden die Teilnehmenden angewiesen, die Bewegungen des Kopfes und des Oberkörpers innerhalb des Aktionsradius zu halten. Der Aufbau mit einem eingeschränkten Bewegungsradius wurde gewählt, um die Teilnehmenden zu einem Einsatz der implementierten Navigationsmethoden zu animieren. Moderate Kopfbewegungen waren jedoch ausdrücklich erlaubt, solange sie den vorgegebenen Rahmen nicht überschritten.

Um sicherzustellen, dass weder die Reihenfolge der getesteten Interaktionsschnittstellen noch die spezifischen Inhalte der inhaltsbasierten Szenarien die Ergebnisse der Evaluation beeinflussen, wurden die Teilnehmenden in vier Testgruppen eingeteilt. Jede Gruppe begann mit einer anderen Kombination aus Interaktionsschnittstelle und inhaltsbasiertem Szenario. Dadurch wurde eine gleichmäßige Verteilung der Bedingungen erreicht und mögliche Verzerrungen minimiert. [Tabelle 5.1](#) veranschaulicht die festgelegten Testgrup-

Bezeichnung	Erste Interaktionsschnittstelle	Erstes inhaltsbasiertes Szenario
ITH	Item Scanning	Szenario Code
ISB	Item Scanning	Szenario Schlüssel
CTH	Cartesian Scanning	Szenario Code
CSB	Cartesian Scanning	Szenario Schlüssel

Tabelle 5.1: Beschreibung der Testgruppen anhand der zuerst verwendeten Interaktionsschnittstelle und des zuerst durchgeführten inhaltsbasierten Szenarios

pen und zeigt, welche Gruppe mit welcher Schnittstelle und welchem inhaltsbasierten Szenario startete.

## 5.3 Ergebnisse

### 5.3.1 Usability (SUS) und User Experience (UEQ)

Die Ergebnisse der SUS zeigen deutliche Unterschiede zwischen den beiden Schnittstellen (vgl. [Abbildung 5.1](#)). Der Mittelwert des SUS-Scores für das Item Scanning beträgt 73,44 (SD = 16,28) bei einem Median von 73,75. Für das Cartesian Scanning liegt der Mittelwert hingegen bei 85,63 (SD = 11,96) mit einem Median von 90.

Die detaillierte Verteilung der Scores ist in [Abbildung 5.2](#) dargestellt. Für das Item Scanning variieren die SUS-Scores zwischen 47,5 und 97,5. Der größte Anteil (drei Personen) liegt im Bereich 52,5-57,5. Insgesamt bewerteten sieben von 16 Teilnehmenden (43,75%) die Schnittstelle mit einem Score von 80 oder höher. Für das Cartesian Scanning liegen die SUS-Scores zwischen 57,5 und 97,5. Der größte Anteil der Teilnehmenden (sechs Personen) bewertete die Schnittstelle mit einem Score im Bereich 92,5-97,5. Insgesamt haben 11 von 16 Teilnehmenden (68,75%) die Schnittstelle mit einem SUS-Score von mindestens 80 bewertet.

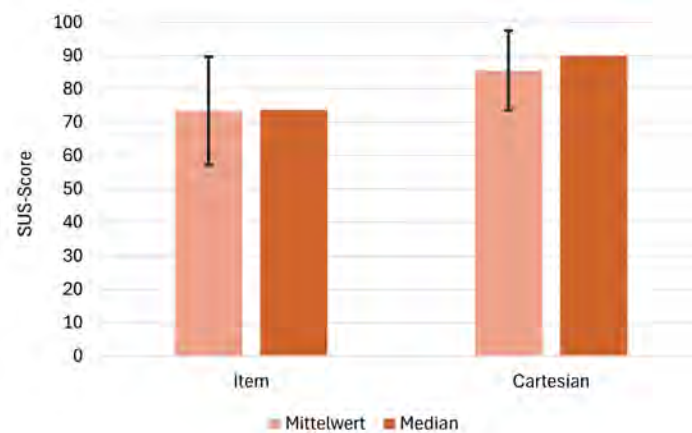
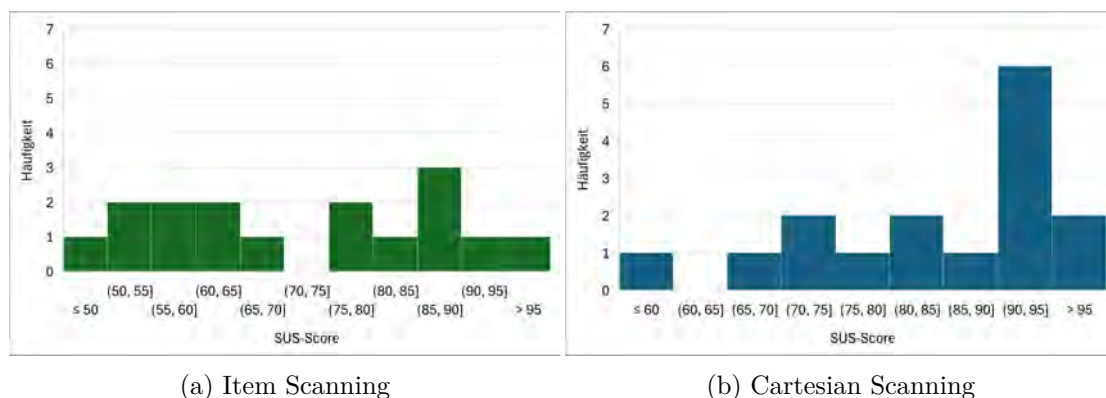


Abbildung 5.1: Mittelwerte (inklusive SD) und Median der SUS-Scores beider Interaktionsschnittstellen



(a) Item Scanning

(b) Cartesian Scanning

Abbildung 5.2: Verteilung der SUS-Scores der beiden Interaktionsschnittstellen. Der SUS-Score liegt im Wertebereich zwischen 0 und 100

Die Ergebnisse des UEQ wurden mit Hilfe des Auswertungstools der Autoren analysiert. Für jeden der sechs Faktoren wurden Mittelwerte und Varianzen berechnet. Die [Tabelle 5.2](#) zeigt die berechneten Mittelwerte und Varianzen für das Item Scanning. In [Abbildung 5.3](#) werden diese Ergebnisse zusätzlich visualisiert.

Für das Item Scanning liegen die Mittelwerte der Bewertungen im positiven Bereich. Den höchsten Wert weist der Faktor Originalität mit einem Mittelwert (MW) von 1,875 bei einer Varianz von 0,69 auf, gefolgt von Durchschaubarkeit (MW = 1,672, Varianz = 0,87) und Stimulation (MW = 1,516, Varianz = 2,50). Am niedrigsten wurden die

UEQ Faktoren (Mittelwert und Varianz)		
<b>Attraktivität</b>	↑ 1,396	2,00
<b>Durchschaubarkeit</b>	↑ 1,672	0,87
<b>Effizienz</b>	↑ 0,844	1,55
<b>Steuerbarkeit</b>	↑ 0,984	1,35
<b>Stimulation</b>	↑ 1,516	2,50
<b>Originalität</b>	↑ 1,875	0,69

Tabelle 5.2: UEQ-Ergebnisse des Item Scannings: Mittelwerte und Varianzen der Faktoren

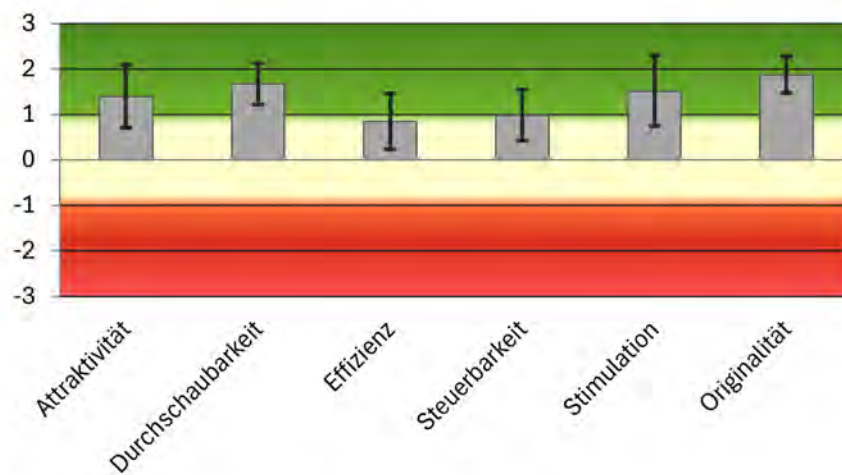


Abbildung 5.3: Visualisierung der UEQ-Ergebnisse des Item Scannings

Faktoren Steuerbarkeit und Effizienz bewertet. Hier liegen die Mittelwerte bei 0,984 und 0,844 mit Varianzen von 1,35 und 1,55.

Die Ergebnisse für das Cartesian Scanning sind in **Tabelle 5.3** dargestellt und in **Abbildung 5.4** visualisiert.

Das Cartesian Scanning wurde insgesamt besser bewertet als das Item Scanning. Der Faktor Durchschaubarkeit wurde mit einem Mittelwert von 2,234 und einer Varianz von 0,53 am höchsten bewertet. Die Faktoren Attraktivität, Stimulation und Originalität wurden mit Mittelwerten von 1,833, 1,734 und 1,750 ähnlich hoch bewertet. Den niedrigsten Mittelwert und gleichzeitig die höchste Varianz weist der Faktor Effizienz auf. Hier liegt der Mittelwert bei 1,063 bei einer Varianz von 1,33.

In **Anhang D** sind ergänzend die detaillierten Verteilungen der Bewertungen der einzelnen Items des UEQ für beide Schnittenstellen aufgeführt.







UEQ Faktoren (Mittelwert und Varianz)		
<b>Attraktivität</b>	 1,833	0,64
<b>Durchschaubarkeit</b>	 2,234	0,53
<b>Effizienz</b>	 1,063	1,33
<b>Steuerbarkeit</b>	 1,469	0,77
<b>Stimulation</b>	 1,734	1,26
<b>Originalität</b>	 1,750	1,12

Tabelle 5.3: UEQ-Ergebnisse des Cartesian Scannings: Mittelwerte und Varianzen der Faktoren

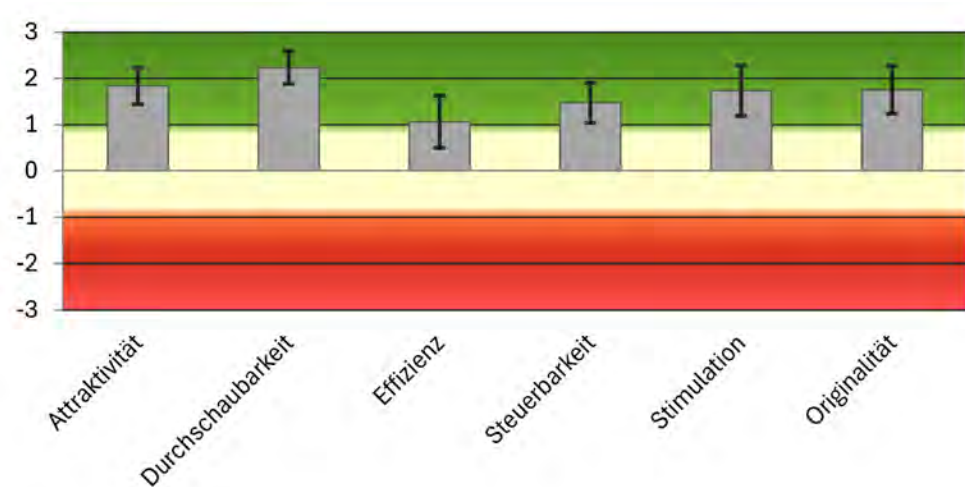


Abbildung 5.4: Visualisierung der UEQ-Ergebnisse des Cartesian Scannings

### 5.3.2 Motion Sickness (SSQ)

Die Ergebnisse des SSQ werden in [Tabelle 5.4](#) dargestellt. Die Tabelle zeigt die im Fragebogen erfassten Symptome sowie die jeweilige Symptomkategorie. Für beide Schnittstellen wird für jedes Symptom die Häufigkeit angegeben, mit der es auftrat. Dies entspricht der Anzahl der Personen, die einen Wert  $>0$  angaben. Zusätzlich wird in einer weiteren Spalte dargestellt, wie viele dieser Personen in der Auswertung jeweils einen Wert  $>1$  angegeben haben und somit eine moderate bis starke Ausprägung des Symptoms empfunden haben. Es zeigt sich, dass beim Item Scanning die Symptome angestrengte Augen (elf Personen, davon vier mit moderater Ausprägung) sowie Ermüdung, Kopfdruck und verschwommenes Sehen (sechs Personen, keine moderaten Ausprägungen) am häufigsten auftraten. Vereinzelt wurde von allgemeinem Unwohlsein, Konzentrationschwierigkeiten, Schwitzen, erhöhtem Speichelfluss, Schwindel und Übelkeit berichtet, wobei diese Symptome jeweils nur eine oder zwei Personen betrafen und nur in leichter Ausprägung auftraten. Symptome wie Aufstoßen oder Gleichgewichtsstörungen wurden nicht genannt. Beim Cartesian Scanning ergab sich eine ähnliche Verteilung. Am häufigsten wurden angestrengte Augen (elf Personen, davon zwei mit moderater Ausprägung) und Ermüdung (acht Personen, keine moderaten Ausprägungen) berichtet. Im Vergleich zum Item Scanning traten verschwommenes Sehen und Kopfdruck etwas seltener auf (drei bzw. vier Personen). Weitere Symptome wie Konzentrationsschwierigkeiten oder Schwitzen traten ebenfalls nur vereinzelt auf, während allgemeines Unwohlsein, Übelkeit und Gleichgewichtsstörungen nicht aufgetreten sind.

Kategorie	Symptom	Item Scanning		Cartesian Scanning	
		Häufigkeit	Davon Bewertungen >1	Häufigkeit	Davon Bewertungen >1
N, O	Allgemeines Unwohlsein	2	1	0	0
O	Ermüdung	6	0	8	0
O	Kopfschmerzen	3	0	3	0
O	Angestrenzte Augen	11	4	11	2
O, D	Verschwommenes Sehen	6	0	3	0
D	Kopfdruck	6	0	4	0
O, D	Schwierigkeiten scharf zu sehen	4	0	4	1
N, O	Konzentrationsschwierigkeiten	1	0	2	0
N	Schwitzen	2	0	2	0
N	Erhöhter Speichelfluss	1	0	1	0
N	Aufstoßen	0	0	0	0
N	Magen macht sich bemerkbar	1	0	0	0
N, D	Übelkeit	1	0	0	0
D	Schwindel (offene Augen)	1	0	1	0
D	Schwindel (geschlossene Augen)	1	0	1	0
D	Gleichgewichtsstörung	0	0	0	0

Tabelle 5.4: Häufigkeit der SSQ-Symptome beider Interaktionsschnittstellen. Die Symptome sind den Kategorien N: Nausea, O: Oculomotor und D: Disorientation zugeteilt

Die weitere Auswertung des SSQ erfolgte nach der Beschreibung der Autoren [Kennedy et al., 1993]. Dazu wurden zunächst die gewichteten Scores der einzelnen Kategorien und anschließend der Gesamtscore bestimmt. Die Ergebnisse sind in [Tabelle 5.5](#) dargestellt. Die Verteilung der Gesamtscores ist darüber hinaus in [Abbildung 5.5](#) visualisiert.

	Item Scanning				Cartesian Scanning			
	Nausea	Oculomotor	Disorientation	Gesamt	Nausea	Oculomotor	Disorientation	Gesamt
Mittelwert	5,366	18,476	16,530	12,155	2,981	16,108	12,180	10,051
Median	0,000	18,950	6,960	11,220	0,000	15,160	0,000	7,480
SD	6,940	15,892	21,092	10,174	5,744	14,873	19,604	9,916

Tabelle 5.5: Mittelwerte, Median und SD der SSQ-Ergebnisse nach Interaktionsschnittstelle und Kategorie

Die Histogramme zeigen, dass die meisten Teilnehmenden bei beiden Schnittstellen einen SSQ-Wert zwischen 0 und 5 aufwiesen (jeweils sechs Personen). Beim Cartesian Scanning liegt der höchste Score bei 37,40 und damit deutlich höher als beim Item Scanning (29,92). Allerdings erreichten beim Item Scanning insgesamt mehr Personen einen Score größer 10 (acht Personen gegenüber sechs Personen). Der Mittelwert des Gesamtscores liegt beim Item Scanning bei 12,155 (SD = 10,174) und beim Cartesian Scanning bei 10,051 (SD = 9,916) (vgl. [Tabelle 5.5](#)).



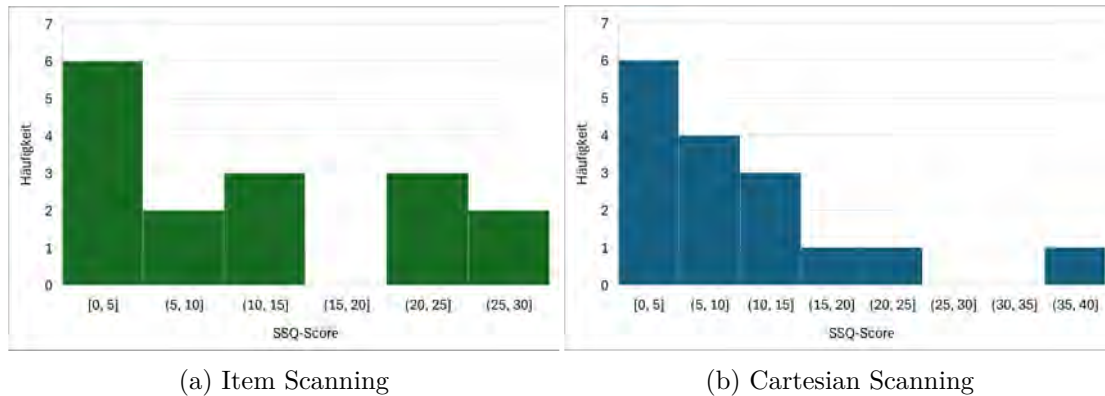


Abbildung 5.5: Verteilung der SSQ-Scores beider Interaktionsschnittstellen

Um genauere Aussagen darüber treffen zu können, in welchen Kategorien die meisten Symptome auftraten, werden die Ergebnisse der Scores der Kategorien folgend genauer betrachtet.

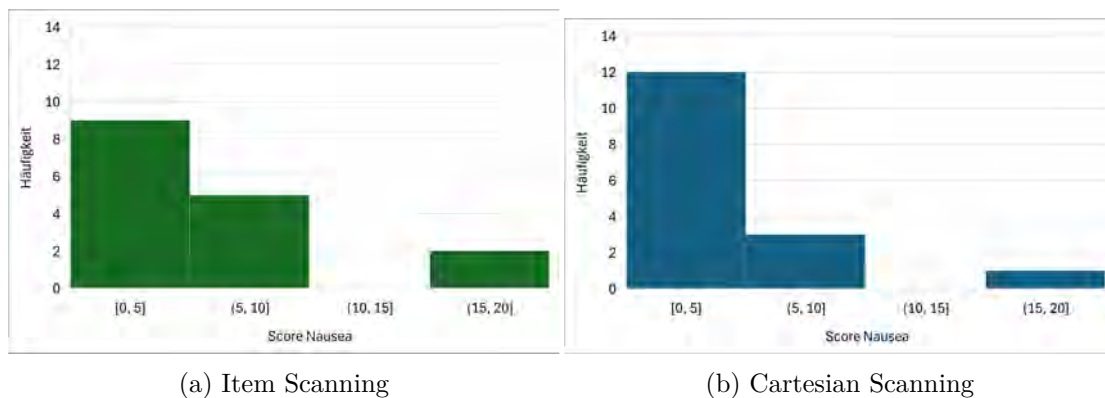


Abbildung 5.6: Verteilung der Scores der SSQ Nausea Scale beider Interaktionsschnittstellen

In der Kategorie Nausea (vgl. [Abbildung 5.6](#)) berichteten beim Item Scanning neun Personen keine Symptome. Fünf Teilnehmende zeigten Scores zwischen 8 und 10, was dem Auftreten eines einzigen Symptoms in leichter Ausprägung entspricht. Zwei Personen erreichten Scores über 15, was dem Auftreten mehrerer Symptome oder mindestens einem Symptom in moderater bis starker Ausprägung entspricht. Der Mittelwert der Scores dieser Kategorie liegt bei 5,366 ( $SD = 6,940$ ) (vgl. [Tabelle 5.5](#)). Beim Cartesian Scanning hingegen zeigten zwölf Personen keine Symptome. Drei Personen erreichten einen Score zwischen 5 und 10 und eine Person einen Score über 15. Der Mittelwert für das

Cartesian Scanning in dieser Kategorie liegt bei 2,981 (SD = 5,744) (vgl. [Tabelle 5.5](#)).

