

Konzeption, Implementierung und Evaluation von Verfahren zur Eingabe von Text in der Virtuellen Realität

Masterarbeit
im Studiengang Intelligente Systeme
an der Technischen Fakultät
der Universität Bielefeld

16. Oktober 2018

vorgelegt von
Dominik Stallmann

Betreuer:
Dr. rer. nat. Thies Pfeiffer
CITEC, Technische Fakultät, Universität Bielefeld
Inspiration 1
33619 Bielefeld

M. Sc. Patrick Renner
CITEC, Technische Fakultät, Universität Bielefeld
Inspiration 1
33619 Bielefeld

Dominik Stallmann, Schongauerstraße 48a, 33615 Bielefeld

Abstract

Dominik Stallmann

Thema der Arbeit

Konzeption, Implementierung und Evaluation von Verfahren zur Eingabe von Text in der Virtuellen Realität

Stichworte

VR, Virtual Reality, Virtuelle Realität, Texteingabe, Sprachverarbeitung, Autocompletion, Autokorrektur, Head-Mounted Display

Kurzzusammenfassung

Diese Arbeit thematisiert die Entwicklung einer Reihe von Verfahren zur Eingabe von Texten zur Nutzung in der Virtuellen Realität mit Head-Mounted-Displays. Die Verfahren werden um Assistenzsysteme erweitert und gemeinsam mit diesen konzeptioniert und implementiert. Anschließend werden die Verfahren aufgrund der Auswertung von zwei Studien bewertet, um so das Verfahren mit der besten Eignung für die Virtuelle Realität zu bestimmen.

Dominik Stallmann

Title of the paper

Conceptualization, Implementation and Evaluation of methods for text-entry in the virtual reality

Keywords

VR, Virtual Reality, text input, text entry, speech recognition, autocompletion, head-mounted display

Abstract

This thesis deals with the development of a set of methods for text input for use in the Virtual Reality with head-mounted displays. The methods will be extended with assistance systems and will be conceptualized and implemented together with them. Subsequently, the methods are evaluated based on two studies to determine the best method for usage in the Virtual Reality.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis	IX
Abkürzungsverzeichnis	XI
Danksagung	XIII
1 Einleitung	1
1.1 Problemdarstellung	1
1.2 Motivation	2
1.3 Zielsetzung	2
1.4 Lösungsansatz	3
1.5 Gliederung	3
2 Stand der Forschung / Stand der Technik	5
2.1 Stand der Technik	5
2.1.1 Hardware	5
2.1.2 Softwareumgebung	7
2.1.3 Levenshtein-Distanz	7
2.2 Stand der Forschung	8
2.2.1 Gazing	8
2.2.2 Kontinuierliches Zoomen	9
2.2.3 Reale Tastaturen	10
2.2.4 Handschrift	11
2.2.5 Virtuelle Tastaturen	13
2.3 Zusammenfassung	14
3 Konzeption	15
3.1 Kriterien	15
3.2 Assistenzsysteme	15
3.2.1 Sprachverarbeitung	15
3.2.2 Wortvorschläge	16
3.2.3 Wortvorhersagen	16
3.2.4 Textunterstützungen	17
3.3 Thumbpad-Konzepte	18
3.3.1 Thumbpad-Tastatur	19
3.3.2 Thumbpad-Kreise	21
3.3.3 Thumbpad-Clusters	23
3.3.4 Thumbpad-Gesten	24
3.4 Controller-Konzepte	26

3.4.1	Touch-Verfahren	27
3.4.2	Drumstick-Verfahren	29
3.4.3	Laser-Verfahren	30
3.5	Gaze-Verfahren	31
3.6	Handschrift / Gesten	32
3.7	Zusammenfassung	33
4	Implementierung	35
4.1	Unity-Assets	35
4.2	Assistenzsysteme	36
4.2.1	Speech Recognition	36
4.2.2	Wortvorschläge	37
4.2.3	Textunterstützungen	40
4.3	Thumbpad-Verfahren	40
4.3.1	Thumbpad-Tastatur	41
4.3.2	Thumbpad-Kreise	42
4.3.3	Thumbpad-Clusters	43
4.3.4	Thumbpad-Gesten	44
4.4	Handschrift / Gesten	45
4.5	Controller-Verfahren	46
4.5.1	Touch	47
4.5.2	Drumstick	48
4.5.3	Laser	49
4.6	Gaze	50
4.7	Trainingsvorbereitung	51
4.8	Diktate	52
4.9	Logging	53
4.10	Installation, Nutzerdokumentation	53
4.11	Zusammenfassung	55
5	Evaluation	57
5.1	Pseudonymisierung	57
5.2	Fehlerquote	57
5.3	Pilotstudie	58
5.3.1	Ergebnisse	58
5.3.2	Auswertung Pilotfragebogen	59
5.4	Hauptstudie	61
5.4.1	Ergebnisse	62
5.4.2	Auswertung Hauptfragebogen	64
5.4.3	Auswertung TLX	65
5.5	Zusammenfassung	67
6	Diskussion	69
6.1	Ausblick	69
6.2	Zusammenfassung	70
	Literaturverzeichnis	71
	Anhang	77
	Logarithmische Regression	77

Diagramme	79
A.1 Pilotstudie	79
A.2 Hauptstudie	83
Fragebögen	89
A.3 Original	89
A.3.1 Pilotstudie	89
A.3.2 Hauptstudie	93
A.4 Allgemeine Fragen	98
A.5 Pilotstudie	99
A.6 Hauptstudie	109
A.6.1 NASA-TLX Fragebogen	114

Abbildungsverzeichnis

2.1	Darstellung einer HTC Vive	5
2.2	Darstellung eines HTC Vive Controllers	6
2.3	Darstellung einer Levenshtein-Matrix	8
2.4	Darstellung einer Eye-Tracking Software	9
2.5	Darstellung des Dasher-Verfahrens	10
2.6	Darstellung minimalistischer Hände auf einer virtuellen Tastatur	11
2.7	Darstellung realistischer Hände auf einer virtuellen Tastatur	11
2.8	Darstellung des Virtual Notepads	12
2.9	Darstellung des Lasers auf Buchstabenfelder	12
2.10	Darstellung der Marionetten-Methode	13
2.11	Darstellung der Verfahren auf einer virtuellen Tastatur	14
3.1	Konzeptdarstellung der Thumbpad-Tastatur	20
3.2	NEO-Tastenlayout	22
3.3	Konzeptdarstellung der Thumbpad-Kreise	23
3.4	Nachbildung der Gamepad-Tastatur von Steam	24
3.5	Konzeptdarstellung der Tastatur der Thumbpad-Gesten	25
3.6	Konzeptdarstellung der virtuellen Tastatur	27
3.7	Konzeptdarstellung der Touch-Methode	28
3.8	Konzeptdarstellung der Drumstick-Methode	30
3.9	Konzeptdarstellung der Laser-Methode	31
4.1	Darstellung der Thumbpad-Tastatur	41
4.2	Darstellung der Thumbpad-Kreise	42
4.3	Darstellung der Thumbpad-Cluster	43
4.4	Darstellung der Thumbpad-Gesten	45
4.5	Darstellung des Handschrift-Verfahrens	46
4.6	Darstellung des Touch-Verfahrens	48
4.7	Darstellung des Drumstick-Verfahrens	49
4.8	Darstellung des Laser-Verfahrens	50
4.9	Darstellung des Gaze-Verfahrens	51
A.1	Ergebnisse der Cluster-Methode der Pilotstudie	79
A.2	Ergebnisse der Gaze-Methode der Pilotstudie	79
A.3	Ergebnisse der Gesten-Methode der Pilotstudie	80
A.4	Ergebnisse der Laser-Methode der Pilotstudie	80
A.5	Ergebnisse der Drumstick-Methode der Pilotstudie	81
A.6	Ergebnisse der Thumbpad-Tastatur-Methode der Pilotstudie	81
A.7	Ergebnisse der Touch-Methode der Pilotstudie	82
A.8	Ergebnisse der Kreise-Methode der Pilotstudie	82

A.9	Vergleich der Verfahren hinsichtlich der Tastenanschläge	83
A.10	Tastenanschläge mit Drumstick-Verfahren	83
A.11	Tastenanschläge mit Thumbpad-Tastatur-Verfahren	84
A.12	Textlänge mit Drumstick-Verfahren	84
A.13	Textlänge mit Thumbpad-Tastatur-Verfahren	85
A.14	Fehlerquote mit Drumstick-Verfahren	85
A.15	Fehlerquote mit Thumbpad-Tastatur-Verfahren	86
A.16	Funktionskurve der Tastenanschläge des Drumstick-Verfahrens	86
A.17	Residuen der Funktionskurve des Drumstick-Verfahrens	87
A.18	Funktionskurve der Tastenanschläge des Thumbpad-Verfahrens	87
A.19	Residuen der Funktionskurve des Thumbpad-Verfahrens	87
A.20	Fragebogen zur Pilotstudie Seite 1	89
A.21	Fragebogen zur Pilotstudie Seite 2	90
A.22	Fragebogen zur Pilotstudie Seite 3	91
A.23	Fragebogen zur Pilotstudie Seite 4	92
A.24	Fragebogen zur Hauptstudie Seite 1	93
A.25	Fragebogen zur Hauptstudie Seite 2	94
A.26	Fragebogen zur Hauptstudie Seite 3	95
A.27	Fragebogen zur Hauptstudie Seite 4	96
A.28	Fragebogen zur Hauptstudie Seite 5	97

Tabellenverzeichnis

5.1	Übersicht der Tastenanschläge in der Pilotstudie	58
5.2	Übersicht der Textlängen in der Pilotstudie	59
5.3	Übersicht der Hilfestellung durch Wortvorschläge in der Hauptstudie	63
5.4	Übersicht der Standardabweichungen beim Thumbpad-Verfahren in der Hauptstudie	64
5.5	Übersicht der Standardabweichungen beim Drumstick-Verfahren in der Hauptstudie	64
A.1	Frage zur Gewohnheit des Schreibens auf Tastaturen	98
A.2	Frage zur Gewohnheit des Schreibens auf Mobilgeräten	98
A.3	Frage zur Gewohnheit an die virtuelle Realität	98
A.4	Frage zur Gewohnheit des Schreibens in der Virtuellen Realität	99
A.5	Frage zum Erlernen der Bedienung in der Pilotstudie	99
A.6	Frage zu Vor- & Nachteilen der Thumbpad-Verfahren in der Pilotstudie ..	100
A.7	Frage zu Vor- & Nachteilen der Controller-basierten Verfahren in der Pilotstudie	101
A.8	Frage zu Verbesserungsvorschlägen in der Pilotstudie	102
A.9	Frage zur positiven Verfahrensauswahl in der Pilotstudie	103
A.10	Frage zur negativen Verfahrensauswahl in der Pilotstudie	103
A.11	Frage zur Fehlerquote in der Pilotstudie	104
A.12	Frage zur Schreibgeschwindigkeit in der Pilotstudie	104
A.13	Frage zur technischen Beeinträchtigung in der Pilotstudie	104
A.14	Frage zu Gründen der technischen Beeinträchtigung in der Pilotstudie	105
A.15	Frage zur geistigen Anforderung in der Pilotstudie	105
A.16	Frage zur körperlichen Anstrengung in der Pilotstudie	106
A.17	Frage zur Erfolgseinschätzung in der Pilotstudie	106
A.18	Frage zur Frustration in der Pilotstudie	107
A.19	Frage zur Präferenz für die Zukunft in der Pilotstudie	107
A.20	Frage zu negativen Auffälligkeiten in der Pilotstudie	108
A.21	Frage zu sonstigen Anmerkungen in der Pilotstudie	108
A.22	Frage zu Vor- & Nachteilen der Drumstick-Verfahren in der Hauptstudie ..	109
A.23	Frage zu Vor- & Nachteilen der Thumbpad-Verfahren in der Hauptstudie ..	110
A.24	Frage zu Verbesserungsvorschlägen in der Hauptstudie	111
A.25	Frage zur Verfahrensauswahl für kurze Texte in der Hauptstudie	112
A.26	Frage zur Verfahrensauswahl für lange Texte in der Hauptstudie	112
A.27	Frage zur technischen Beeinträchtigung in der Hauptstudie	112
A.28	Frage zur Verbesserung der Geschwindigkeit in der Hauptstudie	113
A.29	Frage zur Fehlerquote in der Hauptstudie	113
A.30	Frage NASA-TLX Geistige Anforderung in der Hauptstudie	114
A.31	Frage NASA-TLX Körperliche Anforderung in der Hauptstudie	114

A.32	Frage NASA-TLX Zeitliche Anforderung in der Hauptstudie	115
A.33	Frage NASA-TLX Leistung in der Hauptstudie	115
A.34	Frage NASA-TLX Anstrengung in der Hauptstudie	116
A.35	Frage NASA-TLX Frustration in der Hauptstudie	116
A.36	Frage zu sonstigen Anmerkungen in der Hauptstudie	116

Abkürzungsverzeichnis

API	Programmierschnittstelle	6
Dump	Auszug eines Speicherinhaltes	37
FOV	Field Of View	27
HMD	Head-Mounted-Display	1
OCR	Optischer Zeichenerkennung	32
SDK	Software Development Kit	6
Vive	HTC Vive	5
VR	Virtuelle Realität	1
WSR	Windows Speech Recognition	36

Danksagung

Ich bin froh, mich im Rahmen einer Masterarbeit genauer mit diesem Thema beschäftigen zu können, und möchte meinen Betreuern Herrn Dr. Thies Pfeiffer und Herrn Patrick Renner dafür danken, dass sie mir bei der Umsetzung dieser Arbeit viel Freiheit gegeben und mich stets mit sehr hilfreichen Beantwortungen meiner Fragen unterstützt haben.

1 Einleitung

Die Bereiche Technik und Informatik haben sich in den letzten Jahren stark verändert und entwickelt. Die Virtuelle Realität (VR) erlebt daher seit wenigen Jahren einen neuen Aufschwung, der durch die höhere Leistung aktueller Computer, besserer Displaytechnologie und günstigerer Sensorik begründet ist. Mit diesen kann im Endverbraucherbereich eine stärkere Immersion erreicht werden als noch vor einigen Jahren.

„What is it like to walk in someone else’s shoes? Books allow us to imagine it, and movies allow us to see it, but VR is the first medium that actually allows us to experience it.“

- Nick Mokey, „We Have Virtual Reality. What’s Next Is Straight Out Of The Matrix“, Digital Trends, December 19, 2016

Die VR gewinnt zunehmend an Bedeutung und kommt bereits sowohl in der Lehre und Ausbildung einiger Berufe vor, wie Walsh (2017), Pantelidis (2009) und Alhalabi (2016) zeigen, als auch im Gesundheitswesen, wie bei Krijn et al. (2004), Simone et al. (2006) und Oxford University Press (2014). Gerade weil die VR so vielseitig als technische Grundlage für eine breite Menge von Anwendungsgebieten genutzt werden kann, stellt sich zunehmend die Frage, inwiefern und im Besonderen **wie** die Menschen in der Zukunft in der virtuellen Welt arbeiten und lernen können. Zur umfangreichen Nutzung der VR ist es in jedem Fall erforderlich, kommunizieren zu können, sei es zur Unterhaltung mit anderen Personen innerhalb der VR, oder auch zur Beantwortung von E-Mails und Kurznachrichten aus der echten Welt. Um dies möglich zu machen, ist vor allem eines nötig, die Eingabe von Text.

1.1 Problemdarstellung

Während sich Tastaturen am Schreibtisch sowie Bildschirmtastaturen am Smartphone längst für die Eingabe von Text durchgesetzt haben, gibt es in der virtuellen Welt noch keine einheitliche Methode, wie Text eingegeben wird. Da real existierende Gegenstände wie Tastaturen für den Nutzer eines Head-Mounted-Displays (HMD) üblicherweise nicht sichtbar und an den Schreibtisch gebunden sind, eignen sich diese im Regelfall nicht für die Texteingabe in der VR, bei der sich der Nutzer möglichst frei im Raum bewegen können sollte und in den Händen typischerweise Controller hält.

Existierende Lösungsansätze sind oftmals nur langsam bedienbar, erfordern eine enorme Eingewöhnungszeit oder setzen besondere Hardware voraus, die nur wenige Benutzer besitzen. Aufgrund dessen und auch wegen des aktuell bestehenden Wechsels zwischen verschiedenen Eingabeverfahren in unterschiedlichen VR-Anwendungen, wird die Etablierung eines einheitlichen Texteingabeverfahrens in der VR verhindert. Aus diesem Grund wird in dieser Arbeit von solchen Lösungsansätzen wie bei Nieuwenhuizen (2017) abgesehen. Stattdessen wird das beste Verfahren gesucht, welches Nutzer unter Verwendung eines typischen HMDs mit dessen zugehörigen Controllern nutzen können und dabei auf die Verwendung zusätzlicher Hardware außerhalb des Endverbraucherbereichs verzichtet.

1.2 Motivation

Da für die umfangreiche Nutzung der VR die Möglichkeit zur Eingabe von Text – sei es zur Beantwortung einer Kurznachricht, oder auch zum Verfassen längerer Texte – unabdingbar ist, sollten alle plausiblen Ideen erprobt werden, auf dass sich schon in naher Zukunft eine einheitliche Methode zur Texteingabe in Anwendungen für die VR etabliert.

Diese Arbeit entsteht im Kontext des Projektes *Immersive-Information-Spaces*, bei dem versucht wird, eine virtuelle Suchumgebung zu erschaffen, die den Weg vom Desktop-Arbeitsplatz zum immersiven Informationsraum zu ebnet (Pfeiffer et al., 2017). Das Projekt entsteht als Kooperation zwischen dem CITEC¹ und der Bibliothek der Universität Bielefeld. Daher ist insbesondere die virtuelle Literaturrecherche ein zielorientierter Anwendungsbereich des Projektes. Konkret soll es möglich sein, wie in einer echten Bibliothek, literarische und wissenschaftliche Werke zu finden und mit diesen zu arbeiten. Dabei ist das Ziel nicht, die Literaturrecherche innerhalb einer Bibliothek zu ersetzen, sondern eine VR-Umgebung zu schaffen, die das Recherchieren funktional erweitert.

Ein elementarer Bestandteil dieser Recherche ist selbstverständlich, dass Suchbegriffe eingegeben werden können, um Literaturergebnisse zu finden. Daher ist die Entwicklung eines Texteingabeverfahrens für das Projekt unabdingbar.

Im größeren Rahmen sollte jedoch, wie schon zu Beginn von Kapitel 1 angesprochen, ein Verfahren entwickelt werden, welches ebenfalls zur einheitlichen Verwendung in vielen weiteren VR-Applikationen eingesetzt werden kann.

1.3 Zielsetzung

Das Ziel der Arbeit ist sowohl konzeptionell neue Eingabeverfahren für die VR zu entwickeln, als auch bereits existierende Eingabeverfahren zu implementieren, durch Assistenzsysteme zu verbessern und schließlich jenes der Verfahren zu bestimmen, welches möglichst einheitlich im HMD-Betrieb verwendet werden kann. Der primäre Gesichtspunkt ist dabei die Geschwindigkeit, mit der Nutzer die Verfahren bedienen können. Sekundäre Kriterien sind unter anderem die Intuitivität und Flexibilität sowie die geistige und körperliche Anstrengung (s. Abschnitt 3.1). Der Fokus soll dabei auf solchen Verfahren liegen, die für die breite Masse von HMD-Benutzern geeignet sind und mittels getrackten Controllern und optional zusätzlich mit einem Mikrofon bedient werden können.

Die Zwischenziele der Arbeit setzen sich daher sequentiell aus den folgenden Aspekten zusammen:

- Z1.** Konzeption geeigneter Eingabeverfahren und Assistenzsysteme
- Z2.** Implementierung der Verfahren und Assistenzsysteme zur Bedienung mit einer HTC Vive
- Z3.** Prozessentwicklung für die nachfolgenden Studien
- Z4.** Vorauswahl aus den Verfahren durch die Durchführung einer Pilotstudie
- Z5.** Anpassung bzw. Optimierung der durch die Vorauswahl bestimmten Verfahren
- Z6.** Anpassung des Studienprozesses für die Hauptstudie
- Z7.** Primärziel: Evaluation der Daten der Hauptstudie zur Ermittlung des besten Verfahrens hinsichtlich der Eingabe- und Textproduktionsgeschwindigkeit

¹<https://www.cit-ec.de/de> Letzter Zugriff: 02.10.2018

1.4 Lösungsansatz

Zur Erreichung des Primärziels soll zunächst eine Vielzahl von gängigen und experimentellen Methoden zur Texteingabe in der VR konzeptioniert und implementiert werden. Diese sollen um die nachfolgenden Assistenzsysteme erweitert werden und anschließend die so entstandenen Verfahren untereinander gegenübergestellt werden.

Eines dieser Assistenzsysteme soll ein Sprachverarbeitungssystem sein, welches über zwei Arten von Funktionen verfügen soll. Zum einen soll eine universelle Sprachverarbeitung dazu dienen, beliebige Texte zu erstellen, zum anderen eine Befehlssteuerung dazu genutzt werden können, Schlüsselbegriffe (Keywords) aufzulösen und so die Bedienungs Oberfläche (UI) der Eingabeverfahren zu steuern.

Ein weiteres Assistenzsystem soll dem Nutzer vollständige Wörter vorschlagen für das Wort, an dem von ihm zu einer gegebenen Situation geschrieben wird. Dieses System soll durch die Kombination von Wortvorhersagen und Wortkorrekturen gebildet werden. Die beiden Assistenzsysteme sollen an die Eingabeverfahren gekoppelt und als Teil der Verfahren mit in den Studien ausgewertet werden.

1.5 Gliederung

Im nachfolgenden Kapitel 2 wird zunächst der Stand der Forschung sowie die verwendeten Techniken beschrieben um einen Einblick dahingehend zu erhalten, welche Grundlagen für diese Arbeit bereits als gegeben angesehen werden können.

Kapitel 3 befasst sich mit den Konzepten der Texteingabeverfahren, sowie mit der Bestimmung der Assistenzsysteme, durch die die Texteingabeverfahren verbessert werden sollen. In Kapitel 4 wird der Entwicklungsvorgang, durch den die Konzepte in die Praxis umgesetzt worden sind, aufgezeigt, sowie Einzelheiten zur Vorbereitung der Durchführung der Studie geschildert.

Im Anschluss werden in Kapitel 5 die beiden durchgeführten Studien beschrieben und deren Ergebnisse sowohl aus technischer Sicht, wie auch hinsichtlich der Antworten auf die Fragebögen ausgewertet.

Abschließend werden in Kapitel 6 die Ergebnisse der Studien diskutiert und ein Ausblick darüber gegeben, welche Optimierungen sich an den Verfahren vornehmen lassen, um dem langfristigen Ziel der Vereinheitlichung der Texteingabeverfahren im HMD-Betrieb näher zu kommen.

2 Stand der Forschung / Stand der Technik

In diesem Kapitel wird zunächst die in der Arbeit verwendete Technik in Abschnitt 2.1 erklärt und daraufhin der Bezug der Arbeit zum aktuellen Stand der Forschung in Abschnitt 2.2 verdeutlicht. Dazu werden entsprechende Vorarbeiten vorgestellt und diskutiert. Abschließend werden in Abschnitt 2.3 die dargelegten Grundlagen in Bezug zu dieser Arbeit gesetzt.

2.1 Stand der Technik

Nachfolgend wird die verwendete Technik beschrieben. Dazu zählt die Beschreibung der Hardware in Unterabschnitt 2.1.1, die Wahl der Softwareumgebung in Unterabschnitt 2.1.2 sowie die Erklärung der Levenshtein-Distanz in Unterabschnitt 2.1.3.

2.1.1 Hardware

Zur Umsetzung des Vorhabens werden mehrere HTC Vive (Vive)¹ Geräte verwendet. Bei diesen handelt es sich um HMDs des Herstellers HTC², deren Entwicklung als Kooperation der Firmen Valve³ und HTC stattgefunden hat. Das Display besteht aus zwei einzelnen OLED-Displays, die mit je 1080×1200 Pixeln auflösen und 90 Bilder pro Sekunde darstellen können. Das Sichtfeld (FOV) beträgt etwa 100 Grad in der Horizontalen und 110 Grad in der Vertikalen. Das Tracking funktioniert über präzises inside-out Laser-Tracking (Okreylos, 2016) mittels zwei Basisstationen die Lighthouses genannt werden, sowie über Gyroskope und Accelerometer. Das Gerät kann sowohl sitzend, wie auch stehend und room-scale (raumfüllend) mit einer Fläche von bis zu etwa 4×4 m genutzt werden und verfügt über ein integriertes Mikrofon.

Abbildung 2.1: Darstellung einer HTC Vive⁴



¹<https://www.vive.com/de/> Letzter Zugriff: 28.09.2018

²<https://www.htc.com/de/> Letzter Zugriff: 28.09.2018

³<https://www.valvesoftware.com/de/> Letzter Zugriff: 28.09.2018

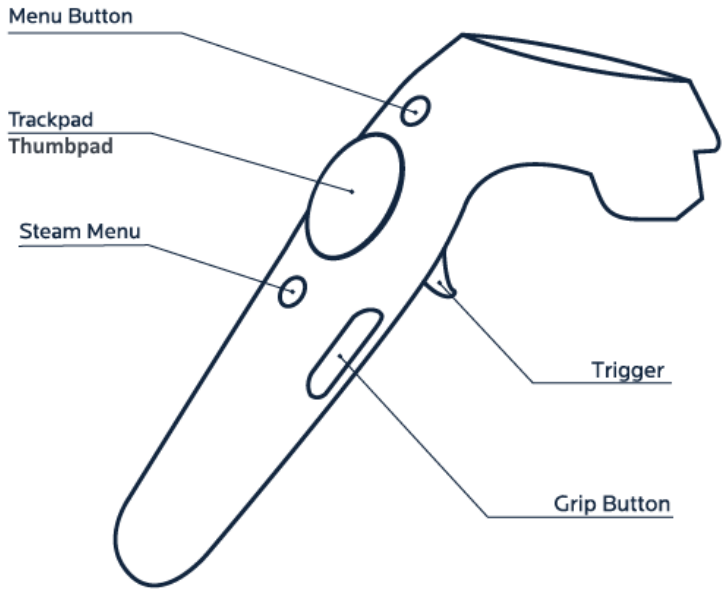
⁴<https://vrreviewz.com/htc-vive-vr-review/> Letzter Zugriff: 28.09.2018

Sowohl das HMD als auch die Controller werden getrackt, d.h. es werden Position, Orientierung und Beschleunigung ermittelt. Die Lighthouses geben jeweils 60 Lichtpulse pro Sekunde und fungieren alternierend, sodass die Geräte mit einer Rate von 120Hz ihren Abstand zu den Lighthouses messen.

Die Software-Schnittstelle bildet das von Valve entwickelte SteamVR⁵. Nativ verwendet werden kann außerdem das Software Development Kit (SDK) OpenVR⁶, welches als Programmierschnittstelle (API) der Unterstützung von HMDs dient und ebenfalls von SteamVR genutzt wird.

In den Händen hält der Benutzer je einen Controller, welcher, wie in Abbildung 2.2 zu sehen, jeweils ein Daumenfeld, nachfolgend Thumbpad oder auch Trackpad genannt, eine analoge Trigger-Taste für den Zeigefinder, zwei Grip-Tasten für Mittel- und ggf. Ringfinger, sowie eine Anwendungs-Taste (Menu Button) und eine System-Taste (Steam Menu) besitzt. Der linke und rechte Grip führen üblicherweise dieselbe Funktion aus, da nur derjenige Grip komfortabel bedient werden kann, welcher an den Fingerspitzen liegt und nicht derjenige im Handballen.

Abbildung 2.2: Darstellung eines HTC Vive Controllers⁷



Das Thumbpad verfügt über insgesamt drei Freiheitsgrade, davon zwei durch die Bewegung des Daumens entlang der Oberfläche und der dritte Freiheitsgrad durch das Drücken des Thumbpads. Die System-Taste wird üblicherweise nicht in Anwendungen von Dritten verwendet, sondern dient durch Gedrückthalten dem An- und Ausschalten der Controller und bei kurzem Drücken der (De-)Aktivierung des SteamVR-Interfaces. Die Anwendungstaste ist bei normaler Handhabung der Controller schlecht zu erreichen und dient innerhalb von Anwendungen typischerweise dafür, Menüs aufzurufen. Effektiv stehen den nachfolgenden Verfahren daher je Controller ein Thumbpad, eine Trigger-Taste und eine Grip-Taste zur Verfügung.

⁵<https://steamcommunity.com/steamvr> Letzter Zugriff: 28.09.2018

⁶<https://github.com/ValveSoftware/openvr> Letzter Zugriff: 28.09.2018

⁷https://survios.com/rawdata/guide/?doing_wp_cron=1538136494.9840700626373291015625

Letzter Zugriff: 28.09.2018

2.1.2 Softwareumgebung

Die Arbeit wird im Kontext des Projektes *Immersive-Information-Spaces* (Pfeiffer et al., 2017) durchgeführt, welches im Jahr 2016 unter Nutzung der Unity Engine⁸ von Unity Technologies⁹ ins Leben gerufen wurde. Die Entwicklungsumgebung dient primär dem Entwickeln von Spielen, doch auch VR-Anwendungen lassen sich mit der Engine erstellen. Als Softwareumgebung wird daher gemäß dem Projekt ebenfalls die Unity Engine verwendet. Das Projekt basiert auf der Version 2018.2.3f1.

2.1.3 Levenshtein-Distanz

Für die in Abschnitt 1.4 erwähnte Vorhersage und Korrektur von Wörtern wird eine fehlertolerante Suche benötigt, die zu einem falsch geschriebenen Wort ebenso wie zu einem unvollständigen Wort eine Klassifizierung gegenüber einer Liste von Wörtern vornehmen kann, um zu prüfen, welche Wörter aus der Liste dem Geschriebenen hinreichend ähneln. Für diese Funktion wird in der Computerlinguistik die sogenannte Levenshtein-Distanz¹⁰ mittels Levenshtein-Algorithmus berechnet. „Der Levenshtein-Algorithmus (auch Edit-Distanz genannt) errechnet die Mindestanzahl von Editierungsoperationen, die notwendig sind, um eine bestimmte Zeichenkette soweit abzuändern, um eine andere bestimmte Zeichenkette zu erhalten.“ (Sulzberger, 2007)

Der Algorithmus funktioniert wie folgt:

Algorithm 1 Levenshtein-Algorithmus

```

1: procedure COMPUTEDISTANCE(a, b)           ▷ a=written text, b=word candidate
2:   n = length(a)
3:   m = length(b)
4:   d = 2D-array[n][m] of 0
5:   for i = 0; i <= n; i++ do
6:     distances[i][0] = i
7:   end for
8:   for j = 0; j <= m; j++ do
9:     distances[0][j] = j
10:  end for
11:  for i = 1; i <= n; i++ do
12:    for j = 1; j <= m; j++ do
13:      c = (b[j-1] == a[i-1])? 0 : 1
14:      d[i][j] = min(min(d[i-1][j]+1, d[i][j-1]+1), d[i-1][j-1]+c)
15:    end for
16:  end for
17:  return d[n][m]                           ▷ The resulting distance
18: end procedure

```

Abbildung 2.3 stellt exemplarisch eine der beiden Möglichkeiten des „günstigsten“ Pfades dar, um von dem Wort „levenshtein“ zu dem Wort „meilenstein“ zu gelangen.

⁸<https://unity3d.com/de/unity> Letzter Zugriff: 28.09.2018

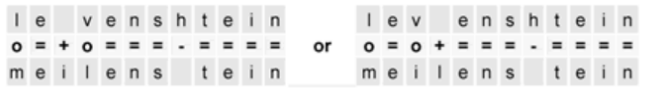
⁹<https://unity3d.com/de/company> Letzter Zugriff: 28.09.2018

¹⁰<http://www.levenshtein.de/> Letzter Zugriff: 29.09.2018

Abbildung 2.3: Darstellung einer Levenshtein-Matrix¹¹

		m	e	i	l	e	n	s	t	e	i	n
l	0	1	2	3	4	5	6	7	8	9	10	11
e	1	1	2	3	3	4	5	6	7	8	9	10
v	2	2	1	2	3	3	4	5	6	7	8	9
e	3	3	2	2	3	4	4	5	6	7	8	9
n	4	4	3	3	3	3	4	5	6	6	7	8
s	5	5	4	4	4	4	3	4	5	6	7	7
t	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
e	8	8	7	7	7	7	6	5	4	5	6	7
i	9	9	8	8	8	8	7	7	6	5	4	5
n	10	10	9	8	9	8	8	7	6	5	4	5
	11	11	10	9	9	9	8	8	7	6	5	4

Es gibt zwei mögliche Wege durch das Tableau, die die geringsten Kosten produzieren. Nämlich



"=" Übereinstimmung; "o" Ersetzen; "+" Einfügen; "-" Löschen

2.2 Stand der Forschung

Nachfolgend werden verschiedene konzeptionelle Ansätze und Lösungen vorgestellt, die im Rahmen der Texteingabe innerhalb und außerhalb der VR in anderen Arbeiten implementiert und evaluiert wurden.

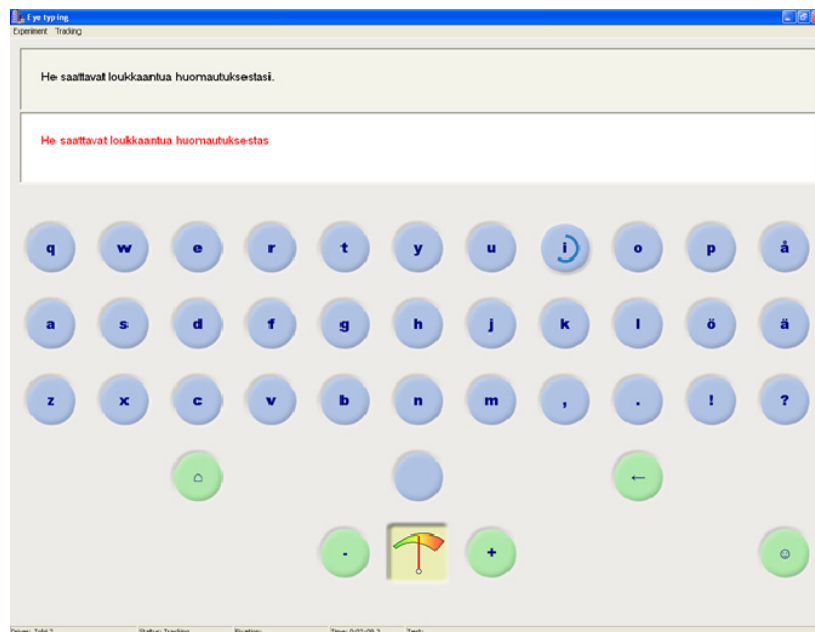
2.2.1 Gazing

Da die verwendete Hardware die Orientierung des Kopfes trackt, bildet sich die Möglichkeit der Gaze-basierten Texteingabe. Diese unterscheidet sich von Eye-Tracking darin, dass die Orientierung des Kopfes als Kriterium für die Selektion von Objekten dient und nicht die Ausrichtung der Augen. In der Literatur wird das Gaze-Tracking teilweise auch als Head-Tracking bezeichnet und das Eye-Tracking stattdessen Gaze-Tracking genannt. In dieser Arbeit entspricht das Gaze-Tracking der Schätzung der Orientierung des Kopfes und das Eye-Tracking der Schätzung der Orientierung der Augen.

