



---


# MASTERARBEIT

---

Herr  
Milas Marco Baumgarten, B.Sc.

**Konzeption und Implementation der  
Softwarearchitektur von Accessibility  
Features im Rahmen eines  
interaktiven Medienproduktes**

Mittweida, März 2023





Fakultät **Angewandte Computer- und Biowissenschaften**

---

# **MASTERARBEIT**

---

## **Konzeption und Implementation der Softwarearchitektur von Accessibility Features im Rahmen eines interaktiven Medienproduktes**

Autor:

**Milas Marco Baumgarten**

Studiengang:

Medieninformatik und interaktives Entertainment

Seminargruppe:

MI19w1-M

Erstprüfer:

Prof. Dr.-Ing Christian Roschke

Zweitprüferin:

Susan Labude, M.Sc.

Einreichung:

Mittweida, 06.03.2023

Verteidigung/Bewertung:

Mittweida, 2023



Faculty of **Applied Computer Sciences and Biosciences**

---

# MASTER THESIS

---

## **Conception and implementation of the software architecture of accessibility features in the context of an interactive media product**

Author:

**Milas Marco Baumgarten**

Course of Study:

Mediainformatics and interactive Entertainment

Seminar Group:

MI19w1-M

First Examiner:

Prof. Dr.-Ing Christian Roschke

Second Examiner:

Susan Labude, M.Sc.

Submission:

Mittweida, 06.03.2023

Defense/Evaluation:

Mittweida, 2023



## **Bibliografische Beschreibung:**

Baumgarten, Milas Marco:

Konzeption und Implementation der Softwarearchitektur von Accessibility Features im Rahmen eines interaktiven Medienproduktes . 2023. 87 S.

Mittweida, Hochschule Mittweida University of Applied Sciences, Fakultät Angewandte Computer und Biowissenschaften, Masterarbeit, 2023.

## **Referat:**

In dieser Arbeit ging es darum, eine Softwarelösung zu entwickeln, die auf die Anforderungen der Game Accessibility Guidelines abgestimmt ist. Die Godot Game Engine wurde als Entwicklungsplattform ausgewählt, da sie sich durch eine hohe Flexibilität und Skalierbarkeit auszeichnet. Durch die Verwendung dieser Engine war es möglich, eine stabile und robuste Basis zu schaffen, auf der die Implementierungen aufgebaut werden konnten. Ein wichtiger Aspekt bei der Entwicklung war die Wahl der richtigen Softwarearchitektur. Es wurde untersucht, welche Architekturen sich am besten eignen, um eine optimale Leistung und eine hohe Benutzerfreundlichkeit zu gewährleisten. Dazu wurden verschiedene Ansätze getestet und verglichen, um die bestmögliche Lösung zu finden. Im nächsten Schritt wurden die Implementierungen von Studierenden getestet und evaluiert. Dabei wurden Aspekte wie die Benutzerfreundlichkeit und der Mehraufwand, der durch die Nutzung der implementierten Features entsteht, untersucht. Es wurde auch darauf geachtet, wie verständlich die Implementierungen gestaltet sind und wie leicht sie zu bedienen sind. Die Ergebnisse der Evaluation wurden dann reflektiert und der Aufwand der Implementationen wurde in Hinblick darauf untersucht, dass es sich bei den Anforderungen fast ausschließlich um solche handelt, die laut den Game Accessibility Guidelines als Anforderungen mit geringer Komplexität eingestuft werden. Dies bedeutet, dass die Implementierung dieser Anforderungen relativ einfach sein und keine großen Schwierigkeiten bereiten sollten. Insgesamt war die Entwicklung der Softwarelösung ein Erfolg und hat gezeigt, dass die Einschätzung der Komplexität der Game Accessibility Guidelines größtenteils bestätigt werden kann. Des Weiteren wurde die Implementation von den Studierenden gut angenommen. Die Ergebnisse der Evaluation können dazu beitragen, einen technischen Ansatzpunkt für zukünftige Bewertungen der Anforderungen und Implementationsversuche zu bieten.





# Inhaltsverzeichnis

<b>Abbildungs- und Tabellenverzeichnis</b>	<b>IV</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielstellung . . . . .	2
1.2 Aufbau . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Accessibility . . . . .	4
2.1.1 Accessibility Guidelines im Internet . . . . .	5
2.1.2 Accessibility Guidelines in Videospielen . . . . .	6
2.2 Anforderungsanalyse . . . . .	7
2.2.1 Ablauf der Anforderungsanalyse . . . . .	8
2.2.2 Anforderungskatalog . . . . .	8
2.3 Softwaredesignpatterns und Softwaregrundlagen . . . . .	10
2.3.1 Softwaredesignpatterns der „Gang of Four“ . . . . .	10
2.3.2 Game Programming Patterns . . . . .	11
<b>3 Konzept und Analyse</b>	<b>13</b>
3.1 Anforderungskatalog . . . . .	13
3.1.1 Motorische Anforderungen . . . . .	14
3.1.2 Kognitive Anforderungen . . . . .	15
3.1.3 Visuelle Anforderungen . . . . .	16
3.1.4 Auditive Anforderungen . . . . .	16
3.1.5 Allgemeine Anforderungen . . . . .	17
3.1.6 Ausgeschlossene Anforderungen . . . . .	18
3.2 Technische Auswahl . . . . .	19
3.2.1 Godot Game Engine . . . . .	19
3.2.2 GitHub . . . . .	20
3.3 Konzept . . . . .	20
3.3.1 Konfigurierbare Tastenbelegung und Sensitivität . . . . .	20
3.3.2 Userinterfacenavigation . . . . .	21
3.3.3 Lesbarkeit von Texten . . . . .	21
3.3.4 Anpassbare Textgröße . . . . .	22

---

3.3.5	Screenreader Kompatibilität . . . . .	23
3.3.6	Untertitel . . . . .	25
3.3.7	Variable Lautstärke . . . . .	32
3.3.8	Accessibility Informationen & Feedback . . . . .	33
3.3.9	Persistente Einstellungen . . . . .	34
<b>4</b>	<b>Implementation</b>	<b>37</b>
4.1	Konfigurierbare Tastenbelegung und Sensitivität . . . . .	37
4.2	Userinterfacenavigation . . . . .	39
4.3	Lesbarkeit von Texten . . . . .	40
4.4	Anpassbare Textgröße . . . . .	41
4.5	Screenreader Kompatibilität . . . . .	42
4.6	Untertitel . . . . .	45
4.7	Variable Lautstärke . . . . .	47
4.8	Accessibility Informationen & Feedback . . . . .	48
4.9	Persistente Optionen . . . . .	49
<b>5</b>	<b>Evaluation und Interpretation der Ergebnisse</b>	<b>51</b>
5.1	Usability . . . . .	51
5.1.1	Godot Wild Jam . . . . .	51
5.1.2	Studierende . . . . .	54
5.2	Erfüllung des Anforderungskatalogs . . . . .	66
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>72</b>
6.1	Ergebnisse . . . . .	72
6.2	Reflexion . . . . .	74
6.2.1	Evaluation . . . . .	74
6.2.2	Implementation . . . . .	76
6.3	Ausblick . . . . .	78
	<b>Literaturverzeichnis</b>	<b>81</b>
	<b>Anhang</b>	<b>89</b>
	<b>A Anforderungskatalog</b>	<b>89</b>
	<b>B Fragebogen: Godot Wild Jam</b>	<b>93</b>
	<b>C Fragebogen: Gruppe A</b>	<b>102</b>
	<b>D Fragebogen: Gruppe B</b>	<b>111</b>
	<b>E Aufgaben Gruppe A &amp; B</b>	<b>120</b>

---

<b>F Antworten: Gruppe A</b>	<b>124</b>
<b>G Antworten: Gruppe B</b>	<b>127</b>
<b>H Offene Antworten</b>	<b>130</b>
<b>I Keybinding Plugin</b>	<b>131</b>
<b>Eidesstattliche Erklärung</b>	<b>133</b>

# Abbildungs- und Tabellenverzeichnis

## Abbildungsverzeichnis

2.1 Accessible Player Experience Pyramide aus [Inc22]	7
2.2 Kreislauf der Anforderungsanalyse aus [Tre22, S. 5]	9
4.1 REQ 011, REQ 054: Konfigurierbare Tastenbelegung, persistente Einstellungen	38
4.2 REQ 013, REQ 054: konfigurierbare Sensitivität, persistente Einstellungen	39
4.3 REQ 012, REQ 022, REQ 034: Userinterfacenavigation, leicht lesbare Schrift, leicht lesbare Schriftgröße	40
4.4 Darstellungsfehler von Texten	41
4.5 REQ 041, REQ 044: grundlegende Untertitel, leicht lesbare Untertitel	46
4.6 Klassendiagramm Untertitelsystem	47
4.7 REQ 042: konfigurierbare Lautstärke	48
4.8 REQ 052. REQ 055: Accessibility Informationen außerhalb des Spiels, Feedbackmöglichkeit	49
4.9 REQ 053. REQ 055: Accessibility Informationen innerhalb des Spiels, Feedbackmöglichkeit	49
5.1 Soziodemografische Antworten	58
5.2 Nutzung von Accessibility Features	59
5.3 Bewertung der Aufgaben	59
5.4 REQ 011, REQ 054: Konfigurierbare Tastenbelegung, persistente Einstellungen	67
5.5 REQ 012, REQ 022, REQ 034: Userinterfacenavigation, leicht lesbare Schrift, leicht lesbare Schriftgröße	68
5.6 REQ 013, REQ 054: konfigurierbare Sensitivität, persistente Einstellungen	68
5.7 REQ 041, REQ 044: grundlegende Untertitel, leicht lesbare Untertitel	69
5.8 REQ 042: konfigurierbare Lautstärke	70
5.9 REQ 052. REQ 055: Accessibility Informationen außerhalb des Spiels, Feedbackmöglichkeit	71
5.10REQ 053. REQ 055: Accessibility Informationen innerhalb des Spiels, Feedbackmöglichkeit	71

# Tabellenverzeichnis

3.1	gekürzter Anforderungskatalog Accessibility Features . . . . .	14
3.2	Laufzeit und Auswertung für 100 Elemente pro Datenstruktur . . . . .	30
3.3	Laufzeit und Auswertung für 1000 Elemente pro Datenstruktur . . . . .	31
5.1	Median der Antworten, getrennt in Aufgabenteil A und B . . . . .	60
5.2	Kolmogorov Smirnov Test für Aufgabenteil A . . . . .	61
5.3	Kolmogorov Smirnov Test für Aufgabenteil B . . . . .	61
5.4	H1: Wilcoxon Test . . . . .	61
5.5	H2: Rangkorrelation nach Spearman für Aufgabenteil A . . . . .	62
5.6	H2: Rangkorrelation nach Spearman für Aufgabenteil B . . . . .	62
5.7	Median der verbleibenden Antworten . . . . .	64
A.1	Anforderungskatalog Accessibility Features . . . . .	89
F.1	Kodierung der binären Fragen . . . . .	124
F.2	Kodierung der Wissensfragen . . . . .	124
F.3	Kodierung der Likert Skalen . . . . .	124
F.4	Antworten von Gruppe A . . . . .	126
G.1	Kodierung der binären Fragen . . . . .	127
G.2	Kodierung der Wissensfragen . . . . .	127
G.3	Kodierung der Likert Skalen . . . . .	127



# 1 Einleitung

Spielen ist ein Teil der modernen menschlichen Gesellschaft, an dem eine signifikante Menge der Bevölkerung teilnimmt [Fri15, S. 9]. Hierbei sind nicht nur Video und Brettspiele die einzigen Arten, um sich spielerisch auszuleben. Ebenfalls lassen sich spielerische Elemente in anderen Umfeldern finden. Hierzu zählen zum Beispiel Spielshows im Fernsehen, Serious Games und Freizeitparks [Fri15, S. 9]. Ein Aspekt, der jedoch häufig wenig Beachtung findet, ist die Barrierefreiheit. Diese wird in der Entwicklung von Videospiele oft initial nicht bedacht und neue Erkenntnisse in dem Feld der Mensch Maschine Schnittstelle werden nicht berücksichtigt [Agu+20, S. 172, 185]. Dadurch werden Möglichkeiten, durch die Spiele auf unterschiedliche Arten erlebt und mit ihnen interagiert werden können, außer Acht gelassen. Dies erschwert Menschen mit Behinderungen die Teilnahme am Spiel, obwohl teilweise bereits minimaler Aufwand diese Barriere überwinden würde. So kann bereits die Wahl einer leicht lesbaren Schriftart, mit einer ausreichend großen Schriftgröße dazu führen, dass der mentale Aufwand für Nutzer\*innen mit eingeschränkter Sicht signifikant reduziert wird [BMM11, S. 138]. Es darf aber nicht vergessen werden, dass, auch wenn Menschen mit Behinderungen eine Minderheit sind, dass Menschen im Laufe ihres Lebens Teil dieser Minderheit werden können [Ken15].

Only fifteen per cent of people with disabilities are born with them. All people should be seen as only temporarily able bodied. [Ken15]

Mit diesem Zitat bringt es Kent sehr eindeutig auf den Punkt, dass Barrierefreiheit nicht nur für eine kleine Zielgruppe relevant ist, sondern potenziell für jeden relevant werden kann. Vicente und López stellen in ihrer Forschung außerdem dar, dass der Anteil an Menschen mit Behinderungen wachsend ist. Als Grund hierfür nennen sie den Anstieg des Altersdurchschnittes und Umweltfaktoren [VL10].

Barrierefreiheit ermöglicht einigen Menschen überhaupt erst Programme, Spiele und Dienste zu verwenden. Außerdem helfen Maßnahmen der Barrierefreiheit auch Menschen, ohne Behinderungen dabei, Informationen einfach aufzunehmen. Ein einfaches Beispiel hierfür sind Untertitel in Filmen, die es zum einen ermöglichen, diese zu verstehen, wenn es keine Synchronisation in einer Sprache gibt, die man versteht. Aber selbst, wenn man die Sprache versteht, hilft es den Inhalt einfacher wahrzunehmen, wenn zusätzlich Untertitel genutzt werden, da die Aussprache nicht immer klar ist und teilweise Dialekte gesprochen werden, die vor allem für Nicht Muttersprachler schwer verständlich sind [KHM13].

Im Web sowie in Filmen und Serien gibt es bereits etablierte Standards, wie Barrierefreiheit gestaltet wird [Cona; BBC22]. Aufgrund dieser Standards ist es für gewöhnlich einfach, Maßnahmen der Barrierefreiheit einzubauen. Zum Beispiel benötigen alternative Texte für Bilder nur einen HTML Tag, um implementiert zu werden. In interaktiven Medien, wie Videospielen, gibt es diese Standards jedoch noch nicht. Es gibt zwar Vorschläge, was getan werden kann, um eine möglichst hohe Barrierefreiheit zu gewährleisten [Inc22; Ell+]. Diese Vorlagen sind aber keine Standards, die etabliert sind.

## 1.1 Zielstellung

Basierend auf diesen identifizierten Lücken, soll im Rahmen dieser Arbeit ein Programm entwickelt werden, das eine Sammlung mehrerer Systeme ist, die Barrierefreiheit ermöglichen. Hierfür wird eine Sammlung an Barrierefreiheitsmaßnahmen, namens „Game Accessibility Guidelines“ als Vorlage genommen [Ell+]. Die Verwendung dieser Systeme soll dabei so simpel gestaltet sein, wie möglich. Damit der Nutzungsaufwand für Entwickler\*innen diese nicht davon abhält, die implementierten Systeme in ihre eigenen Programme aufzunehmen. Bei dem Beispiel von alternativem Text zur Beschreibung von Grafiken bedeutet das, dass Nutzer\*innen der Systeme nur den Text eingeben müssen, damit dieser automatisch verwendet wird. Mit diesem entwickelten Softwarepaket soll daraufhin die folgende Forschungsfrage beantwortet werden. Entspricht der Aufwand der Implementation, der Einordnung, die in den Game Accessibility Guidelines getroffen wurde? Wie hoch ist der Aufwand für Entwickler\*innen bei der Verwendung des Softwarepakets? Zur Beantwortung der Fragen wird eine quantitative Untersuchung durchgeführt, bei der Nutzer\*innen des Softwarepakets befragt werden, wie sie den Nutzungsaufwand und die Verständlichkeit einschätzen. Außerdem werden alternative Ansätze inhaltlich untersucht, um mögliche Verbesserungen zu erkennen.

## 1.2 Aufbau

Nachdem nun das Ziel dieser Arbeit definiert wurde, wird hier beschrieben, wie der vorliegende Text aufgebaut ist. Im 2. Kapitel werden grundlegende Begriffe und Konzepte zur Barrierefreiheit, Barrierefreiheitsmaßnahmen, Anforderungsanalysen und Softwaredesignpatterns erklärt. Im darauf folgenden Kapitel 3 wird der Anforderungskatalog für die Software, die im Rahmen dieser Arbeit entwickelt wird, besprochen. Des Weiteren wird die Auswahl der genutzten Softwareprodukte erklärt. Außerdem werden Konzepte zur Umsetzung des Anforderungskatalogs beschrieben und Alternativen untersucht. Danach wird in Kapitel 4 die explizite Implementation der Barrierefreiheitsmaßnahmen untersucht. Hierbei wird beschrieben, wie die Implementation durchgeführt und welche Erkenntnisse dabei gefunden wurden. Anschlie



---

ßend wird in Kapitel 5 der Ablauf der Evaluation beschrieben. Außerdem werden die Resultate der Umfragen dargestellt und statistisch betrachtet. Des Weiteren wird untersucht, inwieweit der Anforderungskatalog erfüllt wurde. Abschließend werden in Kapitel 6 die Implementationsansätze aus Kapitel 4, sowie die Ergebnisse der Umfrage aus Kapitel 5 reflektiert. Hierbei werden alternative Lösungsansätze, sowie Empfehlungen für zukünftige Forschungsthemen vorgestellt.

## 2 Grundlagen

In diesem Kapitel werden Begriffe und Konzepte erläutert, die zur Entwicklung von barrierefreier Software wichtig ist. Allem voran, was Barrierefreiheit ist. Außerdem werden bestehende Standards im Internet und bei Videospiele betrachtet. Danach wird beschrieben, welche Aspekte Teil einer Anforderungsanalyse sind und wie diese ausgeführt wird. Abschließend werden Software design patterns erklärt, die in den Kapiteln 3.3 und 4 genutzt oder betrachtet werden.

### 2.1 Accessibility

Das Behindertengleichstellungsgesetz versteht unter **Menschen mit Behinderung** „Menschen, die langfristige körperliche, seelische, geistige oder Sinnesbeeinträchtigungen haben, welche sie in Wechselwirkung mit verschiedenen Barrieren an der vollen, wirksamen und gleichberechtigten Teilhabe an der Gesellschaft hindern können.“ [Deu23, §3 Absatz 1]. Des Weiteren besagt es, dass Produkte und Dienste **barrierefrei** sind, wenn sie von Menschen mit Behinderungen ohne besondere Erschwernisse und ohne die Hilfe anderer Personen zugänglich und nutzbar sind [Deu23, §4 Satz 1].

Richtlinien, die sich aus der Forderung nach Barrierefreiheit ergeben, sind in der Kommunikationshilfenverordnung, der Verordnung über barrierefreie Dokumente in der Bundesverwaltung und in der Barrierefreien Informationstechnik Verordnung (BITV) konkretisiert worden. Die BITV beschäftigt sich mit öffentlichen Internetauftritten und angeboten. Deshalb wird sie hier im Gegensatz zu den beiden anderen Verordnungen weiter betrachtet. Laut der Informationsseite zur UN Behindertenrechtskonvention [Men13] basiert die BITV auf den international anerkannten Zugangsrichtlinien für Webinhalte (WCAG3).

Für Softwareprodukte kann ein **universelles Design** angestrebt werden. Laut der UN Behindertenrechtskonvention „bedeutet ‚universelles Design‘ ein Design von Produkten, Umfeldern, Programmen und Dienstleistungen in der Weise, dass sie von allen Menschen möglichst weitgehend ohne eine Anpassung oder ein spezielles Design genutzt werden können. ‚Universelles Design‘ schließt Hilfsmittel, für bestimmte Gruppen von Menschen mit Behinderungen, soweit sie benötigt werden, nicht aus.“ [§2 Absatz 6 UN Behindertenrechtskonvention].

Software ist mit **Accessibility Features** designt, wenn der Zugang zu dem Programm und die Nutzung für den größten Teil der Nutzer\*innen einfach gestaltet ist. Hierzu zählen ebenfalls Nutzer\*innen mit Behinderungen oder Einschränkungen. Durch die Implementation von Accessibility Features wird oft auch die Nutzbarkeit für Menschen ohne Behinderungen erhöht [hic21].

Damit die Barrierefreiheit bei der Entwicklung von Software grundlegend gewährleistet werden kann, gibt es Richtlinien und Vorschläge, die implementiert werden können. Diese Richtlinien werden als **Accessibility Guidelines** betitelt. Im Internet gibt es bereits etablierte Standards, wie das WCAG 2.1 [Cona], die in angepasster Form sogar in Europa als Norm EN 301 [Cou21] für öffentliche Institutionen verbindlich sind. Für Videospiele gibt es jedoch noch nicht solche etablierten Standards. Stattdessen gibt es populäre Guidelines, wie zum Beispiel die Game Accessibility Guidelines [Ell+] und Accessible Player Experience [Inc22]. Diese werden nachfolgend genauer betrachtet.

### 2.1.1 Accessibility Guidelines im Internet

Vorlagen zur Barrierefreiheit gibt es im Internet schon seit mindestens Mai 1999 in Form der Web Content Accessibility Guidelines (WCAG) [Cal+08, S. 1f]. Die aktuelle Version ist WCAG 2.1. In Europa gilt die Norm EN 301 549, um festzulegen, ab wann eine Website als barrierefrei angesehen wird. Diese nutzt die WCAG 2.1 als Grundlage [Cou21, S. 10]. Die WCAG 2.1 ist unterteilt in Wahrnehmung, Steuerung, Verständnis und Robustheit [Cona]. In diesen Kategorien sind explizite Vorschläge aufgeführt, die jeweils fünf Aspekte behandeln. Hierbei handelt es sich zum einen um eine Beschreibung, die ausführliche Informationen über die Anforderung gibt. Außerdem werden Möglichkeiten und Lösungen vorgestellt, mit denen der jeweilige Vorschlag umgesetzt werden kann. Des Weiteren gibt es ausführliche Hinweise, die tiefgründiges Wissen zu den jeweiligen Features vermitteln. Zudem wird eine ausführliche Liste an Fehlerkriterien vorgestellt, mit der spezifische, häufig auftretende Fehlerquellen überprüft werden können. Außerdem gibt es die Möglichkeit, eine sehr ausführliche Erläuterung der Anforderung einzusehen, in der die genauen Gründe, weitere Informationen und Beispiele vermittelt werden.

Die Richtlinien der WCAG beinhalten vier Themengebiete. Wahrnehmung, Steuerung, Verständnis und Robustheit. Nachfolgend wird beispielhaft ein Teil der Accessibility Features aufgelistet, die in diesen Themengebieten eingeordnet sind. In der Wahrnehmungskategorie werden alternativer Text, Alternativen und Unterstützungen zu zeitlichen Medien (Audio und Video), Möglichkeiten, Inhalte auf alternative Arten darzustellen und Unterscheidungshilfen betrachtet. Derweil befasst sich der Steuerungsaspekt unter anderem mit der barrierefreien Nutzung durch eine Tastatur, aus

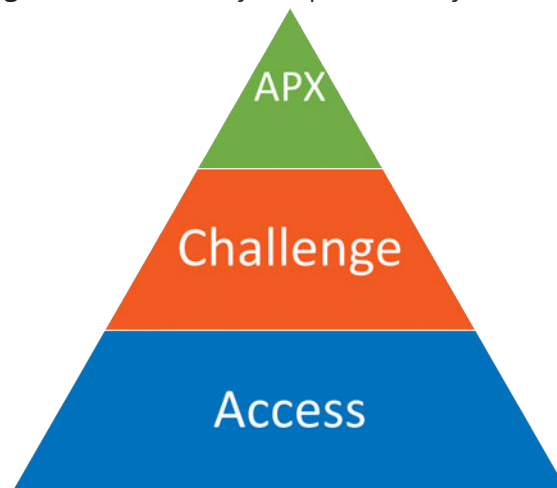
reichender Zeit, um Informationen und Texte zu lesen und zu verarbeiten, der Vermeidung von Anfällen oder anderen physischen Reaktionen, einer einfachen Navigation und alternativen Eingabemöglichkeiten. Der Abschnitt über das Verständnis behandelt vor allem die Lesbarkeit des Textes auf Grundlage von Sprache und Formulierungen, der Vorhersagbarkeit des Verhaltens der Website und der Unterstützung bei Eingaben, damit Fehler vermieden oder korrigiert werden. Den Abschluss macht die Robustheit, die beschreibt, worauf Acht gegeben werden muss, damit die Seite nicht fehlerhaft ist und dadurch von Assistenzsoftware, wie zum Beispiel Screenreadern gelesen werden kann.

### 2.1.2 Accessibility Guidelines in Videospiele

In Videospiele gibt es so wie in der physischen Welt Vorschläge und Richtlinien, um sie inklusiv zu gestalten. Beispiele hierfür stellen die „Game Accessibility Guidelines“ [Inc22], die „Accessible Player Experiences (APX)“ [Inc22], sowie „Includification von AbleGamers“ [BS12] dar, die an dieser Stelle näher vorgestellt werden sollen.

Die „**Game Accessibility Guidelines**“ [Inc22] unterscheiden die Maßnahmen in die Komplexität der Implementation (einfach, mittel und fortgeschritten) und dazu, welche Art der Barrieren überwunden werden sollen (Motorik, Verständnis, Darstellung, Ton, Sprache, Allgemeines). Der Bereich der Motorik umfasst die Steuerung des Spiels, die Navigation in Menüs, Anpassungen der Eingabemöglichkeiten und alternative Eingabemöglichkeiten. Beim Verständnis geht es um die Klarheit von Texten, die Reduzierung der Cognitive Load für Spieler\*innen, der alternativen Darstellung von Informationen und Optionen zur einfacheren mentalen Verarbeitung des Spiels. Die Darstellung umfasst Hinweise, wie Unwohlsein durch zum Beispiel Simulation Sickness vermieden werden, die Darstellung des User Interfaces und alternative Navigationsmöglichkeiten mittels Audio. Bei dem Bereich des Tons wird vor allem darauf eingegangen, wie Untertitel dargestellt werden sollten und dass diese eingebaut werden sollten. Außerdem werden Kommunikationsmöglichkeiten in Mehrspieler Spielen betrachtet. Der Sprachbereich umfasst vor allem den Aspekt der Kommunikation mit anderen Spieler\*innen. Im allgemeinen Bereich werden noch weitere Vorschläge vorgestellt, die nicht in die anderen Bereiche passen. Hierbei handelt es sich vor allem um Optionen zur Steuerung der Schwierigkeit, der Kommunikation von Accessibility Optionen und Assistenzfeatures.

Einige der Vorschläge überschneiden mehrere Themenbereiche und sind deswegen mehrmals vertreten. Ein Beispiel ist, dass ein Textchat alternativ zu einem Sprachchat in Mehrspieler Spielen vorhanden sein sollte. Dieser Vorschlag findet sich im Ton und im Sprachbereich.

**Abbildung 2.1:** Accessible Player Experience – Pyramide aus [Inc22]

Eine weitere Vorlage für Accessibility Features ist die „**Accessible Player Experience (APX)**“ [Inc22]. Diese ist als Pyramide, analog zur Maslowschen Bedürfnishierarchie aufgebaut [AH43]. Diese ist in Grafik 2.1 zu sehen. Die unterste Stufe heißt „Access“ und beinhaltet Accessibility Features, die nötig sind, damit Spieler\*innen das Spiel wahrnehmen und mit dem Spiel interagieren können. Danach kommt die „Challenge“ Stufe. Hierbei geht es um Merkmale, die es ermöglichen, die Schwierigkeit des Spiels auf die persönlichen Bedürfnisse anzupassen. Abschließend kommt die „APX“ Stufe. Diese beinhaltet keine weiteren Vorschläge, sondern sagt aus, dass nur, wenn die beiden vorherigen Stufen erfüllt sind, Spieler\*innen wirklich eine barrierefreie Spieler\*innenerfahrung haben können (engl.: Accessible Player Experience).

## 2.2 Anforderungsanalyse

Nachdem im letzten Unterkapitel die Accessibility Features und Guidelines besprochen wurden, wendet sich dieses Unterkapitel der Anforderungsanalyse zu. Diese wird als Grundlage dienen, um die Guidelines in Kapitel 3.1 in spezifische Aufgabengebiete zu unterteilen und später in Kapitel 4 zu bearbeiten.

Eine Anforderungsanalyse bezeichnet das Ermitteln, die Analyse und die Dokumentation, sowie das Prüfen und Abstimmen von Anforderungen zur Vervollständigung eines Softwareprojektes [Tre22, S. 5f]. Mit einer Anforderungsanalyse wird versucht, einigen Fehlerquellen in der Softwareentwicklung entgegenzuwirken, die offensichtlichsten sind fehlende Anforderungen [Tre22, S. 2]. Dies führt zu Missverständnissen und Unübersichtlichkeit während der Entwicklung und zwischen Kunden und Entwickler\*innen. Ein weiteres Problem sind implizite Anforderungen. Hierbei handelt es sich um Anforderungen, die nicht aufgeschrieben wurden, sondern auf implizitem Wissen basieren. Solche Anforderungen, die als „selbstverständlich“ angesehen werden, treten vor allem bei länder- und kulturübergreifenden Projekten auf. Dem ganzen

Problem sollte entgegengearbeitet werden, indem möglichst viele Anforderungen explizit gemacht werden. Außerdem kann ein Teil des Projektbudgets für implizite Anforderungen eingeplant werden. Des Weiteren sollte bei der Verschriftlichung von Anforderungen darauf geachtet werden, dass diese so gründlich festzulegen und mit den Kunden zu validieren ist, dass Missverständnisse größtmöglich vermieden werden.

Um eine gute Anforderungsanalyse zu erstellen, muss außerdem darauf geachtet werden, unechte Anforderungen auszuschließen [Tre22, S. 3]. Hierbei handelt es sich um solche, die entweder Wünsche und keine richtigen Geschäftsziele darstellen oder welche, die nicht überprüft werden können. Ein Beispiel für letzteres wäre, dass ein System „benutzerfreundlich“ gestaltet werden soll. Hier muss stattdessen definiert werden, welche Anforderungen an das System gestellt werden, damit es als benutzerfreundlich angesehen werden kann.

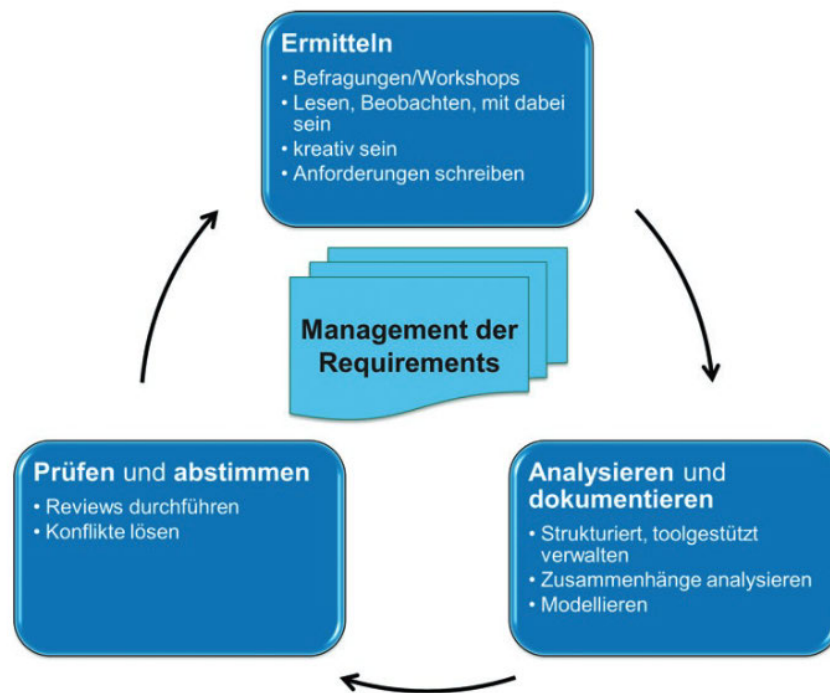
### **2.2.1 Ablauf der Anforderungsanalyse**

Eine Anforderungsanalyse wird in drei Schritte unterteilt, die in einem Kreislauf ablaufen [Tre22, S. 5f]. Dies ist in Grafik 2.2 zu sehen. Diese Schritte bestehen aus „ermitteln“, „analysieren und dokumentieren“ und „prüfen und abstimmen“. Zuerst werden die Quellen für die Anforderungen analysiert. Danach wird das erhobene Material gesichtet und strukturiert. Dabei werden auch Zusammenhänge analysiert und visualisiert. In diesem Schritt bietet sich die Verwendung der Unified Modeling Language (UML) an, um bestimmte Aspekte zu modellieren und die Kommunikation so zu erleichtern. Danach werden die erstellten Artefakte aus der Anforderungsanalyse geprüft. Hierbei wird untersucht, ob die gewählten Methoden und Techniken korrekt angewandt wurden. Außerdem wird mit den betreffenden Stakeholdern untersucht, ob die Artefakte korrekt verstanden wurden.

### **2.2.2 Anforderungskatalog**

Im letzten Unterkapitel wurde erläutert, wie eine Anforderungsanalyse durchgeführt wird. Nun soll das Resultat dieser, ein Anforderungskatalog betrachtet werden. Ein Anforderungskatalog besteht aus funktionalen und nicht funktionalen Anforderungen [Tre22, S. 82]. Funktionale Anforderungen beschreiben die Aufgaben, die das geplante Softwareprodukt lösen soll. Bei nicht funktionalen Anforderungen handelt es sich um Antworten auf die Frage, wie ein System die Leistung erbringen soll. Es handelt sich hierbei um Ansprüche an die Qualität.

Der Anforderungskatalog wird mit mehreren Attributen versehen. Darunter eine Identifikation, eine Beschreibung, eine Kategorisierung, die Verfolgbarkeit, Informationen zur Realisierung, eine Version und Informationen zum Workflow [Tre22, S.

**Abbildung 2.2:** Kreislauf der Anforderungsanalyse aus [Tre22, S. 5]

82 89]. Die Identifikation ist eine Nummer, die für gewöhnlich von Anforderungsmanagementtools automatisch vergeben wird. Sie wird genutzt, um die jeweilige Anforderung eindeutig zu identifizieren. Wenn eine Anforderung obsolet wird, wird sie mit dem Status „obsolet“ versehen. Die Identifikationsnummer bleibt jedoch erhalten. Hierarchien können hierbei auch kenntlich gemacht werden. Eine Beschreibung wird für gewöhnlich in einer natürlichen Sprache verfasst. Dadurch soll eine Verständigung mit Stakeholdern möglich sein. Sie beschreibt in Stichworten den Inhalt der Anforderung. Bei komplexen Sachverhalten kann es von Vorteil sein, eine zusätzliche Kurzbeschreibung zu verfassen, die die Anforderung prägnant zusammenfasst. Außerdem können Erläuterungen, Kommentare und eine Begründung bzw. Business Value verfasst werden. Anforderungen werden mithilfe von Kategorien nach unterschiedlichen Aspekten sortiert. Beispiele hierfür sind die Art der Anforderung, die Verbindlichkeit (Muss, Kann, Nice to have), die Komplexität, die Kritikalität und die Vertraulichkeit. Ein weiterer Teil einer Anforderung ist die Verfolgbarkeit. Hierbei werden Referenzen auf andere Dokumente, Anforderungen oder Stakeholder genommen. Die Realisierung beinhaltet Informationen darüber, zu welchem Produkt eine Anforderung gehört, wie hoch der geschätzte Aufwand ist und zur Priorisierung. Außerdem wird in agilen Projekten eine Zugehörigkeit zu einem Release/ Sprint festgehalten. Des Weiteren wird der Status der Realisierung dokumentiert und im Laufe des Projektes aktualisiert. Dieser fängt an bei offenen Aufgaben, die daraufhin geplant und bestätigt werden und dann freigegeben werden. Diese werden dann für einen Release/ Sprint geplant und später realisiert. Im Laufe der Entwicklung werden

mehrere Versionen in einer Anforderung vermerkt. Diese beinhalten jeweils eine Versionsnummer, einen Autor, ein Erstellungsdatum und den aktuellen Status (in Arbeit, beendet, geprüft, obsolet). Ein weiteres Attribut einer Anforderung ist der Workflow. Hierbei werden die Tätigkeiten und Termine, die zur Vervollständigung der Arbeit nötig sind, aufgeführt.