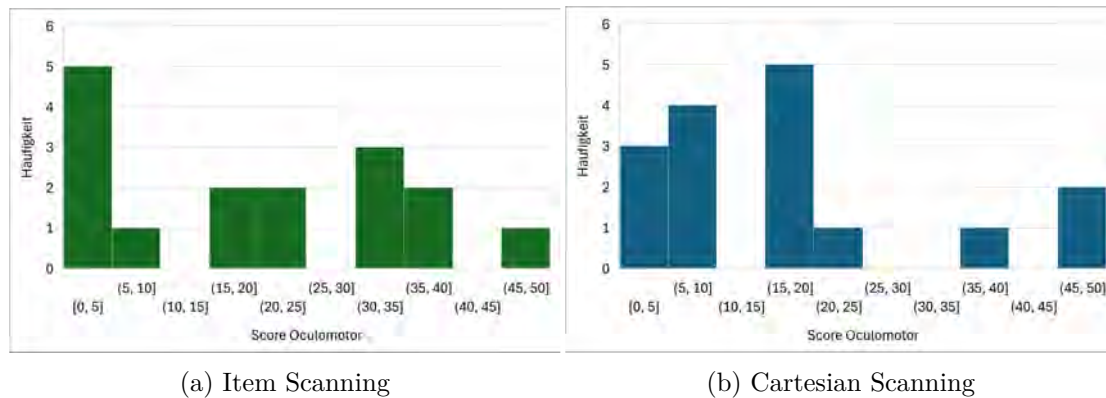


Abbildung 5.7: Verteilung der Scores der SSQ Oculomotor Scale beider Interaktions-schnittstellen

In der Kategorie Oculomotor (vgl. [Abbildung 5.7](#)) traten häufiger Symptome auf. Während beim Item Scanning sechs Personen keine bis nur sehr leichte Symptome (Score <10) aufwiesen, erreichten sieben Teilnehmende einen Score zwischen 15 und 35. Dies entspricht dem Auftreten von zwei oder mehr leichten oder mindestens einem moderaten Symptom. Drei Teilnehmende erreichten einen Score über 35, was auf das Auftreten mehrerer Symptome und mindestens einem Symptom in moderater Ausprägung hinweist. Der Mittelwert der Scores dieser Kategorie liegt bei 18,476 (SD = 15,892) (vgl. [Tabelle 5.5](#)). Beim Cartesian Scanning zeigten sieben Personen keine bis leichte Symptome. Fünf Personen wiesen einen Score zwischen 15 und 20 auf. Vier Personen erreichten einen Score größer 20, wobei zwei Personen einen Score von über 45 erreichten. Der Mittelwert der Scores dieser Kategorie für das Cartesian Scanning liegt bei 16,108 (SD = 14,873) (vgl. [Tabelle 5.5](#)).

In der Kategorie Desorientierung (vgl. [Abbildung 5.8](#)) zeigten beim Item Scanning acht Personen keine Symptome. Zwei Teilnehmende erreichten einen Score zwischen 10 und 15. Sechs Personen erreichten einen Score über 25, wobei eine Person sogar einen Score größer 60 erreichte. Der Mittelwert der Scores dieser Kategorie für das Item Scanning liegt bei 16,530 (SD = 21,092) (vgl. [Tabelle 5.5](#)). Beim Cartesian Scanning blieben neun Personen symptomfrei, während vier Personen einen Score zwischen 10 und 15 erreichten. Nur drei Personen erreichten einen Score über 25, wobei eine Person sogar einen Score von 65 überschritt. Der Mittelwert der Scores dieser Kategorie liegt bei 12,180 (SD = 19,604) (vgl. [Tabelle 5.5](#)).

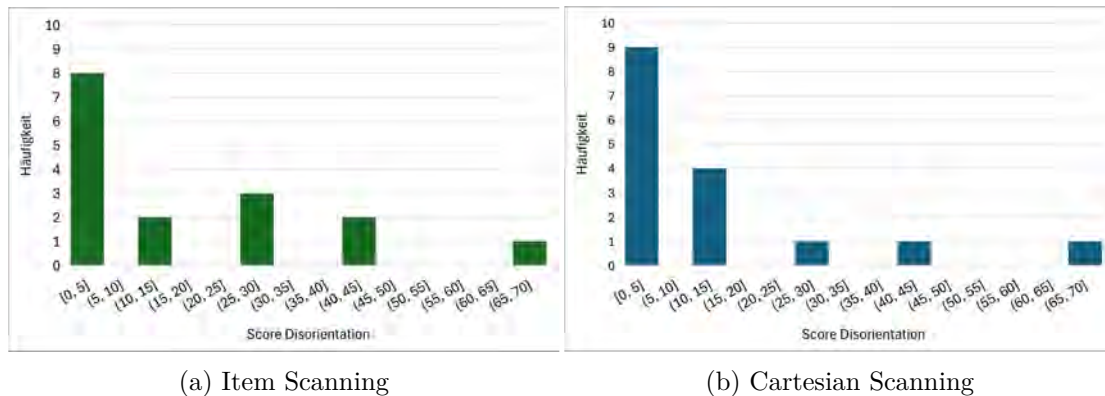


Abbildung 5.8: Verteilung der Scores der SSQ Disorientation Scale beider Interaktionschnittstellen

### 5.3.3 Ablenkung durch die Interaktionsschnittstellen

Die Ergebnisse zur Ablenkung bzw. Presence wurden anhand von drei Fragen erhoben, deren Mittelwerte für beide Schnittstellen in der [Abbildung 5.9](#) dargestellt sind. Der Bewertung der Fragen erfolgte auf einer Skala zwischen 0 und 5.

*Frage 1:* Wie sehr warst Du in die Erfahrung der virtuellen Umgebung involviert? Die Bewertungen für das Item Scanning zeigen einen Mittelwert von 3,813 ( $SD = 1,047$ ). Für das Cartesian Scanning liegt der Mittelwert bei 4,375 ( $SD = 0,719$ ).

*Frage 2:* Wie gut konntest Du Dich auf die zugewiesenen Aufgaben oder erforderlichen Tätigkeiten konzentrieren und nicht auf die Mechanismen, die zur Ausführung dieser Aufgaben oder Tätigkeiten genutzt werden? Bei dieser Frage erreicht das Item Scanning einen Mittelwert von 4,000 ( $SD = 0,817$ ), während das Cartesian Scanning einen Mittelwert von 4,125 ( $SD = 1,025$ ) aufweist.

*Frage 3:* Wie gut konntest Du Dich auf den Inhalt und die visuellen Darstellungen in der Szene konzentrieren? Die Ergebnisse für das Item Scanning zeigen einen Mittelwert von 4,250 ( $SD = 0,577$ ). Beim Cartesian Scanning wird ein Mittelwert von 4,500 ( $SD = 0,730$ ) erzielt.

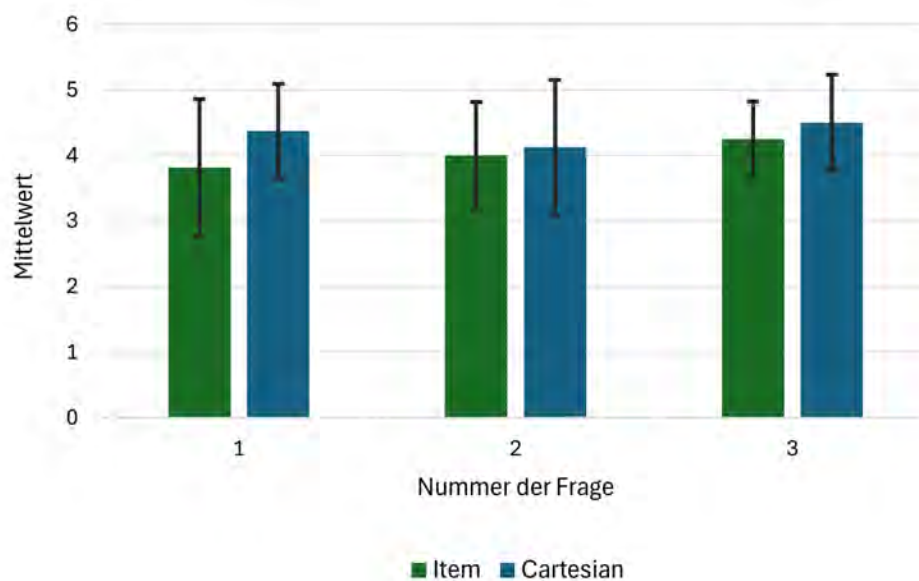


Abbildung 5.9: Ergebnisse (MW und SD) der Fragen zur Ablenkung durch die Interaktionsschnittstellen. Die Bewertung erfolgte auf einer Skala zwischen 0 und 5

### 5.3.4 Geschwindigkeit und Effizienz

Die durchschnittlichen Zeiten für die Durchführung der Evaluationsabschnitte zeigen Unterschiede zwischen den beiden Schnittstellen, dargestellt in [Abbildung 5.10](#). Beim Item Scanning benötigten die Teilnehmenden im technischen Abschnitt durchschnittlich 6 Minuten und 42 Sekunden (SD = 1 Minute und 26 Sekunden). Der schnellste Durchlauf dauerte 4 Minuten und 45 Sekunden, während der längste Durchlauf 8 Minuten und 56 Sekunden dauerte. Im inhaltsbasierten Abschnitt betrug die benötigte Zeit durchschnittlich 5 Minuten und 35 Sekunden (SD = 1 Minute und 43 Sekunden). Der schnellste Durchlauf dauerte hier 2 Minuten und 39 Sekunden, während der längste Durchlauf 9 Minuten und 17 Sekunden dauerte. Beim Cartesian Scanning hingegen betrug die für den technischen Abschnitt benötigte Zeit durchschnittlich 8 Minuten und 5 Sekunden (SD = 1 Minute und 41 Sekunden). Der schnellste Durchlauf dauerte 5 Minuten und 52 Sekunden, während der längste Durchlauf 11 Minuten und 04 Sekunden dauerte. Für den inhaltsbasierten Abschnitt wurden durchschnittlich 8 Minuten 50 Sekunden benötigt (SD = 2 Minuten 52 Sekunden). Der schnellste Durchlauf dauerte 4 Minuten und 26 Sekunden, während der längste Durchlauf 15 Minuten und 42 Sekunden dauerte.

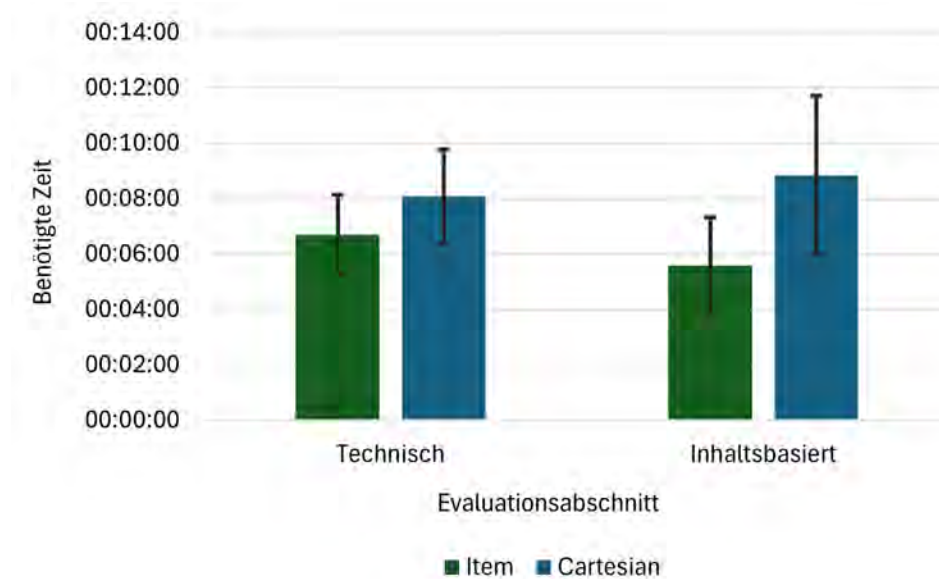


Abbildung 5.10: Benötigte Zeit zum Durchlaufen der Evaluationsabschnitte

Für die Auswertung der Geschwindigkeiten wurden die Interaktionsgeschwindigkeiten aus den Log-Files pro Person zusammengefasst. Daraus wurden Mittelwerte, Varianzen und Spannweiten berechnet, sowohl getrennt für jeden Evaluationsabschnitt als auch zusammengefasst für beide Abschnitte. Die Ergebnisse sind in der [Tabelle 5.6](#) aufgezeigt sowie in der [Abbildung 5.11](#) visualisiert.

Beim Item Scanning beträgt die durchschnittliche Interaktionsgeschwindigkeit im technischen Abschnitt 4,617s (Varianz = 0,381; Spannweite = 2,064). Dagegen ist die Interaktionsgeschwindigkeit im inhaltsbasierten Abschnitt mit 2,927s kürzer (Varianz = 0,522; Spannweite = 2,274). Über beide Abschnitte hinweg ergibt sich eine durchschnittliche Interaktionsgeschwindigkeit von 3,862s (Varianz = 0,293; Spannweite = 1,917). Beim Cartesian Scanning werden im technischen Abschnitt durchschnittlich 8,733s (Varianz = 1,733; Spannweite = 4,840) zur Interaktion benötigt. Im inhaltsbasierten Abschnitt beträgt die mittlere Interaktionsgeschwindigkeit 7,404s (Varianz = 5,616; Spannweite = 7,340). Über beide Abschnitte hinweg ergibt sich eine Interaktionsgeschwindigkeit von 8,004s (Varianz = 3,383; Spannweite = 5,603).

	Item Scanning			Cartesian Scanning		
	Mittelwert	Varianz	Spannweite	Mittelwert	Varianz	Spannweite
Technisch	4,617	0,381	2,064	8,733	1,733	4,840
Inhaltsbasiert	2,927	0,522	2,274	7,404	5,616	7,340
Gesamt	3,862	0,293	1,917	8,004	3,383	5,603

Tabelle 5.6: Mittelwerte, Varianzen und Spannweiten der Interaktionsgeschwindigkeiten beider Interaktionsschnittstellen nach Evaluationsabschnitt in Sekunden

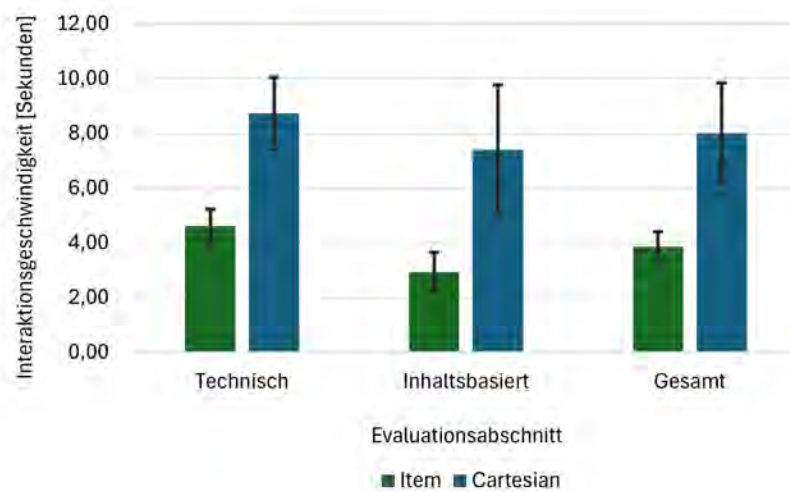


Abbildung 5.11: Darstellung der mittleren Interaktionsgeschwindigkeiten beider Interaktionsschnittstellen nach Evaluationsabschnitt

Um die Veränderung der Interaktionsgeschwindigkeit über die Zeit bzw. über die Anzahl der durchgeführten Interaktionen zu untersuchen, wurde eine lineare Regressionsanalyse durchgeführt und anschließend die Korrelation zwischen den beiden Variablen berechnet. Die Ergebnisse werden in **Tabelle 5.7** aufgezeigt. Der Regressionskoeffizient  $b$  beschreibt die durchschnittliche Veränderung der Interaktionszeit in Abhängigkeit von der Anzahl der bereits durchgeführten Interaktionen. Der Korrelationskoeffizient  $r$  gibt an, wie eng der Zusammenhang zwischen den Variablen ist. Beim Item Scanning ergibt sich für den technischen Abschnitt ein mittlerer Regressionskoeffizient von  $b = -0,839$  (Varianz = 0,503) bei einem Korrelationskoeffizienten von  $r = -0,219$ . Im inhaltsbasierten Abschnitt beträgt der mittlere Regressionskoeffizient  $b = -0,391$  (Varianz = 0,754) bei einem Korrelationskoeffizienten von  $r = -0,066$ . Über beide Abschnitte hinweg ergibt sich ein mittlerer Regressionskoeffizient von  $b = -2,539$  (Varianz = 1,105) und ein Korrelationskoeffizient von  $r = -0,263$ . Beim Cartesian Scanning ergibt sich für den technischen Abschnitt ein durchschnittlicher Regressionskoeffizient von  $b = 0,180$  (Varianz = 0,326)

mit einem Korrelationskoeffizienten von  $r = 0,038$ . Im inhaltsbasierten Abschnitt beträgt der Regressionskoeffizient  $b = 0,241$  (Varianz = 1,177) und der Korrelationskoeffizient  $r = -0,019$ . Über beide Abschnitte hinweg beträgt der Regressionskoeffizient  $b = -0,492$  (Varianz = 1,461) und der Korrelationskoeffizient  $r = -0,085$ .

	Item Scanning			Cartesian Scanning		
	b		r	b		r
	Mittelwert	Varianz		Mittelwert	Varianz	
Technisch	-0,839	0,503	-0,219	0,180	0,326	0,038
Inhaltsbasiert	-0,391	0,754	-0,066	0,241	1,177	-0,019
Gesamt	-2,539	1,105	-0,263	-0,492	1,461	-0,085

Zusammenhang: Keiner Schwach

Tabelle 5.7: Ergebnisse der linearen Regressionsanalyse (Mittelwert und Varianz der Regressionskoeffizienten  $b$  und Korrelationskoeffizienten  $r$ )

Neben der Interaktionsgeschwindigkeit wurde auch die Anzahl der benötigten Scanning Zyklen pro Interaktion im Log-File erfasst. Die [Tabelle 5.8](#) zeigt die Ergebnisse bezüglich der Zyklen für das Item Scanning. Im technischen Abschnitt wurden durchschnittlich 1,542 Zyklen pro Eingabe benötigt (Varianz = 0,007; Spannweite = 0,332). Im inhaltsbasierten Abschnitt hingegen lag der Mittelwert bei 1,423 (Varianz = 0,021; Spannweite = 0,481).

	Zyklen pro getätigter Eingabe		
	Mittelwert	Varianz	Spannweite
Technisch	1,542	0,007	0,332
Inhaltsbasiert	1,423	0,021	0,481

Tabelle 5.8: Benötigte Scanning-Zyklen pro Interaktion beim Item Scanning

### 5.3.5 Robustheit

In beiden Evaluationsabschnitten wurden die aufgetretenen Fehler im Evaluationsprotokoll erfasst. Als Fehler werden in diesem Zusammenhang Eingaben verstanden, die nicht zum gewünschten Ergebnis geführt haben.

Die Ergebnisse des Item Scannings sind in [Abbildung 5.12](#) dargestellt. Im technischen



Abschnitt (vgl. [Abbildung 5.12a](#)) traten bei der Mehrheit der Teilnehmenden Fehler auf. Nur drei Personen konnten den Abschnitt ohne Fehler abschließen. Sieben Personen machten einen Fehler, zwei Personen machten jeweils drei Fehler. Eine Person hat fünf Fehler gemacht. Somit wurden in diesem Abschnitt über alle Teilnehmenden hinweg insgesamt 18 Fehler erfasst. Im inhaltsbasierten Abschnitt (vgl. [Abbildung 5.12b](#)) haben sechs Personen keinen Fehler gemacht, während weitere sechs Personen jeweils einen Fehler machten. Vier Personen machten zwei Fehler. Keine teilnehmende Person hat in diesem Abschnitt mehr als zwei Fehler gemacht. Insgesamt wurden in diesem Abschnitt 14 Fehler verzeichnet.

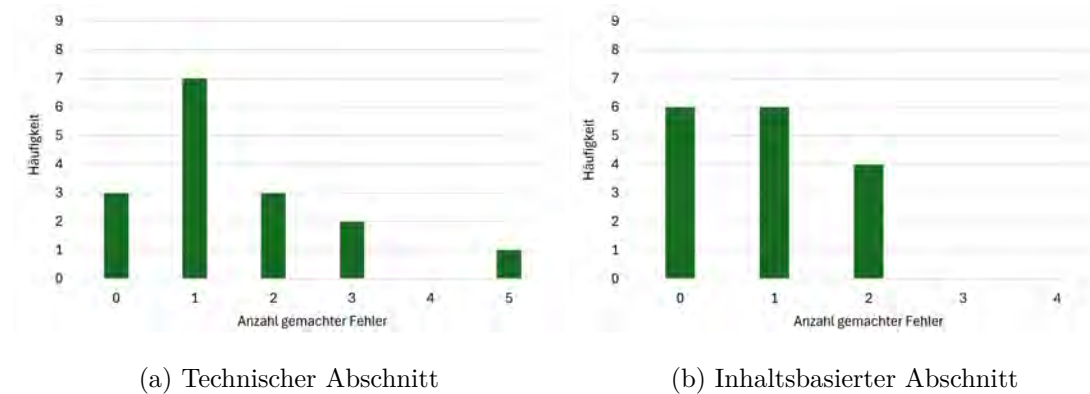


Abbildung 5.12: Häufigkeiten der Anzahl gemachter Fehler im Item Scanning

Im Evaluationsprotokoll wurde für jeden aufgetretenen Fehler der jeweilige Grund vermerkt. Die beim Item Scanning aufgetretenen Gründe sind in [Abbildung 5.13](#) aufgeführt. Es ist zu erkennen, dass die meisten Fehler auf Timing-Probleme zurückzuführen sind. Das bedeutet, dass Eingaben etwas zu früh oder etwas zu spät erfolgten und dadurch versehentlich das vorherige oder nachfolgende Item in der Scanning-Reihenfolge anstelle des Zielelements ausgewählt wurde. Zu späte Eingaben traten insbesondere bei der Navigation auf. Derartige Fehler traten bei 13 der 16 Teilnehmenden auf. Im technischen Abschnitt der Evaluation traten zudem Fehler auf, die auf Verständnisprobleme zurückzuführen sind. Vier Teilnehmenden war das Prinzip bzw. die Aufgabe zu Beginn nicht ganz klar, wodurch es zu fehlerhaften Eingaben kam.