Zur erfolgreichen Selektion von Tasten mittels einer Gaze-Interaktion zählt das Anvisieren des Ziels und das Bestätigen der Auswahl durch einen Auslöser (vgl. Huckauf und Urbina (2008)).

So stellen Majaranta et al. (2009) ein Eye-Tracking Verfahren vor, bei dem Nutzer auf die Tasten einer Bildschirmtastatur schauen und nach dem Ablauf einer Schwellzeit (Dwell time) des Starrens auf eine Taste, diese betätigt wird (vgl. Abbildung 2.4). bei Mott et al. (2017) wird dieses Prozedere mit der Besonderheit versehen, dass die Dwell times jeder Taste unterschiedlich und dynamisch sind, je nachdem für wie wahrscheinlich es befunden wird, dass im Kontext des Geschriebenen die jeweilige Taste als nächstes geschrieben werden soll. So werden 10,62 Wörter pro Minute ohne und 12,39 Wörter pro Minute mit der Verwendung der dynamischen dwell time erreicht (Mott et al., 2017). Bei Majaranta et al. (2009) ist die Dwell time hingegen dynamisch von den Nutzern bestimmbar und in der letzten von zehn Trainingseinheiten erreichten die Teilnehmer durchschnittlich 19,9 Wörter pro Minute mit einer Dwell time von 282ms und einer Fehlquote von 0,36%.

¹¹Abbildung aus Sulzberger (2007) Letzter Zugriff: 02.10.2018

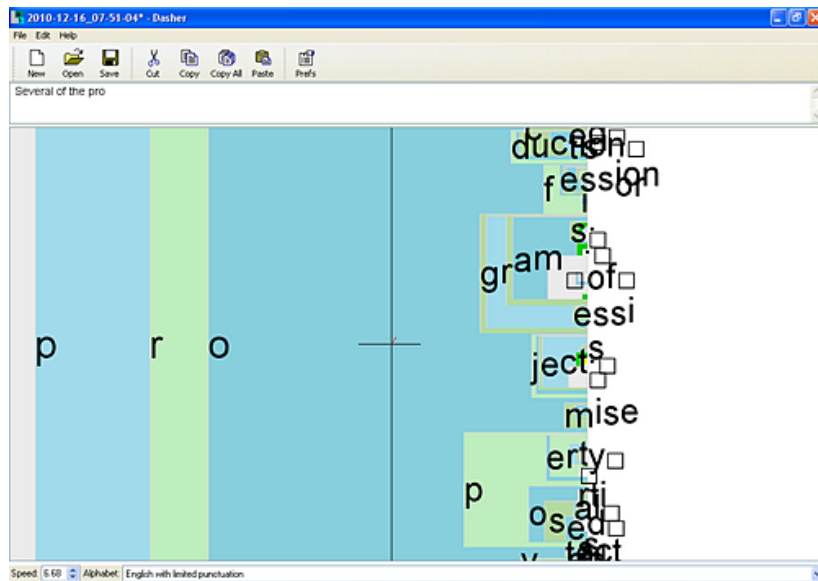
Abbildung 2.4: Darstellung einer Eye-Tracking Software¹²

Die Selektion durch Augen- bzw. Kopforientierung sind jedoch instabil bei Veränderungen in der virtuellen Szene, da Nutzer sich die Umgebung anschauen (Pfeiffer, 2013) und dabei die selektierbaren Objekte nicht bedienen, oder aber sich die selektierbaren Objekte anschauen und dabei diese unfreiwillig bedienen. Letzteres ist durch Jacob (1990) als das Midas-Touch Problem bekannt geworden.

2.2.2 Kontinuierliches Zoomen

Das kontinuierliche Zoomen bildet eine Möglichkeit, das Midas-Touch Problem zu umgehen. Bei diesem Verfahren wird Text geschrieben, indem ein vertikaler Block von Buchstaben und Zeichen von rechts nach links durch das Bild wandert, und der Nutzer durch seine Kopf- bzw. Blickrichtung vorgeben kann, welches Zeichen geschrieben werden soll. Das Eingabesystem namens Dasher realisiert dieses Konzept (Ward und MacKay, 2002) und wird in Abbildung 2.5 dargestellt. Das Verfahren ermöglicht es, ohne die Benutzung der Hände Texte zu schreiben. Einzelne Experten konnten in Ward und MacKay (2002) mit diesem Verfahren bis zu 24 Wörter pro Minute im Durchschnitt schreiben, während normale Benutzer nach einer Trainingsphase durchschnittlich 13 Wörter pro Minute erreichten.

¹²Abbildung aus Majoranta et al. (2009)

Abbildung 2.5: Darstellung des Dasher-Verfahrens¹³

2.2.3 Reale Tastaturen

Ebenso existieren Ansätze, die darauf basieren, weiterhin reale Tastaturen oder Tablets zur Eingabe von Texten zu verwenden. Um die Tastaturen und die eigenen Hände auch in der virtuellen Welt sehen zu können, werden diese bei Grubert et al. (2018) getrackt und im virtuellen Raum dargestellt. Dabei werden entweder minimalistische Repräsentationen der Hände gewählt (vgl. Abbildung 2.6), um nicht durch Falschextrapolationen der tatsächlichen Handpostur den Nutzer abzulenken, oder es wird wie bei Nieuwenhuizen (2017) unter anderem versucht, eine möglichst präzise Erkennung der Handpostur zu erreichen, um eine immersivere Darstellung der Hände zu ermöglichen (s. Abbildung 2.7).

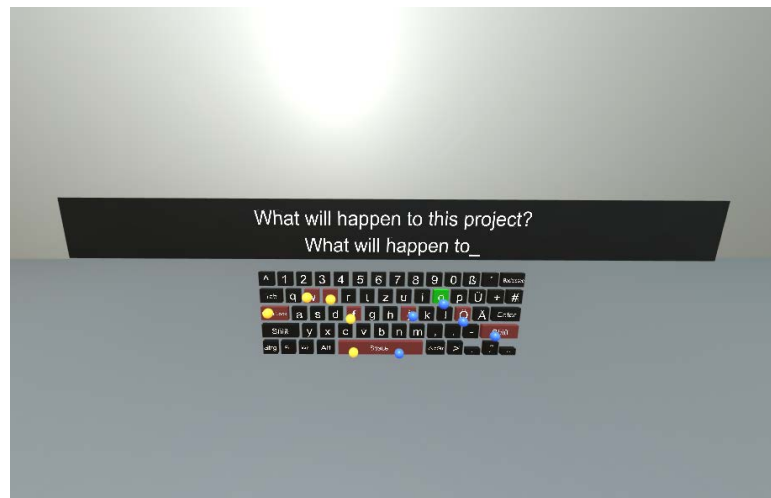
Während bei diesen Verfahren die Position der realen Tastaturen durch Kamerasysteme geschätzt wird, bietet HTC in Kooperation mit Logitech¹⁴ das BridgeSDK¹⁵ an, um die Verwendung von Tastaturen in der VR zu erleichtern und das Tracking der Tastatur über einen HTC Sensor zu ermöglichen.

Nieuwenhuizen (2017) zeigt, dass durch die virtuelle Repräsentation der Tastatur nach ausreichend Eingewöhnungszeit die Eingabegeschwindigkeit gegenüber der Nutzung einer echten Tastatur kaum eingeschränkt wird. Die Teilnehmer erreichten in etwa die Geschwindigkeiten, die sie auch außerhalb der VR auf einer Tastatur erreichten – durchschnittlich 45,92 Wörter pro Minute. Die Verfahren wurden jedoch ausschließlich im Sitzen getestet. Nach Simathamand und Piromsopa (2011), Varcholik et al. (2012) und Hoste und Singer (2014) werden auf realen Tastaturen von erfahrenen Nutzern durchschnittlich etwa 30 Wörter pro Minute erreicht (Gomide et al., 2016).

¹³<http://www.inference.org.uk/dasher/> Letzter Zugriff: 29.09.2018

¹⁴<https://www.logitech.com/de-de> Letzter Zugriff: 02.10.2018

¹⁵https://github.com/Logitech/logi_bridge_sdk Letzter Zugriff: 02.10.2018

Abbildung 2.6: Darstellung minimalistischer Hände auf einer virtuellen Tastatur¹⁶Abbildung 2.7: Darstellung realistischer Hände auf einer virtuellen Tastatur¹⁷

2.2.4 Handschrift

In vielen Publikationen des Bereiches der Texteingabe in der VR wird die Veröffentlichung von Poupyrev et al. (1998) als das Fundament dieses Arbeitsbereiches angesehen, z.B. bei Bowman et al. (2001), MacKenzie und Chang (1999) und Kim (2005). Bei diesem Verfahren trägt der Nutzer in seinen Händen einen Eingabestift und ein Grafiktablett auf dem mit dem Stift geschrieben wird (Abbildung 2.8).

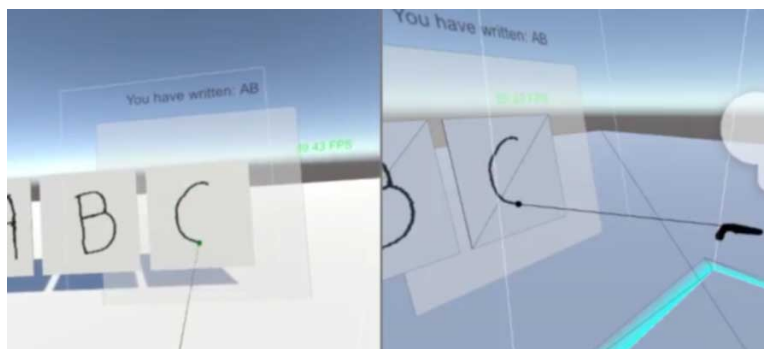
¹⁶Abbildung aus Grubert et al. (2018)

¹⁷Abbildung aus Nieuwenhuizen (2017)

Abbildung 2.8: Darstellung des Virtual Notepads¹⁸

Bei dieser Veröffentlichung wurden keine genauen Angaben zu den erreichten Eingabegeschwindigkeiten gemacht, jedoch greifen Bowman et al. (2002) dieses Verfahren auf und werten es aus. Dabei wurde zum Ende der Trainingsphase durchschnittlich eine Geschwindigkeit von 49,68 Zeichen pro Minute erreicht.

Elmgren (2017) hat den Ansatz verfolgt, mit einem virtuellen Laser einzelne Buchstaben auf virtuelle Felder zu zeichnen (Abbildung 2.9). Dabei mussten die Nutzer den Satz „The quick brown fox jumps over the lazy dog“ schreiben und benötigten im Schnitt bei der zweiten Durchführung der Studie noch 112,28 Sekunden. Somit liegen die durchschnittlich erreichten Zeichen pro Minute bei 22,98.

Abbildung 2.9: Darstellung des Laser auf Buchstabenfelder¹⁹

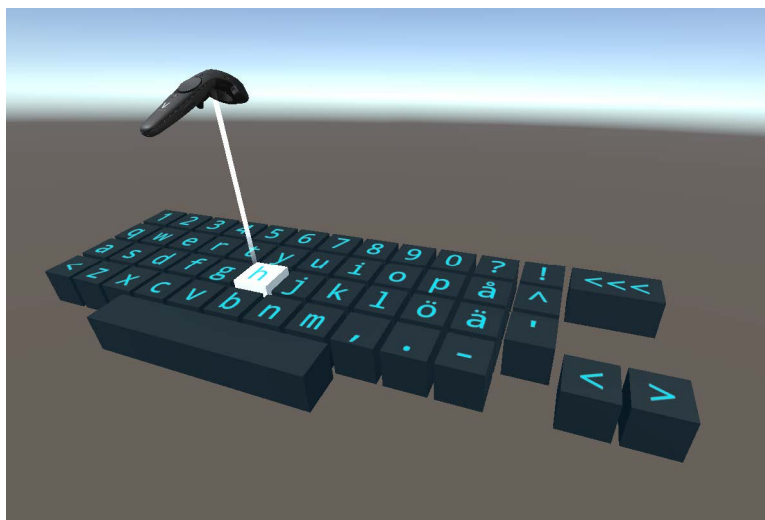
¹⁸Abbildung aus Poupyrev et al. (1998)

¹⁹Abbildung aus Elmgren (2017)

2.2.5 Virtuelle Tastaturen

Einige Lösungsansätze basieren auf dem Schreiben auf einer virtuellen Tastatur. So haben z.B. Olofsson (2017) und Speicher et al. (2018) eine Reihe von Verfahren implementiert, durch die eine virtuelle Tastatur benutzt werden kann. Die Verfahren von Olofsson (2017) setzen sich zusammen aus dem direkten Berühren der virtuellen Tasten und einer sogenannten Marionetten-Methode, bei der – wie in Abbildung 2.10 zu sehen – ein Laser orthogonal zur Griffrichtung des Controllers nach unten zeigt. Außerdem werden eine weitere Laser-Methode, bei der der Laser entlang der Griffrichtung des Controllers gerichtet ist, und eine weitere Methode, die auf der Bedienung der Thumbpads zur Verwendung der virtuellen Tastatur beruht, vorgestellt.

Abbildung 2.10: Darstellung der Marionetten-Methode²⁰



Bei dieser Arbeit wurden lediglich die Zeiten gemessen, die es gedauert hat, zwei vorgegebene Sätze zu schreiben. Diese betragen bei der Thumbpad-Methode 129,58 Sekunden, bei der Touch-Methode 37,83 Sekunden, bei der Marionetten-Methode 37,01 Sekunden und bei der Laser-Methode 33,43 Sekunden.

Ogitani et al. (2018) stellt hingegen eine minimalistische 12-Tasten Tastatur vor, deren Bedienung über Finger-Tracking durch eine Leap-Motion²¹ in jeder Sprache effektiv verwendbar sein soll. Weitere 12-Tasten Tastaturen wurden von MacKenzie et al. (1999) vorgestellt, der für diese Schätzungen für erreichbare Geschwindigkeiten von 22,6 und 44,2 Wörtern pro Minute angibt, jedoch sind die Verfahren nicht speziell für die Verwendung in der VR konzipiert, sondern für Bildschirmstastaturen.

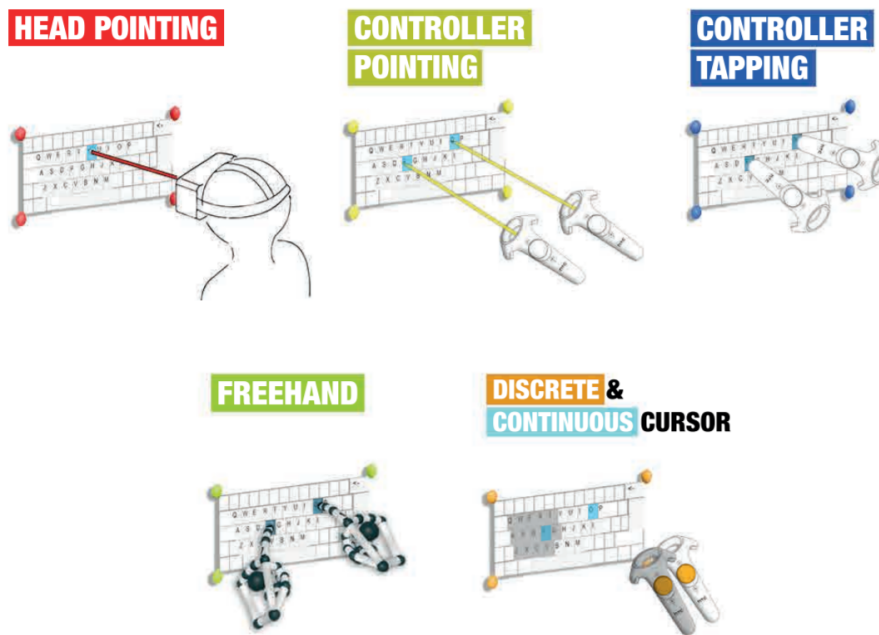
Speicher et al. (2018) stellt über die bereits genannten Verfahren hinaus ebenfalls Konzepte vor, die der Bedienung der Thumbpads zugrunde liegen. Dabei wird jedoch zwischen einer diskreten Position des virtuellen Cursors auf einer der Tasten und der kontinuierlichen Position des Cursors auf der gesamten Tastatur differenziert. Mit dem diskreten Cursor wurden in der Studie von Speicher et al. (2018) durchschnittlich 5,31 Wörter pro Minute geschrieben, während mit dem kontinuierlichen Cursor 8,35 Wörter pro Minute erreicht werden konnten.

²⁰Abbildung aus Olofsson (2017)

²¹<https://www.leapmotion.com/> Letzter Zugriff: 02.10.2018

In dieser Studie kamen außerdem ein Touch-Verfahren, ein Laser-Verfahren, ein Head-Pointing-Verfahren (Gaze-Verfahren) und ein Freihand-Verfahren vor (Abbildung 2.11). Letzteres entspricht dem Touch-Verfahren mit Finger-Tracking. Die Verfahren konnten mit durchschnittlich 12,69 (Touch), 15,44 (Laser), 10,20 (Gaze) und 9,77 (Freihand) Wörtern pro Minute bedient werden.

Abbildung 2.11: Darstellung der Verfahren auf einer virtuellen Tastatur²²



2.3 Zusammenfassung

Zusammenfassend lässt sich sagen, dass bereits eine Vielzahl von Verfahren zur Eingabe von Texten in der VR, als auch solche die außerhalb der VR getestet wurde, sich jedoch durchaus in der VR umsetzen ließen, existiert. Einige der Verfahren setzen zusätzliche Hardware voraus, zeigen sich jedoch vielversprechend hinsichtlich der erreichten Eingabegeschwindigkeiten. Tendenziell erzielen Thumbpad-basierte Lösungen schlechtere Ergebnisse als Verfahren auf einer virtuellen Tastatur.

Die höchsten Schreibgeschwindigkeiten konnten durch die Verwendung von realen Tastaturen erzielt werden, jedoch bringen Verfahren dieser Art eine Reihe von Bedingungen mit sich, die in der typischen Benutzung der VR nicht grundsätzlich gegeben sind, wie z.B., dass die Verfahren auf realen Tastaturen im Sitzen ausgeführt werden müssen, eine getrackte Tastatur voraussetzen oder ein zusätzlicher Eye-Tracker benötigt wird.

Die Auswirkungen von Assistenzsystemen bei dem Schreiben von Texten in der VR sind hingegen noch wenig erforscht und werden in dieser Arbeit genauer behandelt. Im nachfolgenden Kapitel 3 wird insbesondere auf solche Verfahren eingegangen, die mit der Verwendung einer Vive bedienbar sind. In Abschnitt 5.5 werden jedoch alle genannten Verfahren zum Vergleich mit den in diesem Kapitel vorgestellten Methoden herangezogen.

²²Abbildung aus Speicher et al. (2018)

3 Konzeption

In diesem Kapitel werden die Konzepte dargelegt, aufgrund derer die entwickelten Eingabeverfahren umgesetzt werden. In Abschnitt 3.1 werden die Kriterien zur Wahl geeigneter Verfahren ausgearbeitet und nach der Vorstellung der Assistenzsysteme in Abschnitt 3.2 werden in Abschnitt 3.3 Verfahren aufgezeigt, die auf der Benutzung des Thumbpads zur Auswahl von Zeichen beruhen. In Abschnitt 3.4 folgen Verfahrenskonzepte, die mit einer virtuellen Tastatur bedient werden und in Abschnitt 3.6 wird ein Verfahren entwickelt, mit dem Nutzer handschriftlich schreiben können. Die Konzepte werden zum Ende des Kapitels (Abschnitt 3.7) zusammengefasst und bezüglich ihrer jeweiligen Stärken und Schwächen gegenübergestellt.

3.1 Kriterien

In diesem Abschnitt werden die Kriterien bestimmt, anhand derer die Eingabeverfahren ihrer Eignung nach eingestuft werden können. Der wichtigste Betrachtungspunkt ist dabei immer, wie hoch die zu erwartende Schreibgeschwindigkeit des jeweiligen Konzepts ist, jedoch sind auch Faktoren wie die vermutete Fehlerquote (Vertippen) und die erwartete Eingewöhnungszeit, bzw. das Lernverhalten der Umsetzung des Konzepts grundlegende Aspekte der nachfolgenden Gegenüberstellung.

Weitere Kriterien sind die geistige und körperliche Belastung, die Ergonomie und die Eignung zur einhändigen Benutzung.

3.2 Assistenzsysteme

Bevor die Eingabeverfahren im Einzelnen erörtert werden, wird zunächst die Funktionsweise der Assistenzsysteme herausgestellt und die Darstellung, in welcher Form die Systeme in den Verfahren auftreten sollen, diskutiert.

3.2.1 Sprachverarbeitung

Das Sprachverarbeitungssystem soll dem Nutzer die folgenden Funktionen bieten:

- Das Schreiben beliebiger Texte
- Das Editieren zuvor geschriebener Texte
- Das Steuern des Eingabeverfahrens und der Assistenzsysteme

Ein inhärentes Problem der Sprachverarbeitung ist dabei, wie das System erkennen kann, um welche Art von Sprachbefehl es sich bei einem empfangenen Signal handelt. Es ist denkbar, dass der Nutzer die Art des Befehls mit den Controllern während des Sprechens auswählt. Dies würde jedoch die kognitive Belastung erhöhen und einen der Controller blockieren, der nicht nur für das nachfolgende Assistenzsystem benötigt wird, sondern ebenso für die Texteingabeverfahren. Daher wird von einer Zuweisung der Befehlsart durch die Controller abgesehen.

Ebenso sollte es nicht nötig sein vor jedem Befehl auszusprechen, um welche Art von Befehl es sich handelt, wie beispielsweise: „Schreibe: Beispiel“. Stattdessen sollte das System selbst in Lage sein, das Signal in eine der Kategorien zuzuordnen.

Die einzige Ausnahme sollte dabei das Rückgängig machen des letzten Befehls durch z.B. „Befehl, zurück.“ darstellen, sodass der Nutzer mit Sicherheit die Möglichkeit, hat einen Fehler ungeschehen zu machen.

Für die automatische Zuweisung der Befehlsart ist es nötig, eine Sammlung von Steuerbefehlen und Editierbefehlen anzulegen, und den aus dem Audiosignal interpretierten Text mit dieser Sammlung abzugleichen. Liegt der interpretierte Text gemäß der Levenshtein-Distanz hinreichend nah an einem der Befehle, so kann dem assoziierten Befehl entsprechend verfahren werden, anderenfalls soll der Text als Freitext aufgefasst und direkt an die aktuelle Position des Zeigers im Textfeld geschrieben werden.

Die Sprachverarbeitung selbst funktioniert dabei ohne Tasten; jedoch sollte es eine Taste geben, die zum Ein- und Ausschalten der Sprachverarbeitung gedrückt, oder während der Benutzung gedrückt gehalten werden kann. Dafür bietet sich die Menü-Taste des Vive-Controllers an, da diese in den nachfolgenden Verfahren unverwendet bleibt.

3.2.2 Wortvorschläge

Dieses System soll dem Nutzer während des Schreibens eines Wortes Vorschläge anzeigen, mit denen das Wort beendet werden kann. Dabei sollen sowohl Korrekturen vorgeschlagen werden, wie z.B. der Vorschlag „Beispiel“, wenn der Nutzer „Veispiel“ geschrieben hat, wie auch Wortvervollständigungen, wie z.B. „Vervollständigung“, wenn bisher nur „Vervollstä“ geschrieben wurde.

Beide Funktionalitäten lassen sich über dieselbe Mechanik erreichen, die Levenshtein-Distanz, deren Funktionsweise bereits in Unterabschnitt 2.1.3 erklärt wurde. Mithilfe dieses Werkzeuges kann das aktuell geschriebene Wort mit einer Sammlung von deutschsprachigen Wörtern abgeglichen werden, um zu überprüfen, ob ein Wort aus der Sammlung dem geschriebenen Wort hinreichend ähnlich ist. Technische Details hierzu folgen in Kapitel 4, denn zur Umsetzung ist nicht nur eine Liste aller typischen deutschsprachigen Wörter nötig, sondern um die bestmöglichen Vorschläge anzuzeigen muss zusätzlich berücksichtigt werden, wie häufig die hinreichend ähnlichen Wörter in der deutschen Sprache verwendet werden, um von den potentiellen Vorschlägen diejenigen auszuwählen, die mit der höchsten Wahrscheinlichkeit korrekt sind.

Dies kann durch Uni-Gramme erreicht werden, die jedem in der Sammlung auftretenden Wort eine Häufigkeit zuordnen. Ein Eintrag besteht aus einem Wort und einer absoluten Häufigkeit dieses Wortes. Dieses Tupel wird fortan Synonym als Uni-Gramm bezeichnet. Die Sammlung soll nach solchen Wörtern gefiltert werden, die der Eingabe hinreichend ähnlich sind und die drei am häufigsten vorkommenden Wörter sollen vorgeschlagen werden. Die Vorschläge sollen in Form von drei Tasten oberhalb der jeweiligen virtuellen Tastatur der Verfahren angezeigt werden, sodass sie sich als eigenes Element abheben, dennoch im Sichtbereich des Anwenders liegen und schnell auf diese zugegriffen werden kann.

3.2.3 Wortvorhersagen

Eine Erweiterung der Wortvorschläge bildet das System der Wortvorhersagen. Der Unterschied zwischen Wortvorschlägen und Wortvorhersagen ist, dass die Vorhersagen ein nachfolgendes, noch nicht begonnenes Wort vorschlagen. So kommt es z.B. im deutschen Sprachgebrauch derzeit häufig vor, dass nach „Angela“ das Wort „Merkel“ folgt (2-Wort-Kette) und nach „Der zweite“ oft das Wort „Weltkrieg“ folgt (3-Wort-Kette).

Das Assistenzsystem soll so, durch die Hinzunahme kontextueller Informationen, die Unterstützung beim Schreiben verbessern können.

Die Vermutung über das nächste Wort kann dabei schon in dem Moment getroffen werden, in dem entweder der Nutzer selbst ein Leerzeichen schreibt, oder genau dann, wenn einer der Wortvorschläge ausgewählt wird. Theoretisch gibt es auch die Möglichkeit nach einem Punkt oder Komma die Wortvorhersagen einzusetzen, doch ist es komplizierter in diesen Fällen das nachfolgende Wort vorherzusagen, da Wortketten häufig an Satzzeichen enden und eine gute Vorhersage des darauf folgenden Wortes eines allgemeineren Verständnisses des Textes bedarf.

Um diese Funktionalität zu erreichen, werden statt Uni-Gramme höhere N-Gramme benötigt. Ein Bi-Gramm enthält demnach zwei Wörter sowie eine Häufigkeit dieser Wortkette, ein Tri-Gramm stattdessen drei Wörter samt Häufigkeit. Theoretisch können auch noch höhere N-Gramme verwendet werden, jedoch sinkt mit zunehmender Wortkettenlänge die Wahrscheinlichkeit des gewollten Replizierens der Wortketten, da es statistisch unwahrscheinlicher wird, dass Nutzer genau diese Wortkette schreiben möchten.

Konkret sollen bei der Beendigung eines Wortes die Bi- und ggf. Tri-Gramme nach Einträgen gefiltert werden, die mit dem geschriebenen Wort, bzw. den geschriebenen Wörtern anfangen und daraufhin, wie bei den Wortvorschlägen, jene Einträge ausgewählt werden, welche die höchste Häufigkeit besitzen.

Entscheidet sich der Nutzer dazu, statt einem der Vorschläge das nächste Wort selbst zu schreiben, so entsteht so lange eine Phase der Fusion der Wortvorschläge und der Wortvorhersagen, wie das Wortvorhersage-System noch Vorhersagen liefert, die dem bisher Geschriebenen entsprechen.

Dies bedeutet, dass eine Gewichtung dafür gefunden werden muss, in welchen Fällen während des Schreibens eines Wortes das Wortvorhersage-System die in Unterabschnitt 3.2.2 genannten Tasten des Assistenzsystems mit Vorhersagen einnehmen darf, bzw. wie hoch eine Fehlertoleranz sein sollte, bei deren Überschreitung stattdessen ausschließlich Begriffe des Wortvorschlag-Systems nutzbar gemacht werden.

3.2.4 Textunterstützungen

Beim Schreiben von Texten bieten sich einige Gelegenheiten, dem Nutzer unnötige Mühen zu ersparen und die Häufigkeit des störenden Editierens von Texten zu minimieren, wie sie zu Teilen schon von Mobilgerät-Tastaturen bekannt sind.

Zu diesen Unterstützungen zählen:

- Nachdem der Nutzer einen der Wortvorschläge ausgewählt hat, soll auf diesen automatisch ein Leerzeichen folgen und der Zeichencursor hinter das Leerzeichen gesetzt werden.
- Schreibt der Nutzer hinter ein Leerzeichen ein Satzzeichen, dann werden das Leerzeichen und das Satzzeichen automatisch vertauscht. Dies soll unter anderem bei Punkten und Kommata erfolgen, jedoch nicht bei allen Satzzeichen, wie z.B. bei Gedankenstrichen.
- Eine beliebige Stelle in dem Texteingabefeld soll direkt auswählbar sein, z.B. durch Berührung mit dem Controller.
- Die Editierbarkeit des Textes soll durch Cursortasten, welche den Cursor sowohl einzelne Zeichen und ganze Wörter nach links und rechts bewegen, als auch ganz an den Anfang oder an das Ende des Textes springen können, weiter verbessert werden.

- Nach einem Punkt, Ausrufungszeichen und Fragezeichen sollen ausschließlich großgeschriebene Wörter vorgeschlagen werden, jedoch nicht nach gängigen Abkürzungen wie „o.g.“.

Diese Unterstützungen sollen es dem Nutzer sowohl erleichtern die Assistenzsysteme fehlerfrei zu benutzen, als auch Fehler, die beim Schreiben von Texten mit den Verfahren auftreten, möglichst schnell zu korrigieren.

3.3 Thumbpad-Konzepte

Die erste der beiden Hauptgruppen von Konzepten basiert auf der vorwiegenden Nutzung des Thumbpads der Vive-Controller. Diese stellen für die Anwender einen Bezug zu Bildschirmtastaturen her, obgleich sich die Auswahl und die Position der einzelnen Zeichen je nach Konzept unterschiedlich stark von einer typischen Bildschirmtastatur unterscheidet.

Ein Ziel dieser Verfahrensgruppe ist, die körperliche Anstrengung für die Arme zu minimieren und die Verfahren ungehindert nicht nur im Stehen, sondern auch im Sitzen ausführen zu können. Zwar muss zunächst meistens ein ungewohntes Layout der Tasten bedient werden, jedoch bergen durch den Daumen auf einer Touchoberfläche ausgeführte Texteingabeverfahren nach MacKenzie et al. (1999) ein hohes Lernpotential.

Die Controller besitzen außerdem, wie in Abbildung 2.2 dargestellt, weitere Tasten, die in der Verwendung der Konzepte nicht unmittelbar für die Eingabe von Text notwendig sind. Daher kann bei diesen Verfahren die Grip-Taste als Steuertaste benutzt und gefolgt von einer Links-, Mitte- oder Rechts-Auswahl auf dem Thumbpad einer der drei Wortvorschläge genutzt werden. Zusätzlich kann die Trigger-Taste verwendet werden, um schnell zu alternativen Tastenbelegungen zu wechseln und damit effektiv die Anzahl der möglichen Zeichen der Eingabeverfahren zu verdoppeln.

Das Gedrückthalten des Thumbpads bei der Auswahl eines Zeichens löst bei den nachfolgenden Konzepten das Schreiben eines Großbuchstabens bzw. Alternativzeichens, wie z.B. Umlaute, Zahlen, Sonderzeichen oder Satzzeichen aus, indem die Funktionalität der Shift-Taste auf diese Funktionsweise ausgelagert wird.

Ein Nachteil der Verfahrensgruppe ist hingegen, dass bei fast allen der nachfolgenden Verfahren ungewohnte Tastaturlayouts gelernt werden müssen. Es ist daher davon auszugehen, dass durch zunächst eher langsame Schreibgeschwindigkeiten die Frustration während der Anwendung wächst und dadurch die Verfahren insbesondere bei kurzzeitiger Benutzung negativer wahrgenommen werden.

Außerdem sind die Konzepte dieser Gruppe teilweise gegenüber einer realen Tastatur eher unintuitiv, da das Schreiben eines Zeichens bei manchen der Verfahren mehr erfordert, als die Taste isomorph anzusteuern und zu drücken. Dies ist vergleichbar damit, dass es schwieriger ist, auf einer Gitarre einen Akkord zu spielen, als auf einem Klavier, da bei letzterem die Finger direkt zu den Tasten bewegt und diese bedient werden können, während bei einem Akkord auf einer Gitarre weitere Kompetenzen wie die Koordination der Hände notwendig sind.

Darüber hinaus ist zu bedenken, dass Menschen mit unterschiedlichen Handgrößen verschiedene Bereiche des Thumbpads schlechter erreichen können. Bei sehr kleinen Händen bzw. kurzen Daumen ist der dem Daumen am meisten entfernte Bereich nur schwer zu erreichen und bei sehr großen Händen bzw. langen Daumen ist das Gegenteil der Fall und der Bereich des Thumbpads, der dem Daumen am nächsten ist wird schwer erreichbar.

In diesen Fällen kann auch die Bedienung der Grip-Tasten unangenehm sein und nicht am typischen Druckpunkt des Mittel- und ggf. Ringfingers der Hand liegen.