## 2.3 Softwaredesignpatterns und Softwaregrundlagen

Softwaredesignpatterns sind Lösungen für spezifische Probleme im objektorientierten Softwaredesign. Sie stellen Lösungen dar, die sich über die Zeit entwickelt haben und dadurch elegant und simpel sind. Das Resultat sind flexible und wiederverwendbare Designs [Gam+09, S. xi].

### 2.3.1 Softwaredesignpatterns der „Gang of Four“

Das Buch „Design Patterns – Elements of Reusable Object Oriented Software“ von Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides beschreibt viele Software Pattern, mit denen objektorientierte Programme organisiert werden können [Gam+09]. Die Autoren werden häufig als „Gang of Four“ bzw. „GoF“ betitelt. Diese Design Patterns werden seitdem als Lösungen für viele Probleme in der Softwareentwicklung verwendet. Als quasi Goldstandard haben sie noch immer eine hohe Relevanz [AHA21, S. 1]. In den nachfolgenden Unterkapiteln wird auf einige der Pattern eingegangen. Auf diese wird sich später im Implementationsteil (Kapitel 4) bezogen.

#### Singleton Pattern

Das Singleton Pattern beschreibt eine Klasse, für die es immer nur exakt eine Instanz geben darf und die global erreicht werden kann [Gam+09, S. 127]. Hierfür stellt die Klasse selbstständig sicher, dass sie im kompletten Programm nur ein Mal instanziiert werden kann und neue Instanzierungen unterbunden werden. Der Zugriff auf die einzelne Singleton Instanz wird für gewöhnlich durch eine globale, statische Variable in der Klasse gelöst.

#### Observer Pattern

Das Observer Pattern beschreibt eine Eins zu Viele Beziehung zwischen Objekten, bei dem Änderungen des einen Objektes an die vielen Objekte übermittelt werden. Diese können dann selbstständig auf diese Aktualisierung reagieren [Gam+09, S. 293]. Hiermit können Systeme lose miteinander gekoppelt werden. Ein Beispiel hierfür wäre ein Programm, in dem Daten in eine Tabelle eingetragen und visualisiert werden. Die Tabelle ist dann das Subjekt, dass der Visualisierung, dem Observer, mitteilt, wenn Daten geändert werden. Der Tabelle ist dabei egal, wie viele und welche Observer es gibt, sondern nur, dass die Änderungen mitgeteilt werden.



### 2.3.2 Game Programming Patterns

Für die Entwicklung von Videospiele beziehungsweise Game Engines, stellt Robert Nystrom in seinem Buch „Game Programming Patterns“ [Nys14] eine Auswahl der Patterns der Gang of Four vor. Des Weiteren beschreibt er zusätzliche Patterns, die beim Design der Softwarearchitektur von Videospiele genutzt werden können. Ein paar dieser weiteren Patterns sind für diese Arbeit relevant und werden deswegen nachfolgend erklärt.

#### Service Locator

Ein Service Locator [Nys14] soll einen globalen Zugriff zu einem Service ermöglichen, ohne dass dabei die spezifische Implementation gekoppelt wird. Hierfür wird ein abstraktes Interface genutzt, das von dem spezifischen Service implementiert wird. Außerdem wird ein Service Locator erstellt, der eine Referenz auf den Service zur Verfügung stellt. Ein Service Locator stellt eine Alternative für ein Singleton dar, bei dem die Implementation leichter angepasst werden kann. Im Gegensatz dazu, wird der Code komplexer, da zum einen der spezifische Service gefunden werden muss und zum anderen die Möglichkeit gehandhabt werden muss, dass noch kein Service gefunden wurde. Zum Registrieren einer Serviceimplementation kann sogenannte „dependency injecte“ genutzt werden. Hierbei wird die Abhängigkeit des Locators von der Implementation dadurch gehandhabt, dass die Implementation sich selbst beim Locator registriert.

Der Grenzfall, dass noch kein Service bei dem Service Locator registriert ist, kann auf unterschiedliche Arten behandelt werden. Welche dabei gewählt werden sollte, ist von dem spezifischen Anwendungsfall abhängig. Zum einen kann ein Null Service implementiert werden. Dieser implementiert das Interface, aber besitzt keine Funktionalität innerhalb der Methoden. Hierbei wird die Möglichkeit gegeben, dass das Programm weiter ausgeführt wird, ohne mitzubekommen, dass der Service noch nicht läuft. Alternativ kann dies auch genutzt werden, um einen Service temporär zu deaktivieren. Hierbei muss jedoch beachtet werden, dass es im Code nicht ersichtlich ist, ob eine vollständige Implementation oder nur ein Null Service angesprochen wird. Dadurch kann es passieren, dass Informationen verloren gehen ohne, dass dies erkannt wird. Alternativ können drastischere Maßnahmen getroffen werden. So kann die Handhabung einer fehlenden Implementation weitergegeben werden, indem ein Null Objekt vom Locator zurückgegeben wird. Alternativ kann das Programm komplett angehalten werden.

## Event-Queue

Mit einer Event Queue [Nys14] können Nachrichten empfangen und versandt werden. Hierbei werden die Sender und Empfänger voneinander zeitlich entkoppelt. Neue Nachrichten werden an die Queue angehängt und dann nach dem First in, First out Prinzip verarbeitet. Das heißt, dass zuerst die ältesten Nachrichten verarbeitet werden, bevor neuere beachtet werden. Durch die Entkopplung bringen Event Queues klare Vorteile gegenüber anderen Pattern, wie dem Observer dar, die Systeme nur lose mit sich koppelt. Gleichzeitig wird die Komplexität des Codes stark erhöht und neue Probleme müssen bedacht werden. Zum einen sind zentrale Event Queues sehr ähnlich zu globalen Variablen und bringen dadurch Risiken wie Laufzeit fehler mit sich. Zum anderen kann es zu Problemen kommen, wenn die Nachrichten veraltet werden und sich dadurch potenziell auf Objekte beziehen, die nicht mehr vorhanden sind. Außerdem kann es passieren, dass Schleifen in die Queue eingebaut werden, wenn Systeme einander zyklisch benachrichtigen.

## Single Responsibility Prinzip

Außer Software Patterns gibt es auch verschiedene Prinzipien, die in der Programmierung formuliert werden. Diese stellen keine spezifischen Lösungen von allgemeinen Problemen dar, sondern Leitsätze, die helfen sollen, bestimmte Problemquellen zu minimieren beziehungsweise frühzeitig zu erkennen und eine hohe Modularität zu fördern.

Das Single Responsibility Prinzip ist ein Grundsatz der objektorientierten Programmierung. Es wurde 2005 von Robert C. Martin formuliert. Hierbei handelt es sich um einen von vielen Leitsätzen, die bei der Entwicklung von sauberem, modulare Code helfen. Dieses spezifische Prinzip besagt, dass eine Klasse nur exakt einen Grund für Veränderungen haben soll [Mar05]. Diese Definition wurde später in einem Blogbeitrag spezifiziert [Mar14]. Dabei wurde erklärt, dass mit dem Grund einzelne Personengruppen mit einem spezifischen Use Case beziehungsweise eine spezifische Aufgabe gemeint ist. Umformuliert bedeutet dies, dass eine Klasse nur eine Aufgabe erfüllen soll.

## 3 Konzept und Analyse

Nachdem im letzten Kapitel erläutert wurde, wie eine Anforderungsanalyse durchgeführt wird, soll in diesem Kapitel zuerst der im Rahmen dieser Arbeit erstellte Anforderungskatalog vorgestellt werden. Hierbei werden zuerst die Anforderungen, die in dieser Arbeit bearbeitet wurden, beschrieben. Daraufhin werden weitere Anforderungen, die in den Game Accessibility Guidelines [Ell+] aufgeführt werden, aber die in dieser Arbeit nicht implementiert werden können, aufgelistet. Außerdem wird in Kapitel 3.1.6 eine Erklärung gegeben, warum diese nicht weiter bearbeitet werden. Daraufhin werden die Programme und Services, die primär in dieser Arbeit genutzt werden, in Kapitel 3.2 dargestellt. Abschließend werden in Kapitel 3.3 Konzepte beschrieben, mit denen die einzelnen, zu bearbeitenden Anforderungen umgesetzt werden sollen. Falls möglich, werden dabei mehrere Ansätze vorgestellt.

### 3.1 Anforderungskatalog

Basierend auf der Beschreibung aus Kapitel 2.2 wurde der Anforderungskatalog in Tabelle 3.1 beziehungsweise in gesamter Form in Tabelle A.1 erstellt. Die Vergabe der Identifikationsnummern erfolgte nach dem folgenden Muster. Alle Anforderungen beginnen mit „REQ “. Darauf folgt die Überkategorie in Form von „01“ bis „05“. Abschließend wird noch die laufende Nummer der spezifischen Anforderung „0“ bis „9“ angefügt. Bei Anforderungen, die die spezifische Nummer „0“ besitzen, handelt es sich um übergeordnete Anforderungen, die der Einordnung der anderen Anforderung genutzt werden.

Bei der Dokumentation der Anforderungen wurden ein paar Anpassungen vorgenommen. Die Verfolgbarkeit wurde nicht verwendet, da sich die Anforderungen nur auf die Game Accessibility Guidelines [Ell+] und WCAG 2.1 [Cona] beziehen. Ebenso wird die Version nicht aufgeführt. Die Entscheidung hierfür liegt darin begründet, dass dieses Dokument im Gegensatz zu einem normalen Anforderungskatalog nur einen finalen Stand des Anforderungskatalogs und der Entwicklung mit diesem darstellen soll. Deswegen ist die Inklusion von mehreren Versionen unnötig. Aus demselben Grund wird auch der Workflow nicht dokumentiert. Außerdem wurde sich dafür entschieden, das Feld „Kategorie“ nicht aufzuführen. Stattdessen wurden übergeordnete Anforderungen aufgenommen, die die untergeordneten Anforderungen in Kategorien unterteilen. Der Grund für diese Änderungen ist, dass der Anforderungskatalog nur separate Accessibility Features beinhaltet und nicht die komplette Planung eines Softwareproduktes darstellt. Da lediglich Anforderungen mit geringer Komplexität betrachtet werden, wird ebenfalls die Komplexität nicht weiter dokumentiert. Die

einzigsten Ausnahmen für diese Einschränkung ist die Anforderung **REQ-038**. Diese wird auf zwei unterschiedlichen Arten in den Game Accessibility Guidelines aufgeführt. Für mobile Geräte wird dieser Aspekt einer mittleren Komplexität zugeordnet. Für andere Geräte dahingegen einer hohen Komplexität. Die Entscheidung, diese trotzdem aufzunehmen, liegt darin, dass die Verwendung von Screenreadern für Personen mit eingeschränkter Sicht teilweise die Nutzung von Websites, Diensten oder Programmen erst ermöglicht. Nachfolgend werden in dem Unterkapitel 3.1 die Anforderungen genauer beschrieben.

**Tabelle 3.1:** gekürzter Anforderungskatalog – Accessibility Features

<b>Identifikation</b>	<b>Titel</b>	<b>Beschreibung</b>	<b>Priorität</b>
REQ 010	motorische Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 011 bis REQ 015 erfüllt sind.	mittel
REQ 020	kognitive Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 021 bis REQ 027 erfüllt sind.	mittel
REQ 030	visuelle Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 031 bis REQ 037 erfüllt sind.	hoch
REQ 040	auditive Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 041 bis REQ 045 erfüllt sind.	hoch
REQ 050	allgemeine Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 051 bis REQ 055 erfüllt sind.	mittel

### 3.1.1 Motorische Anforderungen

Der erste Bereich der Anforderungen wird mit der übergeordneten Anforderung **REQ-010** zu den motorischen Anforderungen gebündelt. Hierbei handelt es sich um Aspekte, die die Steuerung und Navigation des Spiels betreffen. Nachfolgend werden diese detaillierter beschrieben.

### **REQ-011 konfigurierbare Tastenbelegung**

Die erste Anforderung gehört zum Typ der Bewegung und wird mit einem niedrigen Aufwand eingestuft. In ihr wird verlangt, dass die Steuerung Nutzer\*innen die Möglichkeit bietet, diese neu belegen zu können. Diese Anforderung beruht darauf, dass Menschen mit eingeschränkter Motorik bestimmte Tasten und Tastenkombinationen nur schwer bedienen können. Hier hilft dieses Accessibility Feature, dass sich jede Person die Steuerung so anpassen kann, dass sie individuell einfach bedient werden kann. Ein weiterer Punkt, bei dem dieses Feature eine enorme Hilfe darstellt ist, wenn Spiele mit unterschiedlichen Hardwarelayouts design wurden. So ist es zum Beispiel teilweise schwerer ein Spiel mit einer QWERTZ Tastatur zu steuern, als mit einer QWERTY, wenn die Y Taste mit einer Funktion belegt ist.

### **REQ-012 Userinterfacenavigation**

Diese Anforderung gehört ebenfalls zum Typ der Bewegung. Sie verlangt, dass das Userinterface Nutzer\*innen die Möglichkeit bieten muss, alle Interaktionsflächen über dieselbe Eingabemethode, wie die Steuerung des Spiels erreichen zu können. Das heißt, dass ein Spiel, das komplett per Tastatur gespielt werden kann, es ermöglichen muss, alle Menüs ebenfalls komplett per Tastatur zu bedienen. Im Allgemeinen ist es für Nutzer\*innen eine Störung, wenn sie für einen kleinen Teil des Spiels ihre Eingabemethode wechseln müssen. Für Nutzer\*innen mit physischen Einschränkungen kann dieser Wechsel jedoch potenziell nicht möglich sein, wodurch im negativsten Fall das ganze Spiel nicht spielbar ist.

### **REQ-013 konfigurierbare Sensitivität**

Hierbei wird verlangt, dass Nutzer\*innen die Möglichkeit haben, die Sensitivität der Steuerung anzupassen. Neben persönlichen Präferenzen, gibt es auch Menschen mit eingeschränkter Mobilität, die von einer hohen Sensitivität profitieren. Außer dem kann eine geringe Empfindlichkeit potenziell dabei helfen, Simulation Sickness zu reduzieren. Des Weiteren kann eine feste Einstellung problematisch sein, wenn alternative Eingabegeräte, wie zum Beispiel kopf oder augengesteuerte Geräte verwendet werden. Ebenfalls sollte sich die Möglichkeit der Konfiguration nicht nur auf Mäuse und ähnliche Eingabegeräte beziehen, sondern auch auf Controller. Bei diesen kann es zudem von Vorteil sein, auch Null Zonen einstellbar zu gestalten.

## **3.1.2 Kognitive Anforderungen**

Der nächste Bereich, **REQ-020**, sind die kognitiven Anforderungen. Hierunter zählen solche, die die kognitive Belastung für Spieler\*innen minimieren sollen. Diese Aspekte werden nachfolgend genauer definiert.

### **REQ-022 leicht lesbare Schriftart**

Diese Anforderung verlangt, dass die Standardschriftart so gestaltet sein muss, dass sie für Nutzer\*innen leicht lesbar ist. Um dies zu gewährleisten, wird die Schriftart OpenDyslexic empfohlen, da diese versucht eine möglichst hohe Lesbarkeit der Buchstaben zu ermöglichen [Ell+].

### **3.1.3 Visuelle Anforderungen**

Diese Anforderung **REQ-030** beschäftigt sich mit den visuellen Aspekten des Spiels und wie diese aufbereitet werden können, damit sie von Spieler\*innen einfach aufgenommen und verstanden werden können. Die hierbei untersuchten Anforderungen werden nachfolgend beschrieben.

### **REQ-034 leicht lesbare Schriftgröße**

Bei dieser Anforderung wird verlangt, dass die Textgröße so gewählt wird, dass sie leicht lesbar ist. Ideal ist es, wenn die Textgröße konfigurierbar ist. Alternativ ist eine große Größe ein guter erster Schritt. Hierbei muss außerdem betrachtet werden, auf welchen Endgeräten das Spiel gespielt wird, da der Abstand der Spielenden vom Bildschirm im Allgemeinen bei Konsolenspielen höher ist, als bei PC Spielen.

### **REQ-037 konfigurierbare Schriftgröße**

Texte sollten bis auf die doppelte Größe skaliert werden können, ohne dass externe Tools verwendet werden müssen. Hierbei geht es nicht nur darum, dass dies technisch machbar ist, sondern auch darum, dass diese Anpassungen fehlerfrei dargestellt werden können.

### **REQ-038 Screenreader-Kompatibilität**

Texte und Menüs im Spiel sollten so integriert sein, dass sie von gängigen Screenreadern (JAWS, NVDA) vorgelesen werden können. Bei mobilen Endgeräten gibt es hierfür alternativ die Dienste Voiceover (iOS) und TalkBack (Android). Hiermit soll Spieler\*innen mit stark eingeschränkter Sicht ermöglicht werden, die Inhalte des Spiels wahrnehmen zu können.

### **3.1.4 Auditive Anforderungen**

Nachfolgend werden mehrere Anforderungen beschrieben, die der Anforderung **REQ-040** untergeordnet sind. Sie behandeln Aspekte des Spiels, die das Verständnis von auditiven Elementen erhöhen soll und dabei Menschen mit einem eingeschränkten Gehör berücksichtigt.

### **REQ-041 grundlegende Untertitel**

Untertitel müssen für alle wichtigen gesprochenen Texte vorhanden sein. Diese Anforderung bezieht sich auf alle gesprochenen Texte, die essenziell sind, um das Narrativ des Spiels zu erleben oder wichtige Informationen dazu, wie das Spiel gespielt werden sollte enthalten. Dieses Feature wird nicht nur von Menschen mit eingeschränktem Hörvermögen genutzt, sondern es ist auch hilfreich bei schlechten Lautsprechern, einer hohen Umgebungslautstärke, Lokalisierung, die per Untertitel gelöst wurde und vielem mehr.

### **REQ-42 konfigurierbare Lautstärke**

Wichtig ist ebenfalls, dass die Lautstärke von Musik, Sprache und Effekten separat voneinander einstellbar sind. Hierdurch kann die Unterscheidung der unterschiedlichen Informationen vereinfacht werden und die kognitive Überlastung von zu vielen Eindrücken kann reduziert werden. Für manche Spiele kann es sich ebenfalls lohnen, weitere Einstellungsmöglichkeiten wie zum Beispiel die Lautstärke von spezifischen Effekten, die mit dem UI zu tun haben oder von Hintergrundgeräuschen zu implementieren.

### **REQ-44 leicht lesbare Untertitel**

Texte in Untertiteln müssen meist schnell aufgenommen werden. Außerdem läuft meist gleichzeitig eine visuelle Handlung ab, die Spieler\*innen auch aufnehmen wollen. Deswegen ist es wichtig, dass Untertitel leicht lesbar sind. Dafür ist es essenziell, dass die Textgröße ausreichend groß gewählt wird. Außerdem sollten Untertitel vor einem soliden oder semitransparenten Hintergrund dargestellt werden. Idealerweise wird dies mit Umrissen oder Schatten um die Buchstaben kombiniert. Des Weiteren sollte nicht zu viel Text gleichzeitig angezeigt werden. Im Rahmen dieser Arbeit wird die Länge der Texte außer Acht gelassen, da dies zu spezifisch für den jeweiligen Anwendungszweck ist.

## **3.1.5 Allgemeine Anforderungen**

Die Anforderung **REQ-050** bündelt weitere Aspekte der Accessibility, die beachtet werden sollten, aber nicht eindeutig zu einem bestimmten Sinn zugeordnet werden können. Die Anforderungen, die hierunter zählen, werden nachfolgend beschrieben.

### **REQ-052 Accessibility Informationen außerhalb des Spiels**

Für Spieler\*innen ist es wichtig, sich vor einem potenziellen Kauf über das Vorhandensein von spezifischen Accessibility Features informieren zu können. Diese Informationen sollten einfach zugänglich sein und leicht auffindbar sein. Dies kann über Trailer, Aufschriften auf physischen Exemplaren, Informationen auf der eigenen Website oder Tags bei Verkaufsseiten wie zum Beispiel Steam geschehen.

### **REQ-053 Accessibility Informationen innerhalb des Spiels**

Die Informationen über die implementierten Accessibility Features sollten nicht nur außerhalb des Spiels, sondern auch innerhalb kommuniziert werden. Hiermit ist gemeint, dass es wichtig ist zu beachten, dass die Features beziehungsweise Einstellungen so gestaltet sind, dass Spieler\*innen nicht lange nach ihnen suchen müssen. Möglichkeiten der Darstellung sind zum Beispiel die Darstellung der relevanten Optionen vor dem ersten Start des Spiels. Außerdem Hinweise auf die Existenz der Systeme, die eingeblendet werden. Oder sogar explizite Fragen an die Spielenden, ob bestimmte Features aktiviert werden sollen, falls es scheint, als wäre das Spiel zu schwer.

### **REQ-054 persistente Einstellungen**

Einstellungen, die von Spieler\*innen getätigt werden, müssen gespeichert und bei einem Neustart des Spiels wiederhergestellt werden. Vor allem, wenn viele Anpassungen vorgenommen werden, kann es ein sehr langwieriger Prozess sein, die Einstellungen von neuem vorzunehmen.

### **REQ-055 Feedbackmöglichkeit**

Für die erfolgreiche Umsetzung von Accessibility Features sollte das Feedback von Spieler\*innen (vor allem mit Behinderungen) eingeholt werden. Testversionen mit Fragebögen sind eine Option, die während der Entwicklung genutzt werden kann. Es ist außerdem gut, eine einfache Möglichkeit zu bieten, mit der Nutzer\*innen dem Entwicklungsstudio Feedback bezüglich implementierter und fehlender Accessibility Features geben kann. Hierbei sollte darauf geachtet werden, dass die Kontaktaufnahme möglichst einfach und einladend für die Spieler\*innen gestaltet ist.

### **3.1.6 Ausgeschlossene Anforderungen**

Die Anforderungen, die im Rahmen dieser Arbeit untersucht und implementiert wurden sollen, sind nur ein Teil der Vorschläge, die in den Game Accessibility Guidelines verlangt werden. In diesem Unterkapitel wird nun auf die ausgelassenen Anforderungen eingegangen.



Der Hauptgrund dafür, dass ein Vorschlag aus den Game Accessibility Guidelines hier nicht aufgenommen wurde, liegt darin begründet, dass dieser zu spezifisch für das jeweilige Spiel ist, um in der allgemein nutzbaren Vorlage, die im Rahmen dieser Arbeit entwickelt wird, implementiert zu werden. Des Weiteren wird Virtual Reality Technologie nicht betrachtet. Hierunter zählen folgende Anforderungen:

- **REQ-014** simple Steuerung
- **REQ-021** geringer Einstiegsaufwand
- **REQ-025** interaktives Tutorial
- **REQ-033** Simulation Sickness

Andere Anforderungen beziehen sich wiederum auf inhaltliche und grafische Aspekte, die sich nicht auf programmierbare Systeme beziehen. Eine Möglichkeit, diese Gesichtspunkte automatisch überprüfen zu lassen, sprengt in den meisten Fällen den Rahmen dieser Arbeit. Die Anforderungen, die hierunter zählen sind folgende:

- **REQ-023** einfache Sprache
- **REQ-024** klare Formatierung
- **REQ-027** Muster und flackernde Bilder
- **REQ-031** visuelle Informationsgestaltung
- **REQ-035** klar formatierte Textdarstellung
- **REQ-036** hoher visueller Kontrast
- **REQ-043** auditive Informationsgestaltung
- **REQ-045** optionale Spracheingaben
- **REQ-051** individualisierbare Schwierigkeit

## 3.2 Technische Auswahl

Im Rahmen dieser Arbeit werden vorrangig zwei Softwareprodukte genutzt. Zum einen die Godot Game Engine und zum anderen der Clouddienst GitHub. Diese werden nachfolgend kurz beschrieben. Außerdem wird erläutert, warum die Wahl auf diese beiden Tools fiel.

### 3.2.1 Godot Game Engine

Die Godot Engine ist ein Programm, das genutzt werden kann, um Spiele zu entwickeln. Hierbei nutzt es eine eigene Programmiersprache, namens „GDScript“. Die Entscheidung zur Nutzung dieser Game Engine im Gegensatz zu Alternativen, wie Unity und Unreal ist darin begründet, dass zu Beginn dieser Arbeit ein Programm für eine Messe erstellt werden sollte, bei der die primäre Nutzung von Software, die unter der Open Source Lizenz veröffentlicht wurde, erwünscht war. Während der

Entwicklung des Programms wurde die Messe abgesagt. Da zu diesem Zeitpunkt bereits eine signifikante Menge an Arbeit in die Implementationen investiert wurde, fiel die Entscheidung, bei der Engine zu bleiben.

### 3.2.2 GitHub

Der Dienst GitHub kann genutzt werden, um verschiedene Versionen eines Softwareprojektes zu speichern, zwischen ihnen zu wechseln und sie miteinander zu kombinieren. Diese Funktionalität ist vor allem im kollaborativen Arbeiten hilfreich. Des Weiteren kann der Dienst genutzt werden, um Softwareprojekte zu veröffentlichen, damit andere Personen sie nutzen können. In dieser Arbeit wurde sich darauf beschränkt, die Funktionen des Clouddienstes dazu zu nutzen, inkrementelle Entwicklungsschritte zu speichern und die Implementation an Proband\*innen zu verteilen.

## 3.3 Konzept

Im letzten Unterkapitel wurden die Anforderungen, die in dieser Arbeit bearbeitet werden, vorgestellt. Diese entstammen den Game Accessibility Guidelines [Ell+]. Obwohl dort bereits eine eindeutige Bewertung der Implementation in Form einer Einschätzung der Komplexität vorgenommen wurde, beinhalten die Game Accessibility Guidelines keine Anleitungen, die bei einer Umsetzung der Features hilft. Deswegen wird nun untersucht, auf welche Arten diese umgesetzt werden könnten. Hierfür werden die Programming Patterns der Gang of Four [Gam+09] und von Nystrom [Nys14] als Grundlage genommen. Da es keine standardisierten Lösungsansätze für einen Großteil der Anforderungen gibt, werden nachfolgend unterschiedliche Ansätze vorgestellt und diskutiert. Des Weiteren wird der aktuelle Forschungsstand für die Wahl der Lösungsansätze genutzt. In Kapitel 4 wird dann die spezifische gewählte Implementation beschrieben.

### 3.3.1 Konfigurierbare Tastenbelegung und Sensitivität

Nutzer\*innen brauchen die Möglichkeit, die Tastenbelegung nach ihren eigenen Vorlieben und Anforderungen anzupassen (**REQ-011**). Die Godot Engine verwendet ein eigenes Eingabesystem, bei dem es möglich ist in Laufzeit die Tastenbelegungen zu ändern. Dadurch sind zur Erfüllung dieser Anforderung nur zwei Aspekte nötig. Zum einen muss es ein Menü geben, in dem die aktuellen Tastenbelegungen zu sehen sind und in dem diese auch geändert werden können. Zum anderen müssen die angepassten Tastenbelegungen zwischen Sessions gespeichert und geladen werden. Hierfür bietet die Godot Engine aber ebenfalls Möglichkeiten, um arbiträre Daten zu speichern. Ein weiteres System, über das die aktuelle Steuerung abgefragt werden kann, ist nicht nötig, weil das Engine eigene System diese Aufgabe bereits erfüllt.

Ebenfalls benötigen Nutzer\*innen die Möglichkeit, die Sensitivität der Eingabemöglichkeiten zu definieren (**REQ-013**). Dies findet sich unter anderem darin, wie schnell sich die Kamera in First beziehungsweise Third Person Spielen dreht. Hierfür kann eine Variable genutzt werden, die die Sensitivität als Prozentwert festhält. Dann kann bei der Kamerarotation oder anderen Stellen, an denen es sinnvoll wäre, die Eingabe mit der Sensibilität multipliziert werden.

### 3.3.2 Userinterfacenavigation

Nutzer\*innen müssen die Möglichkeit haben, alle Interaktionsflächen des Userinterfaces auf dieselbe Art zu bedienen, wie das Spiel gesteuert wird (**REQ-012**). Im Rahmen dieser Arbeit bedeutet das, dass Menüs nicht nur per Maus, sondern auch per Tastatur und Controller navigierbar sein müssen. Die Entwicklung von Konsolen und Handyspielen wird hierbei nicht beachtet, da die Navigation des Userinterfaces bei diesen aufgrund der inhärenten Eingabemethoden bereits diese Anforderung erfüllen. Konsolenspiele werden für gewöhnlich komplett per Controller gesteuert und Handyspiele, indem der Bildschirm mit Fingern berührt wird. Die Godot Engine hat für die Erfüllung dieser Anforderung bereits eine Lösung eingebaut. Das Userinterface ist grundlegend mit der Tastatur navigierbar. Hierfür muss lediglich festgelegt werden, welches Element als Eingangspunkt für die Interaktion verstanden wird.

Weil in der Godot Engine ein Element immer nur maximal ein Skript besitzen kann, sollen zwei verschiedene Lösungen dieser Anforderung implementiert werden. Die erste, ist der geradlinige Ansatz, bei dem das Skript den Fokus des UI Systems auf das Element setzt, auf dem das Skript liegt. Die andere Lösung bestimmt ein Element, das separat ausgewählt wird als Fokus.

### 3.3.3 Lesbarkeit von Texten

Die Accessibility Features, die in den Game Accessibility Guidelines vorgestellt werden, sind teilweise sehr vage gehalten. So wird empfohlen, eine leicht lesbare, große Textgröße zu wählen (**REQ-022, REQ-034**). Deswegen muss hier zuerst eine Spezifikation vorgenommen werden.

Schriftarten stellen unter anderem für Menschen mit Dyslexie eine Barriere dar. Bestehende Forschungen haben gezeigt, dass der Abstand zwischen Buchstaben eine signifikante Rolle in der Lesbarkeit spielen kann [Jac14, S. 158; Gal+20, S. 149ff]. Deswegen werden serifenfreie Schriftarten, sowie eine Erhöhung des Abstandes zwischen Buchstaben mit einer gleichzeitigen angemessenen Erhöhung des Abstandes zwischen Wörtern empfohlen. Dedizierte Schriftarten, die die Lesbarkeit für Menschen mit Dyslexie verbessern sollen, wurden nicht empfohlen, da sie zum einen nicht immer Anklang finden und zum anderen neue Probleme bringen [Jac14, S. 159f].

Eine Studie untersuchte die Lesegeschwindigkeit und die Genauigkeit beim Lesen von Texten in serifenfreien und Dyslexie spezifischen Schriftarten, sowie den Einfluss von Abstandserhöhungen zwischen Wörtern und Buchstaben. Das Resultat, das in ihr gefunden wurde, ist, dass die dedizierte Schriftart, die getestet wurde, keinen Einfluss auf die Lesbarkeit der Texte hatte [Gal+20, S. 149ff].

Ein weiterer Aspekt, der die Lesbarkeit beeinflusst, ist die Größe der Schrift. Für Fernsehbildschirme wird von Amazon eine minimale Größe von 28 Pixeln bei einem 1080p Bildschirm empfohlen [Ama]. Diese Textgröße kann als Empfehlung für Konsole Spiele verwendet werden, da diese für gewöhnlich auf einem Fernsehbildschirm dargestellt werden, während die Spieler\*innen sich in einer erhöhten Entfernung dazu befinden. Für die Darstellung auf Computer Bildschirmen ist es schwer eine exakte Aussage zu treffen, da einige bestehende Forschungen nur vage Resultate haben. Rello et al. haben den Einfluss von Textgrößen im Rahmen von 10 bis 26 pt untersucht, sowie Zeilenabstände von 0.8 bis 1.8. Ihre Resultate sind, dass extreme Zeilenabstände die Lesbarkeit verringern und dass größere Schriftarten besser sind [RPM16]. Eine ähnliche Studie von Banerjee et al. hat Textgrößen von 10, 12 und 14 pt in Verbindung zu je zwei serifenlosen und Serifenschriften untersucht. Sie kamen ebenfalls zu dem Ergebnis, dass die größte gewählte Schriftart die beste Lesbarkeit erreicht [BMM11, S. 140]. Cerepinko et al. haben Textgrößen von 10 bis 16 pt in Hinblick auf die Lesbarkeit auf iPads, Papier und PC Bildschirmen untersucht. Sie kamen ebenfalls zu dem Resultat, dass eine höhere Schriftgröße die Lesbarkeit erhöht [ČKP17, S. 1200]. Außerdem fanden sie, dass Texte auf Papier einfacher lesbar sind, als auf Bildschirmen.

Die Godot Engine unterstützt die Möglichkeit, eine beliebige Schriftart zur Darstellung von Texten zu verwenden. Außerdem kann eine projektspezifische Standardchrift eingestellt werden. Hierdurch können bereits viele grundlegende Maßnahmen getroffen werden, um Texte lesbarer zu gestalten. Durch die Verwendung einer zentralen Standardchriftart im Projekt ist es ebenfalls möglich, die Anforderung nach der Skalierbarkeit der Texte umsetzen (**WCAG 1.4.4**). Hierfür muss lediglich die Einstellung der Schriftgröße geändert werden.

### 3.3.4 Anpassbare Textgröße

In Kapitel 3.3.3 hat sich gezeigt, dass die Auswahl einer angemessenen Schriftgröße nicht trivial ist. Deswegen wurde die Anforderung einer von Nutzer\*innen anpassbaren Textgröße aus den „Web Content Accessibility Guidelines“ ebenfalls aufgenommen. Hierbei soll es nicht nur möglich sein, die Textgröße auf bis zu 200% zu erhöhen, sondern die UI Elemente sollen ebenfalls darauf ausgelegt sein, diese Änderung korrekt darzustellen (**REQ-037**). Hierfür wird angestrebt, die in der Godot Engine

eingebauten Konfigurierungsmöglichkeiten zu nutzen. Dabei sollen sich die Interface Elemente so erweitern können, dass sie nicht außerhalb des Sichtbereiches sind. Die Einstellungen, die hierfür relevant sind, sind die „Grow Direction“, „Size Flags“, mit denen, eingestellt wird, wie sich die Elemente verhalten sollen, sowie Anker im Layout. Des Weiteren sollen UI Container verwendet werden, damit die Anordnung der Elemente konstant bleibt.

### 3.3.5 Screenreader-Kompatibilität

Auf Websites wird es Nutzer\*innen mit eingeschränkter Sicht ermöglicht, visuelle Medien wahrzunehmen, indem alternative Textbeschreibungen hinterlegt werden. Diese können dann von Screenreadern verstanden und vorgelesen werden. Die Anforderung **REQ-038** verlangt, dass so eine Möglichkeit implementiert wird. Im Gegensatz zu Websites ist es jedoch potenziell schwerer, Darstellungen Screenreader kompatibel zu gestalten. Der Grund hierfür ist, dass Game Engines die Möglichkeit bieten müssen, parsbare UI Elemente zu verwenden. Die Godot Engine stellt jedoch dahingegen nur ein Fenster, mit einer Pixeldarstellung des Spiels zur Verfügung. Dadurch können hinterlegte Informationen, wie alternative Textbeschreibungen und die Texte von UI Elementen nicht von externen Tools ausgelesen werden. Als Lösung hierfür bieten sich vor allem drei verschiedene Ansätze an. Zum einen können Voice overs für alle UI Elemente eingebaut werden, die automatisch abgespielt werden. Da dieser Ansatz sehr aufwändig ist, kann stattdessen eine Sprachsynthese Engine verwendet werden, die automatisch Text in Sprache umwandelt. Dadurch können Änderungen und dynamische Texte unproblematisch integriert werden. Ein weiterer Ansatz ist es, ein externes Programm zu entwickeln, das Screenreader kompatibel ist und Informationen aus dem Spiel bekommt und gleichzeitig die Interaktionen der Spieler\*innen an das Spiel weitergibt. Die dritte Möglichkeit ist es, die Game Engine anzupassen beziehungsweise zu erweitern, damit UI nativ auslesbar ist. Diese drei Optionen sollen nachfolgend genauer betrachtet werden.