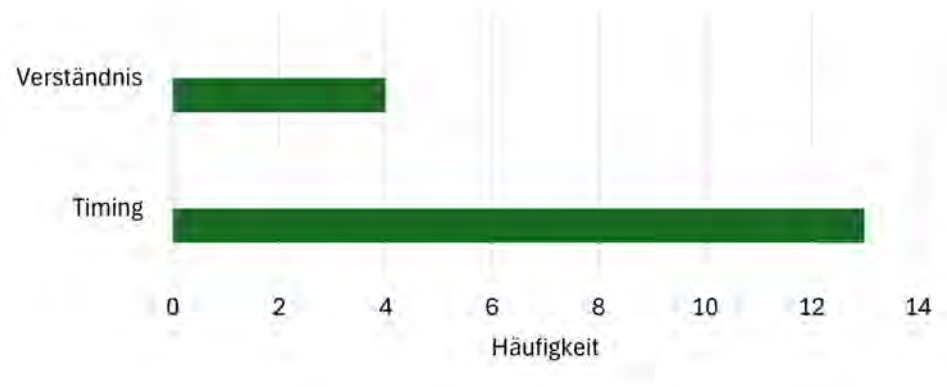


Abbildung 5.13: Gründe und Häufigkeit von Fehlern beim Item Scanning

Die Ergebnisse des Cartesian Scanning sind in [Abbildung 5.14](#) dargestellt. Den technischen Abschnitt (vgl. [Abbildung 5.14a](#)) konnten sieben Personen fehlerfrei abschließen. Drei Personen machten jeweils einen oder zwei Fehler, während jeweils eine Person drei, vier oder sechs Fehler machte. Insgesamt wurden in diesem Abschnitt über alle Teilnehmenden hinweg 22 Fehler erfasst. Im inhaltsbasierten Abschnitt (vgl. [Abbildung 5.14b](#)) blieb die Hälfte der Teilnehmenden fehlerfrei. Fünf Personen machten jeweils einen Fehler, eine Person drei, eine Person vier und eine Person acht Fehler. Die Gesamtzahl der Fehler beträgt hier somit 20.

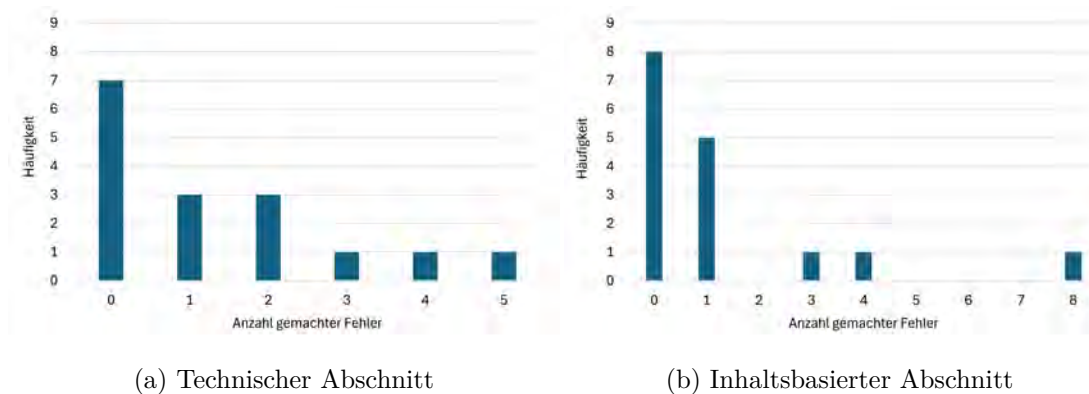


Abbildung 5.14: Häufigkeiten der Anzahl gemachter Fehler beim Cartesian Scanning

Die Gründe für Fehler beim Cartesian Scanning sind in [Abbildung 5.15](#) dargestellt. Es ist zu erkennen, dass neun Personen versehentlich das HUD anstelle des beabsichtigten Elementes ausgewählt haben. Fünf Personen vergaßen den Moduswechsel, was dazu führte, dass statt einer Selektion weitere Navigationsschritte durchgeführt wurden. Drei Personen hatten Probleme beim Schließen von Bildern, da sie den Schließen-Button nicht

wählen konnten. Eine Person hielt den Schalter für den Moduswechsel nicht lange genug gedrückt, sodass der Wechsel nicht ausgelöst wurde und weitere Navigationsschritte statt der gewünschten Selektion ausgeführt wurden. Eine Person selektierte versehentlich einen Wegpunkt, der hinter einem geöffneten Bild lag, obwohl eine leere Eingabe beabsichtigt war.



Abbildung 5.15: Gründe und Häufigkeit von Fehlern beim Cartesian Scanning

Beim Cartesian Scanning wurde zusätzlich die Häufigkeit von leeren Eingaben zur Korrektur von Fehlern, wie z. B. das Setzen der ersten Scanlinie an der falschen Position oder das Vergessen des Wechsels in den Navigationsmodus, erfasst. Die Ergebnisse sind in [Tabelle 5.9](#) dargestellt. Im technischen Abschnitt beträgt die durchschnittliche Anzahl leerer Eingaben 1.000 (Varianz = 2.933; Spannweite = 7.000). Im inhaltsbasierten Abschnitt liegt der Mittelwert mit 1.750 (Varianz = 6.200; Spannweite = 10.000) etwas höher.

	Anzahl leerer Eingaben		
	Mittelwert	Varianz	Spannweite
<b>Technisch</b>	1,000	2,933	7,000
<b>Inhaltsbasiert</b>	1,750	6,200	10,000

Tabelle 5.9: Leere Eingaben beim Cartesian Scanning nach Evaluationsabschnitt

### 5.3.6 Korrelationen und Zusammenhänge

Neben der getrennten Auswertung der Ergebnisse der einzelnen Variablen und Faktoren wurden Korrelationsanalysen durchgeführt, um Zusammenhänge zwischen den Variablen zu untersuchen.

Zunächst wurde für das Cartesian Scanning untersucht, ob die Position der ausgewählten Elemente im Sichtfeld (x- und y-Koordinaten) einen signifikanten Einfluss auf die Interaktionsgeschwindigkeit hat. Die Ergebnisse sind in [Tabelle 5.10](#) zusammengefasst und in [Abbildung 5.16](#) visualisiert.

	Korrelationskoeffizienten	
	X-Koordinate, Geschwindigkeit	Y-Koordinate, Geschwindigkeit
Technischer Abschnitt	0,739	-0,824
Inhaltsbasierter Abschnitt	0,616	-0,685
Gesamt	0,680	-0,756

Tabelle 5.10: Zusammenhang zwischen der Position des Elements im Sichtfeld und der Selektionsgeschwindigkeit beim Cartesian Scanning

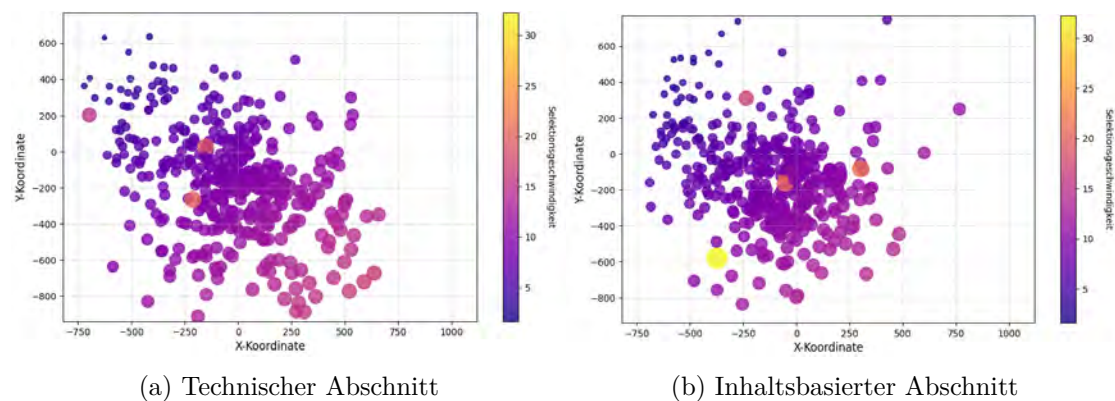


Abbildung 5.16: Zusammenhang zwischen der Position des Elements im Sichtfeld und der Interaktionsgeschwindigkeit beim Cartesian Scanning

Für den technischen Abschnitt zeigt sich eine positive Korrelation zwischen der horizontalen Position (x-Koordinate) und der Interaktionsgeschwindigkeit ( $r = 0,739$ ), während die vertikale Position (y-Koordinate) negativ mit der Geschwindigkeit korreliert ( $r = -0,824$ ). Ähnliche Tendenzen lassen sich im inhaltsbasierten Abschnitt beobachten, wo die Korrelationskoeffizienten für die x-Koordinate  $r = 0,616$  und für die y-Koordinate  $r =$

-0,685 betragen. Werden beide Abschnitte zusammen betrachtet, ergeben sich Korrelationskoeffizienten von  $r = 0,680$  für die x-Koordinate und  $r = -0,756$  für die y-Koordinate. Alle Korrelationen zeigen dabei einen p-Wert  $< 0,000$ . **Abbildung 5.16** veranschaulicht diese Zusammenhänge. Punkte, die eine schnelle Selektion repräsentieren, befinden sich oben links im Sichtfeld und sind klein und dunkelblau dargestellt. Mit zunehmender Entfernung nach rechts oder unten im Sichtfeld werden die Punkte größer und heller. Darüber hinaus ist in dieser Abbildung zu erkennen, dass die meisten Selektionen in der Mitte des Sichtfeldes durchgeführt wurden.

Anschließend wurde für beide Schnittstellen untersucht, ob es Korrelationen zwischen den Ergebnissen des SSQ und den Bewertungen der SUS, des UEQ und der Interaktionsgeschwindigkeit gibt. Die Ergebnisse dieser Untersuchungen sind in **Tabelle 5.11** dargestellt.

	SUS und SSQ			UEQ Attraktivität und SSQ		
	r	t-Wert	p-Wert	r	t-Wert	p-Wert
Item	0,17	0,65	0,53	0,24	0,94	0,36
Cartesian	-0,33	-1,31	0,21	-0,11	-0,41	0,69

	UEQ Durchschaubarkeit und SSQ			Eingabegeschwindigkeit und SSQ		
	r	t-Wert	p-Wert	r	t-Wert	p-Wert
Item	-0,20	-0,77	0,45	0,32	1,26	0,23
Cartesian	-0,66	-3,28	0,01	0,54	2,39	0,03

Zusammenhang:   Schwach   Mäßig   Stark   Signifikant

Tabelle 5.11: Zusammenhänge zwischen den Ergebnissen des SSQ und des SUS, des UEQ sowie der Interaktionsgeschwindigkeit

Für die Korrelation zwischen SSQ und SUS ergab sich beim Item Scanning ein Korrelationskoeffizient von  $r = 0,17$  (p-Wert = 0,53), während beim Cartesian Scanning ein Korrelationskoeffizient von  $r = -0,33$  (p-Wert = 0,21) festgestellt wurde.

Beim Vergleich des SSQ mit dem UEQ-Faktor Attraktivität ergab sich für das Item Scanning ein Korrelationskoeffizient von  $r = 0,24$  (p-Wert = 0,36) und für das Cartesian Scanning ein Wert von  $r = -0,11$  (p-Wert = 0,69). Für den UEQ-Faktor Durchschaubarkeit wurden Korrelationskoeffizienten von  $r = -0,20$  (p-Wert = 0,45) für das Item Scanning und  $r = -0,66$  (p-Wert = 0,01) für das Cartesian Scanning berechnet. Der

Zusammenhang zwischen dem SSQ und der Interaktionsgeschwindigkeit war beim Item Scanning durch einen Korrelationskoeffizienten von  $r = 0,32$  (p-Wert = 0,23) und beim Cartesian Scanning durch einen Korrelationskoeffizienten von  $r = 0,54$  (p-Wert = 0,03) gekennzeichnet.

Darüber hinaus wurden die Zusammenhänge zwischen der Interaktionsgeschwindigkeit im technischen Abschnitt und der Bewertung der SUS sowie zwischen der Interaktionsgeschwindigkeit beider Abschnitte und der Bewertung des UEQ-Faktors Effizienz untersucht. Die Ergebnisse sind in [Tabelle 5.12](#) dargestellt.

	Geschwindigkeit Technischer Abschnitt und SUS			Geschwindigkeit Gesamt und UEQ-Effizienz		
	r	t-Wert	p-Wert	r	t-Wert	p-Wert
Item	0,186	0,710	0,490	0,385	1,559	0,141
Cartesian	-0,289	-1,129	0,278	-0,148	-0,560	0,584

Zusammenhang: Schwach Mäßig

Tabelle 5.12: Zusammenhang zwischen der Interaktionsgeschwindigkeit und der Bewertung der Usability und der UX

Die Korrelationskoeffizienten zwischen der Interaktionsgeschwindigkeit im technischen Abschnitt und dem UEQ-Faktor Effizienz betragen  $r = 0,186$  (p-Wert = 0,490) für das Item Scanning und  $r = -0,289$  (p-Wert = 0,278) für das Cartesian Scanning. Für die Interaktionsgeschwindigkeit über alle Abschnitte und den UEQ-Faktor Effizienz wurden Korrelationskoeffizienten von  $r = 0,385$  (p-Wert = 0,141) für das Item Scanning und  $r = -0,148$  (p-Wert = 0,584) für das Cartesian Scan errechnet.

### 5.3.7 Qualitative Beobachtungen und Feedback der Teilnehmenden

Neben den erhobenen Daten wurden während der Evaluation Beobachtungen notiert. Besonders auffällig war, dass viele Teilnehmende gezielt Kopfbewegungen einsetzten, um das Cartesian Scanning zu beschleunigen oder kleine Fehler zu korrigieren. In [Abbildung 5.17](#) ist dargestellt, wie viele Personen gezielte Kopfbewegungen zu welchem Zweck einsetzten. Neun Personen setzten Kopfbewegungen für beide Zwecke ein, vier Personen ausschließlich zur Fehlerkorrektur und eine Person ausschließlich zur Beschleunigung. Zwei Personen setzten keine gezielten Kopfbewegungen ein.

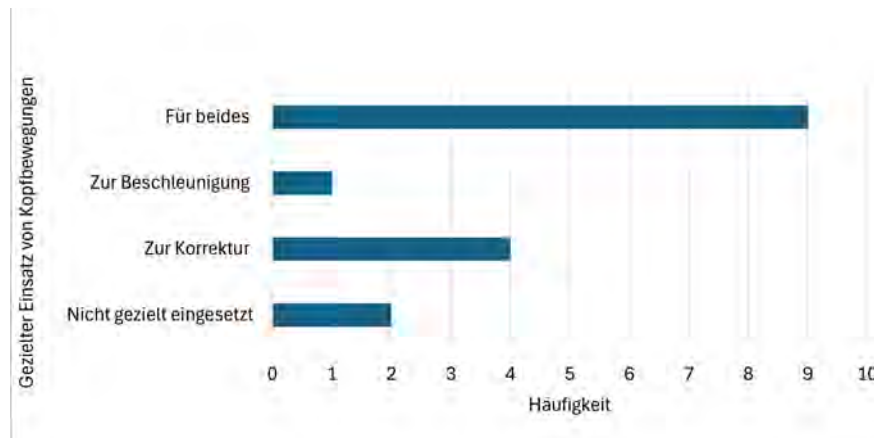


Abbildung 5.17: Einsatz gezielter Kopfbewegungen der Nutzenden beim Cartesian Scanning

Darüber hinaus hatten die Teilnehmenden die Möglichkeit, freies Feedback zu den beiden Schnittstellen zu geben. Dieses Feedback wurde kategorisiert und wird im Folgenden zusammengefasst.

#### Einschätzungen zur Scan Rate:

Die meisten Teilnehmenden äußerten sich bezüglich der Scan Rate der beiden Schnittstellen. **Abbildung 5.18** gibt die diesbezügliche Einschätzung der Teilnehmenden wider. Beim Item Scanning empfanden die meisten Teilnehmenden die Scan Rate als angemessen. Fünf Personen empfanden sie jedoch als zu langsam, eine Person als zu schnell und zwei Personen machten hierzu keine Angabe. Beim Cartesian Scanning empfand keine Person die Scanrate als zu schnell, während sechs Personen sie als angemessen beurteilten. Die meisten Personen gaben an, dass die Scan Rate zu langsam sei. Eine Person äußerte sich nicht.

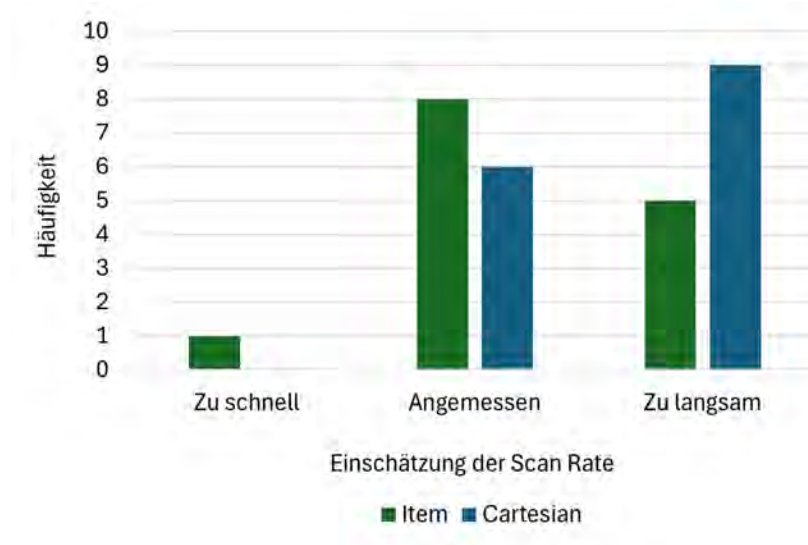


Abbildung 5.18: Einschätzungen der Teilnehmenden zur Scan Rate beider Interaktionschnittstellen

### Probleme und Kritik:

Die meisten geäußerten Kritikpunkte bzw. Einordnungen der aufgetretenen Probleme bezogen sich auf das Item Scanning. Drei Teilnehmende kritisierten die Scanning-Reihenfolge, die als willkürlich oder nicht nachvollziehbar empfunden wurde. Zudem wurde die Animation während des Item Scannings von einer Person als verwirrend beschrieben, da sie dazu führte, dass die Selektion teilweise zu früh erfolgte. Des Weiteren wurde von drei Personen bemängelt, dass das Menü (Navigation und HUD) zu weit oben im Sichtfeld positioniert war. Eine Person äußerte, dass Schwierigkeiten bei der Navigation aufgrund einer zu hohen Scan Rate auftraten. Eine weitere Person merkte an, dass im inhaltsbasierten Abschnitt versteckte Elemente durch das Scanning-Verfahren „gespoilert“ wurden. Dies wurde als schade, aber nicht als ernsthaftes Problem angesehen.

### Vorschläge für Verbesserungen:

Die Teilnehmenden äußerten verschiedene Ideen zur Verbesserung der Schnittstellen. Beim Item Scanning wurde vorgeschlagen, beim ersten Element die Wartezeit etwas zu verlängern, da dieses leicht verpasst werden kann. Außerdem wurde vorgeschlagen, im Navigationsmenü eine Art „Cool Down“ einzuführen, um leichter mehrere Selektionen hintereinander durchführen zu können. Für das Cartesian Scanning wurde vorgeschlagen, die Zeit zu verkürzen, die benötigt wird, um den Schalter für den Moduswechsel gedrückt

zu halten. Zwei Personen äußerten auch den Wunsch, die Geschwindigkeit der Linien dynamisch steuern zu können, um Wartezeiten zu reduzieren und das Scanning-Verfahren effizienter zu gestalten. Außerdem wurde vorgeschlagen, die Scan Rate und die Farben für beide Schnittstellen individuell einstellbar zu gestalten.

#### **Positives Feedback und Aussagen zu Präferenzen zwischen den Schnittstellen:**

Die Teilnehmenden gaben auch Feedback zu Aspekten, die sie als positiv empfanden. Die Teilnehmenden beschrieben das Cartesian Scanning als einfacher und lobten die größere Kontrolle über das Scanning. Es wurde auch betont, dass das Cartesian Scanning intuitiver sei. Es wurde außerdem von zwei Teilnehmenden für die bessere Navigation gelobt, da es einen schnelleren Wechsel des Blickwinkels ermöglichte. Eine teilnehmende Person gab an, dass das Cartesian Scanning zwar einfacher und individueller sei, das Item Scanning jedoch mehr Spaß mache. Auch andere Teilnehmende bevorzugten das Item Scanning, da es als angenehmer empfunden wurde, insbesondere wegen der schnelleren Interaktionsgeschwindigkeit, der geringeren Fehleranfälligkeit und der optischen Gestaltung, die als augenschonender beschrieben wurde.