Nachfolgend stellen die Tasten mit der Aufschrift „<“ und „>“ funktional die Pfeiltasten nach links und rechts dar, „Enter“ steht für die Eingabetaste, „Back“ für die Rücktaste und „Space“ für die Leertaste. Für die Verfahren gilt außerdem, dass die Zuweisung der linken und rechten Hand nach Belieben vertauschbar ist, je nach dominanter Hand und der Vorliebe für die Position der Wortvorschläge sowie die der Textzeile.

3.3.1 Thumbpad-Tastatur

Der primäre Aspekt dieses Konzeptes ist es, eine klassische Bildschirmtastatur, wie sie bei Smartphones und Tablets gebräuchlich ist, als Variante der Methode von MacKenzie und Soukoreff (2002) auf die Thumbpads der Controller zu übertragen. Die Thumbpads haben einen Durchmesser von etwa 4 cm. Die Abbildung der gesamten Tastatur auf jeden der beiden Controller wäre suboptimal, da die QWERTZ-Reihe aus mindestens zehn nebeneinanderliegenden Tasten besteht und jede Taste daher nur eine Kantenlänge von 4 mm haben könnte. Diese Größe würde außerdem dadurch weiter reduziert werden, dass die Thumbpads rund sind und auf der Höhe der QWERTZ-Reihe sogar nur etwa 3 mm übrig bleiben.

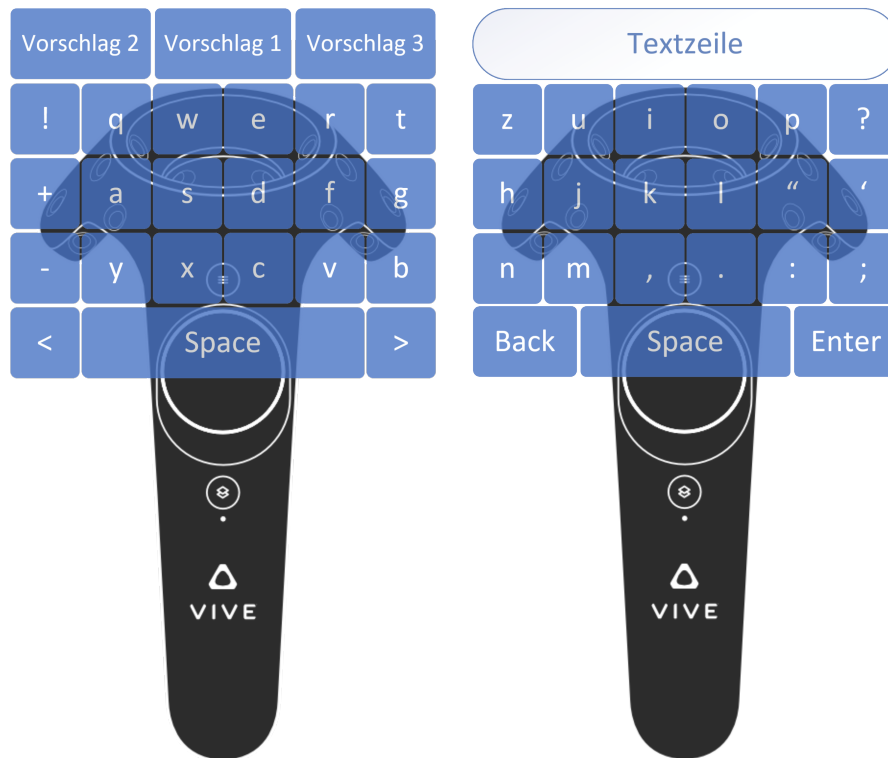
Daher wurde der Entschluss gefasst, ebenso wie in MacKenzie und Soukoreff (2002), die Tastatur in Hälften aufzuteilen. Dabei sollte die Leertaste auf beiden Thumbpads vorhanden sein sollte, sodass diese möglichst schnell mit einer Hand betätigt werden kann, nachdem zuvor der letzte Buchstabe eines Wortes mit der anderen Hand geschrieben wurde.

Außerdem sollten die Funktionstasten „Enter“ und „Back“ größer sein, um mit einer geringeren Fehlerquote betätigt werden zu können. Das rechteckähnliche Format der Tastatur sollte beibehalten werden, um die Benutzung der Tastatur möglichst intuitiv zu halten. Dies hat jedoch zur Folge, dass Ecktasten wie Zirkumflex, Steuerung und die Rücktaste bei einer vollen Ausfüllung der verfügbaren kreisförmigen Thumbpad-Fläche nicht erreichbar wären. Aus diesem Grund wird, wie Abbildung 3.1 zeigt, die Position einiger Tasten angepasst, wobei darauf geachtet wird, dass das QWERTZ-Layout bezüglich der Konsonanten und Vokale beibehalten wird und die Abbildung der Daumenposition auf dem Thumbpad zur Position auf der Tastatur derart ist, dass alle Tasten problemlos erreichbar sind.

Um die rechteckige Form der Tastatur dem Kreis näher zu bringen, können die Funktionstasten „Enter“ und „Back“ auf dem rechten Thumbpad links und rechts neben der Leertaste positioniert werden. Außerdem kann die Zahlenreihe entfallen und deren Zeichen als Teil der alternativen Tastenbelegung beibehalten werden, auf welche mit der Trigger-Taste umgeschaltet werden kann. Dieser Tastenbelegung können zusätzliche Zeichen zugeordnet werden, wie die vier deutschen Umlaute und weitere Sonderzeichen. So kann die Breite der Tastatur insgesamt auf 12 Tasten reduziert werden und ihre Höhe beträgt weiterhin fünf Reihen, da auf einem der Controller die Wortvorschläge auch durch das Thumbpad auswählbar sein sollen und auf dem anderen Controller die Position des Cursors innerhalb der Textzeile mit dem Daumen bestimmt werden können soll.

Da Nutzer bei diesem Verfahren mit der Anordnung der Buchstaben vertraut sind, mit der alternativen Tastenbelegung hingegen nicht, sollte auf den Tasten zusätzlich eine kleine Version der alternativen Belegung jederzeit zu sehen sein. So kann der Nutzer schon während des Umschaltens der Tastenbelegung das gesuchte Zeichen ausfindig machen und wird auch während der Benutzung der Hauptbelegung an die alternative Belegung gewöhnt.

Abbildung 3.1: Konzeptdarstellung der Thumbpad-Tastatur



Insgesamt legt das Verfahren den Schwerpunkt auf die Intuitivität und bietet den Vorteil, dass alle verfügbaren Zeichen auf einen Blick gesehen werden können. Zeichen können mit lediglich einer von Bildschirmtastaturen gewohnten Daumenbewegung gefolgt von kurzem Drücken des Thumbpads geschrieben werden. Ein „Abrutschen“ zu anderen Tasten sollte mit zunehmender Übung nur selten geschehen, da durch ein Cursorobjekt die Referenzposition des Daumens auf der Tastatur angezeigt werden soll, und Nutzer sich daher auf Dauer daran gewöhnen können, die Mitte der Tasten anzusteuern.

Das Verfahren ist in dieser Form nur zweihändig bedienbar. Eine einhändige Variation ist denkbar, jedoch würde dies entweder zusätzliche alternative Belegungen erfordern, was die Navigation zwischen den Belegungen erschwert, oder es müsste die gesamte Tastatur auf nur ein Thumbpad abgebildet werden. Dies würde entweder zu sehr kleinen Tasten, oder zu einer angepassten Variante führen, bei der die Fläche des Thumbpads nur einen Ausschnitt der Tastatur anzeigt und das Positionieren des Daumens an den Rand des Thumbpads den Ausschnitt in die Richtung der Randposition bewegt - entweder kontinuierlich oder diskret.

Das Schreiben von Großbuchstaben sowie Satz- und Sonderzeichen durch Gedrückthalten der Taste auf dem Thumbpad erlaubt es auf der einen Seite, die nötige Menge an Tastenschlägen zu reduzieren, da die Funktionalität der Shift-Taste (Umstell-Taste) abgebildet wird, auf der anderen Seite muss ein zeitlicher Schwellwert (Threshold) festgelegt werden, ab wann diese Funktion einsetzt. Dies hat den Effekt, dass bei einem langen Threshold zwar eine geringere Fehlerquote erwartet, dafür jedoch mehr Zeit in Anspruch genommen wird, während ein kurzer Schwellwert fehleranfälliger ist, jedoch Zeit einsparen kann. Daher sollte es möglich sein, diesen Threshold selbst bestimmen zu können.

3.3.2 Thumbpad-Kreise

Bei diesem Konzept werden die Tasten kreisförmig auf das Thumbpad abgebildet. Anders als bei der Thumbpad-Tastatur liegt die kognitive Herausforderung nicht darin, einem Buchstaben eine zweidimensionale Position, sondern lediglich einen Winkel auf dem Thumbpad zuzuordnen. Die Kreissegmente der Buchstaben sollten dabei an der Spitze zur Mitte des Thumbpads begrenzt sein, sodass eine erhöhte Fehleranfälligkeit durch die eng beieinander liegenden Tasten in dieser Region verhindert wird. So müssen Nutzer den Daumen ausgehend von der Mitte des Thumbpads lediglich in die Richtung einer Taste bewegen, jedoch fernab des Mindestabstandes keinen bestimmten Abstand ansteuern.

Eine Schwierigkeit gegenüber dem Konzept der Thumbpad-Tastatur liegt darin, eine sinnvoll gewählte Anordnung der Zeichen auf den beiden Kreisen zu schaffen. Naheliegende Ideen sind:

1. Tastatur-Hälften, analog zur Thumbpad-Tastatur, auf den Kreisen anzuordnen
2. Die Buchstaben auf den Kreisen in alphabetischer Reihenfolge im Uhrzeigersinn anzuordnen
3. Die Buchstaben so anzuordnen, dass die Controller möglichst oft abwechselnd benutzt werden müssen

Dabei hat (1.) den Nachteil, dass die Zeichen, die sich bei einer Tastatur in der Mitte der abgebildeten Kreise befinden am Rand der Kreise ihren Platz finden müssen und sich dadurch nicht nur selbst an ungewohnten Positionen befinden, sondern auch andere Tasten von ihrer üblichen Position verdrängen würden, um eine gleichverteilte Anordnung der Tasten auf den Kreisen herzustellen.

Bei (2.) ist zu berücksichtigen, dass es bei einer Halbierung des Alphabets gemäß Gartner (2018) bei der deutschen Sprache zu einer vorwiegenden Benutzung von 58,4% des Controllers kommt, der die ersten 13 Buchstaben besitzt, was bei zunehmender Eingabegeschwindigkeit ein größer werdendes Problem darstellen kann. Dem kann jedoch zum Teil entgegengewirkt werden, indem die Funktionstasten so auf die Kreise verteilt werden, dass es näherungsweise zu einer Gleichverteilung der Benutzung der Controller kommt. Den Nutzern ist die alphabetische Reihenfolge im Kontext des Schreibens auf Tastaturen normalerweise nicht geläufig, das Alphabet selbst hingegen sehr wohl. Aus diesem Grund kann angenommen werden, dass während der Übung der einem Buchstaben zugehörige Winkel verhältnismäßig schnell eingepreßt werden kann.

Idee (3.) ist dadurch vielsprechend, dass bei der möglichst oft alternierenden Verwendung der Controller die meiste Zeit mit beiden Händen geschrieben werden kann. Die Idee ist angelehnt an das NEO-Layout¹ (s. Abbildung 3.2). Ebenso verfolgt das primäre Prinzip der Dvorak-Tastenbelegung² diese Idee. Es bleibt zu berücksichtigen, dass diese Layouts insbesondere für die Verwendung von zehn Fingern beim Schreiben konzipiert sind, doch das Konzept des Reduzierens der notwendigen Bewegung zur Optimierung der Geschwindigkeit mit der Schaltflächen benutzt werden können, wurde schon 1954 von Paul Morris Fitts durch Fitts' Gesetz³ etabliert.

¹<https://neo-layout.org> Letzter Zugriff: 28.09.2018

²<http://simon.spruenker.de/dvorak> Letzter Zugriff: 28.09.2018

³<https://m.portal.hogrefe.com/dorsch/fittssches-gesetz/> Letzter Zugriff: 04.10.2018

Abbildung 3.2: NEO-Tastenlayout



Der Nachteil hierbei ist jedoch die hohe Eingewöhnungszeit. Zeichen befinden sich weder an einer gewohnten Position, noch an einer schnell nachvollziehbaren Position. Die einzige Orientierung liefert der Gedanke, immer alternierend auf den Controllern nach dem nächsten Zeichen zu suchen. Obgleich diese Variante für Höchstgeschwindigkeiten bei sehr geübten Schreibern sehr vielversprechend ist, ist sie aufgrund mangelnder Intuitivität für diese Anwendung ungeeignet, wie auch das NEO-Layout auf normalen Tastaturen.

Daher fällt die Entscheidung auf Idee (2.) und Abbildung 3.3 zeigt die Konzeptdarstellung des Verfahrens. Dem linken Controller wird die Leertaste, die Cursorstasten und das Wortvorschlags-System zugeteilt, während der rechte Controller 14 Buchstaben und die Funktionstasten Enter und Back erhält. Die Umlaute und weitere Sonderzeichen befinden sich in der alternativen Tastenbelegung, die erneut über die Trigger-Taste aufrufbar ist. Diese Belegung soll zu einer etwa gleichverteilten Benutzung der Controller führen.

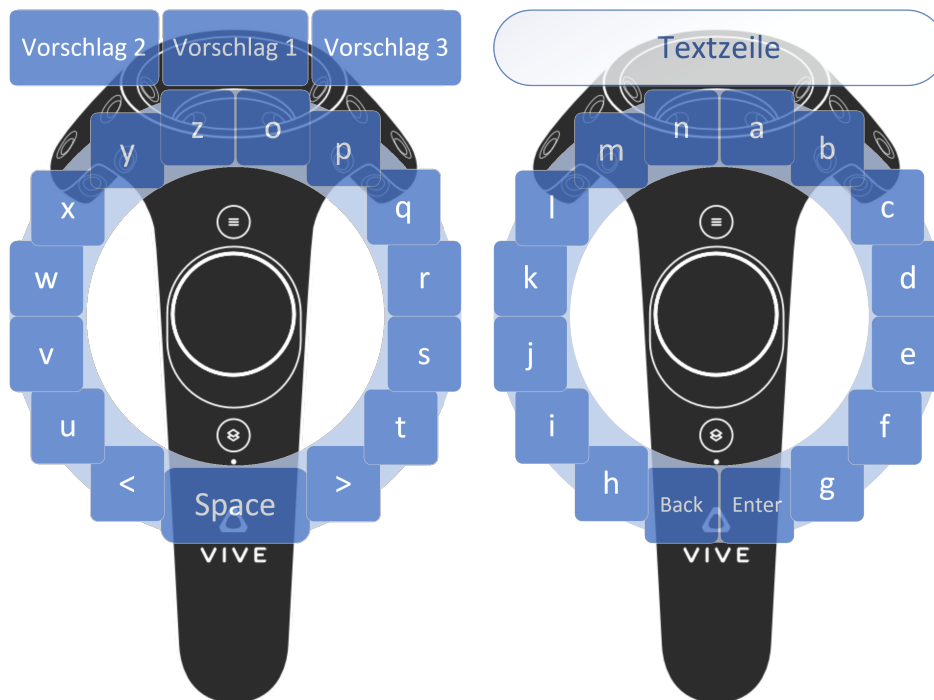
Auch bei diesem Verfahren können alle verfügbaren Zeichen auf einen Blick gesehen werden und aufgrund der zuvor genannten Eigenschaft, dass mit jedem Zeichen lediglich ein Winkel assoziiert werden muss, bietet dieses Verfahren ein erhöhtes Potential dafür, von Experten blind geschrieben werden zu können.

Allerdings ist auch dieses Verfahren nur zweihändig bedienbar. Erneut wäre eine einhändig bedienbare Variation möglich, indem zu zusätzlichen alternativen Belegungen gewechselt werden kann. Dies würde die mögliche Geschwindigkeit jedoch erheblich senken, das Verfahren komplexer machen und in diesem Fall erneut die Frage nach einer optimalen Tastaturbelegung stellen, da der Nutzer möglichst selten zwischen verschiedenen Tastenbelegungen wechseln müsste.

Das Verfahren kann gegenüber dem Abrutschen beim Schreiben eines Großbuchstabens resistenter gemacht werden, indem, sobald das Thumbpad gedrückt wird, der virtuelle Winkelbereich der Taste vergrößert wird. Die Vergrößerung sollte etwa 1/4 der Breite der benachbarten Tasten betragen, sodass es problemlos möglich bleibt, diese zu betätigen, jedoch kleine Verschiebungen des Druckpunktes des Daumens auf dem Thumbpad beim Drücken der Taste nicht mehr zum Abrutschen führen können. Auch dieses Verfahren kann zu Problemen bei zu langen bzw. zu kurzen Daumen bzw. Handgrößen führen.

Ebenfalls tritt die Abwägung der nötigen Dauer für das Gedrückthalten des Thumbpads zum Simulieren der Shift-Taste auf, da auch für dieses Verfahren zwischen potentiell Zeitverlust und einer potentiell erhöhten Fehlerquote entschieden werden muss.

Abbildung 3.3: Konzeptdarstellung der Thumbpad-Kreise

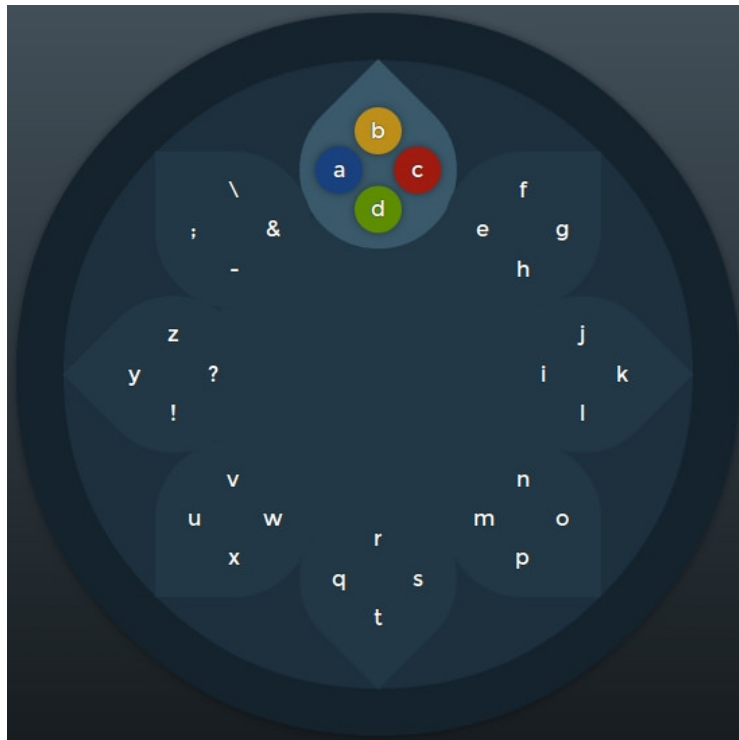


3.3.3 Thumbpad-Clusters

Dieses Konzept beruht auf der Idee, dass die Zeit zur Auswahl und Bestätigung eines Buchstabens reduziert werden kann, indem die Selektion auf beide Hände aufgeteilt wird. Konkret bedeutet dies, dass bei diesem Verfahren mit einem Daumen eine Tastengruppe ausgewählt wird, während mit dem anderen Daumen eine Taste innerhalb der Gruppe selektiert und betätigt wird. Dieses Verfahren kommt typischerweise bei der Verwendung von Controllern und Joysticks vor. Abbildung 3.4 zeigt dies exemplarisch.

Durch diese Funktionsweise ist gegenüber den anderen Verfahren eine geringere Fehlerquote zu erwarten, da bei einer Aufteilung von acht Buchstabengruppen (sogenannten Clustern) und vier Zeichen innerhalb eines Clusters, jedem Zeichen bzw. Cluster mehr Fläche des Thumbpads zur Verfügung steht. Die Fehleranfälligkeit kann zusätzlich dadurch gesenkt werden, dass auf dem Controller, der die vier Zeichen eines Clusters beinhaltet, vom Nutzer während der Wahl des Zeichens innerhalb des Clusters automatisch überprüft wird, ob das richtige Cluster ausgewählt wurde. Durch die geringe Anzahl von Arealen auf den Thumbpads soll zudem die Wahrscheinlichkeit abzurutschen sinken.

Das Verfahren ist in dieser Form nur zweihändig bedienbar, jedoch könnte eine Variation für den einhändigen Betrieb so funktionieren, dass zunächst das Cluster gewählt und durch Drücken bestätigt wird und daraufhin die Auswahl des Buchstabens auf demselben Controller erfolgen kann. Es sollte dann jedoch die Möglichkeit geben, mit der Trigger-Taste zu der Wahl des Clusters zurückzukehren.

Abbildung 3.4: Nachbildung der Gamepad-Tastatur von Steam⁴

Auch mit diesem Verfahren soll es möglich sein, durch die Trigger-Taste zu einer alternativen Tastenbelegung zu wechseln. Außerdem soll das Gedrückthalten der Zeichenselektion innerhalb der Buchstabengruppe zu einem Großbuchstaben führen.

Tendenziell kann es jedoch auch zu Problem führen, dass beide Hände für das Schreiben eines Zeichens notwendig sind, da dies ein erhöhtes Maß an Koordination erfordert. Außerdem muss, solange die Belegung der Tasten auf dem Controller zur Gruppenauswahl nicht verinnerlicht wurde, häufig zwischen den Controllern hin und her geschaut werden. Erst sobald die Gruppenauswahl blind erfolgen kann, reicht es, auf lediglich den Controller zu schauen, der die Wortvorschläge und den eingegebenen Text beinhaltet.

3.3.4 Thumbpad-Gesten

Dieses Konzept entfernt sich von der Idee, eine bestimmte Position auf dem Thumbpad einer bestimmten Taste der Tastatur zuzuordnen. Stattdessen wird bei diesem Verfahren ein Zeichen durch die Bewegung des Daumens zu diesem Zeichen geschrieben und nicht überprüft, ob der Daumen bereits eine bestimmte Zielposition erreicht hat. Das Konzept beruht gegenüber der Handschrift darauf, dass die Zeichen nicht vollständig geschrieben werden müssen, sondern jedem Zeichen ein kurzer, alternativer Bewegungsablauf – eine Geste – zugeordnet wird. Dabei werden acht Gruppen von Zeichen mit je vier Zeichen pro Gruppe um den Controller angeordnet. Ein Zeichen wird immer durch das Bewegen des Daumens in zwei Richtungen und mit fixer Distanz geschrieben.

⁴Bildausschnitt von <http://makegames.com/discussion/2948/text-input-with-a-gamepad> Letzter Zugriff: 28.09.2018

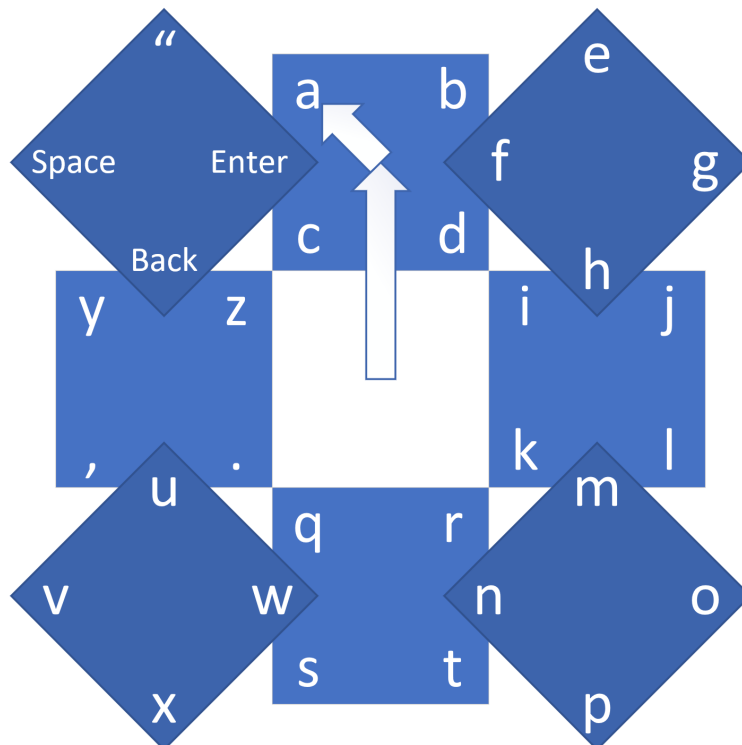
So kann, nachdem der Daumen an einer beliebigen Position auf das Thumbpad gelegt wurde, dieser beispielsweise nach oben bewegt werden, um die dortige Zeichengruppe auszuwählen und wird daraufhin nach oben links bewegt, um denjenigen Buchstaben innerhalb der Gruppe zu wählen, der sich unten links befindet (vgl. Abbildung 3.5). Bei den Gruppen entlang der x- bzw. y-Achse werden die Zeichen innerhalb der Gruppe an den diagonalen Ecken, d.h. oben-links, unten-links, oben-rechts und unten-rechts positioniert, während bei den Gruppen die diagonal liegen gegensätzlich die Buchstaben entlang der x- bzw. y-Achse positioniert werden, sodass niemals das Schreiben eines Zeichens das zweimalige Bewegen in die gleiche Richtung erfordert.

Da für die effektive Anwendung dieses Verfahren letztlich eine alternative Darstellung der möglichen Zeichen gelernt werden muss, ist anzunehmen, dass mit diesem Verfahren vor allem zu Beginn niedrige Eingabegeschwindigkeiten erzielt werden. Das Potential des Verfahrens ist jedoch hoch, da das Thumbpad nur für das Schreiben von Großbuchstaben gedrückt werden muss und das Schreiben von Zeichen allein durch die Bewegung des Daumens erfolgt.

Geübte Schreiber können außerdem die zur Auswahl einer Gruppe (bzw. eines Zeichens) nötige Distanz reduzieren, um ihre Schreibgeschwindigkeit zu erhöhen und nicht für jedes Zeichen erneut in der Mitte des Thumbpads ansetzen zu müssen als auch um mehrere Zeichen in einer einzigen Bewegungskette schreiben zu können.

Abbildung 3.5 zeigt die Richtung, entlang der mit dem Daumen gefahren werden muss.

Abbildung 3.5: Konzeptdarstellung der Tastatur der Thumbpad-Gesten



Je nach Schwerpunkt könnten, wie dargestellt, zugunsten der Intuitivität die Buchstaben in alphabetischer Reihenfolge im Uhrzeigersinn angeordnet werden. Alternativ könnte die Wahrscheinlichkeit erhöht werden, dass nach dem Schreiben eines Zeichens nahtlos das nächste Zeichen geschrieben werden kann. Dafür wäre eine solche Anordnung am besten geeignet, die dafür sorgt, dass auf einen Buchstaben, der den Daumen am Rand des Thumbpads enden lässt, mit hoher Wahrscheinlichkeit ein Buchstabe folgt, der eine derartige Geste erfordert, dass der Daumen vorwiegend in die entgegengesetzte Richtung, also in Richtung der Mitte des Thumbpads, bewegt werden muss.

Dieses Verfahren ist einhändig bedienbar, da durch die Aktivierung der alternativen Belegung mit der Trigger-Taste auf einem Controller bereits $8 \cdot 4 \cdot 2 = 64$ Zeichen zur Verfügung stehen. Für den beidhändigen Betrieb wird diese Anordnung auf den anderen Controller kopiert, sodass der Nutzer sowohl jeden Buchstaben abwechselnd schreiben kann, als auch auf der nicht-dominanten Hand nur Ziffern, Umlaute und Satzzeichen der alternativen Tastenbelegung nutzt, während auf der dominanten Hand alle Buchstaben genutzt werden.

Das Verfahren bietet ein erhöhtes Potential dadurch, dass eine sich zeitlich überschneidende alternierende Benutzung der Controller beim Schreiben von Zeichen möglich ist. Mit einer Hand kann für ein Zeichen bereits die Bewegung zu einer Zeichengruppe durchgeführt werden, während auf dem anderen Controller die Auswahl eines Buchstabens innerhalb der zuvor gewählten Buchstabengruppe erfolgt.

Problematisch bei diesem Verfahren ist jedoch, dass es aufgrund der geringen Intuitivität vor allem zu Beginn schwierig zu lernen ist und daher gegebenenfalls demotivierend auf den Nutzer wirken kann. Außerdem sind zwei Daumenbewegungen pro Zeichen nötig – anders als bei der Thumbpad-Tastatur oder den Thumbpad-Kreisen.

Zwar entfällt in diesem Fall die Wartezeit für die Erfüllung des Thresholds zum Schreiben eines Großbuchstabens, da das Gedrückthalten des Thumbpads während Schreibens eines Buchstabens unmittelbar einen Großbuchstaben erzeugt, jedoch muss entschieden werden, wie lang die geforderten Strecken zur Wahl von Zeichengruppen und Zeichen sind. Je kürzer diese sind, desto anfälliger ist das Verfahren für Fehler; je länger sie sind, desto limitierender ist die Fläche des Thumbpads und die Dauer, die es benötigt, den Daumen längere Strecken zu bewegen.

3.4 Controller-Konzepte

Bei dieser Verfahrensgruppe soll eine Nachbildung einer echten Tastatur als Objekt in der virtuellen Welt existieren. Die nachfolgenden Verfahrenskonzepte liefern Ansätze, wie diese Tastatur bedient werden kann. Fortlaufend werden diese Verfahren auch als Controller-basierte Verfahren bezeichnet.

Für die gesamte Verfahrensgruppe gilt, dass durch das gewohnte Tastaturlayout eine gewisse Vertrautheit zur Eingabemechanik vorhanden ist und weniger nach Zeichen gesucht werden muss, als bei den teilweise alternativen Layouts der Thumbpad-basierten Verfahren. Daher ist außerdem mit einer erhöhten anfänglichen Schreibgeschwindigkeit zu rechnen, die jedoch möglicherweise bereits näher an den Sättigungsgeschwindigkeiten der Nutzer liegt – mehr dazu in Kapitel 5.

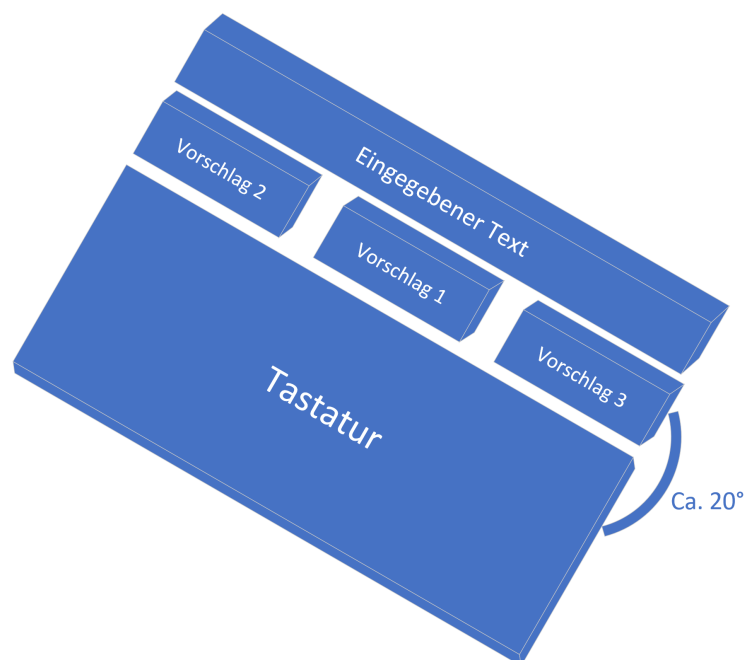
Auch bei diesen Verfahren kann weiterhin die Kombination aus der Grip-Taste und dem Thumbpad zur Auswahl von Wortvorschlägen genutzt werden und je nach Verfahren fallen weitere Funktionen auf die Bedienung der Controller.

Die virtuelle Tastatur ist frei im Raum positionierbar, skalierbar und orientierbar. Einzig die z -Ausrichtung (*roll*-Achse) wird auf 0° festgehalten, d.h. der Nutzer kann die Tastatur beliebig weit zu sich hin und von sich weg neigen, sowie sich mit der Tastatur auf der Stelle drehen, nur sind alle Tasten einer Reihe im virtuellen Raum immer auf derselben Höhe.

Um die Tasten angenehm treffen zu können wird im Nachfolgenden von einer größeren Tastatur ausgegangen, als es bei den Thumbpad-basierten Verfahren der Fall ist. Dadurch wird ein größerer Teil des Field Of View (FOV) eingenommen, was wiederum zur Folge hat, dass bei dem Einsatz der Verfahren in anderen Anwendungen das Eingabeverfahren Elemente der Anwendung verdecken könnte.

Die Konzepte der Verfahren beinhalten Maßnahmen, mit denen diesem Problem entgegen gewirkt werden kann, indem die Tastatur unterhalb des kontextuellen Objektes – z.B. einer Kurznachricht, auf die geantwortet wird – positioniert wird, oder die perspektivische Größe der Tastatur auf andere Arten reduziert werden kann.

Abbildung 3.6: Konzeptdarstellung der virtuellen Tastatur



3.4.1 Touch-Verfahren

Dieses Konzept soll das Schreiben mit zwei Fingern imitieren. Da Controller in den Händen gehalten werden, sollen diese mit einem Bereich ausgestattet werden, der mit den Tasten der Tastatur kollidieren kann und so das Schreiben eines Zeichens auslöst (Collider). Im Fall der Vive-Controller ist es denkbar, dass für diesen Collider das untere Ende der Controller verwendet wird, da mit diesem eine höhere Präzision beim Treffen der Tasten erwartet werden kann als bei dem anderen, größeren Ende. Da die Benutzung der Verfahren jedoch nicht das Drehen der Controller in den Händen bedingen sollen, wird stattdessen das größere Ende der Controller verwendet und an dieses die Collider fixiert.

Die Ausrichtung der Tastatur sollte üblicherweise, ähnlich einer normalen Tastatur, leicht zum Nutzer geneigt sein, sodass dieses Verfahren sowohl mit den Ellenbogengelenken, wie auch zum Teil mit den Handgelenken bedient werden kann, um so die entstehende Belastung aufzuteilen. Aus diesem Grunde sollte ein sichtbarer Bereich an das obere Ende der Controller gesetzt werden, welcher nicht der Griffrichtung der Controller entspricht, sondern der Neigung der Tastatur zugewandt ist. Der Bereich sollte sich zur der Tastatur zugewandten Seite hin verjüngen und daher kegelförmig oder kapselförmig sein, sodass das Treffen von Tasten präzise erfolgen kann (s. Abbildung 3.7).