#### Voice-over beziehungsweise Sprachsynthese

Eine Möglichkeit, Texte und Menüs für Menschen mit eingeschränkter Sicht zugänglich zu gestalten ist, in dem Programm selber eine alternative Audioausgabe einzubauen. Hierfür können vorweg aufgenommene Tonspuren verwendet werden. Dies ist eine technisch einfache Lösung, die jedoch hohe Ressourcen benötigt, um für alle Texte und potenziell für verschiedene Sprachen Voice overs aufzunehmen. Alternativ kann Sprachsynthese genutzt werden, um dynamisch Voice overs zu generieren. Da die Skriptsprache der Godot Engine sehr spezifisch für diese entwickelt ist, ist es nicht möglich, mit ihr externe Bibliotheken einzubinden. Es ist jedoch möglich, mithilfe von Add ons Python Code einzubinden oder als Engine Plugin C++ Code zu nutzen. Somit

können Bibliotheken, wie pytttsx3 und gTTs beziehungsweise Festival genutzt werden. Diese können eine Sprachsynthese durchführen, die dann im Spiel genutzt werden kann.

### **externes Programm**

Experimentell hat sich gezeigt, dass die Ausgabe der Godot Engine, die in einer Konsole stattfindet, Screenreader kompatibel ist. Standardmäßig ist es aber nicht möglich, Eingaben durch die Konsole zu tätigen. Hier könnte eine Anpassung der Engine genutzt werden, um die Steuerung des Spiels per Konsole zu ermöglichen. Wie jedoch im nächsten Kapitel noch genauer beschrieben wird, ist dieser Ansatz sehr aufwendig und stellt eine Barriere für Entwickler\*innen dar, da sie eine spezielle Version der Engine nutzen müssen. Eine weitere Möglichkeit ist es, dass ein separates Programm erstellt wird, das per TCP Kommunikation mit der Godot Engine verbunden ist. Hierdurch müsste lediglich eine Schnittstelle zwischen der Engine und einem externen Tool entwickelt werden. Für die Kommunikation per TCP stellt Godot bereits Möglichkeiten zur Verfügung. Das separate Programm könnte dann entweder mit einem nativen UI entwickelt werden, das Screenreader Kompatibilität ermöglicht oder es könnte eine Konsolenanwendung entwickelt werden, die aus Aus- und Eingabe für das Spiel genutzt werden kann.

### **Anpassung der Engine**

Dieser Ansatz ist der nachhaltigste, da hiervon direkt weitere Entwickler\*innen profitieren und zukünftig die Screenreader Kompatibilität gewährleistet ist. Im optimalen Fall wird dieser Weg gewählt. Er ist jedoch mit einem sehr hohen Aufwand verbunden, da die Godot Engine für diese Änderung potenziell massive Anpassungen benötigt. Zum einen muss auf ein natives UI System umgestiegen werden, das sich zwischen unterschiedlichen Betriebssystemen unterscheidet. Zum anderen müssen die bestehenden Möglichkeiten der Gestaltung der UI Elemente in diesem System korrekt dargestellt werden. Ein Problem, was hierbei ebenfalls aufgeworfen wird, ist, dass sehr wahrscheinlich der Wechsel auf so ein neues UI System inkompatibel mit dem bestehenden System ist. Dadurch würde es potenziell ein Konvertierungstool benötigen, um zu dem neuen Ansatz zu wechseln. Alternativ müssen bestehende Implementierungen komplett neu umgesetzt werden, um die neuen Accessibility Möglichkeiten nutzen zu können. Außerdem müssen solche Anpassungen bei neuen Versionen der Engine aktualisiert werden.

### 3.3.6 Untertitel

Untertitel ermöglichen es, viele Barrieren zu überbrücken (**REQ-041**). Zum einen können Gehöreinschränkungen entgegengewirkt werden, indem relevante Informationen nicht nur per Audio, sondern auch über Texte dargestellt werden. Außerdem können Sprachgrenzen überwunden werden. So ermöglichen Untertitel ein Medium zu verstehen, wenn es eine Fremdsprache verwendet, wenn die durch sie eine Übersetzung zur Verfügung stellen.

In einem Spiel wird maximal ein Untertitelsystem benötigt. Dieses muss dann von unterschiedlichen Stellen im Programm angesprochen werden können und die einkommenden Nachrichten priorisieren. Dadurch muss es möglich sein, dass relevante Informationen und Dialoge sofort dargestellt werden und weniger relevante Dinge, wie zum Beispiel Gespräche, die im Hintergrund passieren, wenn nötig ignoriert werden. Damit ein einzigartiges und leicht erreichbares Untertitelsystem implementiert werden kann, bietet sich ein Singleton an. Alternativ könnte eine statische Klasse genutzt werden, die angesprochen wird. Außerdem könnte es sich hier anbieten, einen Service Locator zu verwenden. Damit die Untertitel zeitlich entkoppelt werden und priorisiert werden können, bietet sich eine Event Queue an.

#### Singleton Pattern

Ein Singleton entspricht definitionsgemäß den Anforderungen eines Untertitelsystems. Es ist nur ein einziges Mal vorhanden und einfach ansprechbar. Die Verwendung eines Singletons bringt aber auch ein paar Konsequenzen mit sich, die beachtet werden müssen. Diese werden häufig kritisiert. Einige der Konsequenzen bzw. Kritikpunkte sind, dass Singletons einen globalen Aspekt in das Programm einbringen, der häufig unnötig ist [Bad17, S. 35; Fre15, S. 129]. Dies kann wiederum zu weiteren Problemen führen. Zum Beispiel kann der gleichzeitige Zugriff auf das Singleton zu unerwünschtem Verhalten führen [Fre15, S. 129; SL08, S. 174f]. So kann es passieren, dass eine Variable der Singleton Instanz gleichzeitig beschrieben und ausgelesen wird. Dann ist das Verhalten des Programms stark abhängig davon, welche Anweisung zuerst ausgeführt wird. Dadurch kann es passieren, dass das Programm nicht mehr deterministisch ist und Laufzeitfehler entstehen, die schwer zu finden sind. Außerdem wird durch die Nutzung von Singletons eine Abhängigkeit zwischen Klassen eingeführt, die lieber vermieden wird. Des Weiteren widersprechen Singletons dem Single Responsibility Prinzip, da die Singleton Klasse neben der eigentlichen Funktionalität auch dafür sorgen muss, dass es nur eine Instanz gibt. Ein weiteres Problem, das Singletons mit sich bringen ist, dass sie Tests erschweren. Vor allem Unit Tests, da diese eigentlich nur eine Klasse testen sollten und hier die Singleton Instanz weitere Komplexität mit sich bringt, die mit der eigentlichen Testklasse wenig zu tun

hat. Außerdem sind Klassen, die die mit der Singleton Instanz interagieren an diese gekoppelt, wodurch es teilweise sogar unmöglich wird sie zu testen, ohne gleichzeitig auch das Singleton zu testen. Es muss außerdem bedacht werden, dass beim Auslesen des Singletons für gewöhnlich sehr viel mehr Informationen abgerufen wird, als nötig ist und dass beim simultanen Ändern und Auslesen der Daten ungewünschtes Verhalten auftreten kann.

### **Statische Klasse**

Eine Alternative zu Singletons sind statische Klassen. Sie sind definitionsgemäß eindeutig referenzierbar und einzigartig. Da die Ähnlichkeit sehr hoch ist, finden sich viele Gefahren von Singletons auch bei statischen Klassen. So zum Beispiel, die enge Bindung zu verschiedenen System im Programm. Außerdem erschweren sie ebenfalls das Debuggen des Programms, weil sie während der Laufzeit von vielen unterschiedlichen Stellen im Code aufgerufen werden können. Weil eine statische Klasse kein Teil der Objekte zur Laufzeit ist, kommen hier ein paar weitere Hürden hinzu. Es ist in Spieleengines, wie Godot, ein bisschen komplexer, Referenzen zwischen einer statischen Klasse und Objekten zu erstellen, da Objekte als Teile von Szenen existieren und gegenseitig anhand ihrer Position in der Szene gefunden werden können. Statische Klassen dahingegen sind nicht in so einer Form an Szenen gebunden. Dadurch ist so eine Referenzierung nicht möglich. Sehr viel einfacher ist es, wenn so eine Klasse dafür verwendet wird, nur Daten zu verarbeiten oder Referenzen, die per Dependency Injection eingeführt werden, festzuhalten. In Singletons findet eine statische Klasse als Referenz zu der Singleton Instanz Verwendung.

### **Service Locator**

Mithilfe eines Service Locators (siehe 2.3.2) kann die spezifische Implementation vom Aufruf durch andere Systeme getrennt werden. Außerdem ist es möglich, die Art der Implementation einfach auszutauschen beziehungsweise zu deaktivieren. Im Vergleich zu einem Singleton bringt ein Service Locator eine höhere Anpassbarkeit, aber auch gleichzeitig eine höhere Komplexität mit sich. Da die Service Implementation zu Laufzeit ausgetauscht werden kann, bieten sich neue Möglichkeiten, wie zum Beispiel die Verwendung unterschiedlicher Versionen, die für unterschiedliche Anwendungszwecke optimiert sind.

### **Event-Queue**

Eine Event Queue hilft, damit einkommende Aufrufe von einem System zeitlich entkoppelt und priorisiert werden können. Für ein Untertitelsystem, das neben den primären Gesprächen auch Hintergrundgespräche und Closed Captions darstellen



soll, ist es nötig eine Priorisierung einzubauen. Bei einer Event Queue werden Nachrichten von anderen Systemen an das Untertitelsystem nicht sofort verarbeitet, sondern zuerst in einer Datenstruktur hinterlegt. Sobald das Untertitelsystem bereit ist, wird ein neuer Eintrag aus der Datenstruktur geholt und verarbeitet. Dieser Prozess wiederholt sich, bis keine unverarbeiteten Nachrichten mehr vorhanden sind. Dann wartet das System darauf, dass neue Nachrichten eintreffen.

Damit die Implementation nicht unnötig kompliziert wird, wird zuerst eine möglichst simple Datenstruktur verwendet, um die Event Queue zu repräsentieren. Hierbei werden vorerst keine Priorisierungen beachtet und die einkommenden Nachrichten werden nach dem Prinzip First In First Out (FIFO) verarbeitet. Das heißt, dass immer die Nachrichten in chronologischer Reihenfolge verarbeitet werden und neue Nachrichten warten müssen, bis alle vorher eingegangenen Nachrichten dargestellt wurden. Wie der Name vermuten lässt, empfiehlt sich hier die Verwendung einer Queue, da diese für das FIFO Prinzip optimiert ist. In Godot ist jedoch die Art der Datenstrukturen, die nutzbar sind, stark eingeschränkt. Es gibt Dictionaries [Conb] und Arrays [Conc]. Für die Arrays gibt es einige spezialisierte Implementationen, die auf bestimmte Datentypen ausgelegt sind. Weitere Datenstrukturen können extern eingebaut werden, dafür ist es aber nötig separat mit C oder C++ Bibliotheken zu erstellen, die daraufhin in der Engine dazugeladen werden. Hiermit könnten Performanceoptimierungen getätigt werden, die mit der Engine eigenen Programmiersprache GDScript nicht möglich sind. Im Rahmen dieser Arbeit erscheint so eine Optimierung jedoch unnötig zu sein, da der Nutzen den Aufwand nicht rechtfertigt.

### **Experimentelle Untersuchung der Performance**

Um zu untersuchen, welche native Datenstruktur die besten Ergebnisse liefert, wurde im Rahmen dieser Arbeit ein experimenteller Test durchgeführt, der nachfolgend genauer beschrieben wird. Bei dem Test wurde die Zeit gemessen wurde, die verschiedene Implementationen einer Event Queue benötigen, um eine hohe Anzahl an Verarbeitungen zu verarbeiten. Eine Verarbeitung entspricht dabei dem Zugriff auf das erste Element, dem Löschen des ersten Elements und dem Hinzufügen eines neuen Elements. Das Experiment besteht jeweils aus einer Million Verarbeitungen, damit ein übermäßig hohes Datenaufkommen simuliert wird. Realistisch ist eine Verarbeitung je drei bis 10 Sekunden. Die geprüften Implementationen werden nachfolgend beschrieben.

Der erste Ansatz ist ein sehr naiver. Hierbei wird ein Array verwendet, bei dem jeweils am Ende neue Elemente hinzugefügt werden und das erste Element nach einem Zugriff gelöscht wird. Hierbei muss beachtet werden, dass vor allem der Löschvorgang einen hohen Aufwand darstellt, da beim Entfernen des ersten Elements alle folgen

den Elemente um je einen Platz im Array verschoben werden müssen. Des Weiteren kann es sein, dass die Erweiterungen des Arrays um neue Elemente eine erhöhte Belastung für den Arbeitsspeicher darstellen, da der verwendete Speicherplatz potenziell jedes Mal neu zugeteilt werden muss. Wie bereits festgelegt, soll der Test nur einen Überblick über den Aufwand der einzelnen Implementationen bringen. Deswegen wird hier nur die Gesamtdauer einer Verarbeitung untersucht und nicht der Einfluss der einzelnen Operationen.

Der nächste Ansatz ist die Verwendung eines Arrays mit fester Länge, bei dem ein Index auf das jeweilige aktuelle Element verweist. Sobald der Index am Ende des Arrays ankommt, springt er wieder zurück zum Anfang. Der Array wird dementsprechend als Kreis angesehen. Dies bringt die Einschränkung mit sich, dass eine Erweiterung des Arrays zu einem späteren Zeitpunkt kompliziert ist. Eine Lösung hierfür könnte es sein, die Länge des Arrays so zu wählen, dass es unwahrscheinlich ist, dass eine Erweiterung nötig ist. Zudem kann eine Möglichkeit eingebaut werden, dass eine Verlängerung möglich ist. Zwei Ansätze hierfür wären zum einen, eine Erweiterung des Arrays um eine Anzahl Elemente und dann das Verschieben der Elemente ab dem aktuellen Element um dieselbe Anzahl Positionen. Zum Anderen könnte ein zweiter Array verwendet werden, der beim Überlaufen des ersten als beliebig erweiterbarer Puffer angesehen werden kann. Wie dieser genau funktioniert, ist dann eine weitere Frage, die neue Möglichkeiten und Hindernisse aufwirft. So kann der Array fortlaufend erweitert werden und über einen Index wird festgehalten, wo sich das aktuelle Element befindet. Sobald das Ende erreicht wird, kann dann wieder der ursprüngliche kreisförmige Array verwendet werden. Hierbei kann es aber theoretisch gesehen passieren, dass der neue Nachrichten zu schnell eintreffen und die Queue ins Unendliche wächst. Für diesen Fall und damit die Nachrichten in der Event Queue aktuell sind, wird eine Möglichkeit benötigt, bei der zu alte Nachrichten gelöscht werden und ein zu hohes Aufkommen von neuen Einträgen als Fehler markiert und blockiert wird.

Wenn ein zyklischer Array verwendet wird, müssen alte Einträge nicht gelöscht werden, sondern es reicht aus, sie zu überschreiben, sobald sie erreicht werden. Da zwei Indizes benötigt werden, einer um das aktuelle hinterste Element zu referenzieren und einen um das aktuelle Element zu referenzieren, ist es simpel zu erkennen, wann der Array überlaufen würde. Dadurch kann erkannt werden, wann neue Nachrichten alte Nachrichten, die noch nicht verarbeitet wurden, überschreiben würden. Hier muss lediglich geprüft werden, ob der Index des zuletzt hinzugefügten Elements einen Schritt vor dem Index des aktuellen Elements ist. Dies würde nämlich bedeuten, dass der nächste Eintrag, wenn er hinzugefügt wird, bevor das aktuelle Element verarbeitet wurde, dieses überschreibt. Hier ist es sinnvoll, eine Warnung auszuge

ben und potenziell einen weiteren Buffer Array zu initialisieren. Um zu verhindern, dass in zu kurzer Zeit zu viele neue Nachrichten gesendet werden, sollte zum Beispiel kontinuierlich geprüft werden, wie schnell Einträge verarbeitet werden und wie schnell sie hinzukommen. Alternativ ist es auch möglich zu warten, bis ein bestimmter Punkt erreicht ist und dann eine Fehlermeldung zu erzeugen. Hierbei kann bereits das Überlaufen des zyklischen Arrays als kritischer Punkt verwendet werden oder erst das Auffüllen des Buffer Arrays. Während der Entwicklung ist es sinnvoll, diesen Fehler nicht nur als Meldung anzuzeigen, sondern auch das Programm zu pausieren beziehungsweise zu beenden. Dieser Punkt sollte nämlich nie in dem fertigen Produkt erreicht werden, da ansonsten die Entscheidung getroffen werden muss, ob ab einem bestimmten Punkt Nachrichten gelöscht oder nicht mehr angenommen werden.

Ein weiter Ansatz besteht darin, ein Dictionary zu verwenden, bei dem fortlaufende Indizes als Schlüssel verwendet werden und die darzustellende Information als Wert besitzen. Im Gegensatz zu einem Array, bei dem alle Einträge hintereinander im Speicher hinterlegt werden, nutzt ein Dictionary die Schlüssel, um die Einträge zu finden. Dementsprechend gibt es auch keinen wirklich „ersten“ Eintrag. Stattdessen wird der Index des aktuellen Elements, der ebenfalls dem Schlüssel entspricht, mitgeführt.

Der letzte Ansatz, der hier untersucht wurde, ist die Verwendung eines Arrays, bei dem anstatt das erste Element zu löschen, alle Elemente in der Reihenfolge invertiert werden. Danach wird das letzte Element gelöscht, das ursprünglich das erste war. Abschließend wird die ursprüngliche Reihenfolge wieder hergestellt.

Nachfolgend werden die Ergebnisse des Experiments in Tabellen 3.2 und 3.3 dargestellt. Für Tabelle 3.2 wurden 100 Elemente bei jeder Datenstruktur als Größe gewählt und es wurden eine Million Verarbeitungen durchgeführt. Bei Tabelle 3.3 wurden dagegen 1000 Elemente als Größe jeder Datenstruktur gewählt. Die beiden Tabellen zeigen außerdem eine Untersuchung des Durchschnitts der ermittelten Zeiten. Des Weiteren zeigen sie, wie lange ein Verarbeitungsschritt dauert und wie viele Verarbeitungsschritte innerhalb von einer Millisekunde stattfinden können. Aus den Tabellen zeigt sich, dass ein Array mit kreisförmig voranschreitendem Index am schnellsten ist. Außerdem zeigt sich keine signifikante Veränderung der Laufzeit bei einer Erhöhung der Arraygröße. Des Weiteren hat sich gezeigt, dass die Laufzeit bei der naiven Nutzung eines Arrays und der Nutzung eines Dictionaries bei 100 Elementen fast gleich ist. Bei einer Erhöhung der Elemente auf 1000 benötigt das Dictionary minimal mehr Zeit, die Laufzeit des naiven Arrays steigt dahingegen fast linear mit der Anzahl der Elemente. Dasselbe lineare Verhalten zeigt sich ebenfalls bei dem Array, der invertiert wird. Dieser benötigt jedoch generell fast die dreifache Zeit des naiven Arrays. Es

kann somit festgehalten werden, dass ein Array mit kreisförmig laufendem Index für eine FIFO Datenstruktur effizient genutzt werden kann. Arrays, bei denen Elemente gelöscht werden oder die invertiert werden, sind nur für eine geringe Anzahl an Elementen geeignet. Dictionaries bringen eine gewisse erhöhte Laufzeit mit sich, sind dann aber nicht sehr stark abhängig von der Anzahl der Elemente.

**Tabelle 3.2:** Laufzeit und Auswertung für 100 Elemente pro Datenstruktur

Testlauf	Laufzeit(1) in ms	Laufzeit(2) in ms	Laufzeit(3) in ms	Laufzeit(4) in ms
1	1227	246	1042	3242
2	1225	249	1054	3242
3	1224	248	1063	3211
4	1228	245	1065	3222
5	1227	251	1061	3218
6	1208	250	1086	3258
7	1209	251	1082	3247
8	1221	250	1085	3246
9	1219	253	1082	3255
10	1233	253	1082	3252

	Impl. 1	Impl. 2	Impl. 3	Impl. 4
Durchschnitt	1222.1	249.6	1070.2	3239.3
Dauer einer Verarbeitung	0.00122	0.00025	0.00107	0.00324
Verarbeitungen pro 1 ms	818.26	4006.41	934.402	308.71

Implementation 1: naiver Array

Implementation 2: Array mit kreisförmigem Index

Implementation 3: Dictionary

Implementation 4: Array mit Invertierung

Alles in allem gibt es mehrere unterschiedliche Optionen, um ein Untertitelsystem umzusetzen. Diese bringen unterschiedliche Vor- und Nachteile mit sich. Es wird sich nun dafür entschieden, dem KISS Prinzip („Keep it simple, stupid“) zu folgen und die einfachste Möglichkeit zu implementieren und dafür Features, die nicht Teil der zugrunde liegenden Anforderungsanalyse sind, zu ignorieren. Deswegen ist die Entscheidung darauf gefallen, ein Singleton für das Untertitelsystem zu verwenden. Dieses stellt die einfachste Art der untersuchten Optionen dar, um eine Möglichkeit zu bieten, von verschiedenen Systemen aus Untertitel darzustellen. Ein Service Locator bringt zwar die Möglichkeit verschiedene Systeme zu implementieren und zwischen diesen zu wechseln, im Rahmen dieser Arbeit zeigt sich jedoch kein Anwendungszweck für diese Features. Die reine Verwendung einer statischen Klasse wurde ebenfalls abgelehnt, da das Untertitelsystem Referenzen auf UI Elemente benötigt. Diese sind sehr viel einfacher umsetzbar, wenn ein Singleton verwendet wird.

**Tabelle 3.3:** Laufzeit und Auswertung für 1000 Elemente pro Datenstruktur

Testlauf	Laufzeit(1) in ms	Laufzeit(2) in ms	Laufzeit(3) in ms	Laufzeit(4) in ms
1	9475	249	1158	28456
2	9467	244	1189	28346
3	9468	251	1191	28403
4	9486	252	1188	28381
5	9444	248	1186	28411
6	9457	243	1187	28345
7	9466	248	1177	28448
8	9454	249	1177	28453
9	9471	250	1190	28627
10	9469	250	1177	28670

	Impl. 1	Impl. 2	Impl. 3	Impl. 4
Durchschnittliche Laufzeit in ms	9465.7	248.4	1182	28454
Dauer einer Verarbeitung in ms	0.00946	0.00025	0.00118	0.02845
Verarbeitungen pro 1 ms	105.64	4025.76	846.02	35.14

Implementation 1: naiver Array  
 Implementation 2: Array mit kreisförmigem Index  
 Implementation 3: Dictionary  
 Implementation 4: Array mit Invertierung

Außerdem ist die Verwendung von statischer Programmierung in Godot stark eingeschränkt. Es gibt statische Methoden, aber weder rein statische Klassen, noch statische Variablen.

### Fazit zur Wahl des Implementationskonzepts

Im Sinne des KISS Prinzips (engl.: „Keep it simple, stupid“, sinngemäß: „In der Kürze liegt die Würzel“) wurde die Entscheidung getroffen, für die Implementation in dieser Arbeit keine Event Queue einzubauen, da diese zwar für komplexere Untertitel nötig sind, aber nicht für den aktuellen Anwendungsfall. Stattdessen soll ein Array implementiert werden, der mit einem fortlaufenden Index, der diesen kreisförmig durchläuft, eine FIFO Datenstruktur darstellt. Diese weist eine hohe Performance und Skalierbarkeit auf.

### Lesbarkeit

Neben der Implementation der Untertitel ist ebenfalls ihre Lesbarkeit wichtig (**REQ-044**). Diese Anforderung verlangt, dass die Schriftgröße ausreichend groß gewählt wird, dass die Untertitel einen transparenten oder opaken Hintergrund besitzen und dass nicht zu viel Text auf einmal dargestellt wird. Zwei dieser drei Kriterien können in dieser Arbeit erfüllt werden. Zum einen wird, wie in Kapitel 3.3.3 beschrieben, oh

nehin angestrebt, eine gut lesbare Schriftart und -größe zu verwenden. Des Weiteren kann ein Hintergrund mithilfe von einem Panel UI Element einfach implementiert werden. Komplexer ist es, die Textlänge zu beeinflussen. Dies ist ein Aspekt, der erst in der Anwendung zum Tragen kommt und von Nutzer\*innen der Implementation spezifiziert werden muss. Es wäre möglich, die Zeichenzahl auf eine bestimmte Maximalanzahl zu reduzieren. Hierfür gibt es laut Karakanta et al. [KNT20, S. 2] einige Richtlinien, die beachtet werden sollten. Zum einen sollte die Zeichenanzahl pro Zeile nur maximal 42 für lateinische, 14 für japanische und 12 bis 16 für chinesische Schriftzeichen betragen. Außerdem sollen die Untertitel weniger als 10% des Bildschirms einnehmen. Daraus ergibt sich, dass für gewöhnlich höchstens zwei Zeilen genutzt werden sollen. Außerdem soll die Anzeigedauer so gewählt sein, dass sie für eine komfortable Lesegeschwindigkeit angemessen sind. Empfohlen sind 21 Zeichen pro Sekunde. Außerdem sollte beachtet werden, dass semantische beziehungsweise syntaktische Einheiten nicht getrennt werden. Des Weiteren sollte bei der Verwendung von zwei Zeilen darauf geachtet werden, dass diese in etwa gleich lang sind. Diese Anforderungen könnten durch eine Überprüfung im Programmcode per Warnung oder Fehlermeldung erzwungen werden. Im Sinne der Nutzungsfreundlichkeit, wird sich hier jedoch dafür entschieden, dass es besser ist, die Entscheidung über die Formatierung der Untertitel den Nutzer\*innen zu überlassen.

### 3.3.7 Variable Lautstärke

Nutzer\*innen sollen die Möglichkeit haben, die Lautstärke von Effekten, Sprache und Musik separat einstellen zu können (**REQ-042**). In Godot gibt es hierfür die Option, mehrere Audiobusse zu verwenden. Diese können von Audioquellen als Ausgabe spezifiziert werden. Dadurch kann die Lautstärke von Effekten, Musik und Sprache getrennt voneinander eingestellt werden. Außerdem wurde sich dafür entschieden, einen Haupt Audiobus zu verwenden, der die Gesamtlautstärke anpasst. Für die Nutzer\*innen wurde ein Menü mit Slidern implementiert, in dem die Lautstärke der Audiobusse angepasst werden kann. Hierbei musste beachtet werden, dass die Slider einen linearen Anstieg besitzen [JG23]. Wahrgenommen Lautstärke verhält sich dahingegen jedoch logarithmisch. Deswegen wurde der Wertebereich der Slider auf knapp über null (0,001) bis zwei festgelegt. Dies entspricht nahezu Stille bis höchste Lautstärke, die die Tonqualität nicht beeinträchtigt.

In Godot wird eine relative Dezibelskala verwendet [JG23]. Jede sechs Dezibel verdoppelt beziehungsweise halbiert sich die Lautstärke. Null Dezibel entspricht der maximalen Lautstärke, die von digitalen Audio Systemen ohne Verzerrung wieder gegeben werden kann [JG23]. Um dieses Verhalten der verwendeten Dezibelskala korrekt zu repräsentieren, wird der lineare Wert der Slider mithilfe von Formel 3.1 verrechnet.

$$y_{Lautstaerke}dB = 20 * \log_{10}(x) \quad (3.1)$$

Des Weiteren muss die aktuelle Lautstärke in Dezibel beim Öffnen des Menüs zu einem linearen Wert für die Slider konvertiert werden. Dafür wird die Umkehrfunktion 3.2 verwendet.

$$y = 10^{(x_{Lautstaerke}dB)/20} \quad (3.2)$$

Da die Umrechnung in Formel 3.1 sich bei null einem Wert von  $-\infty$  nähert, wird der Audiobus deaktiviert, sobald der minimale Wert (0,001) im Slider erreicht wird. Dadurch wird in diesem Grenzfall gewährleistet, dass der minimale Wert des Sliders dazu führt, dass kein Ton mit dem jeweiligen Audiobus wiedergegeben wird.

### 3.3.8 Accessibility Informationen & Feedback

Nutzer\*innen sollten vor dem Start des Spiels die Möglichkeit haben, sich über die implementierten Accessibility Features zu informieren (**REQ-052**). Außerdem sollte es im Spiel einfach sein, diese Informationen ebenfalls einfach zu finden und potenzielle Optionen, die mit ihnen zusammenhängen, auffindig zu machen (**REQ-053**).

Da das Softwareprojekt, das im Rahmen dieser Arbeit entwickelt wird, nur auf GitHub veröffentlicht wird, ist die Möglichkeit der vorherigen Information nur eingeschränkt umsetzbar. Der Grund hierfür ist, dass es auf GitHub keine expliziten Möglichkeiten gibt, Accessibility Features auszuweisen. Projekte, die auf dieser Seite veröffentlicht werden, werden primär mithilfe einer Datei, die sich im Hauptordner des Projektes befindet, beschrieben. Hierbei erfolgt die Beschreibung für gewöhnlich primär durch Text. Weil der komplette Quellcode öffentlich ist, ist es technisch gesehen Nutzer\*innen möglich, im Code alle relevanten Informationen zu finden und zu überprüfen. Für die meisten Anwender\*innen ist dies jedoch keine sinnvolle Lösung. Statt dessen soll die Funktion von „Readme“ Dateien auf GitHub genutzt werden, um direkt Informationen zu dem spezifischen Projekt zu zeigen. Deswegen wurde sich dafür entschieden, eine Auflistung aller implementierter Accessibility Features dort aufzuführen, damit diese leicht zu finden sind. Außerdem wurde die E Mail Adresse des Autors hinterlegt, damit Feedback möglichst einfach gegeben werden kann. Dadurch sollen Nutzer\*innen die Möglichkeit haben, den Autor dieser Arbeit auf Barrieren hinzuweisen, die sich ihnen stellen und Verbesserungswünsche zu spezifischen Features zu formulieren.

Für Wünsche nach weiteren Funktionalitäten oder Meldungen von Fehlern bietet GitHub außerdem eine Funktion namens „Issues“. Hier können andere Nutzer\*innen der Seite spezifische Probleme und Wünsche ansprechen. Da meist für jedes Thema ein neues Issue verfasst wird, in dem wiederum Kommentare geschrieben werden können, wird Entwickler\*innen hier eine strukturierte Feedbackliste gegeben. Während der ersten externen Tests des Projektes wurde diese Funktion bereits verwendet.

Damit Spieler\*innen ebenfalls im Spiel einfach die Informationen über die implementierten Accessibility Features bekommen können, sollen diese in einem Menü, das direkt vom Hauptmenü aus zugänglich ist, dargestellt werden. Dort soll sich ebenfalls die E Mail Adresse des Autors befinden, um Feedback geben zu können (**REQ-055**).

Für ein spezifisches entwickeltes Spiel wäre es besser, andere Möglichkeiten zu bieten, um Feedback geben zu können. Der Grund hierfür ist, dass dieses meist helfen kann, um das Produkt zu verbessern. Deswegen sollte die Barriere der Kommunikationsaufnahme möglichst gering gehalten werden. Hierbei kann es sich lohnen, ein Feedbackformular zur Verfügung zu stellen. Da die Nutzer\*innen des Projekts, das im Rahmen dieser Arbeit entwickelt werden soll, primär Entwickler\*innen sind, wurde sich dafür entschieden, dass GitHub Issues und die Möglichkeit der Kontaktaufnahme per E Mail ausreichend sind.

### 3.3.9 Persistente Einstellungen

Welche Accessibility Features wie verwendet werden, kommt auf die jeweiligen Spieler\*innen an. Deswegen ist es nötig, dass diese Features sehr spezifisch konfigurierbar sind. Hierfür bietet sich die Nutzung eines Optionsmenüs an. Dieses eröffnet aber ein Problem für die Systemarchitektur. Es ist simpel, bestimmte Werte mithilfe eines User Interfaces festzulegen. Da es aber möglich sein sollte, von unterschiedlichen Stellen im Spiel aus dieselben Einstellungen zu treffen, müssen diese persistent festgehalten werden (**REQ-054**). Zwei Orte, an denen ein Optionsmenü mindestens aufrufbar sein sollte, ist im Hauptmenü, damit grundlegende Einstellungen bereits getroffen werden können und im Spiel selber, damit die Accessibility Features detailliert auf die persönlichen Ansprüche angepasst werden können. Außerdem sollten die Konfigurationen zwischen Neustarts des Programms persistent sein. Andernfalls wird unnötiger Mehraufwand erzeugt. In diesem Fall müssten Menschen mit Behinderungen mehr Arbeit leisten, um mit dem Spiel interagieren zu können, als Menschen ohne Einschränkungen. Dadurch würde sich hiermit eine Barriere ergeben, die von vornherein vermieden werden kann, indem die Konfigurationen zwischen Spielsessions festgehalten werden. Ein weiterer wichtiger Punkt, der bei der Implementation von Konfigurationsmöglichkeiten beachtet werden muss, ist, dass diese



von unterschiedlichen Skripten im Programm abgefragt werden müssen. Dadurch muss ermöglicht werden, dass die Einstellungen einfach durch andere Skripte referenziert werden. Zwei Ansätze, zur Umsetzung solcher Konfigurationsmöglichkeiten, werden nachfolgend genauer untersucht. Außerdem stellen Singletons, die in Kapitel 3.3.6 untersucht wurden, eine weitere Möglichkeit dar, die Optionen festzuhalten und abzufragen.

### **Speichern und Laden aus einer Datei**

Eine Möglichkeit, die Optionen zu speichern und an verschiedenen Stellen im Programm auszulesen ist, diese als Datei abzuspeichern und zu laden. Hierfür können alle Optionen in einem einfach auslesbaren Format, wie zum Beispiel JSON, gespeichert werden. Die Skripte, die diese Optionen auslesen wollen, benötigen dann nur eine Möglichkeit, die Datei auszulesen. Hierfür könnte eine Helferfunktion implementiert werden, die Redundanz verhindert. Die vorgenommenen Optionen in einer Datei abzuspeichern und wieder zu laden ist vor allem sinnvoll, um diese zwischen Neustarts des Programms persistent zu gestalten. Hierbei reicht es aber aus, wenn diese Datei nach Änderungen der Optionen überschrieben wird und beim Start des Programms wieder ausgelesen wird. Dieser Ansatz ist jedoch nicht sinnvoll, damit die Einstellungen an vielen Stellen im Programm erreicht werden können. Denn Änderungen der Datei müssen erkannt werden. Eine ineffiziente Möglichkeit ist es, die Datei in einem festen Rhythmus neu auszulesen. Alternativ wäre ein weiteres System nötig, das Änderungen meldet. Eine Möglichkeit für so ein System wäre das Observer Pattern, das im nachfolgenden Unterkapitel 2.3.1 untersucht wird. Für diesen Ansatz lässt sich festhalten, dass es sinnvoll ist, die Einstellungen zwischen Sessions des Programms zu speichern. Während der Laufzeit sind aber andere Systeme nötig, die eine höhere Performance gewährleisten.

### **Observer Pattern**

Änderungen der Optionen können mithilfe eines Observer Patterns mitgeteilt werden. Dabei fungiert das Optionssystem als Subjekt und Systeme, die die Optionen nutzen, sind Observer. Um Accessibility Optionen zu speichern und mitzuteilen, muss das Subjekt eindeutig identifizierbar sein, da es unsinnig und fehleranfällig ist, wenn es mehrere Objekte mit potenziell unterschiedlichen Optionswerten gibt. Damit Observer nicht unterschiedliche Werte für dieselbe Option auslesen, muss also gewährleistet werden, dass es nur exakt ein Subjekt gibt. Somit ist das Subjekt aber eigentlich ein Singleton. Dadurch stellt sich die Frage, ob es überhaupt nötig ist, ein Observer Pattern zu benutzen. Die Optionswerte sind ohnehin im Subjekt (bzw. Singleton) hinterlegt und können einfach ausgelesen werden. Andererseits würde eine Kombination von Singleton und Observer Pattern dazu führen, dass die Systeme, die

die Optionen auslesen, lose gekoppelt sind. Dadurch wäre die Implementation von Tests vereinfacht und die Observer können getestet werden, ohne das Singleton zu benötigen.

