



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

ZweitprüferIn:  
Dr. Daniel Sacristán  
Technische Hochschule Lübeck

Lübeck, den 17. Dezember 2024

# Abstract

Kurzfassung auf Englisch.

# Kurzfassung

Kurzfassung auf Deutsch.

# Danksagung

Danke an ...

# Inhaltsverzeichnis

<b>Abstract</b>	<b>I</b>
<b>Kurzfassung</b>	<b>II</b>
<b>Danksagung</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	2
1.3 Aufbau der Arbeit . . . . .	2
<b>2 Stand der Forschung / Stand der Technik</b>	<b>3</b>
2.1 Virtual Reality . . . . .	3
2.1.1 Hardware - HMD und Controller . . . . .	5
2.1.2 Motion Sickness . . . . .	5
2.2 Barrierefreiheit (in VR) . . . . .	7
2.2.1 Probleme . . . . .	8
2.2.2 Lösungsansätze . . . . .	10
2.3 Binäre Interaktionsschnittstellen . . . . .	11
2.3.1 Switches . . . . .	11
2.3.2 Scanning-Verfahren . . . . .	11
2.4 Usability und User Experience . . . . .	14
2.5 PaneoVR . . . . .	14
2.6 Zusammenfassung . . . . .	14
<b>3 Konzeption</b>	<b>15</b>
3.1 Definition des Design Space . . . . .	15
3.2 Ausprägungen der Interaktionsaufgaben und Komponenten . . . . .	18

---

3.3	Einordnung und Bewertung der Interaktionsaufgaben und Komponenten anhand der Parameter . . . . .	21
3.4	Finale Konzepte . . . . .	34
<b>4</b>	<b>Implementierung</b>	<b>36</b>
4.1	Architektur . . . . .	36
4.2	Gemeinsame Komponenten beider Konzepte . . . . .	39
4.3	Scanning Methoden . . . . .	40
4.3.1	Konzept 1 - Item Scanning . . . . .	40
4.3.2	Konzept 2 - Cartesian Scanning . . . . .	44
4.4	Strategien zur Fehlervermeidung und Debugging . . . . .	48
4.5	Limitationen und mögliche Erweiterungen . . . . .	48
4.6	Installation, Nutzerdokumentation . . . . .	50
<b>5</b>	<b>Evaluation</b>	<b>51</b>
5.1	Konzeption der Evaluation . . . . .	51
5.1.1	Technische Evaluation . . . . .	51
5.1.2	Inhaltsbasierte Evaluation . . . . .	54
5.2	Durchführung . . . . .	56
5.3	Ergebnisse . . . . .	60
5.3.1	UEQ . . . . .	60
5.3.2	SUS . . . . .	60
5.3.3	SSQ . . . . .	60
5.3.4	Ablenkung/Presence . . . . .	60
5.3.5	Geschwindigkeit und Zyklen . . . . .	60
5.3.6	Fehler und leere Eingaben . . . . .	60
5.3.7	Korrelationen . . . . .	60
5.3.8	Feedback/Anmerkungen/Beobachtungen . . . . .	60
<b>6</b>	<b>Diskussion</b>	<b>74</b>
6.1	Ausblick . . . . .	74
6.2	Test . . . . .	74
6.2.1	Tabelle . . . . .	74
6.2.2	Bild . . . . .	75
6.2.3	Begriffe und Abkürzungen . . . . .	75
6.2.4	Mathematik . . . . .	76
6.2.5	Code . . . . .	76
	<b>Literaturverzeichnis</b>	<b>77</b>
	<b>Anhang</b>	<b>81</b>

---

<b>A Test-Anhang</b>	<b>82</b>
A.1 Hinweise zum Layout . . . . .	82
A.1.1 Druckseiten . . . . .	82
A.2 Zeitplanung . . . . .	82
<b>B Test-Anhang 2</b>	<b>83</b>
<b>Erklärung</b>	<b>84</b>
<b>Eidesstaatliche Versicherung</b>	<b>84</b>

## Abbildungsverzeichnis

2.1	Reality-virtuality continuum nach Wohlgenannt et al. [2020]	5
4.1	Grundlegender Informationsfluss der Softwarearchitektur	38
5.1	Ergebnisse UEQ Item Scanning	59
5.2	Ergebnisse UEQ Item Scanning	59
5.3	Ergebnisse UEQ Cartesian Scanning	60
5.4	Ergebnisse UEQ Cartesian Scanning	60
5.5	Ergebnisse SUS	61
5.6	Histogramm SUS Item Scanning	61
5.7	Histogramm SUS Cartesian Scanning	62
5.8	Häufigkeit der Symptome	62
5.9	Histogramm SSQ Item Scanning	63
5.10	Histogramm SSQ Cartesian Scanning	63
5.11	Histogramm Nausea Item Scanning	64
5.12	Histogramm Nausea Cartesian Scanning	64
5.13	Histogramm Oculomotor Item Scanning	65
5.14	Histogramm Oculomotor Cartesian Scanning	65
5.15	Histogramm Disorientation Item Scanning	66
5.16	Histogramm Disorientation Cartesian Scanning	66
5.17	Fragen zur Presence/Ablenkung	67
5.18	Interaktionsgeschwindigkeiten	67
5.19	Interaktionsgeschwindigkeiten Tabelle	68
5.20	Benötigte Zeit	68
5.21	Regressionskoeffizienten Lernkurve	68
5.22	Anzahl Fehler Item Technisch	69
5.23	Anzahl Fehler Item Inhalt	69
5.24	Anzahl Fehler Cartesian Technisch	70
5.25	Anzahl Fehler Cartesian Inhalt	70
5.26	Gründe für Fehler Item Scanning	71
5.27	Gründe für Fehler Cartesian Scanning	71



---

5.28 Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit . . . . .	71
5.29 Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit Inhaltsbasierter Abschnitt . . . . .	72
5.30 Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit Technischer Abschnitt . . . . .	73
5.31 Zusammenhang Interaktionsgeschwindigkeit und Usability/UX . . . . .	73
6.1 Test-Bild . . . . .	75

# Tabellenverzeichnis

3.1	Parameterbewertung Selektion . . . . .	22
3.2	Parameterbewertung Navigation . . . . .	25
3.3	Parameterbewertung Display . . . . .	27
3.4	Parameterbewertung Transition . . . . .	30
3.5	Parameterbewertung Bestätigung . . . . .	31
3.6	Parameterbewertung Initialisierung . . . . .	33
5.1	Testgruppen der Evaluation . . . . .	58
6.1	Tabellen-Test . . . . .	75

# 1 Einleitung

Diese L<sup>A</sup>T<sub>E</sub>X-Vorlage soll als Leitlinie und Anhaltspunkt für das Schreiben von Abschlussarbeiten dienen. Zum einen zeigt die Vorlage eine typische Struktur einer Abschlussarbeit auf, an der sich viele Arbeiten orientieren. Zum anderen gibt diese Vorlage gleichzeitig Tipps und Hinweise sowohl was die Strukturierung als auch was mögliche Technologien, insbesondere L<sup>A</sup>T<sub>E</sub>X-Pakete, angeht.

Typischer Umfang der Einleitung: 1-3 Seiten.

## 1.1 Motivation

Der wesentliche Abschnitt der Einleitung ist die Motivation, in der die Arbeit in einen Kontext gesetzt wird. Wichtig ist dabei vor allem, den Leser abzuholen und relativ zügig in das Thema einzuführen. Dazu eignen sich oft auch Bilder oder Skizzen sehr gut. In vielen Fällen kann dies durch ein konkretes Beispiel gelingen.

Die Motivation darf dabei ruhig auf eine größere Vision hin abzielen, auch wenn im Rahmen der Abschlussarbeit dann vielleicht nur ein bestimmter Aspekt oder eine prototypenhafte Umsetzung realisiert werden kann. Es ist aber wichtig, dem Leser den Gesamtkontext zu vermitteln, damit dann die einzelne Leistung besser eingeordnet werden kann.

„The pursuit of Inclusive Immersion is motivated by four key factors: (i) a moral imperative to make technology advances accessible to all; (ii) VR and AR have demonstrated value as assistive and rehabilitative technologies; (iii) the commercial benefits of reaching the broadest user base possible; and (iv) good design typically yields better usability for all. Expanding on this last factor, there are circumstances in which users may not have a disability but may be situationally impaired. As Wobbrock et al. (2011, pg. 7) observe, “Situational impairments arise when aspects of a user’s environment adversely affect his or her ability to perform certain activities”. A design that addresses a nonsituational im-

pairment may thus also improve the usability for those who are situationally impaired.“ ([Dudley et al., 2023, p. 2990])

## 1.2 Zielsetzung

Das Ziel der Arbeit ist die Konzeption und prototypische Implementierung von zwei 1-Bit-Interaktionsschnittstellen für das Tool PaneoVR. Dadurch soll Menschen mit motorischen Einschränkungen die Nutzung der Anwendung ermöglicht werden. Dies impliziert, dass sämtliche erforderliche Interaktionen mit dem System durch eine einzige, einfache Aktion, wie das Drücken eines Knopfes oder eine Pust-Interaktion, ausgeführt werden können. Dadurch sollen die mit PaneoVR erstellten Trainings für Menschen mit motorischen Einschränkungen erlebbar werden. Die Generierung von Trainings mit dem Autorenwerkzeug von PaneoVR wird im Rahmen dieser Arbeit zunächst nicht berücksichtigt. Das Ziel besteht in der Gestaltung eines geeigneten Interfaces sowie der systematischen und detaillierten Konzeption der Interaktionen, um eine möglichst barrierearme Nutzung für die Zielgruppe zu gewährleisten. Im Anschluss erfolgt eine umfassende Evaluation der implementierten Interaktionsformen. Im Rahmen dieser Arbeit soll untersucht werden, welche Interaktionsform für den Kontext VR besser geeignet ist und welche Empfehlungen für eine barrierearme Interaktion für Menschen mit motorischen Einschränkungen ausgesprochen werden können.

## 1.3 Aufbau der Arbeit

In diesem Abschnitt wird schließlich kurz erklärt, wie der weitere Aufbau der Arbeit ist. Welche Kapitel kommen jetzt noch und mit welchem Thema beschäftigen sich diese? Damit soll dem Leser ein kurzer Überblick gegeben werden. Insbesondere bei einer Bachelorarbeit sollte dieser Abschnitt jedoch sehr knapp gefasst werden.

## 2 Stand der Forschung / Stand der Technik

In diesem Kapitel soll der Bezug der Arbeit zum aktuellen Stand der Forschung, bzw. zum Stand der Technik, je nach Ausrichtung und Schwerpunkt der Arbeit, verdeutlicht werden. Dazu werden entsprechende Vorarbeiten oder alternative Ansätze vorgestellt und diskutiert. Ziel ist es, den Ansatzpunkt der Arbeit genauer zu bestimmen und etwaige Entscheidungen im späteren Verlauf des Textes zu fundieren.

Dieses Kapitel kann je nach Thema der Arbeit *Stand der Forschung* oder *Stand der Technik* heißen. In jedem Kapitel ist es wichtig, wie hier geschehen zu Beginn kurz zu erläutern, um was es in diesem Kapitel geht.

Typischer Umfang des Stands der Forschung: 2-4 Seiten BA, 10-15 Seiten MA.

### 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 und Vance als eine Sammlung von Technologien beschrieben, die es Menschen ermöglichen, immersiv eine Welt jenseits der Realität zu erleben. Nach Bowman und 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 Präsenz als drei zentrale Konzepte von VR. Der Begriff der Präsenz 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 Präsenz 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 Präsenz 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](#)).

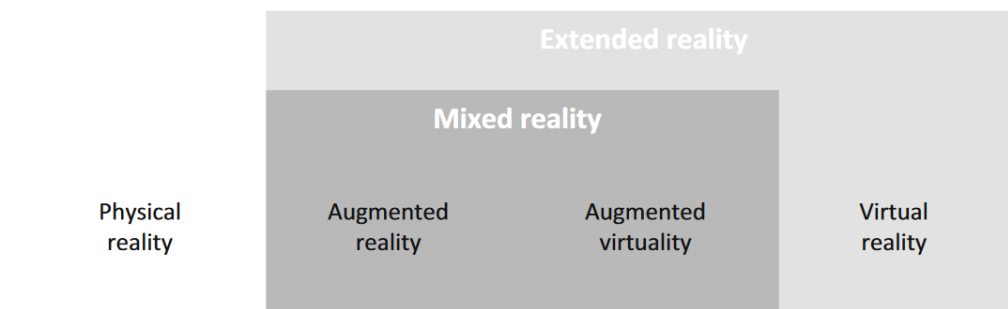


Abbildung 2.1: Reality-virtuality continuum nach Wohlgenannt et al. [2020]

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, S. 32). Wohlgenannt et al. [2020] haben daher das ursprüngliche RVC um diesen Begriff erweitert (vgl. Abbildung 2.1).

### 2.1.1 Hardware - HMD und Controller

### 2.1.2 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", das ursprünglich die Begleiterscheinungen virtueller Umgebungen allgemein beschrieb (McCauley und Sharkey, 1992). In Studien mit HMDs findet zudem der Begriff "VR-Sickness" Verwendung (Cobb et al., 1999; 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, Schweißausbrüche, 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]. Eine der am häufigsten verwendeten subjektiven Messmethoden ist der von Kennedy et al. [1993] entwickelte Simulator Sickness Questionnaire (SSQ). Zu den 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].

## 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 Einschränkung, 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 Einschränkungen bezieht, wird im Folgenden ein kurzer Überblick über den aktuellen Forschungsstand in diesem Bereich gegeben.

- Definition

„Motor impairment is a loss or limitation of function in muscle control or movement or a limitation in mobility. Common causes include arthritis, paralysis, cerebral palsy, or repetitive strain injury. Motor impairment may also include difficulties in speech control and the need to use input devices other than a mouse or keyboard.“ ([Yuan et al., 2011, p. 83])

- Richtlinien/Guidelines:

„So far, guidelines for games rarely consider VR accessibility and few guidelines are exclusively made for VR applications. Many of them are specialized in one specific impairment or device. The way users interact with VR is hardly comparable with other software, so generalized guidelines can not be applied (Cairns et al., 2019a).“ ([Heilemann et al., 2021, p. 3])

– An October 2020 report from XRA (XR Association 2020) provides explicit

guidance on the development of VR and AR applications that are accessible to disabled users.

- Oculus/Meta has also introduced Virtual Reality Check (VRC) guidelines related to accessibility - sind aber sehr beschränkt (z.B. playable without audio, should provide an option to be played with one hand and/or controller, display settings such as brightness and contrast, color blindness options, option to rotate their view without physically moving their head/neck, multiple locomotion styles when possible, provide a setting to enable users to perform all interactions and access information from a fixed position)
- XR Access is a recently established community of university and industry partners focused on addressing the accessibility challenges encountered with VR and AR technologies (Ziel: Inclusive design and accessibility become an unremarkable part of all XR creation)
- Accessibility Guidelines for VR Games - A Comparison and Synthesis of a Comprehensive Set (Heilemann et al.) - -> Zusammenfassung von bestehenden Guidelines mit Fokus auf Games im Allgemeinen, mit einem Absatz speziell für VR Games

### 2.2.1 Probleme

TO DO - Kapiteleinleitung!

#### 1. VR-Gerät einrichten, auf-/absetzen und tragen

Ein grundlegendes Problem für Menschen mit motorischen Einschränkungen 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 der Spielzeitbegrenzung 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 bzgl. des Körpers

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 oft auf Fähigkeiten und Funktionen nicht-behinderter Personen. Dazu gehören beispielsweise die Nutzung im Stehen sowie die Verwendung von Gesten und beiden Hände 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 nicht-behinderter Menschen, was Personen mit Erschöpfung oder chronischen Schmerzen von der Verwendung ausschließt [Wong et al., 2017].

## 3. Controller

Die Nutzung gängiger VR-Controller stellt für Personen mit motorischen Einschränkungen eine erhebliche Herausforderung dar. Aktuelle VR-Systeme sind in erster Linie für Kernnutzende konzipiert, die zwei Hände verwenden, lange stehen und virtuelle Objekte mit Hilfe der Hände manipulieren können. Diese Anforderungen schließen viele Menschen mit motorischen Einschränkungen 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 Einschränkungen Schwierigkeiten, die Tasten auf den Motion-Controllern zu erreichen, zu drücken und gedrückt zu halten, insbesondere wenn gleichzeitig mehrere Tasten betätigt werden müssen [Mott et al., 2020]. Zusätzlich müssen 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].

### 2.2.2 Lösungsansätze

In der Literatur werden bereits einige Ansätze zur Verringerung der genannten Barrieren vorgestellt. So argumentieren [Mott et al. \[2020\]](#) bzgl. der Herausforderungen hinsichtlich der gängigen HMDs, dass Drehknöpfe zum Anpassen des Kopfbandes näher an der Vorder- oder Seite des Headsets positioniert werden sollten, um die Erreichbarkeit zu verbessern. Eine automatische Anpassung des Kopfbandes anstelle einer manuellen Einstellung wäre ebenfalls hilfreich. 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 zugängliche Alternative zu den herkömmlichen Motion-Controllern darstellen. Des Weiteren wird vorgeschlagen, dass Nutzende die Möglichkeit haben sollten, Interaktionsstile oder Steuerungen neu zu konfigurieren, um eine individuellere Nutzung zu ermöglichen. [Dombrowski et al. \[2019\]](#) führen weiter aus, dass weitere potenzielle Verbesserungen die Steuerung von VR-Anwendungen mit alternativen Eingabegeräten wie Schaltern und die Verwendung von Einstellknöpfen an VR-HMDs, die sich automatisch anpassen können, umfassen könnten. Des Weiteren sollten Anwendungen und Eingabegeräte in der Lage sein, ungleichmäßige Eingaben zu tolerieren und sich bemühen, die Absicht des Nutzenden zu interpretieren. Das Design der Anwendung und die Interaktionsformen sollten effizient und komfortabel sein und mit minimaler Ermüdung verwendet werden können. Neben diesen theoretischen Lösungsansätzen wurden in der Forschung bisher nur wenige praktische Untersuchungen für alternative Eingabemethoden 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. \[2018\]](#) hingegen 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. Die Integration von Schaltern jeglicher Form zu Interaktion mit VR-Anwendungen wird in der Literatur bisher kaum betrachtet.

## 2.3 Binäre Interaktionsschnittstellen

TO DO: Kapiteleinleitung

### 2.3.1 Switches

„Binary input is the smallest amount of interaction that can be provided with a switch, because holding down the switch for a certain amount of time may be impossible for a sip and puff device or painful for someone with arthritis.“ ([Yuan et al., 2011, p. 88])

„A switch is an assistive technology device that replaces the use of a mouse, keyboard, controller or joystick which severely motor impaired players may find difficult to use. Switches can be operated by any body part that is able to produce consistent and voluntary movement, and different types of switches can be identified based upon the type of action required to use them (sip and puff, pull, push, or squeeze). Individuals with severe motor impairments may sometimes be able to use only one switch, whereas individuals with less severe motor impairments may be able to use multiple switches.“ ([Yuan et al., 2011, p. 88])

- Schalter/Knöpfe
- Sip and Puff
- Klicken/Geräusche
- noch was?

### 2.3.2 Scanning-Verfahren

Scanning ist eine Eingabemethode, bei der dem Nutzenden eine Auswahl an Selektionsoptionen 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.

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.