## 6 Diskussion

In diesem Kapitel werden die Ergebnisse der Evaluation interpretiert und diskutiert, wobei der Fokus darauf liegt, die im Rahmen der Evaluationsplanung formulierten Fragen sowie die Forschungsfrage FF3 zu beantworten. Dies umfasst insbesondere die Untersuchung der Unterschiede zwischen den entwickelten Interaktionsschnittstellen hinsichtlich der Faktoren Effizienz, Robustheit, Usability, User Experience und Motion Sickness. Dabei wird auch geprüft, ob die erzielten Ergebnisse mit den im Vorfeld formulierten Erwartungen übereinstimmen.

Die Erkenntnisse aus der Analyse der Ergebnisse werden anschließend zusammengefasst, indem die Stärken, Schwächen und Verbesserungsmöglichkeiten der entwickelten Ansätze dargestellt werden. Es wird die Frage beantwortet, welches der beiden Verfahren als geeigneter für den Einsatz in VR eingestuft werden kann. Aufbauend auf diesen Erkenntnissen wird diskutiert, inwieweit die Ergebnisse dieser Arbeit genutzt werden können, um Empfehlungen für die Entwicklung barrierearmer Interaktionsschnittstellen in VR auszusprechen. Abschließend wird reflektiert, inwiefern die Zielsetzungen dieser Arbeit erreicht wurden.

### 6.1 Diskussion der Ergebnisse der Evaluation

#### Effizienz

Die Effizienz der beiden entwickelten Interaktionsverfahren wurde anhand der Dauer zur Durchführung der Evaluationsabschnitte und der Interaktionsgeschwindigkeit bewertet. Zwischen dem Item Scanning und dem Cartesian Scanning konnten dabei deutliche Unterschiede festgestellt werden.

Zunächst wird betrachtet, wie lang die jeweilige Interaktionsgeschwindigkeit beider Schnittstellen ist und ob eine Schnittstelle eine deutlich schnellere Interaktionsgeschwindigkeit bietet. Außerdem wird betrachtet, wie lange es im Durchschnitt dauert, die Evaluati-

onsabschnitte zu durchlaufen und ob eine Schnittstelle dabei einen schnelleren Durchlauf begünstigt. Hinsichtlich der Interaktionsgeschwindigkeiten ist zu erkennen, dass das Item Scanning in beiden Evaluationsabschnitten eine deutlich schnellere mittlere Interaktionsgeschwindigkeit aufweist. Darüber hinaus lassen sich bei beiden Verfahren jeweils Unterschiede zwischen der Interaktionsgeschwindigkeit im technischen und im inhaltsbasierten Abschnitt erkennen. Beim Item Scanning ist dieser Unterschied vor allem auf die geringere Anzahl von Elementen in der Szene zurückzuführen, was dazu führt, dass der Scan schneller das gewünschte Zielelement erreicht und dieses somit schneller ausgewählt werden kann. Beim Cartesian Scanning lässt sich die im Vergleich etwas schnelle Geschwindigkeit mit der Position der Elemente im Sichtfeld erklären. Im inhaltsbasierten Abschnitt wurden tendenziell mehr Selektionen mittig im Sichtfeld ausgeführt und vergleichend zum technischen Abschnitt weniger weiter unten im Sichtfeld (vgl. [Abbildung 5.16](#)). Dies begünstigt aufgrund der vorliegenden Korrelation zwischen der Interaktionsgeschwindigkeit und der Position des Elements im Sichtfeld eine schnellere Interaktion.

Auffallend ist außerdem, dass das Cartesian Scanning eine deutlich größere Spannweite der Interaktionsgeschwindigkeiten aufwies. Dies deutet darauf hin, dass einige Teilnehmende mit dem Verfahren vergleichbare Geschwindigkeiten zum Item Scanning erreichen konnten, während andere erheblich länger benötigten. Diese Variabilität kann auf unterschiedliche Nutzungsstrategien zurückgeführt werden, insbesondere auf die gezielte Nutzung von Kopfbewegungen zur Beschleunigung des Scannings. Personen, die ihre Kopfbewegungen bewusst einsetzten, konnten ihre Interaktionen dadurch beschleunigen, während andere ohne diese Strategie langsamer waren. Außerdem variiert die Interaktionsgeschwindigkeit beim Cartesian Scanning stark zwischen Eingaben zur Navigation und Eingaben zur Selektion. Die Navigation wird dabei oft deutlich schneller ausgeführt als die Selektion. Auch dies trägt zu einer höheren Spannweite und Varianz bei. Das Item Scanning hingegen zeigte eine größere Konsistenz zwischen den Teilnehmenden. Dies liegt insbesondere daran, dass die Schnittstelle den Nutzenden keine Möglichkeit bietet, die Geschwindigkeit des Scannings aktiv zu beeinflussen, vor allem, weil die Nutzenden zwecks der Vergleichbarkeit der Daten nicht die Möglichkeit hatten, die Scan Rate zu verändern.

Darüber hinaus ist auch beim Item Scanning erkennbar, dass die Navigation mit schnelleren Interaktionsgeschwindigkeiten einhergeht. So werden bspw. im Navigationsmenü Interaktionen schneller und direkt hintereinander durchgeführt, wenn Nutzende sich mehrfach in eine Richtung drehen möchten. Die Unterschiede in der Interaktionsgeschwindigkeit spiegelten sich auch in der Gesamtzeit wider, die die Teilnehmenden benötigten, um die Evaluationsabschnitte abzuschließen. Sowohl für den technischen als auch für den inhaltsbasierten Abschnitt benötigten die Teilnehmenden mit dem Cartesian Scanning länger als mit dem Item Scanning, wobei der Unterschied im inhaltsbasierten Szenario besonders deutlich war. Hier waren die Teilnehmenden mit dem Item Scanning im Mittel

über 2 Minuten schneller. Dies unterstreicht, dass das Item Scanning insbesondere bei Szenen mit wenigen Elementen einen Vorteil hinsichtlich der Interaktionsgeschwindigkeit bietet.

Somit kann festgestellt werden, dass die zuvor formulierten Erwartungen hinsichtlich der schnelleren Interaktionsgeschwindigkeit des Item Scannings gegenüber des Cartesian Scannings durch die Evaluationsergebnisse grundsätzlich bestätigt wurden. Allerdings fiel der Unterschied deutlicher aus als angenommen.

Aus diesen Ergebnissen kann geschlossen werden, dass beim Item Scanning insbesondere die Anzahl der Elemente in der Szene als wichtigster Einflussfaktor für die Interaktionsgeschwindigkeit identifiziert werden konnte. Beim Cartesian Scanning lassen sich zwei relevante Einflussfaktoren identifizieren. Dazu zählt zunächst das Ausmaß, in dem Kopfbewegungen zur Beschleunigung des Scannings eingesetzt werden. Je deutlicher und häufiger diese eingesetzt werden, desto schnellere Interaktionsgeschwindigkeiten können erreicht werden. Der zweite Einflussfaktor ist die Position der Elemente im Sichtfeld. Die durchgeführte Korrelationsanalyse hat gezeigt, dass ein signifikanter Zusammenhang zwischen der Interaktionsgeschwindigkeit und der Position vorliegt. Je weiter oben und links sich ein Element im Sichtfeld befindet, desto geringer ist die benötigte Zeit zur Selektion. Dies liegt daran, dass die Scan-Linien entsprechend länger in ihrer Bewegung brauchen, um Elemente weiter rechts oder unten im Sichtfeld zu erreichen.

Als letztes wurde hinsichtlich der Effizienz untersucht, ob die Interaktionsgeschwindigkeit die wahrgenommene Usability und Effizienz beeinflusst. Für die Usability konnte nur ein schwacher Zusammenhang zwischen der Interaktionsgeschwindigkeit und dem SUS-Score festgestellt werden. Dieser war entsprechend auch statistisch nicht signifikant. Beim Cartesian Scanning zeigte sich eine etwas deutlichere Tendenz, dass längere Interaktionsgeschwindigkeiten mit niedrigeren Usability-Bewertungen in Zusammenhang stehen. Aber auch hier besteht insgesamt nur eine schwache, nicht signifikante Korrelation.

Hinsichtlich des Zusammenhangs zwischen der Interaktionsgeschwindigkeit und der Bewertung des UEQ-Faktors Effizienz zeigen die Ergebnisse für beide Verfahren unterschiedliche Tendenzen. Während beim Cartesian Scanning nur eine schwache Korrelation bestand, zeigte das Item Scanning eine mäßige, wenn auch statistisch nicht signifikante, Korrelation. Auffällig ist hier, dass es sich um eine positive Korrelation handelt. Das bedeutet, dass langsamere Geschwindigkeiten tendenziell mit besseren Bewertungen der Effizienz einhergehen.

## Robustheit

Hinsichtlich der Robustheit wurde untersucht, wie häufig Fehler bei der Nutzung der jeweiligen Interaktionsschnittstelle auftreten und was die Gründe für diese Fehler sind. Darüber hinaus wurde analysiert, wie viele Zyklen des Scannings beim Item Scanning durchschnittlich für die Ausführung einer Interaktion nötig waren und wie häufig beim Cartesian Scanning leere Eingaben zur Korrektur von Fehlern eingesetzt wurden.

Beim Item Scanning traten insgesamt mehr Fehler im technischen Abschnitt auf, in dem nur drei Teilnehmende fehlerfrei blieben. Die meisten Teilnehmenden machten hier einen oder direkt mehrere Fehler, die häufig auf Timing-Probleme zurückzuführen waren. Diese resultierten insbesondere aus der Schwierigkeit, den richtigen Zeitpunkt für die Eingabe zu treffen. Dieses Problem trat bei allen Personen auf, die nicht fehlerfrei blieben. Aus der Rückmeldung der Teilnehmenden lässt sich schließen, dass u. a. die Animation zwischen den Elementen das Auftreten dieser Fehler begünstigte, da sie zu einer verfrühten Selektion verleitete. Darüber hinaus traten Timing-Probleme verstärkt im Navigationsmenü auf. Hier kam es oft dazu, dass die Teilnehmenden eine Navigation durchführten, sich kurz umsahen und im Anschluss eine erneute Navigation in die gleiche Richtung durchführen wollten. Oft wechselte jedoch in diesem Moment der Scan zum nächsten hervorgehobenen Element, wodurch eine Fehlselektion entstand, die eine Navigation in die falsche Richtung auslöste. Mögliche Ansätze, um dem Auftreten dieser Fehler entgegenzuwirken, wären bspw. eine Optimierung der Animation und das Einführen einer längeren Wartezeit auf dem zuvor ausgewählten Element im Navigationsmenü.

Im Rahmen der Gestaltung des technischen Abschnitts für das Item Scanning wurden als Herausforderung eine nicht deutliche Scan-Reihenfolge sowie die Positionierung von Elementen am Rand des Sichtfelds genannt. In der Evaluation traten allerdings keine Fehler auf, die sich auf diese Gründe zurückführen lassen. Längere Wartezeiten bei Zielelementen, die sich weiter hinten in der Scan-Reihenfolge befinden, haben das Auftreten von Fehlern ebenfalls nicht begünstigt. Auch die Eingabe auf mehreren Ebenen stellte kein Problem dar.

Beim Cartesian Scanning war das Fehlerprofil vielseitiger. Im technischen Abschnitt blieben mehr Teilnehmende fehlerfrei, aber wenn Teilnehmende Fehler machten, dann häufig auch direkt zwei oder mehr. Eine der Hauptursachen dafür war die versehentliche Selektion des Hauptmenüs. Wie bereits gezeigt, tendieren die Teilnehmenden dazu, ihre Blickrichtung so auszurichten, dass die Zielelemente möglichst mittig im Sichtfeld selektiert werden können. Da das Menü ebenfalls mittig platziert ist, weist es dementsprechend eine räumliche Nähe zu den Zielelementen auf, was das Auftreten von Selektionsfehlern begünstigt. Außerdem befindet sich das Menü auf der UI-Ebene und damit räumlich näher am Nutzenden. Dadurch wird es von der Implementierung bevorzugt selektiert. Daher können bereits leichte Überdeckungen von Elementen oder Eingaben,

die das Zielelement in Richtung Menü knapp verfehlen, dazu führen, dass das Menü anstelle des Zielelements selektiert wird. Eine andere Positionierung des Menüs könnte somit dem Auftreten dieser Fehlselektionen entgegenwirken.

Ein weiterer häufiger Grund für Fehler stellt das Vergessen des Wechsels des Interaktionsmodus dar. Teilnehmende hatten Schwierigkeiten, sich daran zu erinnern, den Modus zu wechseln, bevor sie mit der Interaktion begannen. So wurde meist das Scanning initialisiert und beim Setzen der ersten Linie erkannt, dass es sich um den falschen Modus handelt. Dies zeigt, dass ein Bedarf besteht, den aktuellen Interaktionsmodus auch dann zu visualisieren, wenn das Scanning nicht aktiv ist und die Scan-Linien entsprechend nicht sichtbar sind.

Interessanterweise traten keine Fehler auf, die auf die erwartete Schwierigkeit bei der Selektion von Elementen am Rand des Sichtfelds oder nah aneinander liegender Elemente, wie bspw. bei den Antwortmöglichkeiten von Frage-Feldern, zurückzuführen waren. Hierbei ist anzumerken, dass viele Teilnehmende diese Herausforderungen jedoch auch durch Anpassung der Blickrichtung über die Navigation oder die Bewegung des Kopfes umgingen.

Zusammenfassend bestätigen diese Ergebnisse die im Vorfeld geäußerten Erwartungen hinsichtlich der Unterschiede in der Robustheit der beiden Schnittstellen.

Hinsichtlich der Anzahl der benötigten Zyklen beim Item Scanning kann festgestellt werden, dass die meisten Eingaben bereits im ersten Zyklus des Scannings erfolgten. Das bedeutet, Elemente wurden meistens direkt ausgewählt, sobald sie erstmalig hervorgehoben wurden. Dabei traten keine großen Unterschiede zwischen dem technischen und dem inhaltsbasierten Evaluationsabschnitt auf. Mehrere Zyklen wurden vor allem bei der Navigation benötigt, da sich die Teilnehmenden oft zunächst orientieren oder bewusst einen weiteren Zyklus abwarteten, um Timing-Fehler zu vermeiden. Dies zeigt, dass die Möglichkeit, mehrere Zyklen zu durchlaufen von den Teilnehmenden teilweise als bewusste Strategie zur Fehlervermeidung eingesetzt wurde. Die zuvor beschriebenen Optionen zur Optimierung der Selektion im Navigationsmenü könnten daher ebenfalls dazu führen, dass die Notwendigkeit von mehreren Zyklen geringer wird.

Beim Cartesian Scanning stellte die Nutzung von leeren Eingaben eine Strategie zur Korrektur von Fehlern dar. Im technischen Szenario lag der Mittelwert bei einer leeren Eingabe pro Teilnehmer:in, während im inhaltsbasierten Szenario ein höherer Mittelwert von 1,75 erreicht wurde. Die Spannweite der Nutzung war dabei jedoch recht groß. Während einige Teilnehmende diese Option zur Korrektur gar nicht nutzten, griffen andere häufiger darauf zurück, insbesondere wenn zuvor der Wechsel des Interaktionsmodus vergessen wurde. Bei fehlerhaften Setzen der ersten Scan-Linie wurden leere Eingaben hingegen seltener verwendet. Hier versuchten die meisten Teilnehmenden eher, dies durch gezielte Kopfbewegungen zu korrigieren, anstatt den Prozess durch eine leere Ein-

gabe abzubrechen und neu zu starten.

Die Ergebnisse zeigen, dass leere Eingaben ein essenzielles Werkzeug zur Fehlerkorrektur darstellen und die Robustheit der Schnittstelle erhöhen. Ohne diese Option wären vermutlich mehr Fehler aufgetreten.

### **Erlernbarkeit**

Um zu untersuchen, ob im Verlauf der Evaluation ein messbarer Lerneffekt eingetreten ist, wird analysiert, ob im Verlauf der Evaluation eine Beschleunigung der Interaktionsgeschwindigkeit oder eine Reduktion der Fehleranzahl eingetreten ist.

Beim Item Scanning zeigen die Ergebnisse der durchgeführten linearen Regressionsanalyse, dass die Interaktionsgeschwindigkeit im Verlauf des technischen und im inhaltsbasierten Abschnitt jeweils minimal schneller wird (um 0,839 bzw. 0,391 Sekunden pro Interaktion). Wird die gesamte Evaluation betrachtet, liegt der Effekt bei -2,539 Sekunden und somit deutlich höher. Der Wert wird jedoch durch die geringere Anzahl an Elementen im inhaltsbasierten Abschnitt beeinflusst. Die geringere Anzahl an Elementen hat hier eine geringe Interaktionsgeschwindigkeit begünstigt. Daher ist die Verringerung der Interaktionsgeschwindigkeit über den gesamten Verlauf der Evaluation nicht direkt auf einen Lerneffekt zurückzuführen. Die getrennte Betrachtung der Abschnitte wird daher in diesem Fall als repräsentativer für den Lerneffekt angesehen. Die separate Betrachtung der Entwicklung in den beiden Abschnitten deutet darauf hin, dass keine deutliche Beschleunigung der Interaktionsgeschwindigkeit eingetreten ist, die auf einen Lerneffekt schließen lässt. Die Korrelationskoeffizienten zwischen der Anzahl der getätigten Eingaben und der Interaktionsgeschwindigkeit verdeutlichen dies zusätzlich. Der Zusammenhang ist minimal und im inhaltsbasierten Abschnitt so gering, dass er vernachlässigt werden kann. Dies kann dadurch erklärt werden, dass die Teilnehmenden beim Item Scanning nicht die Möglichkeit hatten, aktiv Einfluss auf die Geschwindigkeit des Verfahrens zu nehmen, da u. a. die Scan Rate für alle Personen zwecks Vergleichbarkeit der Ergebnisse festgelegt wurde und nicht anpassbar war. Ein Lerneffekt hätte sich theoretisch darüber hinaus in einer geringeren Anzahl an benötigten Scanning-Zyklen zeigen können, was wiederum zu einer geringeren Interaktionsgeschwindigkeit geführt hätte. Da die meisten Eingaben jedoch im ersten oder zweiten Durchgang erfolgen, bleibt auch hier nur minimaler Spielraum für eine Verbesserung.

Beim Cartesian Scanning zeigt sich ein noch schwächerer Zusammenhang zwischen der Anzahl getätigter Eingaben und der Interaktionsgeschwindigkeit. Werden die Abschnitte separat betrachtet, verlangsamt sich die Interaktionsgeschwindigkeit tendenziell sogar im Verlauf des Abschnitts (positiver Regressionskoeffizient). Auf die gesamte Evaluation bezogen ist hingegen eine leichte Beschleunigung (um 0,49 Sekunden) erkennbar. Dieser Wert ist hier aussagekräftiger als beim Item Scanning, da beim Cartesian Scanning Faktoren wie die Anzahl der Elemente in der Szene keinen erkennbaren Einfluss

auf die Interaktionsgeschwindigkeit haben. Dennoch bestätigen die Korrelationskoeffizienten, dass auch beim Cartesian Scanning praktisch kein signifikanter Zusammenhang besteht, der das Auftreten eines Lerneffekts aufzeigen könnte.

Ein weiteres Indiz für die Erlernbarkeit eines Verfahrens könnte die Reduktion der Fehlerzahl mit zunehmender Erfahrung sein. Die Ergebnisse zur Robustheit zeigen, dass bei beiden Verfahren die Gesamtzahl der Fehler vom technischen zum inhaltsbasierten Abschnitt abnimmt. Dieser Rückgang könnte auf einen Lerneffekt hindeuten. Die höhere Fehlerzahl im technischen Abschnitt könnte jedoch auch mit der höheren Anzahl an Elementen und der bewusst herausfordernden Gestaltung des Abschnitts zusammenhängen, da diese Faktoren das Auftreten von Fehlern in diesem Abschnitt generell begünstigt haben könnten. Ein Lerneffekt kann daher auch in Bezug auf die Fehlerreduktion anhand der vorliegenden Daten nicht sicher nachgewiesen werden. Die Annahme, dass das Item Scanning leichter zu erlernen ist, kann daher anhand der erhobenen objektiven Daten weder bestätigt noch widerlegt werden.

### **Ablenkung durch die Interaktionsschnittstellen**

Um zu untersuchen, inwieweit die Nutzung der Schnittstellen von den Inhalten der VR-Anwendung ablenkt, wurden in Anlehnung an den Presence Questionnaire von [Witmer and Singer \[1998\]](#) drei Fragen formuliert. Diese Fragen sollen Aufschluss darüber geben, ob die Schnittstellen als störend und ablenkend empfunden wurden.