### **Fazit**

Vor dem Hintergrund, der vorgestellten Lösungsansätze wird sich im Rahmen dieser Arbeit für eine Kombination der drei Ansätze entschieden. Die Optionen werden bei Änderungen in einer Datei gespeichert und bei einem Neustart wieder ausgelesen. In der Laufzeit wird ein Singleton verwendet, das sich die Optionswerte merkt. Dieses funktioniert gleichzeitig als Subjekt, das allen Systemen, die die Informationen benötigen (aka. Observer) Änderungen mitteilt. Somit kann gewährleistet werden, dass die Optionen nur einmalig vorhanden sind und einfach ansprechbar sind. Gleichzeitig sind die Systeme, die diese Informationen nutzen und das Optionssystem nur lose miteinander gekoppelt.

## 4 Implementation

In diesem Kapitel werden die Implementationen beschrieben, die auf der Grundlage der Konzepte in Kapitel 3.3 entwickelt wurden. Dabei wird die Strukturierung von Code und Datenstrukturen, sowie Eigenheiten, die sich während der Entwicklung offenbart haben, betrachtet.

### 4.1 Konfigurierbare Tastenbelegung und Sensitivität

Die Anforderung **REQ-011** verlangt, dass die Tastenbelegung von Nutzer\*innen nach Belieben angepasst werden kann. Hierfür wurde ein Menü erstellt, in dem mithilfe von Labels und Buttons die aktuelle Tastenbelegung beschrieben wird (siehe Grafik 4.1). Dabei gibt das Label eine Auskunft über die Aktion, die mit der Eingabe verknüpft ist (zum Beispiel nach links, rechts, vorne oder hinten gehen). Die Buttons zeigen, welche Taste aktuell für die jeweilige Aktion genutzt wird. Beim Drücken des Buttons wird ein Skript aufgerufen, das auf die nächste Tasteneingabe wartet und die Tastenbelegung einer spezifischen Aktion mit der neuen Eingabe überschreibt. Damit festgelegt wird, welche Interaktion angepasst wird, besitzt das Skript eine Variable, die im Inspektor angezeigt wird, über die der Name der Aktion bestimmt wird. Für das Festlegen der Tastenbelegung bietet die Godot Engine bereits ein ausführliches System, das beliebige Eingabeevents mit bestimmten Tasten verbindet. Zur Identifikation des Events wird ein String verwendet, der den Namen dieser darstellt. Ein Beispiel, wie eine Eingabe namens „move\_forward“ in Godot definiert ist, ist im Quelltext 4.1 dargestellt. Die Verwendung dieser Eingabe ist im Quelltext 4.2 demonstriert.

**Quelltext 4.1:** Tastenbelegung für die „move forward“ Eingabe

```
1 move forwards {
2     "deadzone": 0.1,
3     "events": [
4         Object(
5             InputEventKey,
6             "resource local to scene": false,
7             "resource name": "",
8             "device": 0,
9             "window id": 0,
10            "alt pressed": false,
11            "shift pressed": false,
12            "ctrl pressed": false,
13            "meta pressed": false,
14            "pressed": false,
15            "keycode": 87,
16            "physical keycode": 0,
17            "unicode": 0,
18            "echo": false,
```

```

19         "script": null
20     )
21 ]
22 }

```

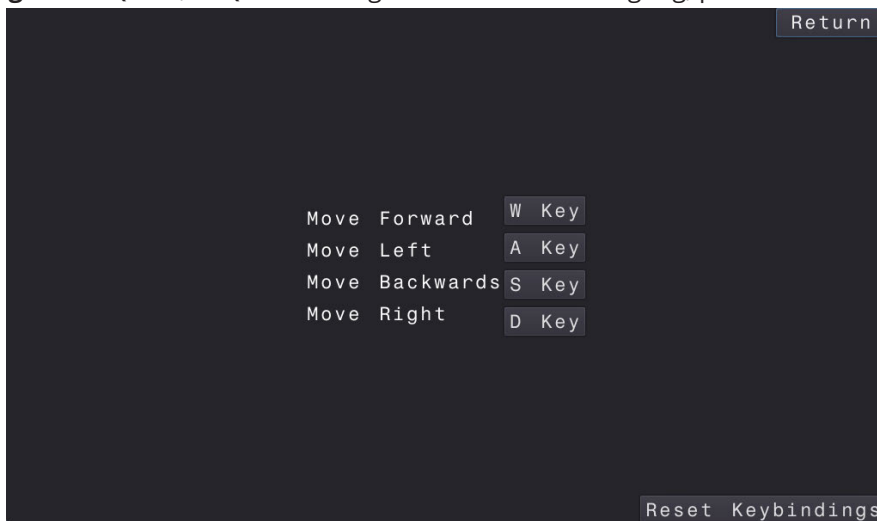
**Quelltext 4.2:** Verwendung der „move forward“ Eingabe

```

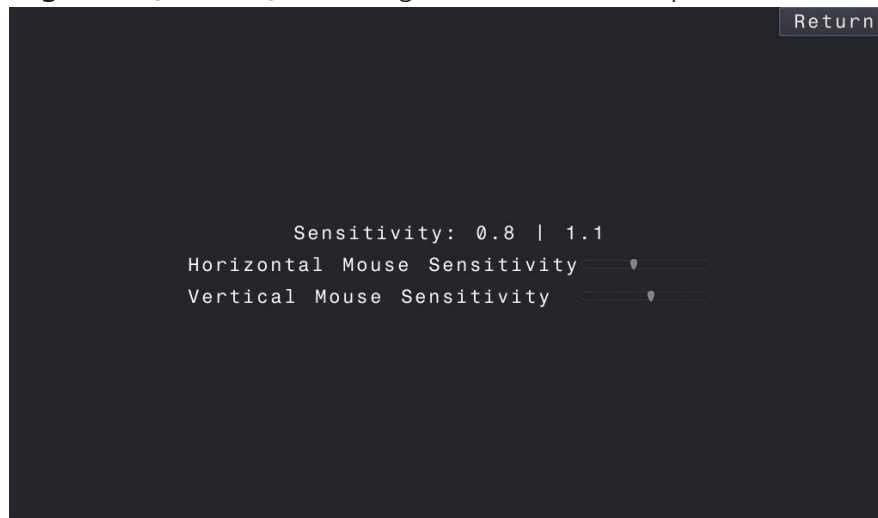
1 if Input.is_action_pressed("move forward"):
2     move_forward()

```

**Abbildung 4.1:** REQ-011, REQ-054: Konfigurierbare Tastenbelegung, persistente Einstellungen



Damit die Tastenbelegungen zwischen Sessions persistent sind, wurde ein Singleton erstellt, das zwei Dictionaries beinhaltet. Eins beinhaltet die Standardbelegung, die in der Engine eingestellt wird und die andere beinhaltet die angepasste Belegung. Bei Änderungen wird das Dictionary mit den angepassten Events in einer Datei gespeichert. Diese wird ebenfalls beim Start des Spiels geladen. Außerdem wird eine Methode bereitgestellt, mit der die Standardtastenbelegung wiederhergestellt werden kann. In der Godot Engine gibt es ein Singleton namens InputMap. Dieses handhabt die Tastenbelegungen. Mit den Funktionen `action_erase_events(action_name)` und `action_add_event(action_name, event)`. Mit diesen werden erst alle bestehenden Eingabeereignisse für eine Aktion gelöscht und dann ein spezifisches Event hinzugefügt. Dies stellt eine einfache Möglichkeit dar, die Anforderung zu erfüllen. Es bringt aber das Problem mit sich, dass nur eine Belegung gleichzeitig möglich ist. Alternativ könnte versucht werden, die Events per Index zu überschreiben. Hierfür wäre jedoch ein komplexeres Menü, mit je einem Button pro Belegung, pro Aktion nötig. Da so etwas zwar potenziell sinnvoll, aber nicht unbedingt notwendig ist, wurde sich entschieden dem KISS Prinzip zu folgen und eine einfache anstelle einer komplexeren Lösung zu bevorzugen.

**Abbildung 4.2:** REQ-013, REQ-054: konfigurierbare Sensitivität, persistente Einstellungen

Eine weitere Anforderung ist, dass Nutzer\*innen die Möglichkeit haben, die Sensitivität der Eingabemöglichkeiten zu definieren (**REQ-013**). Hierfür wird zuallererst eine Möglichkeit benötigt, um eine Variable zu speichern, die die Sensitivität speichert. Hierfür bietet die Godot Engine die Möglichkeit, eine neue Ressource zu definieren. Dafür muss lediglich ein Skript von der Klasse „Ressource“ erben. Bei dieser wurden für diesen Anwendungsfall zwei öffentliche Variablen erstellt, die die horizontale und die vertikale Mausempfindlichkeit darstellen. Außerdem wurde ein Menü erstellt, in dem mithilfe von Slidern der Wert der Variablen angepasst werden kann (siehe Grafik 4.2). Im Quelltext 4.3 ist eine beispielhafte Verwendung der Mausempfindlichkeit dargestellt. Die Variable „MOUSE\_SENSITIVITY“ muss in diesem Fall von der Ressource ausgelesen werden.

**Quelltext 4.3:** Verwendung einer Sensitivitätseinstellung namens: „MOUSE\_SENSITIVITY“

```

1 func input(event):
2     if event is InputEventMouseMotion:
3         self.rotate_y(deg2rad(event.relative.x * MOUSE_SENSITIVITY))

```

## 4.2 Userinterfacenavigation

Damit Nutzer\*innen das Userinterface per Tastatur und Controller steuern können (**REQ-012**), muss wenig implementiert werden. Das Userinterface ist in Godot bereits automatisch so verbunden, dass eine Navigation standardmäßig möglich ist. Hierfür muss nur das Element bestimmt werden, das als Eingangspunkt für die Interaktion verstanden wird. Der Code hierfür ist im Quelltext 4.4 dargestellt.

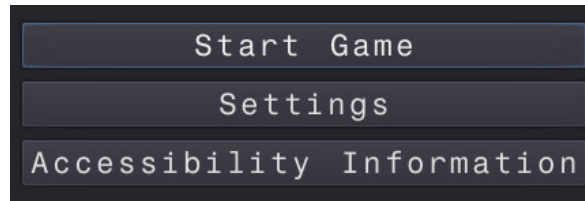
**Quelltext 4.4:** Festlegung des Eingangspunkt für die Navigation des Userinterfaces

```

1 ui_element.grab_focus()

```

**Abbildung 4.3:** REQ-012, REQ-022, REQ-034: Userinterfacenavigation, leicht lesbare Schrift, leicht lesbare Schriftgröße



In dieser Arbeit wurden zwei Skripte implementiert, die diese Funktionalität einbinden. In dem ersten wird während der Initialisierung der Fokus auf das Element gesetzt, auf dem das Skript liegt. Bei dem anderen muss ein Interfaceelement im Inspektor festgelegt werden, das daraufhin fokussiert wird. Dadurch kann das Navigationssystem, das die Godot Engine besitzt, genutzt werden. In Grafik 4.3 ist die Anwendung zu sehen. In diesem Fall ist das fokussierte Element anhand einer blauen Umrandung zu erkennen.

### 4.3 Lesbarkeit von Texten

Die Anforderungen verlangen, dass Texte leicht lesbar sind (**REQ-022, REQ-034**) und die Textgröße anpassbar ist. Die in Kapitel 3.3.3 untersuchten Forschungen haben keine eindeutige beste Schriftgröße für die Darstellung von Texten auf Computer Bildschirmen ergeben. Deswegen wird im Rahmen dieser Arbeit eine Schriftgröße von mindestens 21 Pixel (16 pt) für PC Spiele und von mindestens 28 Pixel für Konsolenspiele empfohlen. Damit die Schriftart einheitlich und groß genug ist, empfiehlt es sich ein „Theme“ zu nutzen oder bei dem Projekt eine Standardschrift einzustellen. Bei einem Theme können viele verschiedene Einstellungen zu allen UI Elementen, wie zum Beispiel Abstände und Farben, aber auch Textart und Textgröße festgelegt werden. Es kann ebenfalls ein Standard Theme eingestellt werden, aber je nachdem, wie das Projekt strukturiert ist, muss dies nicht notwendig sein. Der Grund hierfür liegt daran, dass die Einstellungen auf untergeordnete UI Elemente vererbt wird. In diesem Projekt, wurde eine Hauptmenüszenen erstellt, bei der ein bestimmtes Theme eingestellt wurde. Weitere Menüs werden als Kinder im Hinblick auf die Objekt hierarchie initialisiert. Dadurch werden die Einstellungen automatisch auch auf sie angewandt. Dieser Ansatz ist simpel und von der Engine erwünscht, es gibt aber auch eine große Einschränkung, die vor allem Texte betrifft. Wenn die Textgröße bei einem einzelnen UI Element vom Standard abweichen soll, muss hier nämlich das Theme überschrieben werden. Wenn zum Beispiel eine Überschrift größer sein soll, dann muss ein neues Schriftart Objekt erstellt werden. Dieses muss nicht als Datei gespeichert werden. Es kann aber empfehlenswert sein, dies trotzdem zu tun. Dadurch kann bei ähnlichen Anwendungsfällen dieselbe Datei genutzt werden und Änderungen der Schriftgröße können einfacher übernommen werden. Dies liegt dar

an, dass nicht die spezifischen Stellen im Projekt gefunden werden müssen, an denen die Schriftgröße angepasst wurde, sondern nur die weitere Datei bearbeitet werden muss. Die Resultate der gewählten Schriftgröße sind in Grafik 4.3 zu sehen.

#### 4.4 Anpassbare Textgröße

Nutzer\*innen sollen die Möglichkeit besitzen, die Textgröße selbstständig im Bereich von 100% bis 200% einzustellen (**REQ-037**). Hierfür wurde eine Option eingebaut, mit der die Textgröße per Slider bestimmt werden kann. Da eine Standardschriftart verwendet wird, reicht es aus, den Pfad zu dieser anzugeben und sie zur Laufzeit zu laden. Hierfür wurde eine Stringvariable im Inspektor exportiert, die den Pfad beschreibt. Bei dem Start des Skripts, wird überprüft, ob unter dem angegebenen Pfad eine Datei existiert. Falls dies nicht der Fall ist, wird eine Fehlermeldung ausgegeben und das Programm stoppt. Falls eine Datei unter dem angegebenen Pfad existiert, wird diese als „DynamicFont“ geladen und die aktuelle Größe der Schriftart wird lokal als Standardgröße abgespeichert. Danach wird das „value\_changed“ Event mit einer Methode verbunden. In dieser wird der neue Wert des Sliders genommen und die Größe der geladenen Schriftart wird verändert. Dafür wird die gespeicherte Standardgröße mit dem Wert des Sliders multipliziert. Damit hierbei eine Skalierung im Bereich von 100% bis 200% vorgenommen wird, wurde die Konfigurierung des UI Elements so vorgenommen, dass nur ein Wertebereich von eins bis zwei angegeben werden kann und eine geringe Schrittgröße von 0,01 genutzt wird.

Während der Implementation hat sich gezeigt, dass dieser Ansatz die gewünschten Erfolge erzielt. Gleichzeitig hat sich aber auch herausgestellt, dass es einen Fehler in der Engine gibt, durch den Schriften nicht mehr richtig dargestellt werden, wenn die verwendete „DynamicFont“ Datei bearbeitet wird. Das Resultat davon ist in Grafik 4.4 zu sehen. Im Rahmen der Arbeit wurde keine Möglichkeit gefunden, diesen Fehler zu beheben.

**Abbildung 4.4:** Darstellungsfehler von Texten



## 4.5 Screenreader-Kompatibilität

Screenreader werden von Menschen mit eingeschränkter Sicht genutzt, um Texte und alternative Beschreibungen von visuellen Medien besser wahrnehmen zu können. Hierbei wird der Text beziehungsweise die Beschreibung mit einer Text to Speech Engine auditiv wiedergegeben. Die Anforderung **REQ-038** verlangt, dass die Möglichkeit der Nutzung eines Screenreaders ermöglicht wird. Wie bereits im Konzept untersucht wurde, gibt es vor allem drei verschiedene Ansätze, um die Anforderung zu erfüllen. Eine direkte Nutzung der Godot Engine ohne weitere Arbeit ist nicht möglich, da Screenreader nicht unterstützt werden. Stattdessen könnte ein externes Programm genutzt werden, das mit dem eigentlichen Spiel kommuniziert und eine Schnittstelle zu Spieler\*innen und Screenreadern darstellt. Alternativ kann ein Voice Over beziehungsweise Sprachsynthese genutzt werden, die im Programm integriert ist. Hierdurch wird das implizite Accessibility Ziel der Anforderung erreicht, auch wenn der Wortlaut der Anforderung nicht erfüllt wird. Außerdem gibt es die Möglichkeit, die Engine anzupassen und so einen nativen Ansatz zu bieten, dass Spiele, die mit ihr erstellt werden, Screenreader kompatibel sind. Dieser Weg ist jedoch zu komplex für den Rahmen dieser Arbeit. Stattdessen wurden die beiden anderen Ansätze verfolgt. In der Annahme, dass Sprachsynthese die einfachere Möglichkeit ist, wurde diese primär verfolgt. Der Grund für diese Vermutung liegt darin, dass es sich bei den Ansätzen zur Sprachsynthese lediglich um die Nutzung von existierenden Bibliotheken handelt. Dadurch wird erwartet, dass eine möglichst simple Lösung erreicht wird.

### Sprachsynthese

Sprachsynthese, auch Text to Speech (auch TTS) kann direkt genutzt werden, um arbiträre Texte auditiv auszugeben. Neben Beschreibungen von Szenen, Objekten und Grafiken kann so auch relativ einfach der Text von UI Elementen wiedergegeben werden. Das größte Hindernis bei der Implementation von TTS besteht darin, dass externe Libraries eingebunden werden müssen und dass eine alternative Programmiersprache zu Godots eigener Skriptingsprache, GDScript genutzt werden muss. Eine Lösung für letzteres ist die Nutzung von GDNative, einer Art Kompatibilitätsebene zwischen Godot und C++. Alternativ gibt es durch die Verwendung von Add ons die Möglichkeit, Python in Godot zu verwenden. [Leb23]

Der Ansatz, Python Libraries zu verwenden, um eine Sprachsynthese zu ermöglichen, schien eine simple Lösung zu sein. Zuerst wurde dabei die Bibliothek pyttsx3 [M B20] getestet. Diese ist einfach zu verwenden und kann offline genutzt werden. Es hat sich jedoch schnell gezeigt, dass ein sehr großes Problem in pyttsx3 darin besteht, dass die Ausgabe der Sprache den genutzten Thread blockiert, bis diese abgeschlossen ist.



Als Lösung hierfür wurde versucht, pytttsx3 auf einen separaten Thread auszulagern. Dadurch wurde das bisherige Problem gelöst und der Mainthread wurde nicht mehr blockiert, es war aber nicht möglich, die Aussprache abubrechen, wenn ein neuer Text ausgegeben werden sollte. Wenn weitere Befehle zur Sprachsynthese von neuen Texten ausgelöst wurden, hat dies dazu geführt, dass Fehler von pytttsx3 geworfen wurden. Aufgrund dieser Erfahrungen wurde entschieden, dass diese Library nicht geeignet ist, um für diesen Use Case verwendet zu werden.

Als Nächstes wurde die gTTS Library [Dur23] getestet. Diese kann nur online verwendet werden. Hierdurch können neue Probleme auftreten. An diesem Entwicklungsschritt wurde sich jedoch dafür entschieden, dass die Existenz einer TTS Lösung das Hindernis einer obligatorischen Internetverbindung überwiegt. Bei ersten Versuchen wurde außerdem erkannt, dass der Thread beim Abspielen von Sprachsynthesen nicht blockiert wird. Es hat sich aber schnell gezeigt, dass diese Bibliothek eigentlich dafür gedacht ist, dass die Sprache als Datei abgespeichert wird und nicht direkt ausgegeben wird. Es gibt aber eine Möglichkeit, den Ton im Arbeitsspeicher zwischenzulagern und von dort zum Beispiel mithilfe von pygame [pyg23] auszugeben. In Quelltext 4.5 wird ein Codeausschnitt dargestellt, der diesen Lösungsansatz demonstriert. Außerdem kann die Ausgabe dann abgebrochen werden. Somit eignet sich dieser Workaround im Rahmen der Arbeit als Sprachsynthese.

**Quelltext 4.5:** Wiedergabe von Sprachsynthese mithilfe von pygame und gTTS

```
1 from godot import exposed
2 from godot import *
3
4 from gtts import gTTS
5 import pygame
6 from io import BytesIO
7
8 @exposed
9 class TTS(Node):
10     def ready(self):
11         # setup pygame audio
12         pygame.init()
13         pygame.mixer.init()
14
15     def say(self, text: str):
16         sound = self.generate_tts(self.text)
17         self.play_tts(sound)
18
19     def generate_tts(self, text):
20         mp3_fp = BytesIO()
21         tts = gTTS(text, lang='en', tld='co.uk', slow=False)
22         tts.write_to_fp(mp3_fp)
23         return mp3_fp
24
25     def play_tts(self, sound):
26         pygame.mixer.music.stop()
```

```
27     sound.seek(0)
28     pygame.mixer.music.load(sound, "mp3")
29     pygame.mixer.music.play()
```

In dem Codeausschnitt wird eine Klasse in Python erstellt, die innerhalb eines Godot Programms genutzt werden soll. Dabei wird zuerst pygame sowie ein pygame Mixer, zur Ausgabe von Audio initiiert. Daraufhin wird eine Methode namens „say“ definiert, die von anderen Skripten aufgerufen werden soll. Hierbei wird ein String als Parameter erwartet, in dem der Text definiert wird, der mit der Text to Speech Engine wiedergegeben werden soll. In der Methode wird zuerst die private Methode „\_generate\_tts“ aufgerufen. Hierbei wird ein Bytestream erstellt, in den die Ausgabe von gTTS gespeichert wird. Dieser wird dann zurückgegeben und in der „say“ Methode an die private „\_play\_tts“ Methode weitergegeben. Dort wird bestehende Audioausgabe zuerst gestoppt und daraufhin wird die im Bytestream gespeicherte Sprachsynthese wiedergegeben. Es hat sich jedoch ein Problem an dem Add on gezeigt, durch das Python Code in Godot verwendet werden kann. Es gibt keine direkte Möglichkeit, um von GDScript aus ein Python Skript aufzurufen. Dadurch ist die Verwendung des Python Codes in einem normalen Godot Projekt sehr umständlich. Außerdem werden für den Export von erstellten Spielen weitere Zwischenschritte benötigt, da laut der Dokumentation des Add ons die Python Umgebung manuell exportiert werden muss.

Alternativ können Erweiterungen für die Godot Engine in C++ geschrieben werden. Diese bieten viele Möglichkeiten, um neue Funktionalität einzubauen. Vor allem deswegen, weil einige Einschränkungen, die die Engine an Entwickler\*innen stellt, hier nicht vorhanden sind. So ist es möglich, die Limitation von einem Skript pro Objekt mithilfe von Vererbung zu umgehen. Außerdem können externe Bibliotheken eingebunden werden, die an dieser Stelle Text to Speech ermöglichen können. Ein Hindernis, das sich mit diesem Ansatz ergibt, ist jedoch, dass die Engine mit Verwendung der Erweiterungen neu kompiliert werden muss. Somit müssen Entwickler\*innen entweder eine spezielle Version der Godot Engine nutzen, um von Sprachsynthese Gebrauch machen zu können oder diese Version selber erstellen. Die erste Möglichkeit verlangt ein hohes Vertrauen der Nutzer\*innen, dass die Version der Engine sicher ist. Der andere Ansatz dahingegen verlangt einen signifikanten Extraaufwand. Dadurch ist dieser Ansatz nur eingeschränkt zu empfehlen. Ein möglicher Anwendungsfall ist, wenn bereits eine eigene Version von Godot kompiliert wird und der einzige Mehraufwand darin besteht, dass eine weitere Erweiterung eingefügt werden muss.

## Schnittstelle

Die Godot Engine implementiert GUI im Editor und in exportierten Programmen auf eine Art, dass Screenreader diese nicht auslesen können. Es gibt zwar eine Konsole, die parallel zu den visuellen Fenstern läuft, die lesbar ist, diese verliert aber bei Interaktionen mit dem Programm den Fokus und kann selber nicht verwendet werden, um Eingaben zu tätigen. Ein Ansatz, der eine Alternative darstellt, ist, dass zusätzlich zu dem eigentlichen Spiel auch eine Konsolenanwendung entwickelt wird, die Screenreader kompatibel ist. Das Ziel ist es, dass beide Programme miteinander kommunizieren können und die Konsolenanwendung als Ausgabe und als Eingabe für das Spiel verwendet werden kann. Hierfür kann TCP genutzt werden, um zwischen den beiden Programmen zu kommunizieren. Einen Prototyp dieser Interaktion hat der Twitter Nutzer @nightblade99 demonstriert [Nig20]. Ansätze, um diese Implementation zu testen wurden gefunden, es wurde sich aber aufgrund des sehr hohen Aufwands dafür entschieden, diesen Ansatz in dieser Arbeit nicht weiterzuverfolgen.

## 4.6 Untertitel

In der Anforderung **REQ-041** wird verlangt, dass es Untertitel für die wichtigsten gesprochenen Texte gibt. In Kapitel 3.3.6 wurden verschiedene Ansätze untersucht, mit denen so ein Untertitelsystem umgesetzt werden kann. Schließlich wurde sich dafür entschieden, das System als Singleton zu implementieren und zyklische Arrays für die eingehenden Untertiteltexte zu verwenden. Zuerst wurde eine Szene erstellt, die die nötigen UI Elemente zur Darstellung der Untertitel beinhaltet. Das Stammobjekt ist vom Typ `CanvasLayer`. Dadurch werden die Untertitelinhalte über anderen UIs gerendert. Danach kommt ein `PanelContainer`, der am unteren Bildschirmrand verankert ist und so konfiguriert wurde, dass er sich bei Bedarf noch oben hin erweitert. Diesem Container ist ein `Label` als Kind zugeordnet, in dem die Untertiteltexte abgebildet werden. Der Container wird gleichzeitig auch dazu genutzt, die Lesbarkeit der Untertitel zu steigern und damit Anforderung **REQ-44** zu erfüllen. Grafik 4.5 zeigt das Resultat dieser Implementation.

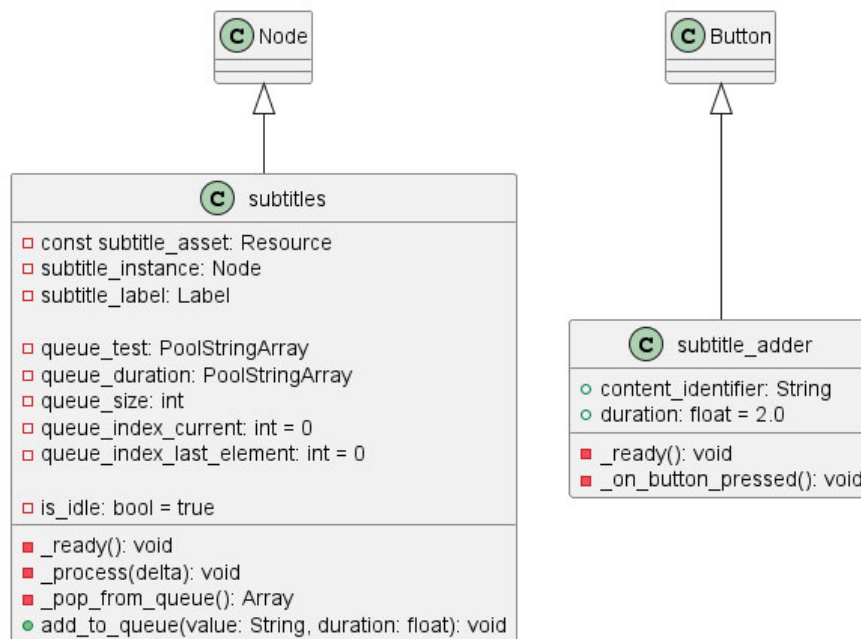
Des Weiteren gibt es ein Skript namens `subtitles`, das als Singleton verwendet wird. Hierfür wurde es in der Godot Engine als `AutoLoad` definiert. Das bedeutet, dass es beim Start des Programmes initialisiert wird und unter einem festgelegten Namen referenziert werden kann. Dieses Skript implementiert den Hauptteil des Untertitelsystems. Bei der Initialisierung instanziiert es die Szene, die das Untertitel UI beinhaltet, als Kindobjekt von sich selbst. Danach holt es sich Referenzen auf das `Label` und auf die Szene, die daraufhin ausgeblendet wird. Außerdem werden zwei zyklische Arrays initialisiert. Diese stellen die Queues für die Untertiteltexte und die Anzeigedauer dar. Zudem ist eine öffentliche Methode namens `add_to_queue`

**Abbildung 4.5:** REQ-041, REQ-044: grundlegende Untertitel, leicht lesbare Untertitel

definiert. Diese wird verwendet, um neue Untertiteltexte mit einer spezifischen Anzeigedauer in die Warteschlange einzureihen. Hierfür wird nach dem Index des aktuell letzten Elements in den Arrays gesucht. Danach wird diese Stelle mit den neuen Werten überschrieben und der Index wird inkrementiert, um bereit für den nächsten Aufruf der Methode zu sein. Falls dabei das Ende der Arrays erreicht wurde, wird der Index auf den Wert „0“ beziehungsweise auf den Anfang der Arrays gesetzt. Während das Programm läuft, wird regelmäßig überprüft, ob aktuell Untertitel angezeigt werden, wenn dies nicht der Fall ist, wird die Szene mit den UI Elementen eingeblendet. Danach wird der Text des Labels mit dem ältesten, noch nicht angezeigten Eintrag in dem Untertitelarray überschrieben. Um zu bestimmen, welche Einträge dazu zählen, wird lediglich ein weiterer Index genutzt, der kontinuierlich auf den zuletzt angezeigten Eintrag verweist. Gleichzeitig wird ein Timer gestartet, der festlegt, wie lange die Untertitel angezeigt werden sollen. Der Wert wird aus dem Array, der die Anzeigedauer speichert, entnommen. Sobald dieser abgelaufen ist, werden die UI Elemente wieder ausgeblendet und das Untertitelsystem wechselt in den Leerlaufmodus.

Ebenfalls gibt es ein weiteres Skript namens `subtitle_adder`, das beispielhaft demonstriert, wie Untertitel zu dem System hinzugefügt werden können. In diesem Fall muss es an ein Objekt des Typen `Button` angehängt werden. Es besitzt Variablen, die im Inspektor angezeigt werden, mit denen der anzuzeigende Text und die Dauer definiert werden können. Sobald der `Button` gedrückt wird, wird die `add_to_queue` Methode des Untertitelsystems aufgerufen und ein neuer Eintrag wird an die Warteschlange angehängt. Der Aufbau der Skripte ist im Klassendiagramm 4.6 visualisiert.

Abbildung 4.6: Klassendiagramm Untertitelsystem

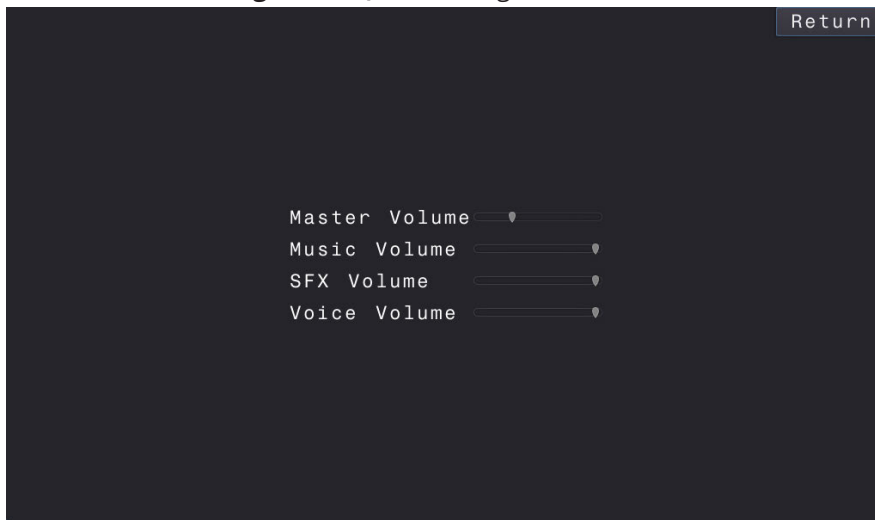


## 4.7 Variable Lautstärke

Die Anforderung **REQ-042** verlangt, dass Nutzer\*innen die Lautstärke von Musik, Effekten und Sprache anpassen können. Hierfür bietet die Godot Engine die Möglichkeit, Audio Busse zu nutzen (siehe Kapitel 3.3.7). Diese können genutzt werden, um Audiodateien über sie abzuspielen und spezifische Lautstärken und Effekte zu definieren. Standardmäßig gibt es einen Hauptbus, der gleichzeitig alle anderen Busse steuert. Damit ist es einfach, alle Audiospuren gleichzeitig still zu stellen. Zur Steuerung der Lautstärke wurde ein Menü mit Slidern erstellt (siehe Grafik 4.7). Diese sind in einem Wertebereich von kurz über 0 bis 2 definiert. Die Audiobusse dahingegen sind im Bereich von  $-80dB$  bis  $6dB$  definiert. Da die Dezibelskala logarithmisch aufgebaut ist, müssen die Werte des Sliders umgerechnet werden. Hierfür wird die Formel 4.1 verwendet. Außerdem wird geprüft, ob der minimale Wert des Sliders ausgewählt wurde. Falls dem so ist, wird der Audiobus stumm gestellt. Die Zuweisung der Lautstärke auf den relevanten Audiobus geschieht per Namen. Hierfür wird im Inspektor eine String Variable veröffentlicht.

$$y_{Lautstaerke} dB = 20 * \log_{10}(x) \quad (4.1)$$

Bei Änderungen der Lautstärke wird ein Timer gestartet. Solange dieser läuft, wird eine Audiodatei abgespielt, die den jeweils relevanten Bus nutzt. Hierdurch sollen Nutzer\*innen die Möglichkeit bekommen, die Anpassung der Lautstärke einschätzen

**Abbildung 4.7:** REQ-042: konfigurierbare Lautstärke

zu können. Damit beim Öffnen des Menüs die Slider die richtigen Werte besitzen, wird mit Formel 4.2 der aktuelle Dezibelwert des Audiobusses in den linearen Wertebereich des Sliders umgewandelt.

$$y = 10^{(x_{Lautstaerke\text{dB}})/20} \quad (4.2)$$

## 4.8 Accessibility Informationen & Feedback

Nutzer\*innen sollen die Möglichkeit haben, sich vor dem Start des Spiels und im Spiel selbst über die implementierten Accessibility Features zu informieren (**REQ-052, REQ-053**). Außerdem soll es möglich sein, Feedback zu geben (**REQ-055**). Hierfür wurde eine Datei erstellt, die die Informationen und Kontaktmöglichkeiten beinhaltet (siehe Grafik 4.8). Diese befindet sich im Hauptordner des Projektes. GitHub, die Website, auf der das Projekt veröffentlicht wurde, stellt diese Datei direkt dar. Dadurch können Interessenten einfach vor dem Download der Dateien einsehen, welche Features eingebaut wurden. Außerdem wurde ein Menü eingebaut, in dem per Labels dieselben Informationen abgebildet werden (siehe Grafik 4.9). Neben der so gegebenen Kontaktmöglichkeit per E Mail, bietet GitHub selber die Möglichkeit, per „Issues“ Vorschläge und Fehlermeldungen zu verfassen. Außerdem wurde der Kontakt zu der Zielgruppe per Discord aufgenommen und die Möglichkeit gestellt, über diesen Kommunikationskanal ebenfalls Rückmeldungen zu geben.