### **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 Benutzereingabe vorgenommen, wird das hervorgehobene Element ausgewählt. Andernfalls wandert die Hervorhebung zum nächsten Element. Der wesentliche Vorteil des Automatic Item Scannings besteht in der Möglichkeit, die Interaktionen des Nutzens auf ein Minimum zu reduzieren. Allerdings ist das Verfahren insgesamt relativ langsam. Des Weiteren erfordert es ein hohes Maß an sensorischer und kognitiver Aufmerksamkeit, um die Reihenfolge der Hervorhebung zu beobachten und zu verfolgen.

### **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.

### **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.

### **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.

Der wesentliche Vorteil des Scannings 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. Darüber hinaus stellt die Festlegung der optimalen Scangeschwindigkeit (Scan Rate) eine der größten Herausforderungen beim Scanning dar. Eine zu hohe Scan Rate kann dazu führen, dass Nutzende Schwierigkeiten haben, im richtigen Moment eine Auswahl zu treffen. Eine zu niedrige Scan Rate hingegen verzögert den Eingabeprozess unnötig. Als bewährte Richtlinie zur Optimierung der Scan Rate kann die sogenannte 0,65-Regel herangezogen werden.

## **2.4 Usability und User Experience**

## **2.5 PaneoVR**

## **2.6 Zusammenfassung**

Jedes Kapitel sollte mit einer eigenen Zusammenfassung abschließen (vielleicht ausgenommen dem einleitenden Kapitel). Der einleitende Text des Kapitels und die Zusammenfassung bilden zugleich eine Klammer um das Kapitel und zeigen einen roten Faden im Übergang zwischen den Kapiteln auf.

Das wesentliche bei der Zusammenfassung insbesondere im Kapitel *Stand der Forschung* ist es, das im Kapitel beschriebene in eigenen Worten kurz und prägnant darzustellen und in Bezug zur eigenen Arbeit zu setzen.

Es könnte in der Zusammenfassung zum Beispiel stehen: „Wie X und Y gezeigt haben, ist noch offen, wie... In dieser Arbeit soll diese Frage so und so angegangen werden.“ Oder „Wie gezeigt werden konnte, gibt es derzeit für X noch keine (zufriedenstellende) Lösung...“.



## 3 Konzeption

TO DO Kapiteleinleitung!

Die Konzeption der Interaktionen erfolgt in vier Schritten:

- Definition des Design Space
- Erarbeitung konkreter Ausprägungen der Interaktionsaufgaben und Komponenten
- Bewertung und Einordnung der erarbeiteten Interaktionsaufgaben und Komponenten anhand der definierten Parameter
- Ableitung und Darstellung der finalen Konzepte

Im ersten Schritt erfolgt die Definition des Design Space der Interaktionen. Dazu werden zunächst die für die Erreichung der Zielsetzung der vorliegenden Arbeit erforderlichen Interaktionsaufgaben abgeleitet. Darüber hinaus erfolgt eine Identifikation von Komponenten und Parametern auf Basis des aktuellen Forschungsstands. Im zweiten Schritt erfolgt die Spezifikation konkreter Ausprägungen der Interaktionsaufgaben sowie der Komponenten. Im dritten Schritt erfolgt eine Einordnung und Bewertung der erarbeiteten Optionen hinsichtlich der zuvor definierten Parameter. Im letzten Schritt werden aus den zuvor erarbeiteten Resultaten zwei finale Konzepte abgeleitet, welche anschließend im Rahmen des zu erstellenden Prototyps implementiert werden.

### 3.1 Definition des Design Space

Die Beschreibung des Design Space für binäre Interaktionen in PaneoVR erfolgt auf Basis einer dreiteiligen, hierarchischen Struktur. Die grundlegende Ebene wird durch die Interaktionsaufgaben definiert. In diesem Kontext erfolgt eine Beschreibung der für die Nutzung der Anwendung erforderlichen Interaktionsarten. Die zweite Ebene umfasst

die Komponenten. Diese definieren spezifische Teilaspekte der Interaktion. Die unterste Ebene bilden die Parameter. Parameter beschreiben in diesem Kontext Designüberlegungen, anhand derer die Interaktionstechniken und -ausprägungen evaluiert werden. Die definierten Parameter beeinflussen das gesamte VR-Erlebnis [L. Franz et al., 2021].

### **Interaktionsaufgaben:**

Die Nutzung der mittels PaneoVR erstellten Trainings basiert auf zwei grundlegenden Interaktionsaufgaben. Dies umfasst die Selektion von Interaktionselementen innerhalb der Szene, d. h. von Elementen wie bspw. Wegpunkten oder Dialogfeldern sowie von UI-Elementen innerhalb des Menüs. Die zweite Interaktionsaufgabe besteht in der Navigation innerhalb der Szene. Dies umfasst die Änderung der Blickrichtung innerhalb der Szene. Da PaneoVR keinen dreidimensionalen Navigationsraum bereitstellt, sondern lediglich 360°-Videos präsentiert, ist keine klassische Navigation im 3D-Raum erforderlich. Stattdessen genügt eine Rotation der Szene, um die Blickrichtung zu ändern und somit innerhalb der virtuellen Welt zu navigieren. Unter Berücksichtigung der Tatsache, dass davon auszugehen ist, dass Nutzende Kopfbewegungen nicht frei und ohne Einschränkungen ausführen können, ist die Entwicklung einer alternativen Interaktionsform unabdingbar.

### **Komponenten:**

In Bezug auf die Interaktionen in PaneoVR lassen sich insgesamt vier verschiedene Komponenten identifizieren, deren konkrete Ausgestaltung einen direkten Einfluss auf die Erfahrung nimmt.

Die Komponente Display definiert die Ebene der Szene, auf der das Interaktionsverfahren stattfindet und visualisiert wird. Im Rahmen der Festlegung der Interaktionskomponente ist insbesondere zu bestimmen, ob sich die Interaktion lediglich auf das Sichtfeld des Nutzenden beschränkt. Die Relevanz dieser Komponente zeigt sich insbesondere in Kontexten, in denen die Nutzenden ihren Kopf bewegen können und somit eine Verschiebung des Sichtfelds unabhängig von der binären Interaktionsschnittstelle umsetzen können.

Eine weitere zu berücksichtigende Komponente stellt die Transition dar. Diese ist insbesondere bei der Realisierung einer Navigation von Relevanz. Die Transition definiert den Übergang von der Ausführung der Interaktion zur Navigation hin zur gewählten neuen Blickrichtung. In diesem Zusammenhang wird definiert, wie die Rotation der First-Person-Kamera erfolgt.

Eine weitere Komponente stellt die Bestätigung dar. In diesem Kontext wird definiert, ob eine Eingabe durch den Nutzenden unmittelbar zur Ausführung der Interaktion führt

oder ob diese zusätzlich bestätigt werden muss, bspw. in Form einer weiteren Eingabe.

Die letzte relevante Komponente ist die Initialisierung. Diesbezüglich erfolgt die Definition, ob das gewählte Selektionsverfahren jederzeit aktiv ist oder vom Nutzenden zunächst durch eine 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 [noa, b], Effektivität [noa, b], Erlernbarkeit [noa, a] sowie Robustheit [noa, a] definiert. Darüber hinaus werden die Parameter Geschwindigkeit [Cook and Polgar, 2015], Realismus [Jerald, 2015], Komfort [Jerald, 2015] sowie Visuelle Komplexität [Steriadis and Constantinou, 2003] berücksichtigt.

*Effizienz* beschreibt das Verhältnis zwischen den eingesetzten Ressourcen und dem damit erreichten Ziel. 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.

*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, 2015].

*Robustheit* beschreibt die Fähigkeit eines Systems, auf Fehler von Nutzenden angemessen zu reagieren. 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.

*Geschwindigkeit* beschreibt die notwendige Zeit, die zur versuchten Erreichung eines Ziels aufgewendet wird. Somit wird die Zeitspanne betrachtet, die für die Durchführung

einer Interaktion benötigt wird. Im Rahmen dieser Betrachtung wird insbesondere die während der Interaktion entstehende Wartezeit analysiert.

*Realismus* beschreibt die wahrgenommene Nähe der Interaktion zur Realität. Dies ist von entscheidender Bedeutung, da eine realitätsnahe Interaktion das Gefühl von Presence bei Nutzenden potenziell verstärken kann, was wiederum das Gesamterlebnis in VR beeinflusst.

*Komfort* bezeichnet die Wahrnehmung der Nutzenden bezüglich der Angenehmheit der Interaktion. In diesem Kontext ist insbesondere die Wahrscheinlichkeit für das Auftreten von Motion Sickness von entscheidender Bedeutung. Des Weiteren werden Faktoren wie Ermüdung oder gesteigerte Konzentration in die Bewertung miteinbezogen.

*Visuelle Komplexität* beschreibt das Ausmaß, in welchem eine Interaktion bzw. deren Darstellung visuell fordernd für die Nutzenden ist. Es ist anzustreben, die visuelle Komplexität auf ein Minimum zu reduzieren, um eine Ablenkung oder Störung der nutzenden Personen zu vermeiden [Steriadis and Constantinou, 2003].

## 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, 2015]. Folgende Scanning-Verfahren können für den Einsatz in PaneoVR in Betracht gezogen werden:

- Continuous Cartesian Scanning
- Automatic Item Scanning
- Step Item Scanning mit Dwell Selection

## 2. Navigation:

Im Hinblick auf eine konsistente Gestaltung sollte die Art der Interaktion bei der Navigation einheitlich mit derjenigen bei der Selektion sein. Dadurch soll erreicht werden, dass das System sich für Nutzende möglichst Erwartungskonform verhält. In dieser Konsequenz bedingt sich die Wahl der Ausprägungen gegenseitig. Der Einsatz von Continuous Cartesian Scanning als Selektionsverfahren eröffnet zwei Optionen für die Gestaltung der Navigation:

- **Direkte Navigation:** Wird während des Scannings ein Punkt gewählt, auf dem kein Interaktionselement liegt, erfolgt eine Drehung der First-Person-Kamera, sodass der gewählte Punkt den neuen Mittelpunkt des Sichtfelds bildet.
- **Navigations- und Selektionsmodus:** Es besteht die Möglichkeit, zwischen zwei verschiedenen Modi zu wechseln. Es stehen ein Navigationsmodus sowie ein Selektionsmodus zur Verfügung. Wird im Selektionsmodus ein Punkt im Raum selektiert, auf dem sich kein Interaktionselement befindet, erfolgt keine Aktion und der Scan wird erneut gestartet. Der Navigationsmodus hingegen zeigt das gleiche Verhalten wie zuvor bezüglich der direkten Navigation beschrieben, wobei jedoch eine Rotation der Kamera auch dann möglich ist, wenn der Schnittpunkt auf ein Interaktionselement gesetzt wird. Dies impliziert, dass im Selektionsmodus keine Navigation und im Navigationsmodus keine Selektion möglich ist.

Bei der Auswahl eines Item-Scanning-Verfahrens für die Selektion besteht die Möglichkeit, die Navigation über ein zusätzliches Navigationsmenü zu implementieren. Innerhalb des Menüs wird für jede mögliche Navigationsrichtung ein entsprechender Button bereitgestellt. In Bezug auf die Funktionsweise der Buttons lassen sich wiederum zwei mögliche Ausprägungen unterscheiden.

- **Schrittweise Rotation:** Die Selektion eines Buttons bewirkt eine Rotation um einen festen Winkel. Um eine größere Rotation zu erzielen, ist eine wiederholte Selektion des entsprechenden Buttons erforderlich.
- **Kontinuierliche Rotation:** Die Selektion eines Buttons initiiert eine kontinuierliche Rotation in die vom Nutzenden gewählte Richtung. Durch eine erneute Selektion des entsprechenden Buttons kann die Rotation jederzeit gestoppt werden. Ein vergleichbares Prinzip wurde bspw. von [Folmer et al. \[2011\]](#) entwickelt und evaluiert.

## Komponenten

## 1. Display:

## HIER FEHLT NOCH NE KURZE EINLEITUNG

- Fixiert im Sichtfeld: Das Scanning-Verfahren ist wie ein HUD im Sichtfeld des Nutzenden fixiert. Dies impliziert, dass lediglich Elemente selektiert werden können, die sich auch im Sichtfeld des Nutzenden befinden. Eine Rotation des Kopfes hat demnach unmittelbaren Einfluss auf die Selektion. Im Rahmen eines Cartesian Scannings besteht dadurch die Möglichkeit, die aktuelle Position des Scans durch eine Kopfbewegung zu beeinflussen. Im Rahmen des Item Scannings besteht die Möglichkeit, die Reihenfolge der Elemente zu modifizieren.
- In der gesamten Szene: Das Scanning-Verfahren stellt einen integralen Bestandteil der Szene dar. Dies bedeutet, dass sich der Scan bei einer Rotation nicht aktiv mitdreht und impliziert, dass stets die vollen 360° der Szene abgedeckt werden. Folglich werden auch Interaktionselemente hervorgehoben, die sich außerhalb des Sichtfelds der nutzenden Person befinden. Somit hat eine mögliche Bewegung des Kopfes keinerlei Einfluss auf den Verlauf des Scans.
- Beschränkt auf das Sichtfeld mit Unterbrechung des Scans: Im Rahmen dieser Option findet eine Beschränkung des Scanning-Verfahrens auf das Sichtfeld des Nutzenden statt. Allerdings wird der Scan-Vorgang unterbrochen, sobald der Kopf des Nutzers eine Rotation vollzieht. Mit erneutem Innehalten der Kopfbewegung wird ein neuer Scan initiiert.

## 2. Transition:

Die Rotation der Kamera kann auf unterschiedliche Arten realisiert werden. In der Literatur (vgl. [L. Franz et al. \[2021\]](#), [Farmani and Teather \[2020\]](#), [Onuki and Kumazawa \[2019\]](#)) werden insbesondere die beiden folgenden Optionen diskutiert, die auch in der Praxis Anwendung finden:

- 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. Bestätigung

HIER FEHLT NOCH NE KURZE EINLEITUNG

- Keine Bestätigung: Eine zusätzliche Bestätigung der Interaktion ist nicht erforderlich. Die Reaktion der Anwendung erfolgt unmittelbar nach Betätigung des Schalters.
- Pop-Up-Menü: Bei Betätigung des Schalters wird ein Pop-Up-Menü angezeigt, welches dem Nutzenden die Möglichkeit bietet, eine Bestätigung der zuvor ausgelösten Interaktion vorzunehmen oder diese abubrechen.
- Bestätigungszeit: Wird durch Betätigung des Schalters eine Interaktion initialisiert, ist innerhalb eines vorgegebenen Zeitintervalls eine weitere Betätigung desselben Schalters erforderlich, um die zuvor initiierte Auswahl zu bestätigen und die entsprechende Funktion zu aktivieren. Eine Nicht-Wiederholung der Interaktion führt zur Nicht-Ausführung der Interaktion.

#### 4. Initialisierung

HIER FEHLT NOCH NE KURZE EINLEITUNG

- 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.
- Ein-/Ausschaltbares Scanning: Das Scanning-Verfahren muss durch eine initiale Betätigung des Schalters aktiviert werden. Des Weiteren existiert ein UI-Element in der Szene, welches eine Deaktivierung bzw. eine Pausierung des Scanning-Verfahrens ermöglicht.

### 3.3 Einordnung und Bewertung der Interaktionsaufgaben und Komponenten anhand der Parameter

#### Interaktionsaufgaben

##### 1. Selektion

*Effizienz:* Eine effiziente binäre Interaktionsmethode für PaneoVR zeichnet sich durch

Parameter	Cartesian	Automatic	Step
Effizienz	Hoch	Hoch	Gering-Mittel
Effektivität	Mittel	Hoch	Hoch
Erlernbarkeit	Hoch	Hoch	Hoch
Robustheit	Mittel	Mittel	Hoch
Geschwindigkeit	Mittel	Langsam	Mittel
Realismus	-	-	-
Komfort	Mittel	Mittel	Gering
Visuelle Komplexität	Mittel	Mittel	Gering

Tabelle 3.1: Parameterbewertung Selektion

eine möglichst geringe Anzahl an erforderlichen Aktivierungen des Schalters aus. Beim Cartesian Scanning sind grundsätzlich zwei Aktivierungen erforderlich, um eine gewünschte Interaktion auszuführen. Folglich kann die Effizienz dieser Selektionsmethode als hoch eingestuft werden. Demgegenüber weist das Automatic Item Scanning lediglich eine geringfügig höhere Effizienz auf. Die Selektion eines Elements erfolgt durch einmalige Aktivierung des Schalters. Eine Gruppierung der Elemente, etwa mit dem Ziel einer Erhöhung der Interaktionsgeschwindigkeit, bedingt eine größere Anzahl an Interaktionen, was zu einer leichten Reduzierung der Effizienz führt. Das Step Item Scanning zeigt die geringste Effizienz, da mehrere Aktivierungen erforderlich sind, um ein Element gezielt auszuwählen. Eine Gruppierung der Elemente kann in diesem Fall die Effizienz der Selektionsmethode erhöhen, da insbesondere bei einer großen Anzahl an Interaktionselementen in der Szene die Anzahl der erforderlichen Interaktionen deutlich verringert werden kann.

*Effektivität:* Die Effektivität des Cartesian Scanning wird insgesamt als mittel bewertet. Das Timing spielt eine entscheidende Rolle bei der Auswahl des gewünschten Interaktionselements. Ein Verfehlen des Elements ist insbesondere bei ungenauem Timing eine potenzielle Gefahr. Demgegenüber weist das Item Scanning eine leicht höhere Effektivität auf, da ausschließlich interaktive Elemente selektiert werden können und keine leeren Eingaben möglich sind. Beim Automatic Item Scanning ist jedoch auch ein gutes Timing von entscheidender Bedeutung, um das gewünschte Element nicht zu verpassen. Insgesamt weist das Step Item Scanning die höchste Effektivität auf, da den Nutzenden hier die größte Kontrolle über das Timing und die Auswahl gegeben ist [Cook and Polgar, 2015].

*Erlernbarkeit:* Personen, die bereits Erfahrung mit Schaltersteuerungen an anderen Geräten gesammelt haben, dürften alle beschriebenen Scanning-Verfahren als vertraut wahrnehmen, da sie bspw. in gängigen Betriebssystemen Anwendung finden. Die Verfahren



sind sowohl in den Betriebssystemen iOS und macOS von Apple [Apple, 2024] als auch in Android [Google, 2024] integriert. Daher wird die Erlernbarkeit aller drei Verfahren als hoch bewertet.