Die Analyse der Ergebnisse zeigt, dass beide Schnittstellen in diesem Aspekt überwiegend positiv bewertet und als wenig störend wahrgenommen wurden, wobei das Cartesian Scanning tendenziell etwas besser bewertet wurde. Die Ergebnisse zeigen, dass sich die Teilnehmenden bei beiden Verfahren gut auf die Inhalte und Aufgaben in der virtuellen Szene konzentrieren konnten. Dies deutet darauf hin, dass die Interaktionsschnittstellen keine übermäßige kognitive Belastung erzeugten. Das Gefühl, dass die Nutzenden in die virtuelle Umgebung involviert sind, war beim Cartesian Scanning durchschnittlich etwas stärker ausgeprägt. Dies könnte auf eine höhere Immersion und ein verstärktes Gefühl von Presence hinweisen. Die Streuung der Bewertungen war hierbei beim Item Scanning höher, was darauf hindeutet, dass die Wahrnehmung der Schnittstelle diesbezüglich stärker von individuellen Präferenzen abhing.

Während beide Schnittstellen es den Teilnehmenden ermöglichten, sich auf die Inhalte zu fokussieren, gab es beim Cartesian Scanning diesbezüglich größere Unterschiede zwischen den Personen. Dies deutet darauf hin, dass sich einige Teilnehmende stärker auf die Bedienung konzentrieren mussten als andere.

Insgesamt bestätigen die Ergebnisse, dass beide Verfahren so gestaltet sind, dass sie die Aufmerksamkeit der Nutzenden nicht von den Inhalten ablenken und keine übermäßigen Störungen verursachen. Die Annahme, dass die entwickelten Schnittstellen eine geringe Ablenkung vom Inhalt aufweisen, konnte damit bestätigt werden. Diese Ergebnisse zeigen, dass binäre Interaktionsschnittstellen trotz ihrer Einfachheit so gestaltet werden können, dass sie das Erlebnis nicht wesentlich beeinträchtigen und zugleich ein Gefühl von Presence erzeugt werden kann.

### Usability

Die Usability der beiden Interaktionsschnittstellen wurde von den Teilnehmenden unterschiedlich bewertet, wobei das Cartesian Scanning im Vergleich zum Item Scanning deutlich besser abschnitt. Die folgenden Einordnungen der SUS-Scores basieren auf den Ergebnissen von [Aaron Bangor and Miller \[2008\]](#). Der SUS-Score des Cartesian Scanning lag mit einem Durchschnitt von 85,63 im Bereich einer guten bis exzellenten Usability, was etwa einer Schulnote „2“ (amerikanisches B) entspricht. Dies deutet darauf hin, dass die Mehrheit der Teilnehmenden die Schnittstelle als sehr nutzerfreundlich empfand.

Beim Cartesian Scanning bewerteten 11 von 16 Personen die Usability mit einem Score über 80, was als gut bis exzellent eingeordnet werden kann. Sechs dieser Personen erreichten sogar Werte über 92, die mit einer Schulnote „1“ vergleichbar ist. Lediglich zwei Teilnehmende vergaben Scores im Bereich zwischen 50 und 60, was als unterdurchschnittlich betrachtet wird und an der Grenze zur Nicht-Akzeptanz liegt. Insgesamt verdeutlicht die Verteilung der Scores, dass die Bewertungen recht einheitlich ausfielen und nur wenige Personen Schwierigkeiten bei der Nutzung des Cartesian Scannings hatten.

Im Gegensatz dazu erzielte das Item Scanning einen durchschnittlichen SUS-Score von 73,44, der ebenfalls über dem Durchschnitt liegt, aber nur knapp im Bereich einer guten Usability (Schulnote „3“, amerikanisches C). Die Bewertungen zeigen außerdem eine deutlich größere Streuung der Scores. Während neun Personen eine Bewertung über 68 (durchschnittliche Usability) vergaben, bewerteten 7 sieben Personen die Schnittstelle mit einem geringeren Score, wobei vier Teilnehmende sogar Scores unter 60 vergaben, die als mangelhaft oder nicht akzeptabel einzustufen sind. Eine Person bewertete die Schnittstelle mit einem Score unter 50, was auf eine klare Nicht-Akzeptanz hinweist.

Diese Verteilung deutet darauf hin, dass die Bewertung des Item Scanning stärker von anderen Einflussfaktoren, wie möglicherweise individuellen Präferenzen, abhängt als das Cartesian Scanning. Während ein Großteil der Teilnehmenden die Usability als gut bis sehr gut empfand, gab es auch einen deutlichen Anteil, der die Schnittstelle als verbesserungswürdig oder gar nicht akzeptabel bewertete.

Insgesamt zeigt die Analyse, dass das Cartesian Scanning hinsichtlich der Usability als überzeugender bewertet wurde. Beim Item Scanning besteht hingegen noch deutlicher



Verbesserungsbedarf, um die Usability für alle Anwender:innen auf ein gleichmäßig hohes Niveau zu bringen. Die Ergebnisse lassen jedoch keine Rückschlüsse darauf zu, welche spezifischen Faktoren beim Item Scanning die Bewertung beeinflusst und damit die hohe Streuung der Ergebnisse hervorgerufen haben.

### User Experience

Die Evaluation der UX zeigt, dass das Cartesian Scanning in den Faktoren durchschnittlich besser bewertet wurde als das Item Scanning, mit Ausnahme des Faktors Originalität. Beide Verfahren wurden in allen UX-Faktoren durchschnittlich positiv bewertet ( $MW > 0,8$ ), wobei sich deutliche Unterschiede in den Einzelbewertungen und damit in den Varianzen zeigen.

Beim Cartesian Scanning wurden vor allem die Faktoren Durchschaubarkeit und Attraktivität positiv hervorgehoben, während die Effizienz am schlechtesten bewertet wurde. Die geringeren Varianzen in den Bewertungen zeigen, dass die Teilnehmenden sich in ihren Einschätzungen weitgehend einig waren, insbesondere in Bezug auf Durchschaubarkeit und Attraktivität. Items, die besonders positiv wahrgenommen wurden, waren *verständlich*, *leicht zu lernen*, *angenehm*, *übersichtlich* und *aufgeräumt*. Trotz der allgemein positiven Rückmeldungen wurde das Item *langsam* deutlich negativ bewertet, was die geringe Bewertung des Faktor Effizienz bedingt.

Die UX des Item Scanning wurde ebenfalls positiv bewertet, jedoch mit insgesamt größeren Varianzen, was auf stärkere Uneinigkeit unter den Teilnehmenden hindeutet. Besonders niedrig bewertet wurden die Faktoren Effizienz und Steuerbarkeit. Die niedrigere Bewertung der Steuerbarkeit könnte u. a. durch die Scan-Reihenfolge bedingt sein. Diese wird durch das System festgelegt und lässt sich nicht aktiv durch den Nutzenden beeinflussen. Insbesondere im technischen Abschnitt empfanden einige Teilnehmende die Reihenfolge der Elemente als nicht nachvollziehbar oder nicht erwartungskonform, was aus dem Feedback der Personen hervorgeht. Darüber hinaus könnte auch die geringe Kontrolle über die Interaktionsgeschwindigkeit ein möglicher Grund für die geringere Bewertung des Faktor darstellen. Besonders positiv bewertet wurde hingegen der Faktor Originalität. Insbesondere die Items *kreativ* und *neuartig* wurden positiv hervorgehoben. Aber auch das Item *leicht zu lernen* erhielt eine durchaus positive Bewertung.

Ein zentraler Kritikpunkt bei beiden Schnittstellen ist die Effizienz, die in beiden Fällen am niedrigsten bewertet wurde. Dies ist verständlich, da Scanning-Verfahren grundsätzlich als langsam zu bewerten sind, insbesondere im Vergleich zu anderen gängigen Interaktionsverfahren. Eine Verbesserung der Bewertung der Effizienz könnte vermutlich bereits durch eine Einstellung zur Anpassung der Scan Rate erreicht werden. Wie die Ergebnisse aus dem Feedback zeigen, wurde diese von vielen Teilnehmenden als zu langsam empfunden, insbesondere beim Cartesian Scanning. Eine individuelle Einstellung könnte

die wahrgenommene Effizienz deutlich beeinflussen.

Um die Bewertung der Steuerbarkeit des Item Scannings zu verbessern, könnte neben der Anpassbarkeit der Scan Rate auch die Anpassung der Scan-Reihenfolge hilfreich sein. Es wäre denkbar, eine Möglichkeit zu implementieren, die bei Erstellung einer Szene oder im Editor-Modus die Option bietet, die Scan-Reihenfolge festzulegen, damit eine nachvollziehbare Lösung für jedes individuelle Szenario geschaffen werden kann.

Anschließend zur Auswertung des UEQ wurde ergänzend überprüft, inwieweit die subjektiven Angaben hinsichtlich der Effizienz mit den gemessenen technischen Daten übereinstimmen. Dazu wurde eine Korrelationsanalyse der Bewertung des Faktors Effizienz und der Interaktionsgeschwindigkeit durchgeführt. Beim Item Scanning liegt zwischen den Variablen eine mäßige, positive Korrelation vor. Diese zeigt, dass langsamere Interaktionsgeschwindigkeiten tendenziell mit besseren Bewertungen der Effizienz einhergingen. Dieses Ergebnis deutet darauf hin, dass die Bewertung der Effizienz nicht allein durch Geschwindigkeit bestimmt wird und stattdessen viel mehr subjektive Faktoren eine entscheidende Rolle spielen.

Beim Cartesian Scanning wurde nur eine leicht negative Korrelation festgestellt, was bedeutet, dass höhere Geschwindigkeiten tendenziell mit besseren Bewertungen der Effizienz zusammenhängen. In diesem Fall scheint die Geschwindigkeit etwas stärker mit der Wahrnehmung von Effizienz verknüpft zu sein als beim Item Scanning. Der Zusammenhang ist insgesamt jedoch als recht schwach zu bewerten. Beide Korrelation weisen keine statistische Signifikanz auf, trotzdem verdeutlichen sie, dass eine schnellere Interaktionsgeschwindigkeit kein alleiniger ausschlaggebender Faktor für die Bewertung der Effizienz ist.

### **Motion Sickness**

Die Evaluation der beiden Schnittstellen zeigt, dass Motion Sickness Symptome bei beiden Schnittstellen auftraten, jedoch mit Unterschieden in ihrer Ausprägung und Verteilung. Im Durchschnitt war der SSQ-Score beim Item Scanning leicht höher (12,155) als beim Cartesian Scanning (10,051). Beide Werte weisen auf das Vorhandensein signifikanter Symptome hin, wobei die Verteilung der Scores zwischen den Verfahren variiert. Beim Cartesian Scanning hatten mehr als die Hälfte der Teilnehmenden (10 Personen) einen Score unter 10, was vernachlässigbaren bis minimalen Symptomen entspricht, während drei Personen einen Score über 20 erreichten, was als besorgniserregend gilt. Beim Item Scanning zeigten hingegen acht Personen einen Score unter 10, während 5 Personen einen Score über 20 aufwiesen. Dies deutet darauf hin, dass insgesamt beim Item Scanning häufiger stärkere Symptome auftraten.

Eine Betrachtung der spezifischen Symptome zeigt, dass in der Kategorie Oculomotor die meisten Symptome auftraten, während in der Kategorie Nausea bei beiden Verfahren die wenigsten Symptome berichtet wurden. Beim Item Scanning berichteten die Teilnehmenden besonders häufig über angestrengte Augen (elf Personen, davon vier mit moderater Ausprägung) sowie Ermüdung, Kopfdruck und verschwommenes Sehen. Beim Cartesian Scanning traten ebenfalls häufig angestrengte Augen (elf Personen, davon zwei moderat) sowie leichte Ermüdung (acht Personen) auf, wobei Symptome wie Kopfdruck und verschwommenes Sehen weniger verbreitet waren als beim Item Scanning. Die stärkeren Ausprägung oculomotorischer Symptome bei beiden Schnittstellen deutet darauf hin, dass beide Schnittstellen eine kognitive Belastung hervorrufen.

In der Kategorie Disorientation wurden bei beiden Schnittstellen die größten Streuungen festgestellt. Während acht Personen beim Item Scanning und neun beim Cartesian Scanning vernachlässigbare Symptome berichteten (Score  $<5$ ), erreichten einige Teilnehmende sehr hohe Scores ( $>40$ ), was auf erhebliche Einschränkungen hindeutet. Es lässt sich erkennen, dass Disorientation zwar selten, jedoch bei einzelnen Personen besonders ausgeprägt auftreten kann.

Ein entscheidender Einflussfaktor auf die Symptome könnte der Aufbau der Evaluation selbst gewesen sein. Beide Schnittstellen wurden in einer Sitzung getestet, die 45–60 Minuten dauerte, und erforderten die Erlernung neuer Steuerungen sowie die Bearbeitung vieler Aufgaben. Diese intensive kognitive und visuelle Belastung könnte das Auftreten von oculomotorischen Symptomen wie angestrengte Augen, Ermüdung und Konzentrationsschwierigkeiten begünstigt haben. Dennoch sollte über Optimierungen der Verfahren nachgedacht werden, um die Symptome insgesamt zu reduzieren.

Insgesamt zeigen die Ergebnisse, dass die Annahmen hinsichtlich des Komforts teilweise bestätigt werden können. Die Ergebnisse zeigen, dass insbesondere Symptome auftreten, die auf kognitive oder visuelle Anstrengung zurückzuführen sind. Allerdings sind die Symptome teilweise stärker ausgeprägt als erwartet. Inwieweit dies durch das Evaluationsdesign begünstigt wurde, lässt sich nicht mit Sicherheit beurteilen. Entgegen den Erwartungen zeigen die Ergebnisse zudem einen leichten Unterschied zwischen den Schnittstellen.

Nach der Auswertung des SSQ wurde untersucht, welche Zusammenhänge zwischen dem Auftreten von Motion Sickness und der Bewertung der Usability und UX sowie der Interaktionsgeschwindigkeit bestehen. Für die Usability ergab die Korrelationsanalyse beim Item Scanning einen schwachen positiven Zusammenhang. Dies bedeutet, dass höhere SSQ-Scores tendenziell mit einer etwas besseren Bewertung der Usability einhergingen. Beim Cartesian Scanning wurde hingegen ein mäßiger negativer Zusammenhang festgestellt. Hier schienen niedrigere SSQ-Scores zu einer besseren Bewertung der Usability

beizutragen. Dies entspricht eher der allgemeinen Erwartung, dass eine Schnittstelle, die weniger physische oder kognitive Belastungen verursacht, auch als einfacher und angenehmer wahrgenommen wird. Diese Korrelation war jedoch nicht statistisch signifikant. Ein wichtiger Aspekt, der bei der Interpretation dieser Ergebnisse berücksichtigt werden muss, ist der Zeitpunkt der Datenerhebung. Die SUS wurde unmittelbar nach dem technischen Abschnitt ausgefüllt, während der SSQ erst am Ende der Evaluation erhoben wurde. Symptome, die erst während des inhaltsbasierten Abschnitts auftraten, bspw. aufgrund der längeren Nutzungszeit oder erhöhter kognitiver Anforderung, konnten die SUS-Bewertung daher nicht beeinflussen. Dies könnte die positive Korrelation beim Item Scanning erklären und darauf hindeuten, dass ggf. beim Item Scanning die Symptome erst im weiteren Verlauf der Evaluation aufgetreten sind oder sich hier stärker ausgeprägt haben.

Hinsichtlich der UX wurden in der Korrelationsanalyse insbesondere die Faktoren Attraktivität und Durchschaubarkeit betrachtet. Beim Cartesian Scanning zeigte sich ein leichter negativer Zusammenhang zwischen dem SSQ-Score und der Bewertung der Attraktivität. Dies deutet darauf hin, dass die Schnittstelle eher als angenehm und sympathisch wahrgenommen wird, wenn weniger körperliche Symptome auftreten. Beim Item Scanning hingegen wurde ein schwacher positiver Zusammenhang festgestellt. Dadurch, dass der Zusammenhang bei beiden Schnittstellen eher schwach ist, kann angenommen werden, dass das Auftreten von Motion Sickness nur einen geringen Einfluss auf die Bewertung der Attraktivität hat und hier andere Faktoren eine wichtigere Rolle einnehmen.

Ein deutlicherer Zusammenhang besteht zwischen dem SSQ und dem UX-Faktor Durchschaubarkeit. Hier wurde bei beiden Schnittstellen eine negative Korrelation festgestellt, die beim Cartesian Scanning als stark bewertet werden kann und sogar eine statistische Signifikanz aufwies. Dies legt nahe, dass weniger Symptome mit einer besseren Bewertung der Durchschaubarkeit einhergehen. Dies deutet darauf hin, dass eine klar strukturierte und leicht verständliche Schnittstelle die Teilnehmenden weniger belastet und somit weniger Symptome hervorruft. Es wäre aber auch möglich, dass eine Schnittstelle, die weniger Symptome verursacht, wiederum als übersichtlicher und intuitiver empfunden wird. Die Frage nach der Kausalität kann daher an dieser Stelle nicht beantwortet werden.

Die Korrelationsanalyse zwischen den SSQ-Scores und der Interaktionsgeschwindigkeit zeigt, dass schnellere Interaktionsgeschwindigkeiten tendenziell mit niedrigeren SSQ-Scores und somit weniger Symptomen einhergehen. Beim Cartesian Scanning war dieser Zusammenhang stark und statistisch signifikant, während er beim Item Scanning mäßig ausgeprägt war. Dieser Zusammenhang kann auf zwei Arten interpretiert werden. Einerseits wäre es möglich, dass das Auftreten der Symptome die Interaktionsgeschwindigkeit

beeinflusst. Das bedeutet, dass Personen schneller interagieren können, wenn sie weniger Symptome verspüren. Symptome wie angestrenzte Augen, Ermüdung oder Konzentrationsschwierigkeiten könnten die Reaktionszeit und somit die Interaktionsgeschwindigkeit beeinträchtigen. Hinsichtlich des Cartesian Scannings wäre es auch möglich, dass Personen, die Symptome verspüren, eher dazu neigen, Kopfbewegungen zu vermeiden, wodurch eine Beschleunigung der Interaktion nicht vorgenommen werden kann. Andererseits könnte auch die Geschwindigkeit die Symptome beeinflussen. Personen, die in der Lage sind, schneller zu interagieren, könnten weniger Symptome entwickeln, da kürzere Interaktionszeiten die okulomotorische und kognitive Belastung verringern könnten. Beim Item Scanning war die Korrelation schwächer ausgeprägt, was möglicherweise darauf zurückzuführen ist, dass die Interaktionsgeschwindigkeit hier weitgehend durch das System vorgegeben war und nicht aktiv von den Nutzenden beeinflusst werden konnte. Beim Cartesian Scanning hingegen hatten die Teilnehmenden mehr Kontrolle über die Geschwindigkeit, was zu einer stärkeren Korrelation geführt haben könnte.

### **Limitation der Evaluation hinsichtlich der Teilnehmenden**

Bei der Einordnung der Ergebnisse ist zu berücksichtigen, dass die Evaluation mit einer vergleichsweise kleinen Anzahl von Teilnehmenden durchgeführt wurde. Die Ergebnisse können daher vor allem Tendenzen und Hinweise liefern, wobei statistische Signifikanzen aufgrund der begrenzten Datenmenge schwer zu erreichen sind.

Ein weiterer zentraler Aspekt ist die Zielgruppe, auf die die Ergebnisse übertragen werden sollen. Die Teilnehmenden setzten sich ausschließlich aus Personen ohne motorische Beeinträchtigungen zusammen, die in ihrem Alltag keine binären Interaktionsschnittstellen oder Scanning-Verfahren nutzen. Dies könnte die Ergebnisse in mehrfacher Hinsicht beeinflusst haben. So könnten bspw. Bewertungen im UEQ zu Faktoren wie Originalität oder Stimulation in dieser Evaluation besonders positiv ausgefallen sein, da das Konzept für die Teilnehmenden eine Neuartigkeit darstellt. Personen aus der Zielgruppe, die in ihrem Alltag bereits Schalter und Scanning-Verfahren verwenden, könnten diese Aspekte hingegen anders bewerten, da ihnen vergleichbare Mechanismen vertrauter sind.

Darüber hinaus könnten sich auch technische Ergebnisse wie Interaktionsgeschwindigkeiten oder Fehlerraten bei Personen mit motorischen Beeinträchtigungen unterscheiden. Unterschiede in der motorischen Kontrolle oder der Vertrautheit mit binären Schnittstellen könnten dazu führen, dass die Interaktionszeiten und die Präzision der Eingaben abweichend ausfallen. Insgesamt ist daher zu beachten, dass die Ergebnisse der Evaluation in dieser Arbeit nicht ohne Weiteres auf die tatsächliche Zielgruppe übertragbar sind, sondern eine weiterführende Untersuchung mit entsprechenden Teilnehmenden erforderlich ist, um belastbare und repräsentative Ergebnisse zu erzielen.

## 6.2 Zusammenfassung und Schlussfolgerungen

Basierend auf den oben dargestellten Ergebnissen werden im Folgenden die Stärken und Schwächen sowie mögliche Verbesserungsansätze der beiden Schnittstellen zusammenfassend dargestellt.