**Abbildung 4.8:** REQ-052. REQ-055: Accessibility Informationen außerhalb des Spiels, Feedbackmöglichkeit

```
# Godot Game Accessibility Guidelines
This project can be used as a starting point for a new Godot project.
It includes basic systems to use accessibility features as described by the [Game accessibility
guidelines](https://gameaccessibilityguidelines.com/).

The features are mostly implemented in addons. Also please read the Readme files inside of the
specific addons. They will explain specific things about the features and how to use them.

## Implemented Accessibility Features:
- the controls can be remapped
- the UI can be controlled via mouse or keyboard
- the sensitivity of controls can be adjusted
- the default font is easily readable
- the default font size is chosen to be easily readable
- separate volume controls are provided for effects, speech, background and music
- information about the implemented accessibility features is provided to the user

## Notes
- for different UI elements (e.g. ingame UI), the "ui_theme" might have to be reapplied
  - otherwise the separation of elements might be too small

## Other
Feedback: milasbaumgarten@gmail.com
```

**Abbildung 4.9:** REQ-053. REQ-055: Accessibility Informationen innerhalb des Spiels, Feedbackmöglichkeit

```
Implemented Accessibility Features: Return
- the controls can be remapped
- the UI can be controlled via mouse or
  keyboard
- the sensitivity of controls can be adjusted
- the default font is easily readable
- the default font size is chosen to be easily
  readable
- separate volume controls are provided for
  effects, speech and music
- information about the implemented
  accessibility features is provided to the user
- information about the implemented
  accessibility features is provided to the user
  before starting the game
- information about the implemented
  accessibility features is provided to the user
  before starting the game
- basic subtitles exist

Feedback: milasbaumgarten@gmail.com
```

## 4.9 Persistente Optionen

Bestimmte Optionen und Konfigurationen müssen zwischen Spielesessions gespeichert werden (**REQ-054**). Gleichzeitig müssen die Einstellungen an verschiedenen Stellen im Code zugänglich sein. Hierfür bietet sich eine Struktur, wie zum Beispiel ein Observer Pattern oder ein Singleton an. Für diesen Anwendungsfall, in dem lediglich Variablen gesetzt und ausgelesen werden müssen, kann jedoch auch ein anderer Ansatz verwendet werden, den die Godot Engine bietet. Es gibt die Möglichkeit, eigene Ressourcen zu definieren. Diese sind Dateien im Projektordner, die arbiträre, definierte Informationen speichern können. Laut der Dokumentation <sup>1</sup> sind Ressour

<sup>1</sup>[https://docs.godotengine.org/en/stable/tutorials/best\\_practices/node\\_alternatives.html](https://docs.godotengine.org/en/stable/tutorials/best_practices/node_alternatives.html)

cen performanter als Nodes. Dadurch ist es aus Sicht der Optimierung sinnvoller diese, anstelle eines Singeltons zu verwenden. Außerdem bieten sie inhärent die Möglichkeit, serialisiert und deserialisiert zu werden. Dadurch ist es einfach, einen Speicher und Ladevorgang zu gestalten. Godot bietet hierfür die ResourceSaver und ResourceLoader Klassen. Die Verwendung ist in Listing 4.6 dargestellt.

**Quelltext 4.6:** Speichern und Laden einer benutzerdefinierten Ressource namens „settings“

```
1 # speichern
2 ResourceSaver.save(settings file path , settings)
3
4 # laden
5 settings ResourceLoader.load(settings file path)
```



## 5 Evaluation und Interpretation der Ergebnisse

In Kapitel 3.1 wurden die Anforderungen vorgestellt, die in dieser Arbeit bearbeitet wurden. Daraufhin wurden in Kapitel 3.3 Konzepte dargestellt, um diese Anforderungen zu implementieren. Anschließend wurden im letzten Kapitel beschrieben, wie die Implementation durchgeführt wurde. Nun wird in diesem Kapitel die Evaluation der Implementation beschrieben. Ebenfalls wird die Evaluation ausgewertet. Dabei werden zwei verschiedene Aspekte untersucht. Zum einen die Usability, die mithilfe von Umfragen und Statistik evaluiert wird und zum anderen die Erfüllung des Anforderungskatalogs. Bei letzterem wird die Evaluation inhaltlich durchgeführt.

### 5.1 Usability

Um die Usability der implementierten Accessibility Features zu untersuchen, wurden zwei Umfragen im Rahmen dieser Arbeit durchgeführt. Zum einen wurde Teilnehmer\*innen des Godot Wild Jams die Implementation zur Verfügung gestellt. Eine detaillierte Beschreibung des Godot Wild Jams erfolgt nachfolgend in Kapitel 5.1.1. Die Teilnehmer\*innen konnten dann die Implementationen, die in dieser Arbeit entstanden sind, bei der Entwicklung eines eigenen Spiels nutzen und testen. Danach wurden die Entwickler\*innen nach ihrer Nutzung und ihrer Meinung betreffend der Implementationen befragt. Zum anderen wurde Studierenden eine Liste an Aufgaben gegeben, mit denen sie die implementierten Features anwenden sollten (siehe Kapitel 5.1.2). Außerdem sollten analoge Aufgaben ausgeführt werden, die nur die Möglichkeiten nutzen, die die Godot Engine bietet. Hiermit sollte die Verwendung der Accessibility Features simuliert werden. Parallel dazu wurde von den Studierenden ein Fragebogen ausgefüllt, in dem der Aufwand und das Verständnis der durchgeführten Lösungswege bewertet wurde. Hierdurch sollte untersucht werden, wie viel Extraaufwand die Nutzung der Implementation dieser Arbeit für eine\*n neue\*n Nutzer\*in darstellt und wie verständlich diese sind.

#### 5.1.1 Godot Wild Jam

Bei dem Godot Wild Jam handelt es sich um einen Game Jam. Dieser findet jeweils am zweiten Freitag jedes Monats für neun Tage statt [Wil23]. Er wurde im September 2018 gegründet. Eine Besonderheit gegenüber anderen Game Jams ist, dass nur die Godot Game Engine verwendet werden darf. Außerdem wird in der Bewertung der Spiele die Barrierefreiheit auf Grundlage der Game Accessibility Guidelines [EII+] betrachtet. Für die Auswertung der Implementationen, die in dieser Arbeit durchgeführt

wurden, sollten Teilnehmer\*innen des Godot Wild Jams im Rahmen des Game Jams die Accessibility Features für ein eigenes Projekt nutzen. Dazu wurde ein Fragebogen auf Englisch erstellt (siehe Anhang B). In diesem werden zuerst soziodemografische Fragen gestellt, mit denen eine Einordnung der Teilnehmer\*innen betreffend ihrer Kenntnisse der Godot Engine und von Accessibility Features getätigt wird. Danach wird für die einzelnen implementierten Features gefragt, ob diese genutzt werden und falls dem so ist, wird die Nutzbarkeit auf Grundlage der System Usability Scale [Bro95, S. 3f] abgefragt. Hierbei werden die Bereiche Tastenbelegung und Maussensitivität, Tastatursteuerung des UIs, Textlesbarkeit, Lautstärkeinstellungen, Untertitel und Informationen über die implementierten Accessibility Features behandelt. Mit dieser Aufteilung werden alle Teile des Anforderungskatalogs, die implementiert wurden, abgedeckt.

Es wurden ursprünglich hohe Erwartungen in die Evaluation durch die Teilnehmenden des Godot Wild Jams gelegt, da diese wahrscheinlich einen hohen Wissensstand in der Nutzung der Godot Engine besitzen und eine extrinsische Motivation haben, Accessibility Features in ihre Projekte einzubauen. Es wurde jedoch keine einzige Antwort erhalten. Dadurch kann dieser Teil der Evaluation nicht weiter durchgeführt werden. Eine Nichtteilnahme an der Evaluation kann verschiedene Ursachen haben. Eine Auswahl wird nachfolgend vorgestellt. Der erste Grund für die fehlende Teilnahme an der Umfrage kann darin begründet sein, dass Evaluationen für Teilnehmer\*innen Druck verursachen und sich teilweise wie eine Prüfung anfühlen können [HM21]. Diese Hürde kombiniert sich damit, dass Teilnehmende einer Game Jam dies für gewöhnlich in ihrer eigenen Freizeit machen und in ihrem normalen Alltag bereits weiteren Beschäftigungen nachgehen. So hat eine Studie von Borg et al. [Bor+20, S. 5f] gezeigt, dass die Personen, die an einer Game Jam teilnehmen, primär Studierende und Menschen sind, die professionell in der Spieleindustrie oder der Softwareentwicklung arbeiten. Melo et al. [Mel+19, S. 4f] kamen zu einem ähnlichen Ergebnis. Die Untersuchung von Umbach [Umb04, S. 28-35] betrachtete verschiedene Aspekte, die zu einer Verbesserung der Akzeptanz einer Online Umfrage führen sollen. Hierbei schreibt er zuerst davon, dass das grafische Design der Umfrage eine große Rolle spielen kann, aber dass in der vorhandenen Literatur kein explizites Design empfohlen wird. Aus diesem Grund wurde für die Evaluation die Entscheidung getroffen, Google Forms zu nutzen. Hierbei wurde davon ausgegangen, dass das Design bereits für eine hohe Akzeptanz entwickelt wurde und dass diese Darstellung den meisten Teilnehmenden bereits bekannt ist. Der Grund hierfür ist die hohe Präsenz von Google als Softwaredienstleistungsunternehmen im Internet. Dadurch wird vermutet, dass das vorgegebene Design ausreichend getestet und für die Teilnahme an Umfragen optimiert ist.

Außerdem schreibt Umbach [Umb04, S. 28-35], dass es innerhalb der Umfrage keine Barrieren, wie zum Beispiel Passwörter geben sollte, die Proband\*innen außerhalb der gewählten Stichprobenmenge an der Teilnahme hindert. In dieser Arbeit wurde die Umfrage so gestaltet, dass keine Barrieren implementiert wurden, die Personen an der Teilnahme hindern könnten. Des Weiteren sollten Erinnerungen an die Zielgruppe der Umfrage geschickt werden [LH17, S. 82f], damit die Teilnahme erhöht wird. Ohne weitere Kontaktaufnahmen sinkt die Antwortrate laut Cook et al. auf circa 30% [CHT00, S. 11]. Außerdem ist die Teilnahme laut Crawford et al. [CCL01, S. 3] innerhalb der ersten Stunden und Tagen am höchsten und fällt dann signifikant. Des Weiteren hatten Crawford et al. [CCL01, S. 3] beobachtet, dass bereits eine einzige Erinnerung die Teilnahme verdoppelt. In der hier durchgeführten Umfrage wurde ebenfalls nach ein paar Tagen eine Erinnerungsnachricht verfasst und an die potenziellen Proband\*innen geschickt. Außerdem hat die Untersuchung von Umbach [Umb04, S. 28-35] herausgefunden, dass es wenig Forschung dazu gibt, wie effektiv es ist, die Aufrufe zur Teilnahme an einer Online Umfrage zu personalisieren. Bei physischen Umfragen gibt es jedoch eindeutige Vorteile, wenn der Aufruf zur Teilnahme und Erinnerungen personalisiert sind. Dieser Aspekt wurde in der durchgeführten Evaluation nicht ausreichend genutzt. Vor der Durchführung der Umfrage wurde informell untersucht, wie hoch der Zuspruch für das zu evaluierende Projekt unter den Proband\*innen ist. Hier wurde in kurzer Zeit viel positives Interesse geäußert. Mit dem Ziel, Barrieren zu vermeiden und möglichst vielen Personen die Nutzung der in dieser Arbeit entwickelten Implementation zu ermöglichen, wurde der Zugriff öffentlich ermöglicht, ohne dass weitere Schritte oder Einschreibungen nötig sind. Hierbei hat sich jedoch im Nachhinein gezeigt, dass es so nicht möglich ist nachzuvollziehen, wie groß die Nutzung der zu evaluierenden Implementation ist. Es war lediglich möglich, bei einer Person herauszufinden, dass sie das zur Verfügung gestellte Projekt genutzt hat und die implementierten Accessibility Features in zwei Projekten eingebaut wurden. Dies liegt darin, dass diese Person ihr Interesse an dem Projekt in Form von Nachrichten und der Meldung von Verbesserungsvorschlägen gezeigt hat. Über die Kontaktadresse der Person war es dann möglich, unter den eingesendeten Spielen des Godot Wild Jam, diese zu finden, an denen die Person mitgewirkt hat. Ein Test des Spiels, sowie eine Untersuchung des Quellcodes haben daraufhin bestätigt, dass Teile der Implementation dieser Arbeit genutzt wurden.

Durch diese fehlende Nachvollziehbarkeit war es ebenfalls nicht möglich, spezifische Personen zur Teilnahme an der Umfrage aufzufordern oder zu einem Interview einzuladen. Ein weiteres Hindernis, das sich aus dem Zeitdruck eines Game Jams und der vorhandenen Arbeitslast der Teilnehmer\*innen ergeben haben kann, ist, dass die Nutzung von externem Code eine gewisse Einarbeitungszeit benötigt. Dies in Kombination mit der potenziellen Ungewissheit der Entwickler\*innen, ob sich die Nutzung

für sie lohnt, kann dazu geführt haben, dass das in dieser Arbeit entwickelte Add on nur geringe Verwendung genossen hat. Abschließend kann somit betrachtet werden, dass die gewählte Herangehensweise, um diese spezifische Zielgruppe zur Teilnahme an einer Evaluation zu motivieren, nicht erfolgreich ist. Für zukünftige Forschungen mit dieser Zielgruppe ist es nötig, die Umfrage auf eine andere Art durchzuführen.

### 5.1.2 Studierende

Da die ursprünglich geplante Evaluation mit den Teilnehmer\*innen des Godot Wild Jam keinen Erkenntnisgewinn zuließ, wurde sich kurzfristig dafür entschieden, eine weitere Evaluation mit einer anderen Zielgruppe durchzuführen. Für die Teilnahme an der Umfrage zur Nutzbarkeit der implementierten Lösungen wurden primär Studierenden und Alumni der Hochschule Mittweida kontaktiert. Hierfür werden ihnen Arbeitsblätter zur Verfügung gestellt, mit denen eine Nutzung der implementierten Accessibility Features in einem kompakten zeitlichen Rahmen simuliert wird. Bei den Proband\*innen handelt es sich primär um Studierende des Studienganges „Medieninformatik und interaktives Entertainment“. Da öffentliche Kommunikationskanäle genutzt wurden, um die Studierenden zu erreichen, ist es aber auch möglich, dass Studierende anderer Studiengänge an der Umfrage teilgenommen haben. Dies sollte jedoch wenig Einfluss auf die Antworten haben, da der Fragebogen so gestaltet wurde, dass keine Vorkenntnisse nötig sind. Somit ist es auch für Personen, die nicht der primären Zielgruppe angehören möglich einen Beitrag zu der Evaluation der in dieser Arbeit getätigten Evaluationen zu leisten.

Die Entscheidung, den spezifischen Studiengang als primäres Anlaufziel für die Suche nach Proband\*innen zu nutzen, liegt darin begründet, dass während des Studiums Grundlagen der Informatik und der Spieleentwicklung vermittelt werden. Dadurch kann davon ausgegangen werden, dass ein grundlegendes Verständnis für Anforderungen und Prozesse der Spieleentwicklung vorhanden sind und erfolgreiche Einschätzungen der Implementation der Accessibility Features getätigt werden können.

### Fragebögen

Die Fragebögen (siehe Anhang C und D) sind so aufgebaut, dass die Proband\*innen danach befragt werden, wie sie das Verständnis und die Schwierigkeit der Aufgabe, sowie den Aufwand und den Mehrwert des Lösungsansatzes einschätzen. Außerdem wird gefragt, ob sie den Lösungsansatz erneut verwenden würden. Der Fragebogen umfasst außerdem soziodemografische Fragen und die Möglichkeit, freies Feedback zu den Aufgabenstellungen oder der Implementation zu geben. Im Gegensatz zu dem Fragebogen, den die Teilnehmer\*innen des Godot Wild Jams erhalten haben, ist dieser auf Deutsch verfasst worden.

Es wird davon ausgegangen, dass die Studierenden wenig bis keine Erfahrung mit der Nutzung der Godot Engine besitzen. Deswegen werden die Aufgaben so beschrieben und mit Anleitungen versehen, dass möglichst wenig vorherige Kenntnisse nötig sind, um an der Evaluation teilnehmen zu können (siehe Anhang E). Dadurch soll auch ein expliziter Lösungsansatz genutzt werden, der für die einzelnen Accessibility Features angedacht ist. Außerdem wurde sich dafür entschieden, die Aufgaben in zwei Teile aufzuspalten. In Teil A wird die Nutzung der Implementationen dieser Arbeit untersucht. In Teil B dahingegen werden dieselben Aufgaben nur mit den Werkzeugen, die die Engine besitzt, gelöst. Außerdem kommt im zweiten Teil ein Accessibility Feature hinzu, das bereits komplett in der Engine implementiert ist und in dieser Arbeit keine weitere Bearbeitung benötigte. Dahingegen werden zwei Aufgaben, die im ersten Teil inkludiert sind, ausgelassen, da sie nur mit weiterem Programmieraufwand lösbar wären und es keine native Alternative gibt.

Im Fragebogen werden die Proband\*innen zuerst darum gebeten, ihren eigenen Wissensstand in den Bereichen Godot Engine, Game Development, Implementation von Accessibility Features und der Nutzung von Add ons in der Godot Engine zu bewerten. Die Antwortmöglichkeiten hierfür sind „kein Wissen“, „Anfänger“, „Fortgeschrittener“ und „Experte“. Weitere Fragen betreffend des Alters und des Geschlechtes werden nicht in der Umfrage aufgenommen, da diese in der Bewertung der in dieser Arbeit implementierten Accessibility Features nicht weiter betrachtet werden. Stattdessen würden sie ungenutzt bleiben und potenziell die Privatsphäre der Proband\*innen beeinträchtigen. Die Wahl der Fragen zur Bewertung der Aufgabenbereiche fiel auf fünf Variablen, die im nächsten Absatz genauer vorgestellt werden. Es wurde sich dafür entschieden, möglichst wenig Fragen zu nutzen, damit der Aufwand für die Teilnehmenden möglichst gering gehalten wird. Hiermit soll die Antwortrate erhöhen werden. Die Annahme hierfür liegt darin begründet, dass die Forschung von Rolstad et al. [RAR11, S. 1107] eine potenzielle negative Korrelation zwischen der Länge einer Umfrage und der Antwortrate entdeckt haben. Diese Erkenntnis wurde von den Forschungen von Sharma, sowie Galesic und Bosnjak bestätigt [Sha22, S. 68; GB09, S. 358].

Zuerst wird gefragt, ob die Aufgabenstellung schwer zu verstehen ist. Danach, ob es einfach war, die Aufgabe zu erfüllen. Als nächstes, ob die Proband\*innen den vorgestellten Lösungsansatz auch in zukünftigen Anwendungen verwendet werden würden. Daraufhin, ob die Lösung als unnötig komplex bewertet wird und abschließend, ob der Mehrwert des Lösungsansatzes für die Teilnehmenden ersichtlich ist. Die Antwortmöglichkeiten sind eine Likert Skala mit fünf Möglichkeiten, von trifft zu,

bis trifft nicht zu. Durch die Wahl von einer Skala mit einer neutralen Option sollten die Teilnehmer\*innen der Befragung die Möglichkeit haben, eine neutrale Antwort zu geben.

Mit den gewählten Variablen sollen verschiedene Erkenntnisse über die evaluierte Implementation gewonnen werden. Die erste Frage nach der Verständlichkeit der Aufgabenstellung dient als Vergleichswert, damit unterschieden werden kann, ob eine negative Einschätzung der implementierten Accessibility Features darin begründet ist, dass diese schwer nutzbar ist oder ob die Aufgabenstellung, die von den Proband\*innen bearbeitet werden sollen, nicht präzise genug formuliert ist. Die nächste Variable, ob die Aufgabe einfach zu lösen war, soll zeigen, wie schwierig die durch die Aufgabenstellung simulierte Nutzung der Implementierung wahrgenommen wird. Als Nächstes soll die Frage nach einer potenziellen Wiederverwendung des Lösungsansatzes eine Auskunft darüber geben, ob die Teilnehmer\*innen den genutzten Lösungsansatz als zielführend empfinden, um vergleichbare Aufgaben und Probleme zu lösen. Die vorletzte Variable ist eine Einschätzung der relativen Komplexität der Aufgabenlösung. Hiermit soll ein Rückschluss darüber getroffen werden, ob Aspekte der implementierten Skripte von den Nutzenden als hinderlich empfunden wurden und die Anwendung des jeweiligen Features spezifischer gestaltet werden sollte. Abschließend wird untersucht, ob Teilnehmende den Mehrwert des Lösungsansatzes erkannt haben. Hiermit wird angestrebt, dass die vorherigen Einschätzungen insoweit eingeordnet werden können, dass negative Bewertungen eine potenzielle Korrelation damit besitzen, dass der Nutzen des Accessibility Features nicht erkannt wurde.

Um einer ungewollten Tendenz der Antworten entgegenzuwirken, werden die Fragen so gestellt, dass zwei negativ und drei positiv formuliert sind. Hiermit soll einem Bias aufgrund von Zustimmungen zu den Fragestellungen entgegengewirkt werden. Die Relevanz so einer Maßnahme zeigt die Forschung von Hill und Roberts [HR23, S. 13f], sowie von De Juan und Koos [DK21, S. 636f]. Es gibt jedoch auch Forschungen, die davon ausgehen, dass diese Praxis wenig Nutzen besitzt und die negativen Aspekte, wie Missverständnisse und ein Vergessen der Forschenden, die Skalen bei der Auswertung zu invertieren, den Nutzen überwiegt [SL11, S. 3]. Da die Verwendung von negativen und positiven Fragen jedoch einen Quasi Standard darstellt [Bro95, S. 3f], wurde sich dafür entschieden, diesem in dieser Arbeit ebenfalls zu folgen.

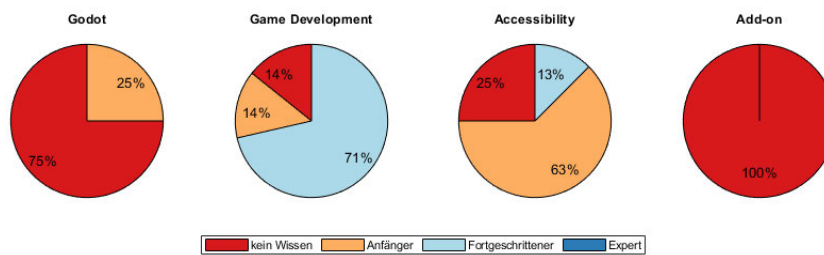
Die Evaluation wurde so durchgeführt, dass die Teilnehmer\*innen zufällig einer von zwei Gruppen zugeordnet wurden. In beiden Gruppen werden dieselben Fragen und Aufgaben bearbeitet, aber die Reihenfolge der Teile A und B ist getauscht. Während der Erstellung der Evaluation wurde davon ausgegangen, dass die Proband\*innen

während der Bearbeitung der Aufgaben potenziell einen hohen Lerneffekt haben und dadurch den zuletzt bearbeiteten Teil positiver, als den zuerst bearbeiteten Teil bewerten, da sie ein höheres Verständnis der Godot Game Engine und der Aufgaben gebiete besitzen. Hierfür wurde eine selbst erstellte Website als Anlaufstelle genutzt, die die Proband\*innen komplett zufällig eine Gruppe zuweist. Da hierbei keine Statistik geführt wurde, ist keine perfekte Verteilung von 50% Teilnahmen je Gruppe gewährleistet. Diese Einteilung hat über die Reihenfolge entschieden, in der die Aufgaben bearbeitet werden sollten. Entweder wurden zuerst die Aufgaben gestellt, die nur die Godot Engine verwenden und dann die, in denen die Implementation dieser Arbeit genutzt werden. Hierdurch wird gewährleistet, dass der Lerneffekt bei Proband\*innen während der Bearbeitung der Aufgaben in der Auswertung beachtet werden kann.

Damit eine statistische Auswertung der Antworten durchgeführt werden kann, wurden im Nachhinein die Antworten der Likert Skalen als Zahlenwerte kodiert. Hierfür wurden die Antworten in der Reihenfolge „trifft nicht zu“ bis „trifft zu“ mit den Werten 1 bis 5 ersetzt. Außerdem wurden die Antworten der negativ gestellten Fragen invertiert. Dadurch wird die Interpretation der Ergebnisse vereinfacht und laut Sauro und Lewis werden so Fehler in der Auswertung reduziert [SL11, S. 3]. Des Weiteren kann nun eine Bewertung mit einem hohen Zahlenwert „positiv“ verstanden werden, da dieser Wert mit einer hohen Zustimmung der positiven Aspekte der Variablen korreliert.

### **Deskriptive Statistik**

Es gab acht Teilnahmen der Umfrage. Fünf wurden zu Gruppe A zugewiesen (siehe Anhang F) und drei zu Gruppe B (siehe Anhang G). Die geringe Teilnahmerate lässt vermuten, dass die Entscheidungen in der Erstellung des Fragebogens, die in Kapitel 5.1 getroffen wurden, notwendig waren. Aufgrund des geringen Probenumfangs können im Weiteren keine eindeutigen Aussagen getroffen werden, aber es ist möglich, Tendenzen zu erkennen. In der Konzeption der Umfrage wurde davon ausgegangen, dass die Teilnehmer\*innen wenig bis keine Erfahrung im Umgang mit der Godot Engine besitzen. Diese Vermutung wurde bestätigt, da lediglich 25% (2 Personen) der Befragten sich als Anfänger in der Verwendung der Godot Engine eingestuft haben und die anderen Probandinnen ausgesagt haben, dass sie keinerlei Erfahrung besitzen. Dahingegen stufen 62,5% (5 Personen) der Teilnehmenden ihr Wissen in der Spieleentwicklung als fortgeschritten ein. Eine Person (12,5%) beschreibt ihren Wissensstand als dem Anfängerlevel entsprechend, eine weitere Person gibt an, kein Wissen in diesem Bereich zu besitzen und es gab eine Enthaltung. 62,5% (5 Personen) der Proband\*innen geben an, sich auf einem Anfängerlevel mit der Implementation

**Abbildung 5.1:** Soziodemografische Antworten

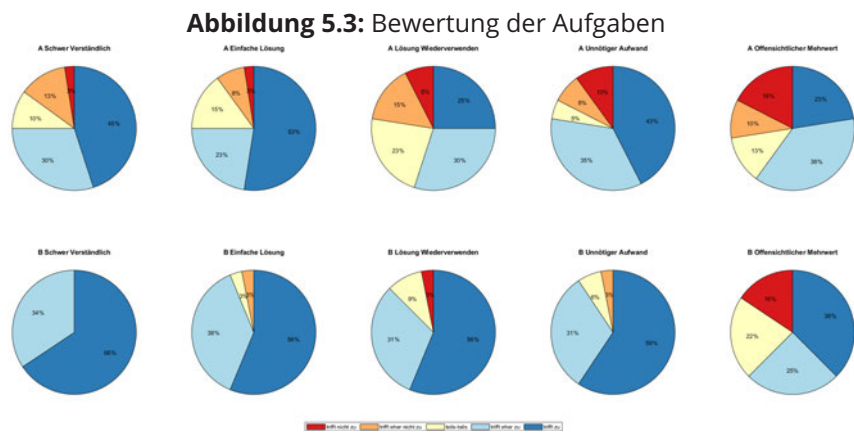
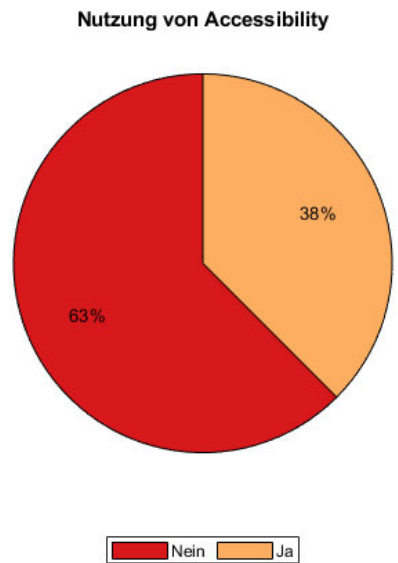
von Accessibility Features auszukennen. Zwei Personen (25%) geben an, kein Wissen auf diesem Feld zu besitzen und eine Person (12,5%) bewertet ihr Wissen als fortgeschritten. Dies ist in Grafik 5.1 zu sehen.

Hiermit wird gezeigt, dass die Umfrage eine Personengruppe umfasst, die größtenteils Erfahrung im Bereich der Spieleentwicklung besitzen und teilweise Vorkenntnisse im Bereich der Entwicklung von Accessibility Features besitzen. Dadurch kann für die weitere Evaluation davon ausgegangen werden, dass bei den meisten Proband\*innen zumindest ein grundlegendes Verständnis von Game Engines vorhanden ist. Somit kann vermutet werden, dass die Nutzung der Godot Engine, auch wenn sie für viele der Teilnehmenden unbekannt ist, keine signifikanten Hindernisse darstellen sollte. Bei der Nutzung von Add-ons in der Godot Engine hat keiner der Teilnehmenden Vorerfahrung. Außerdem haben 3 Proband\*innen (37,5%) angegeben, dass sie selber Accessibility Features benutzen (siehe Grafik 5.2). Im Vergleich zu der Forschung von PopCap Games erscheint diese Zahl repräsentativ für Videospiele zu sein, da laut der Untersuchung rund 20% der Konsumierenden mit einer Behinderung leben [Con08].

Des Weiteren hat sich gezeigt, dass 75% der Proband\*innen die Aufgaben im Aufgabenbereich A (eher) nicht als schwer verständlich einschätzen. Aufgabenbereich B wird von allen Teilnehmenden (eher) nicht als schwer verständlich eingeschätzt. Die Lösungen im Bereich A werden von 76% der Teilnehmer\*innen als (eher) leicht bewertet. In Teil B sind es dahingegen 94%. Der Wunsch, die Lösungsansätze wieder zu verwenden, ist dahingegen nicht so eindeutig. In 55% der Antworten wird einer Wiederverwendung (eher) zugestimmt. 22,5% sind sich unsicher und in 22,5% der Antworten wird eine erneute Nutzung (eher) abgelehnt. Diese Bewertungen sind in Grafik 5.3 dargestellt.



**Abbildung 5.2:** Nutzung von Accessibility Features



**Schließende Statistik**

Nachdem nun die erhobenen Daten beschrieben wurden, sollen in diesem Abschnitt Hypothesen beantwortet werden, die sich aus der Zielstellung der Arbeit und aus den Annahmen der Erstellung der Evaluation ergeben haben. Die Hypothesen sind die folgenden:

- **H1** Die Bewertung der Aufgaben in Teil A unterscheidet sich signifikant von denen von Teil B.
- **H2** Je besser die Aufgabenstellung verstanden wurde, desto positiver wurden die anderen Bereiche der Aufgabe bewertet.
- **H3** Aufgaben werden positiver bewertet, wenn bereits ähnliche Aufgaben bearbeitet wurden.

Zuerst soll nun die erste Hypothese untersucht werden. Hierfür müssen die Bewertungen der Aufgaben anhand der fünf festgelegten Variablen betrachtet werden. Zuerst wird mithilfe des Medians untersucht, ob es einen Unterschied zwischen den Aufgabenteilen gibt. Hierbei werden die Antworten beider Gruppen zusammengeführt und in die Aufgabenbereiche A und B aufgeteilt. Daraufhin werden Antworten der Aufgaben zusammengefasst, damit alle Antworten, die zu einer Variable gehören, miteinander gruppiert sind. Hierbei werden jedoch nur die Aufgaben in die Auswertung aufgenommen, die zueinander äquivalent sind. Dadurch ergibt sich, dass die Aufgabenstellungen A2, A3 und B4 an dieser Stelle nicht weiter beachtet werden. Die Wahl der Metrik begründet sich darin, dass alle Bewertungen zu einer Metrik zusammengeführt werden sollten, um einen Gesamteindruck zu gewähren. Dadurch bieten sich vor allem der Mittelwert und der Median an. Damit potenzielle Ausreißer keinen starken Einfluss auf die Auswertung haben, wurde sich für den Median entschieden. Die berechneten Mediane sind in Tabelle 5.1 zu sehen.

**Tabelle 5.1:** Median der Antworten, getrennt in Aufgabenteil A und B

<b>Variable</b>	<b>Median Teil A</b>	<b>Median Teil B</b>
Leicht Verständlich	4,0	5,0
Einfache Lösung	4,5	5,0
Lösung Wiederverwenden	4,0	5,0
Passender Aufwand	4,0	5,0
Offensichtlicher Mehrwert	4,0	4,0

Aus der Tabelle 5.1 zeigt sich, dass der Teil B in allen Variablen, außer im offensichtlichen Mehrwert eine positivere Bewertung erhalten hat. Nun muss geprüft werden, ob dieses Ergebnis eine statistische Signifikanz aufweist und somit auf die Grundgesamtheit übertragen werden kann. Hierfür muss zunächst überprüft werden, ob es sich um eine Normalverteilung handelt, da dies über die weiteren Tests entscheidet. Für die Wahl des richtigen Tests muss zuerst untersucht werden, wie hoch die Stichprobengröße ist. Diese errechnet sich aus der Anzahl der Teilnehmenden und der Anzahl an Fragen, da die Antworten aller untersuchten Fragen kombiniert werden. Da bei Teil A und Teil B je drei Fragen untersucht werden und es acht Proband\*innen gibt, ergibt sich eine Stichprobengröße von  $n = 24$ . Da diese relativ klein ist, wird in dieser Arbeit der Kolmogorov Smirnov Test angewandt. Dieser kann auch bei kleinen Datenmengen ( $n > 20$ ) genutzt werden. Bei diesem Test wird der maximale Abstand zwischen der spezifischen Verteilungsfunktion und der Normalverteilung überprüft. Dabei ist die Nullhypothese, dass eine Normalverteilung vorliegt. Wenn dahingegen die Signifikanz unter dem Grenzwert von 0,05 liegt, kann nicht davon ausgegangen werden, dass es sich um diese spezifische Verteilung handelt [SH06, S. 337-339].

**Tabelle 5.2:** Kolmogorov-Smirnov-Test für Aufgabenteil A

Variable	p
Leicht Verständlich	1,36e 21
Einfache Lösung	1,36e 21
Lösung Wiederverwenden	1,36e 21
Passender Aufwand	3,90e 18
Offensichtlicher Mehrwert	1,60e 16

**Tabelle 5.3:** Kolmogorov-Smirnov-Test für Aufgabenteil B

Variable	p
Leicht Verständlich	1,38e 22
Einfache Lösung	1,36e 21
Lösung Wiederverwenden	1,01e 20
Passender Aufwand	1,36e 21
Offensichtlicher Mehrwert	2,43e 17

Wie sich aus den Tabellen 5.2 und 5.3 zeigt, handelt es sich bei keiner der überprüften Variablen um eine Normalverteilung. Dadurch kann kein t Test angewendet werden [SL21, S. 351]. Stattdessen kann ein nicht parametrischer Wilcoxon Test genutzt werden. Die Ergebnisse des Tests sind in Tabelle 5.4 zu sehen.

**Tabelle 5.4:** H1: Wilcoxon-Test

Variable	p	z	r
Leicht Verständlich	0,02	2,43	0,50
Einfache Lösung	0,23		
Lösung Wiederverwenden	0,03	2,15	0,44
Passender Aufwand	0,04	2,03	0,41
Offensichtlicher Mehrwert	0,66		

Dieser zeigt, dass für die Variablen „leicht verständlich“, „Lösung wiederverwenden“ und „passender Aufwand“ ein signifikanter Unterschied in der Bewertung der Aufgaben zwischen Teil A und Teil B besteht, da für die Signifikanz jeweils  $p < 0,05$  gilt. Die Variablen „einfache Lösung“ ( $p = 0,23$ ) und „offensichtlicher Mehrwert“ ( $p = 0,66$ ) zeigen dahingegen keinen signifikanten Unterschied auf. Als Nächstes wird die Effektstärke anhand von Gleichung 5.1 berechnet. Diese kann dann mit den Effektgrenzen von Cohen verglichen werden [Coh88, S. 79 81].

$$r = \left| \frac{z}{\sqrt{n}} \right| \quad (5.1)$$

Hiermit ergibt sich eine Stärke, die größer ist, als 0,4 für die Variablen „leicht verständlich“ ( $r = 0,50$ ), „Lösung wiederverwenden“ ( $r = 0,44$ ) und „passender Aufwand“ ( $r = 0,41$ ). Laut den Effektgrenzen von Cohen bedeutet dies, dass es sich hier um einen starken Effekt handelt. Zusammenfassend kann also gesagt werden, dass bei den Variablen „einfache Lösung“ und „offensichtlicher Mehrwert“ kein signifikanter Unterschied zwischen der Bearbeitung der Aufgabenbereiche besteht. Die Bewertung für diese Variablen wurden mit einem Median im Wert von 4 beziehungsweise der Kategorie „trifft eher zu“ bewertet. Dadurch kann hier davon ausgegangen werden, dass die Proband\*innen die Komplexität der Nutzung der implementierten Accessibility Features und den Mehrwert äquivalent zu der vergleichbaren Nutzung der Godot Engine einschätzen. Dahingegen zeigt sich ein signifikanter Unterschied der Bewertung der Variablen „leicht verständlich“, „Lösung wiederverwenden“ und „passender Aufwand“. Diese werden ebenfalls für die Accessibility Features mit einem Median Wert von 4 beziehungsweise „trifft eher zu“ bewertet. Es besteht jedoch ein signifikanter Unterschied zu der äquivalenten Bewertung der Godot Engine. Diese wird mit einem Median Wert von 5 beziehungsweise „trifft zu“ bewertet. Dieser Unterschied besitzt einen starken Effekt. Da die Hypothese **H1** für die Variablen „einfache Lösung“ und „offensichtlicher Mehrwert“ abgelehnt werden muss, wird die komplette Hypothese abgelehnt. Somit ergibt diese Untersuchung, dass es keinen signifikanten Unterschied in der Bewertung der Aufgabenbereiche gibt.