*Robustheit:* Der Einsatz von Scanning-Verfahren kann zu Fehlern führen, sofern die Scan Rate nicht den individuellen Bedürfnissen des jeweiligen Nutzenden entspricht. Diesbezüglich ist insbesondere eine zu hohe Scan-Rate problematisch, da den Nutzer:innen keine ausreichende Zeit zur Verfügung steht, um eine präzise Auswahl zu treffen [Cook and Polgar, 2015]. Dies trifft insbesondere auf die Optionen des Cartesian Scanning sowie des Automatic Item Scanning zu. Um dem zuvor beschriebenen Problem entgegenzuwirken, sollte den Nutzenden die Möglichkeit gegeben werden, die Scan Rate individuell einzustellen, um sie somit an die spezifischen Bedürfnisse anzupassen. Demgegenüber bietet das Step Item Scanning den Vorteil, dass die Scan Rate jederzeit von den Nutzenden kontrolliert werden kann, wodurch das Auftreten von Fehlern vermieden werden kann [Cook and Polgar, 2015]. Daher wird die Robustheit dieser Option als am höchsten eingestuft. Des Weiteren können beim Item Scanning Fehler auftreten, wenn die Reihenfolge der Elemente nicht nachvollziehbar ist und ein anderes Interaktionselement hervorgehoben wird, als von den Nutzenden erwartet. Um diesem Umstand entgegenzuwirken, wird empfohlen, die Reihenfolge der Elemente so nachvollziehbar wie möglich zu gestalten sowie eine deutliche Visualisierung zu wählen, die bereits Hinweise auf das jeweils nächste Element liefert. Als weitere Maßnahme zur Erhöhung der Robustheit kann die Implementierung eines Toleranzbereichs für das Cartesian Scanning in Erwägung gezogen werden. Dadurch können geringe Ungenauigkeiten von der Anwendung toleriert und die Intentionen der Nutzenden möglichst exakt interpretiert werden.

*Geschwindigkeit:* Scanning-Verfahren führen im Vergleich zu direkten Interaktionsmethoden in der Regel zu einer langsameren Interaktion [Cook and Polgar, 2015]. Es ist zu berücksichtigen, dass die folgenden Bewertungen lediglich einen Vergleich der verschiedenen Scanning-Methoden darstellen. Auch ein schnelles Scanning-Verfahren weist grundsätzlich eine geringere Geschwindigkeit auf als eine direkte Interaktion. In Bezug auf die Interaktionsgeschwindigkeit lassen sich Unterschiede zwischen Cartesian Scanning und Item Scanning hinsichtlich der Anzahl der Interaktionselemente in einer Szene feststellen. Insbesondere bei einer geringen Anzahl an Interaktionselementen in einer Szene ist die Interaktionsgeschwindigkeit beim Cartesian Scanning vergleichsweise langsam. Die Dauer des Scan-Vorgangs ist insbesondere davon abhängig, an welcher Stelle sich das gesuchte Element innerhalb der Szene befindet. Es kann zu einer gewissen Verzögerung kommen, bis die Scanning-Linie die gewünschte Position des Interaktionselements erreicht. In diesem Fall erweist sich die Selektion mittels Item Scanning als schneller. Bei einer hohen Anzahl an Interaktionselementen ist das Item Scanning hingegen relativ langsam. Die Anzahl der in einer Szene enthaltenen Interaktionselemente wirkt sich direkt auf die Geschwindigkeit des Scans aus. Mit einer größeren Anzahl von Elementen

muss eine entsprechend größere Anzahl von Elementen während des Scans durchlaufen werden, was zu einer Verlangsamung der gesamten Interaktionsgeschwindigkeit führt. 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, 2015]. Das Step Item Scanning kann insgesamt als schneller bewertet werden als das Automatic Item Scanning, da die Nutzenden hier schneller zum Zielobjekt gelangen und überflüssige Wartezeiten umgehen können [Cook and Polgar, 2015].

*Realismus:* In Bezug auf die Selektion ist der Parameter Realismus als unerheblich zu betrachten. Insgesamt weisen Scanning-Verfahren im Vergleich zu direkten Interaktionsmethoden einen geringen Realismus auf. In Bezug auf den Grad des Realismus zeigen sich zwischen den einzelnen Scanning-Verfahren lediglich geringfügige Unterschiede.

*Komfort:* In Hinsicht auf den Komfort der Selektion ist insbesondere die Wahrscheinlichkeit für das Auftreten einer motorischen Ermüdung zu betrachten. Des Weiteren findet die erforderliche Konzentration des Nutzenden zur korrekten Ausführung der Selektion Berücksichtigung. Das Cartesian Scanning erfordert eine geringe Anzahl an Aktivierungen des Schalters durch den Nutzenden. Daher kann das Risiko einer motorischen Ermüdung als relativ gering eingestuft werden. Von entscheidender Bedeutung für das Scanning-Verfahren ist jedoch ein exaktes Timing. Daher wird die erforderliche Konzentration als erhöht angenommen. Beim Automatic Item Scanning sind wenige Aktivierungen des Schalters erforderlich, sodass die Wahrscheinlichkeit für das Auftreten von motorischer Ermüdung als gering bewertet werden kann. Allerdings ist hier eine hohe visuelle Aufmerksamkeit sowie ein exaktes Timing erforderlich, was eine gesteigerte Konzentration erfordert. Beim Step Item Scanning ist die Anzahl der erforderlichen Aktivierungen gegenüber den anderen Scanning-Verfahren um ein Vielfaches höher. Infolgedessen wird der physische Aufwand als hoch bewertet, was zu einer raschen motorischen Ermüdung führen kann. Dafür ist der Aufwand an visueller Aufmerksamkeit seitens des Nutzenden gering und die erforderliche Konzentration niedriger [Cook and Polgar, 2015].

*Visuelle Komplexität:* Die visuelle Komplexität der Szene wird durch den Einsatz eines Scanning-Verfahrens insgesamt erhöht. Die Erhöhung der visuellen Komplexität erfolgt beim Item Scanning durch das kontinuierliche Hervorheben von Interaktionselementen, beim Cartesian Scanning hingegen durch die Bewegung der Scanning-Linie. Das Step Item Scanning weist die geringste visuelle Komplexität auf, da den Nutzenden hier die größte Kontrolle über das Scanning gegeben ist und der Wechsel der Hervorhebungen eigenständig gesteuert werden kann.

## 2. Navigation

Parameter	Cartesian Scanning	Modi	Menü Schritte	Menü Kontin.
Effizienz	Hoch	Hoch	Mittel	Mittel
Effektivität	Mittel	Hoch	Gering	Hoch
Erlernbarkeit	hhh	hhh	hhh	?
Robustheit	Gering	Hoch	Hoch	Hoch
Geschwindigkeit	Schnell	Schnell	Mittel	Mittel
Realismus	Mittel	Mittel	Gering	Mittel
Komfort	Mittel	Mittel	Hoch	Mittel
Visuelle Komplexität	Gering	Gering-Mittel	Gering-Mittel	Gering-Mittel

Tabelle 3.2: Parameterbewertung Navigation

*Effizienz:* Es kann festgehalten werden, dass die Optionen mit Moduswechsel oder zusätzlichem Menü eine geringere Effizienz aufweisen als die Option ohne. Dies ist darauf zurückzuführen, dass in diesen Fällen jeweils ein Zwischenschritt erforderlich ist, um die Navigation auszuführen. Die Menüoption mit festen Rotationsschritten erweist sich als zunehmend ineffizient, je größer der gewünschte Rotationswinkel ist. Mit steigendem Rotationswinkel sind entsprechend mehr Interaktionen erforderlich, um diesen zu erreichen. Um die Effizienz dieser Option zu erhöhen, wäre es in der Implementierung erforderlich, das Menü nach einer Interaktion geöffnet zu lassen und eine kurze Pause im Scanning anzusetzen. Dadurch wird es den Nutzenden erleichtert, denselben Button problemlos mehrfach hintereinander auszuwählen, sodass keine zusätzlichen Wartezeiten entstehen. Bei einer gewünschten Rotation um beide Achsen weisen die auf dem Cartesian Scanning basierenden Optionen eine höhere Effizienz auf, da hier eine simultane Rotation um beide Achsen möglich ist.

*Effektivität:* In Bezug auf die Effektivität der Navigation wird betrachtet, in welchem Umfang die Navigation kontrolliert werden kann und inwiefern der Rotationswinkel durch den Nutzenden exakt bestimmt werden kann. Diesbezüglich erweisen sich sowohl der Interaktionsmodus als auch das Navigationsmenü mit kontinuierlicher Rotation als die flexibelsten Optionen, was eine hohe Effektivität gewährleistet [Folmer et al., 2011]. Die Effektivität der Navigation, die auf dem Cartesian Scanning basiert, ist durch die Lage und Anzahl der Interaktionselemente eingeschränkt. Die Effektivität der Navigation ist umso geringer, je mehr Interaktionselemente sich in der Szene befinden. Dies ist darauf zurückzuführen, dass die Punkte im Raum, an denen sich die Elemente befinden, nicht gewählt werden können. Im Falle des Rotationsmenüs, welches eine schrittweise Rotation vorsieht, ist der Rotationswinkel vorgegeben. In der Konsequenz ist lediglich eine begrenzte Anzahl an Rotationswinkeln möglich, wodurch die angestrebte Position ggf. nicht exakt erreicht werden kann [Folmer et al., 2011].

*Erlernbarkeit:* TO DO!

*Robustheit:* Hinsichtlich der Robustheit erweisen sich Optionen, welche eine Abgrenzung zwischen Selektions- und Navigationsaufgabe schaffen, als vorteilhafter. Eine fehlende Abgrenzung kann insbesondere bei einer Vielzahl an Interaktionselementen innerhalb einer Szene oder bei einem ungenauen Timing zu Verwechslungen und ungewollten Reaktionen des Systems führen, was die Robustheit der Interaktion beeinträchtigt. Eine Trennung in Form eines separaten Modus oder Menüs kann derartige Fehler vermeiden. Bei der Implementierung einer Modus-Option ist jedoch von essenzieller Bedeutung, dass der aktuelle Modus für den Nutzenden deutlich erkennbar ist. In diesem Kontext erweist sich eine aussagekräftige Visualisierung als maßgeblich. Die Menü-Optionen bieten die umfassendste Kontrolle über die Navigation, insbesondere hinsichtlich der Rotationsachsen. Dadurch kann einer Fehlerentstehung effektiv vorgebeugt werden.

*Geschwindigkeit:* Die höchste Geschwindigkeit bei der Navigation wird durch die Option auf Grundlage des Cartesian Scanning erzielt, da keine Zwischenschritte in der Interaktion erforderlich sind und der Rotationswinkel beliebig groß gewählt werden kann. Infolgedessen kann die gewünschte Position durch eine einzige Auswahl erreicht werden. Die Geschwindigkeit ist jedoch bei einer hohen Anzahl an Interaktionselementen in einer Szene limitiert, da dies die Möglichkeiten für auswählbare Nutzereingaben einschränkt. Die Option, die zwei Interaktionsaufgaben in verschiedenen Modi auszuführen, bedingt einen zusätzlichen Zwischenschritt in Form eines Modus-Wechsels. Dies resultiert in einer geringfügigen Verlangsamung der Geschwindigkeit. Demgegenüber stellt die Anzahl der Interaktionselemente für die Geschwindigkeit der Interaktion keinen Einfluss dar. Die Vorteile, die sich aus einem beliebig großen Rotationswinkel ergeben, bleiben bestehen. Des Weiteren bieten diese beiden Optionen den Vorteil, dass eine Rotation auf zwei Achsen gleichzeitig mit der Auswahl eines einzigen Punktes innerhalb der Szene möglich ist. Dies resultiert in einer Steigerung der Geschwindigkeit. Die Navigation über ein zusätzliches Navigationsmenü, welches mittels Item Scanning bedient wird, weist eine leicht reduzierte Geschwindigkeit auf. In diesem Fall ist zunächst das Öffnen des entsprechenden Menüs erforderlich, um eine Navigation zu ermöglichen. Bei der schrittweisen Rotation sind ggf. mehrere Interaktionsschritte nacheinander erforderlich, um die gewünschte Position zu erreichen. Dies resultiert in einer Verringerung der allgemeinen Geschwindigkeit. Die kontinuierliche Rotation ist jedoch nicht als wesentlich schneller zu bewerten, da eine zu schnelle Rotation das Risiko von Motion Sickness erhöhen würde. Bei beiden Optionen ist lediglich eine Rotation um eine Achse zur Zeit möglich. Ist eine Rotation um zwei Achsen gewünscht, muss dies in zwei Schritten erfolgen, was sich deutlich auf die Geschwindigkeit auswirkt.

*Realismus:* TO DO

*Komfort:* In Hinblick auf den Komfort der Navigation ist insbesondere die benötigte Konzentration zur Erreichung des Ziels von Relevanz. Die Optionen, welche auf Grundlage des Cartesian Scanning basieren, erfordern insgesamt eine höhere Konzentration als eine Navigation über ein zusätzliches Menü. Dies ist darauf zurückzuführen, dass ein präzises Timing notwendig ist, um den gewünschten Rotationspunkt zu erreichen. Bei der Option ohne Modi ist zusätzlich zu berücksichtigen, dass eine unbeabsichtigte Auswahl eines Interaktionselements möglich ist. Die Menüoption mit vorgegebenen Rotationswinkeln erfordert die geringste Konzentration. Bei der Option der kontinuierlichen Rotation ist darüber hinaus zu erwähnen, dass eine durchgängige Rotation der First-Person-Kamera das Risiko des Auftretens von Motion Sickness erhöht [Farmani and Teather, 2020, Onuki and Kumazawa, 2019].

*Visuelle Komplexität:* Die visuelle Komplexität der Szene wird durch das Hinzufügen eines zusätzlichen Interaktionselements marginal erhöht. Dies trifft sowohl auf die Menüoptionen als auch auf die Modi-Optionen zu. Um die visuelle Komplexität bei den Menüoptionen auf ein Minimum zu reduzieren, ist es empfehlenswert, das Menü als ein- und ausklappbare Komponente zu gestalten. Infolgedessen werden die Buttons zur Rotation lediglich angezeigt, sofern diese benötigt werden. Die visuelle Komplexität der Option mit Modus-Wechsel ist maßgeblich von der Art der Visualisierung des aktuellen Modus abhängig. An dieser Stelle sollte das Ziel darin bestehen, eine Visualisierung zu finden, die die visuelle Komplexität auf ein Minimum reduziert, gleichzeitig aber deutlich und unmissverständlich ist.

### Komponenten:

#### 1. Display

Parameter	Fest am Sichtfeld	Gesamte Szene	Sichtfeld mit Neustart
Effizienz	Hoch	Gering	Hoch
Effektivität	Gering-Mittel	Hoch	Mittel
Erlernbarkeit	ddd	asd	asda
Robustheit	Gering	Mittel	Hoch
Geschwindigkeit	Schnell	Langsam	Schnell
Realismus	-	-	-
Komfort	Gering	Mittel	Mittel
Visuelle Komplexität	Hoch	Gering	Gering

Tabelle 3.3: Parameterbewertung Display

*Effizienz:* Eine Steigerung der Effizienz der Interaktion kann durch eine Beschränkung des Scanning-Verfahrens auf das Sichtfeld erzielt werden. In diesem Kontext kann die

Effizienz durch gezielte Kopfbewegungen, sofern diese für die Nutzenden möglich sind, zusätzlich erhöht werden. Ein Scanning-Verfahren, welches sich über die gesamten  $360^\circ$  der Szene erstreckt, weist eine deutlich geringere Effizienz auf. Eine Unterbrechung des Scanning-Verfahrens bei jeder Kopfbewegung kann zu einer Verringerung der Effizienz führen, da dadurch potenziell unbeabsichtigte Bewegungen zu einer Verzögerung des Verfahrens führen können. In anderen Situationen kann die Effizienz hingegen potenziell gesteigert werden. Beispielsweise bei gewünschter Interaktion mit Zielelementen, die sich am Rand des Sichtfelds befinden und deren Auswahl aufgrund von Timing-Problemen verpasst wurde. In derartigen Fällen erlaubt eine Kopfbewegung, gefolgt von einem unmittelbaren Neustart des Scannings, dass die erneute Wartezeit auf die Wiederholung des Scannings entfällt. Dies kann insbesondere von Vorteil sein, wenn eine Vielzahl von Interaktionselementen in einer Szene vorhanden ist und ein Item Scanning-Verfahren zum Einsatz kommt.

*Effektivität:* Die auf das Sichtfeld beschränkten Optionen sind anfällig für unbeabsichtigte Verschiebungen durch Kopfbewegungen, was die Wahrscheinlichkeit erhöht, dass Interaktionselemente verfehlt werden oder die Genauigkeit von Auswahlen bei der Navigation beeinträchtigt wird. Dies hat eine entsprechend negative Auswirkung auf die Effektivität dieser Optionen. Insbesondere beim Cartesian Scanning ist diese Wahrscheinlichkeit deutlich erhöht. Durch einen Neustart des Scannings kann die Effektivität hingegen gesteigert werden. Bei der  $360^\circ$ -Option ist dieser Umstand generell nicht gegeben. Daher wird die Effektivität dieser Option am höchsten eingeschätzt.

*Erlernbarkeit:* TO DO!

*Robustheit:* Beim  $360^\circ$ -Scanning wird dem Auftreten von Fehlern grundsätzlich dadurch vorgebeugt, dass Kopfbewegungen keinen Einfluss auf das Scanning haben und dadurch keine unbeabsichtigten Verschiebungen auftreten können. Allerdings besteht in diesem Zusammenhang das Risiko unbeabsichtigter Selektionen, sofern sich der Scan außerhalb des Sichtfelds befindet und somit für die Nutzenden nicht wahrnehmbar ist. In diesem Kontext ist eine eindeutige visuelle Rückmeldung essenziell, um zu signalisieren, dass das Scanning aktiv ist und ordnungsgemäß funktioniert. Die Implementierung einer zusätzlichen Bestätigung der Eingabe könnte in diesem Zusammenhang als sinnvoll erachtet werden, um Fehlern vorzubeugen. Ein Neustart des Scannings nach einer Kopfbewegung unterbricht den Scan und wirkt somit präventiv gegen Fehler. Da während der Bewegung keine Eingaben möglich sind, werden unbeabsichtigte Selektionen weitestgehend verhindert. Die höchste Fehlerwahrscheinlichkeit besteht bei einem festen Scanning im Sichtfeld. Selbst minimale Kopfbewegungen wirken sich unmittelbar auf das Scanning aus, sodass eine Verschiebung der Auswahl einfach möglich ist. Dies trifft insbesondere auf die Anwendung des Cartesian Scannings zu. In Bezug auf das Step Item Scanning erweist sich die Option mit Neustart als die effektivste Lösung. Es wird angenommen,

dass eine simultane Bewegung des Kopfes und Betätigung des Schalters für Nutzende schwierig bis unmöglich ist. Wird das Scanning während der Kopfbewegung nicht unterbrochen, besteht die Möglichkeit einer unbeabsichtigten Auswahl eines Elements, sofern die Dauer der Kopfbewegung das Zeitintervall der Auswahl überdauert.

*Geschwindigkeit:* Die Interaktionsgeschwindigkeit ist bei den beiden Optionen, bei denen das Scanning-Verfahren auf das Sichtfeld des Nutzenden beschränkt ist, wesentlich höher als bei einem 360°-Scanning, bei dem eine deutliche Zeitverzögerung auftreten kann. Die schnellste Option stellt das fixierte Verfahren am Sichtfeld dar, da in diesem Fall keinerlei Verzögerungen auftreten. Die Verzögerungen bei der Neustart-Option können jedoch ebenfalls als relativ gering bewertet werden. Für beide Optionen gilt, sofern Kopfbewegungen für den Nutzenden möglich sind, dass diese gezielt eingesetzt werden können, um die Interaktionsgeschwindigkeit zu erhöhen. Durch Kopfbewegungen kann die Scan-Linie beim Cartesian Scanning gezielt in die Nähe der Position des Zielelements gelenkt werden, wodurch eine schnellere Auswahl des Zielelements möglich ist. Im Rahmen des Item Scannings besteht die Möglichkeit, das Zielelement weiter an den Anfang der Scan-Reihenfolge zu bringen, was den gleichen Effekt wie bei der zuvor beschriebenen Vorgehensweise hat.