Da bei diesem Verfahren jede Taste mit beiden Controllern erreichbar ist, kann das Verfahren ohne Einschränkungen einhändig bedient werden. Die Shift-Funktionalität kann sowohl durch das Gedrückthalten der Trigger-Taste erzielt, als auch durch die tatsächliche Shift-Taste der Tastatur verwendet werden. Die Shift-Taste sollte nicht gedrückt gehalten werden müssen, sondern aktivierbar sein und sich beim Schreiben des nächsten Zeichens wieder deaktivieren, wie es bei Mobilgeräten üblich ist. Das Verfahren sollte für Nutzer daher intuitiv bedienbar sein. Außerdem kann die Tastatur weiter unten im Sichtfeld positioniert werden, um mehr FOV für das Kontextobjekt zur Verfügung zu behalten.

Jedoch muss die Tastatur innerhalb der Armlänge des Nutzers positioniert werden, um bedient werden zu können, und nimmt daher ein größeres Sichtfeld als die nachfolgenden Verfahren ein, da auch die Skalierung der Tastatur nicht zu klein sein sollte, um die Fehleranfälligkeit des Verfahrens nicht zu steigern. Zur uneingeschränkten Nutzung sollte das Verfahren außerdem im Stehen ausgeführt werden. Es ist durchaus möglich, das Verfahren auch im Sitzen zu verwenden, jedoch schränkt dies potentiell die Freiheit der Unterarme ein.

Abbildung 3.7: Konzeptdarstellung der Touch-Methode



Im Vergleich zu den Thumbpad-basierten Verfahren ist außerdem mit einer höheren körperlichen Anstrengung zu rechnen, da häufige – und mit zunehmender Geschwindigkeit abruptere – Armbewegungen erforderlich sind. Außerdem kann es zur Kollision der Controller kommen, wenn mit beiden Händen nah beieinanderliegende Tasten getroffen werden sollen. Der Balanceakt bei diesem Verfahren liegt darin, den Nutzer eine für ihn passende Transformation der Tastatur finden zu lassen. Große Skalierungen erhöhen die körperliche Anstrengung, da größere Bewegungen erforderlich sind, während kleinere Skalierungen die Fehleranfälligkeit erhöhen, da die Tasten präziser angesteuert werden müssen.

3.4.2 Drumstick-Verfahren

Dieses Verfahren ist eine Abwandlung des Touch-Verfahrens, bei dem sich an dem oberen Ende der Controller die Collider nicht in unmittelbarer Nähe des Controllers befinden, sondern diese auf Abstand von dem Controller positioniert werden. Um die von den Controllern entfernten Collider besser mit den Controllern zu assoziieren, sollen die Collider wie der Kopf eines Schlagzeugstocks (Drumstick) aussehen und der Stock die Verbindung zwischen dem Controller und dem Collider darstellen – daher die Namensgebung (s. Abbildung 3.8).

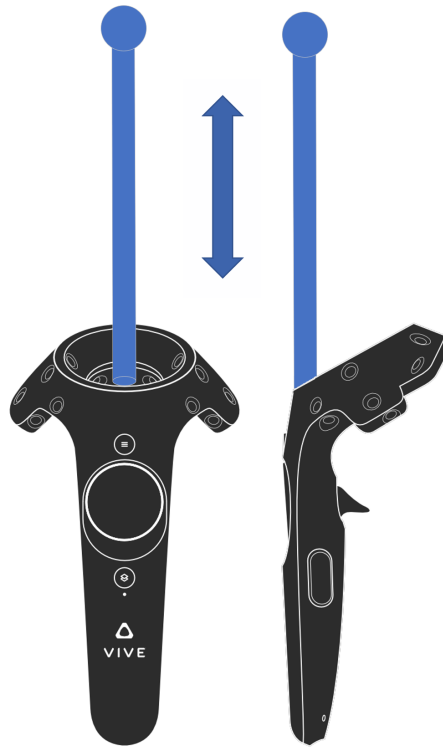
Durch den Abstand der Collider zum Controller soll gegenüber dem Touch-Verfahren zum einen die Kollision der Controller verhindert werden, da zum Bedienen nah beieinanderliegender Tasten die Controller weniger nah zusammengeführt werden müssen, und zum anderen nimmt die Tastatur einen kleineren Bereich des Sichtfeldes ein, ohne dabei zu weit entfernt zu sein, da die Drumsticks die effektive Reichweite der Arme erhöhen. Außerdem kommt es durch die im Vergleich zum Modell der Controller verhältnismäßig dünnen Modelle der Drumsticks zu weniger Verdeckungen der Tasten, als es bei dem Touch-Verfahren der Fall ist.

Das Verfahren ist ebenfalls einhändig bedienbar und die Tastatur kann weiter unten im Sichtfeld positioniert werden. In Kombination mit der erhöhten Distanz vom Nutzer kann diese einen noch kleineren Bereich des Sichtfeldes einnehmen. Die Länge der Drumsticks soll von Nutzern eingestellt werden können, sodass diese sich nach Belieben die entstehende Belastung auf die Unterarme oder die Handgelenke verteilen können. Längere Drumsticks resultieren in vermehrter Nutzung der Handgelenke, während kurze Sticks im Limit äquivalent zu dem Touch-Verfahren sind und daher eine vermehrte Nutzung der Ellenbogengelenke bedeuten.

Auch dieses Verfahren sollte im Stehen ausgeführt werden und auch hier ist mit einer größeren körperlichen Anstrengung zu rechnen, als bei den Thumbpad-basierten Verfahren. Allerdings sollte die Anstrengung geringer sein als bei dem Touch-Verfahren, da kleine Bewegungen der Handgelenke aufgrund der Verlängerung der Arme zu größeren Bewegungen des Drumstick-Kopfes führen.

Es wird angenommen, dass die Bedienung des Verfahrens unintuitiver ist als bei dem Touch-Verfahren, da für Nutzer die Verlängerung der Arme ungewohnt ist. Auch bei diesem Verfahren müssen die passende Ausrichtung und Skalierung der Tastatur gefunden werden, sodass sowohl die Fehleranfälligkeit, wie auch die nötige Bewegung, gering gehalten werden.

Abbildung 3.8: Konzeptdarstellung der Drumstick-Methode



3.4.3 Laser-Verfahren

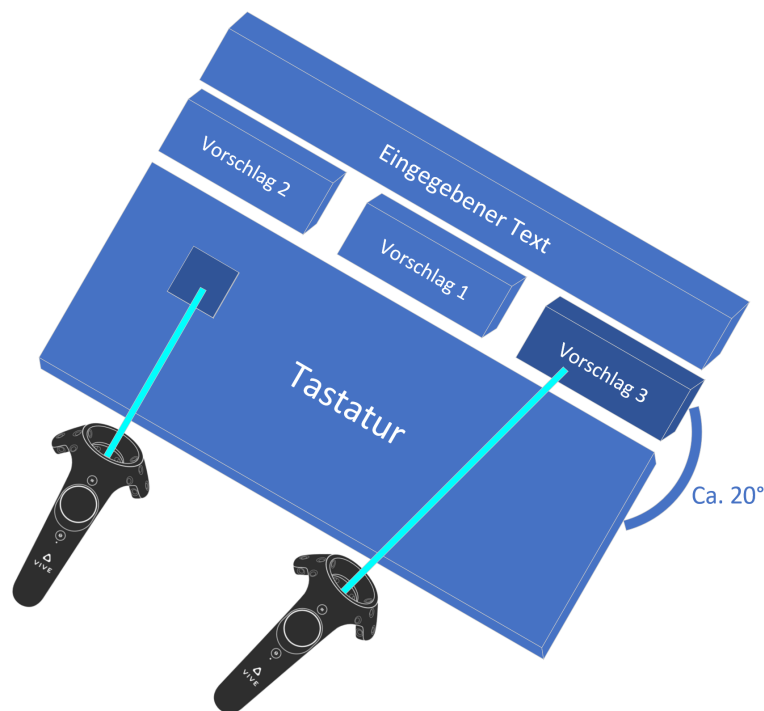
Bei diesem Verfahren können Nutzer Zeichen auswählen, indem sie diese mit einem Laserstrahl selektieren und daraufhin per Trigger-Taste bestätigen können, wie Abbildung 3.9 zeigt. Dabei wird ein Zeichen erst beim Loslassen der Trigger-Taste geschrieben, sodass Nutzer ihre Selektion sehen und bestätigen, oder durch Wegbewegen des Lasers von der Tastatur abbrechen können. Dies liefert gegenüber den anderen Verfahren den Vorteil, dass die Tastatur deutlich weiter vom Nutzer entfernt sein kann und nur kleine Bewegungen notwendig sind, um zwischen den Zeichen der Tastatur zu wechseln, ähnlich dazu, wie wenn mit einem Laserpointer auf einer Leinwand etwas selektiert wird.

Auch dieses Verfahren ist einhändig bedienbar, allerdings kann die Trigger-Taste nicht als Shift-Taste verwendet werden. Da das Drücken des Thumbpads jedoch ungenutzt ist, kann diese Funktionalität auf das Thumbpad gelegt werden. Die Tastatur sollte bei diesem Verfahren dem Nutzer stärker zugeneigt sein, sodass die dem Nutzer zugewandte Fläche jedes Zeichens maximal groß ist und dadurch die Tastatur kleiner skaliert werden kann.

Dadurch nimmt dieses Verfahren tendenziell einen kleineren Bereich des FOV ein. Außerdem wird erwartet, dass die körperliche Belastung dieses Verfahrens geringer ist, da wie auch beim Drumstick-Verfahren sowohl die Ellenbogengelenke als auch die Handgelenke genutzt werden können und für jedes Zeichen nur eine Ausrichtung des Controllers und das Bestätigen mit dem Zeigefinger nötig sind – gegenüber dem Positionieren des Controllers gefolgt von einer ab- und auf-Bewegung. Außerdem kann das Verfahren besser im Sitzen verwendet werden. Dennoch schränkt das Sitzen auch bei diesem Verfahren potentiell die Bewegung der Unterarme ein.

Durch die Auswahl mit der Trigger-Taste kann es jedoch auch zu Fehlselektionen kommen. Dies kann sowohl dadurch passieren, dass das Drücken der Trigger-Taste das obere Ende des Controllers leicht senkt, als auch durch minimale Tracking-Fehler (Jitter), die gemäß des Strahlensatzes aufgrund der größeren Distanz zu der Tastatur bereits zum Verfehlen der Taste führen können. Aus diesem Grund ist wichtig, dass selektierte Tasten farblich markiert werden und Nutzer die genaue Zielposition des Lasers auf den Tasten sehen können, sodass sie selbst Fehlselektionen detektieren und ihnen entgegenwirken können.

Abbildung 3.9: Konzeptdarstellung der Laser-Methode



3.5 Gaze-Verfahren

Wie aus Kapitel 2 hervorgeht, ist es eine verhältnismäßig erprobte Zeichenselektionen durch die Ausrichtung des Kopfes (nachfolgend Gaze-Tracking genannt) bzw. die Ausrichtung der Augen (Eye-Tracking) durchzuführen. Diese Selektion kann dabei auf verschiedene Weisen funktionieren, wie z.B. bei dem bereits in Unterabschnitt 2.2.1 aufgeführten Dasher-Verfahren (Ward und MacKay, 2002). Mit diesem Verfahren erreichte Geschwindigkeiten werden in Kapitel 5 mit den vorgestellten Methoden verglichen.

Stattdessen soll jedoch in dieser Arbeit mit Gaze-Tracking auf der virtuellen Tastatur geschrieben werden, während die Controller für zusätzliche Funktionen genutzt werden können. Das Schreiben eines Zeichens erfolgt durch die Orientierung des Kopfes auf die entsprechende Taste, gefolgt vom Warten auf das Füllen eines Ladekreises, welcher im Fovea-Bereich angezeigt wird. Sobald der Kreis gefüllt ist, wird das Zeichen geschrieben, der Kreis geleert und dieser fängt unmittelbar an, sich erneut zu füllen. Die grundlegende Bedienung sollte daher schnell zu erlernen sein.

Das Verfahren ist theoretisch ohne Controller bedienbar. Diese sollten dennoch genutzt werden, um das Verfahren um weitere Funktionen zu erweitern. Die Geschwindigkeit, mit der sich der Ladekreis füllt, sollte durch die Controller angepasst werden können und könnte außerdem, wie bei Mott et al. (2017) automatisch bestimmt und dynamisch je nach Zeichen angepasst werden, sodass Nutzer je nach Verfassung und Expertise diese erhöhen bzw. senken können. Die Trigger-Taste der Controller soll dazu genutzt werden können, die Auswahl eines Zeichens vorzeitig zu bestätigen, da Nutzer so weitere Zeit bei der Benutzung des Verfahrens einsparen können.

Generell ist eine Variation denkbar, bei der es keinen sich füllenden Ladekreis gibt, sondern alle Tasten durch die Trigger-Taste bestätigt werden. Das Verfahren kann uneingeschränkt im Sitzen verwendet werden, da die Einschränkung der Unterarme weniger relevant ist und die nötigen Ausrichtungen des Kopfes im Sitzen nicht erschwert sind.

Die Rotation des Kopfes könnte auf Dauer jedoch sehr anstrengend sein und sogar zu Nackenschmerzen führen. Insbesondere bei erhöhter Schreibgeschwindigkeit und damit einhergehenden abrupteren Bewegungen des Kopfes könnte es zu diesem Effekt kommen. Bei der Variante des Ladekreises ist außerdem denkbar, dass unabsichtlich etwas geschrieben wird, wenn der Nutzer gerade nachdenkt, sich umschaute oder sich anderweitig beschäftigt und in diesem Moment nicht schreiben möchte. Es sollte daher möglich sein, den Threshold zum Schreiben eines Zeichens so einzustellen, dass ungewollten Zeichen-selektionen nicht länger auftreten und nur noch über die Benutzung der Trigger-Taste ein Zeichen geschrieben wird.

Ein weiteres Problem des Verfahrens ist, dass es nicht "blind" geschrieben werden kann, sondern immer auf die Zeichen, die geschrieben werden sollen, geschaut werden muss und daher ein gleichzeitiges Lesen des Geschriebenen und Fortsetzen des Schreibens nicht möglich ist.

3.6 Handschrift / Gesten

Eine weitere Möglichkeit sich das Tracking der Controller zu Nutze zu machen, ohne unmittelbar auf einer Tastatur zu schreiben, ist es, den Nutzer handschriftlich in die Luft schreiben zu lassen. Dieses Verfahren hat gegenüber allen anderen genannten Verfahren den Vorteil, dass es keine Tastatur benötigt, die das Sichtfeld blockiert. Dennoch sind Steuermöglichkeiten erforderlich um Wörter zu editieren, Vorschläge zu nutzen usw.; daher kommt auch dieses Verfahren nicht ohne eine grafische Benutzeroberfläche aus.

Generell sollte das Verfahren jedoch intuitiv bedienbar sein und sich für den Nutzer ähnlich anfühlen, wie etwas in die Luft zu schreiben. Es gibt jedoch das Problem, dass es bei in die Luft geschriebenen Text verhältnismäßig schwierig ist, eine erfolgreiche Texterkennung durchzuführen, mit Hilfe derer die Handschrift in digitalen Text verwandelt wird.

Aus diesem Grund liegt die Idee nahe, dass die Handschrift auf einer virtuellen zweidimensionalen Fläche durchgeführt wird, sodass die Repräsentation des Geschriebenen auf einem Bild stattfinden kann, welches mittels Optischer Zeichenerkennung (OCR) die Zeichen in dem Text extrahiert.

Diese Fläche kann auf mehrere Arten beschrieben werden. Eine Möglichkeit ist, dem Nutzer einen virtuellen Stift (z.B. Kreide) in die Hand zu legen, mit dem an einer virtuellen Tafel geschrieben werden kann.

Alternativ könnte der Nutzer den Text mit einem Laserstrahl an die Tafel schreiben, indem er während der Bewegung des Controllers mit der Trigger-Taste steuert, ob der Laser auf der Tafel schreiben soll oder nicht. Da dies bei Elmgren (2017) jedoch zu einer langsamen Eingabegeschwindigkeit geführt hat, wird in dieser Arbeit die Stift-Variante weiter verfolgt.

Bei dieser kann der Nutzer als haptisches Feedback jedoch lediglich das Vibrieren des Controllers erfahren, die Hand allerdings an nichts anlegen. Die Benutzung eines realen flachen Objektes auf dem geschrieben werden soll wird nicht näher betrachtet, da die Verfahren, wie in Abschnitt 1.2 beschrieben, ohne zusätzliche Hardware oder Gegenstände, stehend, sowie nach Möglichkeit auch sitzend, genutzt werden können sollen.

Daher kann die Position der Tafel nur abgeschätzt werden und beim Schreiben muss entschieden werden, was passieren soll, wenn sich der Stift etwas von der Tafel entfernt, oder durch diese hindurch sticht. Der Laserstrahl ist hingegen von der Distanz unabhängig und es kann präzise bestimmt werden, wann geschrieben wird.

Das Verfahren ist generell einhändig bedienbar, in der zweiten Hand könnte jedoch ein Werkzeug gehalten werden, mit dem handschriftlich geschriebener Text gelöscht werden kann, wie ein Radiergummi bzw. ein Schwamm. Die Tasten der Controller bleiben dabei größtenteils unbenutzt und können für die Assistenzsysteme, sowie der Steuerung der Tafel bzw. des Whiteboards und das Editierung des Textes genutzt werden.

3.7 Zusammenfassung

Zusammenfassend lässt sich sagen, dass eine breite Vielfalt von Möglichkeiten besteht, wie in der VR Texte geschrieben werden können. Das Verfahren der Thumbpad-Gesten verspricht viel Potential nach dem Überwinden einer steilen Lernkurve, während Verfahren wie die Thumbpad-Tastatur und das Touch-Verfahren auf der virtuellen Tastatur durch eine sehr intuitive Bedienung leicht zu erlernen sein sollten.

Die Thumbpad-Kreise, wie auch die Thumbpad-Cluster, zeigen zur Literatur alternative Möglichkeiten auf, die nach einiger Lernzeit schnell und intuitiv bedienbar sein sollten und das Drumstick-Verfahren kann gegebenenfalls einige der Nachteile des Touch-Verfahrens vermeiden.

Die Gaze- und Laser-Verfahren sind bereits erprobter, allerdings könnten durch die Bedienung der Controller gepaart mit den Assistenzsystemen, um die die Verfahren erweitert werden, potentiell höhere Geschwindigkeiten erreicht werden als bei Speicher et al. (2018), Elmgren (2017) und Majaranta et al. (2009).

Die Sprachverarbeitung bietet einen alternativen Ansatz Texte einzugeben und kann potentiell mit jedem der Verfahren kombiniert werden. Allgemein wird die Erwartung festgehalten, dass die Wortvorschläge und Wortvorhersagen die Textproduktionsgeschwindigkeiten jedes der Verfahren erhöhen können und insbesondere bei den für Fehler anfälligeren Verfahren zu einer schnellen Behebung der Fehler beitragen können.

4 Implementierung

In diesem Kapitel werden Einzelheiten der Umsetzung beschrieben. Darunter fällt auch, welche Probleme aufgetreten sind und welche Lösungen gefunden wurden. Außerdem wird die Eignung der einzelnen Verfahren für die Studie ermittelt.

Wie bereits in Kapitel 2 erwähnt, wird für die Umsetzung der Konzepte die Unity Engine verwendet. Die Verfahren werden in das Projekt *Immersive-Information-Spaces* integriert. Die Basis dieses Projektes wurde von Pfeiffer et al. (2017) erschaffen.

4.1 Unity-Assets

Bei Unity-Assets handelt es sich um fertige bzw. vorbereitete Inhalte von Entwicklern aus dem Unity Asset Store¹. Diese Inhalte können als Grundlage dienen, um schnell in die Kernaufgaben der Arbeit einsteigen zu können sowie um Zeit und Mühen zu sparen, Funktionalitäten zu entwickeln und virtuelle Objekte selbst zu erstellen.

Bei dem virtuellen Raum, in dem die Verfahren ihre Anwendung finden sollen, handelt es sich um das „3D Virtual Shop Vol 2“-Asset² des Entwicklers Barking Dog³. Zur Darstellung von Text, der auch nah vor den Augen der Nutzer scharf aussieht und nicht verpixelt, wird das mittlerweile in die Engine integrierte Asset „TextMesh Pro“⁴ verwendet, welches Signed Distance Fields (SDFs) benutzt, um eine von der Distanz unabhängige Textschärfe zu erzielen.

Die Schnittstelle zur Verwendung der HTC Vive durch SteamVR erfolgt durch das „SteamVR Plugin“⁵ der Valve Cooperation. Die Funktionalität eines Webbrowsers für die VR, welcher für die in Abschnitt 5.3 näher beschriebene Studie wichtig ist, kann durch das Asset „Embedded Browser“⁶ von Zen Fulcrum LLC⁷ erzielt werden.

Um die Implementierung der Verfahren zu beschleunigen wurden außerdem die Assets „Curved VR Keyboard“⁸ von Handcrafted VR⁹, das „VR Keyboard“¹⁰ von Yunhan¹¹,

¹<https://assetstore.unity.com> Letzter Zugriff: 29.09.2018

²<https://assetstore.unity.com/packages/3d/environments/urban/3d-virtual-shop-vol-2-73361>
Letzter Zugriff: 29.09.2018

³<https://assetstore.unity.com/publishers/21180> Letzter Zugriff: 29.09.2018

⁴<https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126>
Letzter Zugriff: 29.09.2018

⁵<https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
Letzter Zugriff: 29.09.2018

⁶<https://assetstore.unity.com/packages/tools/gui/embedded-browser-55459>
Letzter Zugriff: 29.09.2018

⁷<https://assetstore.unity.com/publishers/12287> Letzter Zugriff: 29.09.2018

⁸<https://assetstore.unity.com/packages/tools/input-management/curved-vr-keyboard-77177>
Letzter Zugriff: 29.09.2018

⁹<https://assetstore.unity.com/publishers/25336> Letzter Zugriff: 29.09.2018

¹⁰<https://assetstore.unity.com/packages/tools/input-management/vr-keyboard-95936>
Letzter Zugriff: 29.09.2018

¹¹<https://assetstore.unity.com/publishers/30481> Letzter Zugriff: 29.09.2018

„VRKeys“¹² von The Campire Union¹³ sowie die von außerhalb des Asset-Stores bezogene VR-Tastatur „Punchkeyboard“¹⁴ des Entwickler J. Ravasz in das Projekt importiert und auf Funktionalitäten und Ideen geprüft, die die Umsetzung der Verfahren unterstützen können.

In dem Projekt befinden sich weitere Assets von Drittanbietern, die jedoch nicht zur Umsetzung der Verfahren oder der Durchführung der Studien genutzt werden.

4.2 Assistenzsysteme

Zunächst soll erneut, abstrahiert von den umgesetzten Verfahren, beschrieben werden, wie die Assistenzsysteme implementiert wurden und wie sie ihre Anwendung in den einzelnen Verfahren finden können.

4.2.1 Speech Recognition

Die Sprachverarbeitung wurde in zwei Komponenten implementiert. Zum einen in Form eines sogenannten Dictionary-Recognizers, zum anderen als Universal-Recognizer. Bei dem Dictionary-Recognizer handelt es sich um eine Klasse, die überprüft, ob ein Audiosignal einem der vorgegebenen Sätze (Dictionary) entspricht. Es ist also möglich, Phrasen – typischerweise Kommandos – vorzugeben, welche wiedererkannt werden und daraufhin Funktionen des Programmcodes ausführen können.

Die Zuordnung des Audiosignals zu Einträgen des Dictionaries geschieht über die *KeywordRecognizer*¹⁵-Klasse des *UnityEngine.Windows.Speech* Paketes. Im Gegensatz dazu versucht der Universal-Recognizer beliebige gesprochene Wörter in Audiosignalen zu detektieren. Diese Variante ermöglicht das Eingeben beliebiger Texte, solange diese vom System erkannt werden können. Für diese Funktionalität wird die *DictationRecognizer*¹⁶-Klasse des gleichen Paketes verwendet. Bei diesem handelt es sich um eine Schnittstelle, zu der von Microsoft¹⁷ entwickelten Windows Speech Recognition (WSR)¹⁸.

Die Anwendung wurde unter Windows entwickelt und getestet. Da andere Betriebssysteme wie OS X und Linux nicht auf WSR zugreifen können, schränkt dies die Nutzbarkeit der Anwendung auf solchen Betriebssystemen ein. Als Alternative zu WSR könnte das Sprachverarbeitungssystem von IBM¹⁹ mit der Sprachverarbeitung durch Watson²⁰ fungieren. Falls die Anwendung unter anderen Betriebssystemen als Windows genutzt werden soll, könnte auf dieses System zurückgegriffen werden, da IBM im Asset Store das Asset „IBM Watson SDK for Unity“²¹ mit einer geeigneten Schnittstelle anbietet.

¹²<https://assetstore.unity.com/packages/tools/input-management/vrkeys-99222>

Letzter Zugriff: 29.09.2018

¹³<https://assetstore.unity.com/publishers/31646> Letzter Zugriff: 29.09.2018

¹⁴<https://github.com/rjth/Punchkeyboard> Letzter Zugriff: 29.09.2018

¹⁵<https://docs.unity3d.com/ScriptReference/Windows.Speech.KeywordRecognizer.html>

Letzter Zugriff: 29.09.2018

¹⁶<https://docs.unity3d.com/ScriptReference/Windows.Speech.DictationRecognizer.html>

Letzter Zugriff: 29.09.2018

¹⁷<https://www.microsoft.com/de-de> Letzter Zugriff: 29.09.2018

¹⁸<https://developer.microsoft.com/de-de/windows/speech> Letzter Zugriff: 29.09.2018

¹⁹<https://www.ibm.com/de-de/> Letzter Zugriff: 29.09.2018

²⁰<https://www.ibm.com/watson/> Letzter Zugriff: 29.09.2018

²¹<https://assetstore.unity.com/packages/tools/ai/ibm-watson-sdk-for-unity-108831>

Letzter Zugriff: 29.09.2018

In diesem Fall fiel die Entscheidung jedoch auf WSR, da erste Testversuche deutlich bessere Resultate lieferten. Um die Liste der von WSR angebotenen Recognizer zu vervollständigen, wird außerdem der *GrammarRecognizer*²² erwähnt. Da jedoch keine komplexen, eigenständig entworfene Grammatiken benötigt werden, kommt dieser nicht zum Einsatz.

Da nicht beide Systeme gleichzeitig ausgeführt werden können, findet die Sprachverarbeitung ausschließlich über die Universal-Recognition statt, da das Assistenzsystem durch diesen Recognizer nicht eingeschränkt wird. Auch aus der Universal-Recognition interpretierte Phrasen können für die Suche nach Kommandos mit einem Dictionary abgeglichen werden. Dies geschieht fehlertolerant, da es sonst zu Problemen bei der Benutzung kommen kann, wenn aufgrund von Störgeräuschen, Sprachfehlern oder technischen Problemen schon bei kleinen Abweichungen zwischen gesprochenen Phrasen und vorgegebenen Kommandos kein Bezug hergestellt und der Text stattdessen geschrieben wird.

Daher wird der interpretierte Text über die Levenshtein-Distanz mit Einträgen des Dictionarys abgeglichen. Liegt die Distanz zu einem der Kommandos unterhalb einer Toleranzgrenze, so wird die Phrase als das entsprechende Kommando aufgefasst.

4.2.2 Wortvorschläge

Zur Implementierung des Wortvorschläge-Systems wurden zunächst die Sprachprimitive von Google Books (Google LLC, 2018b) inspiziert. Auf der Webseite werden große Datenbestände von N-Grammen in verschiedenen Sprachen frei zum Download angeboten. Die N-Gramme sind sortiert nach dem Zeichen, mit dem sie beginnen, oder alternativ auch nach den Worttypen (Adjektive, Nomen etc.). Jedoch konnte frühzeitig festgestellt werden, dass die Dauer der Verarbeitung der vorliegenden Menge an Daten zeitlich nicht praktikabel wäre. Eine Gesamtgröße der Uni- und Bi-Gramme wird leider nicht angegeben, kann jedoch durch das Herunterladen einzelner Dateien grob abgeschätzt werden. So ist allein die Sammlung der Wörter, die mit „s“ beginnen, entpackt bereits 1,8 Gigabyte groß.

Hochgerechnet auf alle Buchstaben und unter Hinzunahme aller Kombinationsmöglichkeiten der Bi-Gramme wäre der Bestand über einem Terabyte groß. Bei dieser Schätzung wurde sowohl bedacht, dass bei den Uni-Grammen mit „s“ ein überdurchschnittlich großer Bestand existiert, und auch bedacht, dass die Bi-Gramme aufgrund höherer Vielfalt weniger effizient gespeichert werden können.

Erschwerend kommt hinzu, dass die Bestände nicht nach deutschsprachigen Wörtern gefiltert sind, sondern zu großen Teilen Buchstaben- und Zeichenketten enthalten, die nicht gebräuchlichen Wörtern entsprechen. Dies ist mit der Texterkennung zu begründen, mit der die Bestände aus Büchern, PDFs usw. extrahiert wurden. Es wäre möglich, jedoch mit einem hohen Zeitaufwand verbunden, den gesamten Bestand herunterzuladen, zu formatieren und durch eine Filterung der Häufigkeit von Einträgen die meisten ungewünschten Zeichenketten zu eliminieren.

Stattdessen wurde im Rahmen der Arbeit ein Auszug eines Speicherinhaltes (Dump) der deutschsprachigen Version von Wikimedia²³ heruntergeladen. Der mit „20180620“ seinem Erstellungsdatum nach benannte Dump enthält die Bestände aller 2.192.364 deutschsprachigen Artikel, Stub-Pages und Disambiguitätsseiten, die es zu diesem Zeitpunkt gab.

²²<https://docs.unity3d.com/ScriptReference/Windows.Speech.GrammarRecognizer.html>

Letzter Zugriff: 29.09.2018

²³<https://dumps.wikimedia.org/backup-index.html> Letzter Zugriff: 22.08.2018

Da der Inhalt des Dumps nicht ohne Verarbeitung verwendet werden kann, wurde das Python-Tool WikiExtractor²⁴ hinzugezogen, welches den Text aus einem solchen Dump extrahiert und anschließend Steuerbefehle und Metadaten entfernt. Nach der Ausführung der *WikiExtractor.py* und zusätzlicher manueller Bereinigung der Daten bildeten die Inhalte des Dumps 20.505.850 Zeilen und Titel.

Für die Weiterverarbeitung der Daten wurde daraufhin in Python ein N-Gramm-Generator entwickelt (*NgramGenerator.py*). Dieser erstellt Uni-Gramme und Bi-Gramme aus den Zeilen und Titeln und zählt deren Vorkommen, also die absolute Häufigkeit. Die Wahl der Programmiersprache fiel für diese Aufgabe auf Python, da sich das Programm in einer Weiterführung der Arbeit zusammen mit dem WikiExtractor in eine Prozesskette kombinieren lässt. So wurde eine 196 Megabyte große Uni-Gramm-Liste, sowie eine 1,3 Gigabyte große Bi-Gramm-Liste angelegt. Die Uni-Gramm-Liste enthält 11.418.288 Einträge und die Bi-Gramm-Liste enthält 60.092.226 Einträge.

Erste Tests zeigten angemessene Wortvorschläge, jedoch benötigt die Verarbeitung der N-Gramme mehrere Sekunden, währenddessen keine neuen Bilder für das HMD erzeugt werden können. Da der Abgleich mit dieser Menge von Einträgen viel Rechenleistung benötigt, wurde davon abgesehen, zusätzlich Tri-Gramme zu verwenden, da diese nur selten zum Tragen kämen und die Zahl der Bildausfälle erhöht hätten.

Es wurden mehrere unterschiedlich stark reduzierte Versionen der N-Gramme erstellt, deren Einträge verschiedene Mindesthäufigkeiten überschreiten, da N-Gramme mit sehr niedrigen Häufigkeiten ebenfalls nicht das typische deutsche Sprachbild repräsentieren. In der Anwendung werden jene N-Gramme verwendet, deren Mindesthäufigkeit 100 beträgt. Das Uni-Gramm Dictionary besteht somit noch aus 237.847 Einträgen und das Bi-Gramm Dictionary aus 131.751 Einträgen. Außerdem wurde das System in einen eigenen Thread ausgelagert, sodass bei der Benutzung keine Standbilder für Nutzer mehr entstehen, sondern allenfalls die neue Vorhersage mit einer kleinen Verzögerung erscheint.

Nun, da die N-Gramme vorhanden sind und beliebig zwischen möglichst vielen N-Grammen versus möglichst kurzer Ladezeiten gewichtet werden kann, werden im Nachfolgenden die Verfahren zu Wortvorschlägen und Vorhersagen konkret entwickelt.

4.2.2.1 Korrektur & Vervollständigung

Bei der Berechnung der Levenshtein-Distanz wurde eine Begrenzung der maximalen Wortlänge festgelegt. Diese Begrenzung gilt allerdings nicht für die Länge der Vorschläge, sondern für die Länge, wie lang ein vom Nutzer geschriebenes Wort sein darf um noch Vorschläge aus dem Assistenzsystem zu erhalten. Der Grund für diese Entscheidung ist die Begrenzung der Größe der Levenshtein-Matrix aus Abbildung 2.3, mit dem Ziel, die benötigte Rechenzeit bei langen Worten zu umgehen. Die Länge der Wörter in dem Uni-Gramm Dictionary sollte hingegen nicht begrenzt sein, da die potentielle Zeichensparnis größer ist, wenn lange Wörter frühzeitig korrekt vorgeschlagen werden. Allerdings muss im Gegenzug die Betrachtungslänge des Geschriebenen begrenzt werden, da sonst bei der Eingabe jedes Zeichens zunehmend große Matrizen berechnet werden müssen. Die Begrenzung konnte nach einigen Optimierungsvorgängen zur Hauptstudie auf 16 Zeichen gesetzt werden, während sie zuvor bei der Pilotstudie noch 14 Zeichen betrug.

Der maßgebliche Faktor zur Selektion der Uni-Gramme ist die minimal und vor allem maximal erlaubte Levenshtein-Distanz. Der Algorithmus wurde so implementiert, dass die fehlenden Zeichen von potentiell unvollständigen geschriebenen Wörtern gegenüber einem Eintrag des Uni-Gramms nicht als Fehler gewertet werden.