Als Nächstes wird die Hypothese **H2** untersucht, die besagt, dass Aufgaben positiver bewertet wurden, je besser die Aufgabenstellung verstanden wurde. Da es sich bei den zu betrachtenden Variablen um ordinal skalierte Variablen handelt, kann die Rangkorrelation nach Spearman verwendet werden, um den Zusammenhang zwischen „leicht verständlich“ und den anderen evaluierten Aspekten zu untersuchen [BM03, S. 307-310].

**Tabelle 5.5:** H2: Rangkorrelation nach Spearman für Aufgabenteil A

Variable	$\rho$	p
Einfache Lösung	0,70	0,00
Lösung Wiederverwenden	0,41	0,01
Passender Aufwand	0,68	0,00
Offensichtlicher Mehrwert	0,35	0,03

**Tabelle 5.6:** H2: Rangkorrelation nach Spearman für Aufgabenteil B

Variable	$\rho$	p
Einfache Lösung	0,65	0,00
Lösung Wiederverwenden	0,16	0,38
Passender Aufwand	0,11	0,56
Offensichtlicher Mehrwert	0,17	0,35

Tabellen 5.5 und 5.6 zeigen die Ergebnisse der Rangkorrelation nach Spearman. Hier bei zeigt sich, dass in Teil A, in dem die implementierten Accessibility Features bearbeitet wurden, eine signifikante Korrelation zwischen allen untersuchten Variablen ( $p < 0,05$ ) und der Variable „leicht verständlich“ existiert. Die Effektstärke, die in der Tabelle als  $\rho$  angegeben ist, kann mit den Effektgrenzen von Cohen verglichen werden. Hierbei zeigt sich, dass es sich bei den Korrelationen zu „einfache Lösung“ ( $\rho = 0,70$ ) und „passender Aufwand“ ( $\rho = 0,68$ ) um einen starken Effekt handelt. Bei „Lösung wiederverwenden“ ( $\rho = 0,41$ ) und „offensichtlicher Mehrwert“ ( $\rho = 0,35$ ) besteht dahingegen nur ein mittlerer Effekt. Bei Teil B besteht dahingegen nur eine Korrelation zu der Variable „einfache Lösung“ ( $p < 0,05$ ). Hier existiert sogar starker Effekt ( $\rho = 0,65$ ). Zusammenfassend zeigt sich somit, dass ein hohes Verständnis der Aufgaben in Teil A mit einer positiveren Bewertung der anderen Variablen korreliert. Für diesen Aufgabenteil kann somit die Hypothese belegt werden. Bei Teil B dahingegen besteht nur eine Korrelation zu der Variable „einfache Lösung“. Für die anderen Variablen muss hier von der alternativen Hypothese ausgegangen werden, dass keine Korrelation besteht.

Es soll nun die Hypothese **H3**, dass Aufgaben positiver bewertet werden, wenn bereits ähnliche Aufgaben bearbeitet wurden, betrachtet werden. Hierfür werden die Antworten getrennt nach den Gruppen der Proband\*innen untersucht. Dafür werden wieder alle äquivalenten Aufgaben in Teil A und in Teil B zusammengefasst. Dazu werden jedoch diesmal nur die Antworten einer Gruppe betrachtet. Hierdurch ergibt sich eine Stichprobengröße von  $n_A = 15$  für Gruppe A und von  $n_B = 9$  für Gruppe B. Diese ist für eine weitere Auswertung mit dem Wilcoxon Test nicht ausreichend, da für diesen eine Stichprobengröße von  $n > 20$  vorhanden sein muss. Somit kann an dieser Stelle aufgrund der geringen Anzahl an Teilnahmen an der Umfrage keine Aussage über diese Hypothese getroffen werden.

Abschließend wird noch eine Untersuchung der Aufgaben vorgenommen, die nicht in den Hypothesen aufgenommen wurden. Hierbei handelt es sich um die Aufgaben (A2, A3, B4), die kein äquivalentes Gegenstück besitzen. Dadurch kann der Wilcoxon Test nicht genutzt werden, da es sich nicht um eine verbundene Stichprobe handelt. Da die Menge der verbliebenen Daten mit einer Größe von  $n = 16$  für Teil A und von  $n = 8$  für Teil B zu gering für eine statistisch relevante Auswertung ist, kann hier lediglich der Median aufgestellt werden, um einen Überblick zu gewähren. Dieser ist in Tabelle 5.7 dargestellt. Hierbei zeigt sich, dass die Aufgaben im Teil A ähnlich den anderen Aufgaben bewertet werden (siehe Tabelle 5.1). Die Variable „Lösung wiederverwenden“ wird dahingegen mit einem Wert von 3 beziehungsweise „teils teils“ negativer bewertet. Die Aufgabe B4 wird dahingegen in allen Variablen, außer im „offensichtlichen Mehrwert“ negativer bewertet.

**Tabelle 5.7:** Median der verbleibenden Antworten

<b>Variable</b>	<b>Median (A2, A3)</b>	<b>Median (B4)</b>
Leicht Verständlich	4,0	4,0
Einfache Lösung	5,0	4,0
Lösung Wiederverwenden	3,0	4,5
Passender Aufwand	4,0	4,0
Offensichtlicher Mehrwert	4,0	4,5

### Qualitative Auswertung

Am Ende der Umfrage wurde den Teilnehmenden die Möglichkeit gegeben, ihre Gedanken zu den Aufgaben, der Godot Engine und der Implementation in einer offenen Antwort zu äußern. Hierbei wurden vier Antworten erhalten. Diese sind im Anhang unter Abschnitt H in ihrer Vollständigkeit aufgeführt. Die Antworten wurden mit einem Buchstaben als Referenz versehen. Dieser dient lediglich dafür, dass in der nachfolgenden Evaluation einfach auf spezifische Antworten verwiesen werden kann.

Person A beschreibt, dass die Aufgaben grundsätzlich verständlich waren. Außer dem wird die Vermutung geäußert, dass es hilfreich gewesen wäre, wenn die Anwendungsziele der Aufgaben beschrieben worden wären. Dadurch wäre es leichter, die Wiederverwendbarkeit zu bewerten. Dieser Vorschlag findet sich ebenfalls bei den Proband\*innen B und C. Diese schreiben außerdem, dass die Relevanz einzelner Aufgaben für das Anwendungsgebiet der Accessibility nicht klar war. Hierfür werden die Aufgaben beispielhaft genannt, die die Anforderungen **REQ-52**, **REQ-53** und **REQ-55** behandeln. In diesen sollen in einer Textdatei und in einem Menü im Spiel Informationen hinterlegt werden, die Auskünfte über die inkludierten Accessibility Features geben. Außerdem soll die Möglichkeit gegeben werden, dass Nutzer\*innen Feedback geben. In der Aufgabenstellung ist jedoch nicht beschrieben, dass diese Informationen Nutzer\*innen die Möglichkeit geben sollen, sich im Spiel und vorab darüber zu informieren, ob es Accessibility Features gibt, die ihnen die Nutzung des Spiels ermöglicht oder vereinfacht. Zudem sollen sie die Möglichkeit besitzen, Feedback an die Entwickler\*innen zu schicken, damit die Features verbessert oder weitere eingebaut werden können. Dieses Unverständnis kann darin begründet sein, dass die Proband\*innen potenziell wenig eigene Verbindung zu Accessibility Features besitzen. In der Umfrage haben lediglich drei der acht Teilnehmer\*innen angegeben, dass sie Accessibility Features nutzen. Bei Proband\*innen B und C handelt es sich jedoch um zwei der Personen, die angegeben haben, selber Accessibility Features zu nutzen. Somit liegt die Vermutung nahe, dass das Ziel dieser Features nicht eindeutig genug zu erkennen war. Außerdem hat nur eine Person ihren Kenntnisstand in der Implementation dieser Features als fortgeschritten eingestuft. Somit ist die Vermu

tung naheliegend, dass gewisse Barrierefreiheitsmaßnahmen für die Teilnehmenden unbekannt sind. Als Lösung für diese Wissenslücke hätte es hilfreich sein können, den Nutzen der Aufgaben zu beschreiben und auf die relevanten Anforderungen zu verweisen.

Des Weiteren haben Proband\*innen B und C beschrieben, dass sie sich eine ausführlichere Beschreibung der Aufgaben gewünscht hätten, damit es für sie einfacher gewesen wäre, diese auszuführen. Person C schlägt dabei vor, dass Bilder genutzt werden könnten, um das Verständnis der Aufgaben zu fördern. Hier zeigt sich, dass der Versuch, die Aufgaben möglichst verständlich und detailliert zu formulieren, sinnvoll war. Trotzdem werden Diskrepanzen in der Verständlichkeit kritisiert. Diese wären in einem Pre Test aufgefallen und es wäre möglich gewesen, sie auszubessern. Im zeitlichen Rahmen dieser Arbeit war dies jedoch nicht möglich. Die unerwartet ausbleibende Teilnahme der Entwickler\*innen des Godot Wild Jams hatte dazu geführt, dass eine alternative Evaluationsstrategie verfolgt werden musste. Hierbei fiel die Entscheidung auf die simulierte Anwendung mit anschließender Befragung, an der Studierende teilgenommen haben, die an dieser Stelle ausgewertet wird. Durch den sehr geringen verbleibenden zeitlichen Rahmen zur Durchführung dieser Umfrage, wurde sich dafür entschieden, keinen Pre Test durchzuführen und stattdessen den Proband\*innen mehr Zeit zur Teilnahme zu gewähren. Die Diskrepanzen, die sich dadurch zeigen, sollten jedoch in folgenden Forschungen beachtet und verbessert werden. Eine weitere Ungenauigkeit, die erst während der Umfrage aufgefallen ist und durch Proband\*in B angesprochen wurde, ist, dass in der Aufgabenstellung teilweise Formulierungen zwischen deutsch und englisch wechseln. Solche Kritikpunkte sollte ebenfalls in zukünftigen Evaluationen durch Pre Tests gefunden und ausgebessert werden.

Ein weiterer Kritikpunkt, der von Person B geäußert wurde, ist, dass es ihr schwerfiel eine Einschätzung des Mehrwerts vorzunehmen, da kein Vergleich bekannt war. Da die Person in Gruppe A eingeteilt wurde, kann davon ausgegangen werden, dass sie zuerst die Aufgaben bearbeitet hat, die sich mit den implementierten Accessibility Features befassen. Dadurch stellt sich hier die Frage, ob es nötig ist, in der Aufgabenbeschreibung auch auf vergleichbare Lösungsansätze einzugehen oder ob es besser wäre, wenn die Teilnehmer\*innen zuerst den reinen Godot Teil bearbeiten und dann erst die Erweiterungen durch die Accessibility Features verwenden. Da in der schließenden Statistik aufgrund der geringen Antwortmenge keine Aussage über den Nutzen der Aufteilung der Proband\*innen in zwei Gruppen mit einer unterschiedlichen Bearbeitungsreihenfolge getroffen werden kann, fällt es schwer, den Nutzen dieser Entscheidung zu bewerten. Es wäre jedoch möglich, in einer folgenden Studie die Teilnehmer\*innen die Aufgaben in derselben Reihenfolge bearbeiten zu lassen

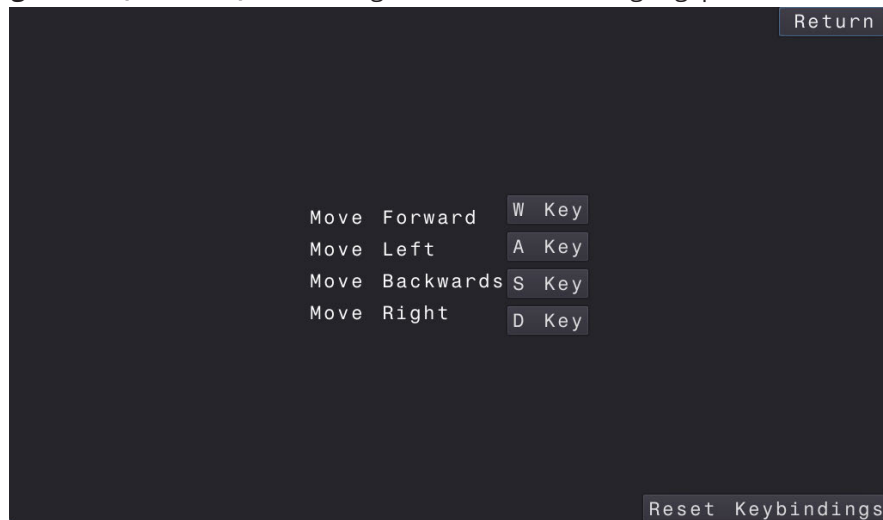
und mit den Aufgaben, die lediglich die Godot Game Engine nutzen, zu beginnen. Dadurch wäre eine Vergleichbarkeit der Lösungsansatz, die die implementierten Accessibility Features nutzen, vergleichbar. Für die reine Verwendung der Godot Engine gäbe es dann jedoch weiterhin keinen Vergleich. Hier stellt sich jedoch die Frage, ob dieser nötig ist und ob es effektiver wäre, diese Aufgaben lediglich als Vergleich für eine Bewertung der Accessibility Features zu verwenden. Der Wechsel auf eine Einschätzung, die immer im Vergleich zu einem Standard Lösungsansatz steht, könnte zudem einen besseren Überblick darüber liefern, wie der zusätzliche Aufwand der Verwendung der implementierten Features bewertet wird. In einer realen Anwendung kann davon ausgegangen werden, dass die Standard Lösungsansätze bereits verwendet werden. Dadurch bringt eine Evaluation dieser nur einen geringen Wissensgewinn, wenn nicht das Ziel ist, die Nutzbarkeit der Engine zu optimieren. Da in dieser Arbeit die Evaluation von Godot als Vergleichswert genutzt wird, ist es nötig, beide Aufgabenteile zu evaluieren, aber eine zukünftige Untersuchung könnte potenziell darüber optimiert werden, dass lediglich eine Bewertung in Relation zu dem Standard Lösungsansatz untersucht wird.

Das Feedback von Person D beinhaltet lediglich, dass ihr die Nutzung der Godot Engine schwerfiel. In dieser Antwort zeigt sich, wie auch schon in der Kritik, von Personen B und D, die den Wunsch nach verständlicheren Aufgaben geäußert haben, dass die Nutzung der Godot Engine ein Hindernis für sich darstellt. Diese Barriere wurde während der Erstellung des Fragebogens bedacht. Deswegen wurde sich dafür entschieden, die Aufgaben nicht lediglich als Ziele, die erreicht werden sollen, zu formulieren, sondern ebenfalls den Lösungsweg zu beschreiben. Hierdurch sollte es auch für Laien möglich sein, die Aufgaben zu bearbeiten und an der Umfrage teilzunehmen. In den Antworten hat sich gezeigt, dass dieser Ansatz zielführend war, aber noch ausgebaut werden kann.

## 5.2 Erfüllung des Anforderungskatalogs

In Kapitel 3.1 wurden die Anforderungen beschrieben, die durch die Game Accessibility Guidelines aufgestellt werden. Hierbei wurde bereits eine Einschränkung getroffen. In dieser wurden einzelne Anforderungen als nicht im Rahmen dieser Arbeit erfüllbar eingestuft. Bei diesen handelt es sich vor allem um Anforderungen, die sich mit dem Game Design oder der visuellen Darstellung befasst. Da das Ziel der Implementationen, die in dieser Arbeit getroffen werden, eine Vorlage für die Inklusion von Accessibility Features darstellen soll, sind solche Anforderungen zu spezifisch für das jeweilige Spiel, um umgesetzt werden zu können. Für die verbleibenden Anforderungen soll nun untersucht werden, ob diese erfüllt wurden. Zuerst wird die Anforderung **REQ-011** betrachtet. Diese verlangt die Möglichkeit, die Tastenbelegung zu konfigurieren.



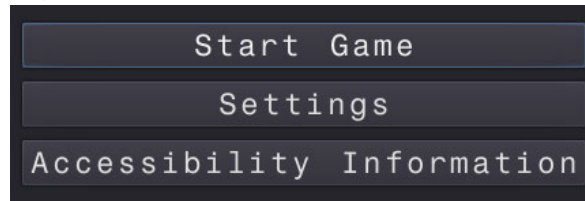
**Abbildung 5.4:** REQ-011, REQ-054: Konfigurierbare Tastenbelegung, persistente Einstellungen

rieren. In Bild 5.4 ist ein Menü zu sehen, das im Rahmen der Implementation erstellt wurde und Einstellungen zur Konfiguration der Tastenbelegung zeigt. Dieses kann von Entwickler\*innen um weitere Eingabeaktionen erweitert werden. Nutzer\*innen können mit diesem Menü die Steuerung frei nach ihren Wünschen anpassen. Somit kann diese Anforderung als erfüllt bewertet werden. Des Weiteren wird an dieser Stelle und später in den Anforderungen **REQ-013** und **REQ-042** die Anforderung **REQ-054** erfüllt, da die getroffenen Einstellungen persistent gespeichert werden und bei einem Neustart des Programms geladen werden.

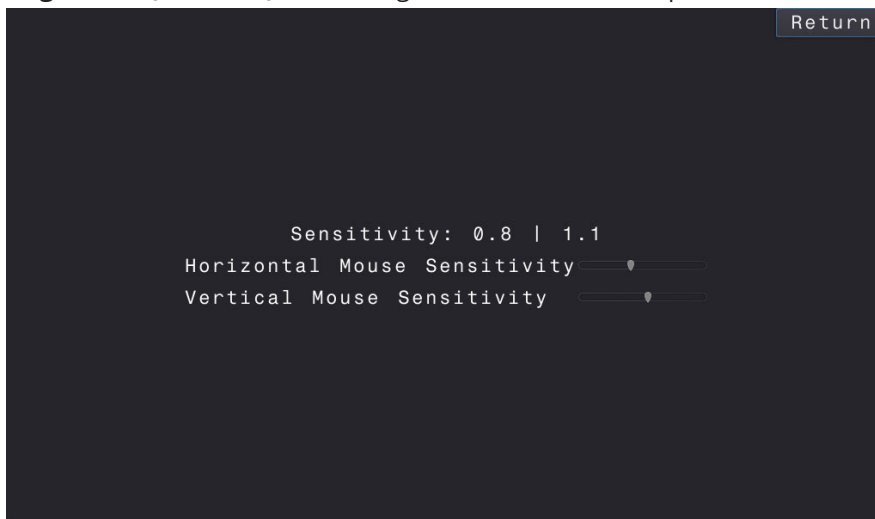
Als Nächstes wird Anforderung **REQ-012** betrachtet. Diese verlangt, dass Menüs auf dieselbe Art navigiert werden können, wie das Spiel. Da hier von einer Steuerung mit Maus und Tastatur ausgegangen wird, müssen lediglich diese beiden Eingabemöglichkeiten überprüft werden. Standardmäßig ist es in der Godot Engine möglich, UI Elemente per Maus zu bedienen. Damit eine Navigation per Tastatur ermöglicht wird, muss lediglich ein UI Element fokussiert werden. Dies wird durch eine Umrandung visualisiert. Diese ist in Grafik 5.5 zu sehen. Hier ist die Umrandung blau und markiert den Knopf mit der Aufschrift „Start Game“. Dadurch kann diese Anforderung als erfüllt bewertet werden. Des Weiteren zeigt diese Grafik die Konfiguration der Schriftart. Diese ist so gewählt, dass sie leicht lesbar ist aufgrund ihrer Schriftfamilie und ihrer Größe. Hierbei wurde sich an die Erkenntnisse aus Kapitel 3.3.3 gehalten. Dadurch werden die Anforderungen **REQ-022** und **REQ-034** erfüllt.

Die nächste Anforderung, die untersucht wird, ist **REQ-013**. Diese verlangt, dass die Sensitivität der Eingabegeräte konfiguriert werden kann. Da im Rahmen dieser Arbeit von einer Steuerung mit Maus und Tastatur ausgegangen wird, bedeutet dies, dass eine Möglichkeit benötigt wird, die horizontale und die vertikale Sensitivität der Maus einzustellen. Als beispielhafte Anwendung wurde hierfür ein Menü im

**Abbildung 5.5:** REQ-012, REQ-022, REQ-034: Userinterfacenavigation, leicht lesbare Schrift, leicht lesbare Schriftgröße



**Abbildung 5.6:** REQ-013, REQ-054: konfigurierbare Sensitivität, persistente Einstellungen



plementiert, dass in Grafik 5.6 zu sehen ist. In diesem können die jeweiligen Sensitivitäten konfiguriert werden und das Resultat wird in einer globalen Variable gespeichert. Des Weiteren wurde eine Anleitung hinterlegt, die erklärt, wie diese Variable zum Beispiel für die Kamerasteuerung in einem First Person Spiel genutzt werden kann. Die Anleitung kann im Anhang I und im Projektordner gefunden werden ([https://github.com/MilasBaumgarten/Godot\\_GameAccessibilityGuidelines\\_Addon/tree/main/addons/keybindings](https://github.com/MilasBaumgarten/Godot_GameAccessibilityGuidelines_Addon/tree/main/addons/keybindings)). Somit wird diese Anforderung als erfüllt angesehen.

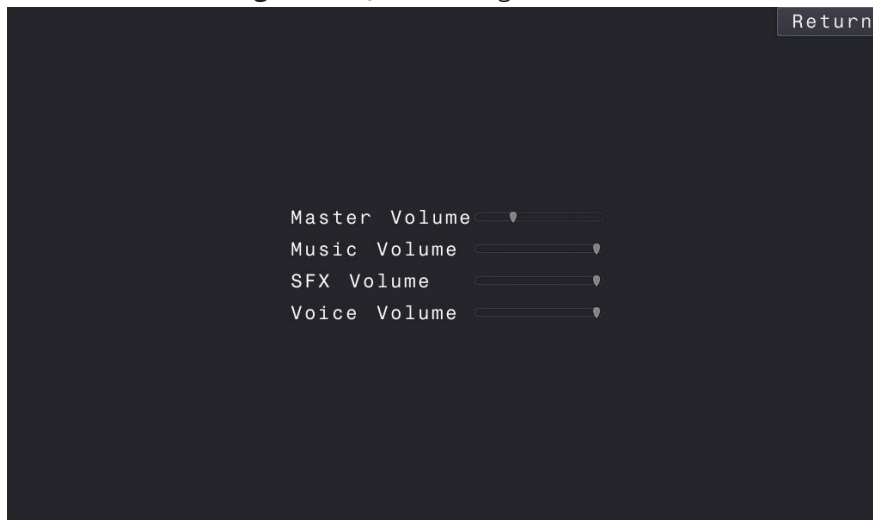
Die nächste Anforderung, die bearbeitet wurde, ist **REQ-037**. Diese verlangt, dass die Textgröße angepasst werden kann. Hier hat sich jedoch während der Implementation in Kapitel 4.4 gezeigt, dass es einen Bug in der Godot Engine gibt, durch den Schriften nach einer Anpassung ihrer Größe falsch angezeigt werden. Deswegen war es nicht möglich, diese Anforderung zu erfüllen. Dieser Fehler ist in der Godot Engine beginnend mit Version 4.0 gelöst. Da diese Version jedoch erst offiziell seit März 2023 veröffentlicht ist [Cond] und die Implementation in dieser Arbeit für eine Evaluation im Februar 2023 entwickelt wurde, war es nicht möglich, die neuere Version der Godot Engine zu verwenden.

**Abbildung 5.7:** REQ-041, REQ-044: grundlegende Untertitel, leicht lesbare Untertitel

Außerdem wurde die Anforderung **REQ-038** bearbeitet. Hierbei sollte es ermöglicht werden, dass Texte im Spiel per Screenreader ausgelesen werden können. Hierfür wurden verschiedene Ansätze ausprobiert, damit eine Sprachausgabe in das Programm eingebaut wird. Keiner war jedoch zufriedenstellend. Die alternativen Ansätze, die in Kapitel 3.3.5 vorgeschlagen wurden, konnten aufgrund des eingeschränkten zeitlichen Rahmens dieser Arbeit nicht evaluiert werden. Somit war es nicht möglich, diese Anforderung zu erfüllen.

Weitere Anforderungen, die bearbeitet wurden, sind **REQ-041** und **REQ-044**. Bei diesen beiden Anforderungen wird die Existenz von Untertiteln verlangt und dass diese leicht lesbar sind. Hierfür wurde ein einfaches Untertitelsystem implementiert, das eine Nachrichtenqueue beinhaltet, an die Texte übergeben werden können, die als Untertitel angezeigt werden sollen. Für Demonstrationszwecke wurde ebenfalls ein Skript programmiert, das mit dem Druck auf einen Button eine beliebige Nachricht an das Untertitelsystem übergibt. Die Verwendung hiervon ist in Bild 5.7 zu sehen. Des Weiteren wurden die Untertitel in Hinblick auf **REQ-022** und **REQ-034** so konfiguriert, dass sie leicht lesbar sind. Ebenfalls wurde ein Hintergrund erstellt, damit sich der Text von den anderen Bildelementen abhebt. Somit werden die beiden Anforderungen **REQ-041** und **REQ-044** als erfüllt angesehen.

Des Weiteren wurde die Anforderung **REQ-042** bearbeitet, die verlangt, dass die Lautstärke von Effekten, Musik und Sprache separat voneinander konfiguriert werden können. Hierfür wurde ein Menü implementiert, in dem mithilfe von Slidern die Lautstärke von mehreren Audiobussen angepasst werden kann. Dies ist in Bild 5.8 zu sehen. Die Audiobusse sollten dann während der Entwicklung des Spiels genutzt

**Abbildung 5.8:** REQ-042: konfigurierbare Lautstärke

werden, um die Tonausgaben zu konfigurieren. Da hierdurch alle Vorarbeiten gewährleistet sind, die im Rahmen dieser Arbeit für diese Anforderung implementiert werden können, gilt diese Anforderung ebenfalls als erfüllt.

Die letzten Anforderungen, die bearbeitet wurden, sind **REQ-052**, **REQ-053** und **REQ-055**. Diese erwarten, dass Nutzer\*innen außerhalb und innerhalb des Spiels Informationen über die implementierten Accessibility Features erhalten. Ebenfalls soll Spieler\*innen die Möglichkeit gegeben werden, Feedback an das Entwickler\*innenteam zu senden. Um diese Ziele zu erfüllen, wurde ein Menü implementiert, in dem alle relevanten Informationen in einer Liste aufgeführt werden. Außerdem wird die E-Mail-Adresse des Autors als Kontaktmöglichkeit hinterlegt. Analog dazu wurden in einer Textdatei im Hauptordner des Projektes dieselben Informationen beschrieben. Da das Projekt über GitHub zur Verfügung gestellt wird, werden diese Informationen sofort beim Aufrufen der Projektseite angezeigt. Sobald ein Spiel mit der Implementation dieser Arbeit veröffentlicht wird, sollten die Informationen auf der Seite der Veröffentlichung angezeigt werden. Im gegebenen Rahmen wird dieser Lösungsansatz jedoch als ausreichend für die Erfüllung der Anforderung verstanden. Somit werden die Anforderungen **REQ-052**, **REQ-053** und **REQ-055** als erfüllt angesehen.

**Abbildung 5.9:** REQ-052. REQ-055: Accessibility Informationen außerhalb des Spiels, Feedbackmöglichkeit

```
# Godot Game Accessibility Guidelines
This project can be used as a starting point for a new Godot project.
It includes basic systems to use accessibility features as described by the [Game accessibility
guidelines](https://gameaccessibilityguidelines.com/).

The features are mostly implemented in addons. Also please read the Readme files inside of the
specific addons. They will explain specific things about the features and how to use them.

## Implemented Accessibility Features:
- the controls can be remapped
- the UI can be controlled via mouse or keyboard
- the sensitivity of controls can be adjusted
- the default font is easily readable
- the default font size is chosen to be easily readable
- separate volume controls are provided for effects, speech, background and music
- information about the implemented accessibility features is provided to the user

## Notes
- for different UI elements (e.g. ingame UI), the "ui_theme" might have to be reapplied
  - otherwise the separation of elements might be too small

## Other
Feedback: milasbaumgarten@gmail.com
```

**Abbildung 5.10:** REQ-053. REQ-055: Accessibility Informationen innerhalb des Spiels, Feedbackmöglichkeit

```
Implemented Accessibility Features: Return
- the controls can be remapped
- the UI can be controlled via mouse or
  keyboard
- the sensitivity of controls can be adjusted
- the default font is easily readable
- the default font size is chosen to be easily
  readable
- separate volume controls are provided for
  effects, speech and music
- information about the implemented
  accessibility features is provided to the user
- information about the implemented
  accessibility features is provided to the user
  before starting the game
- information about the implemented
  accessibility features is provided to the user
  before starting the game
- basic subtitles exist

Feedback: milasbaumgarten@gmail.com
```

## 6 Zusammenfassung und Ausblick

Nachdem im letzten Kapitel die Resultate der Evaluationen vorgestellt wurden, werden diese in diesem Kapitel zusammengefasst und anschließend reflektiert. Außer dem werden Anstöße für folgende Forschungen dargelegt und Empfehlungen für weitere Umfragen vorgestellt. Des Weiteren wird die Implementation der Accessibility Features, die in dieser Arbeit vorgenommen wurde, reflektiert. Hierbei wird untersucht, inwieweit die Entscheidungen für spezifische Implementationskonzepte als erfolgreich angesehen werden können und wo es Verbesserungsbedarf gibt. Des Weiteren wird untersucht, inwieweit die Forschungsfrage beantwortet werden kann.

### 6.1 Ergebnisse

In dieser Arbeit wurde eine Vorlage entwickelt, die von Entwickler\*innen innerhalb der Godot Engine genutzt werden kann, um einen Großteil der Anforderungen der Features mit geringer Komplexität der Game Accessibility Guidelines [EII+] zu erfüllen. Der erstellte Code wurde durch Studierende im Rahmen einer Umfrage evaluiert. In Kapitel 5.1.2 wurden die Antworten ausgewertet und drei Hypothesen untersucht. Bei diesen handelt es sich um folgende: **H1** Die Bewertung der Aufgaben in Teil A unterscheidet sich signifikant von denen von Teil B. **H2** Je besser die Aufgabenstellung verstanden wurde, desto positiver wurden die anderen Bereiche der Aufgabe bewertet. **H3** Aufgaben werden positiver bewertet, wenn bereits äquivalente Aufgaben bearbeitet wurden. Hierbei wurde für die erhaltenen Antworten bewiesen, dass die Hypothese **H1** widerlegt ist und es keinen signifikanten Unterschied in der Bewertung zwischen Teil A und Teil B gibt. Somit kann davon ausgegangen werden, dass der zusätzliche Aufwand, den die Nutzung der Accessibility Features mit sich bringt, nicht signifikant ist.

Mit der Hypothese **H2** wurde daraufhin untersucht, wie hoch der Zusammenhang zwischen dem Verständnis der von den Proband\*innen bearbeiteten Aufgabenstellungen und ihrer Bewertung der anderen Variablen ist. Hierbei wurde bewiesen, dass es einen signifikanten Zusammenhang gibt, für die Aufgaben, die die Nutzung der Accessibility Features beinhalten. Dieser besitzt einen mittleren bis starken Einfluss. Daraus kann in Zusammenhang zu den offenen Antworten, die ebenfalls in Kapitel 5.1.2 ausgewertet wurden, vermutet werden, dass das Verständnis der Aufgaben ein signifikantes Problem für die Teilnehmenden dargestellt hat. Somit stellte sich die Frage, wie stark sich eine Überarbeitung der Aufgabenstellung auf die Bewertungen der Proband\*innen auswirken würde und ob dann immer noch ein signifikanter

Unterschied zwischen der Bewertung der unterschiedlichen Aufgabenbereiche zu erkennen wäre. Hierfür ist es nötig, weitere Untersuchungen durchzuführen, die im Rahmen dieser Arbeit nicht aufgenommen werden konnten.

Des Weiteren wurde in Hypothese **H3** untersucht, wie signifikant die Entscheidung war, die Teilnehmenden in zwei Gruppen aufzuteilen und die Aufgaben in unterschiedlichen Reihenfolgen bearbeiten zu lassen. Aufgrund der geringen Teilnahme an der Umfrage lässt sich jedoch für diese Hypothese keine Aussage treffen. Da hingegen geben die offenen Antworten einiger Proband\*innen einen Rückschluss darüber, dass die Vergleichbarkeit der Lösungsansätze schwer war. Für zukünftige Evaluationen könnte es sich deswegen anbieten, die Nutzung der reinen Godot Engine als Standard zu verstehen und die Teilnehmer\*innen nur den Vergleich zu dem Mehraufwand der Nutzung der Accessibility Features bewerten zu lassen. Alternativ kann eine Beschränkung auf Personen vorgenommen werden, die genug Erfahrung im Umgang mit der Godot Engine besitzen, damit die Aufgaben, die nur die Engine nutzen, trivial beziehungsweise nicht nötig sind.

Ein Großteil der Teilnehmer\*innen der Umfrage haben angegeben, dass sie Erfahrungen in der Implementation von Accessibility Features besitzen (75%). Dahingegen hat nur ein kleiner Teil angegeben, Accessibility Features zu nutzen (37,5%). In Hinblick darauf, welche Aspekte laut der Game Accessibility Guidelines [EII+] zur Barrierefreiheit für Spiele gehören, liegt die Vermutung nahe, dass die reale Zahl der Nutzungen weit höher ist. Offensichtliche Hilfsmittel, wie Brillen, Untertitel und als solche beschriftete Accessibility Features sind nur ein kleiner Teil, der Möglichkeiten. Weitere Aspekte, die potenziell als solche nicht im Bewusstsein der Proband\*innen waren, sind zum Beispiel Schwierigkeitseinstellungen, die Möglichkeit, die Tastenbelegung und Maussensitivität anzupassen und die simultane Darstellung von Informationen über mehrere Sinne [EII+]. Um diese Hypothese zu untersuchen, könnten Proband\*innen mit einer Liste an verschiedenen Aspekten von Videospiele konfrontiert werden und eine Einschätzung darüber treffen, welche davon Accessibility Features sind und welche nicht.

Laut dem Statistischen Bundesamt leben circa 7,8 Millionen Menschen mit Behinderungen in Deutschland [Bun]. Ebenfalls wird in dieser Statistik aufgezeigt, dass der Anteil an Menschen mit Behinderungen mit zunehmendem Alter steigt. Dies unterstützt die Aussage von Kent, dass alle Menschen den Zustand, dass sie nicht mit einer Behinderung leben, nur als temporär verstehen sollten [Ken15]. Somit ist es wichtig, dass das Bewusstsein über Arten von Accessibility Features erhöht wird und dass diese bei der Entwicklung von Spielen mehr fokussiert werden.

## 6.2 Reflexion

Nachdem nun die Ergebnisse der Evaluation vorgestellt wurden, soll die Durchführung der Umfragen kritisch reflektiert werden. Dabei werden ebenfalls Vorschläge für Verbesserungen und Alternativen dargelegt. Außerdem werden die Implementierungen, die in dieser Arbeit durchgeführt wurden, kritisch reflektiert und die Erkenntnisse, die während der Arbeit an ihnen gesammelt wurden, werden dargelegt.