*Realismus:* Die beschriebenen Optionen unterscheiden sich hinsichtlich des Parameters Realismus kaum. In der Konsequenz wird dieser Parameter als unerheblich für die Komponente erachtet.

*Komfort:* In Bezug auf den Komfort ist insbesondere die notwendige Konzentration des Nutzenden zu berücksichtigen. In diesem Kontext weist die 360°-Option die geringste erforderliche Konzentration auf. Die Nutzenden sind nicht gefordert, sich Gedanken über Bewegungen des Kopfes zu machen, da diese keinen Einfluss auf das Scanning-Verfahren ausüben. Die höchste Konzentration ist erforderlich, wenn das Scanning-Verfahren auf das Sichtfeld beschränkt ist und eine Bewegung des Kopfes einen direkten Einfluss auf das Scanning-Verfahren hat. In diesem Fall können bereits minimale Bewegungen eine Verschiebung des Scannings hervorrufen, wodurch in Momenten geringer Konzentration potenziell Fehler auftreten können. Des Weiteren ist bei der Einschätzung des Komforts auch die mögliche motorische Ermüdung zu berücksichtigen, insbesondere im Hinblick auf das Step Item Scanning. Ein 360°-Scanning kann aufgrund der hohen Anzahl erforderlicher Interaktionen schnell zu Ermüdung führen. Ein Neustart kann ebenfalls zusätzliche Interaktionen nach sich ziehen, die zu einer schnelleren Ermüdung führen können.

*Visuelle Komplexität:* Ein Neustart des Scannings bei einer Kopfbewegung resultiert in einer relativ geringen visuellen Komplexität, da sich dadurch die Bewegung des Kopfes nicht mit der Bewegung des Hervorhebens der Elemente beim Item Scanning bzw.

mit der Bewegung der Scan-Linie beim Cartesian Scanning überlagert. Folglich ist die visuelle Komplexität bei festem Scanning am Sichtfeld ohne Unterbrechung höher. Insbesondere im Hinblick auf das Cartesian Scanning kann diese Option bei Nutzenden Irritation auslösen, da die gleichmäßige und kontinuierliche Bewegung des Scannings durch die Rotation des Kopfes unterbrochen bzw. modifiziert wird. Beim 360°-Scanning erfolgt ebenfalls eine Überlagerung der Bewegungen, jedoch wird die visuelle Komplexität dennoch als vergleichsweise gering eingestuft, da der Ablauf des Scannings stets gleichmäßig und vorhersehbar bleibt.

## 2. Transition

Parameter	Direkt	Kontinuierlich
Effizienz	-	-
Effektivität	-	-
Erlernbarkeit	Hoch	Hoch
Robustheit	-	-
Geschwindigkeit	Schnell	Langsam
Realismus	Gering	Hoch
Komfort	Hoch	Gering
Visuelle Komplexität	Mittel	Gering

Tabelle 3.4: Parameterbewertung Transition

*Effizienz:* Die Gestaltung der Transition hat keinen Einfluss auf die Effizienz der Interaktion.

*Effektivität:* Die Gestaltung der Transition hat keinen Einfluss auf die Effektivität der Interaktion.

*Erlernbarkeit:* Beide Transitionen werden bereits in einer Vielzahl bekannter VR-Anwendungen implementiert, bspw. in -. Folglich lässt sich der Wiedererkennungswert beider Optionen als hoch einschätzen.

*Robustheit:* Die Gestaltung der Transition hat keinen Einfluss auf die Robustheit der Interaktion.

*Geschwindigkeit:* Im Vergleich zur kontinuierlichen Bewegung weist die direkte Rotation eine höhere Geschwindigkeit auf, da die Rotation unmittelbar und ohne Übergangszeit erfolgt [Onuki and Kumazawa, 2019].

*Realismus:* Die kontinuierliche Rotation der Szene imitiert die natürliche Bewegung des



Kopfes und wirkt daher für die Nutzenden realistisch. Demgegenüber ist die direkte Rotation eine vergleichsweise unrealistische Transition [Onuki and Kumazawa, 2019].

*Komfort:* Eine kontinuierliche Rotation der Kamera kann bei Nutzenden relativ rasch Symptome von Motion Sickness hervorrufen [Farmani and Teather, 2020, Onuki and Kumazawa, 2019], wodurch der Komfort deutlich geschwächt wird. Im Gegensatz dazu birgt die direkte Rotation ein vergleichsweise geringes Risiko für das Auftreten von Motion Sickness [Farmani and Teather, 2020, Onuki and Kumazawa, 2019] und erweist sich somit für Nutzende als angenehmer.

*Visuelle Komplexität:* Der unmittelbare Wechsel des Sichtfelds bei der direkten Rotation kann dazu führen, dass Nutzende kurzzeitig die räumliche Orientierung verlieren und sich erst wieder visuell an das neue Bild gewöhnen müssen [L. Franz et al., 2021]. Eine potenzielle Maßnahme, um diesem Umstand entgegenzuwirken, wäre die Implementierung eines geringen Rotationswinkels, sodass die Nutzenden zeitnah einen visuellen Anker im Raum ausfindig machen können. Demgegenüber kann die kontinuierliche Bewegung dazu führen, dass das räumliche Bewusstsein bei den Nutzenden erhöht wird [L. Franz et al., 2021].

### 3. Bestätigung

Parameter	Keine	Pop-Up-Menü	Zeitinterval
Effizienz	Hoch	Gering	Hoch
Effektivität	-	-	-
Erlernbarkeit	?	?	?
Robustheit	Gering	Hoch	Hoch
Geschwindigkeit	Hoch	Mittel	Langsam-Mittel
Realismus	Hoch	Gering	Mittel
Komfort	Mittel	Mittel	Mittel
Visuelle Komplexität	Gering	Mittel	Mittel

Tabelle 3.5: Parameterbewertung Bestätigung

*Effizienz:* Bezüglich der Bewertung der Effizienz muss zwischen korrekt beabsichtigten Eingaben und fehlerhaften Eingaben unterschieden werden. Bei korrekten Eingaben ist bietet die Implementierungsoption ohne zusätzliche Bestätigung die höchste Effizienz, da durch eine einzige Eingabe das Interaktionsziel erreicht werden kann. Bei fehlerhaften Eingaben hingegen wird die Effizienz deutlich gemindert, da der Aufwand zu Korrektur des Fehlers höher ist als bei einer Option mit zusätzlicher Bestätigung.

*Effektivität:* TO DO!

*Erlernbarkeit:* TO DO!

*Robustheit:* Die Robustheit der Interaktion wird durch eine zusätzliche Bestätigung der Eingabe erhöht, da dadurch Fehler bzw. fehlerhafte Eingaben unmittelbar und mit minimalem Aufwand korrigiert werden können. Wichtig bei der Option mit Zeitintervall ist, dass deutlich visualisiert wird, dass die Eingabe erfolgreich war und das Bestätigungsintervall aktiv ist. Ohne die Bestätigung wird die Robustheit als geringer bewertet.

*Geschwindigkeit:* Bezüglich der Geschwindigkeit ist klar, dass eine zusätzliche Bestätigung sich auf jeden Fall negativ auf die insgesamt Interaktionsgeschwindigkeit auswirkt. Die Option ohne zusätzliche Bestätigung ist demnach die schnellste Option. Eine zusätzliche Eingabe als Bestätigung erfordert jedoch nur einen geringen zeitlichen Zusatzaufwand. Bei einem Bestätigungsintervall ist der Zusatzaufwand abhängig von der Länge der gesetzten Zeit.

*Realismus:* Bezüglich des Realismus kann festgehalten werden, dass eine zusätzliche Bestätigung der Interaktion zu einer Verringerung des Realismus führt und damit vermutlich das Gefühl von Presence mindert. Bei einem zusätzlichen Pop-Up-Menü ist der Bruch im Realismus zu die visuelle Unterbrechung vermutlich am höchsten.

*Komfort:* Wenn keine zusätzliche Bestätigung benötigt wird, muss auch entsprechend keine zusätzliche Eingabe erfolgen, wodurch der physische Aufwand so gering wie möglich gehalten wird. Jedoch ist gleichzeitig eine höhere Konzentration erforderlich, da fehlerhafte Eingaben nicht unmittelbar korrigiert werden können. Wird eine zusätzliche Bestätigung gefordert, in welcher Form auch immer, sind die Auswirkungen entsprechend umgekehrt.

*Visuelle Komplexität:* Auch bzgl. der visuellen Komplexität kann gesagt werden, dass eine zusätzliche Bestätigung dazu führt, dass die visuelle Komplexität erhöht wird, da in beiden Optionen zusätzliche visuelle Hinweise bzw. Informationen benötigt werden. Generell ist bei beiden Bestätigungsoptionen die visuelle Komplexität im Detail abhängig von der Gestaltung des Pop-Up-Menüs bzw. der Visualisierung des Zeitintervalls.

#### 4. Initialisierung

*Effizienz:* Beide Varianten weisen eine hohe Effizienz auf. Ist das Scanning durchgehend aktiv, können zu jedem Zeitpunkt Interaktionen durchgeführt werden. Die Initialisierung erfordert einen zusätzlichen Zwischenschritt, der jedoch mit minimalem Aufwand verbunden ist, sodass die Effizienz nicht maßgeblich beeinträchtigt wird.

*Effektivität:* Das Vorhandensein einer Initialisierung hat keinen Einfluss auf die Effektivität.

Parameter	Immer an	Ein-/Ausschaltbar
Effizienz	Hoch	Hoch
Effektivität	-	-
Erlernbarkeit	Hoch	Gering
Robustheit	-	-
Geschwindigkeit	Schnell	Schnell
Realismus	Gering	Hoch
Komfort	Mittel	Hoch
Visuelle Komplexität	Hoch	Mittel

Tabelle 3.6: Parameterbewertung Initialisierung

vität der Interaktion.

*Erlernbarkeit:* Die Initialisierung des Scanning-Verfahrens muss den Nutzenden zunächst erklärt werden. Die Erlernbarkeit wird daher insgesamt als gering bewertet. Sofern das Scanning-Verfahren durchgängig aktiv ist, erübrigt sich eine zusätzliche Erläuterung der Funktionsweise.

*Robustheit:* Das Vorhandensein einer Initialisierung hat keinen Einfluss auf die Robustheit der Interaktion.

*Geschwindigkeit:* Bei dauerhaft aktiviertem Scanning-Verfahren ist die Durchführung von Interaktionen zu jedem Zeitpunkt möglich. Dies hat eine entsprechende positive Auswirkung auf die benötigte Interaktionszeit. Die Initialisierung des Scannings erfolgt jedoch durch eine einfache Interaktion, sodass die Geschwindigkeit durch diesen Zwischenschritt lediglich minimal beeinflusst wird. In der praktischen Anwendung ist eine solche minimale Beeinflussung vermutlich nicht von Relevanz.

*Realismus:* Ist das Scanning-Verfahren nicht durchgängig aktiv, besteht für die Nutzenden die Möglichkeit, das Video in der Szene genauer zu betrachten und sich auf dieses zu konzentrieren, wodurch ein tieferes Eintauchen in die Szene ermöglicht wird. Es ist anzunehmen, dass sich dadurch das Gefühl von Presence erhöht. Ein permanentes Scanning hingegen dürfte eher zu einer weniger realistischen Wahrnehmung führen.

*Komfort:* Die permanente Überlagerung des Videos in der Szene mit dem Scanning-Verfahren könnte von Nutzenden als störend empfunden werden. Eine Fokussierung der Nutzenden auf den Inhalt des Videos erfordert eine deutlich höhere Konzentration, um eine Ablenkung durch die Visualisierung des Scanning-Verfahrens zu vermeiden. Sofern das Verfahren eigenständig aktiviert und deaktiviert werden kann, ist die erforderliche

Konzentration geringer. Diese Option bedingt jedoch eine minimale Steigerung der erforderlichen Interaktionen.

*Visuelle Komplexität:* Die permanente Überlagerung des Videos in der Szene und der Bewegung Scanning-Verfahrens erhöht die insgesamt visuelle Komplexität. Für die Nutzenden ist es, wie bereits unter dem Parameter Komfort beschrieben, ggf. schwierig, sich auf das Video zu konzentrieren. Durch die Möglichkeit, das Scanning zu aktivieren oder auch zu deaktivieren ermöglicht es den Nutzenden hingegen, die visuelle Komplexität der Szene eigenständig zu reduzieren.

### 3.4 Finale Konzepte

In der Bewertung der Optionen hinsichtlich der Interaktionsaufgaben sowie der Komponenten zeigt sich, dass eine hohe Geschwindigkeit und Effizienz oft mit einer geringeren Robustheit und einem reduzierten Komfort einhergehen. Diese Parameter stehen in einer Wechselwirkung zueinander, wodurch eine gleichzeitige Optimierung nicht uneingeschränkt möglich ist. Daher wird ein Konzept aufgestellt, das primär auf die Optimierung von Geschwindigkeit und Effizienz abzielt, während ein alternatives Konzept den Fokus auf Robustheit und Komfort legt. Bei Situationen, in denen die Entscheidung zwischen den Ausprägungsoptionen aufgrund ähnlicher Bewertungen unklar ist, werden die weiteren Parameter herangezogen, um festzustellen, welche Option insgesamt mehr Vorteile aufweist. Auf Basis dieser Überlegungen ergeben sich die folgenden Konzepte.

#### 1. Konzept: Fokus Geschwindigkeit und Effizienz

Das erste Konzept fokussiert sich auf eine schnelle und effiziente Interaktion. Für die Selektion wird das Cartesian Scanning gewählt, da es eine mittlere Geschwindigkeit und hohe Effizienz bietet. Bei der Navigation werden Menü-Optionen aufgrund des gewählten Selektionsverfahrens ausgeschlossen, wodurch zwischen der Modi-Option sowie der direkten Navigation entschieden werden muss. Aufgrund der höheren Effektivität wird hier die Modus-Option bevorzugt. Bzgl. der Komponente Display steht entweder eine fest im Sichtfeld verankerte Darstellung oder ein Neustart des Scannings zur Auswahl. Aufgrund der geringeren visuellen Komplexität und höheren Effektivität wird die Neustart-Option gewählt. Für die Transition wird aufgrund der höheren Geschwindigkeit eine direkte Rotation implementiert. Eine Bestätigung entfällt, um die Geschwindigkeit und Effizienz weiter zu steigern. Außerdem ist das Scanning dauerhaft aktiv, da dies trotz einer knappen Entscheidung insgesamt als die schnellere Lösung eingeschätzt wird.

## 2. Konzept: Fokus auf Robustheit und Komfort

Das zweite Konzept legt den Fokus auf Robustheit und Komfort. Für die Selektion wird Automatic Item Scanning gewählt. Obwohl das Step Item Scanning eine höhere Robustheit bietet, fällt der Komfort dabei deutlich negativ aus, weshalb die automatische Variante bevorzugt wird. Bei der Navigation sind aufgrund der gewählten Selektion nur Menü-Optionen möglich. Hier wird ein Menü mit festen Navigationsschritten gewählt, da es einen höheren Komfort bietet. Bezüglich der Komponente Display wird die Neustart-Option gewählt, um eine höhere Robustheit zu gewährleisten. Für die Transition wird erneut die direkte Rotation gewählt, da diese deutlich mehr Komfort für Nutzende bietet. Hinsichtlich der Bestätigung stehen Pop-Up und Zeit zur Auswahl. Beide Optionen bieten eine hohe Robustheit und nur geringe Unterschiede im Komfort. Es wird die Zeit-Option gewählt, da sie zu einem höheren wahrgenommenen Realismus führt. Das Scanning-Verfahren ist nicht dauerhaft aktiv, sondern erfordert eine Initialisierung, da dies sowohl mehr Komfort als auch mehr Realismus bietet.

## 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 Cartesain 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 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 dem beim Laden einer neuen Szene alle interaktiven Paneo-Elemente als Child-Objekte angehängt werden.
- *Scene Manager*: Enthält unter anderem 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 unter anderem die Main Camera sowie den UI-Anchor. Dem UI-Anchor sind zentrale UI-Komponenten wie das Ingame-Menü 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. 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 zudem, ob der Schalter länger als zwei Sekunden gehalten wurde, um anschließend 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 Objekts.
- *CartesianScanHandler*: Dies ist das zentrale Script für das Cartesian Scanning. Es steuert, zu welchem Zeitpunkt die Animationen der Linien aufgerufen werden, berechnet den Schnittpunkt der 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 der Feedback-Audioausgaben bei Interaktionen zuständig.

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 si-

cherzustellen, 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 Objekt 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 Animation der Linien im Cartesian Scanning wurden direkt an die animierten Objekte gebunden. Dies erleichtert die Konfiguration der Animationen 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.

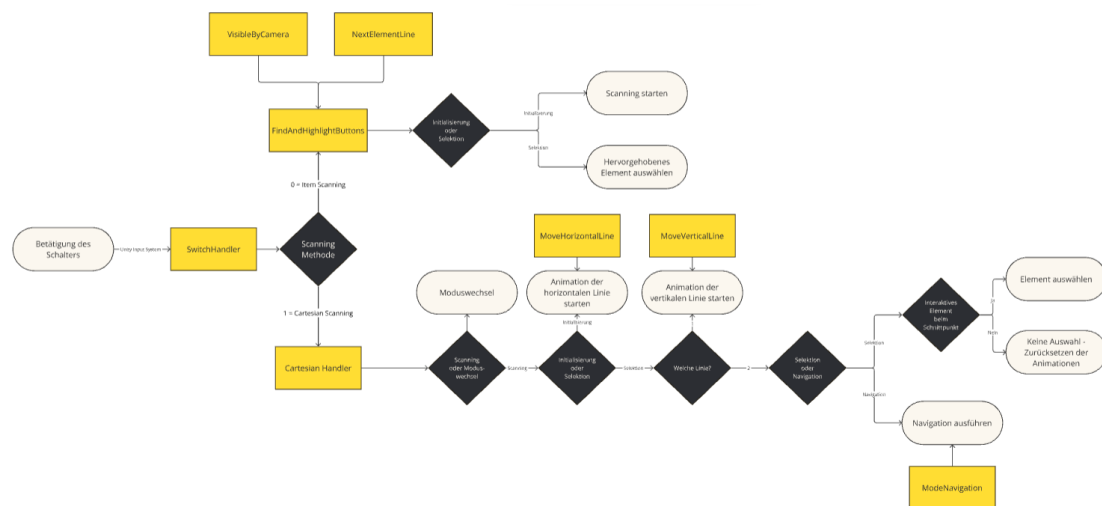


Abbildung 4.1: Grundlegender Informationsfluss der Softwarearchitektur

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 Scanning Handler 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 Auswahl der interaktiven



Elemente.

## 4.2 Gemeinsame Komponenten beider Konzepte

Die Implementierung der beiden Scanning-Methoden basiert unter anderem auf gemeinsam genutzten Komponenten, die Funktionalitäten beider Verfahren unterstützen. Dazu gehören die Eingabeverarbeitung, die Audioausgabe sowie ein zusätzliches Highlight-GameObject, das allen interaktiven Objekten 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 leicht 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 Nutzenden 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 Benutzerfeedback von zentraler Bedeutung ist.