²⁴<https://github.com/attardi/wikiextractor> Letzter Zugriff: 01.10.2018

Dies hat den Grund, dass so die minimal geforderte Levenshtein-Distanz auf null gesetzt werden kann und dennoch Einträge akzeptiert werden, die das aktuelle Wort vervollständigen könnten.

Das heißt die Distanz zwischen „Nu“ und „Null“ liegt in dieser abgewandelten Variante bei null, wodurch „Null“ als potentieller Vorschlag akzeptiert wird. Am anderen Ende der Skala limitiert die maximale Distanz, wie weit ein Eintrag vom Geschriebenen entfernt sein darf um noch als potentieller Vorschlag zu gelten. Für diesen Filter wurde schließlich der Wert zwei festgelegt, sodass „diesss“ noch den Vorschlag „dieser“ erhält, jedoch „dissss“ von „dieser“ zu weit entfernt ist.

Die maximale Distanz muss allerdings nicht zwangsläufig auf einen festen Wert gesetzt werden. Es ist denkbar mit einer maximalen Distanz von eins zu starten und sobald für die aktuelle Distanz weniger als drei Vorschläge gefunden werden, die N-Gramme mit einer um +1 erhöhten maximalen Distanz erneut zu durchsuchen. Es ist davon abzuraten, statt multipler Durchläufe schon beim ersten Durchlauf bereits Einträge abzuspeichern, die außerhalb der gewünschten Distanz liegen, da sonst gerade zu Beginn eines Wortes tausende von Wörtern mit ihren Distanzen abgespeichert werden würden. Außerdem wird beim Überschreiten der maximal zulässigen Distanz die Berechnung der Matrix sofort abgebrochen, um weitere Rechenzeit zu sparen.

Um das System zusätzlich zu beschleunigen, wird bei dem Filtern der Einträge außerdem zwischen Groß- und Kleinschreibung unterschieden. Die Großschreibung des ersten Buchstabens kann entweder wie ein Fehler angesehen werden für Worte die mit dem gleichen, jedoch kleingeschriebenen Buchstaben anfangen, oder dies könnte als kein Fehler gewertet werden, da Nutzer vom Schreiben auf Mobilgeräten gewohnt sein können, dass das ausschließliche Schreiben von Kleinbuchstaben keinen Einfluss auf die Vorschläge hat. Da, wie in Kapitel 5 beschrieben wird, zur Evaluation der Verfahren in der Studie ein Text abgeschrieben wird, ist davon auszugehen, dass Nutzer auf die Verwendung von Groß- und Kleinschrift achten.

Daher wird stattdessen die Groß- und Kleinschrift als harter Filter dafür genutzt, welche Einträge zur Berechnung der Levenshtein-Distanz weiter betrachtet werden. Fängt ein geschriebenes Wort mit einem Großbuchstaben an, so werden sämtliche Einträge, die mit Kleinbuchstaben anfangen, nicht weiter berücksichtigt. Der zugehörige Code befindet sich in der *Levenshtein*-Klasse.

4.2.2.2 Wortvorhersage

Zur Ermittlung der Vorhersagen künftiger Wörter stellen bei den Bi-Grammen, anders als bei dem Korrektur- & Vervollständigungssystem, weder kleingeschriebene, noch großgeschriebene vorherige Wörter einen Filter dar, der die Vorhersage des nächsten Wortes beeinflusst. Dies ist der Fall, da sowohl auf z.B. „der“, wie auch auf „Der“ die gleichen Vorschläge, mit einer auf die deutsche Sprache gesehen durchschnittlich gleich hohen Wahrscheinlichkeit, folgen können.

Konkret wird die Vorhersage bei jedem Leerzeichen und jeder Auswahl eines vorherigen Wortvorschlages genutzt. Diese füllt die Tasten des Systems daraufhin mit Vorschlägen, welche bei der Übereinstimmung mit dem nachfolgend geschriebenen beibehalten und anderenfalls durch das Korrektur- & Vervollständigungssystem ersetzt werden.

Das *PredictionCanvas*-Prefab beinhaltet die Funktionalität des Assistenzsystems sowie die Tasten der Vorhersage und dient als modulares Objekt, welches den Eingabeverfahren hinzugefügt wurde. Dabei handelt es sich um eine instanzierbare Objektklasse, an der Änderungen auf alle Instanzen angewendet werden.

4.2.3 Textunterstützungen

Die meisten der in Unterabschnitt 3.2.4 genannten Textunterstützungen wurden implementiert. Dazu zählt das automatische Hinzufügen von Leerzeichen nach der Wahl eines Wortvorschlages sowie das Vertauschen von Punkten, Kommata, Ausrufungszeichen und Fragezeichen mit einem Leerzeichen, wenn diese hinter dem Leerzeichen geschrieben werden.

Es ist jedoch nicht möglich, die Cursorposition in dem Textfeld durch die Berührung mit einem Controller zu bestimmen. Bei der Thumbpad-Tastatur, sowie den Verfahren auf der virtuellen Tastatur wäre es hilfreich, wenn der Nutzer durch eine direkte Auswahl, sei es durch das Positionieren des Cursorobjektes bei der Thumbpad-Tastatur (Unterabschnitt 4.3.1), oder durch Berühren des Textfeldes an entsprechender Stelle bei den Verfahren Touch, Drumstick und Laser den Cursor an die passende Position setzen könnte. Für die Umsetzung dieser Funktionalität ist es notwendig, zu einer Position auf der Oberfläche des Textfeldes eine zugehörige Textposition bestimmen zu können. Leider verfügt weder die native *InputField*-Komponente, noch die von TextMeshPro angebotene *TMP_InputField*-Komponente über diese Funktion.

Theoretisch könnte dies über die Kenntnisse der Breite jedes einzelnen Zeichens, unter Berücksichtigung auf eventuelle Verschiebungen des Textes – aufgrund des Überschreitens der Länge des Eingabefeldes – und unter Hinzunahme der relativen Position der Selektion auf dem Eingabefeld ausgerechnet werden, jedoch misslang dies aufgrund dessen, dass die *InputField*-Komponenten die Verschiebung des Textes nicht nutzbar machen.

Die Editierbarkeit des Textes durch Cursortasten ist dennoch gegeben. Durch Betätigen der Pfeiltasten springt der Cursor ein Zeichen nach links bzw. rechts und durch die Nutzung der Funktionsweise über die die Shift-Taste abgebildet wird kann bis zum nächsten bzw. vorherigen Leerzeichen gesprungen werden. Durch Gedrückthalten der Pfeiltasten kann der Cursor außerdem zum Beginn und Ende des Textes bewegt werden.

Nach einem Punkt, Ausrufungszeichen oder Fragzeichen erscheinen die nachfolgenden Vorschläge immer beginnend mit einem Großbuchstaben. Dies geschieht, anders als Unterabschnitt 3.2.4 konzipiert, jedoch nach jedem dieser Zeichen und unterscheidet nicht nach Abkürzungen, da nicht eindeutig ermittelt werden konnte, ob Abkürzungen öfter innerhalb eines Satzes oder am Ende eines Satzes auftreten. Daher müsste bei jedem Punkt semantisch bestimmt werden, ob er das Ende eines Satzes darstellt.

4.3 Thumbpad-Verfahren

Bei dieser Gruppe von Verfahren liegt ein zusätzliches Augenmerk darauf, die Position der Darstellung der Tastaturen für Nutzer so festzulegen, dass die Eingabe sowohl bequem im Sitzen, als auch uneingeschränkt im Stehen durchgeführt werden kann.

Für die Darstellung der Zeichen und des Eingabefeldes wurden Textfelder von TextMeshPro benutzt, um verpixelte Schrift bei genauerer Betrachtung der grafischen Darstellung der Verfahren zu verhindern. Die nachfolgend abgebildeten Tastaturen werden an die Controller geheftet und befinden sich perspektivisch leicht oberhalb der virtuellen Modelldarstellung der Controller. Die Grafikelemente wurden bei allen Verfahren der Gruppe näher zum Kopf der Controller hin positioniert, sodass der Blickwinkel zwischen dem Kontextobjekt und dem Eingabeverfahren reduziert wird.

4.3.1 Thumbpad-Tastatur

Das Verfahren wurde der Konzeption entsprechend implementiert. Bezüglich des Layouts der Tasten wurde lediglich in der untersten Zeile etwas abgerundet, um dem kreisförmigen Thumbpad besser zu entsprechen (s. Abbildung 4.1). Am oberen Ende wurde hingegen auf eine Abrundung verzichtet, da dies eine Verkleinerung der Schriftgröße der Vorschläge mit sich bringen und die Breite des Textfeldes reduzieren würde.

Dem Verfahren wurde eine kleine rote Kugel als Cursorobjekt hinzugefügt, welche die relative Position des Berührungspunktes des Daumens auf dem Thumbpad verdeutlicht. So kann der Nutzer besser abschätzen, ob sich der Auflagepunkt nah am Rand einer Taste befindet, oder mittig auf der Taste liegt. Dadurch soll ungewolltes Abrutschen zu anderen Tasten, insbesondere beim Schreiben von Großbuchstaben, verringert werden. Lässt man das Thumbpad los, so springt die Kugel zurück auf ihre Ursprungsposition, wie in Abbildung 4.1 auf der rechten Seite zu sehen. Bei dem linken Controller ist hingegen zu sehen, wie eine Taste, über die die Kugel schwebt, farblich hervorgehoben wird. Sobald die Taste gedrückt wird, verfärbt sich das helle Cyan zu einem kräftigeren Cyan.

Die Tastatur wurde bewusst nicht so positioniert, dass sie mittig über dem Thumbpad liegt, da sie sonst so weit unten läge, dass Nutzer ihren Blick vermehrt auf die Hände richten müssten. Durch die erhöhte Position müssen die Arme weniger angehoben werden und der Winkel zwischen dem Kontextobjekt und der Tastatur ist verringert.

Zusätzlich ist, wie ebenfalls in Abbildung 4.1 zu sehen, bei allen Tasten, die eine alternative Belegung besitzen, eine kleine Vorschau der alternativen Zeichen an der oberen rechten Ecke der Tasten zu sehen. Entsprechend ist erkennbar, dass zum Zeitpunkt der Implementierung weitere 17 Tasten für zusätzliche Zeichen frei geblieben sind; beziehungsweise sogar 23 Zeichen, falls auch die Funktionstasten in der untersten Zeile alternativ belegt werden sollen.

Abbildung 4.1: Darstellung der Thumbpad-Tastatur



Das Betätigen der Trigger-Taste ändert hierbei das Layout auf beiden Tastaturhälften. Dies hat den Grund, dass eine Zeitersparnis dadurch möglich ist, dass mit einem Controller das Layout gewechselt werden kann um mit dem anderen Controller – insbesondere bei der Verwendung von Umlauten – in den meisten Fällen nur ein einziges Zeichen mit dem alternativen Layout geschrieben wird. Daraufhin kann mit dem ersten Controller zurück zum Standardlayout gewechselt werden. Es wird davon ausgegangen, dass diese Funktionsweise die Intuitivität des Verfahrens verbessert und die Geschwindigkeit der Eingabe erhöhen kann.

Das Verfahren wurde in der *Thumbpad-Tastatur-Szene* realisiert und das *Thumbpad-Typing*-Skript beinhaltet die Implementierung der Funktionalität.

4.3.2 Thumbpad-Kreise

Eine der großen Fragen bei der Darstellung der Thumbpad-Kreise ist, ob die einzelnen Tasten tatsächlich als Kreisausschnitte, oder als konventionelle quadratische Tasten dargestellt werden sollen. Die Entscheidung fiel auf Letzteres, da davon ausgegangen wird, dass solche Tasten Nutzern tendenziell vertrauter sind und die Intuitivität für wichtiger befunden wird als ein optisch ansprechenderes Aussehen durch Tasten in Form von Kreisausschnitten (vgl. Abbildung 4.2).

Bei diesem Verfahren wurde außerdem bewusst auf den Einsatz eines Cursorobjektes verzichtet, da die Tasten durch einen Winkelbereich statt einer Position definiert sind und die Kugel naheläge, dass diese bis zu der Position der Tasten bewegt werden sollte, obwohl dies nicht der Fall ist, sondern bereits eine kleine Distanz von der Mitte des Thumbpads in Richtung einer Taste ausreicht um eine Taste zu selektieren. Würde sich das Cursorobjekt stattdessen nur auf dem Kreis der Zeichen bewegen können, so würde sie nahelegen, dass der Daumen entlang des Zeichenkreises bewegt werden soll, was ebenfalls nicht der Fall ist.

Wird eine Taste gedrückt, so wird sie nicht nur entsprechend eingefärbt, sondern auch etwas größer skaliert, um dem Nutzer zu suggerieren, dass nun ein größerer virtueller Winkel existiert, der der Taste zugeordnet ist. Durch diese Funktion kann das Abrutschen zu benachbarten Tasten beim Gedrückthalten einer Taste reduziert werden.

Abbildung 4.2: Darstellung der Thumbpad-Kreise



Da bei diesem Verfahren die Vorschläge nicht direkt über das linke Thumbpad ausgewählt werden können, ist es bei stattdessen möglich, wie in Abschnitt 3.3 konzipiert, die Grip-Taste als Steuertaste zu verwenden, um durch die horizontale Position des Daumens auf dem Thumbpad einen der Vorschläge auszuwählen.

Das Verfahren wurde in der *ThumbpadKreise*-Szene realisiert und das *ThumbpadKreise*-Skript beinhaltet die Implementierung der Funktionalität.

4.3.3 Thumbpad-Clusters

Die Implementierung dieses Verfahrens unterscheidet sich von den anderen Verfahren dieser Gruppe unter anderem darin, dass sowohl die Wortvorschläge, als auch die Textzeile auf dem gleichen Controller liegen (s. Abbildung 4.3). Diese Entscheidung wurde getroffen, da dies für geübte Schreiber ermöglicht, beim Nutzen des Verfahrens lediglich auf den in der Darstellung rechten Controller schauen zu müssen. Anfänglich ist dadurch jedoch gegensätzlich ein vermehrtes alternierendes Schauen auf die Controller nötig, da sowohl die Wahl des richtigen Clusters auf einem Controller, wie auch die Überprüfung der Auswahl eines Zeichens innerhalb eines Clusters auf dem anderen Controller betrachtet wird.

Eine Unterscheidung gegenüber dem in Unterabschnitt 3.3.3 vorgestellten Konzept dieses Verfahrens ist, dass ein neuntes Cluster in der Mitte der Fläche des Thumbpads hinzugefügt wurde. Durch das zusätzliche Cluster verfügt das Verfahren über 36 Tasten pro Belegung und daher, unter Einbezug der alternativen Tastenbelegung, welche erneut über die Trigger-Taste erreicht werden kann, insgesamt über 72 Belegungen. Damit hat dieses Verfahren in der Gruppe der Thumbpad-Verfahren die meisten verfügbaren Tasten. Auf dem neunten Cluster liegen die vier Funktionstasten „<“, „>“, Back und Enter.

Abbildung 4.3: Darstellung der Thumbpad-Cluster



Bei diesem Verfahren wird die alternative Tastenbelegung nicht als Vorschau angezeigt, da vermutet wird, dass die Menge an Zeichen und Gruppen bereits ohne die Vorschauen der alternativen Tastenbelegung komplex erscheint, da alle Zeichen auf einem Controller zu sehen sind und nicht auf beide Controller verteilt werden. Generell ist es jedoch für erfahrene Nutzer potentiell interessanter, die Vorschauen zu sehen und es wird nicht pauschal davon abgeraten, diese anzuzeigen.

Das Großschreiben eines Buchstabens erfolgt auch bei diesem Verfahren über das Gedrückthalten des Buchstabens bei dem in Abbildung 4.3 rechts abgebildeten Controller. Theoretisch ist es möglich, dass die Schwellzeit des Gedrückhaltens bereits mit dem linken Controller beginnen kann, mit dem tendenziell zunächst das gewünschte Cluster ausgewählt wird, bevor an dem anderen Controller das gewünschte Zeichen innerhalb des Clusters gedrückt wird. Diese Idee kam jedoch erst nach Beginn der Studie auf und wurde daher nicht implementiert.

Das Verfahren wurde in der *ThumbpadClusters*-Szene realisiert und das *Thumbpad-Cluster*-Skript beinhaltet die Implementierung der Funktionalität.

4.3.4 Thumbpad-Gesten

Die Implementierung des Thumbpad-Gesten Verfahrens ist in Abbildung 4.4 zu sehen. Unabhängig davon, wo der Nutzer den Daumen auf das Thumbpad legt, beginnt grundsätzlich ein zunächst roter Punkt in der Mitte der Zeichengruppen, also am unteren Ende der in der Abbildung zu sehenden blauen Linie. Von diesem Punkt aus kann der Nutzer das Thumbpad entlang fahren und dadurch eine Linie in der virtuellen Welt zeichnen. Wurde durch die erste Hälfte der Geste eine Gruppe ausgewählt, so verfärbt sich die Linie von rot zu grün und die bisher gezeichnete Geste, die möglicherweise nur knapp die gewünschte Zeichengruppe getroffen hat, wird durch einen Linienprototypen ersetzt, der dem Nutzer anzeigt, wie eine perfekte Geste von der Mitte zu der Zeichengruppe ausgesehen hätte und so dem Nutzer das zukünftige Ansteuern der korrekten Zeichengruppe erleichtert. In dem Moment der Auswahl einer Zeichengruppe wird außerdem die Gruppe in hellem cyan markiert.

Nun hat der Nutzer die Möglichkeit entweder das Thumbpad loszulassen, um so das Schreiben des Zeichens abubrechen, die Buchstabengruppe farblich zurückzusetzen und die Linie zu löschen, oder er kann den Daumen weiterbewegen um innerhalb der Gruppe eines der Zeichen auszuwählen. In Abbildung 4.4 wurde so der Buchstabe „e“ ausgewählt. Sobald ein Zeichen ausgewählt und die Geste somit beendet ist, färbt sich die Linie von grün zu blau, der Controller vibriert kurz, die Buchstabengruppe wird farblich zurückgesetzt und der Buchstabe geschrieben. Wenn während diesem Moment das Thumbpad gedrückt gehalten wird, erscheint ein Großbuchstabe. Dabei kann der Nutzer selbst bestimmen, ab welchem Zeitpunkt das Thumbpad gedrückt gehalten wird. Es ist lediglich von Belang, ob es zum Moment des Schreibens des Buchstabens gedrückt ist oder nicht.

Durch diese Anpassung der Funktionsweise von Großbuchstaben entfällt die zeitliche Verzögerung, die bei den anderen Verfahren mit dem Schreiben eines Großbuchstabens einhergeht, und so wird das Potential hoher Schreibgeschwindigkeiten weiter gesteigert. Bei diesem Verfahren befinden sich außerdem sämtliche Zeichen auf jedem Controller.

Mit ausreichend Expertise ist es daher möglich, mit einem Controller bereits die gewünschte Zeichengruppe anzusteuern, während mit dem anderen Controller simultan das Beenden einer Geste durchgeführt wird.

Das Verfahren wurde in der *ThumbpadGesten*-Szene realisiert und das *Thumbpad-Gesten*-Skript beinhaltet die Implementierung der Funktionalität.

Abbildung 4.4: Darstellung der Thumbpad-Gesten



4.4 Handschrift / Gesten

Das Handschrift-Verfahren wurde während der Implementierungsphase abgebrochen. Es zeichnete sich nicht als vielversprechend ab, da in vorab durchgeführten Tests erreichte Geschwindigkeiten weit unter den Geschwindigkeiten der Tests anderer Verfahren lagen. Der Fokus der Arbeit wurde daher vermehrt auf die anderen Verfahren konzentriert. Dennoch soll an dieser Stelle ein kurzer Überblick über die Funktionsweise und das Vorgehen gegeben werden, da es sich bei dem Verfahren dennoch um eine Schreibmethode handelt, die es Nutzern erlaubt ohne die Einführung zusätzlicher Abstraktionsschichten in der VR Texte zu schreiben.

Dem Nutzer werden bei diesem Verfahren, wie in Abbildung 4.5 zu sehen, statt der Controller ein Permanentmarker und ein Whiteboard-Schwamm in den Händen angezeigt. Vor dem Nutzer befindet sich ein Whiteboard auf dem geschrieben werden kann. Links neben dem Whiteboard befinden sich Funktionstasten. Mit diesen kann das Geschriebene in digitale Schrift umgewandelt werden, das Whiteboard bereinigt werden, die vorherige Umwandlung rückgängig gemacht werden und die bisherige digitale Schrift gelöscht werden. Die Implementierung befindet sich in der *CloudVision*-Szene und in dem Skript *WhiteboardDrawer* wurden der Permanentmarker und der Schwamm implementiert.

Die Umwandlung des Geschriebenen in digitalen Text findet über die Cloud Vision API²⁵ von Google²⁶ statt. In der *CloudVision*-Klasse wird die Textur angelegt, die während der Benutzung beschrieben wird. Diese wird auf Befehl des Nutzers zu einem Server von Google mit der Information geschickt, dass Text in dem Bild erkannt werden sollen. Der Server antwortet daraufhin mit einem Datenpaket in dem unter anderem der Text angegeben ist.

Diese Funktionsweise schien zunächst sehr vielversprechend zu sein, da Google mit dieser Technik eigene Datenbestände an Text aus PDFs und von Bildern extrahiert (Google LLC, 2018a). Es hat sich jedoch herausgestellt, dass der Dienst nicht auf die Interpretation von Handschrift spezialisiert ist und diese daher gegenüber digital erstelltem Text mit einer erhöhten Fehlerquote detektiert wird.

²⁵<https://cloud.google.com/vision/> Letzter Zugriff: 30.09.2018

²⁶<https://www.google.de/about/> Letzter Zugriff: 30.09.2018

Erschwerend kommt hinzu, dass zum Zeitpunkt der Arbeit nur 1.000 solcher Erkennungen pro Monat kostenlos sind und jede weiteren 1.000 Erkennungen 1,50\$²⁷ kosten. Bei einer Fortsetzung des Verfahrens wäre es wünschenswert gewesen, wenn schon während des Schreibens regelmäßig das Bild interpretiert wird um das Assistenzsystem der Wortvorschläge nutzen zu können, doch würde dies mehrere Anfragen pro Sekunde bedeuten. Dadurch würde die Zahl der benötigten Erkennungen immens steigen und die Kosten für die Nutzung des Dienstes zunehmend störender werden.

Für die Weiterentwicklung dieses Verfahrens würde sich eventuell das „WritePad® Handwriting Recognition SDK“²⁸ eignen, dessen Beschreibung eine akkurate, schnelle, kompakte, einfach zu benutzende, multilinguale, plattformübergreifende Handschrifterkennungstechnologie verspricht. Das SDK kann außerdem offline und kostenlos benutzt werden.

Abbildung 4.5: Darstellung des Handschrift-Verfahrens



4.5 Controller-Verfahren

Die bei der Gruppe der Controller-basierten Verfahren verwendete Tastatur beruht auf einer Abwandlung der in Abschnitt 4.1 vorgestellten VRKeys Tastatur. Die Position der Funktionstasten wurde angepasst, unnötige Tasten (wie die „.com“ Taste) entfernt, das deutsche QWERTZ-Layout übernommen, die Tasten der Wortvorhersage hinzugefügt und das Textfeld ersetzt. Ferner wurde die Benutzung der Tasten mit der Verwendung von Laserstrahlen und Gaze-Rays kompatibel gemacht sowie die Funktionalität der Tastatur in einigen Punkten erweitert. So ist es z.B. bei jedem der Verfahren möglich, die Trigger-Taste während des Schreibens eines Buchstabens gedrückt zu halten um einen Großbuchstaben zu schreiben.

Das Aussehen der Tastatur unter Verwendung der jeweiligen Verfahren ist für Touch in Abbildung 4.6, für Drumsticks in Abbildung 4.7, für Laser in Abbildung 4.8 und für Gazing in Abbildung 4.9 zu sehen. Unterhalb der Tastatur befindet sich eine Kugel, die in den torusförmigen Kopf der Vive-Controller passt.

²⁷<https://cloud.google.com/vision/pricing> Letzter Zugriff: 30.09.2018

²⁸<https://github.com/phantware/WritePadSDK> Letzter Zugriff: 30.09.2018

Wird der Controller entsprechend positioniert, so kann durch das Drücken der Grip-Taste die Tastatur starr an den Controller fixiert werden. Nutzer können sich mit der Tastatur drehen, diese bewegen und die Neigung der Tastatur zum Nutzer bestimmt werden. Durch erneutes Drücken der Grip-Taste verlässt die Tastatur ihre Haftung an den Controller und schwebt erneut frei im Raum.

Zusätzlich kann die Tastatur, während sie an einen Controller fixiert ist, durch die Benutzung des Thumbpads skaliert werden. So können Nutzer mit kürzeren Armen oder zur Verringerung der körperlichen Anstrengung die Tastatur kleiner skalieren und Personen mit längeren Armen oder zur Verringerung der Fehlerquote diese größer skalieren.

Auf dem linken Controller sind in den Abbildungen außerdem Pfeile nach links und rechts zu erkennen. Diese können als Tasten genutzt werden um den Cursor des Textfeldes nach links und rechts zu bewegen. Bei Gedrückthalten des Thumbpads springt der Cursor nach einer kurzen Zeit (350ms) um ein ganzes Wort bis zum nächsten, bzw. vorherigen Leerzeichen. Bei noch längerem Gedrückthalten (700ms) springt der Cursor ganz bis zum Ende, bzw. zum Anfang des Eingabefeldes.

Die Pfeile nach oben und unten dienen hingegen als Tasten zur Skalierung der Tastatur. Der rechte Controller stellt durch die Zahlen „1 2 3“, am besten zu sehen in Abbildung 4.9, die Bereiche des Thumbpads dar, mit denen auch in diesen Verfahren die Grip-Taste gedrückt gehalten werden kann, um gefolgt von einer Auswahl des linken, mittleren oder rechten Bereichs des Thumbpads den linken, mittleren oder rechten Wortvorschlag zu nutzen. Mit jedem der nachfolgenden Verfahren können die Vorschläge jedoch auch direkt über die jeweilige Interaktionsmethode ausgewählt werden.

Die Tasten der Wortvorschläge und das Textfeld wurden außerdem, wie in Abbildung 3.6 dargestellt, zum Nutzer geneigt, sodass der Nutzer die Tastatur weiter unten im Sichtfeld positionieren kann und gleichzeitig – wie beim Schauen auf einen Bildschirm während des Schreibens auf einer Tastatur – zur Überprüfung des Geschriebenen den Blick geradeaus richten kann. So kann verhindert werden, schräg auf den weiter entfernten Text blicken zu müssen, wie es bei einer flachen Ausrichtung der Wortvorschläge und Textzeile der Fall gewesen wäre.

Die Verfahren befinden sich in der *VirtualKeyboard*-Szene. Bei dieser können im Inspector des GameObjects *Tastatur* die Verfahren durch die Kontrollkästchen *Touch*, *Drumstick*, *Laser* und *Gaze* ausgewählt werden.

4.5.1 Touch

Statt für spitz zulaufende Kegel, wie sie in Kapitel 3 erörtert werden, wurden Kapseln für die Collider verwendet, da mit diesen besser ersichtlich ist, wie weit der Collider von einer Taste entfernt ist, als bei einem spitz zulaufenden Ende. Die Kapseln wurden so eingefärbt, dass sie sich von der Tastatur stark abheben und wie ein fester Bestandteil des Controllers wirken, obgleich sie sich auch von diesen als eindeutiges Objekt unterscheiden.

Wie in Abbildung 4.6 zu sehen ist, wurden die Kapseln, wie vorgesehen, geneigt an die Controller fixiert, sodass die Hände während des Schreibens eine ergonomischere Haltung beibehalten können. Es wurde abgewogen, wie weit die Kapseln aus den Controllern herausstehen sollen, da weiter entfernte Kapseln für weniger Verdeckung sorgen, jedoch auch kleinere Rotationen der Handgelenke und Unterarme größere Bewegungen mit sich ziehen, während sehr nah am Controller anliegende Kapseln den gegenteiligen Effekt erzielen.

Die Entscheidung fiel auf nah anliegende Kapseln vor allem aus dem Grund, dass die Intuitivität des Verfahrens möglichst hoch gehalten werden soll und bei dieser Position der Kapseln ein deutlicherer Bezug zum Zwei-Finger-Schreiben existiert, als es bei weiter entfernten Kapseln der Fall wäre.

Das *KeyboardDevice*-Skript enthält die Implementierung der Controller, sowohl für das Touch-, als auch für das Drumstick-Verfahren.

Abbildung 4.6: Darstellung des Touch-Verfahrens



4.5.2 Drumstick

Bei diesem Verfahren wird im Gegensatz zu der Beschreibung von Unterabschnitt 4.5.1 die Collider weiter vom Controller entfernt. Die Collider werden entlang der Griffrichtung der Controller positioniert und wie in Abbildung 4.7 zu sehen ist, deutlich kleiner skaliert. Dies hat in Kombination mit den dünnen Stäben der Drumsticks den Effekt, dass die Menge der Verdeckung gering gehalten wird und sich das Verfahren in der Bedienung wie das Spielen eines Xylophons oder Schlagzeuges anfühlen soll. Die Drumsticks wurden, wie die Kapseln in Unterabschnitt 4.5.1, so eingefärbt, dass sie sich von der Tastatur abheben und die Kugel am Ende der Sticks so skaliert, dass sie einen klar erkennbaren Abschluss der Controller bilden.

Funktional sind das Touch- und das Drumstick-Verfahren fast identisch. Der einzige Unterschied liegt darin, dass bei diesem Verfahren die Nutzer mit den Pfeiltasten nach oben und unten auch die Länge der Drumsticks steuern können. Dies funktioniert nur, während die Tastatur sich nicht im Griff des Controllers befindet. Die Länge der Drumsticks der beiden Controller wird dabei gleich gehalten und der Kopf der Drumsticks bleibt kugelförmig.

Durch diese Funktion können Nutzer, indem sie die Drumsticks verlängern, einen vermehrten Wert auf eine geringere körperliche Belastung legen, oder zugunsten der Verringerung der Fehlerquote die Drumsticks kürzer einstellen, sodass die Verstärkung kleiner Rotationen der Hände und Arme durch die Länge der Drumsticks geringer ist.

Abbildung 4.7: Darstellung des Drumstick-Verfahrens



4.5.3 Laser

Bei diesem Verfahren werden, wie in Abbildung 4.8 zu sehen, Strahlen entlang der Griff-richtung aus den Controllern in den virtuellen Raum dargestellt. Diese visualisieren Raycasts, die auf Schnittpunkte mit Objekten im Raum getestet werden. Trifft ein Strahl auf eine Taste, so wird diese farblich hervorgehoben und der Strahl endet auf ihr. Der Nutzer kann sie daraufhin durch Betätigung der Trigger-Taste benutzen. Die optionale Funktion der Großschreibung wird bei diesem Verfahren mit dem Thumbpad durchgeführt, da die Trigger-Taste bereits zum Schreiben von Zeichen verwendet wird. Dies bedeutet, dass durch Gedrückthalten des Thumbpads während des Schreibens eines Zeichens die Shift-Funktionalität auf die Betätigung der Taste angewendet wird.

Damit beide Thumbpads für diese Funktionsweise verfügbar sind, ist für das Bewegen des Cursors mit dem Controller bei diesem Verfahren das Gedrückthalten der Grip-Taste notwendig. Auf diese Weise wird eine zeitliche Verzögerung durch das Warten auf einen Threshold auch bei diesem Verfahren eliminiert und die Shift-Taste der Tastatur muss nicht länger benutzt werden.

Zusätzlich bietet das Verfahren den Vorteil, dass davon auszugehen ist, dass die Controller weniger translatiert werden müssen, als bei den beiden Verfahren aus Unterabschnitt 4.5.1 und Unterabschnitt 4.5.2. Die Unterarme können in angewinkelter Orientierung vor den Körper gehalten werden und der Großteil der Selektion der Tasten kann über die Rotation der Handgelenke und Unterarme stattfinden.

Das *Laser*-Skript der Controller enthält die Implementierung der Funktionsweise des Schreibens auf der Tastatur und ermöglicht die Wahl der Wortvorschläge über die Laserstrahlen. Das *KeyboardDevice*-Skript enthält außerdem die abgewandelte Version der Tastatursteuerung über die Controller.

Abbildung 4.8: Darstellung des Laser-Verfahrens



4.6 Gaze

Die Implementierung des Gaze-Verfahrens befindet sich in der *GazeRaycaster*-Klasse. Wie vorgesehen füllt sich ein kleiner Kreis im Zentrum des Sichtfeldes. Dazu wird bei jedem neuen Bild (Frame) geprüft, ob die Kopfausrichtung des Nutzers noch auf die gleiche Taste fällt und wenn dem so ist, wird der Kreis weiter gefüllt. Abbildung 4.9 zeigt eine Momentaufnahme in der gerade der Kreis gefüllt wurde, woraufhin dieser im nächsten Frame wieder verschwinden und der Buchstabe „j“ geschrieben wird. Der Bildausschnitt des Kreises wurde zu Anschauungszwecken vergrößert dargestellt.

Über den linken Controller lässt sich oben bzw. unten auf dem Thumbpad die Schwellzeit einstellen, die ein Blick auf einer Taste ruhen muss, um diese zu benutzen. Außerdem kann durch das Betätigen der Trigger-Taste ein Zeichen vorzeitig geschrieben werden. Dadurch beginnt sich der Kreis erneut zu füllen, der bereits erreichte Anteil der Schwellzeit wird also zurückgesetzt. Das Gedrückthalten des Thumbpads während des vorzeitigen Schreibens eines Zeichens fügt die Shift-Funktion der Taste hinzu, bzw. löscht ein ganzes Wort bei der Rücktaste.

Um unabsichtliches Abbrechen des Schreibens eines Zeichens durch Kopfbewegungen, die zum Verlassen des Fokus von der zu schreibenden Taste führen, zu verhindern, wurde zunächst der Collider der Taste mit zunehmender Füllung des Kreises größer. Dies hat jedoch den Effekt, dass der Collider über andere Tasten hinausragt, was bei einer kurzen Schwellzeit zum Schreiben eines Zeichens führen kann, dessen Nachbarzeichen geschrieben werden sollte. Daher wird stattdessen der erreichte Anteil der Schwellzeit beim Verlassen einer Taste kurz zwischengespeichert, sodass für den Nutzer die Möglichkeit besteht, die Füllung des Kreises fortzusetzen, wenn er kurzfristig zu der Taste zurückkehrt.