### 6.2.1 Evaluation

Im Rahmen dieser Arbeit wurden zwei Umfragen durchgeführt. Die erste Umfrage war an Teilnehmer\*innen des Godot Wild Jams gerichtet. Bei dieser wurde ein hoher Erkenntnisgewinn erhofft, da die Entwickler\*innen bereits Erfahrung im Umgang mit der Godot Engine besitzen und im Rahmen der Game Jam mit Accessibility in Kontakt gekommen sind. Hier gab es jedoch keine einzige Antwort auf die Umfrage. Hierfür kann es verschiedene Gründe geben, auf die in Kapitel 5.1.1 eingegangen wurde. Hier könnte es hilfreich sein, einen erneuten Versuch zu starten und eine weitere Evaluation mit den Entwickler\*innen durchzuführen. Dies ist jedoch im Rahmen dieser Arbeit zeitlich nicht möglich. Für eine Verbesserung der Teilnahme wird empfohlen, einen engeren Kontakt zu den Teilnehmenden der Game Jam aufzubauen und Informationen darüber zu sammeln, welche Personen die implementierten Accessibility Features nutzen, damit diese nach Ende der Game Jam befragt werden können. Hierdurch ist es ebenfalls möglich, vor der Durchführung der Umfrage einen Überblick darüber zu bekommen, wie groß die Nutzung ist. Ebenfalls wird empfohlen, das Wissen über die Existenz der Implementation bei den Teilnehmenden zu erhöhen. Hierfür könnten die Organisator\*innen der Game Jam kontaktiert werden. Des Weiteren hat sich in der Evaluation der Studierenden gezeigt, dass es hilfreich ist, die Proband\*innen direkt zu kontaktieren, anstatt einen Aufruf an alle Personen, die sich dafür interessieren könnten, zu verfassen. Eine persönliche Nachricht stellt im Gegensatz zu einem öffentlichen Aufruf einen Zwang für die Zielperson dar, eine Antwort zu verfassen beziehungsweise in diesem Fall an der Umfrage teilzunehmen.

Des Weiteren ist während der Durchführung der Umfrage mit den Studierenden aufgefallen, dass vor allem eine größere Anzahl an Teilnehmenden nötig ist, um zu verlässigere Aussagen treffen zu können. Obwohl bei dieser Untersuchung bereits ein Teil der Proband\*innen persönlich kontaktiert und dazu aufgefordert wurden, an der Studie teilzunehmen, war dies nicht ausreichend. Hier stellt sich jetzt die Frage, welche weiteren Maßnahmen getroffen werden könnten, um eine höhere Antwort rate zu erreichen. Die Vorschläge von Umbach [Umb04, S. 28 35], sowie Liao und Hsieh [LH17, S. 82f], Erinnerungen an die Teilnehmenden zu schicken und die Nachrichten zu personalisieren, wurden bereits inkludiert. Eine Möglichkeit, die bleibt, ist



die Anzahl an personalisierten Aufrufen zur Teilnahme an der Umfrage zu erhöhen. Sechs Personen wurden direkt kontaktiert, mit der Bitte, an der Evaluation teilzunehmen. Davon haben lediglich zwei bestätigt, dass sie die Umfrage ausgefüllt haben. Die anderen sechs Proband\*innen haben sich durch Aufrufe in verschiedenen Kommunikationsgruppen gefunden. Hierbei wurden rund 500 Personen kontaktiert, bei denen jedoch von einem Großteil unbekannt ist, ob sie den Aufruf gelesen haben. Da auf beiden Wegen nur eine sehr geringe Antwortrate erreicht wurde, stellt sich hier für den Autor die Frage, ob es zielführender wäre, andere Möglichkeiten der Evaluation zu nutzen. Eine Möglichkeit wären Expert\*inneninterviews. Diese könnten zum Beispiel mit Teilnehmenden des Godot Wild Jams durchgeführt werden. Hierdurch würde auch eine weit geringere Anzahl an Proband\*innen ausreichend sein. Diese Möglichkeit wurde ebenfalls in dieser Arbeit bedacht. Da jedoch nur eine Person ausfindig gemacht wurde, die nachweisbar die implementierten Accessibility Features im Rahmen des Game Jams genutzt hat, sowie eine weitere Person, die aus persönlichem Interesse den Code genutzt hat, schien es nicht möglich zu sein, eine theoretische Sättigung zu erreichen. Deswegen wurde sich dafür entschieden, diesen Ansatz nicht zu nutzen.

Ebenfalls hat sich während der Auswertung der Antworten der Umfrage gezeigt, dass die geringe Auswahl der soziodemografischen Fragen für die Privatsphäre der Proband\*innen vorteilhaft war, da es keine Möglichkeit gab, einzelne Teilnehmer\*innen zu identifizieren. Es hat sich aber auch gezeigt, dass weitere Fragen nötig sind, damit die Auswahl der Testgruppe reproduzierbar ist. Vor allem Fragen bezüglich des Alters und Geschlechts, sowie des Bildungsgrades, sind nötig, damit folgende Studien die Testgruppe reproduzieren können oder Vergleiche zu dieser aufstellen können. Außerdem war es nicht möglich, alle implementierten Accessibility Features zu evaluieren, da es nicht möglich war, für alle Aufgaben eine vergleichbare Aufgabenstellung zu formulieren, die einen Vergleich ermöglicht hätte. Ein Vergleich der Mediane, der übrig gebliebenen Aufgaben, mit den Medianen der untersuchten Aufgaben ermöglicht jedoch Vermutungen aufzustellen. So zeigt sich, dass die Aufgaben im Teil A bis auf den Wunsch, den Lösungsansatz in Zukunft erneut zu verwenden, circa gleichbleibend bewertet wurden (siehe Kapitel 5.1.2). Dahingegen ist die Bewertung der verbliebenen Aufgabe in Teil B negativer ausgefallen (siehe Kapitel 5.1.2). Dadurch kann die Vermutung aufgestellt werden, dass die Aufgaben, die sich mit der Menüführung per Tastatur und Untertiteln befassen, zu vergleichbaren Resultaten führen würde, wie die anderen Aufgaben, die sich mit den Accessibility Features beschäftigen. Für die Aufgabe in Teil B, die sich mit der Konfiguration einer Standardschriftart befasst, fällt es schwerer Rückschlüsse zu treffen. Es kann sein, dass die Aufgabenstellung für die Proband\*innen schwerer zu verstehen war, als die anderen Aufgaben und dadurch analog zu den Erkenntnissen der Hypothese H2, die Bewertung der wei

teren Aspekte negativer stattgefunden hat. Ebenfalls ist es aber auch möglich, dass Nutzer\*innen den Lösungsweg, den die Godot Engine vorschreibt, weniger positiv bewerten, da er im Vergleich zu bekannten anderen Programmen, wie Microsoft Word oder der Unity Game Engine komplex und aufwändig erscheint. Für eine genauere Erkenntnis ist es nötig, eine weitere Untersuchung durchzuführen. Hierbei sollte darauf geachtet werden, dass auf die Existenz von äquivalenten Aufgaben geachtet wird, falls die Auswertung in Form eines Vergleiches stattfindet.

### 6.2.2 Implementation

Während der Implementation der Lösungsansätze der Anforderungen hat sich vor allem gezeigt, dass in der Godot Engine bereits viel Vorarbeit geleistet wurde, damit die Accessibility Features einfach eingebaut werden können. Ebenfalls hat sich jedoch gezeigt, dass es hierbei ein paar Ausnahmen gibt. Diese sind die Screenreader Kompatibilität und Untertitel. Im Rahmen dieser Arbeit wurden verschiedene Bibliotheken und Ansätze ausprobiert, um die Ausgabe eines Spiels, das mit der Godot Engine erstellt wurde, von einem Screenreader auslesen zu können. Schlussendlich hat es sich jedoch herausgestellt, dass die gewählten Ansätze nicht zu einem zufriedenstellenden Ergebnis führen. So hat es sich gezeigt, dass der Einbau von Sprachsynthese über externe Bibliotheken möglich ist, aber einigen Mehraufwand an die Entwickler\*innen mit sich bringt. Der Grund hierfür ist, dass die eigene Skriptingsprache der Engine nicht verwendet werden kann, sondern andere Programmiersprachen genutzt werden müssen. Hierfür gibt es Ansätze, wie zum Beispiel die Nutzung von Plugins oder Erweiterungen des Quellcodes. Diese benötigen jedoch einen signifikanten Aufwand von Nutzer\*innen, bevor die eigentliche Entwicklung von Spielen und Programmen begonnen werden kann. Aus diesem Grund wurde sich während der explorativen Entwicklung dafür entschieden, diese Ansätze als nicht zielführend zu bewerten. Wenn die Screenreader Kompatibilität in folgenden Arbeiten erneut angestrebt wird, wird empfohlen, einen anderen Ansatz zu verfolgen. Hierbei wäre es zum einen möglich, eine Erweiterung der Engine zu schreiben und dabei anzustreben, dass diese in die offizielle Version inkludiert wird. Aufgrund des stark eingeschränkten zeitlichen Rahmens dieser Arbeit war dieses Ziel nicht möglich. Alternativ kann versucht werden, ein separates Programm als Schnittstelle zwischen Spiel und Screenreader zu implementieren. Dieses würde dann Ausgaben des Spiels darstellen und Eingaben der Spieler\*innen weitergeben. Für Projekte, deren Budget groß genug ist, um eine manuell erstellte Audiobeschreibung zu erstellen, kann diese ebenfalls verwendet werden, um Accessibility für Menschen mit eingeschränkter Sicht zu gewährleisten. Im Sinne der Accessibility wäre es jedoch, wenn eine Möglich

keit gefunden wird, mit der Programme, die mit der Godot Engine erstellt werden, ohne weitere Arbeit Sprachsynthese beinhalten oder von Screenreadern gelesen werden können.

Die andere Implementation, die komplexer als erwartet war, sind Untertitel. Hier bei hat sich schnell gezeigt, dass selbst ein sehr simples System zur Darstellung dieser bereits eine nicht triviale Systemarchitektur benötigt. Der Grund hierfür liegt daran, dass das System zum einen von verschiedenen Stellen im Code angesprochen werden muss. Zum anderen daran, dass die Darstellung der Untertitel zeitlich entkoppelt sein sollte von der Übermittlung der Untertitelnachricht an das System. Wenn sich darauf beschränkt wird, dass Untertitel nur in Zwischensequenzen dargestellt werden, in denen es keine Nutzer\*inneninteraktion gibt, könnte dies ohne ein spezifisches System funktionieren. Hier würde es ausreichen, wenn die darzustellenden Texte als Teil der Animation eingebaut wird oder, wenn direkt eine Videodatei abgespielt wird, in der Untertitel angezeigt werden. Dies ist jedoch eine starke Einschränkung, die nicht sonderlich wiederverwendbar ist und nur in sehr spezifischen Situationen funktioniert. Deswegen wurde sich dafür entschieden, das komplexere System zu bevorzugen und zu implementieren. Damit die bestehende Lösung für Closed Captions oder sekundäre Untertitel genutzt werden kann, ist es nötig, die Architektur zu erweitern. Es ist vor allem nötig, dass eine Priorisierung der Nachrichten getätigt werden kann. Dabei müssen die wichtigsten Textbeschreibungen immer zuerst angezeigt werden und nebensächliche Informationen können nur dargestellt werden, wenn die Zeit dafür besteht. Dadurch müssen auch Nachrichten, die in der Eventqueue sind, bei Bedarf nach einer bestimmten Zeit gelöscht werden, da veraltete Untertitel ab einem bestimmten zeitlichen Rahmen unsinnig sind. Hierbei muss ebenfalls bedacht werden, dass Closed Captions nicht unbedingt immer optionale Informationen darstellen, sondern dass hierbei vor allem relevante Informationen, die primär auditiv dargestellt werden, vermittelt werden sollen.

Während der Implementation hat sich bei einer Anforderung gezeigt, dass diese aktuell nicht zufriedenstellend erfüllt werden kann. Die variable Einstellung der Textgröße unterliegt aktuell einem Fehler in der Godot Engine, wodurch die Darstellung von Texten fehlerhaft ist. Zudem besitzt die gewählte Implementation ein paar Einschränkungen. Die erste ist, dass nicht überprüft wird, ob die Schriftart, auf die der Pfad verweist, wirklich verwendet wird. So kann es passieren, dass die Größe einer Schriftart geändert wird, dies aber keinen Einfluss auf die Darstellung im Programm hat. Die andere Einschränkung ergibt sich daraus, wie Änderungen an den Darstellungen von einzelnen UI Elementen in Godot vorgenommen werden. Wenn ein Element sich von der Standardkonfiguration unterscheiden soll, dann muss diese für das Element überschrieben werden. In dem Fall, dass die Schriftgröße geändert wird, hat dies

zur Folge, dass Änderungen der Standardschriftart keinen Effekt auf dieses Objekt hat. Hier müsste ein Ansatz verwendet werden, bei dem Anpassungen des Standards weiterhin beachtet werden. Eine Möglichkeit hierfür könnte es sein, dass Elemente mit Texten eine spezielle Variable zur Skalierung benutzen, die erst beim Start des Programms die Größe verändert. Hierbei muss untersucht werden, inwieweit dies eine Lösung darstellt, oder ob das Problem weiterhin besteht. Dieser Ansatz bringt jedoch potenziell ein neues Hindernis mit sich. In Godot kann ein Objekt nur ein Skript besitzen. Somit bringt die Entscheidung, ein kleines Hilfsskript an viele Objekte anzuhängen, die Gefahr mit sich, eine große Einschränkung zu sein. Es ist möglich von Entwickler\*innen zu verlangen, Vererbung zu verwenden, um dieses Problem zu umgehen, aber diese Lösung bringt eine hohe Fehleranfälligkeit mit sich. Hier scheint eine Anpassung der Engine potenziell einen besseren Ansatz darzustellen. Durch eine Änderung des bestehenden Systems würde dieses jedoch sehr wahrscheinlich inkompatibel mit bestehenden Projekten sein. Außerdem muss abgewägt werden, inwieweit es sinnvoll ist, ein System grundlegend zu verändern, bei dem sich Entwickler\*innen an einen bestimmten Workflow gewöhnt haben.

### 6.3 Ausblick

Nachdem nun die Ergebnisse der Arbeit untersucht wurden, stellt sich nun die Frage, ob die Forschungsfragen dieser Arbeit beantwortet werden können. Die primäre Forschungsfrage untersucht, ob die Einordnung der Komplexität, die von den Game Accessibility Guidelines [EII+] getroffen wird, in dieser Arbeit bestätigt wird. Außerdem stellt sich die Frage, wie hoch der Aufwand für die Entwickler\*innen bei der Verwendung des implementierten Softwarepakets ist. Es lässt sich nicht mit einem klaren „Ja“ oder „Nein“ beantworten, ob der geschätzte Aufwand der Game Accessibility Guidelines [EII+] bestätigt werden konnte. Stattdessen müssen die einzelnen Anforderungen betrachtet werden. In dieser Arbeit hat sich gezeigt, dass die meisten Anforderungen mit sehr geringem Aufwand umgesetzt werden können. Dies liegt darin begründet, dass die nötigen Softwaresysteme und Voraussetzungen bereits in der Godot Engine inkludiert sind. Lediglich die Anforderung der Screenreader (Screenreader KompatibilitätREQ 038) und die Anforderung nach Untertiteln (**REQ-041**) waren eindeutig komplexer. Da es sich bei der Kompatibilität für Screenreader um eine mittlere bis komplexe Anforderung handelt, war dies zu erwarten. Da die Implementation für Geräte, die nicht Smartphones sind, bereits von den Game Accessibility Guidelines [EII+] als komplex bewertet wurde, kann diese Einordnung in dieser Arbeit bestätigt werden. Die Art, mit der Spiele heutzutage dargestellt werden, stellt eine Barriere für die Nutzung von Screenreadern dar und kann nur mit signifikantem Aufwand umgangen werden.

Die Einschätzung, dass ein grundlegendes Untertitelsystem eine geringe Komplexität besitzt [EII+], kann nicht bestätigt werden. Das System muss von unterschiedlichen Stellen aus angesprochen werden können und zeitlich entkoppelt sein. Die daraus resultierende Softwarearchitektur entspricht nicht der Komplexität, die sich in den anderen Anforderungen gezeigt hat. Die Implementation war komplexer, als der Großteil der anderen Anforderungen. Gleichzeitig jedoch simpler, als die Umsetzung der Screenreader Kompatibilität. Dadurch wird diese Anforderung im Rahmen dieser Arbeit einer mittleren Komplexität zugeordnet. Dies entspricht nicht der Einordnung der Game Accessibility Guidelines [EII+]. Somit kann die Forschungsfrage für alle untersuchten Anforderungen, bis auf die Implementation von grundlegenden Untertiteln zustimmend beantwortet werden.

Die sekundäre Forschungsfrage untersucht den Aufwand, den Entwickler\*innen bei der Nutzung der implementierten Accessibility Features haben. Hierfür wurde eine Evaluation mit Studierenden durchgeführt, die mit mehreren Aufgaben die Anwendung des implementierten Softwarepakets simulieren sollten. Dabei hat sich gezeigt, dass es für die untersuchte Stichprobe keinen signifikanten Unterschied in der Bewertung der Aufgaben gab, der davon abhängen würde, ob die Accessibility Features genutzt wurden oder nicht. Stattdessen hat sich gezeigt, dass es eine starke Korrelation zwischen dem Verständnis der Aufgaben und ihrer Bewertung gibt. Somit kann die Forschungsfrage damit beantwortet werden, dass kein signifikanter Mehraufwand für die Entwickler\*innen bei der Nutzung der Implementationen dieser Arbeit entsteht. Dahingegen hat es sich jedoch als wichtig herausgestellt, eine sehr gute Dokumentation zu erstellen, da eine unverständliche Beschreibung der Nutzung der Systeme, eine Barriere für Entwickler\*innen darstellt.

In dieser Arbeit konnte gezeigt werden, dass ein signifikanter Teil der untersuchten Anforderungen der Game Accessibility Guidelines [EII+] in der korrekten Komplexität eingeordnet wurden. Dieses Ergebnis war zu erwarten, da es sich bei diesen Guidelines um ein Projekt handelt, das von mehreren Studios, Expert\*innen und Forscher\*innen entwickelt wurde [EII+]. Da jedoch keine Quellen für die Entscheidungen der Komplexitätsbewertungen angegeben wurden, leistet diese Arbeit einen Beitrag zum aktuellen Forschungsstand. Der Grund hierfür ist, dass diese Arbeit zum einen Vorschläge bietet, nach denen Accessibility Features implementiert werden können und zum anderen, dass diese Arbeit die Einordnung einiger Features untersucht und bestätigt hat. Nachfolgende Forschungen sollten die Ergebnisse dieser Arbeit überprüfen und weiterführen, vor allem, in Hinblick auf die Evaluation des Mehraufwands für Entwickler\*innen. Des Weiteren wurde in dieser Arbeit die Vermutung aufgestellt, dass das Wissen über Accessibility Features in der Stichprobe an Studierenden gering ist. Hier können weitere Untersuchungen durchgeführt werden, um herauszufinden,

ob es nötig ist, das Wissen über Accessibility Features zu verbreiten oder ob es bereits bei, vor allem, Entwickler\*innen ein Bewusstsein über die Existenz spezifische Accessibility Features gibt. Außerdem kann analog zu dieser Arbeit eine Untersuchung von Implementationskonzepten für die weiteren Accessibility Features durchgeführt werden. Hierbei sind vor allem komplexere Features von großem Interesse, da diese mit hoher Wahrscheinlichkeit auf viele unterschiedliche Arten implementiert werden können. Dadurch bietet es sich an, eine Gegenüberstellung der verschiedenen Ansätze aufzustellen und miteinander zu vergleichen.

## Literaturverzeichnis

- [Fri15] Valerie Frissen, Hrsg. *Playful identities: the ludification of digital media cultures*. MediaMatters. OCLC: 881214749. Amsterdam: Amsterdam University Press, 2015. ISBN: 978 90 8964 639 2 978 90 485 2303 0.
- [Agu+20] Juan Aguado Delgado u. a. „Accessibility in video games: a systematic review“. en. In: *Universal Access in the Information Society* 19.1 (März 2020), S. 169 193. ISSN: 1615 5297. DOI: 10 . 1007 / s10209 018 0628 2. URL: [https://doi.org/10.1007/s10209\\_018\\_0628\\_2](https://doi.org/10.1007/s10209_018_0628_2) (besucht am 04.04.2023).
- [BMM11] Jayeeta Banerjee, Deepti Majumdar und Dhurjati Majumdar. „Readability, Subjective Preference and Mental Workload Studies on Young Indian Adults for Selection of Optimum Font Type and Size during Onscreen Reading“. In: *Al Ameen Journal of Medical Sciences* 4 (Jan. 2011).
- [Ken15] Mike Kent. „Disability and eLearning: Opportunities and Barriers“. In: *Disability Studies Quarterly* 35.1 (Feb. 2015). ISSN: 2159 8371, 1041 5718. DOI: 10 . 18061 / dsq . v35i1 . 3815. URL: [http://dsq\\_sds.org/article/view/3815](http://dsq_sds.org/article/view/3815) (besucht am 17. 11. 2022).
- [VL10] María Rosalía Vicente und Ana Jesús López. „A Multidimensional Analysis of the Disability Digital Divide: Some Evidence for Internet Use“. en. In: *The Information Society* 26.1 (Jan. 2010), S. 48 64. ISSN: 0197 2243, 1087 6537. DOI: 10 . 1080 / 01615440903423245. URL: <http://www.tandfonline.com/doi/abs/10.1080/01615440903423245> (besucht am 17. 11. 2022).
- [KHM13] Jan Louis Kruger, Esté Hefer und Gordon Matthew. „Measuring the impact of subtitles on cognitive load: eye tracking and dynamic audiovisual texts“. In: *Proceedings of the 2013 Conference on Eye Tracking South Africa*. ETSA '13. New York, NY, USA: Association for Computing Machinery, Aug. 2013, S. 62 66. ISBN: 978 1 4503 2110 5. DOI: 10 . 1145 / 2509315 . 2509331. URL: <https://doi.org/10.1145/2509315.2509331> (besucht am 18.04.2023).
- [Cona] World Wide Web Consortium. *How to Meet WCAG (Quickref Reference)*. URL: <https://www.w3.org/WAI/WCAG21/quickref/> (besucht am 11.11.2022).
- [BBC22] BBC. *BBC Subtitle Guidelines*. en. Version 1.2.1. Juli 2022. URL: <https://www.bbc.co.uk/accessibility/forproducts/guides/subtitles/subtitles/> (besucht am 17. 11. 2022).

- [Inc22] The AbleGamers Foundation Inc. *Accessible Player Experiences (APX)*. Accessible Games. 2022. URL: <https://accessible.games/accessible-player-experiences/> (besucht am 10.10.2022).
- [Ell+] Barrie Ellis u. a. *Game accessibility guidelines | A straightforward reference for inclusive game design*. URL: <https://gameaccessibilityguidelines.com/> (besucht am 10.10.2022).
- [Deu23] Bundesrepublik Deutschland. *BGG nichtamtliches Inhaltsverzeichnis*. 2023. URL: <https://www.gesetze-im-internet.de/bgg/> (besucht am 19.04.2023).
- [Men13] Netzwerk Menschenrechte. *Barrierefreiheit UN Behindertenrechtskonvention*. de. Juli 2013. URL: <https://www.behindertenrechtskonvention.info/barrierefreiheit-3881/> (besucht am 19.04.2023).
- [hic21] hickey. *Accessibility (Design basics) Win32 apps*. en us. Juni 2021. URL: <https://learn.microsoft.com/en-us/windows/win32/uxguide/inter-accessibility> (besucht am 04.04.2023).
- [Cou21] Council of European Union. *EN 301 549 Accessibility requirements for ICT products and services*. [https://www.etsi.org/deliver/etsi\\_en/301500\\_301599/301549/03.02.01\\_60/en\\_301549v030201p.pdf](https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf). 2021.
- [Cal+08] Ben Caldwell u. a. „Web content accessibility guidelines (WCAG) 2.0“. In: *WWW Consortium (W3C) 290* (2008), S. 1–34.
- [BS12] Mark C. Barlet und Steve D. Spohn. *Includification*. 1.4. The AbleGamers Foundation, 2012. 48 S. ISBN: 978 1 4792 8935 6. URL: <https://accessible.games/includification/> (besucht am 10.10.2022).
- [AH43] Maslow AH. „Maslow’s hierarchy of needs“. In: *Psychol Rev* 50 (1943), S. 370–96.
- [Tre22] Hansruedi Tresp. *Agile objektorientierte Anforderungsanalyse: Planen Ermitteln Analysieren Modellieren Dokumentieren Prüfen*. erfolgreich studieren. Wiesbaden: Springer Fachmedien Wiesbaden, 2022. ISBN: 978 3 658 37194 4. DOI: 10.1007/978-3-658-37194-4. URL: <https://link.springer.com/10.1007/978-3-658-37194-4> (besucht am 26.10.2022).
- [Gam+09] Erich Gamma u. a. *Design patterns: elements of reusable object oriented software*. OCLC: 624525693. Reading, MA: Addison Wesley, 2009. ISBN: 978 0 321 70069 8.



- [AHA21] Sara H. S. Almadi, Danial Hooshyar und Rodina Binti Ahmad. „Bad Smells of Gang of Four Design Patterns: A Decade Systematic Literature Review“. en. In: *Sustainability* 13.18 (Jan. 2021). Number: 18 Publisher: Multidisciplinary Digital Publishing Institute, S. 10256. ISSN: 2071 1050. DOI: 10.3390/su131810256. URL: [https://www.mdpi.com/2071\\_1050/13/18/10256](https://www.mdpi.com/2071_1050/13/18/10256) (besucht am 04.04.2023).
- [Nys14] Robert Nystrom. *Game programming patterns*. eng. s.l.: genever benning, 2014. ISBN: 978 0 9905829 0 8. URL: <https://gameprogrammingpatterns.com>.
- [Mar05] Robert C. Martin. *The Principles of OOD*. 2005. URL: <http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod> (besucht am 17.02.2023).
- [Mar14] Robert C. Martin. *The Single Responsibility Principle*. Blog. Mai 2014. URL: [https://blog.cleancoder.com/uncle\\_bob/2014/05/08/SingleReponsibilit.html](https://blog.cleancoder.com/uncle_bob/2014/05/08/SingleReponsibilit.html) (besucht am 17.02.2023).
- [Jac14] James E. Jackson. „Towards Universally Accessible Typography: A Review of Research on Dyslexia“. In: *Journal on Technology and Persons with Disabilities* Santiago, J. (Eds): Annual International Technology and Persons with Disabilities Conferenc (2014).
- [Gal+20] Jessica Galliussi u. a. „Inter letter spacing, inter word spacing, and font with dyslexia friendly features: testing text readability in people with and without dyslexia“. en. In: *Annals of Dyslexia* 70.1 (Apr. 2020), S. 141 152. ISSN: 1934 7243. DOI: 10.1007/s11881\_020\_00194\_x. URL: [https://doi.org/10.1007/s11881\\_020\\_00194\\_x](https://doi.org/10.1007/s11881_020_00194_x) (besucht am 11.01.2023).
- [Ama] Amazon. *Design and User Experience Guidelines (Fire TV) | Amazon Fire TV*. URL: [https://developer.amazon.com/docs/fire\\_tv/design\\_and\\_user\\_experience\\_guidelines.html#foot\\_ui](https://developer.amazon.com/docs/fire_tv/design_and_user_experience_guidelines.html#foot_ui) (besucht am 12.01.2023).
- [RPM16] Luz Rello, Martin Pielot und Mari Carmen Marcos. „Make It Big! The Effect of Font Size and Line Spacing on Online Readability“. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, S. 3637 3648. ISBN: 978 1 4503 3362 7. DOI: 10.1145/2858036.2858204. URL: <https://doi.org/10.1145/2858036.2858204> (besucht am 12.01.2023).
- [ČKP17] Darijo Čerepinko, Damira Keček und Mario Periša. „Text readability and legibility on iPad with comparison to paper and computer screen“. en. In: *Tehnicki vjesnik Technical Gazette* 24.4 (Aug. 2017). ISSN: 13303651, 18486339. DOI: 10.17559/TV\_20160225140202. URL: <http://hrcak.srce.hr/185497> (besucht am 12.01.2023).

- [Bad17] Wessel Badenhorst. „The Singleton Pattern“. en. In: *Practical Python Design Patterns: Pythonic Solutions to Common Problems*. Hrsg. von Wessel Badenhorst. Berkeley, CA: Apress, 2017, S. 23–35. ISBN: 978 1 4842 2680 3. DOI: 10.1007/978 1 4842 2680 3\_2. URL: [https://doi.org/10.1007/978 1 4842 2680 3\\_2](https://doi.org/10.1007/978 1 4842 2680 3_2) (besucht am 04.04.2023).
- [Fre15] Adam Freeman. „The Singleton Pattern“. en. In: *Pro Design Patterns in Swift*. Hrsg. von Adam Freeman. Berkeley, CA: Apress, 2015, S. 113–136. ISBN: 978 1 4842 0394 1. DOI: 10.1007/978 1 4842 0394 1\_6. URL: [https://doi.org/10.1007/978 1 4842 0394 1\\_6](https://doi.org/10.1007/978 1 4842 0394 1_6) (besucht am 04.04.2023).
- [SL08] Michael Suess und Claudia Leopold. „Problems, workarounds and possible solutions implementing the singleton pattern with C++ and OpenMP“. In: *A Practical Programming Model for the Multi Core Era: 3rd International Workshop on OpenMP, IWOMP 2007, Beijing, China, June 3–7, 2007 Proceedings 3*. Springer. 2008, S. 173–184.
- [Conb] Godot Contributors. *Dictionary*. en. URL: [https://docs.godotengine.org/en/stable/classes/class\\_dictionary.html](https://docs.godotengine.org/en/stable/classes/class_dictionary.html) (besucht am 19.04.2023).
- [Conc] Godot Contributors. *Array*. en. URL: [https://docs.godotengine.org/en/stable/classes/class\\_array.html](https://docs.godotengine.org/en/stable/classes/class_array.html) (besucht am 19.04.2023).
- [KNT20] Alina Karakanta, Matteo Negri und Marco Turchi. *MuST Cinema: a Speech to Subtitles corpus*. en. arXiv:2002.10829 [cs]. Feb. 2020. URL: <http://arxiv.org/abs/2002.10829> (besucht am 24.02.2023).
- [JG23] Ariel Manzur Juan Linietsky und the Godot community. *Audio buses*. en. 2023. URL: [https://docs.godotengine.org/en/stable/tutorials/audio/audio\\_buses.html](https://docs.godotengine.org/en/stable/tutorials/audio/audio_buses.html) (besucht am 12.01.2023).
- [Leb23] Emmanuel Leblond. *touilleMan/godot python*. original date: 2016-09-25T13:03:54Z. Apr. 2023. URL: <https://github.com/touilleMan/godot-python> (besucht am 19.04.2023).
- [M B20] Natesh M Bhat. *pyttsx3: Text to Speech (TTS) library for Python 2 and 3. Works without internet connection or delay. Supports multiple TTS engines, including Sapi5, nsss, and espeak*. Juli 2020. URL: <https://github.com/nateshmbhat/pyttsx3> (besucht am 19.04.2023).
- [Dur23] Pierre Nicolas Durette. *gTTS: gTTS (Google Text to Speech), a Python library and CLI tool to interface with Google Translate text to speech API*. Jan. 2023.
- [pyg23] pygame. *pygame* · PyPI. März 2023. URL: <https://pypi.org/project/pygame/> (besucht am 19.04.2023).

- [Nig20] Nightblade [@nightblade99]. *Doesn't look flashy, but got two way communication going between a #GodotEngine concentration game and a #DotNetCore terminal (for screen readers. Godot prints messages (top left), sends messages on click; C# can read and write messages, too. #GameDev #Prototype #WIP #A11y* <https://t.co/XWhBWtwLY9>. en. Tweet. Dez. 2020. URL: <https://twitter.com/nightblade99/status/1337225618442891266> (besucht am 19.04.2023).
- [Wil23] Godot Wild Jam. *Welcome to Godot Wild Jam!* en US. 2023. URL: <https://godotwildjam.com/> (besucht am 19.04.2023).
- [Bro95] John Brooke. „SUS: A quick and dirty usability scale“. In: *Usability Eval. Ind.* 189 (Nov. 1995).
- [HM21] Imke Hedder und Jana Michels. „Evaluationen zwischen Wunsch und Realität“. de DE. In: *Wissenschaftskommunikation.de* (März 2021). URL: <https://www.wissenschaftskommunikation.de/evaluationen-zwischen-wunsch-und-realitaet-46349/> (besucht am 04.04.2023).
- [Bor+20] Markus Borg u. a. „Video Game Development in a Rush: A Survey of the Global Game Jam Participants“. In: *IEEE Transactions on Games* PP (Sep. 2020), S. 1 1. DOI: 10.1109/TG.2019.2910248.
- [Mel+19] Leandro de Almeida Melo u. a. „What Motivates Different People to Participate in Game Jams?“ en. In: *Anais Estendidos do Simpósio Brasileiro de Sistemas Colaborativos (SBSC)*. ISSN: 0000 0000. SBC, Okt. 2019, S. 135 140. URL: <https://sol.sbc.org.br/index.php/sbsestendido/article/view/8365> (besucht am 04.04.2023).
- [Umb04] Paul D. Umbach. „Web surveys: Best practices“. In: *New Directions for Institutional Research* 2004.121 (2004), S. 23 38. DOI: <https://doi.org/10.1002/ir.98>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ir.98>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ir.98>.
- [LH17] Pei Wen Liao und Jun Yi Hsieh. „Does Internet Based Survey Have More Stable and Unbiased Results than Paper and Pencil Survey?“ en. In: *Open Journal of Social Sciences* 5.1 (Jan. 2017). Number: 1 Publisher: Scientific Research Publishing, S. 69 86. DOI: 10.4236/jss.2017.51006. URL: <http://www.scirp.org/Journal/Paperabs.aspx?paperid=73441> (besucht am 04.04.2023).
- [CHT00] Colleen Cook, Fred Heath und Russel L. Thompson. „A Meta Analysis of Response Rates in Web or Internet Based Surveys“. en. In: *Educational and Psychological Measurement* 60.6 (Dez. 2000). Publisher: SAGE Publica

- tions Inc, S. 821–836. ISSN: 0013-1644. DOI: 10.1177/00131640021970934. URL: <https://doi.org/10.1177/00131640021970934> (besucht am 04.04.2023).
- [CCL01] Scott D. Crawford, Mick P. Couper und Mark J. Lamias. „Web Surveys: Perceptions of Burden“. en. In: *Social Science Computer Review* 19.2 (Mai 2001). Publisher: SAGE Publications Inc, S. 146–162. ISSN: 0894-4393. DOI: 10.1177/089443930101900202. URL: <https://doi.org/10.1177/089443930101900202> (besucht am 04.04.2023).
- [RAR11] Sindre Rolstad, John Adler und Anna Rydén. „Response Burden and Questionnaire Length: Is Shorter Better? A Review and Meta analysis“. en. In: *Value in Health* 14.8 (Dez. 2011), S. 1101–1108. ISSN: 1098-3015. DOI: 10.1016/j.jval.2011.06.003. URL: <https://www.sciencedirect.com/science/article/pii/S1098301511015245> (besucht am 07.04.2023).
- [Sha22] Hunny Sharma. „How short or long should be a questionnaire for any research? Researchers dilemma in deciding the appropriate questionnaire length“. en. In: *Saudi Journal of Anaesthesia* 16.1 (März 2022). Publisher: Wolters Kluwer – Medknow Publications, S. 65. DOI: 10.4103/sja.sja\_163\_21. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8846243/> (besucht am 07.04.2023).
- [GB09] Mirta Galesic und Michael Bosnjak. „Effects of Questionnaire Length on Participation and Indicators of Response Quality in a Web Survey“. In: *Public Opinion Quarterly* 73.2 (Jan. 2009), S. 349–360. ISSN: 0033-362X. DOI: 10.1093/poq/nfp031. URL: <https://doi.org/10.1093/poq/nfp031> (besucht am 07.04.2023).
- [HR23] Seth J. Hill und Margaret E. Roberts. „Acquiescence Bias Inflates Estimates of Conspiratorial Beliefs and Political Misperceptions“. en. In: *Political Analysis* (Jan. 2023). Publisher: Cambridge University Press, S. 1–16. ISSN: 1047-1987, 1476-4989. DOI: 10.1017/pan.2022.28. URL: <https://www.cambridge.org/core/journals/political-analysis/article/acquiescence-bias-inflates-estimates-of-conspiratorial-beliefs-and-political-misperceptions/36461392CC22A6AC6813767D64490A64> (besucht am 07.04.2023).
- [DK21] Alexander De Juan und Carlo Koos. „Survey participation effects in conflict research“. In: *Journal of Peace Research* 58.4 (Juli 2021). Publisher: SAGE Publications Ltd, S. 623–639. ISSN: 0022-3433. DOI: 10.1177/0022343320971034. URL: <https://doi.org/10.1177/0022343320971034> (besucht am 07.04.2023).