### Highlight Objekt

Allen interaktiven Objekten 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 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 Objekt ausgewählt werden kann. Zusätzlich wird der Collider verwendet, um zu prüfen, ob das Objekt 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 Linien ein interaktives Objekt 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 Methoden

### 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 Menüs und einen Button zum Öffnen des Navigationsmenüs. Sobald der *SwitchHandler* eine Eingabe registriert, prüft er, ob das Scanning bereits aktiv ist oder initialisiert werden muss.

Diese Information wird über einen Methodenaufruf an das Skript *FindAndHighlightButtons* übergeben. Ist das Scanning noch nicht aktiv, wird die Methode *StartItemScanning* aufgerufen, andernfalls die Methode *SwitchSelection*. Diese beiden Methoden stellen die zentrale Funktionalität der Scanning-Methode 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  IEnumerator ItemScanning()
2  {
3      while (isScanActive)
4      {
5          foreach (GameObject button in allButtons)
6          {
7              // Versuche, das Highlight-Objekt zu finden
8              GameObject highlight = button.transform.Find("Highlight").
9              gameObject;
10
11              if (highlight != null)
12              {
13                  // Prüfe, ob das Highlight-Objekt sichtbar ist
14                  if (visibleByCamCheck.IsVisibleByCam(cam, highlight.
15                  GetComponent<Collider>()))
16                  {
17                      // Aktiviere das Highlight-Objekt
18                      highlight.GetComponent<MeshRenderer>().enabled = true
19
20                      ;
21                      highlightedObject = button;
22
23                      // Starte die Animation zum naechsten Objekt
24                      GameObject nextObject = GetNextHighlightableObject();
25                      scriptNextElement.DrawLineBetweenElements(
26                      highlightedObject, nextObject);
27
28                      // Warte fuer die Scan-Rate
29                      yield return new WaitForSeconds(ScanRate);
30
31                      // Deaktiviere das Highlight-Objekt, falls es noch
32                      existiert
33                      if (highlight != null)
34                      {
35                          highlight.GetComponent<MeshRenderer>().enabled =

```

```

31     false;
32     }
33     }
34     }
35     }
36 }

```

Listing 4.1: Vereinfachter Pseudocode des Item Scannings

Die Iteration läuft, solange das Scanning aktiv ist. Für das aktuelle Objekt wird zunächst geprüft, ob es ein Highlight-GameObject als Kind-Objekt enthält. Diese Prüfung stellt sicher, dass wirklich nur Objekte 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*. Ist diese Prüfung positiv, wird das Highlight durch Einschalten des Mesh Renderers aktiviert. Gleichzeitig wird das hervorgehobene Objekt in einer entsprechenden Variablen gespeichert, um bei der Selektion darauf zugreifen zu können. Anschließend wird das nächste hervorgehobene Objekt bestimmt und eine Animation gestartet, die eine Linie zwischen dem aktuellen und dem nächsten Objekt 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 Objekt noch existiert. Dies ist notwendig, da das Objekt unter Umständen während der Wartezeit gelöscht werden kann. Würde an dieser Stelle keine erneute Überprüfung erfolgen, käme es zu einer `NullPointerException`.

### Auswahl eines interaktiven Elements

Die Auswahl eines hervorgehobenen Elements erfolgt über die Methode `SwitchSelection`. Hier wird geprüft, ob ein Objekt aktuell hervorgehoben ist. Ist dies der Fall, wird die Button-Komponente des Objekts gesucht und deren `OnClick`-Event ausgelöst. Der Ablauf ist im folgenden Pseudocode [Listing 4.2](#) veranschaulicht.

```

1
2 public void SwitchSelection()
3 {
4     if (highlightedObject != null)
5     {
6         // Suche nach der Button-Komponente
7         GameObject buttonTrigger = highlightedObject.transform.Find("
8 Btn_Trigger").gameObject;
9         Button buttonToClick = buttonTrigger.GetComponent<Button>();
10
11         // 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 4.2: Vereinfachter Pseudocode der Selektion des Item Scanning

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  $40^\circ$  (rechts) bzw.  $-40^\circ$  (links).

### Animation zwischen dem aktuellen und folgenden Element

Um die Reihenfolge zu verdeutlichen, in der die interaktiven Objekte hervorgehoben werden, wird eine Linie zwischen dem aktuellen und dem nächsten Element gezogen. Dazu wurde der *IngameScene* das GameObject *NextElementLine* hinzugefügt. Dieses enthält einen Line Renderer zur Darstellung der Linie sowie das Skript *NextElementLine*, das die Steuerung der Animation übernimmt. 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 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-Interactable beschrieben. Wird das Dialog-Objekt innerhalb des ItemScanning ausgewählt, wird das OnClick-Ereignis ausgelöst. Dadurch 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 Objekt aktiv ist bzw. sich auf der vordersten Ebene befindet. Dies erfordert besondere Aufmerksamkeit bei der Implementierung, da verschiedene Fälle berücksichtigt werden müssen. Für jedes Objekt, das eine weitere Ebene öffnet, muss geprüft werden, wie sich die Auswahl der Scan-Optionen dadurch ändert.

### 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 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 Linie bewegt sich von oben nach unten vor dem Nutzenden, während die vertikale Linie sich von links nach rechts bewegt. Die Bewegung der Linien wird über die genannten Skripte gesteuert. Der Aufbau dieser Skripte wird im Folgenden näher beschrieben.

#### Animation der Linien

Die Methode *StartAnimation* startet die Coroutine *MoveLine*, in der die Bewegung der Linie gesteuert wird. Hier wird zunächst die Start- und Endposition der Linie definiert. Bei der horizontalen Linie sind dies der obere und der untere Rand des Canvas. Bei der vertikalen Linie entspricht dies dem linken und rechten Rand des Canvas. Nachfolgend ist in Listing 4.3 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 4.3: Ausschnitt aus der Methode zur Animation der Linien im Cartesian Scanning

Sobald die Linie ihre Endposition erreicht hat, wird über eine Zählervariable geprüft, ob die Animation bereits dreimal hintereinander ausgeführt wurde. Ist die Linie noch nicht

dreimal durchgelaufen, wird die Animation zurückgesetzt und erneut gestartet. Erst nach dem dritten Durchlauf wird die *Reset*-Methode im *CartesianScanHandler* aufgerufen, wo der gesamte Fortschritt des Scanning zurückgesetzt wird. Dieses wiederholte Durchlaufen wurde zur Fehlerkorrektur implementiert. Wird die gewünschte Position der Linie beim ersten Durchlauf verfehlt, kann einfach abgewartet werden, bis die Linie wieder die gewünschte Position erreicht hat. Durch den Abbruch nach drei Durchläufen werden zudem vermeidbare Fehlentscheidungen vermieden.

Eine weitere wesentliche Methode innerhalb des Skripts ist *StopAnimation*. Diese bewirkt, dass die aktuell laufende Coroutine gestoppt wird, sodass die Linie in ihrer aktuellen Position verbleibt. Die Methode wird aufgerufen, sobald Nutzende die Position der Linie im Zuge des Scanning 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 Stadium sich der Scanning-Prozess befindet, und ruft entsprechend die nächste erforderliche Methode auf (vgl. [Listing 4.5](#)).