### Stärken der Schnittstellen

Das Item Scanning zeichnet sich durch eine schnelle Interaktionsgeschwindigkeit aus, insbesondere wenn die Anzahl der Elemente in der Szene gering ist. Zudem ist die Schnittstelle weniger fehleranfällig und kann insgesamt als robust bezeichnet werden. Die Schnittstelle ermöglicht es den Teilnehmenden, sich gut auf die Erfüllung ihrer Aufgaben zu konzentrieren, ohne dass die Steuerung als störend oder ablenkend vom Inhalt der virtuellen Umgebung empfunden wird. Die UX des Item Scannings wurde insgesamt positiv bewertet, wobei insbesondere der Faktor Originalität hervorgehoben wurde. Die Schnittstelle wurde als kreativ und innovativ bewertet. Darüber hinaus wird sie von den Teilnehmenden als verständlich und leicht erlernbar beurteilt.

Das Cartesian Scanning bietet ein hohes Maß an Kontrolle und Selbstwirksamkeit. Unterstützt wird dies durch die Möglichkeit, die Interaktionsgeschwindigkeit durch Kopfbewegungen zu beschleunigen, sofern diese für die Person ausführbar sind. Darüber hinaus bietet die Schnittstelle effektive Maßnahmen zur Fehlerkorrektur, wie z. B. die Möglichkeit, leere Eingaben zu tätigen oder eine bereits gesetzte Scan-Linie nachträglich durch Kopfbewegungen zu verschieben.

Wie das Item Scanning lenkt auch das Cartesian Scanning nur wenig vom Inhalt der virtuellen Umgebung ab, sodass die Teilnehmenden die Aufgaben zielgerichtet bearbeiten konnten. Beim Cartesian Scanning fühlten sich die Nutzenden tendenziell etwas stärker in die Erfahrung involviert als beim Item Scanning. Sowohl die Usability als auch die UX wurden beim Cartesian Scanning positiv bewertet. In Bezug auf die UX wurde beim Cartesian Scanning insbesondere der Faktor Durchschaubarkeit positiv bewertet. Die Ergebnisse zeigen, dass die Schnittstelle als angenehm, übersichtlich, leicht erlernbar und aufgeräumt empfunden wurde.

### Schwächen der Schnittstellen

Eine zentrale Schwäche des Item Scanning liegt in der geringen Kontrolle der Nutzenden über das Scanning-Verfahren. Weder die Interaktionsgeschwindigkeit noch die Scan-Reihenfolge können aktiv beeinflusst werden, was zu einer geringeren Bewertung der Effizienz und Steuerbarkeit führt. Diese Einschränkungen spiegeln sich in der Bewertung der UX wider, wo die Schnittstelle als langsam, ineffizient und nicht erwartungskonform

wahrgenommen wurde. Darüber hinaus wurde die Scan-Reihenfolge als nicht nachvollziehbar empfunden, was zu Verwirrung führte und damit zu der niedrigen Bewertung der Steuerbarkeit beitrug. Die häufigste Fehlerursache beim Item Scanning waren Timing-Probleme. Während die Robustheit der Schnittstelle an sich positiv bewertet wurde, schränken diese Fehler die Zuverlässigkeit in der Nutzung ein. Auch die Bewertung der Usability zeigte Verbesserungspotential. Zudem traten beim Item Scanning tendenziell mehr Symptome von Motion Sickness auf. Diese betrafen vor allem angestrengte Augen, Müdigkeit und Kopfdruck, was auf eine mögliche Überforderung der visuellen Wahrnehmung hinweist. Des Weiteren wurde von den Teilnehmenden bemängelt, dass das Navigationsmenü zu hoch platziert ist, was als störend empfunden wurde.

Das Cartesian Scanning zeigte insbesondere Schwächen hinsichtlich der Interaktionsgeschwindigkeit und der Robustheit. Die Interaktionsgeschwindigkeit war deutlich langsamer als beim Item Scanning. Zudem besteht eine starke Abhängigkeit zwischen der Position der Zielelemente im Sichtfeld und der Interaktionsgeschwindigkeit. Diese Aspekte führten zu einer niedrigen Bewertung der Effizienz. Hinsichtlich der Robustheit zeigte sich, dass das Cartesian Scanning im Vergleich zum Item Scanning anfälliger für das Auftreten von Fehlern war und auch die Fehlerursachen vielfältiger waren. Auch beim Cartesian Scanning traten Symptome von Motion Sickness auf, insbesondere angestrengte Augen und Ermüdung, was auf eine visuelle Belastung durch das Verfahren hinweist. Darüber hinaus wurde festgestellt, dass die voreingestellte Scan-Rate etwas zu langsam war, was wahrscheinlich die wahrgenommene Ineffizienz verstärkte. Auch die Zeit, die zum Drücken des Schalters für den Wechsel des Interaktionsmodus benötigt wurde, wurde als zu lang empfunden.

### **Ansätze zur Verbesserung der Schnittstellen**

Für das Item Scanning ist die Optimierung der Animation und der Wartezeiten ein zentraler Ansatz, um Timing-Probleme zu reduzieren und die Robustheit des Verfahrens weiter zu erhöhen. Eine längere Verweildauer auf dem zuvor ausgewählten Element im Navigationsmenü sowie eine etwas längere Wartezeit auf dem ersten Element in der Scan-Reihenfolge könnten die Nutzenden dabei unterstützen, präzisere Selektionen vorzunehmen. Darüber hinaus könnte die Möglichkeit, die Scan-Reihenfolge bereits bei der Erstellung einer Szene festzulegen, die Scan-Reihenfolge nachvollziehbarer erscheinen lassen und damit die Bewertung der Steuerbarkeit verbessern. Eine weitere Optimierung besteht darin, die Scan Rate konfigurierbar zu machen. Dies würde es den Nutzenden ermöglichen, aktiv Einfluss auf den Prozess und die Interaktionsgeschwindigkeit zu nehmen, was wiederum nicht nur die Effizienz, sondern auch die Steuerbarkeit verbessern könnte. Auch eine neue Positionierung der Menüs könnte die Interaktion erleichtern und die visuelle Belastung reduzieren.

Beim Cartesian Scanning liegt der Schwerpunkt der Verbesserungen insbesondere auf der Erhöhung der Robustheit sowie der Interaktionsgeschwindigkeit und damit der Effizienz. Um die Fehleranfälligkeit zu reduzieren, sollte eine Neupositionierung des Hauptmenüs vorgenommen werden, um Fehlselektionen zu vermeiden. Eine weitere Verbesserungsoption ist die Visualisierung des Interaktionsmodus auch dann, wenn der Scan nicht aktiv ist und die Scan-Linien somit nicht sichtbar sind. Um die Interaktionsgeschwindigkeit zu verbessern, sollte auch hier die Möglichkeit geschaffen werden, die Scan Rate individuell anzupassen. Dies könnte sich gleichzeitig positiv auf die Wahrnehmung der Steuerbarkeit auswirken. Eine weitere Idee zur Beschleunigung des Prozesses liegt in der gezielten Anpassung der Scan-Bewegungen. Da die meisten Interaktionen in der Mitte des Sichtfeldes stattfinden, könnte es sinnvoll sein, den Scan so zu gestalten, dass er diese Bereiche schneller erreicht.

### **Eignung der Schnittstellen für den Einsatz in VR**

Die Analyse der Ergebnisse zeigt, dass sowohl das Item Scanning als auch das Cartesian Scanning Vor- und Nachteile aufweisen und in ihrer aktuellen Implementierung noch Verbesserungspotenzial bieten. Das Cartesian Scanning wurde im Rahmen der Evaluation hinsichtlich der subjektiven Bewertungen in den Fragebögen insgesamt etwas besser und einheitlicher bewertet, während das Item Scanning hinsichtlich der Interaktionsgeschwindigkeit und Fehlerhäufigkeit bessere objektive Ergebnisse aufweist. Die subjektiven Bewertungen beim Item Scanning wiesen zudem häufig größere Streuungen auf. Dies deutet darauf hin, dass die Akzeptanz des Cartesian Scanning im Durchschnitt höher ist, während die Wahrnehmung des Item Scanning stärker von anderen Einflussfaktoren wie etwa persönlichen Präferenzen abhängt.

Trotz der besseren Gesamtbewertung des Cartesian Scanning offenbart insbesondere das freie Feedback, dass einige Teilnehmende das Item Scanning deutlich bevorzugten. Auch dies unterstreicht, dass die Wahrnehmung und Akzeptanz der Schnittstellen von persönlichen Präferenzen abhängig ist.

Im derzeitigen Zustand der Implementierung lässt sich daher zwar feststellen, dass das Cartesian Scanning etwas besser angenommen wird, es ergibt sich jedoch keine klare Präferenz für eine der Schnittstellen im Hinblick auf ihre Eignung für den Einsatz in VR. Vielmehr deuten die Ergebnisse darauf hin, dass beide Ansätze Potenzial für den Einsatz in virtuellen Umgebungen bieten und es sich lohnt, ihre Entwicklung und Optimierung weiter voranzutreiben. Durch gezielte Verbesserungen könnten beide Schnittstellen stärker auf die Bedürfnisse unterschiedlicher Nutzenden abgestimmt werden, was ihre Einsatzmöglichkeiten in VR-Anwendungen weiter steigern würde.



### **Ableitung von Empfehlungen für die Entwicklung binärer Interaktionsschnittstellen**

Die durchgeführte Evaluation liefert eine Grundlage, um Empfehlungen für die Entwicklung binärer Interaktionsschnittstellen auszusprechen. Es konnten sowohl Stärken als auch Schwächen der beiden getesteten Schnittstellen identifiziert werden. Darüber hinaus bieten die Ergebnisse durch die Kombination von technischen Daten und subjektivem Nutzerfeedback einen umfassenden Einblick in die Nutzererfahrung und Interaktionsqualität der Schnittstellen. Dennoch müssen die Limitationen der Evaluation berücksichtigt werden. Die Evaluation liefert zwar relevante Hinweise, die Übertragbarkeit auf die Zielgruppe ist jedoch begrenzt. Die Ergebnisse bieten vor allem Ansätze für Empfehlungen, die in weiterführenden zielgruppenspezifischen Untersuchungen validiert und präzisiert werden sollten.

Aus den Ergebnissen können folgende Empfehlungen abgeleitet werden:

- **Scanning-Verfahren**

Gängige Scanning-Verfahren können auf den Einsatz in VR übertragen werden. Sowohl das Automatic Item Scanning als auch das Continuous Cartesian Scanning eignen sich für den Einsatz in VR.

- **Anpassbarkeit und Flexibilität**

Es ist sinnvoll, verschiedene Scanning-Verfahren zu implementieren, die Nutzende nach ihren individuellen Präferenzen auswählen können. Es sollte eine Option zur individuellen Anpassung der Scan-Rate angeboten werden.

- **Klare Visualisierung und Feedback**

Interaktionsprozesse sollten verständlich durch visuelle und akustische Hinweise dargestellt werden, die die Scan-Reihenfolge, aktive Interaktionsmodi oder andere relevante Zustände verdeutlichen.

- **Verständlichkeit**

Bei Verwendung eines Item Scanning-Verfahrens sollte die Scan-Reihenfolge nachvollziehbar und erwartungskonform gestaltet sein.

- **Fehlerkorrekturmechanismen**

Es sollten Möglichkeiten zur schnellen und intuitiven Fehlerkorrektur implementiert werden, wie z. B. Abbruchfunktionen oder das Zurücksetzen des Scannings, um die Robustheit zu erhöhen.

- **Komfort**

Nicht notwendige visuelle oder kognitive Belastungen sollten so weit wie möglich vermieden werden, um das Auftreten von Motion Sickness Symptomen zu minimieren.

- **Positionierung von interaktiven Elementen**

UI- und Interaktionselemente sollten so platziert werden, dass sie komfortabel erreichbar sind. Interaktionselemente sollten vorzugsweise in der Mitte des Sichtfeldes platziert werden.

## 7 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde untersucht, wie binäre Interaktionsschnittstellen für VR-Anwendungen gestaltet und im Rahmen des Tools PaneoVR implementiert werden können, um die Zugänglichkeit für Menschen mit motorischen Einschränkungen zu verbessern. Die Untersuchung adressierte systematisch die in **Kapitel 1** aufgestellten Forschungsfragen (FF1-FF4), welche anhand einer Konzeption, Umsetzung und Evaluation bearbeitet wurden.

Zur Beantwortung der ersten Forschungsfrage (FF1), wie binäre Interaktionen in Virtual Reality gestaltet werden können, wurde eine strukturierte Konzeption durchgeführt. Dabei wurde zunächst der Design Space definiert, indem relevante Interaktionsaufgaben und -komponenten sowie Parameter zur Einordnung von Gestaltungsoptionen identifiziert wurden. Anschließend wurden mögliche Gestaltungsoptionen detailliert ausgearbeitet und eingeordnet, woraus zwei finale Ansätze abgeleitet wurden. Diese basieren auf den etablierten Scanning-Verfahren Automatic Item Scanning und Continuous Cartesian Scanning, die für den Einsatz in VR angepasst und erweitert wurden.

Die prototypische Implementierung der entwickelten Konzepte ermöglichte die Untersuchung der zweiten Forschungsfrage (FF2), welche die Herausforderungen bei der Implementierung binärer Interaktionsschnittstellen sowie mögliche Lösungsansätze adressiert. Im Rahmen der Implementierung zeigten sich für beide Ansätze spezifische technische Herausforderungen. Beim Item Scanning stellte insbesondere die dynamische Anpassung des Scannings an unterschiedliche Interaktionsebenen eine Herausforderung dar. Ebenso erforderte die Beschränkung des Scannings auf das Sichtfeld der Nutzenden die Entwicklung eines ressourcenschonenden Verfahrens, um die Performance der VR-Anwendung nicht signifikant zu beeinträchtigen. Das Cartesian Scanning erwies sich als technisch aufwendiger und anspruchsvoller, da die korrekte Berechnung der Schnittpunkte der Scan-Linien sowie die Festlegung eines angemessenen Toleranzbereichs für die Interaktion schwerfällig waren.

Zur Beantwortung der dritten Forschungsfrage (FF3) - der Untersuchung von Unterschieden zwischen den Ansätzen hinsichtlich Effizienz, Robustheit, Usability und User Experience - wurde eine umfassende Evaluation durchgeführt. Die Ergebnisse zeigen

signifikante Unterschiede zwischen den Schnittstellen. Während das Item Scanning hinsichtlich Interaktionsgeschwindigkeit und Robustheit bessere technische Ergebnisse erzielte, überzeugte das Cartesian Scanning in den subjektiven Bewertungen, insbesondere bei Usability und User Experience. Beide Ansätze weisen spezifische Stärken und Schwächen auf, die sich weitgehend mit den in der Konzeption formulierten Erwartungen deckten. Auffällig war die starke Variation in der Wahrnehmung des Item Scanning zwischen den Teilnehmenden, wohingegen das Cartesian Scanning eine gleichmäßigere Akzeptanz fand. Zum aktuellen Entwicklungsstand lässt sich festhalten, dass das Cartesian Scanning tendenziell besser angenommen wird, sich jedoch keine eindeutige Präferenz für eine der beiden Schnittstellen hinsichtlich ihrer Eignung für VR-Anwendungen ableiten lässt. Vielmehr zeigen die Ergebnisse, dass beide Ansätze Potenzial für den Einsatz in virtuellen Umgebungen besitzen und es sich lohnt, ihre Weiterentwicklung und Optimierung voranzutreiben.

Die vierte Forschungsfrage (FF4) widmete sich der Ableitung allgemeiner Empfehlungen für die Implementierung binärer Interaktionsschnittstellen in VR-Anwendungen. Die durchgeführte Evaluation lieferte eine fundierte Grundlage für die Formulierung solcher Empfehlungen. Zu den zentralen Empfehlungen gehören die Möglichkeit zur individuellen Anpassung der Scan-Rate, die Implementierung effektiver Mechanismen zur Fehlerkorrektur und die ergonomische Platzierung von UI- und Interaktionselementen.

Eine wesentliche Limitation der durchgeführten Evaluation besteht darin, dass sie ausschließlich mit Teilnehmenden ohne motorische Beeinträchtigungen durchgeführt wurde. Die Übertragbarkeit der Ergebnisse auf die Zielgruppe ist daher nur eingeschränkt möglich. Zukünftige Arbeiten sollten sich primär der Umsetzung der identifizierten Verbesserungsansätze widmen, insbesondere der Erhöhung der Interaktionsgeschwindigkeit und Robustheit sowie der Reduktion der visuellen Belastung. Darauf aufbauend sollte eine zielgruppenspezifische Evaluation durchgeführt werden, um die Ergebnisse zu validieren und darauf aufbauend weitere Anpassungen vorzunehmen. Dies ist unerlässlich, um die Nutzbarkeit der entwickelten Schnittstellen für die eigentliche Zielgruppe sicherzustellen.

Darüber hinaus ergeben sich aus den Ergebnissen neue Fragestellungen, die in weiterführenden Arbeiten adressiert werden sollten. Dazu gehören die Untersuchung der Einflussfaktoren auf die subjektive Bewertung des Item Scannings sowie eine detaillierte Analyse der Erlernbarkeit beider Schnittstellen. Des Weiteren sollten die Ursachen für das Auftreten von Motion Sickness untersucht und das Evaluationsdesign so angepasst werden, dass externe Einflussfaktoren minimiert werden. Schließlich könnte eine umfassende Evaluation im Hinblick auf Aspekte wie Presence und Immersion weitere Einblicke in die Wechselwirkungen zwischen den Schnittstellen und dem VR-Erlebnis liefern.

---

Insgesamt zeigt diese Arbeit, dass binäre Interaktionsschnittstellen ein erhebliches Potenzial für barrierearme VR-Anwendungen bieten. Durch gezielte Verbesserungen, zielgruppenspezifische Validierung und iterative Weiterentwicklung könnten die entwickelten Ansätze einen wichtigen Beitrag zur Gestaltung inklusiver virtueller Umgebungen leisten und die Zugänglichkeit für Menschen mit motorischen Beeinträchtigungen nachhaltig verbessern.

## Literaturverzeichnis

Philip T. Kortum Aaron Bangor and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24 (6):574–594, 2008. 10.1080/10447310802205776. URL <https://doi.org/10.1080/10447310802205776>. [VII, 21, 22, 26, 107]

Dmitry Alexandrovsky, Susanne Putze, Valentin Schwind, Elisa D. Mekler, Jan David Smeddinck, Denise Kahl, Antonio Krüger, and Rainer Malaka. Evaluating user experiences in mixed reality. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380959. 10.1145/3411763.3441337. URL <https://doi.org/10.1145/3411763.3441337>. [22]

Apple Inc. Einführung in die Funktionen der Bedienungshilfen auf dem iPhone, 2024. URL <https://support.apple.com/de-de/guide/iphone/iph3e2e4367/ios>. Accessed: 28.06.2024. [1, 36]

Woodrow Barfield, David Zeltzer, Thomas Sheridan, and Mel Slater. *Presence and performance within virtual environments*, page 473–513. Oxford University Press, Inc., USA, 1995. ISBN 0195075552. [7]

Leif P. Berg and Judy M. Vance. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality*, 21(1):1–17, March 2017. ISSN 1434-9957. 10.1007/s10055-016-0293-9. URL <https://doi.org/10.1007/s10055-016-0293-9>. [6]

Pauline Bimberg, Tim Weissker, and Alexander Kulik. On the usage of the simulator sickness questionnaire for virtual reality research. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 464–467, 2020. 10.1109/VRW50115.2020.00098. [11]

Susie Blackstien-Adler, Fraser Shein, Janet Quintal, Shae Birch, and Patrice L. Tamar Weiss. Mouse manipulation through single-switch scanning. *Assistive Technology*, 16

- (1):28–42, 2004. 10.1080/10400435.2004.10132072. URL <https://doi.org/10.1080/10400435.2004.10132072>. PMID: 15359467. [20]
- Doug A. Bowman and Ryan P. McMahan. Virtual reality: How much immersion is enough? *Computer*, 40(7):36–43, 2007. 10.1109/MC.2007.257. [6]
- John Brooke. SUS - A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194), 1996. [21, 73]
- Chris Nolet. Quick Outline | Particles/Effects | Unity Asset Store, 2022. URL <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488?srsId=AfmB0oq6CrS1BbXrAXyWrK9F2zahK5r5DKs3pDxYBNQNR-1aZTUK0hX7>. Accessed: 11.01.2025. [58]
- Brendan A. Ciccone, Shannon K. T. Bailey, and Joanna E. Lewis. The Next Generation of Virtual Reality: Recommendations for Accessible and Ergonomic Design. *Ergonomics in Design*, 31(2):24–27, 2023. 10.1177/10648046211002578. URL <https://doi.org/10.1177/10648046211002578>. [1]
- Albert M. Cook and Janice M. Polgar. Chapter 6 - Making the Connection: User Inputs for Assistive Technologies. In Albert M. Cook and Janice M. Polgar, editors, *Assistive Technologies (Fourth Edition)*, pages 117–138. Mosby, St. Louis (MO), fourth edition edition, 2015a. ISBN 978-0-323-09631-7. <https://doi.org/10.1016/B978-0-323-09631-7.00006-5>. URL <https://www.sciencedirect.com/science/article/pii/B9780323096317000065>. [18, 19, 20, 29, 30, 36, 37, 38]
- Albert M. Cook and Janice M. Polgar. Chapter 7 - Control Interfaces for Assistive Technologies. In Albert M. Cook and Janice M. Polgar, editors, *Assistive Technologies (Fourth Edition)*, pages 139–170. Mosby, St. Louis (MO), fourth edition edition, 2015b. ISBN 978-0-323-09631-7. <https://doi.org/10.1016/B978-0-323-09631-7.00007-7>. URL <https://www.sciencedirect.com/science/article/pii/B9780323096317000077>. [17]
- Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. Inclusive AR/VR: accessibility barriers for immersive technologies. *Universal Access in the Information Society*, 23(1):59–73, March 2024a. ISSN 1615-5297. 10.1007/s10209-023-00969-0. URL <https://doi.org/10.1007/s10209-023-00969-0>. [14, 15]
- Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. Inclusive augmented and virtual reality: A research agenda. *International Journal*