Es wurde darauf geachtet, das Zeitfenster für diese Funktion sehr klein zu wählen, da es sonst zum unabsichtlichen Schreiben eines Zeichens kommen kann, wenn die Kopfausrichtung wiederholt über eine Taste fährt, während sich der Nutzer z.B. umschauf oder beim Lesen des geschriebenen Textes hoch und wieder herunter schaut.

Das *GazeRaycaster*-Skript enthält die Implementierung der Schwellzeit und die Darstellung des Ladekreises, sowie die Logik zur Selektion von Tasten. Die Steuerung der Tastatur befindet sich auch bei diesem Verfahren im *KeyboardDevice*-Skript.

Abbildung 4.9: Darstellung des Gaze-Verfahrens²⁹



4.7 Trainingsvorbereitung

Das Training, welches die Teilnehmer während der Durchführung der Studie vor den Diktaten durchlaufen, unterscheidet sich zwischen Pilotstudie und Hauptstudie. So haben während der Pilotstudie, durch die Verwendung des Embedded Browsers, die Teilnehmer die Möglichkeit, auf der Website <https://codeshare.io/> nach eigenem Ermessen Freitext zu schreiben. Alternativ können geschriebene Texte anschließend in eine Textdatei gespeichert und den Teilnehmern geschickt werden, sodass diese sich Notizen erstellen können. Da bei der Pilotstudie jedes Verfahren nur für zehn Minuten trainiert, bevor die Verwendung in einem Diktat getestet wird, ist dies ausreichend, um den Teilnehmern eine Aktivität zu liefern, die zur Benutzung der Eingabeverfahren führt.

Die Skripte des Embedded Browser Unity-Assets wurden für dieses Vorhaben angepasst. Die Steuereinheit des Webbrowsers stellt das *VRBrowser*-Prefab dar. Dies erlaubt mit dem *VRMainControlPanel*-Skript die Steuerung des Browsers und instanziiert das *Browser-Pane*-Prefab, welches die Steuerung des Browserfensters beinhaltet.

²⁹Der Kreis befindet sich rechts vom Zentrum des Bildes, da die Perspektive des linken Auges gezeigt wird.

Für das Training während der Hauptstudie werden hingegen 300 PhD Comics³⁰ verwendet. Die Verwendung dieser wird in Abschnitt 5.4 näher erklärt. Es wurde ein System entwickelt, welches das Umschalten von Comics durch die Betätigung der Enter-Taste ermöglicht. Die Teilnehmer der Studie können so selbst steuern, wie lange ihnen ein Comic angezeigt wird. Die Implementierung dessen befindet sich als Teil des *ComicTranslator*-Prefabs in dem *ComicTranslator*-Skript.

4.8 Diktate

Zur Durchführung der Pilotstudie sollen die Versuchspersonen zum Ende der Benutzung jedes Verfahrens ein fünfminütiges Diktat schreiben. Für die Pilotstudie wurden zur Erstellung der Diktate randomisiert Worte aus der Fassung der Uni-Gramm-Liste ausgewählt, in der die Einträge eine Mindesthäufigkeit von 10 besitzen. So wurden Diktate, bestehend aus je 1.000 Wörtern, durch die Methode „Ziehen mit Zurücklegen“ aus über 1,3 Millionen Möglichkeiten bestimmt. Die Wahrscheinlichkeit der Ziehung eines Wortes skalierte dabei mit der diesem Wort zugeordneten Häufigkeit, um ein realistisches Verhältnis der Zahl der Vorkommen des Wortes zu generieren.

Durch das Testen der Verfahren anhand solcher Diktate kann neben den Eingabeverfahren das Assistenzsystem der Korrektur & Vervollständigung getestet werden. Das System der Wortvorhersage ist hingegen für Diktate dieser Art nicht geeignet, da kein Kontext zwischen den Wörtern existiert, der die deutsche Sprache abbildet.

In der Hauptstudie wurden hingegen Sätze aus der deutschen Fassung von Stephen Hawking's „A Brief History of Time“³¹ („Eine kurze Geschichte der Zeit“) für die Erstellung der Diktate gewählt. Die Entscheidung, die Diktate aus Buchausschnitten zusammenzustellen, ist damit begründet, dass diese das realitätsnahe Schreiben von Texten simulieren. Die Wahl fiel auf dieses Buch, da es sich um ein populärwissenschaftliches Buch handelt, in dem sowohl Beschreibungen vorkommen, die strukturell solchen Texten ähneln, aus denen die N-Gramme zusammengesetzt wurden, jedoch ebenso zu Teilen sehr persönlich geschrieben, und damit stilistisch anders als die Quelle der N-Gramme ist. Dadurch sollen in den Diktaten sowohl Sätze enthalten sein, bei denen die Assistenzsysteme sehr hilfreich sein können, als auch solche, bei denen sie verhältnismäßig wenig Hilfestellung liefern können. Dies bildet eine stärker verallgemeinerbare Aussage über das Maß der Hilfestellung durch die Assistenzsysteme, als wenn die Diktate ausschließlich aus solchen Texten bestünden, die den Quelltexten der N-Gramme sehr ähneln, oder ausschließlich solche, die den Texten in ihrer Schreibweise stilistisch sehr entfernt sind. Auf diese Weise kann bei der Betrachtung der Ergebnisse die Verbesserung der Geschwindigkeit durch die Verwendung der Assistenzsysteme gerechtfertigt abgeschätzt werden.

Der Text des Buches wurde extrahiert und in einzelne Sätze zerlegt. Extrahierte Sätze, die nicht Teil des Haupttextes des Buches sind, wurden entfernt, wie z.B. Abbildungsbeschreibungen, die Danksagung, etc. Die Reihenfolge der Sätze wurde für die Erstellung jedes der zehn Diktate der Hauptstudie randomisiert und auf 100 Sätze begrenzt.

Das *DictationSystem*-Prefab beinhaltet die Darstellung des Diktatsystems, sowie dessen Implementierung in dem *DictationSystem*-Skript.

³⁰<http://phdcomics.com> Letzter Zugriff: 01.10.2018

³¹<http://www.hawking.org.uk/a-brief-history-of-time.html> Letzter Zugriff: 01.10.2018

4.9 Logging

Zur Auswertung der Daten muss nach der Durchführung der Studie eingesehen werden können, wann der Nutzer welche Aktion durchgeführt hat. Dabei sind nicht nur bewusste Entscheidungen wichtig, sondern auch die Menge an technisch gesehen unnötigen Bewegungen sowie die Häufigkeit des Blickwechsels zwischen dem Geschriebenen und der Tastatur der Verfahren können Aufschluss über Vor- & Nachteile der Eingabemethoden geben.

Bei der Pilotstudie wurden die Daten ausschließlich während des Diktates gesammelt, während bei der Hauptstudie, auf die Vermutung hin, dass sich auch aus dem Eingabeverhalten während der Trainingszeit wichtige Erkenntnisse gewinnen lassen, die gesamte Verwendungsdauer über Daten aufgezeichnet wurden. Während der Nutzung der Verfahren wurden die folgenden Daten bei jedem Frame, also etwa 90-mal pro Sekunde aufgezeichnet:

- Ist das Diktat gerade aktiv? – **Boolean**
- Funktioniert das Tracking? – **Boolean**
- Die Uhrzeit, inklusive Millisekunden – **System.DateTime**
- Veränderung der Geschwindigkeit der Controller und des HMDs – **3 · Vector3**
- Geschwindigkeit der Controller und des HMDs – **3 · Vector3**
- Position der Controller und des HMDs – **3 · Vector3**
- Rotation der Controller und des HMDs – **3 · Vector3**
- Position der Daumen auf den Thumbpads, falls verfügbar, sonst „0“ – **2 · Vector2**
- Die aktuellen drei Wortvorschläge – **3 · String**
- Die getätigte Eingabe, falls verfügbar, sonst „_“ – **String**

Das Fettgedruckte hinter den Einträgen gibt den Datentyp, bzw. Vielfache des Datentyps an. Die Veränderung der Geschwindigkeit dient der Approximation der Beschleunigung. Die Daten werden in dem *DataCollector*-Skript gesammelt, welches Teil des *DataCollector*-Prefabs ist.

4.10 Installation, Nutzerdokumentation

Zur Verwendung der Verfahren werden zunächst die Unity Engine³² sowie Steam³³ benötigt. Um Fehler bei der Portierung des Projektes zu anderen Versionen zu entgehen, sollte die Unity Engine vorzugsweise in der Version 2018.2.3f1 verwendet und auf dem Betriebssystem Microsoft Windows ausgeführt werden. Steam wird benötigt, um die für die Vive nötige Schnittstelle SteamVR nutzen zu können.

Es sollte ein Rechner benutzt werden, der leistungsstark genug ist, um eine Framerate von 90 Bildern pro Sekunde für das HMD zu erreichen und die Verfahren uneingeschränkt nutzen zu können. Zur Ausführung in der VR wird eine HTC Vive benötigt, für deren Einrichtung das Vive-Setup (2018) genutzt werden kann.

³²<https://unity3d.com/de/get-unity/download> Letzter Zugriff: 01.10.2018

³³<https://store.steampowered.com/about/> Letzter Zugriff: 01.10.2018

Das Unity-Projekt befindet sich auf der beigefügten *DVD1* in */immersivedataexploration/*. Zur Verwendung muss dieses in einen beschreibbaren Datenträgerpfad kopiert werden. Die gekürzten Fassungen der N-Gramme der Studie sind unter *./Assets/Resources/WordPrediction/* gespeichert und die während der Hauptstudie durchgeführten Diktate befinden sich unter *./Assets/Dictation/*. Die Diktate der Pilotstudie befinden sich in dessen Unterordner *./pilotstudie/*. Unter *./Logs/readme.txt* befindet sich eine Übersicht der Angaben des Formats der aufgezeichneten Daten.

Auf *DVD1* befinden sich außerdem die Archive */ngrams.rar* und */pilotstudylogs.rar*. Erstes beinhaltet die vollständige Version der bereinigten N-Gramme, während Letzteres die bei der Pilotstudie aufgezeichneten, pseudonymisierten Daten enthält.

Der Inhalt von *DVD2* besteht aus */mainstudylogs.rar* sowie */extracted-text.rar*. Dabei handelt es sich um die aufgezeichneten, pseudonymisierten Daten der Hauptstudie und um die unverarbeitete Fassung der von dem *WikiExtractor.py* ausgegebenen Textdateien des Wikimedia Dumps.

Zur Entpackung der Archive wird ein Datenkompressionsprogramm benötigt welches *.rar*-Dateien öffnen kann, wie z.B. 7-Zip³⁴ oder WinRAR³⁵.

DVD1 beinhaltet außerdem die Python-Programme. Diese befinden sich unter */WikiExtractor/* und */N-Gram-Generator/*. Zur Ausführung der Skripte wird Python 2.7³⁶ benötigt. Die Erstellung weiterer N-Gramme ist nicht notwendig, doch sollte dies dennoch ausgeführt werden, so ist zunächst die *WikiExtractor.py* gemäß den Instruktionen der zugehörigen *README.md* auszuführen.

Die resultierenden Textdateien sollten von zusätzlichen Tags, Leerzeilen usw. bereinigt werden, zu einer Textdatei zusammengefasst werden und können daraufhin von dem *N-Gram-Generator.py* genutzt werden um neue N-Gramme zu erstellen. Die so entstandenen N-Gramme müssen daraufhin entsprechend dem Format der mitgelieferten N-Gramme angepasst werden und können schließlich im Unity-Projekt verwendet werden.

Die Szenen, in denen die Verfahren implementiert sind befinden sich unter *./Assets/Scenes/TextInput/*. In den Szenen enthalten ist jeweils das *DictationController-GameObject*, in welchem mit *Logname* der Name der Logs angepasst werden kann. Mit den Variablen *Dictation_id* kann ein Diktat ausgewählt werden und mit *ActivateLogging* kann bestimmt werden, ob das Logging aktiviert werden soll. *TimeRunning* kann während des Ablaufs der Trainingseinheit zur Pausierung der Zeit genutzt werden und mit *PreparationTime* kann angegeben werden, wie viel Vorbereitungszeit der Nutzer zwischen dem Wechsel seiner Trainingsphase und der Diktatphase zur Verfügung hat.

Die Verwendung der Eingabeverfahren im Einzelnen geht dabei aus dem Kapitel Konzeption sowie Abschnitt 4.2, Abschnitt 4.3 und Abschnitt 4.5 hervor.

Zur Verwendung der universellen Spracherkennung muss zudem die personalisierte Spracherkennung unter Windows aktiviert sein. Diese ist erreichbar über Windows-Einstellungen → Erleichterte Bedienung → Spracherkennung → Zusätzliche Spracherkennungs-Einstellungen → Datenschutzeinstellungen für Sprache, Freihand und Eingabe → Spracherkennungsdienste und Eingabevorschläge aktivieren.

³⁴<http://www.7-zip.de/> Letzter Zugriff: 11.10.2018

³⁵<https://www.winrar.de/> Letzter Zugriff: 11.10.2018

³⁶<https://www.python.org/downloads/> Letzter Zugriff: 01.10.2018

4.11 Zusammenfassung

Zusammenfassend lässt sich sagen, dass die Verwendung der Unity Engine während der Arbeit keine größeren Probleme bereitet hat und durch die Unity-Assets schnell eine Grundlage für die Implementierung der Verfahren geschaffen werden konnte.

Die Sprachverarbeitung eignet sich generell als Assistenzsystem, wird jedoch zugunsten der Aussagekraft der Vergleiche zwischen den Verfahren nicht in der Studie benutzt, da dieses System eher wie ein eigenes Eingabeverfahren statt wie ein Assistenzsystem genutzt werden könnte. Die Windows-Speech-Recognition liefert scheinbar gute Ergebnisse, insbesondere bei der komplexen Aufgabe der universellen Interpretation.

Für das Assistenzsystem der Wortvorschläge und Wortvorhersagen wurde ein System entwickelt, für welches eine Priorisierung zwischen Umfang und Geschwindigkeit gewählt werden kann. Es ist zu erwarten, dass bei der Verwendung der gesamten Google Books Bibliothek ein ähnlich gutes, wenn nicht besseres Ergebnis erzielt werden könnte, als es Nutzer von modernen Mobilgeräten gewohnt sind.

Zu den Verfahren selbst lässt sich festhalten, dass die ausgearbeiteten Konzepte umfangreich übernommen und umgesetzt werden konnten. Die Verfahren wurden, wie vorgesehen, um die Assistenzsysteme erweitert und durch Hinzunahme der Bedienelemente der Controller um weitere Funktionen bereichert. Die Arbeit an dem Handschrift-Verfahren wurde bei diesem Prozess eingestellt, da die sich abzeichnende Geschwindigkeit weit unter der Geschwindigkeit der anderen Verfahren lag und auch technische Aspekte die Möglichkeiten des Verfahrens beschränkt haben.

Außer der Verfahren und Assistenzsysteme wurde zudem ein Programm zur Extraktion eines Wikimedia Dumps modifiziert, ein weiteres Programm zur Erstellung von N-Grammen aus dem extrahierten Text entwickelt, Trainingsvorbereitungen in Form von einem eingebetteten Browser und der Nutzung von Comics getroffen, verschiedene Arten von Diktaten für die Pilot- und Hauptstudie erstellt und die Aufzeichnung der Daten implementiert.

5 Evaluation

Es folgt die Evaluation der Studien. Dabei ist zu erwähnen, dass in den Tabellen und Listen, auf die im Anhang verwiesen wird, die Darstellung „ n “ bedeutet, dass eine sinngemäß äquivalente Antwort n -mal gegeben wurde. Dabei wird die als erste erfasste Formulierung übernommen und nachfolgende Antworten auf sinngemäße Übereinstimmung mit bereits aufgenommenen Antworten überprüft.

Außerdem tauchen in den Tabellen und Listen der Hauptstudie zu Teilen Pfeile (->) auf. Diese zeigen eine Veränderung gegenüber einer vorherigen Trainingseinheit auf. Die Teilnehmer haben bei jeder Trainingseinheit der Hauptstudie denselben Fragebogen erhalten und konnten so frühere Eindrücke revidieren und verdeutlichen, dass sich ein Eindruck verändert hat.

Die meisten Abbildungen und Tabellen der Evaluation wurden dem Anhang hinzugefügt, um die Übersichtlichkeit des Kapitels zu wahren, abgesehen von solchen Tabellen, die bereits anschaulich die Ergebnisse der Studien zusammenfassen.

5.1 Pseudonymisierung

Alle Daten, die während der Studien gesammelt wurden, sind pseudonymisiert worden. Die Pseudonyme der Teilnehmer tragen die Kennung VP (Versuchsperson) und sind nummeriert von VP1 bis VP8 zur Kennung der Teilnehmer der Pilotstudie sowie VP11 bis VP20 zur Kennung der Teilnehmer der Hauptstudie.

Es wurden keine persönlichen oder sensiblen Daten aufgezeichnet. Die Daten wurden ausschließlich auf einem privaten Computer und durch ein privates Passwort geschützte Universitätscomputer gespeichert, um den Zugriff Dritter zu verhindern.

5.2 Fehlerquote

Zur Bestimmung der Fehlerquote wird zwischen dem gesamten geschriebenen Text und dem vorgegebenen Diktatext die Levenshtein-Distanz berechnet und die Fehlerquote durch das Verhältnis der Distanz zur Länge des Textes bestimmt. Selbstverständlich ist bei diesem Vergleich – anders als bei der Berechnung der Levenshtein-Distanz im Assistenzsystem der Wortvervollständigung – ein Unterschied in der Textlänge ebenso ein Fehler, der die Distanz erhöht, wie es im originalen Algorithmus vorgesehen ist.

Da die Diktate immer in ganzen Sätzen (bzw. in der Pilotstudie als Gruppe von 10 Wörtern) diktieren werden und die Versuchspersonen mit dem Ende des Diktates üblicherweise Sätze nicht exakt abschließen, wird aus dem Vergleichstext nur der Anteil zur Berechnung hinzugezogen, bis zu dem die Versuchspersonen während des Diktates geschrieben haben. Bei der Berechnung der Distanz wurden auch solche Fälle als Fehler gewertet, bei denen Wörter der Diktate nach der neuen deutschen Rechtschreibung eine andere Schreibweise haben, so z.B. „dass“ statt „daß“, oder auch wenn nach der Nutzung eines Wortvorschlages hinter dem automatisch hinzugefügten Leerzeichen ein weiteres Leerzeichen geschrieben wurde.

5.3 Pilotstudie

Bei der Pilotstudie wurde an zwei aufeinanderfolgenden Tagen je eine Verfahrensgruppe durch acht Teilnehmer getestet. Die Zuweisung der Reihenfolge der Verfahren zu den Versuchspersonen, erfolgte randomisiert, jedoch mit der Beschränkung, dass mit beiden Gruppen, sowie mit jedem Verfahren innerhalb der Gruppen, möglichst gleich häufig begonnen wurde.

In der Pilotstudie betrug das durchschnittliche Alter 24 Jahre, wobei die jüngste Person 23 und die älteste Person 26 Jahre alt waren. Die Personen gaben an, mit sieben bis zehn Fingern zu schreiben. Der Durchschnitt liegt bei etwa 9 Fingern. Zwei Teilnehmer waren weiblich, sechs Teilnehmer männlich. Ein Teilnehmer war linkshändig, sieben Teilnehmer rechtshändig. Dem Teilnehmer mit dem Pseudonym VP8 war es leider aus privaten Gründen nicht möglich, die Verfahrensgruppe der Controller-basierten Verfahren zu testen, daher entfallen diese Daten. Alle nachfolgenden Berechnungen und Schlüsse berücksichtigen diesen Faktor.

Jedem Verfahren wurden in der Pilotstudie 15 Minuten gewidmet, davon 10 für das Einüben des Verfahrens und weitere 5 für das Schreiben des Diktates. Für die Pilotstudie wurden die acht Diktate verwendet und ebenso wurde die Zuteilung der Diktate zu den Verfahren randomisiert, mit der Beschränkung, dass jedes Diktat mit jedem Verfahren möglichst gleich häufig verwendet wird und jedes Diktat von jeder Person nur einmal geschrieben wird. Das Gaze-Verfahren wird zugunsten der Übersichtlichkeit nachfolgend der Verfahrensgruppe der Controller-basierten Verfahren zugeordnet.

5.3.1 Ergebnisse

Tabelle 5.1 zeigt die Tastenanschläge, die die Teilnehmer während des Schreibens der Diktate in der Pilotstudie erzielt haben. Es ist zu sehen, dass mit dem Drumstick-Verfahren durchschnittlich mehr als doppelt so viele Tastenanschläge durchgeführt wurden, wie mit der Thumbpad-Tastatur, welche unter den Thumbpad-basierten Verfahren am besten abgeschnitten hat.

Tabelle 5.1: Übersicht der Tastenanschläge in der Pilotstudie

	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	∅
Tastatur	228	172	217	189	220	196	205	191	202,3
Kreise	159	137	131	176	155	143	152	135	148,5
Gesten	114	87	103	85	111	108	96	106	101,3
Cluster	136	114	142	151	141	131	115	102	129
Drumstick	375	597	502	433	350	358	225	-	405,5
Touch	396	428	517	441	311	208	409	-	387
Laser	273	290	373	282	250	273	193	-	276,3
Gaze	145	143	155	166	138	90	151	-	141,1

Tabelle 5.2 zeigt die Textlängen, die mit den Verfahren erreicht wurden. Die durchschnittlich erzielten Textlängen der Laser- und Gaze-Verfahren sind höher als die durchschnittliche Anzahl der Tastenanschläge mit diesen Verfahren, während die Textlängen bei den anderen Verfahren leicht bis deutlich unterhalb der Anzahl der Tastenanschläge liegen.

Der Unterschied ist in die entgegengesetzte Richtung am stärksten bei dem Verfahren der Thumbpad-Kreise ausgeprägt. Mit diesem wurde mit den zweitmeisten Tastenanschlägen in der Gruppe der Thumbpad-Verfahren nur 85,72% der Tastenanschläge an Text produziert.

Insgesamt schneidet jedoch auch hier das Drumstick-Verfahren mit dem besten Ergebnis von durchschnittlich 377,1 Zeichen ab und aus der Gruppe der Thumbpad-Verfahren liefert erneut die Thumbpad-Tastatur mit durchschnittlich 198,5 Zeichen die besten Ergebnisse. Die Fehlerquote im geschriebenen Text bewegte sich bis auf sehr wenige Ausreißer stets zwischen 0% bis 3%, während einzelne Teilnehmer scheinbar Probleme mit der Fortsetzung des Diktates hatten und zwischenzeitig ganze Abschnitte der Diktate ausgelassen haben.

Tabelle 5.2: Übersicht der Textlängen in der Pilotstudie

	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	Ø
Tastatur	104	212	230	168	277	212	188	197	198,5
Kreise	126	113	107	218	139	105	130	80	127,3
Gesten	91	141	99	153	93	89	67	32	95,6
Cluster	118	106	178	130	103	127	51	114	115,9
Drumstick	400	195	400	326	396	389	534	-	377,1
Touch	311	443	467	157	306	375	502	-	365,9
Laser	368	380	378	258	321	267	375	-	335,3
Gaze	241	211	248	236	118	150	185	-	198,4

Die Abbildungen A.1 bis A.8 zeigen die Tastenanschläge der Versuchspersonen mit den jeweiligen Verfahren während der Diktate im Detail. Es ist zu erkennen, dass nur selten Unregelmäßigkeiten aufgetreten sind und die meiste Zeit die Teilnehmer mit einer relativ konstanten Geschwindigkeit schreiben konnten. Den Trackingdaten kann entnommen werden, dass Phasen ohne Tastenanschläge tendenziell dann auftreten, wenn das Geschriebene in der Textzeile gelesen wird.

Abbildung A.9 zeigt die Mittelwertfunktionen der Abbildungen A.1 bis A.8, also die durchschnittlichen Tastenanschläge der Teilnehmer während der Diktate. Der Abbildung kann entnommen werden, dass das Drumstick-Verfahren und das Touch-Verfahren weitaus schneller bedient wurden. Außerdem fällt auf, dass mit den Thumbpad-Verfahren im Schnitt erheblich schlechtere Ergebnisse erzielt wurden, als die Controller-basierten Verfahren.

Die vorliegenden Ergebnisse zeigen deutlich, dass in allen technischen Betrachtungspunkten das Drumstick-Verfahren die besten Resultate geliefert hat, während aus der Gruppe der Thumbpad-Verfahren mit der Thumbpad-Tastatur die besten Ergebnisse erzielt wurden.

5.3.2 Auswertung Pilotfragebogen

Die Teilnehmer haben nach den beiden Teilnahmen an der Studie jeweils einen Fragebogen ausgefüllt. Eine Kopie des Fragebogens kann Unterabschnitt A.3.1 entnommen werden. Dabei erhielten die Teilnehmer an dem Tag, an dem sie die Controller-basierten Verfahren getestet haben, den dargestellten Fragebogen, während sie an dem anderen Tag eine abgewandelte Version erhalten haben, die sich lediglich in der Bezeichnung der Verfahren innerhalb des Fragebogens unterscheidet.

Tabelle A.1 zeigt, dass die meisten Nutzer regelmäßig bis sehr häufig mittellange bis sehr lange Texte auf Tastaturen schreiben, wodurch von einer relativ hohen Vertrautheit mit dem QWERTZ-Layout ausgegangen werden kann.

Tabelle A.2 zeigt außerdem, dass ein Teil der Teilnehmer eher wenig mit dem Schreiben auf Mobilgeräten vertraut ist, während ein anderer Teil angibt, dieses häufig bis sehr häufig zu tun. Dadurch kann für diese Teilnehmer angenommen werden, dass sie leichter mit dem Umgang des Schreibens auf dem Thumbpad zurechtkommen werden, während für die übrigen Teilnehmer ein geringerer Bezug zu Bildschirmtasten von Mobilgeräten existiert. Tabelle A.3 zeigt, dass die Nutzer nur wenig an die Nutzung der VR gewohnt sind und Tabelle A.4 verdeutlicht, dass zuvor keiner der Teilnehmer regelmäßig in der VR Texte geschrieben hat.

Durch diese Antworten lässt sich bereits vermuten, dass den Teilnehmern die Bedienung der weniger intuitiven Verfahren schwer gefallen ist. Dies spiegelt sich in den Ergebnissen von Tabelle 5.1 wider. So zeigt Tabelle A.5 auf, dass die Nutzer nach eigener Empfindung die Bedienung des Touch- und Drumstick-Verfahrens, sowie das Verfahren der Thumbpad-Tastatur am schnellsten erlernen konnten.

Tabelle A.6 zeigt die Meinungen der Teilnehmer zu den Vor- & Nachteilen der Thumbpad-Verfahren¹ und Tabelle A.7 die Meinungen zu den Controller-Verfahren. Hier fallen insbesondere das Gaze- und das Gesten-Verfahren eher negativ auf.

In Tabelle A.8 werden die Versuchspersonen nach Verbesserungsvorschlägen gefragt, die zu Teilen für die Hauptstudie übernommen wurden, wie z.B. die einstellbare Länge der Drumsticks. Tabelle A.9 zeigt die präferierten Verfahren der Teilnehmer. Auch hier schneiden die Thumbpad-Tastatur sowie das Touch- und das Drumstick-Verfahren am besten ab. Nach Tabelle A.10 sind hingegen das Gesten- und das Gaze-Verfahren am wenigsten beliebt. Die meisten Nutzer haben in Tabelle A.11 diesen beiden Verfahren außerdem die höchste und der Thumbpad-Tastatur die geringste Fehlerquote zugeschrieben. Die Schreibgeschwindigkeit wurde jedoch im Schnitt bei dem Touch-Verfahren am höchsten eingeschätzt, wie Tabelle A.12 zeigt.

Des Weiteren wurden die Teilnehmer, wie in Tabelle A.13 zu sehen ist, hin und wieder technisch beeinträchtigt. VP6 gibt sogar an, sehr stark technisch beeinträchtigt worden zu sein. Dies spiegelt sich jedoch nicht in den Ergebnissen wider. Nur ein weit unterdurchschnittliches Ergebnis bei dem Gaze-Verfahren in Abbildung A.2 fällt auf. Dabei handelt es sich jedoch um ein Verfahren aus der anderen Verfahrensgruppe, für die von VP6 eine technische Beeinträchtigung von 0 angegeben wurde. Welche technischen Aspekte die Teilnehmer genau beeinträchtigt hat, kann Tabelle A.14 entnommen werden. VP1 fühlte sich vor allem durch die schlechte Benutzbarkeit der Grip-Tasten unabhängig von dem Verfahren beeinträchtigt.

Die geistige Anforderung der Aufgabe wurde, wie in Tabelle A.15 zu sehen ist, bei den Thumbpad-Verfahren, aufgrund der Gewöhnung an die unterschiedlichen Tastenanordnungen, tendenziell höher eingeschätzt, während, wie Tabelle A.16 zeigt, die körperliche Anstrengung tendenziell bei den Controller-basierten Verfahren größer ist, obgleich auch diese zumeist ohne große körperliche Anstrengungen genutzt werden konnten.

Tabelle A.17 zeigt, dass bei den Thumbpad-Verfahren nur zwei Teilnehmer mit ihrer Schreibgeschwindigkeit zufrieden waren, während bei den Controller-Verfahren fünf von sieben Teilnehmern schon nach der 15-minütigen Benutzung die Aufgabe nach eigener Einschätzung erfolgreich bewältigt haben.

¹Bei fehlender Angabe, ob es sich bei einer Antwort um einen Vor- oder Nachteil handelt, wurde diese Zuordnung nach bestem Wissen und Gewissen vorgenommen.

In Tabelle A.18 ist zudem zu sehen, dass die meisten Teilnehmer sehr wenig bis gar nicht von dem Umgang mit den Verfahren frustriert waren und lediglich das Gaze-Verfahren von zwei Teilnehmern als frustrierend bzw. stressend empfunden wurde. In optimierter Form würden die meisten Teilnehmer laut Tabelle A.19 gerne die Thumbpad-Tastatur benutzen, bzw. das Touch- oder Drumstick-Verfahren bei den Controller-basierten Verfahren. Erneut wird das Touch-Verfahren tendenziell angenehmer empfunden, obgleich die Daten gezeigt haben, dass mit dem Drumstick-Verfahren bessere Ergebnisse erzielt wurden.

In Tabelle A.20 und Tabelle A.21 konnten die Teilnehmer außerdem nennen, ob Ihnen etwas besonders negativ aufgefallen ist, oder sie sonstige Anmerkungen, Kritik oder Ideen haben.

Insgesamt hat sich gezeigt, dass aus technischer Sicht je Verfahrensgruppe die Thumbpad-Tastatur und das Drumstick-Verfahren am besten für die Fortsetzung in der Hauptstudie geeignet sind, während aus der Sicht der Teilnehmer die Wahl auch auf die Thumbpad-Tastatur fällt, jedoch das Touch- etwas besser abschneidet als das Drumstick-Verfahren.

5.4 Hauptstudie

Bei der Hauptstudie wurden das Drumstick-Verfahren, sowie die Thumbpad-Tastatur von zehn Personen für fünf Trainingseinheiten mit einer Dauer von je 30 Minuten pro Verfahren eingeübt und getestet. Es wurde versucht, die Personen jeden Tag zur gleichen Uhrzeit teilnehmen zu lassen, jedoch war dies aus privaten Gründen nicht bei allen Teilnahmen möglich. Daher wurden die Trainingseinheiten nach bester Möglichkeit zeitlich aufgeteilt, der Großteil der Teilnehmer hat regulär täglich zur gleichen Uhrzeit teilgenommen. Die Teilnahmezeiten können den Namen und Inhalten der aufgezeichneten Daten entnommen werden.

Zur Übung der Verfahren wurden die in Abschnitt 4.7 erwähnten PhD Comics verwendet. Während der 25-minütigen Trainingsphase bestand die Aufgabe der Teilnehmer darin, die Comics vom Englischen ins Deutsche zu übersetzen. Die Teilnehmer haben mitgeteilt bekommen, dass nicht die exakte Übersetzung der Comics von Bedeutung ist, sondern dass die Eingabeverfahren möglichst durchgehend trainiert werden sollen. Die Wahl fiel dabei auf Comics, da die Teilnehmer durch diese weniger gelangweilt sein sollen, als es beim bloßen Abschreiben von beliebigem Text der Fall gewesen wäre. Außerdem unterscheiden sich die Comics sowohl thematisch als auch von der Sprachwahl her von den Diktaten, sodass potentiellen Erwartungshaltungen der Teilnehmer an die Diktate nicht entsprochen wird.

Das durchschnittliche Alter betrug 25,2 Jahre, wobei die jüngste Person 23 und die älteste Person 30 Jahre alt waren. Die Personen gaben an, mit sechs bis zehn Fingern auf Tastaturen zu schreiben, mit einem Durchschnitt von 8,6 Fingern. Vier Teilnehmer waren weiblich, sechs Teilnehmer männlich. Alle Teilnehmer waren rechtshändig.

Der Fragebogen der Hauptstudie kann in Unterabschnitt A.3.2 eingesehen werden. Die Antworten der Fragen lassen sich in Abschnitt A.4 und Abschnitt A.6 im Anhang finden. Außerdem wurde die deutsche Fassung des NASA TLX Fragebogens (NASA, 2018) verwendet. Bei diesem Fragebogen handelt es sich um eine Beurteilung der Belastung (Workload), die von der gestellten Aufgabe ausgeht. Allerdings wird auf die Gewichtung der Kategorien verzichtet, da Hart (2006) zeigt, dass die Gewichtung üblicherweise keinen nennenswerten Einfluss auf das Korrelationsverhältnis der Fragen liefert. Daher wird in dieser Arbeit der RAW TLX-Score berechnet.

Der RAW TLX-Score dient auch bei Mott et al. (2017), Grubert et al. (2018) und Nieuwenhuizen (2017) dafür, zu bemessen, wie belastet die Teilnehmer von der Durchführung eines Experimentes oder der Verwendung einer Technologie sind.