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

Listing 4.4: Methode *Scanning* im *CartesianScanHandler*

Die Methode *ScanningFirstLine* aktiviert das *GameObject* der horizontalen 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 Linie aktiviert und die Animation gestartet. In der Methode *Selection* wird anschließend die Animation der zweiten Linie gestoppt, die beiden *GameObjects* wieder deaktiviert, die Animationen 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 Linien berechnet. Dies wird im Folgenden anhand von Pseudocode veranschaulicht.

```

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 4.5: Pseudocode zur Veranschaulichung der Methode *FindIntersection* im *CartesianScanHandler*

Je nachdem, welcher Modus aktiv ist, wird anschließend unterschiedlich verfahren. Ist der Selektionsmodus aktiviert, wird mit Hilfe der Methode *CheckForInteractiveElement* überprüft, ob sich hinter dem Schnittpunkt der Linien ein interaktives Objekt 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 Rays gewählt, um einen Toleranzbereich in Form eines Radius einführen zu können. Dadurch wird ein Objekt auch dann ausgewählt, wenn der Schnittpunkt das Objekt 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 Objekts ausgelöst. Trifft der Ray auf mehrere Objekte, z.B. wenn zwei Ebenen mit interaktiven Elementen übereinander liegen, wird stets das Objekt ausgewählt, auf das der Ray zuerst trifft. Das heißt, es wird das Element ausgewählt, das näher an der Kamera positioniert ist. Dadurch wird sichergestellt, dass UI-Elemente wie das Ingame-Menü oder das Hint-Panel bevorzugt ausgewählt werden und keine unbeabsichtigten Selektionen bei Verdeckungen auftreten. Wurde kein Objekt 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 2 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. Bei der Auswahl der Farben wurde darauf geachtet, dass diese einen relativ hohen Kontrast aufweisen, damit der Unterschied möglichst auch für Menschen mit Farbschwäche wahrnehmbar 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 robust. Ein Vorteil dieses Verfahrens liegt insbesondere darin, dass Objekte 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 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. Die Festlegung des Radius des *SphereCasts* stellte eine Herausforderung dar, da hier ein Kompromiss gefunden werden musste. Der Radius musste groß genug sein, um kleine Verfehlungen zu tolerieren, aber klein genug, um bei dicht beieinan-

derliegenden Objekten eine korrekte Auswahl zu ermöglichen. Es galt also, verschiedene Werte auszuprobieren und zu prüfen, ob sich das gewünschte 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 NullPointerExceptions werden so abgefangen. Dies wird z.B. beim Item Scanning verwendet. Wenn bspw. ein Objekt kein Highlight-Objekt hat oder das Objekt zur Laufzeit der Coroutine gelöscht wird, wird dies abgefangen und das Objekt 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, also 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. Eine zentrale Einschränkung beim Item Scanning liegt in der fehlenden Konfigurierbarkeit der Reihenfolge, in der interaktive Objekte hervorgehoben werden. Gegenwärtig basiert die Reihenfolge auf der Reihenfolge, in der die Objekte beim Laden der Szene erzeugt werden. Dies kann dazu führen, dass in der Szene räumlich nahe beieinander liegende

Objekte 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 Objekte integriert werden. Durch eine Sortierung der Liste der interaktiven Objekte anhand dieses Parameters könnte die Reihenfolge an die räumliche Anordnung angepasst werden. Im Rahmen dieser Arbeit wurde diese Option nicht implementiert, da kein Zugriff auf den Webeditor von PaneoVR besteht und somit dieser Parameter nicht in den Webeditor und damit nicht in die Gestaltung von Szenarien integriert werden konnte.

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 im Kapitel Evaluation 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 Cartesian Scanning stellt die Platzierung der 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 Scanbereich vorsehen, um die Benutzerfreundlichkeit zu erhöhen. Eine weitere Einschränkung liegt in der planaren Darstellung der 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 beispielsweise ein Audiosignal hinzugefügt, das jedes Mal ertönt, wenn ein neues Objekt hervorgehoben wird. Ein ähnliches Prinzip könnte beim Cartesian Scanning eingeführt werden. Hier könnte bspw. ein Audiosignal ertönen, wenn eine Linie während der Animation mit einem interaktiven Objekt kollidiert.

## 4.6 Installation, Nutzerdokumentation

Unbedingt zur Implementierung gehört dann am Ende auch eine Beschreibung, wie man die Implementierung zu nutzen hat. Dazu gehört die Beschreibung, wie man die Implementierung bezieht und installiert, genauso wie eine kurze Beschreibung der Bedienung. Falls letztere zu umfangreich werden würde, aber notwendig ist, kann diese auch im Anhang aufgeführt und mit einem Verweis gearbeitet werden.

## 5 Evaluation

Die Evaluation der entwickelten Interaktionsverfahren erfolgt in zwei Hauptabschnitten: einer technischen und einer inhaltsbasierten Evaluation. Ziel ist es, sowohl die technische Leistungsfähigkeit der Verfahren 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 Teil der Evaluation dient der Überprüfung der Annahmen zu technischen Parametern wie Geschwindigkeit, Effizienz, Robustheit und Erlernbarkeit, die im Rahmen der Konzeption getroffen wurden. Für jedes Verfahren wird ein spezifisches Szenario entwickelt, in dem die technischen Fähigkeiten der Verfahren 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 Objekten im Vordergrund. Im zweiten Teil hingegen liegt der Schwerpunkt auf der Interaktion mit Dialogfeldern und damit auf der Interaktion auf verschiedenen Ebenen sowie auf der Selektion von Objekten, 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 Testpersonen. Die Anordnung der Elemente innerhalb der Szenen ist absichtlich herausfordernd gestaltet, um insbesondere die Robustheit und Effektivität der Verfahren 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 Scanning-Verfahren

Eine Herausforderung bei der Verwendung von Item Scanning ist zunächst eine hohe Anzahl von Elementen in der Szene, 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 können von den Nutzenden leicht übersehen oder in der Scanreihenfolge nicht erwartet werden, was zu fehlerhaften oder unbeabsichtigten Selektionen führen kann. Darüber hinaus stellen längere Wartezeiten für Elemente, die erst spät in der Scanning-Reihenfolge erscheinen, eine weitere Herausforderung dar. Diese können zu Ungeduld und damit zu fehlerhaften Eingaben führen. Auch die Interaktion auf mehreren Ebenen, z.B. das Schließen von Popups oder das Navigieren in Dialogfeldern, wird als weitere Herausforderung angesehen.

Im Gegensatz dazu ist beim Cartesian Scanning die größte Herausforderung, das gewünschte Element bei 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 wird auch bei diesem Scanning-Verfahren die Positionierung von Elementen am Rand des Sichtfeldes als Herausforderung angesehen. Insbesondere wenn sich Objekte weit oben oder weit links befinden, kann es schnell passieren, dass Nutzende die Selektion verpassen und auf einen weiteren Durchlauf der Selektionslinie 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, kann es schnell zu Verschiebungen und damit zu Fehleingaben kommen. Auch hier ist somit eine erhöhte Konzentration erforderlich.

Die Szenarien für den technischen Teil 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 Objekte aufgegriffen. Zur Minimierung von Zufallseffekten wurde eine Mindestanzahl von fünf gleichartigen Objekten pro Szenario integriert.

## Fragestellungen und Datenerhebung

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

*Geschwindigkeit*

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

#### *Robustheit*

- Wie häufig treten Fehler bzw. unbeabsichtigte Eingaben bei den jeweiligen Verfahren auf?
- Was sind die Gründe für auftretende Fehler?
- Wie häufig werden zwei oder mehr Durchgänge im Scanning benötigt?

#### *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?

Die Erfassung der Daten erfolgt durch die Kombination von automatischem Logging direkt aus Unity und einem Evaluationsprotokoll. 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, während zusätzliche Beobachtungen dokumentiert werden.

### **Erwartete Ergebnisse der technischen Evaluation**

Die Erwartungen an die Verfahren basieren auf den theoretischen Überlegungen zur je-

weiligen Methodik. Beim Item Scanning wird erwartet, dass es intuitiver und schneller erlernbar ist als das Cartesian Scanning. Dies könnte sich durch eine geringere Anzahl von Fehlern und selten erforderlichen Korrekturen zeigen. Allerdings wird das Item Scanning voraussichtlich bei der Navigation insgesamt langsamer sein, da größere Navigationen erforderlich sind, um alle platzierten Elemente erreichen zu können. Im Cartesian Scanning wird ein holpriger Start und eine entsprechend steilere Lernkurve erwartet. Es wird davon ausgegangen, dass die Navigation effizienter erfolgt und größere Winkel schneller abgedeckt werden. Herausforderungen wie eng beieinanderliegende Objekte oder solche am Rand des Sichtfelds könnten zu einer höheren Fehlerquote führen.

### 5.1.2 Inhaltsbasierte Evaluation

Der zweite Teil der Evaluation widmet sich der Wechselwirkung zwischen Interaktionsschnittstelle und Inhalt. Hier stehen die Parameter Komfort, Effizienz und visuelle Komplexität im Vordergrund. Die inhaltsbasierte Evaluation ergänzt den technischen Teil, indem die Verfahren in realistischeren Anwendungsszenarien getestet werden. Es werden zwei Szenarien in Form einfacher Rätsel nach dem Prinzip von Escape Rooms erstellt. Die Aufgabe der Testpersonen besteht darin, einen hinter interaktiven Objekten wie Bildern oder Audiodateien versteckten Zahlencode zu ermitteln bzw. einen Schlüssel zur Lösung des Szenarios zu finden. Die Rätsel sind bewusst einfach gehalten, um die kognitiven Fähigkeiten der Testpersonen nicht zu 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 Objekte erfolgt in Abstimmung mit den 360°-Videos der Szene.