- of *Human-Computer Interaction*, 40(20):6200–6219, 2024b. 10.1080/10447318.2023.2247614. URL <https://doi.org/10.1080/10447318.2023.2247614>. [16, 25]
- DIN EN ISO 9241-110:2020-10. DIN EN ISO 9241-110:2020-10, Ergonomie der Mensch-System-Interaktion - Teil 110:Interaktionsprinzipien. Din, 3 . [29, 30]
- DIN EN ISO 9241-11:2018-11. DIN EN ISO 9241-11:2018-11, Ergonomie der Mensch-System-Interaktion - Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte. Din, 5 . [20, 21, 29]
- DIN EN ISO 9421-210:2020-03. DIN EN ISO 9421-210:2020-03, Ergonomie der Mensch-System-Interaktion - Teil 210: Menschzentrierte Gestaltung interaktiver Systeme. Din, 3 . [1, 21]
- Matt Dombrowski, Peter A. Smith, Albert Manero, and John Sparkman. Designing inclusive virtual reality experiences. In *Virtual, Augmented and Mixed Reality. Multimodal Interaction: 11th International Conference, VAMR 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part I*, page 33–43, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-21606-1. 10.1007/978-3-030-21607-8\_3. URL [https://doi.org/10.1007/978-3-030-21607-8\\_3](https://doi.org/10.1007/978-3-030-21607-8_3). [2, 14, 15, 25, 37]
- John Dudley, Lulu Yin, Vanja Garaj, and Per Ola Kristensson. Inclusive Immersion: a review of efforts to improve accessibility in virtual reality, augmented reality and the metaverse. *Virtual Reality*, 27(4):2989–3020, December 2023. ISSN 1434-9957. 10.1007/s10055-023-00850-8. URL <https://doi.org/10.1007/s10055-023-00850-8>. [2]
- Ralf Dörner, Wolfgang Broll, Paul Grimm, and Bernhard Jung, editors. *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer, Berlin, Heidelberg, 2019. ISBN 978-3-662-58860-4 978-3-662-58861-1. 10.1007/978-3-662-58861-1. URL <http://link.springer.com/10.1007/978-3-662-58861-1>. [8, 9, 29]
- Yasin Farmani and Robert J. Teather. Evaluating discrete viewpoint control to reduce cybersickness in virtual reality. *Virtual Reality*, 24(4):645–664, December 2020. ISSN 1434-9957. 10.1007/s10055-020-00425-x. URL <https://doi.org/10.1007/s10055-020-00425-x>. [33, 42, 47]
- Åsa Fast-Berglund, Liang Gong, and Dan Li. Testing and validating Extended Reality (xR) technologies in manufacturing. *Procedia Manufacturing*, 25:31–38, 2018.



- ISSN 2351-9789. <https://doi.org/10.1016/j.promfg.2018.06.054>. URL <https://www.sciencedirect.com/science/article/pii/S2351978918305730>. [8]
- Eelke Folmer, Fangzhou Liu, and Barrie Ellis. Navigating a 3d avatar using a single switch. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, FDG '11, page 154–160, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308045. 10.1145/2159365.2159386. URL <https://doi.org/10.1145/2159365.2159386>. [31, 32, 37, 40, 41, 42]
- Kathrin Gerling and Katta Spiel. A critical examination of virtual reality technology in the context of the minority body. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. 10.1145/3411764.3445196. URL <https://doi.org/10.1145/3411764.3445196>. [13, 14]
- Carina Girvan. What is a virtual world? Definition and classification. *Educational Technology Research and Development*, 66(5):1087–1100, October 2018. ISSN 1556-6501. 10.1007/s11423-018-9577-y. URL <https://doi.org/10.1007/s11423-018-9577-y>. [7]
- Google. Tips for using switch access - android accessibility help, 2024. URL [https://support.google.com/accessibility/android/answer/6395627?hl=en&ref\\_topic=6151780&sjid=16986929134949672065-EU](https://support.google.com/accessibility/android/answer/6395627?hl=en&ref_topic=6151780&sjid=16986929134949672065-EU). Accessed: 05.08.2024. [36]
- Marc Hassenzahl. User experience (ux): towards an experiential perspective on product quality. In *Proceedings of the 20th Conference on l'Interaction Homme-Machine*, IHM '08, page 11–15, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582856. 10.1145/1512714.1512717. URL <https://doi.org/10.1145/1512714.1512717>. [22]
- Fiona Heilemann, Gottfried Zimmermann, and Patrick Münster. Accessibility Guidelines for VR Games - A Comparison and Synthesis of a Comprehensive Set. *Frontiers in Virtual Reality*, 2, 2021. ISSN 2673-4192. 10.3389/frvir.2021.697504. URL <https://www.frontiersin.org/journals/virtual-reality/articles/10.3389/frvir.2021.697504>. [12, 25, 26]
- Hochschule Bielefeld. ViRDIPA- Über das Projekt, 2023. URL <https://www.hsbi.de/inbvg/projekte/bildungsforschung/virtual-reality-basierte-digital-reusable-learning-objects-in-der-pflegeausbildung/ueber-dasprojekt>. Accessed: 03.01.2025. [23]

- Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Morgan & Claypool, September 2016. ISBN 978-1-970001-13-6. [6, 10, 11, 29, 30]
- Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993. 10.1207/s15327108ijap0303\_3. URL [https://doi.org/10.1207/s15327108ijap0303\\_3](https://doi.org/10.1207/s15327108ijap0303_3). [9, 10, 11, 75, 83]
- Robert S. Kennedy, Julie Drexler, and Robert C. Kennedy. Research in visually induced motion sickness. *Applied Ergonomics*, 41(4):494–503, 2010. ISSN 0003-6870. <https://doi.org/10.1016/j.apergo.2009.11.006>. URL <https://www.sciencedirect.com/science/article/pii/S0003687009001574>. [10]
- Hyun K. Kim, Jaehyun Park, Yeongcheol Choi, and Mungyeong Choe. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. *Applied Ergonomics*, 69:66–73, 2018. ISSN 0003-6870. <https://doi.org/10.1016/j.apergo.2017.12.016>. URL <https://www.sciencedirect.com/science/article/pii/S000368701730282X>. [9]
- Rachel L. Franz, Sasa Junuzovic, and Martez Mott. Nearmi: A framework for designing point of interest techniques for vr users with limited mobility. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383066. 10.1145/3441852.3471230. URL <https://doi.org/10.1145/3441852.3471230>. [16, 28, 29, 33, 47]
- Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and Evaluation of a User Experience Questionnaire. In Andreas Holzinger, editor, *HCI and Usability for Education and Work*, pages 63–76, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-89350-9. [22, 75]
- Joseph J. LaViola. A discussion of cybersickness in virtual environments. *SIGCHI Bull.*, 32(1):47–56, January 2000. ISSN 0736-6906. 10.1145/333329.333344. URL <https://doi.org/10.1145/333329.333344>. [10]
- Michael E. McCauley and Thomas J. Sharkey. Cybersickness: Perception of self-motion in virtual environments. *Presence: Teleoper. Virtual Environ.*, 1(3):311–318, January 1992. ISSN 1054-7460. [9]
- Meta. Meta Quest Virtual Reality Check (VRC) für Meta Quest, 2024. URL

- <https://developers.meta.com/horizon/resources/publish-quest-req/>. Accessed: 13.01.2025. [12]
- Leonard Meyer. Paneovr Prototyp v1.2a Handbuch für Interaktionsfelder, 2025a. URL <https://paneovr.net/handbuch/>. Accessed: 03.01.2025. [24]
- Leonard Meyer. Paneovr, 2025b. URL <https://paneovr.net/>. Accessed: 03.01.2025. [23]
- Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In Hari Das, editor, *Telemanipulator and Telepresence Technologies*, volume 2351, pages 282 – 292. International Society for Optics and Photonics, SPIE, 1995. 10.1117/12.197321. URL <https://doi.org/10.1117/12.197321>. [VII, 7]
- Katsumi Minakata, John Paulin Hansen, I. Scott MacKenzie, Per Bækgaard, and Vijay Rajanna. Pointing by gaze, head, and foot in a head-mounted display. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367097. 10.1145/3317956.3318150. URL <https://doi.org/10.1145/3317956.3318150>. [15]
- Martez Mott, Edward Cutrell, Mar Gonzalez Franco, Christian Holz, Eyal Ofek, Richard Stoakley, and Meredith Ringel Morris. Accessible by design: An opportunity for virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 451–454, 2019. 10.1109/ISMAR-Adjunct.2019.00122. [14]
- Martez Mott, John Tang, Shaun Kane, Edward Cutrell, and Meredith Ringel Morris. “i just went into it assuming that i wouldn’t be able to have the full experience”: Understanding the accessibility of virtual reality for people with limited mobility. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371032. 10.1145/3373625.3416998. URL <https://doi.org/10.1145/3373625.3416998>. [1, 13, 14, 15]
- Charles M. Oman. Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology*, 68(2):294–303, 1990. 10.1139/y90-044. URL <https://cdnsiencepub.com/doi/abs/10.1139/y90-044>. [10]
- Yoshikazu Onuki and Itsuo Kumazawa. Reorient the gazed scene towards the center: Novel virtual turning using head and gaze motions and blink. In *2019 IEEE Conference*

- on *Virtual Reality and 3D User Interfaces (VR)*, pages 1864–1871, 2019. 10.1109/VR.2019.8797722. [33, 42, 47]
- Gary E. Riccio and Thomas A. Stoffregen. An ecological Theory of Motion Sickness and Postural Instability. *Ecological Psychology*, 3(3):195–240, 1991. 10.1207/s15326969eco0303\_2. URL [https://doi.org/10.1207/s15326969eco0303\\_2](https://doi.org/10.1207/s15326969eco0303_2). [10]
- Heidi Koester Richard Simpson and Ed LoPresti. Selecting an appropriate scan rate: The “.65 rule”. *Assistive Technology*, 19(2):51–60, 2007. 10.1080/10400435.2007.10131865. URL <https://doi.org/10.1080/10400435.2007.10131865>. PMID: 17727073. [20]
- Maria V. Sanchez-Vives and Mel Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, April 2005. ISSN 1471-0048. 10.1038/nrn1651. URL <https://doi.org/10.1038/nrn1651>. [6, 7]
- Dimitrios Saredakis, Ancret Szpak, Brandon Birckhead, Hannah A. D. Keage, Albert Rizzo, and Tobias Loetscher. Factors Associated With Virtual Reality Sickness in Head-Mounted Displays: A Systematic Review and Meta-Analysis. *Frontiers in Human Neuroscience*, 14, 2020. ISSN 1662-5161. 10.3389/fnhum.2020.00096. URL <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2020.00096>. [9]
- Andrej Somrak, Iztok Humar, M. Shamim Hossain, Mohammed F. Alhamid, M. Anwar Hossain, and Jože Guna. Estimating VR Sickness and user experience using different HMD technologies: An evaluation study. *Future Generation Computer Systems*, 94: 302–316, 2019. ISSN 0167-739X. <https://doi.org/10.1016/j.future.2018.11.041>. URL <https://www.sciencedirect.com/science/article/pii/S0167739X18325044>. [8, 9, 10]
- Kay M. Stanney, Robert S. Kennedy, and Julie M. Drexler. Cybersickness is Not Simulator Sickness. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 41(2):1138–1142, 1997. 10.1177/107118139704100292. URL <https://doi.org/10.1177/107118139704100292>. [11, 26]
- Constantine E. Steriadis and Philip Constantinou. Designing human-computer interfaces for quadriplegic people. *ACM Trans. Comput.-Hum. Interact.*, 10(2):87–118, June 2003. ISSN 1073-0516. 10.1145/772047.772049. URL <https://doi.org/10.1145/772047.772049>. [19]
- Shari Trewin, Mark Laff, Vicki Hanson, and Anna Cavender. Exploring Visual and Motor Accessibility in Navigating a Virtual World. *ACM Trans. Access. Comput.*, 2

- (2):11:1–11:35, June 2009. ISSN 1936-7228. 10.1145/1530064.1530069. URL <https://doi.org/10.1145/1530064.1530069>. [31]
- Aikaterini Valakou, George Margetis, Stavroula Ntoa, and Constantine Stephanidis. A Framework for Accessibility in XR Environments. In Constantine Stephanidis, Margherita Antona, Stavroula Ntoa, and Gavriel Salvendy, editors, *HCI International 2023 – Late Breaking Posters*, pages 252–263, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-49215-0. [16, 25, 32]
- Kenneth R. Walsh and Suzanne D. Pawlowski. Virtual Reality: A Technology in Need of IS Research. *Communications of the Association for Information Systems*, 8, 2002. ISSN 15293181. 10.17705/1CAIS.00820. URL <https://aisel.aisnet.org/cais/vol8/iss1/20>. [6]
- Ker-Jiun Wang, Quanbo Liu, Yifan Zhao, Caroline Yan Zheng, Soumya Vhasure, Quan-feng Liu, Prakash Thakur, Mingui Sun, and Zhi-Hong Mao. Intelligent wearable virtual reality (vr) gaming controller for people with motor disabilities. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 161–164, 2018. 10.1109/AIVR.2018.00034. [16]
- Debbie Weber and Maryann Demchak. Using assistive technology with individuals with severe disabilities. *Computers in the Schools*, 12(3):43–56, 1996. 10.1300/J025v12n03\\_06. URL [https://doi.org/10.1300/J025v12n03\\_06](https://doi.org/10.1300/J025v12n03_06). [17]
- Johann Wentzel. Bring-Your-Own Input: Context-Aware Multi-Modal Input for More Accessible Virtual Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, pages 1–5, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9422-2. 10.1145/3544549.3577056. URL <https://doi.org/10.1145/3544549.3577056>. [31]
- Bob G. Witmer and Michael J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, June 1998. 10.1162/105474698565686. URL <https://doi.org/10.1162/105474698565686>. \_\_eprint: <https://direct.mit.edu/pvar/article-pdf/7/3/225/1836425/105474698565686.pdf>. [6, 7, 75, 106]
- Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. Ability-based design: Concept, principles and examples. *ACM Trans. Access. Comput.*, 3(3), April 2011. ISSN 1936-7228. 10.1145/1952383.1952384. URL <https://doi.org/10.1145/1952383.1952384>. [2]

- Isabell Wohlgenannt, Alexander Simons, and Stefan Stieglitz. Virtual Reality. *Business & Information Systems Engineering*, 62(5):455–461, October 2020. ISSN 1867-0202. 10.1007/s12599-020-00658-9. URL <https://doi.org/10.1007/s12599-020-00658-9>. [VII, 6, 7, 8]
- Alice Wong, Hannah Gillis, and Ben Peck. Vr accessibility: Survey for people with disabilities. Technical report, 2017. URL [https://www.ben-peck.com/papers/VR\\_Accessibility\\_Survey.pdf](https://www.ben-peck.com/papers/VR_Accessibility_Survey.pdf). [13, 14, 25]
- XR Access - Virtual, Augmented & Mixed Reality for People with Disabilities. XR Access, 2022. URL <https://xraccess.org/about/>. Accessed: 13.01.2025. [12]
- Ilsun Rhiu Yong Min Kim and Myung Hwan Yun. A systematic review of a virtual reality system from the perspective of user experience. *International Journal of Human-Computer Interaction*, 36(10):893–910, 2020. 10.1080/10447318.2019.1699746. URL <https://doi.org/10.1080/10447318.2019.1699746>. [22]
- Bei Yuan, Eelke Folmer, and Frederick C. Harris. Game accessibility: a survey. *Universal Access in the Information Society*, 10(1):81–100, March 2011. ISSN 1615-5297. 10.1007/s10209-010-0189-5. URL <https://doi.org/10.1007/s10209-010-0189-5>. [1, 17]
- F. Zwicky. The morphological approach to discovery, invention, research and construction. In Fritz Zwicky and Albert G. Wilson, editors, *New Methods of Thought and Procedure*, pages 273–297, Berlin, Heidelberg, 1967. Springer Berlin Heidelberg. ISBN 978-3-642-87617-2. [27]

## **Anhang A**

### **Evaluationsprotokoll**

Verfahren:	<b>Evaluationsprotokoll</b>	Testgruppe:
------------	-----------------------------	-------------

**Technisches Szenario**

Gesamtzeit:

**Abschnitt 1**

Item	Anzahl Verfehlungen	Grund Verfehlung
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Kommentare/Ergänzungen:

**Abschnitt 2**

Item	Anzahl Verfehlungen	Grund Verfehlung
F1		
F2		
F3		
F4		
F5		
A1		
A2		
A3		
A4		
A5		

Kommentare/Ergänzungen:

Abbildung A.1: Evaluationsprotokoll Seite 1



Verfahren:	<b>Evaluationsprotokoll</b>	Testgruppe:
<b>Inhaltsbasiertes Szenario</b>		
Gesamtzeit:		
Item	Anzahl Verfehlungen	Grund Verfehlung
Kommentare/Ergänzungen:		

Abbildung A.2: Evaluationsprotokoll Seite 2

## Anhang B

### Screenshots aus dem technischen Evaluationsabschnitt

#### B.1 Item Scanning

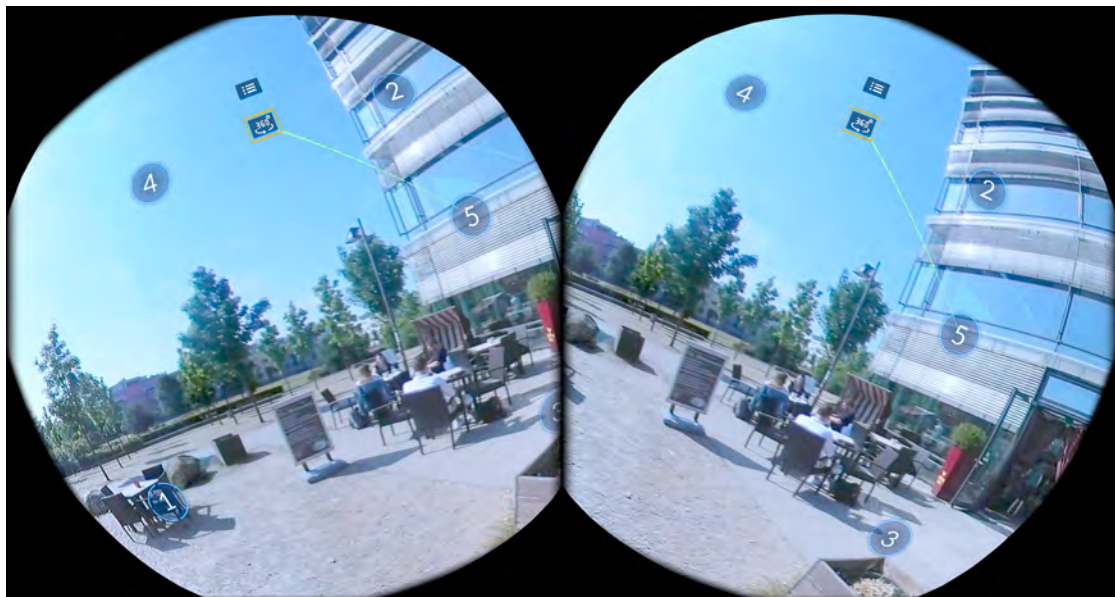


Abbildung B.1: Ausschnitt aus der technischen Evaluation des Item Scannings. Aktives Scanning

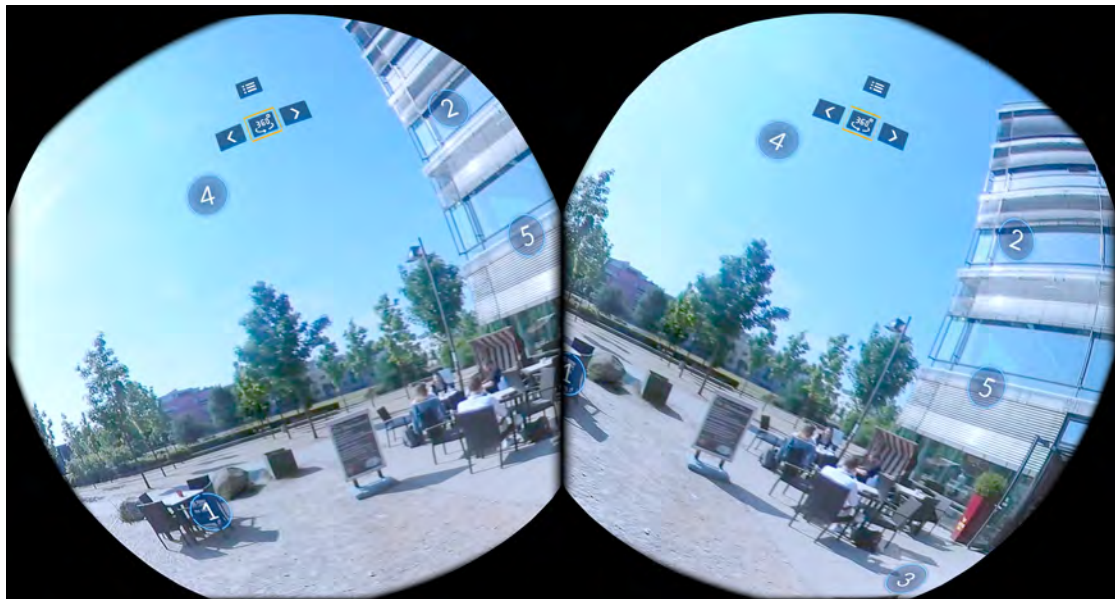


Abbildung B.2: Ausschnitt aus der technischen Evaluation des Item Scannings. Geöffnetes Navigationsmenü