5.4.1 Ergebnisse

Zur Übersichtlichkeit werden die den Diagrammen zugehörigen Daten als Teil der Abbildung aufgelistet. So zeigt zunächst Abbildung A.10 die Entwicklung der Tastenanschläge, die die Versuchspersonen bei den Trainingseinheiten mit dem Drumstick-Verfahren erreicht haben. Die gepunktete Linie zeigt die durchschnittlichen Tastenanschläge je Trainingseinheit (T). Die Entwicklung ist größtenteils positiv, vereinzelt gibt es jedoch Einbrüche. Generell steigt der Durchschnitt gerade zu Beginn stark an und flacht von T4 zu T5 am meisten ab. Gegenüber der jeweils vorherigen Trainingseinheit stieg die durchschnittliche Leistung bei T2 um 12,1%, bei T3 um 6,48%, dann um 8,9% und zu T5 um weitere 1,22%.

Aus diesen Daten lässt sich mithilfe der in Anhang A beschriebenen logarithmischen Regression eine Trend-Funktion bilden. Die Regression der Daten führt zu der Funktion $y = 98,86 \cdot \ln(x) + 531,68$, wobei x der Nummer der Durchführungen und y den erreichten Tastenanschlägen entspricht. Die Residuenquadratsumme dieser Funktion liegt bei $RSS = 213,59$ und das Bestimmtheitsmaß bei $R^2 = 99,9892\%$. Die Funktionskurve ist in Abbildung A.16 zu sehen und Abbildung A.17 zeigt die Residuen.

Im Falle des Thumbpad-Verfahrens zeigt Abbildung A.11 eine noch deutlichere positive Entwicklung, mit abflachender Steigung für den Durchschnittswert. Vereinzelt Durchführungen resultierten auch hier in weniger Tastenanschlägen als bei der vorherigen Durchführung, jedoch stieg die Leistung durchschnittlich um 14,18% bei T2, 12,98% bei T3, 9,8% bei T4 und weitere 6,5% zu T5. Die zugehörige Funktionskurve ist in Abbildung A.18 zu sehen und Abbildung A.19 zeigt die Residuen.

Hierbei zeichnet sich noch deutlicher ein logarithmischer Trend ab. Die Regression der Daten führt in diesem Fall zu der Funktion $y = 86,82 \cdot \ln(x) + 310,71$. Die Residuenquadratsumme dieser Funktion liegt bei $RSS = 156,8$ und das Bestimmtheitsmaß bei $R^2 = 99,9801\%$.

Abbildung A.12 und Abbildung A.13 zeigen die während der Diktate erreichten Textlängen. Auch diese steigen tendenziell bei jeder Trainingseinheit an, jedoch ist in beiden Fällen die durchschnittlich erreichte Textlänge bei T5 geringer als bei T4. Dies ist darauf zurückzuführen, dass sich die Diktate $d8$ und $d9$, die bei der letzten Durchführung geschrieben wurden, von den übrigen Diktaten $d0$ bis $d7$ unterscheiden und die Teilnehmer bei den Diktaten der letzten Durchführung durchschnittlich schlechtere Wortvorschläge erhielten, wie in Tabelle 5.3 ersichtlich ist. So betrug die mögliche Zeichensparnis bei den Diktaten $d0$ bis $d7$ im Schnitt 26,23% und liegt damit um etwa 3,72% höher als die 22,51% der letzten Durchführung. Die im Gegensatz dazu leicht erhöhte Häufigkeit, mit der Wörter korrekt vorgeschlagen wurden, ist hingegen bei der letzten Durchführung minimal höher. Dies bedeutet zusammengefasst, dass bei $d8$ und $d9$ die Vorschläge etwas häufiger genutzt werden konnten, jedoch erst später erschienen und eine kleinere Ersparnis mit sich brachten als bei den restlichen Diktaten.

Die Berechnung der von MacKenzie (2002) vorgestellten KSPC-Metrik (Keystrokes per Character) zur Bestimmung der Effizienz von Texteingaben setzt sich aus dem Verhältnis der getätigten Tastenanschläge geteilt durch die Länge des produzierten Textes zusammen. Diese beträgt zur letzten Trainingseinheit 1,033 bei dem Drumstick-Verfahren und 1,0765 bei der Thumbpad-Tastatur.

Zur Berechnung der Werte in Tabelle 5.3 wurden die ersten 1.000 Zeichen der Diktate verwendet, da eine vollständige Auswertung der Diktate nicht die tatsächlichen Verhältnisse während der Studie widerspiegeln würde. Dabei wurden nur exakt übereinstimmende Wortvorschläge gezählt.

Plurale, deren Singular vorgeschlagen wird, Vorschläge, deren Suffix sich minimal von der Diktatvorgabe unterscheidet sowie Wörter die wiederum aus mehreren Wörtern bestehen wie „Beobachtungsdaten“ werden dabei nicht berücksichtigt, obwohl deren Benutzung in der Praxis eine zusätzliche Zeichensparnis bringen könnte. Ferner wurde für die Auswertung der Vorschläge eine gekürzte Fassung der N-Gramme verwendet. Es ist anzunehmen, dass unter Verwendung der vollständigen N-Gramme sowohl die mögliche Zeichensparnis als auch der Anteil korrekt vorgeschlagener Wörter erhöht werden kann.

Tabelle 5.3: Übersicht der Hilfestellung durch Wortvorschläge in der Hauptstudie

Diktat	Durchführung-zuordnung	Mögliche Zeichensparnis	Häufigkeit korrekt vorgeschlagener Wörter
D0	T1 - T4	30,12%	52,94%
D1	T1 - T4	21,87%	40,98%
D2	T1 - T4	22,65%	36,9%
D3	T1 - T4	26,52%	47,65%
D4	T1 - T4	27,2%	49,3%
D5	T1 - T4	25%	44,5%
D6	T1 - T4	29,2%	46,53%
D7	T1 - T4	27,3%	46,94%
D8	T5	20,72%	41,72%
D9	T5	24,3%	50,33%

D = Diktatnummer

Abbildung A.14 zeigt die Fehlerquote des Drumstick-Verfahrens und Abbildung A.15 die Fehlerquote des Thumbpad-Verfahrens. Bei beiden Verfahren gibt es keine nennenswerte Entwicklung der Fehlerquoten. Diese lag durchschnittlichen bei weniger als 1,5%. In der Literatur wird die Fehlerquote oft vernachlässigt (Wobbrock, 2007), daher zeigt Bhattacharya et al. (2008) die Wichtigkeit der Berücksichtigung der Fehlerquote beim Messen von Schreibgeschwindigkeiten auf virtuellen Tastaturen auf.

Außerdem wurden die Standardabweichungen σ der Tastenanschläge, Textlängen und Fehlerquoten für beide Verfahren zu jeder Trainingseinheit berechnet (Tabelle 5.4 und Tabelle 5.5). Diese zeigen eine Vergrößerung der Standardabweichung bei den erzielten Tastenanschlägen, was eine unterschiedliche Lerneffizienz der Teilnehmer bedeutet. Die Standardabweichungen der Textlängen stiegen bei dem Drumstick-Verfahren tendenziell an, während bei dem Thumbpad-Verfahren keine deutliche Entwicklung sichtbar ist. Außerdem wurden die Standardabweichungen der Fehlerquote in Prozent angegeben, die ebenso wie die durchschnittliche Fehlerquote keine strukturelle Entwicklung aufzeigt. Beim technischen Vergleich der erreichten Tastenanschläge und Textlängen der Verfahren wird deutlich, dass das Drumstick-Verfahren überlegen ist.

Nicht nur sind die erreichten Tastenanschläge und Textlängen bei der letzten Durchführung um über 50% höher, sondern auch die logarithmische Regression resultiert für dieses Verfahren in einer höheren Erwartung für die langzeitige Benutzung.

Tabelle 5.4: Übersicht der Standardabweichungen beim Thumbpad-Verfahren in der Hauptstudie

Trainingseinheit	Anschläge	Textlänge	Fehlerquote
T1	48,85	51,10	1,62%
T2	77,17	68,76	1,49%
T3	100,18	53,08	0,53%
T4	129,37	95,12	1,27%
T5	148,57	44,76	1,66%

Tabelle 5.5: Übersicht der Standardabweichungen beim Drumstick-Verfahren in der Hauptstudie

Trainingseinheit	Anschläge	Textlänge	Fehlerquote
T1	55,11	50,01	0,72%
T2	95,20	84,73	1,77%
T3	125,43	84,23	1,42%
T4	173,27	96,42	0,81%
T5	178,17	89,73	1,01%

5.4.2 Auswertung Hauptfragebogen

Da der subjektive Eindruck der Teilnehmer ebenso wegweisend für die Bestimmung des besten Verfahrens ist wie die technische Betrachtung, folgt nun die Auswertung des Fragebogens, den die Teilnehmer nach jeder Trainingseinheit erneut zur Anpassung und Erweiterung ihrer Antworten erhalten haben.

Tabelle A.1 zeigt, dass die Teilnehmer der Hauptstudie gegenüber denen der Pilotstudie eine bessere Gewöhnung daran haben, auf Tastaturen zu schreiben. Auch die Vertrautheit, auf Mobilgeräten zu schreiben ist, wie in Tabelle A.2 zu sehen ist, größer. Die Nutzer waren außerdem, wie Tabelle A.3 entnommen werden kann, im Durchschnitt bereits vor der Studie etwas gewohnter an die Benutzung der VR und drei der Teilnehmer gaben bei Tabelle A.4 an, wiederholt Texte in der VR geschrieben zu haben. Tendenziell wird daher von einer etwas erhöhten Vertrautheit mit verschiedenen Eingabemethoden und der besseren Bekanntheit des Assistenzsystems der Wortvorschläge ausgegangen.

In Tabelle A.22 sind die Antworten auf die Frage zu sehen, welche Vor- & Nachteile die Teilnehmer dem Drumstick-Verfahren zuschreiben². Die Handhabung und Benutzung wird dort größtenteils positiv beschrieben, jedoch sei das Verfahren auf Dauer anstrengend zu verwenden, insbesondere da es im Stehen ausgeführt werden müsse.

Bei dem Thumbpad-Verfahren hingegen liegt, wie in Tabelle A.23 entnommen werden kann, die Kritik vor allem bei der Präzision, mit der Tasten getroffen werden müssen. Wiederholt wird positiv empfunden, dass das Verfahren körperlich weniger anstrengend ist.

²Bei fehlender Angabe, ob es sich bei einer Antwort um einen Vor- oder Nachteil handelt, wurde diese Zuordnung nach bestem Wissen und Gewissen vorgenommen.

Die Teilnehmer haben daraufhin die Frage beantwortet, welche Verbesserungen Sie sich an den getesteten Verfahren erhoffen. Die Ergebnisse dazu sind in Tabelle A.24 zu sehen und zeigen sehr verschiedene Ansätze auf. Die Vorschläge überschneiden sich nur selten. Einzig der Wunsch, dass bei der Verwendung des Drumstick-Verfahrens die Auswahl des Controllers, dessen Trigger gedrückt gehalten werden kann, um einen Großbuchstaben zu schreiben, beliebig sein sollte, wurde wiederholt geäußert.

Bei der direkten Gegenüberstellung der Verfahren für das Schreiben kurzer Texte (Tabelle A.25) gaben acht Teilnehmer an, dass ihnen das Drumstick-Verfahren besser gefällt und nannten dafür als Hauptgründe, dass es schnell bedienbar ist, einen größeren Spaßfaktor liefert und intuitiver bedient werden kann. Bei der Frage, welches Verfahren zum Schreiben langer Texte von den Teilnehmern bevorzugt wird, fielen in Tabelle A.26 die Ergebnisse ähnlich aus, jedoch mit einer leichten Tendenz zu dem Thumbpad-Verfahren, da dieses körperlich weniger Anstrengung bereitet.

Tabelle A.27 ist zu entnehmen, dass es zwischenzeitig zu kleineren technischen Problemen kam, die jedoch durch eine Pause des Trainings und einer Behebung des Problems keinen Einfluss auf die Durchführung der Diktate ausgeübt haben sollten. Bei den technischen Aspekten, die sich auf die Hardware selbst beziehen, wird wiederholt die Unschärfe von Texten genannt. Es könnte daher sein, dass HMDs mit einer höheren Auflösung bzw. einer höheren Pixeldichte zu besseren Ergebnissen führen können, da weniger Zeit für das Lesen von Wortvorschlägen und dem Geschriebenen benötigt wird.

In Tabelle A.28 befinden sich die Schätzungen der Teilnehmer hinsichtlich der Verbesserung der erzielten Schreibgeschwindigkeit mit den beiden Verfahren. Die Werte variieren stark, sind teilweise negativ und es fiel den Teilnehmern scheinbar generell schwer, eine gute Einschätzung der Werte zu geben. So hat beispielsweise VP6 wiederholt Verschlechterungen für die Entwicklung der Geschwindigkeit angegeben, jedoch ist die Zahl der Tastenanschläge bei jeder Trainingseinheit gestiegen, außer bei T4 mit dem Drumstick-Verfahren – genau die Durchführung, bei der eine Verbesserung von 30% geschätzt wurde.

Außerdem haben die Nutzer bei jeder Durchführung ihre Fehlerquote in Tabelle A.29 geschätzt. Auch hier weicht diese vor allem bei den hohen Angaben stark von den tatsächlichen Daten aus Abbildung A.14 und Abbildung A.15 ab. Generell ist jedoch bei den meisten Teilnehmern die geschätzte Fehlerquote im Laufe der Durchführungen gesunken, obgleich auch hier einzelne Ausnahmen existieren.

Am Ende des Fragebogens wurden die Teilnehmer nach sonstigen Anmerkungen, Kritik oder Ideen gefragt (Tabelle A.36). Dort gab es neben etwas Lob und dem Hinweis, dass Menschen mit beeinträchtigten Händen mit dem Drumstick-Verfahren dennoch gut schreiben könnten, keine weiteren Aussagen.

5.4.3 Auswertung TLX

Der NASA-TLX Fragebogen dient schon seit über 20 Jahren der Auswertung von Belastungen (Hart, 2006). Aus den in Abschnitt 5.4 genannten Gründen wird in dieser Arbeit der RAW TLX-Score berechnet. Der Wert ergibt sich aus der von den Versuchspersonen gewählten Angaben auf einer Skala von 0 bis 20, multipliziert mit 5, um einen Prozentwert zu erhalten. Die Durchschnitte wurden sowohl je Versuchsperson als auch je Trainingseinheit berechnet, um individuelle Hoch- und Niedrig-Einschätzungen zu identifizieren, als auch um die Entwicklung der Eindrücke der Personen festzuhalten.

Hat ein Benutzer zu einer Trainingseinheit keine neue Angabe getätigt, so wurde die Angabe der vorherigen Trainingseinheit übernommen. Der aussagekräftigste Wert ist der Durchschnitt der Versuchspersonen zur letzten Durchführung, da dieser widerspiegelt, mit welcher Erwartung, welcher Einstellung und welchem Eindruck gegenüber den Verfahren die Nutzer die Studie beendet haben.

Bei der Frage nach der geistigen Anforderung der Aufgabe (Tabelle A.30) fällt eine starke Varianz zwischen den Teilnehmern auf. Insgesamt, als Durchschnitt der Durchschnitte der Ergebnisse der einzelnen Durchführungen bzw. Personen – fortan Gesamtdurchschnitt genannt – ergibt sich ein Score von 58,9%. Am fünften Tag betrug die geistige Anforderung hingegen lediglich 46%. Im Verlauf der Tage ist eine deutliche Abnahme des Scores erkennbar. Die Nutzer konnten die Verfahren daher intuitiver nutzen und haben sich an die Funktionalitäten der Verfahren zunehmend gewöhnt.

Die körperliche Anstrengung aus Tabelle A.31 hat sich bei der vierten und fünften Durchführung ebenfalls verringert, mit einem Endergebnis von 42% gegenüber dem Gesamtdurchschnitt von 48,5%. Da die Verfahren allerdings durch mehr Übung nicht sonderlich weniger körperlich anstrengend werden, ist davon auszugehen, dass hektisches Bewegen und Revidieren von Falschbewegungen mit zunehmender Übung seltener geworden ist und daher diese Entwicklung begründet.

Tabelle A.32 zeigt die Ergebnisse zu der Frage, als wie hoch die zeitliche Anforderung empfunden wurde. Wie zu erwarten ist, wird bei einem Diktat, welches möglichst schnell abgeschrieben werden soll, ein zeitlicher Druck präsent empfunden, jedoch bleibt dieser über die Tage verteilt bei den meisten Teilnehmern konstant. Der Gesamtdurchschnitt liegt bei 39,7% und der Durchschnitt des letzten Tages bei 38,5%.

In Tabelle A.33, der Frage nach einer Selbsteinschätzung des erzielten Erfolges, ist eine steigende Entwicklung zu beobachten. Hierbei ist anzumerken, dass bei dieser Frage ein höherer Score besser ist, da die Skala nicht, wie in NASA (2018) beschrieben, invertiert wurde. Bis auf einen Einbruch an Tag drei, dessen Ursprung größtenteils durch eine völlig andere Einschätzung von VP2 rührt, entwickelt sich der Score im Laufe der Durchführungen von 55,5% bis zu 69%, mit einem Gesamtdurchschnitt von 59,8%. Dies weist erneut auf eine zunehmende Vertrautheit mit den Verfahren hin und lässt sich unter anderem auch aus den typischerweise mit jedem Durchgang besser werdenden Ergebnissen begründen.

Tabelle A.34 zeigt hingegen keine deutliche Entwicklung bei dem Grad der Anstrengung, den die Teilnehmer leisten mussten. Auch hier stammen die Schwankungen größtenteils von VP2. Der Score bleibt weitestgehend konstant; der Score der letzten Durchführung liegt bei 54,5% und der Gesamtdurchschnitt bei 58,2%.

Der Grad der Frustration hat, wie in Tabelle A.35 zu sehen ist, im Laufe der Studie abgenommen. Zunächst hält sich diese auf einem Plateau, doch bei den letzten beiden Durchführungen fällt sie ab. Der Gesamtdurchschnitt liegt bei 33,6%, während der Score bei der letzten Durchführung lediglich 22% betrug. Dies lässt ebenso auf die steigende Zufriedenheit und Vertrautheit mit den Verfahren schließen.

Zur Berechnung der Durchschnitte werden die in Tabelle A.33 gelisteten Scores auf der Skala invertiert. Der durchschnittliche TLX-Score der Gesamtdurchschnitte liegt somit bei 46,52% und der durchschnittliche Score der letzten Durchführung liegt bei 39%.

5.5 Zusammenfassung

Insgesamt zeichnet sich bei der Evaluation ab, dass eine bemerkenswerte Eingewöhnung der Verfahren stattgefunden hat. Bei noch längerer Trainingszeit wäre gemäß den Approximation der Entwicklung der Tastenanschläge mit noch besseren Ergebnissen zu rechnen. Jedoch wurden auch mit dieser begrenzten Trainingszeit bereits durchschnittlich 137,36 Tastenanschläge pro Minute während eines Diktates mit dem Drumstick-Verfahren erreicht; mit einem Spitzenwert von 166,4 Tastenanschlägen pro Minute.

Dabei wurden durchschnittlich 133 und maximal 163,8 Zeichen pro Minute produziert. Mit dem Thumbpad-Verfahren wurden durchschnittlich 90,88 und eine Höchsthäufigkeit von 105,6 Tastenanschlägen pro Minute erreicht. Die Textproduktionsgeschwindigkeit liegt bei durchschnittliche 84,42 und maximal 95,4 Zeichen pro Minute.

Da der Vergleich von Wörtern pro Minute hinsichtlich der Differenz der durchschnittlichen Wortlängen in verschiedenen Sprachen nicht aussagekräftig verwendbar ist, wird zur Auswertung der Ergebnisse gegenüber der Verfahren, deren durchschnittliche Geschwindigkeiten in Kapitel 2 genannt wurden das Verfahren von Mayzner (2017) verwendet. Für die englische Sprache wird daher von einer durchschnittlichen Wortlänge von 4,79 Zeichen ausgegangen. Dieser Wert berücksichtigt die Frequenz mit der Wörter im Englischen durchschnittlich verwendet werden. Durch Yamada (1980) wird jedoch gerade in älteren Publikationen eine durchschnittliche Wortlänge von 5 Zeichen angenommen. Da die durchschnittliche Wortlänge von Mayzner (2017) hingegen auf historischen Daten beruht, wird sie für die weiteren Berechnungen genutzt.

Gegenüber dem in Unterabschnitt 2.2.2 beschriebenen kontinuierlichem Zoomen mit Geschwindigkeiten von durchschnittlich 13 bis 24 Wörtern pro Minute, also 62,27 bis 114,96 Zeichen pro Minute, zeigt sich das Drumstick-Verfahren überlegen. Das Thumbpad-Verfahren liegt innerhalb dieses Wertebereiches. Die bei (Majaranta et al., 2009) maximal erreichten 95,3 Zeichen pro Sekunde liegen unterhalb der Maximalgeschwindigkeiten der vorgestellten Verfahren. Die erreichten Geschwindigkeiten liegen jedoch deutlich unterhalb der in Unterabschnitt 2.2.3 auf einer echten Tastatur durchschnittlich erzielten 219,96 Zeichen pro Minute. Das Virtual Notepad aus Unterabschnitt 2.2.4 liegt mit 22,98 Zeichen pro Minute weit abgeschlagen und die beiden vorgestellten Verfahren konnten schneller benutzt werden als sämtliche in Unterabschnitt 2.2.5 von Speicher et al. (2018) vorgestellten Verfahren. Die Geschwindigkeiten in Zeichen pro Minute liegen bei diesen Verfahren bei 25,43 mit dem diskreten Cursor, 40 mit dem kontinuierlichen Cursor, 60,79 mit dem Touch-Verfahren, 73,96 mit dem Laser-Verfahren, 48,86 mit dem Gaze-Verfahren und 46,80 mit dem Freihand-Verfahren und damit allesamt unter den erzielten Geschwindigkeiten.

Zusammenfassend sind die Ergebnisse sehr zufriedenstellend, jedoch können sie nicht mit der Geschwindigkeit des Schreibens auf realen Tastaturen während der Benutzung der VR mithalten. Es konnte ein Sättigungsverhalten der Entwicklung der Geschwindigkeiten festgestellt werden, von dem aus Nutzer zunehmend langsamer die Frequenz der Tastenanschläge und die Länge der Texte erhöhen können.

Wie stark diese Sättigung in der Hauptstudie eingesetzt hat, kann nicht genau bestimmt werden, jedoch kann die logarithmische Regression als Approximation der Sättigung angesehen werden. Werden die Funktionen $y = 98,86 \cdot \ln(x) + 531,68$ bei dem Drumstick-Verfahren und $y = 86,82 \cdot \ln(x) + 310,71$ bei dem Thumbpad-Verfahren auf lange Trainingszeiten hochgerechnet, so nähert sich vor allem das Drumstick-Verfahren den Geschwindigkeiten von Nieuwenhuizen (2017) an.

6 Diskussion

In diesem Kapitel wird zunächst in Abschnitt 6.1 ein Ausblick darüber gegeben, an welchen Punkten eine vertiefende Arbeit an diesem Thema anknüpfen könnte und durch welche Aspekte die Verfahren verbessert werden könnten. Im Anschluss wird in Abschnitt 6.2 zusammenfassend reflektiert, in welchem Umfang die Zielsetzungen der Arbeit erreicht werden konnten.

6.1 Ausblick

Obwohl in der Arbeit erfolgreich eine Reihe von Verfahren konzeptioniert, implementiert und evaluiert werden konnten, ließen sich diese dennoch in einigen Bereichen modifizieren, um die Eingabegeschwindigkeiten weiter zu verbessern.

So wurde das Assistenzsystem der Wortvorschläge nur begrenzt nutzbar gemacht, da die Durchsuchung der N-Gramme auf reduzierten Fassungen beruht. Daher liegt eine mögliche Optimierung des Assistenzsystems darin, durch eine verbesserte Parallelisierung und eine effizientere Datenrepräsentation einen größeren Ausschnitt der N-Gramme verwenden zu können.

Zusätzlich könnte künftig durch die Verwendung von höheren N-Grammen eine bessere Wortvorhersage erreicht werden. Zur Optimierung der Gewichtung zwischen Vorschlägen verschieden hoher N-Gramme kann ein Algorithmus entwickelt werden, der je nach tatsächlicher Übereinstimmung des Geschriebenen mit den zuvor gewählten Vorschlägen die Gewichtung automatisch anpasst.

Die Vorhersagen lassen sich außerdem durch eine Personalisierung der N-Gramme optimieren. Bisher wird zur Erhaltung vergleichbarer Ergebnisse in der Studie zur Laufzeit keine Änderung an den N-Grammen vorgenommen. In der Praxis sollte geschriebener Text den N-Grammen als neuer Eintrag hinzugefügt bzw. deren Häufigkeit angepasst werden.

An der Berechnung der Levenshtein-Distanz der Wortvorschläge könnte außerdem eine Modifikation vorgenommen werden, die Buchstabenersetzungen nicht mit der konstanten Distanz von 1 wertet, sondern eine dynamische Distanz zuordnet, die von dem Abstand der Tasten auf der virtuellen Tastatur abhängig ist und so die Wahrscheinlichkeit eines korrekten Wortvorschlages erhöht. Zusätzlich ist während der Studie aufgefallen, dass vermehrt unabsichtliche Vertauschungen von zwei Buchstaben innerhalb eines Wortes auftreten, daher wäre die Damerau-Levenshtein Distanz möglicherweise noch hilfreicher zur Bestimmung der Wortvorschläge. Diese unterscheidet sich von der klassischen Levenshtein-Distanz darin, dass die Vertauschung zweier in der Schreibreihenfolge benachbarter Buchstaben als Elementar-Operation eingeführt wird und somit Fehler dieser Art eine Distanz von 1 statt 2 verursachen. Diese Distanz wird z.B. von Jimenez (2007) dazu verwendet, leicht fehlerbehaftete Passworteingaben zu tolerieren.

Außerdem könnten die Wortvorschläge zusätzlich davon beeinflusst werden, welche Objekte sich in der virtuellen Szene befinden, sodass der Nutzer im Kontext des *Immersive-Information-Spaces*-Projekt Literatur und sonstige Objekte, die sich bereits im virtuellen Raum befinden, schnell aufrufen kann.

Ein weiterer Aspekt ist die Wahl des Sprachverarbeitungswerkzeuges.

Während bei dieser Arbeit frühzeitig entschieden wurde, dass die Sprachunterstützung nicht Bestandteil der Studien sein wird, bleibt sie dennoch für den Einsatz in der Praxis eine Möglichkeit, Texte einzugeben und Befehle auszuführen. Aus diesem Grund sollten für die Verwendung in der Praxis weitere Sprachverarbeitungswerkzeuge evaluiert werden.

Auch an den Eingabeverfahren selbst ließen sich einige Optimierungen vornehmen, um diese intuitiver bedienbar zu machen, oder um die Fehleranfälligkeit der Verfahren zu reduzieren. So könnte z.B. bei den Thumbpad-Gesten das Mapping der Daumenbewegung zur dargestellten Linie in der Art angepasst werden, dass der Nutzer innerhalb einer Buchstabengruppe mit der Linie genau auf das virtuelle Zeichen zusteuern kann und nicht von der angezeigten Position des Buchstabens auf eine Geste abstrahieren muss. Das Gaze-Verfahren könnte möglicherweise durch Eye-Tracking verbessert werden um den Nacken weniger zu belasten und das Blinzeln als Interaktionsoption einzuführen. Die Collider des Drumstick- und des Touch-Verfahrens könnten zudem etwas transparent gemacht werden, um Reduzierungen der Eingabegeschwindigkeit durch Verdeckung zu mindern. Außerdem könnten einige der Verbesserungsvorschläge aus Tabelle A.24 genutzt werden, um den Nutzern eine erleichterte Bedienung oder umfangreichere Funktionalität zu bieten.

Bei einigen der Verfahren hätte die Funktionserweiterung einer Swipe-Mechanik das Potential, die Schreibgeschwindigkeit weiter zu erhöhen. Bei dieser erfolgt die Zeichenselektion allein über die Position und Verweildauer des auswählenden Objektes, ohne die Bestätigung der Selektion durch das Drücken einer Taste. Es ist anzunehmen, dass besonders die Thumbpad-Tastatur und die Thumbpad-Kreise davon profitieren könnten. Die Mechanik wurde in der VR bereits von Bard (2007) evaluiert.

6.2 Zusammenfassung

Bezüglich der Zielsetzung wurde die Bestimmung geeigneter Eingabeverfahren in Kapitel 3 vollzogen. Es wurden insgesamt neun Konzepte vorgestellt. Die Verfahren wurden so, wie in Kapitel 4 beschrieben, umgesetzt und zur Bedienung mit einer HTC Vive entwickelt. Die Verfahren wurden daraufhin, wie vorgesehen, um die Assistenzsysteme erweitert. Für die Entwicklung des Durchführungsprozesses der Studie ist deutlich geworden, dass das Handschrift-Verfahren, sowie die Sprachverarbeitung als Assistenzsystem nicht Bestandteil der Studie sein sollten.

Zur Ermittlung von erreichbaren Schreibgeschwindigkeiten bei der langzeitigen Benutzung wurde entschieden, zwei Studien durchzuführen, von der die erste nicht nur die Verfahren auf die Probe gestellt und den Erhalt von frühzeitigem Feedback geschaffen hat, sondern auch der Auswahl der Verfahren gedient hat, die in der Hauptstudie trainiert wurden. Mit der Durchführung der Pilotstudie konnte daraufhin die Auswahl der Verfahren getroffen werden und die Thumbpad-Tastatur sowie das Drumstick-Verfahren ausgewählt werden. Zwischen der Pilotstudie und der Hauptstudie konnten die ausgewählten Verfahren anhand des Fragebogens und Auffälligkeiten während der Durchführung der Pilotstudie angepasst werden und so in optimierter Form in der Hauptstudie genutzt werden. Für die Hauptstudie wurde der Durchführungsprozess angepasst, sodass die Teilnehmer die Zeit der Trainingseinheiten effizienter für die Bedienung der Eingabeverfahren nutzen konnten. Schließlich folgte die Evaluation der aufgezeichneten Daten und Fragebögen, durch die sich das Drumstick-Verfahren mit einer Schreibfrequenz von bis zu 166,4 Tastenanschlägen pro Minute und 163,8 Zeichen pro Minute bei lediglich 2,5 Stunden Trainingszeit als bestes Verfahren etablieren konnte. Im Vergleich zu allen anderen Verfahren der Literatur schneidet das Drumstick-Verfahren besser ab, jedoch mit der Ausnahme des Schreibens auf einer realen Tastatur während der Nutzung der VR.