#### Fragestellungen und Datenerhebung

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

##### *Ablenkung durch die Verfahren*

- Werden die Interaktionsschnittstellen als störend empfunden?
- Lenken die Interaktionsschnittstellen die Aufmerksamkeit der Testpersonen vom Inhalt ab?

##### *User Experience*



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

#### *Motion Sickness*

- Tritt Motion Sickness auf? Wenn ja, wie stark sind die Symptome ausgeprägt?
- Welche Symptome treten auf meisten auf?
- Gibt es Unterschiede zwischen den Verfahren in Bezug auf die Häufigkeit oder 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 User Experience Questionnaire (UEQ) verwendet. Der Simulator Sickness Questionnaire (SSQ) 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 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?

Zusätzlich wird auch für das inhaltsbasierte Szenario ein automatisierter Log erstellt. Auch hier werden, wie zuvor im technischen Szenario, Daten zur Geschwindigkeit erfasst. Auch Fehler werden weiterhin im Evaluationsprotokoll vermerkt.

Die erhobenen Daten werden statistisch ausgewertet. Für den UEQ werden die Ergebnisse mit dem Auswertungstool der Autoren analysiert. Für den SSQ werden die Ergebnisse nach der Methode der Entwickler ausgewertet und die entsprechenden SSQ-Scores ermittelt. Zusätzliche Fragen werden mit Hilfe von Mittelwerten und Standardabweichungen interpretiert, um Einblicke in spezifische Aspekte wie z.B. die inhaltliche Fokussierung zu erhalten.

Es wird erwartet, dass das Item Scanning aufgrund der geringen Anzahl von Elementen in den jeweiligen Szenen schneller ist und daher hinsichtlich der UX besser bewertet wird, insbesondere hinsichtlich Effizienz. Beide Verfahren sollten eine geringe Ablenkung vom Inhalt aufweisen und nur minimal zur Entstehung von Motion Sickness beitragen.

## 5.2 Durchführung

Die Evaluation der entwickelten Interaktionsschnittstellen wurde im Zeitraum vom 11. bis 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 Benutzerfreundlichkeit der Implementierungen. Die Auswahl einer heterogenen Gruppe ohne motorische Beeinträchtigungen ermöglicht eine Bewertung der allgemeinen Nutzbarkeit der Verfahren. Darüber hinaus können potenzielle Schwachstellen in der Interaktion und technische Herausforderungen unabhängig von spezifischen Beeinträchtigungen identifiziert werden. Dies schafft eine solide Basis für weitere Optimierungen und ermöglicht es, die Implementierungen so anzupassen, dass sie zukünftig sowohl für die Zielgruppe als auch für einen breiteren Kreis von Nutzenden geeignet sind.

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 eigentlichen Tests 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 des ersten Interaktionsverfahrens:**

Die Funktionsweise des ersten zu testenden Scanning-Verfahrens 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 nahm die Testperson 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), ebenfalls in digitaler Form, ausgefüllt.

### **6. Wechsel zum zweiten Verfahren:**

Die gleiche Prozedur wurde danach für das zweite Verfahren 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 Interaktionsverfahren noch die spezifischen Inhalte der Szenarien die Ergebnisse der Evaluation beeinflussen, wurden die Teilnehmenden in vier Testgruppen eingeteilt. Jede Gruppe begann mit einer anderen Kombination von Interaktionsverfahren und Szenario. Dadurch wurde eine gleichmäßige Verteilung der Bedingungen erreicht und mögliche Verzerrungen minimiert. **Tabelle 5.1** veranschaulicht die festgelegten Testgruppen und zeigt, welche Gruppe mit welchem Scanning-Verfahren und welchem inhaltsbasierten Szenario startete.

Bezeichnung	Erste Interaktionsschnittstelle	Inhaltsbasiertes Szenario
ITH	Item Scanning	Szenario 1
ISB	Item Scanning	Szenario 2
CTH	Cartesian Scanning	Szenario 1
CSB	Cartesian Scanning	Szenario 2

Tabelle 5.1: Testgruppen der Evaluation

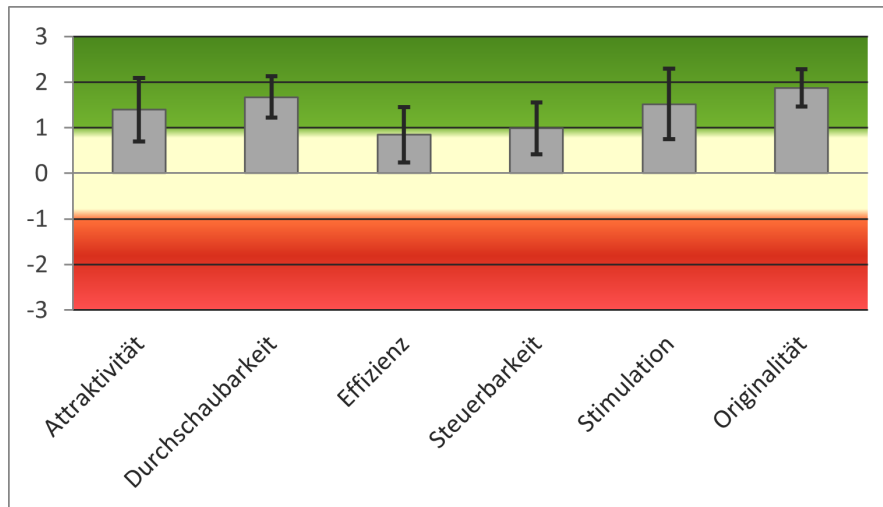


Abbildung 5.1: Ergebnisse UEQ Item Scanning

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

Abbildung 5.2: Ergebnisse UEQ Item Scanning

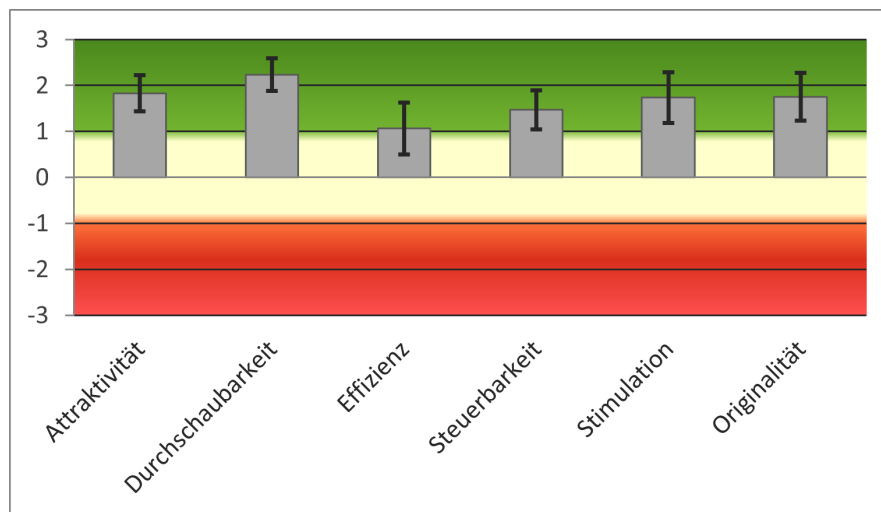


Abbildung 5.3: Ergebnisse UEQ Cartesian Scanning

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

Abbildung 5.4: Ergebnisse UEQ Cartesian Scanning

## 5.3 Ergebnisse

### 5.3.1 UEQ

### 5.3.2 SUS

### 5.3.3 SSQ

### 5.3.4 Ablenkung/Presence

### 5.3.5 Geschwindigkeit und Zyklen

### 5.3.6 Fehler und leere Eingaben

### 5.3.7 Korrelationen

### 5.3.8 Feedback/Anmerkungen/Beobachtungen

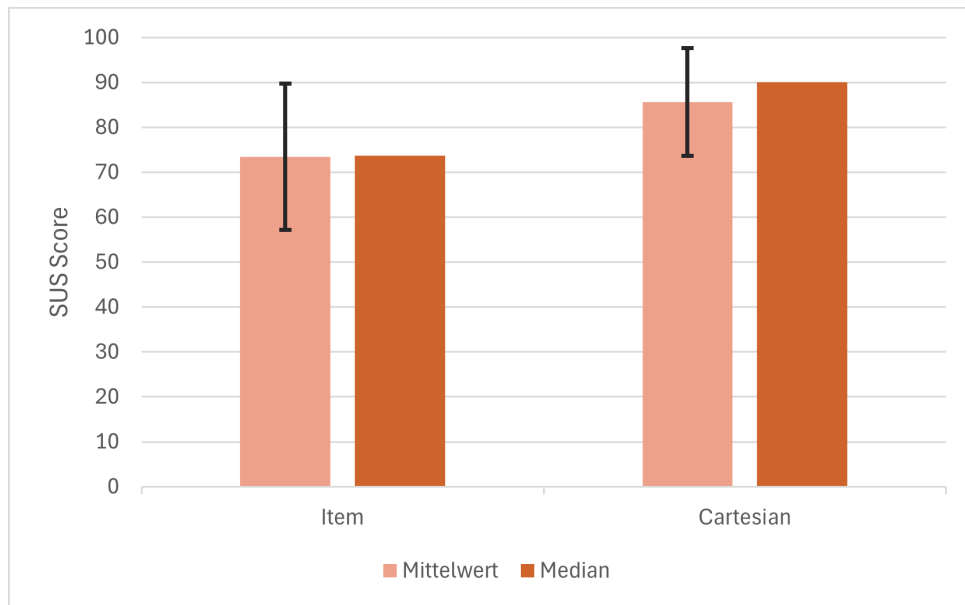


Abbildung 5.5: Ergebnisse SUS

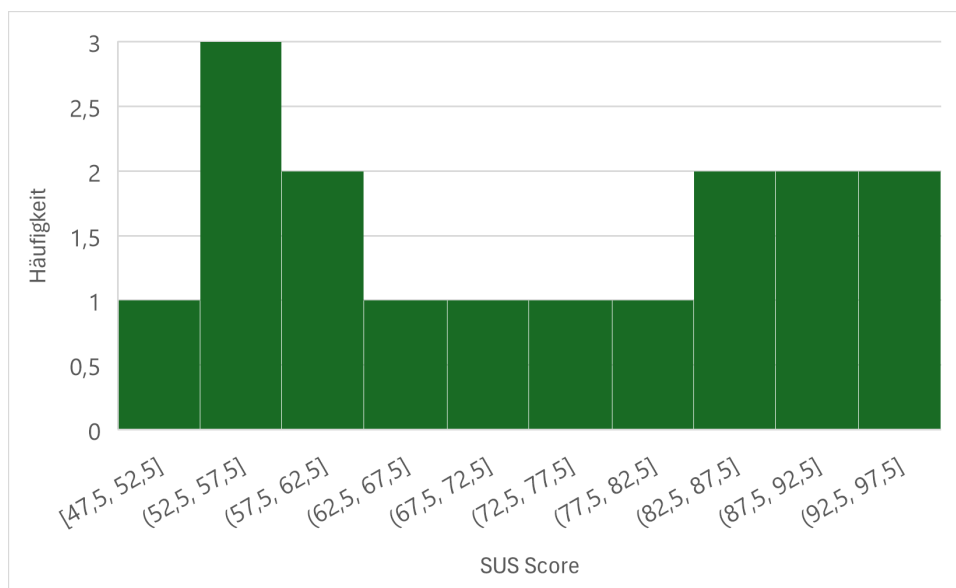


Abbildung 5.6: Histogramm SUS Item Scanning

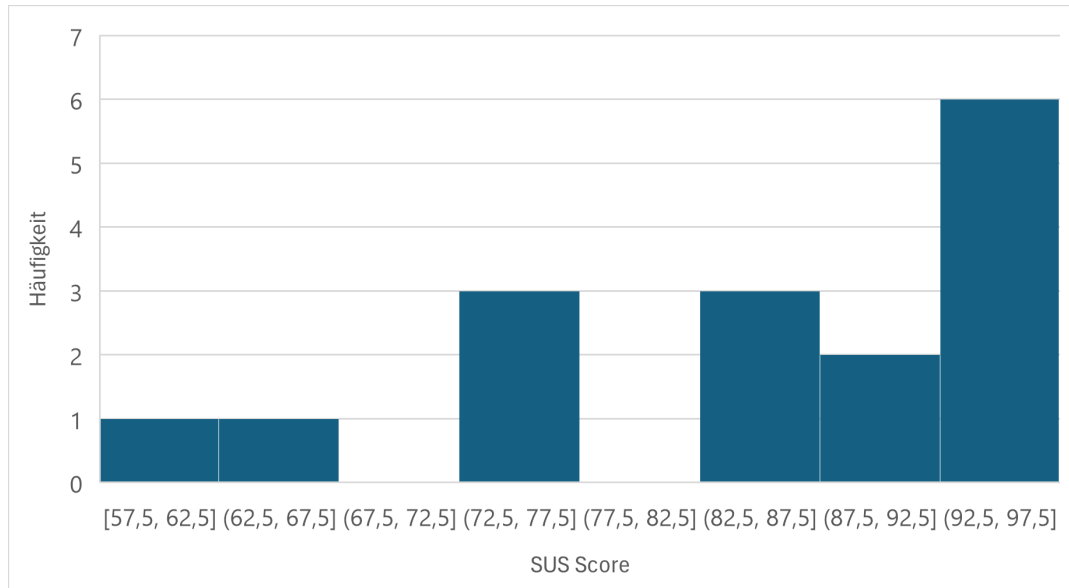


Abbildung 5.7: Histogramm SUS Cartesian Scanning

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

Abbildung 5.8: Häufigkeit der Symptome



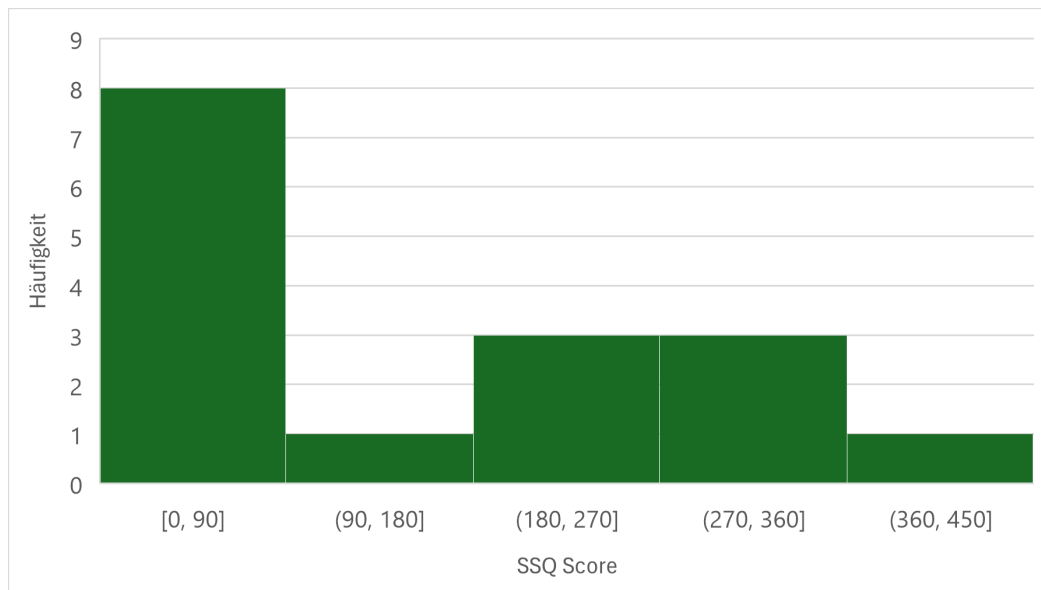


Abbildung 5.9: Histogramm SSQ Item Scanning

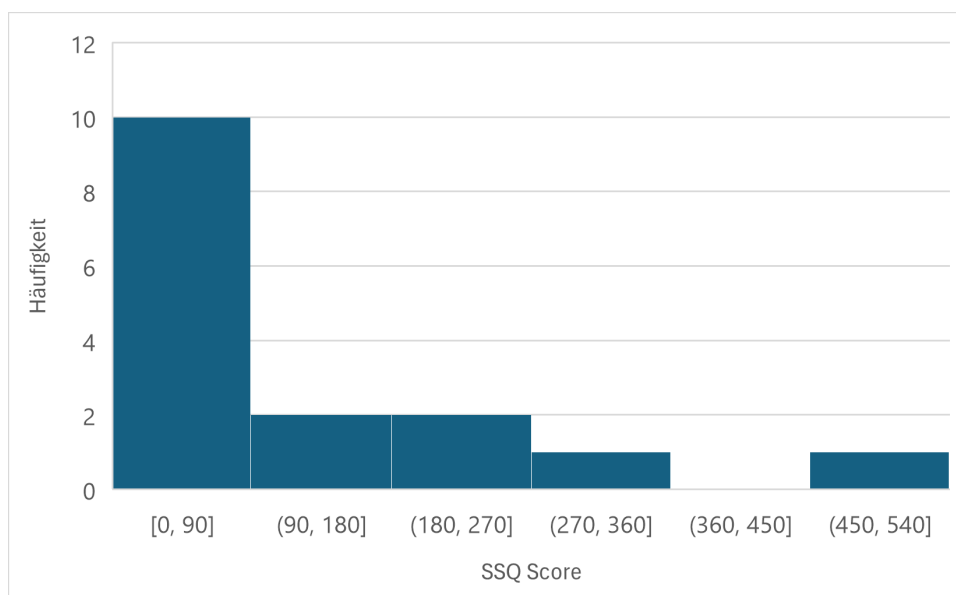


Abbildung 5.10: Histogramm SSQ Cartesian Scanning

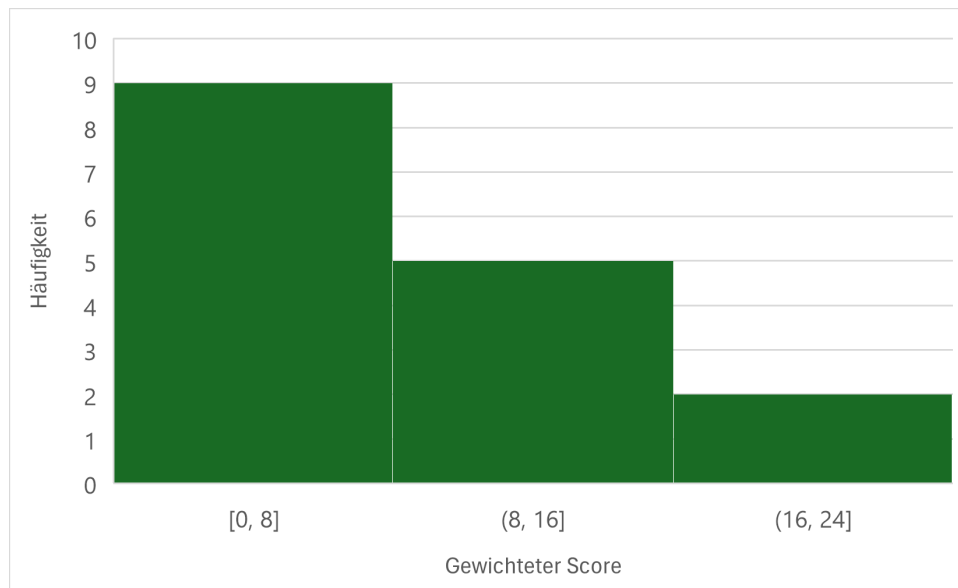


Abbildung 5.11: Histogramm Nausea Item Scanning

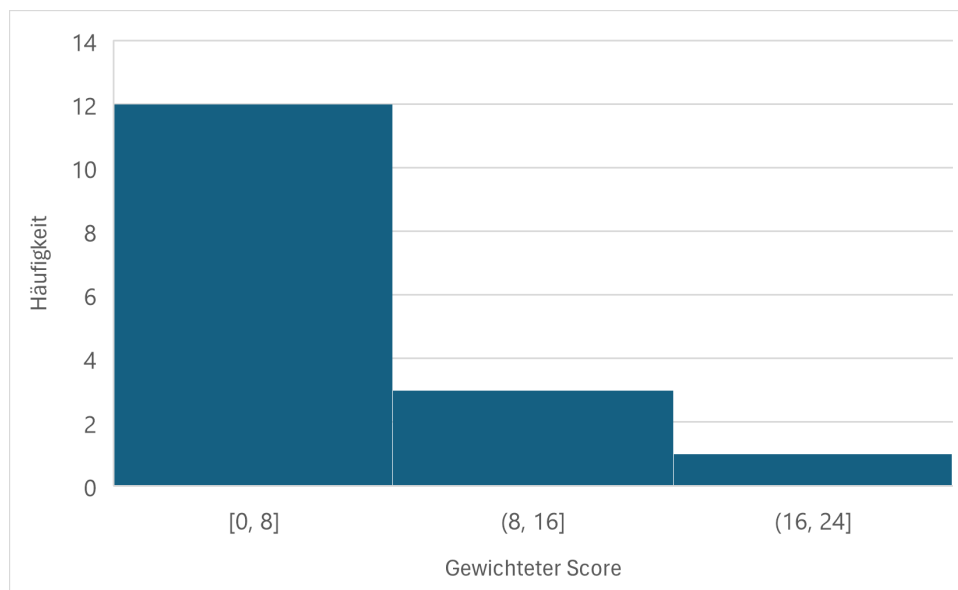


Abbildung 5.12: Histogramm Nausea Cartesian Scanning

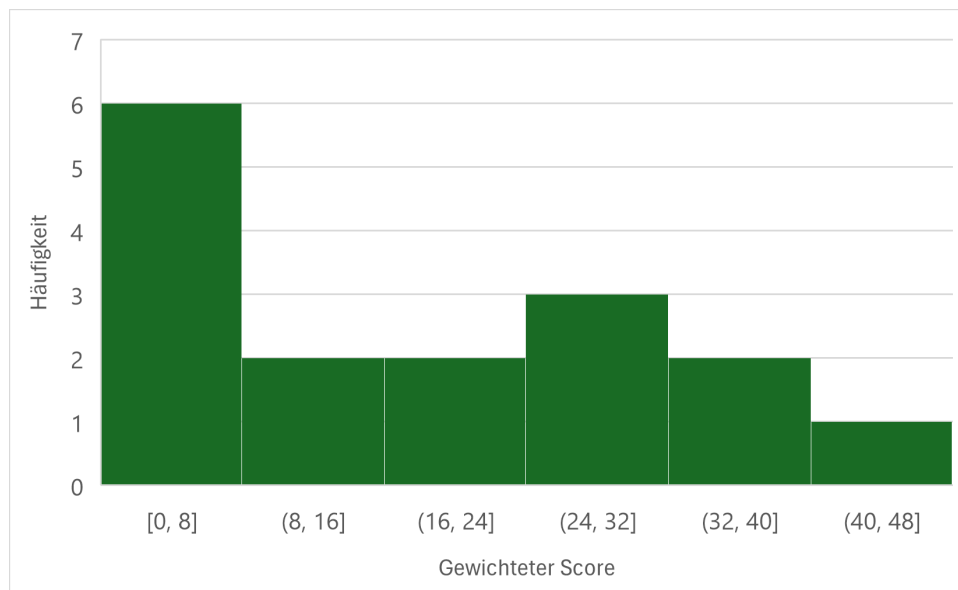


Abbildung 5.13: Histogramm Oculomotor Item Scanning

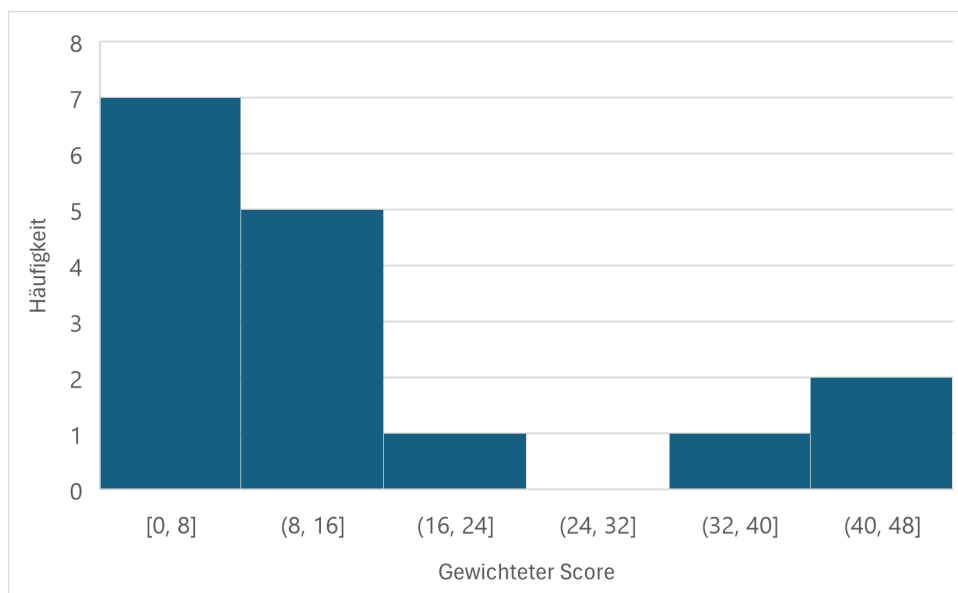


Abbildung 5.14: Histogramm Oculomotor Cartesian Scanning

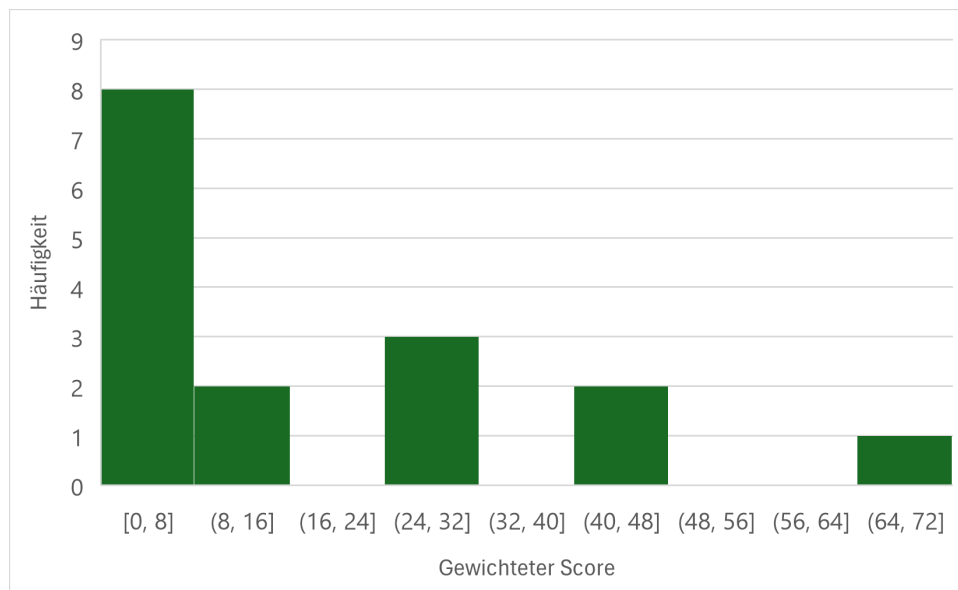


Abbildung 5.15: Histogramm Disorientation Item Scanning

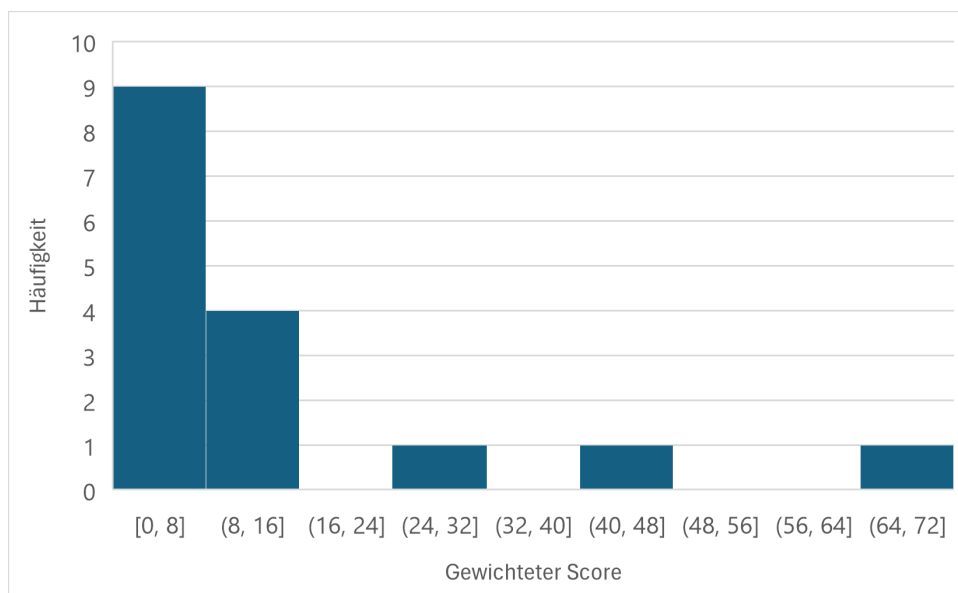


Abbildung 5.16: Histogramm Disorientation Cartesian Scanning

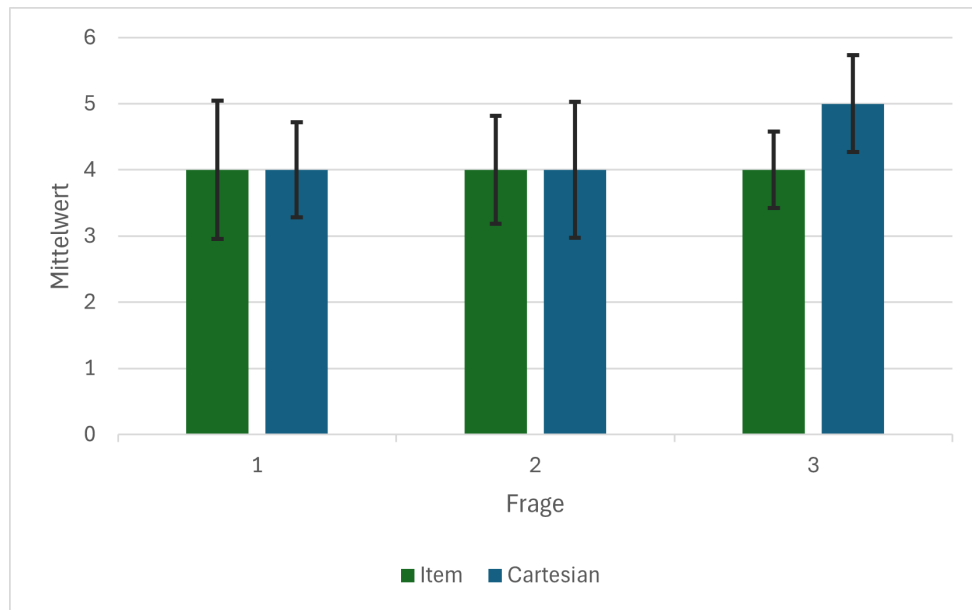


Abbildung 5.17: Fragen zur Presence/Ablenkung

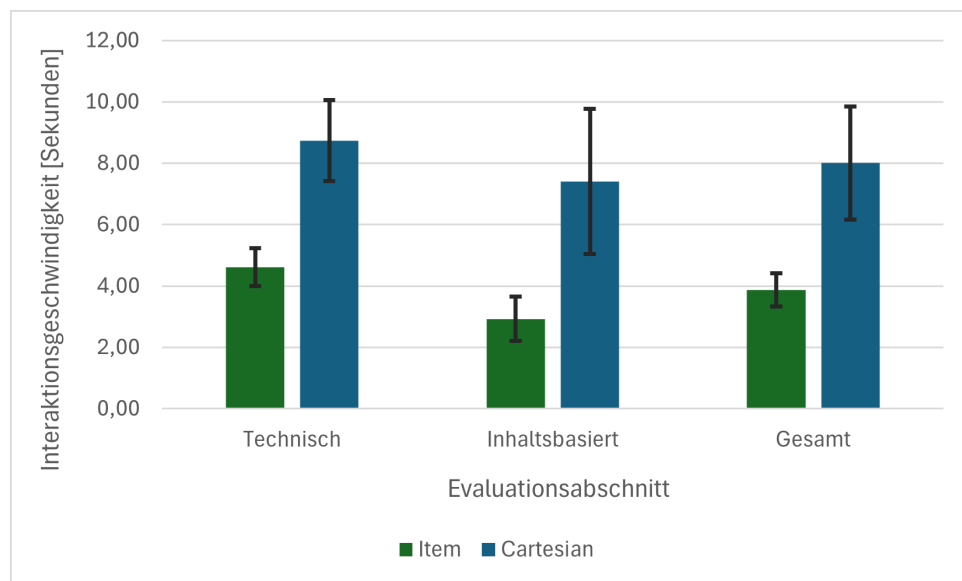


Abbildung 5.18: Interaktionsgeschwindigkeiten

	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

Abbildung 5.19: Interaktionsgeschwindigkeiten Tabelle

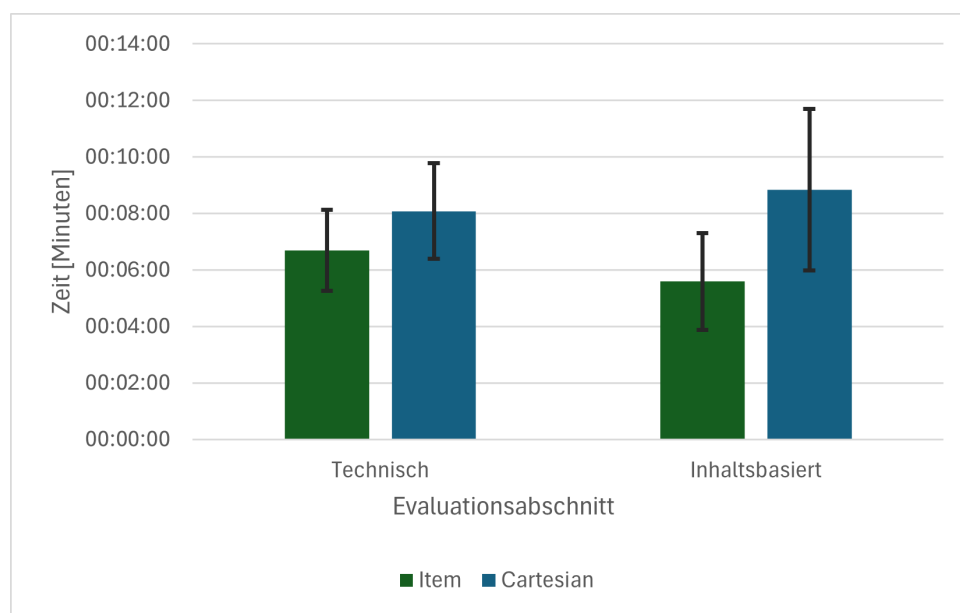


Abbildung 5.20: Benötigte Zeit

	Item Scanning		Cartesian Scanning	
	Regressionskoeffizient	Korrelationskoeffizient	Regressionskoeffizient	Korrelationskoeffizient
Technisch	-0,839	0,503	0,180	0,326
Inhaltsbasiert	-0,391	0,754	0,241	1,177
Gesamt	-2,539	1,105	-0,492	1,461

Kein Zusammenhang
Schwacher Zusammenhang

Abbildung 5.21: Regressionskoeffizienten Lernkurve

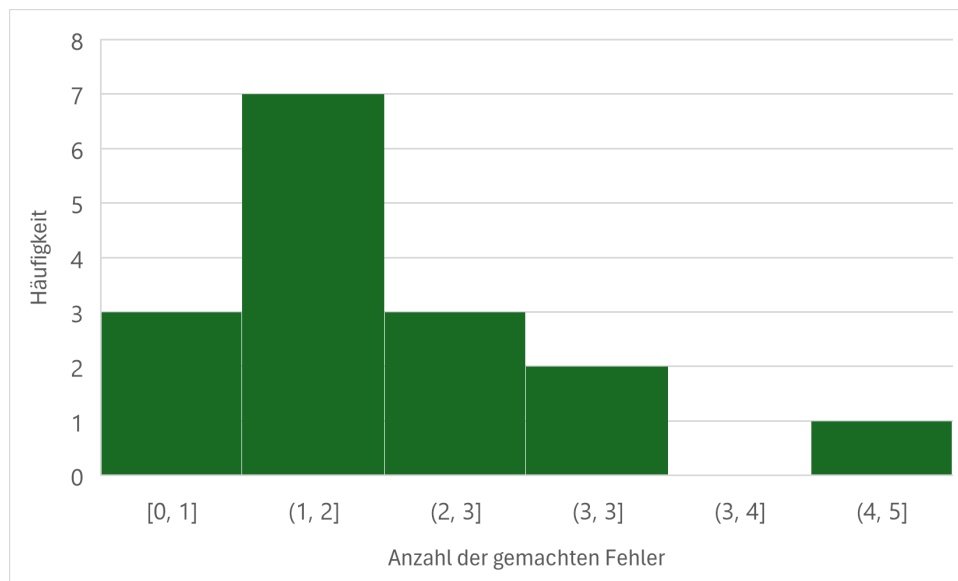


Abbildung 5.22: Anzahl Fehler Item Technisch

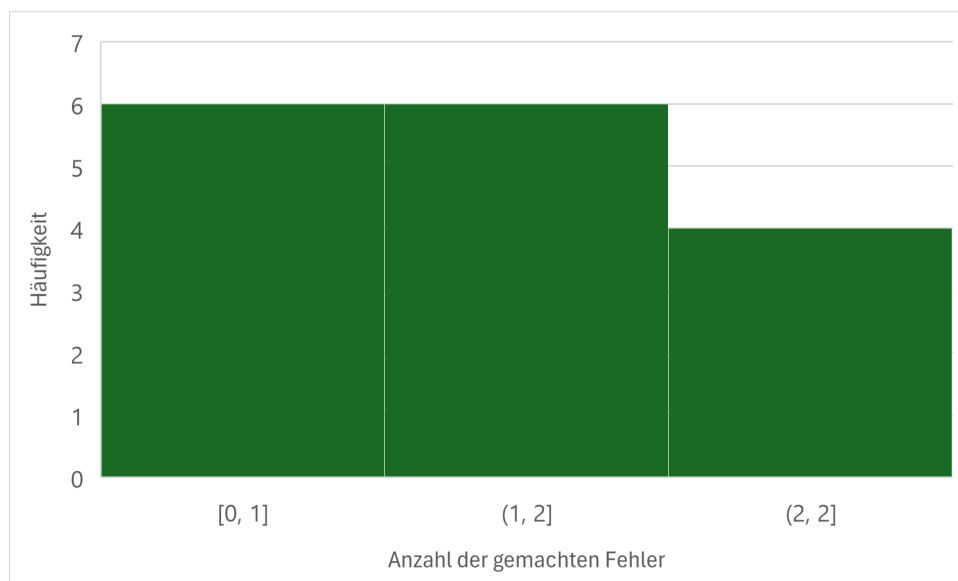


Abbildung 5.23: Anzahl Fehler Item Inhalt

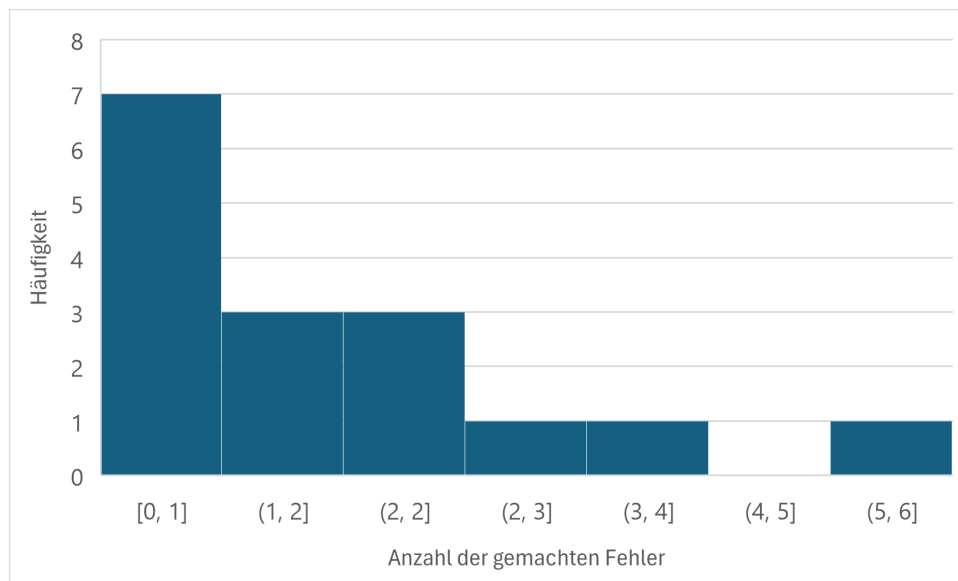


Abbildung 5.24: Anzahl Fehler Cartesian Technisch

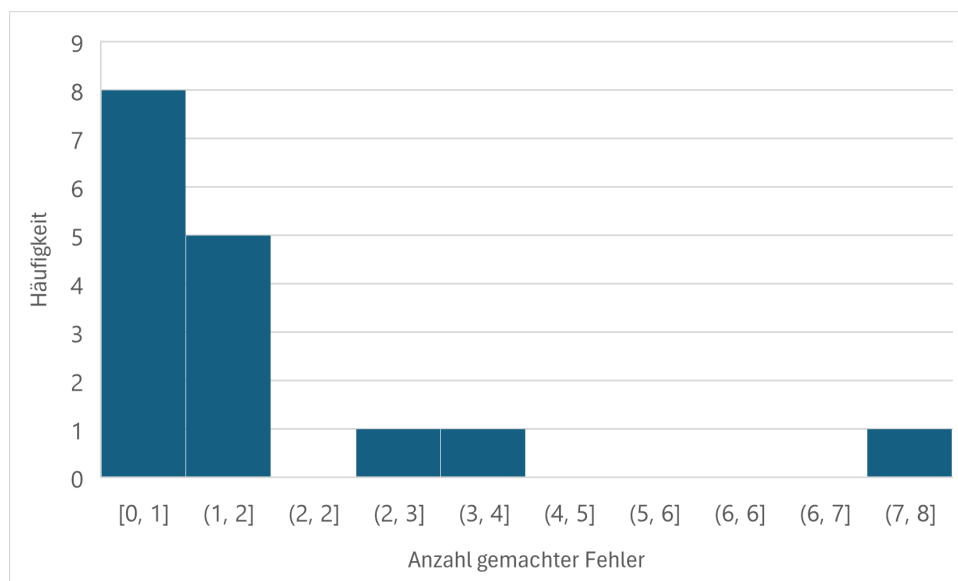


Abbildung 5.25: Anzahl Fehler Cartesian Inhalt



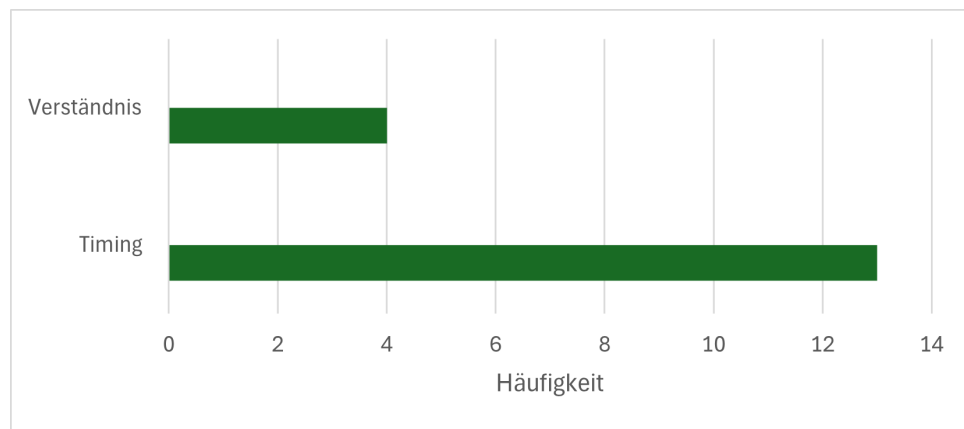


Abbildung 5.26: Gründe für Fehler Item Scanning

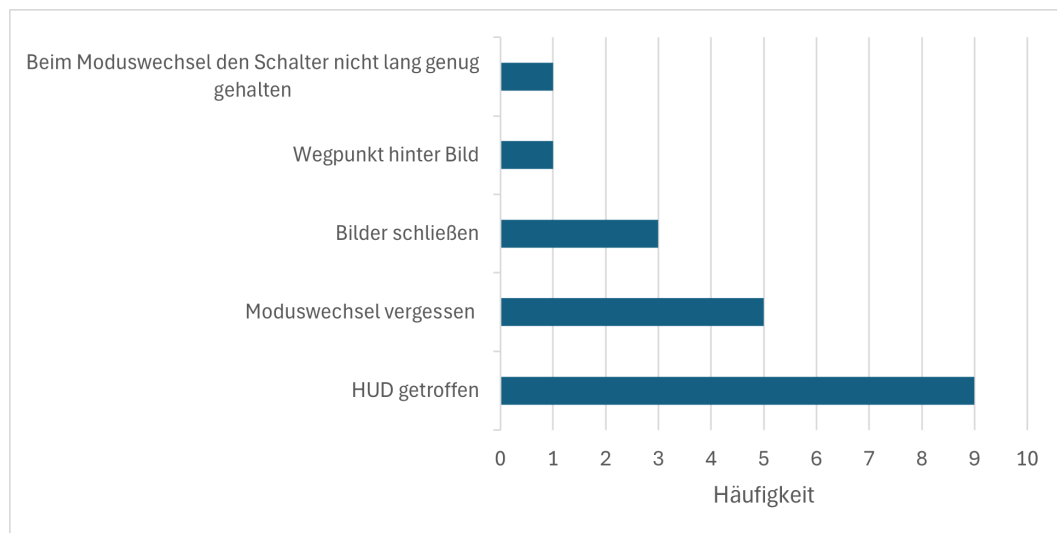


Abbildung 5.27: Gründe für Fehler Cartesian Scanning

	Korrelationskoeffizient	
	X-Koordinate, Geschwindigkeit	Y-Koordinate, Geschwindigkeit
Technischer Abschnitt	0,7395	-0,8240
Inhaltsbasierter Abschnitt	0,6156	-0,6849
Gesamt	0,6804	-0,7562

Abbildung 5.28: Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit

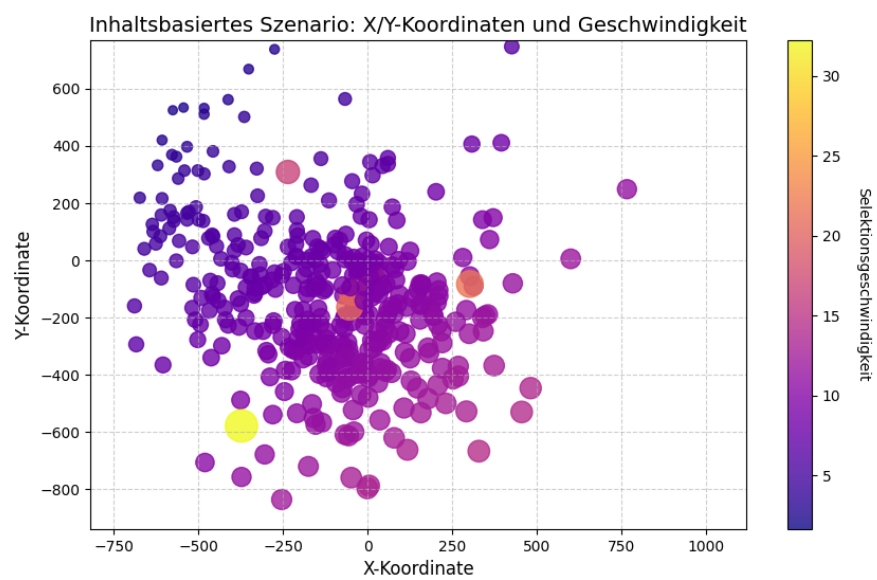


Abbildung 5.29: Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit Inhaltsbasierter Abschnitt

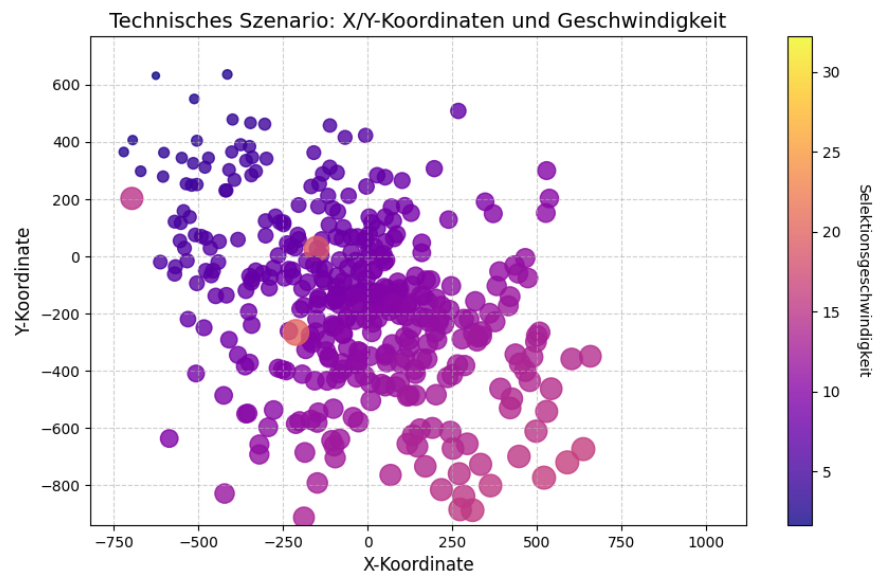


Abbildung 5.30: Zusammenhang Position des Objekts im Sichtfeld und Selektionsgeschwindigkeit Technischer Abschnitt

Item	Geschwindigkeit Technischer Abschnitt und SUS			Geschwindigkeit Gesamt und UEQ-Effizienz		
	Korrelationskoeffizient	t-Wert	p-Wert	Korrelationskoeffizient	t-Wert	p-Wert
Item	0,19	0,71	0,49	0,38	1,56	0,14
Cartesian	-0,29	-1,13	0,28	-0,15	-0,56	0,58

Kein Zusammenhang  
 Schwacher Zusammenhang  
 Mäßiger Zusammenhang  
 Starker Zusammenhang

Abbildung 5.31: Zusammenhang Interaktionsgeschwindigkeit und Usability/UX

## 6 Diskussion

Hier wird reflektiert, in welchem Umfang die Zielsetzungen der Arbeit erreicht werden konnten. In der Regel gelingt dies durch den Bezug auf die Evaluation.

Typischer Umfang: 1-2 Seiten.

### 6.1 Ausblick

Manchmal konnte im Rahmen einer Bachelor- oder Masterarbeit ein wissenschaftliches Thema auch so tief bearbeitet werden, dass wir es auf einem wissenschaftlichen Workshop oder einer Konferenz veröffentlichen konnten.

Für Themen aus dem Bereich Virtuelle oder Erweiterte Realität bietet sich dazu zum Beispiel der jährlich stattfindende Workshop der Fachgruppe VR/AR der Gesellschaft für Informatik an. [Bluhm et al. \[2009\]](#) ist nur ein Beispiel von mehreren Veröffentlichungen aus unserer Gruppe.

### 6.2 Test

Kleine Tests, die später noch passend in den übrigen Text eingearbeitet werden.

#### 6.2.1 Tabelle

Das ist [Tabelle 6.1](#).

Test 1	Test 2
abc	123
def	456
ghi	789

Tabelle 6.1: Tabellen-Test

### 6.2.2 Bild

Das ist **Abbildung 6.1**.



Abbildung 6.1: Test-Bild

### 6.2.3 Begriffe und Abkürzungen

The **Latex** typesetting markup language is specially suitable for documents that include **mathematics**. **Formulas** are rendered properly and easily once one gets used to the commands.

Given a set of numbers, there are elementary methods to compute its **Greatest Common Divisor**, which is abbreviated **GCD**. This process is similar to that used for the **Least Common Multiple (LCM)**.

### 6.2.4 Mathematik

$$\sin \alpha = \left( \frac{a}{c} \right) \quad (6.1)$$

### 6.2.5 Code

Listing 6.1 ist ein Code-Beispiel in Python.