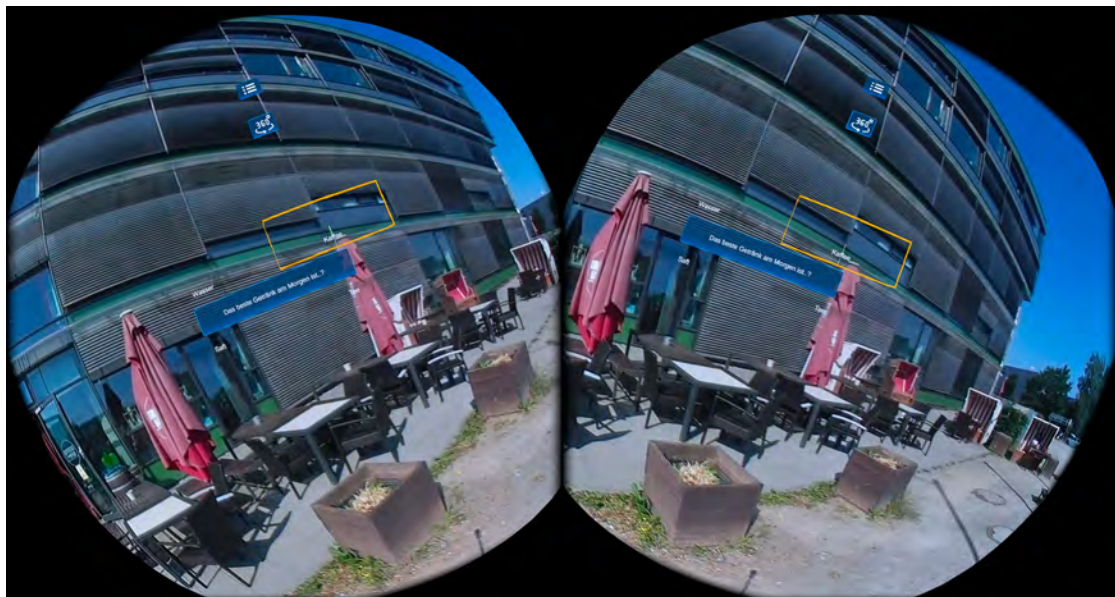


Abbildung B.3: Ausschnitt aus der technischen Evaluation des Item Scannings

## B.2 Cartesian Scanning

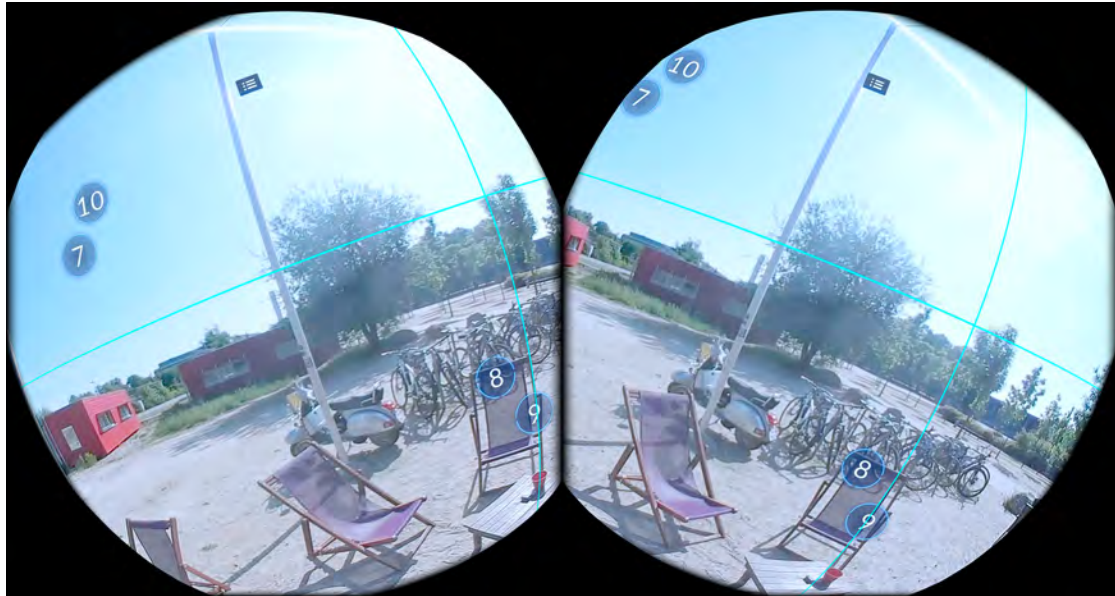


Abbildung B.4: Ausschnitt aus der technischen Evaluation des Cartesian Scannings. Aktiver Navigationsmodus



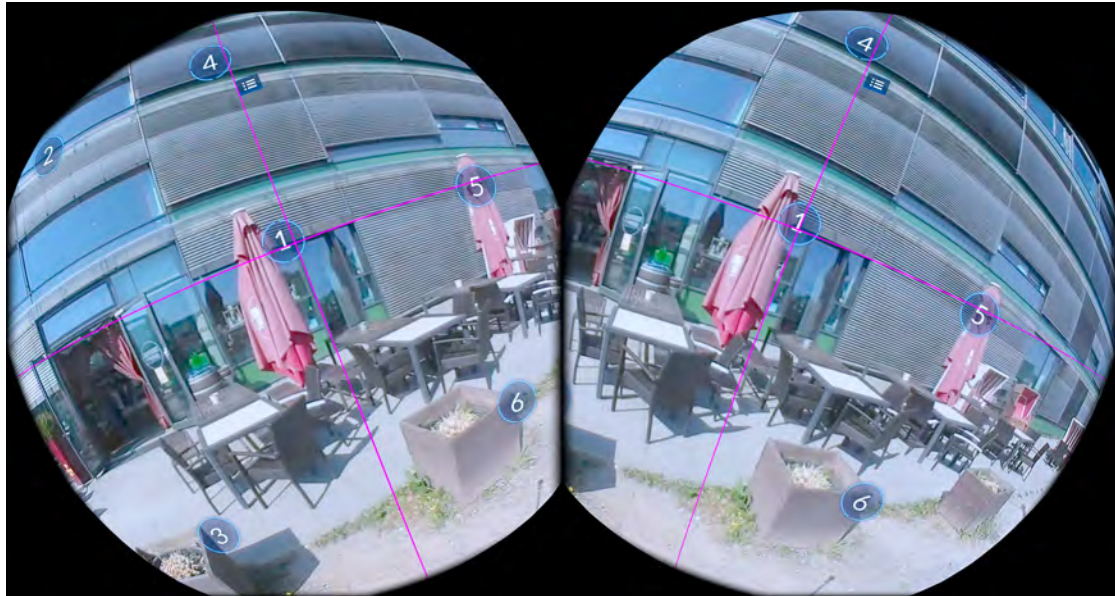


Abbildung B.5: Ausschnitt aus der technischen Evaluation des Cartesian Scannings. Aktiver Selektionsmodus

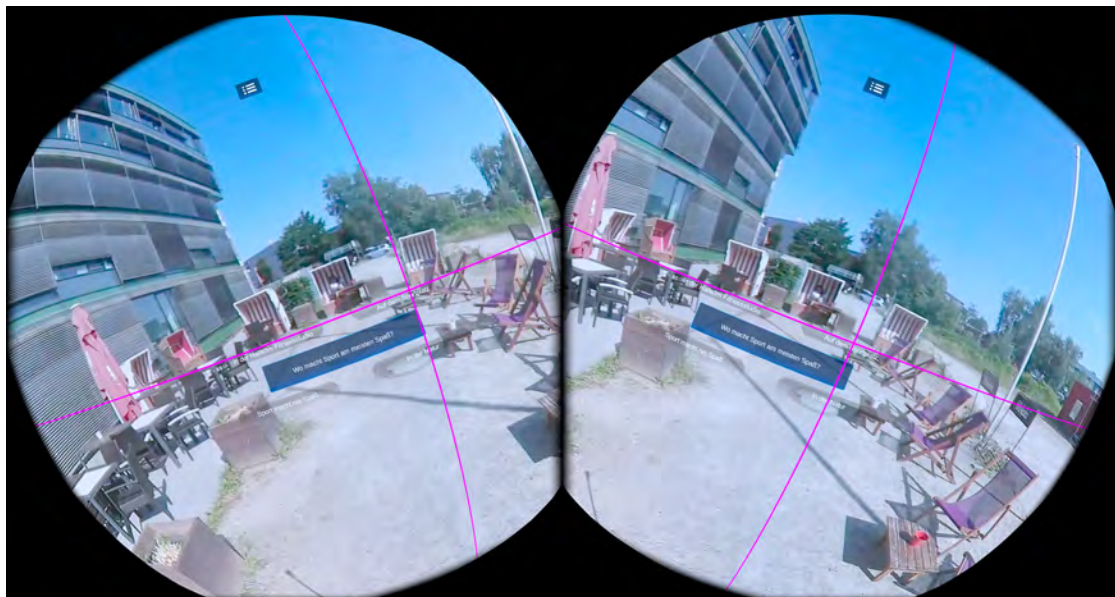


Abbildung B.6: Ausschnitt aus der technischen Evaluation des Cartesian Scannings

## Anhang C

### Screenshots aus dem inhaltsbasierten Evaluationsabschnitt

#### C.1 Szenario Code



Abbildung C.1: Ausschnitt aus dem inhaltsbasierten Szenario Code: Start des Szenarios





Abbildung C.2: Ausschnitt aus dem inhaltsbasierten Szenario Code: Raum Beispiel



Abbildung C.3: Ausschnitt aus dem inhaltsbasierten Szenario CodeGeöffnetes Bild mit versteckter Zahl



Abbildung C.4: Ausschnitt aus dem inhaltsbasierten Szenario Code: Finale Eingabe des Zahlencodes





Abbildung C.5: Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Start des Szenarios

## C.2 Szenario Schlüssel



Abbildung C.6: Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Raum Beispiel

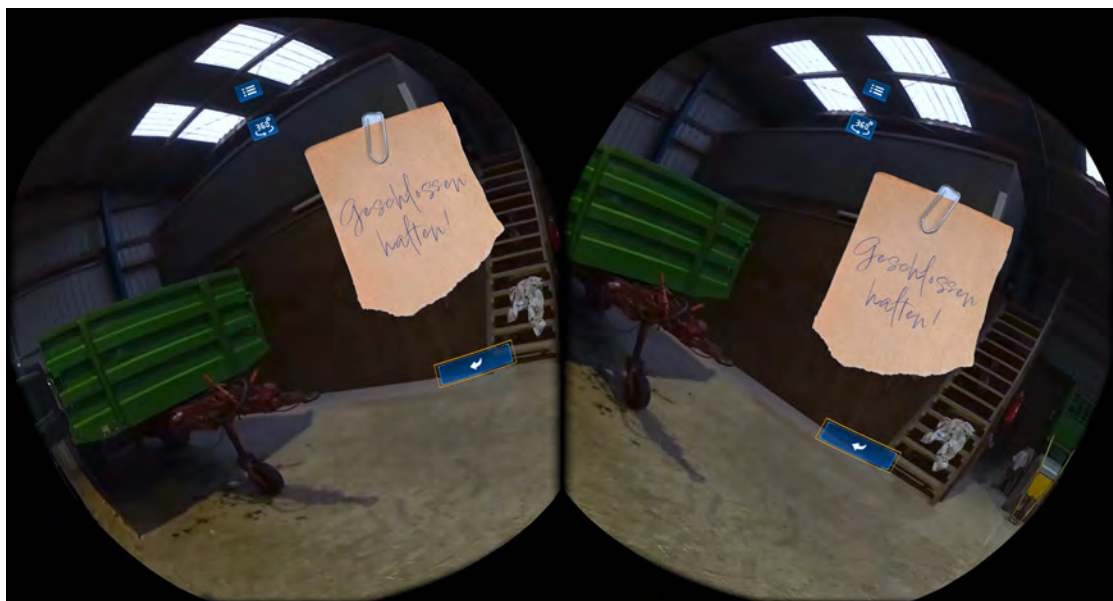


Abbildung C.7: Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Geöffnetes Bild

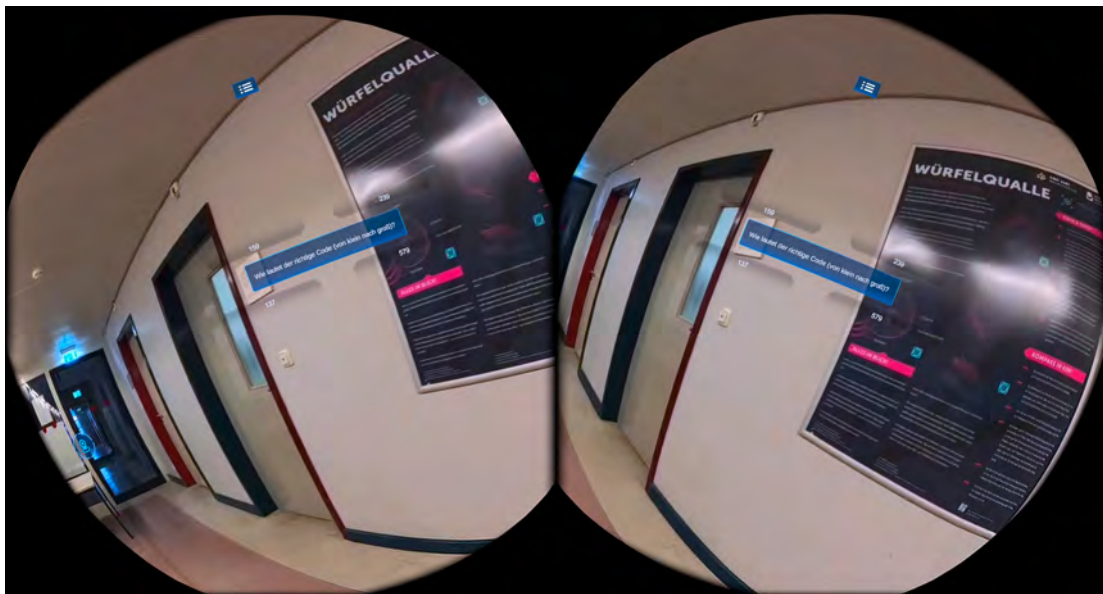


Abbildung C.8: Ausschnitt aus dem inhaltsbasierten Szenario Schlüssel: Finaler Schlüssel

## Anhang D

### Ergebnisse UEQ

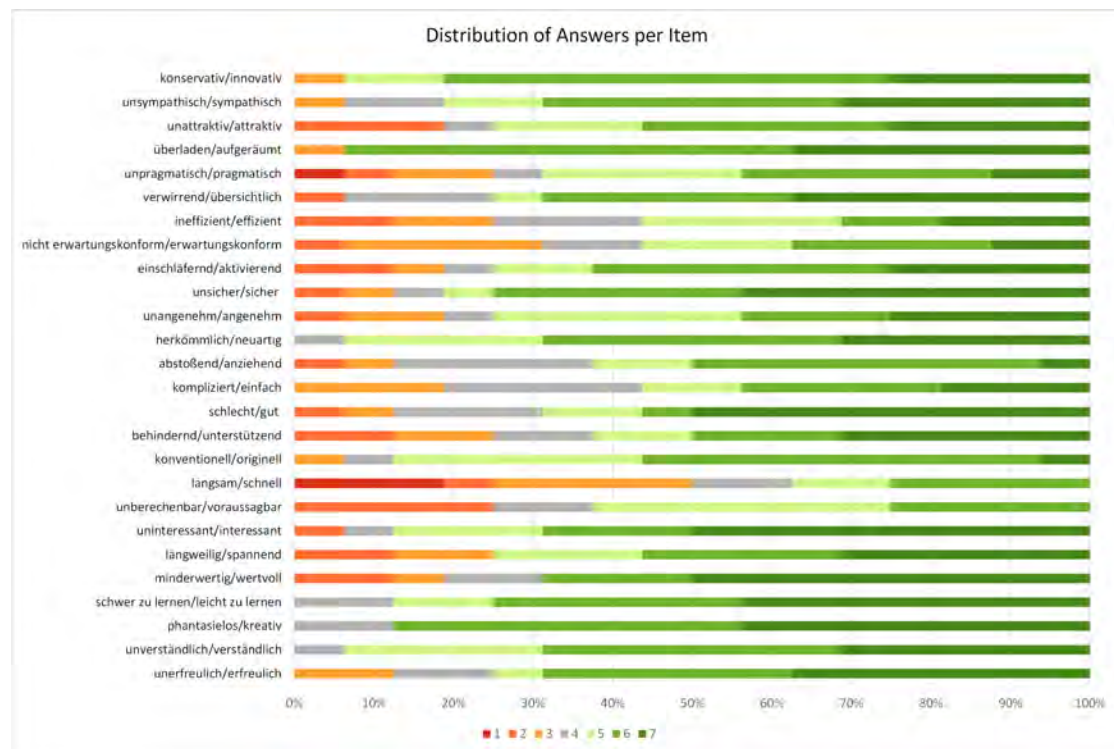


Abbildung D.1: Genaue Verteilung der Bewertungen der Items des UEQ beim Item Scanning

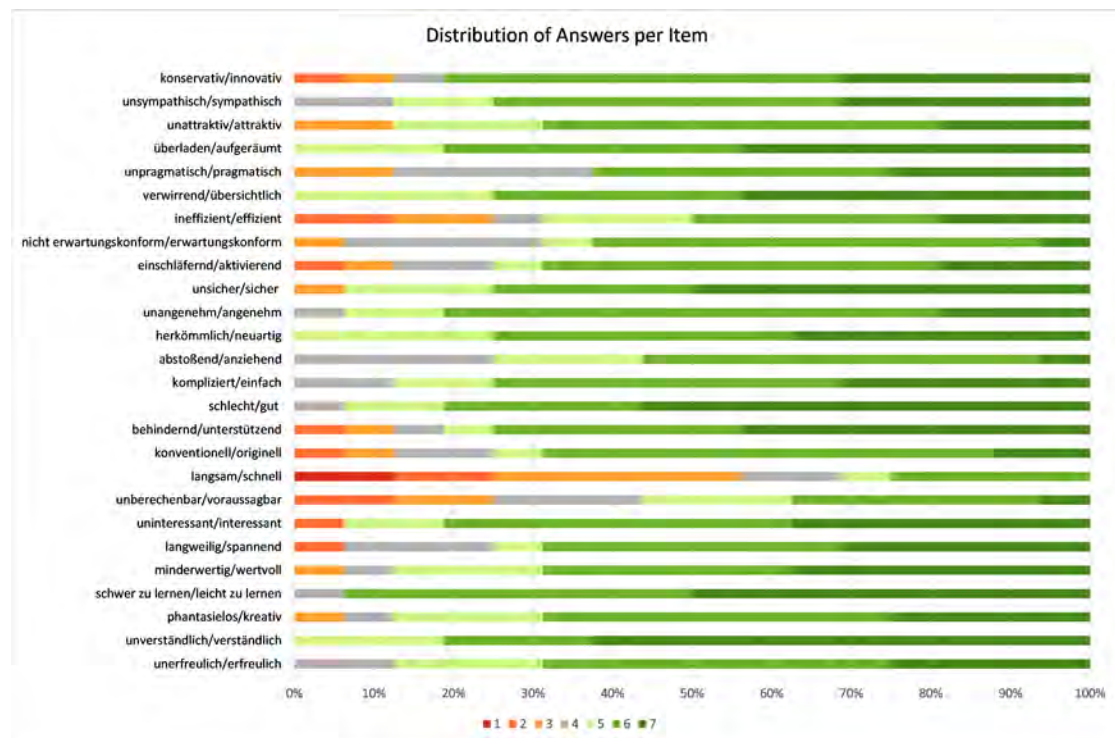


Abbildung D.2: Genaue Verteilung der Bewertungen der Items des UEQ beim Cartesian Scanning

## Anhang E

### Digitaler Anhang

Der digitale Anhang dieser Arbeit umfasst folgendes:

- 01: Eine digitale Version der Arbeit (pdf)
- 02: Zwei .apk-Dateien, die als ausführbare Anwendung auf einer Quest 3 installiert werden können (Eine Datei für das Item Scanning und eine Datei für das Cartesian Scanning)
- 03: Das vollständige Unity-Projekt
- 04: Ergänzende Videos der Durchläufe der verschiedenen Szenarien der Evaluation