- [SL11] Jeff Sauro und James R. Lewis. „When designing usability questionnaires, does it hurt to be positive?“ en. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vancouver BC Canada: ACM, Mai 2011, S. 2215–2224. ISBN: 978 1 4503 0228 9. DOI: 10.1145/1978942.1979266. URL: <https://dl.acm.org/doi/10.1145/1978942.1979266> (besucht am 07.04.2023).
- [Con08] GamesIndustry International Contributor. *PopCap Games research*. en. Juni 2008. URL: <https://www.gamesindustry.biz/popcap-games-research-publisher-s-latest-survey-says-that-casual-games-are-big-with-disabled-people> (besucht am 07.04.2023).
- [SH06] Lothar Sachs und Jürgen Hedderich. *Angewandte Statistik*. de. Springer Berlin Heidelberg, 2006. ISBN: 978 3 540 32160 6. DOI: 10.1007/978 3 540 32161 3. URL: <http://link.springer.com/10.1007/978 3 540 32161 3> (besucht am 14.04.2023).
- [SL21] Philipp Sibbertsen und Hartmut Lehne. *Statistik: Einführung für Wirtschafts- und Sozialwissenschaftler*. de. Berlin, Heidelberg: Springer, 2021. ISBN: 978 3 662 62695 5 978 3 662 62696 2. DOI: 10.1007/978 3 662 62696 2. URL: <https://link.springer.com/10.1007/978 3 662 62696 2> (besucht am 14.04.2023).
- [Coh88] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. 2. Aufl. New York: Routledge, Juli 1988. ISBN: 978 0 203 77158 7. DOI: 10.4324/9780203771587.
- [BM03] Frank E. Beichelt und Douglas C. Montgomery. „Mathematische Statistik“. de. In: *Teubner Taschenbuch der Stochastik: Wahrscheinlichkeitstheorie, Stochastische Prozesse, Mathematische Statistik*. Hrsg. von Frank E. Beichelt und Douglas C. Montgomery. Wiesbaden: Vieweg+Teubner Verlag, 2003, S. 229–402. ISBN: 978 3 322 80067 1. DOI: 10.1007/978 3 322 80067 1\_4. URL: [https://doi.org/10.1007/978 3 322 80067 1\\_4](https://doi.org/10.1007/978 3 322 80067 1_4) (besucht am 14.04.2023).
- [Cond] Godot Contributors. *Godot 4.0 sets sail: All aboard for new horizons*. en. URL: <https://godotengine.org/article/godot-4-0-sets-sail/> (besucht am 14.04.2023).
- [Bun] Statistisches Bundesamt. *Behinderte Menschen*. de. URL: [https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/\\_inhalt.html](https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/_inhalt.html) (besucht am 13.04.2023).



## Anhang A: Anforderungskatalog

**Tabelle A.1:** Anforderungskatalog – Accessibility Features

<b>Identifikation</b>	<b>Titel</b>	<b>Beschreibung</b>	<b>Priorität</b>
REQ 010	motorische Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 011 bis REQ 015 erfüllt sind.	mittel
REQ 011	konfigurierbare Tastenbelegung	Die Steuerung muss Nutzer*innen die Möglichkeit bieten, sie neu belegen zu können.	hoch
REQ 012	Userinterface navigation	Das Userinterface muss Nutzer*innen die Möglichkeit bieten, alle Interaktionsflächen über dieselbe Eingabemethode, wie die Steuerung des Spiels zu erreichen.	mittel
REQ 013	konfigurierbare Sensitivität	Die Steuerung muss Nutzer*innen die Möglichkeit bieten, die Sensitivität dieser anzupassen.	gering
REQ 014	simple Steuerung	Die Steuerung sollte so gestaltet sein, dass sie so einfach wie möglich ist.	mittel
REQ 015	konfigurierbares haptisches Feedback	Haptisches Feedback muss Nutzer*innen die Möglichkeit bieten, die Intensität einzustellen und sie komplett zu deaktivieren.	gering
REQ 020	kognitive Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 021 bis REQ 027 erfüllt sind.	mittel

REQ 021	geringer Einstiegsaufwand	Das Spiel muss Nutzer*innen die Möglichkeit bieten, es zu starten, ohne dass mehrere Menüs navigiert werden müssen.	hoch
REQ 022	leicht lesbare Schriftart	Die Standardschriftart muss so gestaltet sein, dass sie für Nutzer*innen leicht lesbar ist.	hoch
REQ 023	einfache Sprache	Texte sollen so gestaltet sein, dass sie einfache, klare Sprache verwenden.	gering
REQ 024	klare Formatierung	Texte sollten klar formatiert sein.	gering
REQ 025	interaktives Tutorial	Das Spiel sollte so gestaltet sein, dass es interaktive Tutorials für die Nutzer*innen gibt.	mittel
REQ 026	individuelle Textgeschwindigkeit	Texte sollten Nutzer*innen die Möglichkeit bieten, sie in einer individuellen Geschwindigkeit voranschreiten zu lassen.	gering
REQ 027	Muster und flackernde Bilder	Das Spiel muss so gestaltet werden, dass flackernde Bilder und sich wiederholende Muster vermieden werden.	mittel
REQ 030	visuelle Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 031 bis REQ 037 erfüllt sind.	hoch
REQ 031	visuelle Informationsgestaltung	Essenzielle Informationen müssen so gestaltet werden, dass sie nicht nur über Farben dargestellt werden.	hoch
REQ 032	konfigurierbares Sichtfeld	Das Sichtfeld (FOV) soll so eingestellt werden, dass Übelkeit bei Nutzer*innen vermieden wird.	gering



REQ 033	Simulation Sickness	Auslöser für Simulation Sickness in VR müssen vermieden werden.	hoch
REQ 034	leicht lesbare Schriftgröße	Die Standardtextgröße muss so gewählt werden, dass sie für Nutzer*innen leicht lesbar ist.	hoch
REQ 035	klar formatierte Textdarstellung	Texte sollen so dargestellt werden, dass sie klar formatiert sind.	gering
REQ 036	hoher visueller Kontrast	Textbereiche müssen so gestaltet sein, dass es einen hohen visuellen Kontrast zwischen Text und Hintergrund gibt.	hoch
REQ 037	konfigurierbare Schriftgröße	Das Spiel sollte Nutzer*innen die Möglichkeit bieten, die Textgröße um bis zu 200% zu erhöhen, ohne dass dabei Einschränkungen der Inhalte oder der Funktionalität ergeben. (siehe WCAG 1.4.4)	hoch
REQ 038	Screenreader Kompatibilität	Texte und Menüs im Spiel sollten so integriert sein, dass sie von gängigen Screenreadern (JAWS, NVDA) vorgelesen werden können.	gering
REQ 040	auditive Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 041 bis REQ 045 erfüllt sind.	hoch
REQ 041	grundlegende Untertitel	Untertitel müssen für alle wichtigen gesprochenen Texte vorhanden sein.	hoch
REQ 042	konfigurierbare Lautstärke	Die Lautstärke muss Nutzer*innen die Möglichkeit bieten, diese separat für Effekte, Sprache und Musik einzustellen.	hoch

REQ 043	auditive Informationsgestaltung	Essenzielle Informationen müssen so gestaltet werden, dass sie nicht nur über Töne dargestellt werden.	hoch
REQ 044	leicht lesbare Untertitel	Untertitel müssen so gestaltet sein, dass sie von Nutzer*innen leicht gelesen werden können.	hoch
REQ 045	optionale Spracheingaben	Spracheingaben dürfen nicht eine nötige Eingabemöglichkeit sein, sondern nur eine weitere bzw. alternative.	hoch
REQ 050	allgemeine Anforderungen	Diese Anforderung ist erfüllt, wenn die untergeordneten Anforderungen REQ 051 bis REQ 055 erfüllt sind.	mittel
REQ 051	individualisierbare Schwierigkeit	Die Schwierigkeit soll so gestaltet sein, dass es für Nutzer*innen eine breite Auswahl gibt.	gering
REQ 052	Accessibility Informationen außerhalb des Spiels	Informationen über die Accessibility Features müssen bereits vor dem Start des Spiels für Nutzer*innen zugänglich sein.	mittel
REQ 053	Accessibility Informationen innerhalb des Spiels	Informationen über die Accessibility Features müssen im Spiel für Nutzer*innen zugänglich sein.	mittel
REQ 054	persistente Einstellungen	Einstellungen, die von Nutzer*innen im Spiel getätigt wurden, müssen gespeichert und nach einem Neustart geladen werden.	hoch
REQ 055	Feedback möglichkeit	Das Spiel sollte Nutzer*innen die Möglichkeit bieten, Feedback zu den Accessibility Features zu geben.	mittel



**Sociodemographic Questions**

What is your experience level in the following tools/ skills?

	Novice	Advanced Beginner	Competent	Proficient	Expert
Godot Engine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Game Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Development of Accessi bility Features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usage of Add ons	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you use accessibility options yourself?      yes      no  
     

**Usage of the Add-on**

Have you used the Godot Game Accessi  
bility Addon, that was developed for this  
study?      yes      no  
     

If not, can you please elaborate, why you didn't use the Add on?

.....  
 .....

**Control Remapping and Mouse Sensitivity**

The following questions are about the examples and systems for the control remapping and the mouse sensitivity.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
 .....

### Keyboard Navigation of the UI

The following questions are about the examples and systems for keyboard navigation of the UI.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
.....

**Font Readability**

The following questions are about the examples and systems for readability of the font. Especially the readability and size of the default font, as well as the system for resizing the font.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
 .....

**Volume Controls**

The following questions are about the examples and systems for controlling the volume.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
 .....



## Subtitles

The following questions are about the examples and systems for the basic subtitle system, as well as the readability of the subtitles.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
 .....

## Accessibility Information

The following questions are about the examples and systems for informing the users about the implemented accessibility features. Especially the information outside the project (README file) as well as through an in game menu. This also concerns the ways to give feedback, either via email, Discord or GitHub Issues.

	strongly disagree	somewhat disagree	unsure	somewhat agree	strongly agree
I think I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the various functions in this system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the system very cumbersome to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Do you have anything additional you want to say/ include?

.....  
.....

# Anhang C: Fragebogen: Gruppe A

## Evaluation Usability

Vielen Dank, dass Sie sich die Zeit genommen haben, mir bei meiner Masterarbeit zu helfen.

Bei dieser Bewertung geht es um die Benutzerfreundlichkeit der Systeme und Beispiele, die in dem Godot Game Accessibility Guidelines Projekt implementiert wurden. Ihre Antworten werden verwendet, um zu analysieren, wie effektiv die Umsetzung im Hinblick auf die Benutzerfreundlichkeit war. Sie können diese Umfrage jederzeit abbrechen, ohne dass dies negative Folgen hat.

Bitte fangen Sie zuerst mit der Bearbeitung dieser Umfrage an und führen Sie nur die Aufgaben im Aufgaben Dokument aus, wenn Sie hierzu in der Evaluation aufgefordert werden. Geschätzte Zeit: 30 Minuten

Bitte beachten Sie, dass Sie 18 Jahre oder älter sein müssen, um an dieser Studie teilnehmen zu können.

Ihre Antworten werden anonym gespeichert und für die Forschung während dieser und folgender Studien verwendet.

	Ja	Nein
Geben Sie Ihre Zustimmung?	<input type="radio"/>	<input type="radio"/>

**Soziodemografische Fragen**

Was ist Ihr Wissenslevel in den folgenden Programmen/ Skills?

	kein Wissen	Anfänger	Fortge- schrittener	Experte
Godot Engine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Game Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Implementation von Accessibility Features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nutzung von Add Ons in der Go dot Engine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Ja	Nein
Nutzen Sie Accessibility Features in Spielen oder Programmen?	<input type="radio"/>	<input type="radio"/>

## Teil A – Accessibility

Bearbeiten Sie bitte den Teil A im Aufgaben Dokument und beantworten Sie danach oder analog die nachfolgenden Fragen.

Aufgaben Dokument: <https://tinyurl.com/3e3jweux>

Wie schätzen Sie die Aufgabe **A1 Steuerung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A2 Menüführung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A3 Untertitel** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A4 Lautstärke** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A5 Information** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



**Teil B – Godot Nativ**

Bearbeiten Sie bitte den Teil B im Aufgaben Dokument und beantworten Sie danach oder analog die nachfolgenden Fragen.

Aufgaben Dokument: <https://tinyurl.com/3e3jweux>

Wie schätzen Sie die Aufgabe **B1 Steuerung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **B2 Lautstärke** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **B3 Informationen** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **B4 Schriftart** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A5 Information** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Sonstiges**

Falls Sie noch sonstiges Feedback haben, können Sie dieses gerne hier schreiben.

.....  
.....

## Anhang D: Fragebogen: Gruppe B

### Evaluation Usability

Vielen Dank, dass Sie sich die Zeit genommen haben, mir bei meiner Masterarbeit zu helfen.

Bei dieser Bewertung geht es um die Benutzerfreundlichkeit der Systeme und Beispiele, die in dem Godot Game Accessibility Guidelines Projekt implementiert wurden. Ihre Antworten werden verwendet, um zu analysieren, wie effektiv die Umsetzung im Hinblick auf die Benutzerfreundlichkeit war. Sie können diese Umfrage jederzeit abbrechen, ohne dass dies negative Folgen hat.

Bitte fangen Sie zuerst mit der Bearbeitung dieser Umfrage an und führen Sie nur die Aufgaben im Aufgaben Dokument aus, wenn Sie hierzu in der Evaluation aufgefordert werden. Geschätzte Zeit: 30 Minuten

Bitte beachten Sie, dass Sie 18 Jahre oder älter sein müssen, um an dieser Studie teilnehmen zu können.

Ihre Antworten werden anonym gespeichert und für die Forschung während dieser und folgender Studien verwendet.

	Ja	Nein
Geben Sie Ihre Zustimmung?	<input type="radio"/>	<input type="radio"/>

### Soziodemografische Fragen

Was ist Ihr Wissenslevel in den folgenden Programmen/ Skills?

	kein Wissen	Anfänger	Fortge- schrittener	Experte
Godot Engine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Game Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Implementation von Accessibility Features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nutzung von Add Ons in der Go dot Engine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Ja	Nein
Nutzen Sie Accessibility Features in Spielen oder Programmen?	<input type="radio"/>	<input type="radio"/>

**Teil B – Godot Nativ**

Bearbeiten Sie bitte den Teil B im Aufgaben Dokument und beantworten Sie danach oder analog die nachfolgenden Fragen.

Aufgaben Dokument: <https://tinyurl.com/3e3jweux>

Wie schätzen Sie die Aufgabe **B1 Steuerung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **B2 Lautstärke** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **B3 Informationen** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Wie schätzen Sie die Aufgabe **B4 Schriftart** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A5 Information** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Teil A – Accessibility

Bearbeiten Sie bitte den Teil A im Aufgaben Dokument und beantworten Sie danach oder analog die nachfolgenden Fragen.

Aufgaben Dokument: <https://tinyurl.com/3e3jweux>

Wie schätzen Sie die Aufgabe **A1 Steuerung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A2 Menüführung** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A3 Untertitel** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A4 Lautstärke** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wie schätzen Sie die Aufgabe **A5 Information** ein?

	trifft zu	trifft eher zu	teils teils	trifft eher nicht zu	trifft nicht zu
Die Aufgabenstellung war schwer zu verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fiel mir einfach, die Aufgabe zu erfüllen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Für zukünftige Anwendungen würde ich den selben Lösungsansatz verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Lösung der Aufgabe war unnötig aufwendig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Mehrwert des Lösungsansatzes ist für mich offensichtlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Sonstiges**

Falls Sie noch sonstiges Feedback haben, können Sie dieses gerne hier schreiben.

.....  
.....

## Anhang E: Aufgaben Gruppe A & B

Laden Sie bitte Godot herunter <https://godotengine.org/download/3.x/windows/> und entpacken Sie es. Wichtig ist, dass Sie nicht die „Godot Engine .NET“ Version wählen, sondern die normale Version nutzen.

Für die nachfolgenden Bearbeitungen ist es vorab relevant, dass nur die Antworten aus der Evaluation gespeichert und in meiner Masterarbeit ausgewertet werden. Die Ausarbeitung der nachfolgenden Aufgaben bleibt komplett bei Ihnen und keine Informationen, die Sie dabei angeben wird in jeglicher Form weitergegeben.

### Teil A – Accessibility

Laden Sie das Projekt unter dem Link: [https://github.com/MilasBaumgarten/Godot\\_GameAccessibilityGuidelines\\_Addon/tree/Evaluation](https://github.com/MilasBaumgarten/Godot_GameAccessibilityGuidelines_Addon/tree/Evaluation) herunter. Hierfür können Sie über den Button „Code“ das Projekt als eine ZIP Datei herunterladen. Starten Sie dann die Godot Engine, indem Sie auf „Godot\_v3.5.2 stable\_win64.exe“ klicken. Importieren Sie das „Godot Game Accessibility Guidelines“ Projekt, indem Sie auf „Import“ drücken und dort das Projekt auswählen.

### A1 Steuerung

Erstellen Sie nun die Input Actions:

- „move\_left“
- „move\_right“
- „move\_forward“
- „move\_back“

Zur Hilfe finden Sie die offizielle Anleitung unter folgendem Link: [https://docs.godotengine.org/en/3.6/getting\\_started/first\\_3d\\_game/02.player\\_input.html#creating\\_input\\_actions](https://docs.godotengine.org/en/3.6/getting_started/first_3d_game/02.player_input.html#creating_input_actions). Erweitern Sie als Nächstes das Tastaturbelegungsmenü, dass sie im FileSystem Menü unten links in Godot finden (scenes/keybinding\_menu.tscn). Editieren Sie dafür das Label unter „KeybindingMenu/Panel/HBoxContainer/VBoxContainer\_Labels/Label“ und schreiben Sie in das Feld „Text“ im Inspektor den Namen der Input Action, für den dieses Label steht („move\_left“, „move\_right“, „move\_forward“, „move\_back“).

Konfigurieren Sie nun den Button unter „KeybindingMenu/Panel/HBoxContainer/VBoxContainer\_Buttons/Button“, indem Sie diesen auswählen und im Feld „Action“

den Namen der Input Action eingeben, die mit diesem Button von Nutzer\*innen konfiguriert werden soll. Duplizieren Sie danach das Label und den Button dreimal und wiederholen Sie die Schritte 1 und 2 für die weiteren Input Actions.

### **A2 Menüführung**

Ziehen Sie das Skript „addons/ui\_keyboard\_navigation/other\_grab\_focus.gd“ auf die Root Node des Tastaturbelegungsmenüs (scenes/keybinding\_menu.tscn). Konfigurieren Sie dann im Inspektor unter dem Feld „UI Element Path“ das UI Element, das initial fokussiert werden soll.

### **A3 Untertitel**

Erstellen Sie in einer beliebigen Szene einen Button und ziehen Sie das Skript „addons/subtitles/subtitle\_adder.gd“ auf diesen. Konfigurieren Sie dann den Text, der beim Drücken des Buttons als Untertitel angezeigt wird, sowie die Anzeigedauer.

In einem Projekt, das dieses Untertitelsystem nutzen möchte, würde anstelle des „subtitle\_adder.gd“ Skripts das Untertitelsystem auf andere Wege angesprochen. Hierfür ist es lediglich nötig, die Codezeile: „Subtitles.add\_to\_queue(text, duration)“ auszuführen. Für Demonstrationszwecke wird in dieser Evaluation und in dem Projekt die beispielhafte Anwendung mittels eines Buttons dargestellt.

### **A4 Lautstärke**

Erstellen Sie jetzt einen weiteren Audiobus namens „Music“ und ordnen Sie diesem dem Master Audiobus unter. Die offizielle Dokumentation hierfür finden Sie auf der Seite: [https://docs.godotengine.org/en/3.6/tutorials/audio/audio\\_buses.html](https://docs.godotengine.org/en/3.6/tutorials/audio/audio_buses.html). Öffnen Sie dann das Audiomenu (scenes/audio\_mixer\_menu.tscn) und erweitern Sie es um ein Label und einen Slider für den „Music“ Audiobus. Beachten Sie dabei, dass im Slider der Name des Busses eingetragen werden muss.

### **A5 Informationen**

Die folgenden Aufgaben können Sie nur teilweise innerhalb der Godot Engine lösen. Nutzen Sie hierfür geläufige, andere Programme. Unter Windows bietet sich hier die Verwendung des Explorers und des Programms Notepad an.

Erstellen Sie im Projektordner eine Textdatei namens „README“ und beschreiben Sie in dieser kurz, welche Elemente das Programm besitzt (z. B. welche Eingabemöglichkeiten, für die Input Actions implementiert wurden). Geben Sie außerdem eine Möglichkeit an, wie Nutzer\*innen Sie für Feedback erreichen könnten (z. B. E Mail Adresse, Social Media).

Navigieren Sie im Projekt zudem zum Accessibility Informationsmenü (scenes/accessibility\_information.tscn) und schreiben Sie dort ebenfalls eine Beschreibung. Hierfür können Sie auch denselben Text, den Sie bereits in die README geschrieben haben, nutzen.

Hinterlegen Sie bitte ebenfalls eine Kontaktmöglichkeit. Hierfür können Sie das Label unten rechts im Menü nutzen.

## **Teil B – Godot Nativ**

Starten Sie die Godot Engine, indem Sie auf „Godot\_v3.5.2 stable\_win64.exe“ klicken. Erstellen Sie dann ein neues Projekt, indem Sie auf „New Project“ klicken. Vergeben Sie nun im Feld „Project Name“ einen beliebigen Namen für das Programm und suchen Sie im Feld „Project Path“ den Ort aus, an dem das Projekt gespeichert werden soll. Drücken Sie hierfür auf den Button „Browse“ und wählen Sie den Ordner mit dem Button „Select Current Folder“ aus. Beachten Sie, dass es sich um einen leeren Ordner handeln muss. Alternativ können Sie am ausgewählten Ort einen leeren Ordner erstellen, indem Sie auf den Button „Create Folder“ neben dem Projektnamen klicken. Erstellen Sie jetzt das Projekt mit einem Klick auf den Button „Create & Edit“.

### **B1 Steuerung**

Erstellen Sie die Input Actions „move\_left“, „move\_right“, „move\_forward“ und „move\_back“. Zur Hilfe finden Sie die offizielle Anleitung unter folgendem Link: [https://docs.godotengine.org/en/3.6/getting\\_started/first\\_3d\\_game/02.player\\_input.html#creating\\_input\\_actions](https://docs.godotengine.org/en/3.6/getting_started/first_3d_game/02.player_input.html#creating_input_actions).

### **B2 Lautstärke**

Erstellen Sie jetzt einen weiteren Audiobus namens „Music“ und ordnen Sie diesem dem Master Audiobus unter. Die offizielle Dokumentation hierfür finden Sie auf der Seite: [https://docs.godotengine.org/en/3.6/tutorials/audio/audio\\_buses.html](https://docs.godotengine.org/en/3.6/tutorials/audio/audio_buses.html).

### **B3 Informationen**

Die folgenden Aufgaben können Sie nicht innerhalb der Godot Engine lösen. Nutzen Sie hierfür geläufige, andere Programme. Unter Windows bietet sich hier die Verwendung des Explorers und des Programms Notepad an.

Erstellen Sie im Projektordner eine Textdatei namens „README“ und beschreiben Sie in dieser kurz, welche Elemente das Programm besitzt (z. B. welche Eingabemöglichkeiten, für die Input Actions implementiert wurden). Geben Sie außerdem eine Möglichkeit an, wie Nutzer\*innen Sie für Feedback erreichen könnten (z. B. E Mail Adresse, Social Media).



#### **B4 Schriftart**

Konfigurieren Sie nun eine Schriftart als Standard, die für Darstellungen in dem Programm verwendet werden soll. Besorgen Sie sich hierfür zuerst eine beliebige Schriftart (z. B. bei <https://fonts.google.com/>). Fügen Sie diese nun zu dem Programm hinzu, indem Sie die Datei in das Godot Fenster ziehen.

Erstellen Sie nun eine neue DynamicFont Ressource, indem Sie im Abschnitt FileSystem, der Godot Engine mit Rechtsklick eine neue Ressource erstellen. Wählen Sie dabei DynamicFont als Typ aus. Bestätigen Sie dann die Erstellung mit dem Button „Save“. Wähle Sie als Nächstes die neu erstellte Datei mit Doppelklick aus. Klappen Sie im Inspector den Unterpunkt „Font“ aus und ziehen Sie die Schriftart Datei auf das Feld „Font Data“.

Stellen Sie jetzt die erstellte und konfigurierte DynamicFont Ressource als Standard textart ein, indem Sie „Project“ und dann „Project Settings auswählen“. Dort navigieren Sie dann unter dem Reiter „General“ zu „GUI“ und dann zu „Theme“ und wählen dort mithilfe des Buttons neben dem Feld „Custom Font“ die DynamicFont Ressource aus. Abschließend müssen Sie die Engine neu starten. Drücken Sie hierfür auf den Button „Save & Restart“.

## Anhang F: Antworten: Gruppe A

Nachfolgend sind die Antworten der Proband\*innen aus Gruppe A dargestellt. Die Abkürzung „P“ steht in dem Tabellenkopf für „Proband\*inn“. Die Fragen sind aufgrund des Platzes abgekürzt. Sie entsprechen jedoch den Fragen in Anhang C und sind in derselben Reihenfolge aufgelistet. Des Weiteren wurden die Antworten kodiert. Die Tabellen F.1, F.2 und F.3 zeigen das genutzte Vorgehen.

Antwort	Kodierung
Nein	0
Ja	1

**Tabelle F.1:** Kodierung der binären Fragen

Antwort	Kodierung
kein Wissen	0
Anfänger	1
Fortgeschrittener	2
Experte	3

**Tabelle F.2:** Kodierung der Wissensfragen

Antwort	Kodierung
trifft nicht zu	1
trifft eher nicht zu	2
teils teils	3
trifft eher zu	4
trifft zu	5

**Tabelle F.3:** Kodierung der Likert-Skalen

Frage	Unterthema	P 1	P 2	P 3	P 4	P 5
Zustimmung	Antwortspeicherung	1	1	1	1	1
Wissenslevel	Godot Engine	0	1	0	0	0
	Game Development	2		2	0	2
	Implementation von Accessibility Features	1	1	2	0	1
	Add ons in der Godot Engine	0	0	0	0	0
Nutzung von Accessibility Features		1	0	0	1	1
A1	Schwer Verständlich	4	3	2	3	1
	Einfache Lösung	4	2	5	3	4
	Lösung Wiederverwenden	3	2	4	4	5
	Unnötiger Aufwand	5	3	2	2	2
	Offensichtlicher Mehrwert	3	4	3	1	4
A2	Schwer Verständlich	4	2	1	1	1
	Einfache Lösung	4	3	5	5	5
	Lösung Wiederverwenden	2	1	3	4	4
	Unnötiger Aufwand	2	4	1	1	1
	Offensichtlicher Mehrwert	4	2	2	1	4
A3	Schwer Verständlich	2	2	2	4	1
	Einfache Lösung	5	4	5	2	5
	Lösung Wiederverwenden	4	3	3	3	5
	Unnötiger Aufwand	2	2	5	1	1
	Offensichtlicher Mehrwert	4	4	2	1	5
A4	Schwer Verständlich	2	2	3	4	1
	Einfache Lösung	5	4	3	3	4
	Lösung Wiederverwenden	5	2	2	4	5
	Unnötiger Aufwand	1	2	4	2	1
	Offensichtlicher Mehrwert	5	5	3	1	5
A5	Schwer Verständlich	2	1	2	1	1
	Einfache Lösung	5	5	4	5	5
	Lösung Wiederverwenden	5	5	4	5	2
	Unnötiger Aufwand	2	2	4	1	1
	Offensichtlicher Mehrwert	3	5	4	1	3
B1	Schwer Verständlich	1	2	1	1	1
	Einfache Lösung	5	4	5	5	5
	Lösung Wiederverwenden	5	4	4	5	5
	Unnötiger Aufwand	1	2	2	1	1
	Offensichtlicher Mehrwert	5	4	5	1	3
B2	Schwer Verständlich	1	1	1	1	1
	Einfache Lösung	5	5	5	5	5

	Lösung Wiederverwenden	5	4	4	5	5
	Unnötiger Aufwand	1	2	1	1	1
	Offensichtlicher Mehrwert	5	3	3	1	4
B3	Schwer Verständlich	1	1	2	1	1
	Einfache Lösung	5	5	4	5	2
	Lösung Wiederverwenden	3	5	3	5	1
	Unnötiger Aufwand	1	2	3	1	4
	Offensichtlicher Mehrwert	3	4	4	1	3
B4	Schwer Verständlich	1	1	2	2	1
	Einfache Lösung	5	5	4	4	4
	Lösung Wiederverwenden	4	5	4	4	5
	Unnötiger Aufwand	2	2	3	1	2
	Offensichtlicher Mehrwert	3	5	1	1	5

**Tabelle F.4:** Antworten von Gruppe A

## Anhang G: Antworten: Gruppe B

Nachfolgend sind die Antworten der Proband\*innen aus Gruppe B dargestellt. Die Abkürzung „P“ steht in dem Tabellenkopf für „Proband\*inn“. Die Fragen sind aufgrund des Platzes abgekürzt. Sie entsprechen jedoch den Fragen in Anhang D und sind in derselben Reihenfolge aufgelistet. Des Weiteren wurden die Antworten kodiert. Die Tabellen G.1, G.2 und G.3 zeigen das genutzte Vorgehen.

Antwort	Kodierung
Nein	0
Ja	1

**Tabelle G.1:** Kodierung der binären Fragen

Antwort	Kodierung
kein Wissen	0
Anfänger	1
Fortgeschrittener	2
Experte	3

**Tabelle G.2:** Kodierung der Wissensfragen

Antwort	Kodierung
trifft nicht zu	1
trifft eher nicht zu	2
teils teils	3
trifft eher zu	4
trifft zu	5

**Tabelle G.3:** Kodierung der Likert-Skalen

Frage	Unterthema	P 1	P 2	P 3
Zustimmung	Antwortspeicherung	1	1	1
Wissenslevel	Godot Engine	1	0	0
	Game Development	2	2	1
	Implementation von Accessibility Features	1	0	1
	Add ons in der Godot Engine	0	0	0
Nutzung von Accessibility Features		0	0	0
B1	Schwer Verständlich	2	1	2
	Einfache Lösung	4	5	3
	Lösung Wiederverwenden	5	5	3
	Unnötiger Aufwand	1	1	2
	Offensichtlicher Mehrwert	5	3	4
B2	Schwer Verständlich	2	1	1
	Einfache Lösung	4	5	4
	Lösung Wiederverwenden	5	5	4
	Unnötiger Aufwand	1	1	1
	Offensichtlicher Mehrwert	5	5	4
B3	Schwer Verständlich	2	1	1
	Einfache Lösung	4	5	4
	Lösung Wiederverwenden	5	5	4
	Unnötiger Aufwand	1	1	2
	Offensichtlicher Mehrwert	5	5	4
B4	Schwer Verständlich	2	2	2
	Einfache Lösung	4	5	4
	Lösung Wiederverwenden	5	5	4
	Unnötiger Aufwand	1	1	2
	Offensichtlicher Mehrwert	5	5	4
A1	Schwer Verständlich	1	3	2
	Einfache Lösung	5	4	3
	Lösung Wiederverwenden	3	2	4
	Unnötiger Aufwand	1	5	2
	Offensichtlicher Mehrwert	4	2	5
A2	Schwer Verständlich	1	5	1
	Einfache Lösung	5	1	3
	Lösung Wiederverwenden	4	1	5
	Unnötiger Aufwand	2	5	2
	Offensichtlicher Mehrwert	4	1	5
A3	Schwer Verständlich	2	1	4
	Einfache Lösung	5	5	2

	Lösung Wiederverwenden	3	1	4
	Unnötiger Aufwand	1	1	3
	Offensichtlicher Mehrwert	4	1	4
A4	Schwer Verständlich	1	1	2
	Einfache Lösung	5	5	4
	Lösung Wiederverwenden	3	5	4
	Unnötiger Aufwand	1	1	2
	Offensichtlicher Mehrwert	4	5	4
A4	Schwer Verständlich	1	1	1
	Einfache Lösung	5	5	5
	Lösung Wiederverwenden	4	3	5
	Unnötiger Aufwand	1	1	1
	Offensichtlicher Mehrwert	4	4	5

## Anhang H: Offene Antworten

**A:** „Es war grundsätzlich nachvollziehbar, was getan werden sollte, ich hätte es viel leicht noch praktisch gefunden, wenn die Aufgabe ganz kurz das Anwendungsziel definiert hätte, weil damit die Einordnung, ob die Lösung für weitere Male sinnvoll ist oder nicht dadurch machbar wird. Viel Erfolg für die Masterarbeit :)“

**B:** „Es ist schwer den Mehrwert von den verschiedenen Ansätzen zu benennen, wenn man bisher keine Ansätze der Umsetzung kannte. In den Aufgabenstellungen wäre es schön gewesen, wenn die Bezeichnungen entweder auf deutsch oder englisch gewesen wären, nicht aber wechselnd zwischen den Aufgaben. Mir hat sich nicht immer erschlossen, was das ausführen einer Aufgabe bringt/mit Accessibility zu tun hat, v.a. in Hinsicht auf die txt Datei. Einige Aufgaben/gewünschten Vorgehensweisen hätten genauer ausformuliert bzw. präziser sein können, damit sie auch für Laien nachvollziehbar sind.“

**C:** „Ein paar Bilder hätten bei den Aufgaben nicht geschadet, um diese besser verstehen zu können. Zudem wäre es Kontext für die Aufgaben gut gewesen, um die anschließende Bewertung besser treffen zu können. Da mir nicht immer klar war, warum ich gewisse Sachen gemacht habe, unter anderem die Informationen als README und noch mal im Programm angeben.“

**D:** „Godot ist fucking umständlich zu bedienen...“



# Anhang I: Keybinding Plugin

The code in this plugin is mostly copied from: [https://github.com/godotengine/godot-demo-projects/tree/master/gui/input\\_mapping](https://github.com/godotengine/godot-demo-projects/tree/master/gui/input_mapping).

The keymap will be saved to the user directory.

- 1 Win: %APPDATA%/<Project Name>
- 2 Linux, macOS: ~/.local/share/godot/app\_userdata/<Project Name>

## Change Keybindings

### Setup

- implement your own keybinding menu (addons/keybinding\_menu can be used as a reference)
- the translation file (addons/keybindings/localisation/keybind\_menu\_strings.en.translation) for the example menu has to be added manually to work (Project Settings > Translations)

### Note

- the Directory of the keymap might not be the project name, if a custom user dir was set
  - Advanced Settings must be enabled to view „Use Custom User Dir“
  - Project Settings > General > Application > Config > Use Custom User Dir
- right now this plugin only cares about the primary binding and ignores the rest
- potentially it might remove additional bindings
  - has to be checked

## Change Sensitivity

The horizontal and vertical mouse sensitivity gets saved in a custom resource (resources/input\_settings\_data). Additional settings can be added by customizing resources/input\_settings.gd.

The sensitivity settings can be used by implementing an export variable. An example of the usage is given in Examples/input\_settings\_usage\_example.gd.

The sensitivity can be used similar to the example from the Godot FPS Tutorial ([https://docs.godotengine.org/en/3.3/tutorials/3d/fps\\_tutorial/part\\_one.html](https://docs.godotengine.org/en/3.3/tutorials/3d/fps_tutorial/part_one.html)). `MOUSE_SENSITIVITY` would be `settings.mouse_horizontal_sensitivity` in the implementation in this project.

```
1 func input(event):
2     if event is InputEventMouseMotion and Input.get mouse mode() == Input.MOUSE_MODE_CAPTURED:
3         rotation helper.rotate x(deg2rad(event.relative.y * MOUSE_SENSITIVITY))
4         self.rotate y(deg2rad(event.relative.x * MOUSE_SENSITIVITY * 1))
5
6     var camera rot = rotation helper.rotation degrees
7     camera rot.x = clamp(camera rot.x, -70, 70)
8     rotation helper.rotation degrees = camera rot
```

## Eidesstattliche Erklärung

Hiermit versichere ich Milas Marco Baumgarten an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 20. April 2023

Ort, Datum

A solid black rectangular box used to redact the signature of the author.

Milas Marco Baumgarten, B.Sc.