```

1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
28
29    return M

```

Listing 6.1: Code-Beispiel mit Python

## Literaturverzeichnis

DIN EN ISO 9241-110:2020-10, Ergonomie der Mensch-System-Interaktion\_ - Teil\_110: Interaktionsprinzipien (ISO\_9241-110:2020); Deutsche Fassung EN\_iso\_9241-110:2020, a. URL <https://www.dinmedia.de/de/-/-/320862700>. [17]

DIN EN ISO 9241-11:2018-11, Ergonomie der Mensch-System-Interaktion\_ - Teil\_11: Gebrauchstauglichkeit: Begriffe und Konzepte (ISO\_9241-11:2018); Deutsche Fassung EN\_iso\_9241-11:2018, b. URL <https://www.dinmedia.de/de/-/-/279590417>. [17]

Apple. Einführung in die Funktionen der Bedienungshilfen auf dem iPhone, 2024. URL <https://support.apple.com/de-de/guide/iphone/iph3e2e4367/ios>. [23]

Andreas Bluhm, Jens Eickmeyer, Tobias Feith, Nikita Mattar, and Thies Pfeiffer. Exploration von sozialen Netzwerken im 3D Raum am Beispiel von SoNVR für Last.fm. In Andreas Gerndt and Marc Erich Latoschik, editors, *Virtuelle und Erweiterte Realität - Sechster Workshop der GI-Fachgruppe VR/AR*, pages 269–280, Aachen, 2009. Shaker Verlag. [74]

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), January 2015. ISBN 978-0-323-09631-7. 10.1016/B978-0-323-09631-7.00006-5. URL <https://www.sciencedirect.com/science/article/pii/B9780323096317000065>. [17, 18, 22, 23, 24]

Matt Dombrowski, Peter A. Smith, Albert Manero, and John Sparkman. Designing Inclusive Virtual Reality Experiences. In Jessie Y.C. Chen and Gino Fragomeni, editors, *Virtual, Augmented and Mixed Reality. Multimodal Interaction*, pages 33–43, Cham, 2019. Springer International Publishing. ISBN 978-3-030-21607-8. 10.1007/978-3-030-21607-8\_3. [9, 10]

Ralf Dörner, Wolfgang Broll, Paul Grimm, and Bernhard Jung, editors. *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Aug-*

- mentierten 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>. [17]
- Yasin Farmani and Robert J. Teather. Evaluating discrete viewpoint control to reduce cybersickness in virtual reality. *Virtual Real.*, 24(4):645–664, dec 2020. ISSN 1359-4338. 10.1007/s10055-020-00425-x. URL <https://doi.org/10.1007/s10055-020-00425-x>. [20, 27, 31]
- 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>. [19, 25]
- 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, pages 1–14, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8096-6. 10.1145/3411764.3445196. URL <https://doi.org/10.1145/3411764.3445196>. [8, 9]
- 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). [23]
- Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool, September 2015. ISBN 978-1-970001-12-9. [17]
- 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, July 1993. ISSN 1050-8414. 10.1207/s15327108ijap0303\_3. URL [https://doi.org/10.1207/s15327108ijap0303\\_3](https://doi.org/10.1207/s15327108ijap0303_3). Publisher: Taylor & Francis \_eprint: [https://doi.org/10.1207/s15327108ijap0303\\_3](https://doi.org/10.1207/s15327108ijap0303_3). [7]
- Robert S. Kennedy, Julie Drexler, and Robert C. Kennedy. Research in visually induced motion sickness. *Applied Ergonomics*, 41(4):494–503, July 2010. ISSN 0003-6870. 10.1016/j.apergo.2009.11.006. URL <https://www.sciencedirect.com/science/article/pii/S0003687009001574>. [6]



- 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>. [10, 16, 17, 20, 31]
- Joseph J. LaViola. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, January 2000. ISSN 0736-6906. 10.1145/333329.333344. URL <https://dl.acm.org/doi/10.1145/333329.333344>. [6]
- Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. pages 282–292, Boston, MA, December 1995. 10.1117/12.197321. URL <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981543>. [4]
- 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, pages 1–9, New York, NY, USA, June 2019. Association for Computing Machinery. ISBN 978-1-4503-6709-7. 10.1145/3317956.3318150. URL <https://doi.org/10.1145/3317956.3318150>. [10]
- 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, October 2019. 10.1109/ISMAR-Adjunct.2019.00122. URL <https://ieeexplore.ieee.org/document/8951960>. [9]
- 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*, pages 1–13, Virtual Event Greece, October 2020. ACM. ISBN 978-1-4503-7103-2. 10.1145/3373625.3416998. URL <https://dl.acm.org/doi/10.1145/3373625.3416998>. [8, 9, 10]
- Charles M. Oman. Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology*, 68(2):294–303, February 1990. ISSN 0008-4212. 10.1139/y90-044. URL <https://cdnsiencepub.com/doi/abs/10.1139/y90-044>. Publisher: NRC Research Press. [6]

- 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. [20, 27, 30, 31]
- Gary E. Riccio and Thomas A. Stoffregen. An ecological Theory of Motion Sickness and Postural Instability. *Ecological Psychology*, 3(3):195–240, September 1991. ISSN 1040-7413. 10.1207/s15326969eco0303\_2. URL [https://doi.org/10.1207/s15326969eco0303\\_2](https://doi.org/10.1207/s15326969eco0303_2). Publisher: Routledge \_eprint: [https://doi.org/10.1207/s15326969eco0303\\_2](https://doi.org/10.1207/s15326969eco0303_2). [6]
- 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://www.nature.com/articles/nrn1651>. Publisher: Nature Publishing Group. [4]
- Dimitrios Saredakis, Ancret Szpak, Brandon Birkhead, 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, March 2020. ISSN 1662-5161. 10.3389/fnhum.2020.00096. URL <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2020.00096/full>. Publisher: Frontiers. [6]
- 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, May 2019. ISSN 0167739X. 10.1016/j.future.2018.11.041. URL <https://linkinghub.elsevier.com/retrieve/pii/S0167739X18325044>. [5, 6, 7]
- 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>. [17, 18]
- 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>. [3]
- 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 Vir-

- tual Reality (VR) Gaming Controller for People with Motor Disabilities. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 161–164, Taichung, Taiwan, December 2018. IEEE. ISBN 978-1-5386-9269-1. 10.1109/AIVR.2018.00034. URL <https://ieeexplore.ieee.org/document/8613653/>. [10]
- 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>. [4]
- 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, 3, 5]
- Alice Wong, Hannah Gillis, and Ben Peck. Survey for People with Disabilities. Technical report, 2017. [9]

# **Anhang A**

## **Test-Anhang**

### **A.1 Hinweise zum Layout**

#### **A.1.1 Druckseiten**

Zur besseren Korrektur der Arbeit sollten die im Prüfungsamt abzugebenden Exemplare einseitig gedruckt werden. Die Vorlage ist entsprechend ausgelegt. Für sich selbst, für Angehörige, etc, kann man später dann leicht noch mal eine beidseitig bedruckte Fassung erstellen.

### **A.2 Zeitplanung**

- Erfahrungsgemäß ist das Aufschreiben der Arbeit für die meisten mit am schwersten. Daher sollte damit frühzeitig begonnen werden. Spätestens zwei Wochen vor der Abgabe wird es allerdings allerhöchste Eisenbahn!
- Der Druck der Arbeit kann unter Umständen einen Tag in Anspruch nehmen, daher empfiehlt es sich, einen Abgabetermin ab Dienstag-Freitag zu wählen.

## Anhang B

### Test-Anhang 2

Test

## **Erklärung**

Soweit meine Rechte berührt sind, erkläre ich mich einverstanden, dass die vorliegende Arbeit Angehörigen der Hochschule Emden/Leer für Studium / Lehre / Forschung uneingeschränkt zugänglich gemacht werden kann.

## **Eidesstaatliche Versicherung**

Ich, die Unterzeichnende, erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Quellenangaben und Zitate sind richtig und vollständig wiedergegeben und in den jeweiligen Kapiteln und im Literaturverzeichnis wiedergegeben. Die vorliegende Arbeit wurde nicht in dieser oder einer ähnlichen Form ganz oder in Teilen zur Erlangung eines akademischen Abschlussgrades oder einer anderen Prüfungsleistung eingereicht.

Mir ist bekannt, dass falsche Angaben im Zusammenhang mit dieser Erklärung strafrechtlich verfolgt werden können.

Lübeck, den 17. Dezember 2024

Finja Wegener