Literaturverzeichnis

- Alhalabi, W. (2016). Virtual reality systems enhance students' achievements in engineering education. *Behaviour & Information Technology*, 35(11):919–925. [1]
- Bard, G. V. (2007). A Prototype for Text Input in Virtual Reality with a Swype-like Process Using a Hand-tracking Device. <https://escholarship.org/uc/item/43r3t7m8>. (Accessed on 10/06/2018). [70]
- Bhattacharya, S., Basu, A., und Samanta, D. (2008). Computational modeling of user errors for the design of virtual scanning keyboards. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 16(4):400–409. [63]
- Bowman, D. A., Kruijff, E., LaViola, J. J., und Poupyrev, I. (2001). An Introduction to 3-D User Interface Design. *Presence: Teleoperators and Virtual Environments*, 10(1):96–108. [11]
- Bowman, D. A., Rhoton, C. J., und Pinho, M. S. (2002). Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 46(26):2154–2158. [12]
- Elmgren, R. (2017). Handwriting in VR as a Text Input Method. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1107665&dswid=-7451>. (Accessed on 10/05/2018). [12, 33]
- Gartner, L. (2018). Häufigkeitstabellen. <http://www.mathe.tu-freiberg.de/~hebis/caf/kryptographie/haeufigkeitstabellen.html>. (Accessed on 10/05/2018). [21]
- Gomide, R. d. S., Loja, L. F. B., Lemos, R. P., Flôres, E. L., Melo, F. R., und Teixeira, R. A. G. (2016). A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques. *Research on Biomedical Engineering*, 32(2):176–198. [10]
- Google LLC (2018a). Big-Data-Lösungen | Google Cloud. <https://cloud.google.com/solutions/big-data/>. (Accessed on 10/04/2018). [45]
- Google LLC (2018b). Google Ngram Viewer. <https://storage.googleapis.com/books/ngrams/books/datasetv2.html>. (Accessed on 09/27/2018). [37]
- Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., und Kristensson, P. O. (2018). Text Entry in Immersive Head-Mounted Display-based Virtual Reality using Standard Keyboards. *CoRR*, abs/1802.00626. [10, 11, 62]
- Hart, S. G. (2006). Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908. [61, 65]

- Hoste, L. und Signer, B. (2014). SpeeG2. In Epps, J., Chen, F., Oviatt, S., Mase, K., Sears, A., Jokinen, K., und Schuller, B., Herausgeber, *Proceedings of the 2014 ACM International Conference on Multimodal Interaction*, Seiten 213–220, New York, NY. ACM. [10]
- Huckauf, A. und Urbina, M. (2008). On object selection in gaze controlled environments | Huckauf | Journal of Eye Movement Research. <https://bop.unibe.ch/index.php/JEMR/article/view/2276>. (Accessed on 10/05/2018). [8]
- Jacob, R. J. K. (1990). What you look at is what you get: eye movement-based interaction techniques. <https://dl.acm.org/citation.cfm?id=97246>. (Accessed on 10/05/2018). [9]
- Jimenez, J. G. (2007). Spelling-Error Tolerant, Order-Independent Pass-Phrases via the Damerau-Levenshtein String-Edit Distance Metric. (Accessed on 10/06/2018). [69]
- Kim, G. (2005). *Designing Virtual Reality Systems: The Structured Approach*. Springer London, London. [11]
- Krijn, M., Emmelkamp, P. M., Biemond, R., de Wilde Ligny, C., Schuemie, M. J., und van der Mast, C. A. (2004). Treatment of acrophobia in virtual reality: The role of immersion and presence. *Behaviour Research and Therapy*, 42(2):229–239. [1]
- MacKenzie, I. und Chang, L. (1999). A performance comparison of two handwriting recognizers. *Interacting with Computers*, 11(3):283–297. [11]
- MacKenzie, I. S. (2002). KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. In Paternò, F., Herausgeber, *Human Computer Interaction with Mobile Devices*, Band 2411 in *Lecture Notes in Computer Science*, Seiten 195–210. Springer, Berlin and Heidelberg. [62]
- MacKenzie, I. S., Zhang, S. X., und Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & information technology*, 18(4):235–244. [13, 18, 77]
- MacKenzie, S. und Soukoreff, W. (2002). *Text entry for mobile computing: Special issue*, Band vol.17, no's 2 & 3 in *Human-computer interaction*. Lawrence Erlbaum, Mahwah, NJ. and London. [19]
- Majaranta, P., Ahola, U.-K., und Špakov, O. (2009). Fast gaze typing with an adjustable dwell time. In Olsen, D. R., Arthur, R. B., Hinckley, K., Morris, M. R., Hudson, S., und Greenberg, S., Herausgeber, *CHI 2009 - digital life, new world*, Seite 357, New York, NY. ACM. [8, 9, 33, 67]
- Mayzner, M. (2017). English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU. <http://norvig.com/mayzner.html>. (Accessed on 10/06/2018). [67]
- Mott, M. E., Williams, S., Wobbrock, J. O., und Morris, M. R. (2017). Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In Mark, G., Fussell, S., Lampe, C., schraefel, m., Hourcade, J. P., Appert, C., und Wigdor, D., Herausgeber, *Explore, innovate, inspire*, Seiten 2558–2570, New York, NY. Association for Computing Machinery Inc. (ACM). [8, 32, 62]
- NASA (2018). TLX @ NASA Ames - Home. <https://humansystems.arc.nasa.gov/groups/TLX/>. (Accessed on 09/28/2018). [61, 66, 115]

- Nieuwenhuizen, F. (2017). Evaluation of typing performance in virtual reality on a physical keyboard. <https://elib.uni-stuttgart.de/handle/11682/9895>. (Accessed on 10/04/2018). [1, 10, 11, 62, 67]
- Ogitani, T., Arahori, Y., Shinyama, Y., und Gondow, K. (2018). Space Saving Text Input Method for Head Mounted Display with Virtual 12-key Keyboard. In Barolli, L., Herausgeber, *32nd IEEE International Conference on Advanced Information Networking and Applications*, Seiten 342–349, Piscataway, NJ. IEEE. [13]
- Okreylos (2016). Lighthouse tracking examined | Doc-Ok.org. <http://doc-ok.org/?p=1478>. (Accessed on 10/05/2018). [5]
- Olofsson, J. (2017). Input and Display of Text for Virtual Reality Head-Mounted Displays and Hand-held Positionally Tracked Controllers. <http://ltu.diva-portal.org/smash/record.jsf?pid=diva2%3A1117389&dswid=-3846>. (Accessed on 10/05/2018). [13]
- Oxford University Press (2014). *The Oxford handbook of virtuality*. Oxford handbooks. Oxford and New York and Auckland and Cape Town and Dar es Salaam and Hong Kong and Karachi and Kuala Lumpur. [1]
- Pantelidis, V. S. (2009). Reasons to Use Virtual Reality in Education and Training Courses and a Model to Determine When to Use Virtual Reality | Pantelidis | Themes in Science and Technology Education. Seiten 59–70. (Accessed on 10/06/2018). [1]
- Pfeiffer, T. (2013). *Gaze-based assistive technologies*, Seiten 90–109. IGI Global. [9]
- Pfeiffer, T., Summann, F., Hellriegel, J., Wolf, S., und Pietsch, C. (2017). Virtuelle Realität zur Bereitstellung integrierter Suchumgebungen: 94-107 Seiten / o-bib. Das offene Bibliotheksjournal / herausgegeben vom VDB, Bd. 4, Nr. 4 (2017). [2, 7, 35]
- Poupyrev, I., Tomokazu, N., und Weghorst, S. (1998). Virtual Notepad: handwriting in immersive VR. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*, Seiten 126–132. IEEE Comput. Soc. [11, 12]
- Simathamand, J. und Piromsopa, K. (2011). Performance optimization of Thai virtual keyboard for social networking. In Mitranont, J. L., Herausgeber, *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE), 2011*, Seiten 210–213, Piscataway, NJ. IEEE. [10]
- Simone, L. K., Schultheis, M. T., Rebimbas, J., und Millis, S. R. (2006). Head-mounted displays for clinical virtual reality applications: pitfalls in understanding user behavior while using technology. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, 9(5):591–602. [1]
- Speicher, M., Feit, A. M., Ziegler, P., und Krüger, A. (2018). Selection-based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, Seiten 647:1–647:13, New York, NY, USA. ACM. [13, 14, 33, 67]
- Sulzberger, C. (2007). Effiziente Levenshtein Implementierungsansätze. <http://www.levenshtein.de/>. (Accessed on 10/05/2018). [7, 8]
- Varcholik, P. D., LaViola, J. J., und Hughes, C. E. (2012). Establishing a baseline for text entry for a multi-touch virtual keyboard. *International Journal of Human-Computer Studies*, 70(10):657–672. [10]

- Vive-Setup (2018). HTC Vive Installation Guide. https://support.steampowered.com/steamvr/HTC_Vive/. (Accessed on 10/06/2018). [53]
- Walsh, K. (2017). Real Uses of Virtual Reality in Education: How Schools are Using VR | Emerging Education Technologies. <https://www.emergingedtech.com/2017/06/real-uses-of-virtual-reality-in-education-how-schools-are-using-vr/>. (Accessed on 10/04/2018). [1]
- Ward, D. J. und MacKay, D. J. (2002). Fast Hands-free Writing by Gaze Direction. <http://www.inference.org.uk/mackay/eyeshortpaper.pdf>. (Accessed on 09/29/2018). [9, 31]
- Wobbrock, J. O. (2007). Measures of Text Entry Performance. In MacKenzie, I. S. und Tanaka-Ishii, K., Herausgeber, *Text entry systems*, The Morgan Kaufmann series in interactive technologies, Seiten 47–74. Boston, Amsterdam and s.l. [63]
- Yamada, H. (1980). A historical study of typewriters and typing methods, from the position of planning japanese parallels. [67]

Appendix

Logarithmische Regression

Zur Bestimmung einer Lernkurve werden in Unterabschnitt 5.4.1 logarithmische Regressionen durchgeführt, um eine Funktion zu entwickeln, die die Schreibgeschwindigkeiten abschätzt, welche die Nutzer der Eingabeverfahren durch weiteres Training dieser Verfahren erreichen können.

Regressionen sind ein statistisches Analyseverfahren, das Bezüge zwischen Regressoren, das sind unabhängige Variablen, oder auch erklärende Variablen genannt, und Regressanden, das sind abhängige Variablen, oder auch erklärte Variablen genannt, herstellt.

Im Kontext der Regression der Ergebnisse bildet die Nummer der Trainingseinheit die erklärende Größe, die einen Einfluss auf die Zahl der erwarteten Tastenanschläge, also die erklärte Größe, hat. Die Regression sollte in dieser Anwendung logarithmisch sein, da nach MacKenzie et al. (1999) ein Sättigungsverhalten der Geschwindigkeiten angenommen werden kann.

Bei den Regressionen werden diejenigen Funktionen bestimmt, die die gegebenen Datenpunkte bestmöglich abbilden. Diese Funktionen haben die Form $y = a \cdot \ln(x) + b$. Dabei ist y die im späteren Kontext resultierende Geschwindigkeit der Tastenanschläge, x die Anzahl der Verwendungen des jeweiligen Eingabeverfahrens und a und b die zu bestimmenden Parameter der Funktion.

Die Güte einer Regression wird durch das Bestimmtheitsmaß R^2 festgelegt und die Abweichungen der Datenpunkte von der Regressionskurve durch die Residuenquadratsumme RSS bemessen.

Das Bestimmtheitsmaß ist definiert als

$$R^2 = \frac{SQE}{SQT} = \frac{\sum_{n=1}^m (\hat{y}_n - \bar{y})^2}{\sum_{n=1}^m (y_n - \bar{y})^2} \quad (\text{A.1})$$

Dabei ist SQE die sogenannte *Erklärte Abweichungsquadratsumme* und SQT die *Gesamte Abweichungsquadratsumme*.

$\hat{y}(x_n)$ ist eine Schätzfunktion, die jedem Wert x_n einen geschätzten Funktionswert zuweist, und deren Abweichungen von den tatsächlichen Werten Residuen genannt werden. \bar{y} ist der empirische Mittelwert und wird wie folgt berechnet:

$$\bar{y} = \frac{1}{n} \cdot \sum_{n=1}^m y_n \quad (\text{A.2})$$

Die Residuenquadratsumme wird berechnet durch

$$RSS = \sum_{n=1}^m (y_n - \hat{y}_n)^2 = \sum_{n=1}^m e_n^2 \quad (\text{A.3})$$

Dabei sind die e_n^2 die Residuen.

Diagramme

A.1 Pilotstudie

Abbildung A.1: Ergebnisse der Cluster-Methode der Pilotstudie

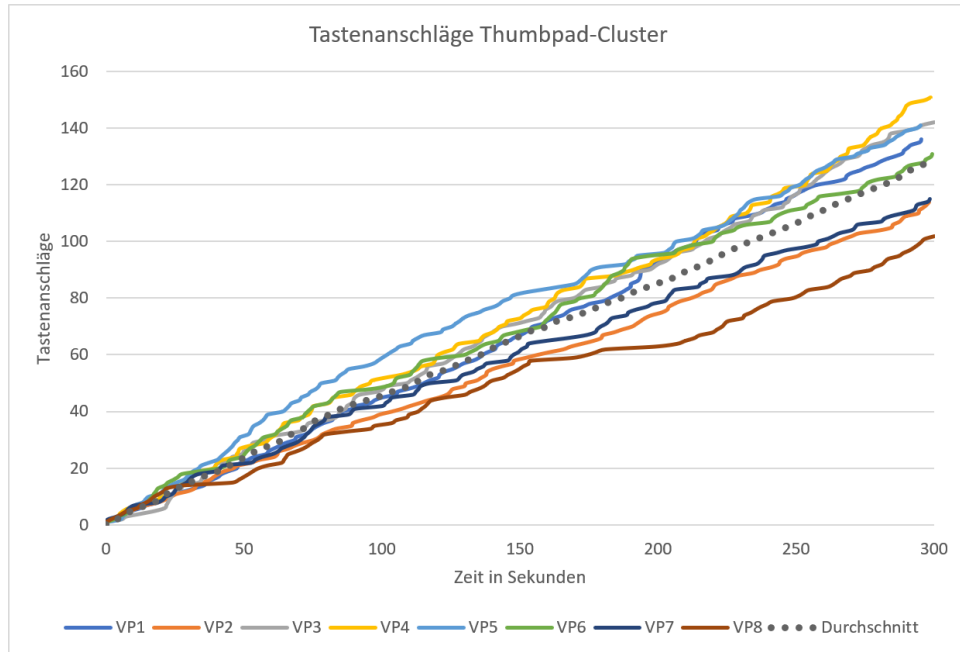


Abbildung A.2: Ergebnisse der Gaze-Methode der Pilotstudie

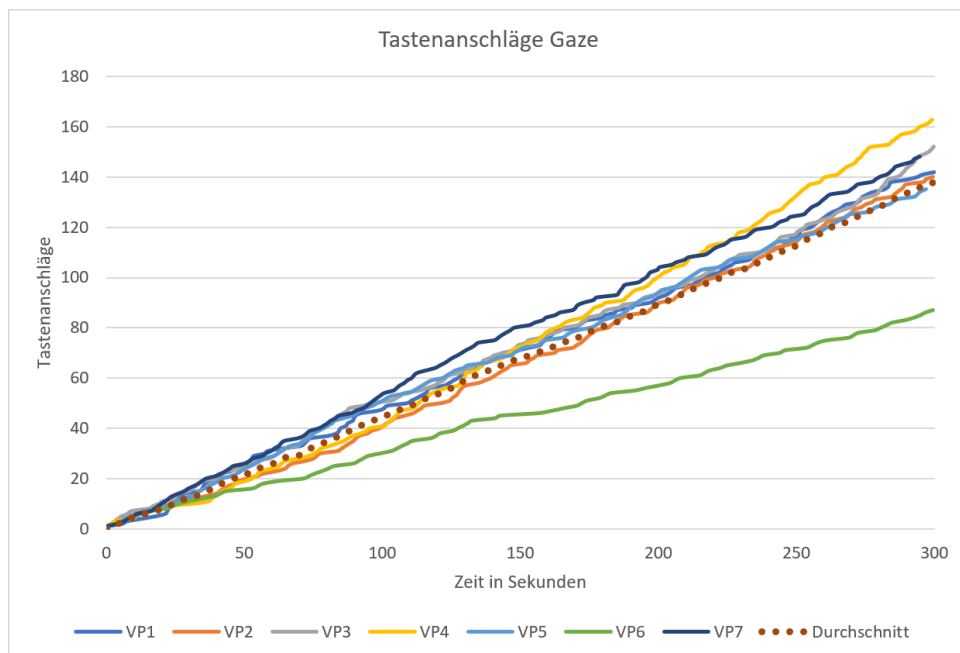


Abbildung A.3: Ergebnisse der Gesten-Methode der Pilotstudie

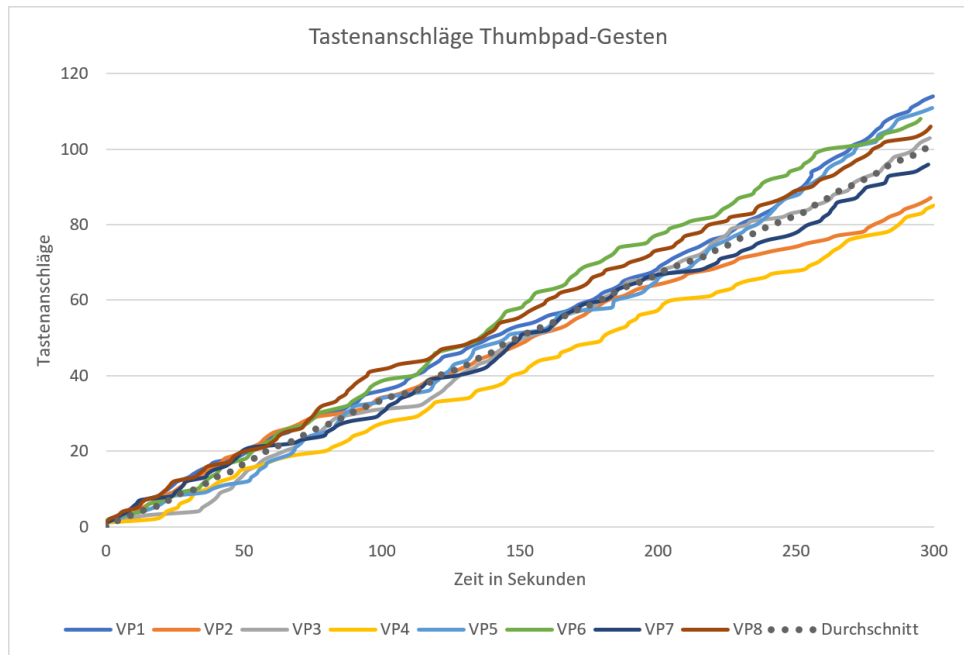


Abbildung A.4: Ergebnisse der Laser-Methode der Pilotstudie

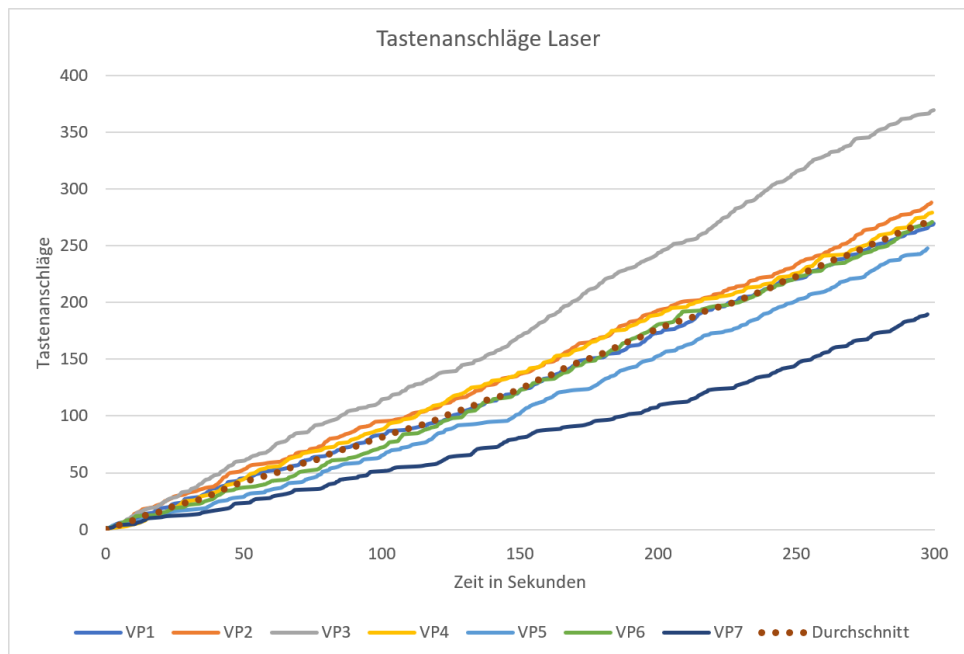


Abbildung A.5: Ergebnisse der Drumstick-Methode der Pilotstudie

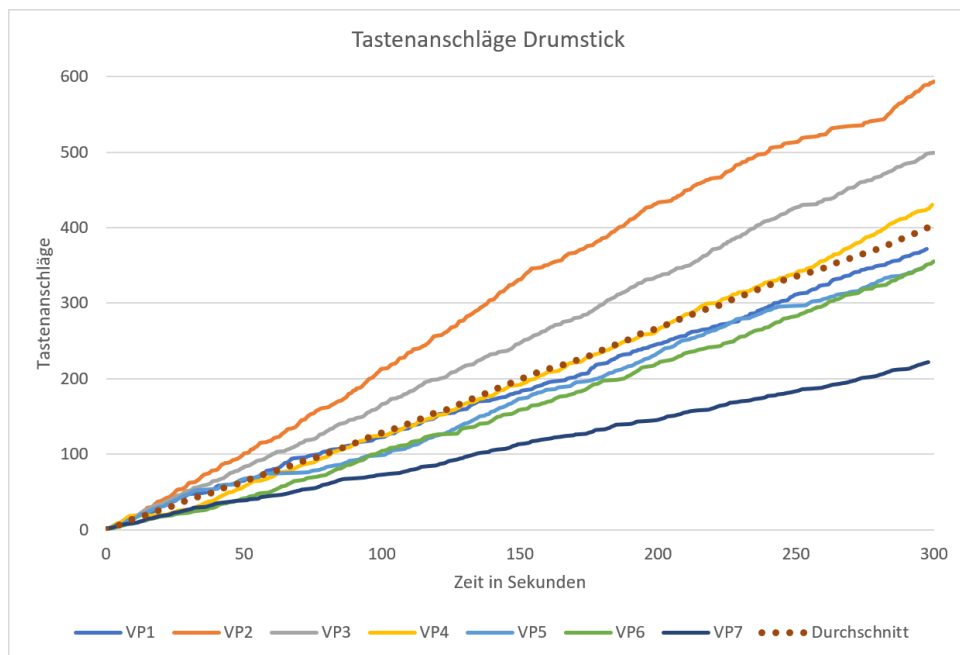


Abbildung A.6: Ergebnisse der Thumbpad-Tastatur-Methode der Pilotstudie

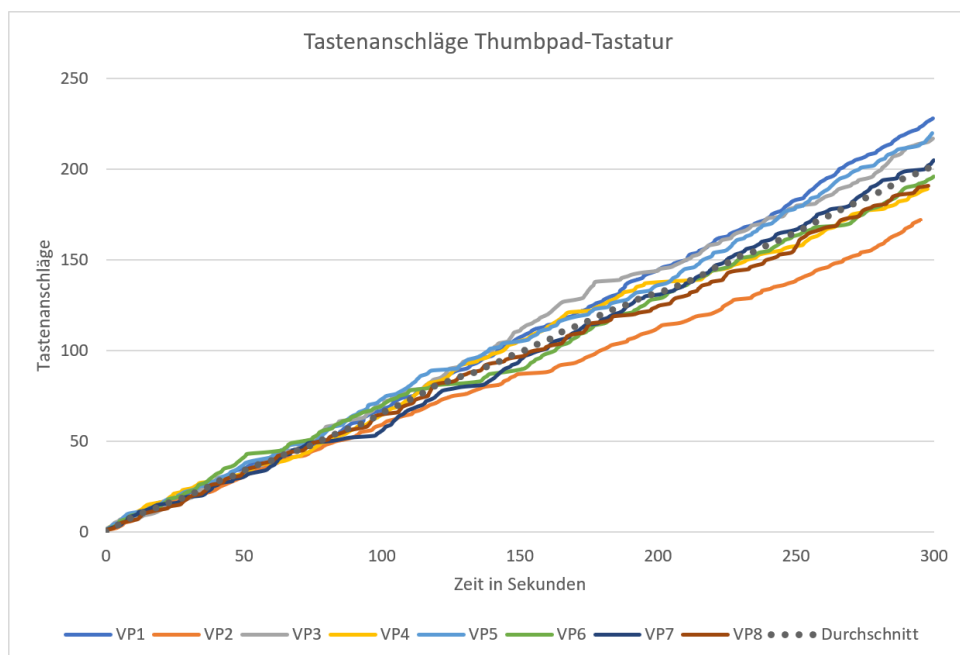


Abbildung A.7: Ergebnisse der Touch-Methode der Pilotstudie

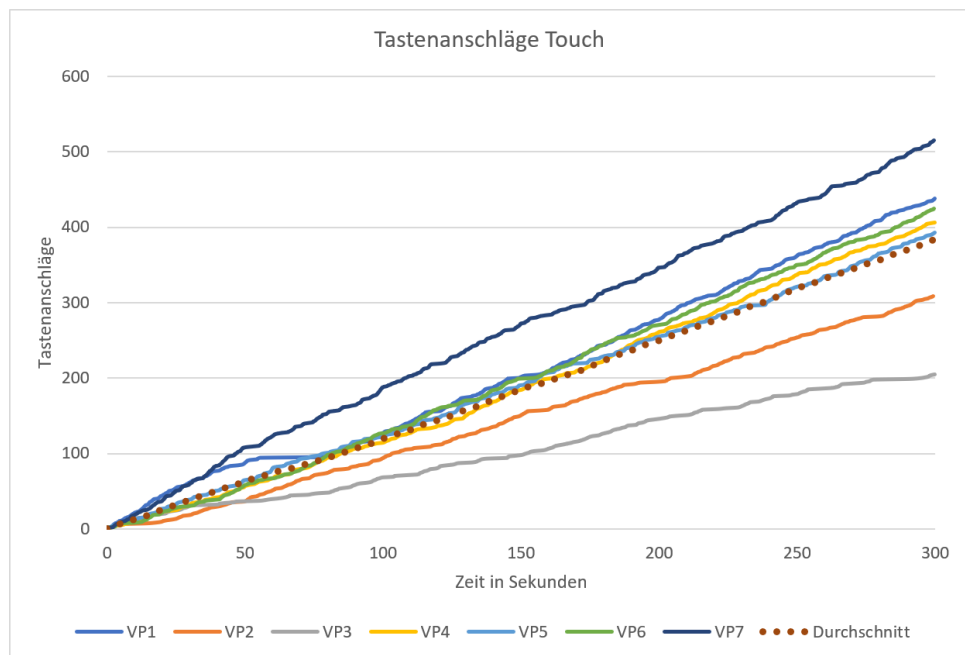


Abbildung A.8: Ergebnisse der Kreise-Methode der Pilotstudie

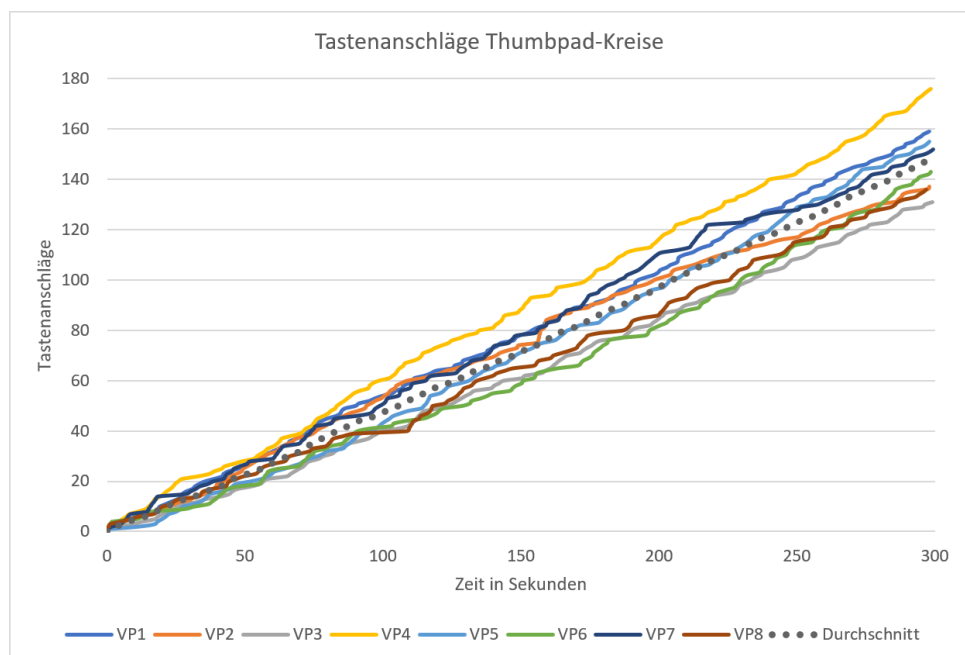
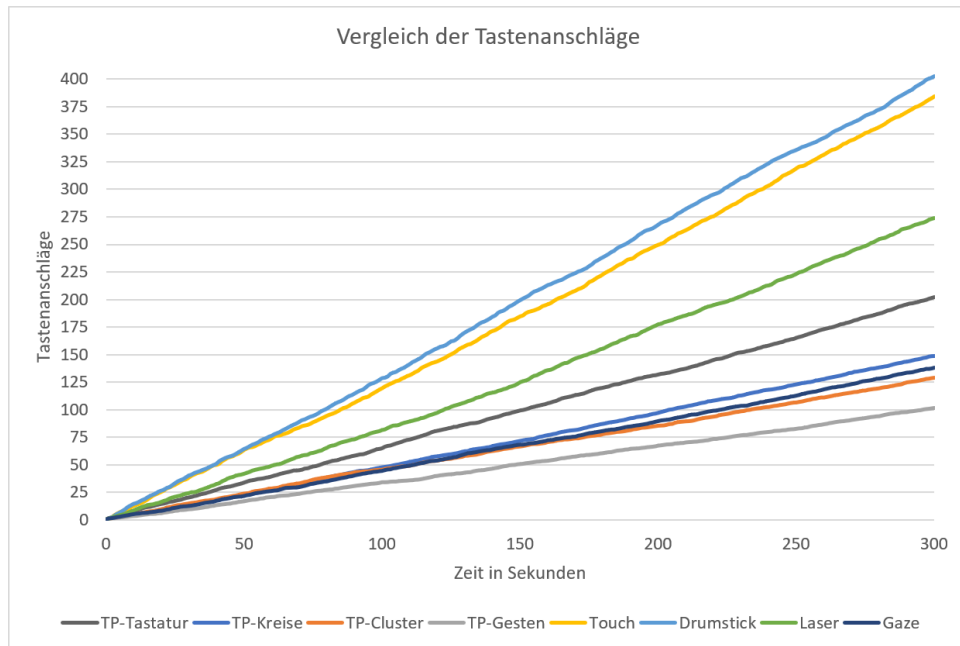


Abbildung A.9: Vergleich der Verfahren hinsichtlich der Tastenanschläge



TP = Thumbpad

A.2 Hauptstudie

Abbildung A.10: Tastenanschläge mit Drumstick-Verfahren

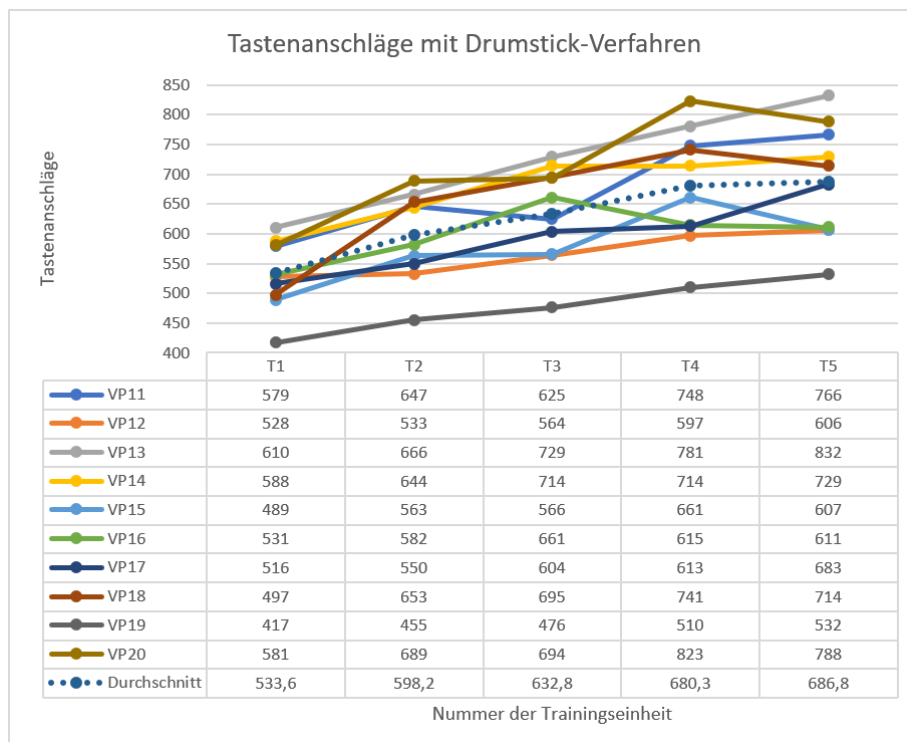


Abbildung A.11: Tastenanschläge mit Thumbpad-Tastatur-Verfahren

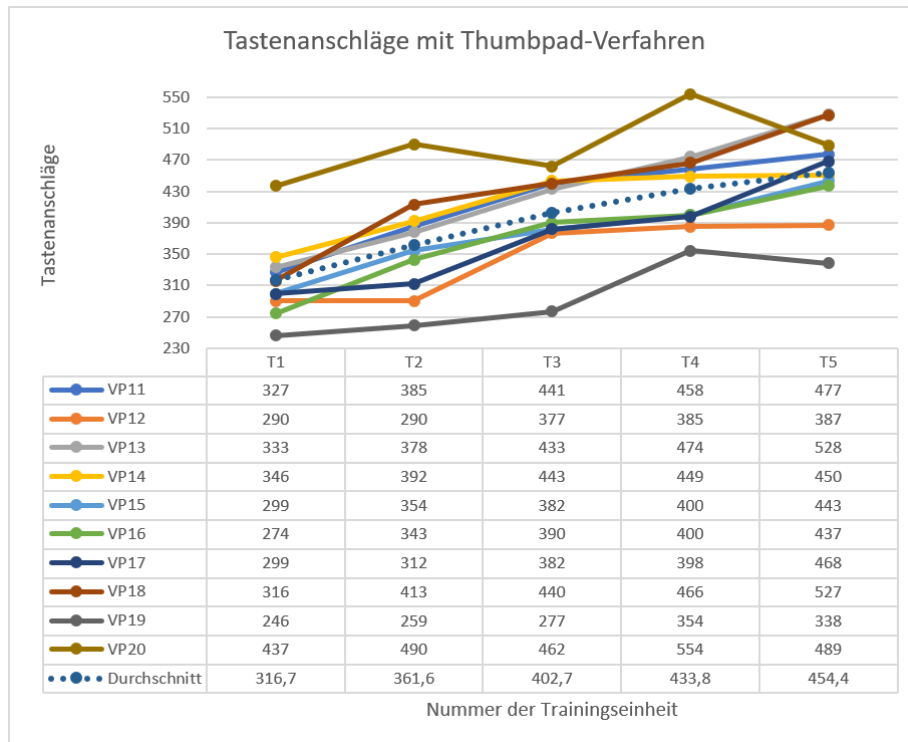


Abbildung A.12: Textlänge mit Drumstick-Verfahren

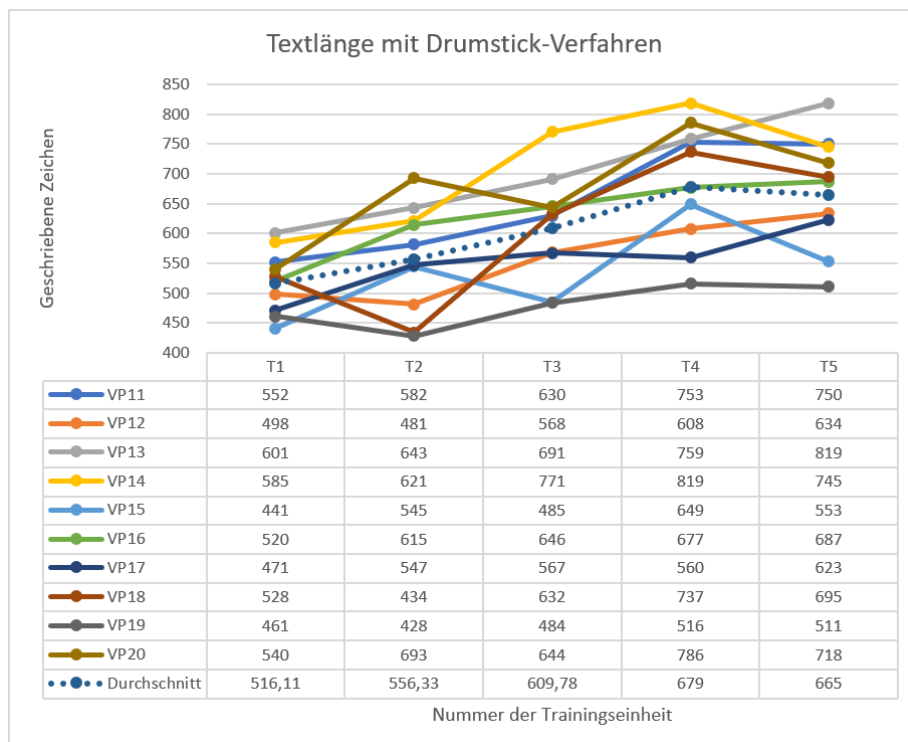


Abbildung A.13: Textlänge mit Thumbpad-Tastatur-Verfahren

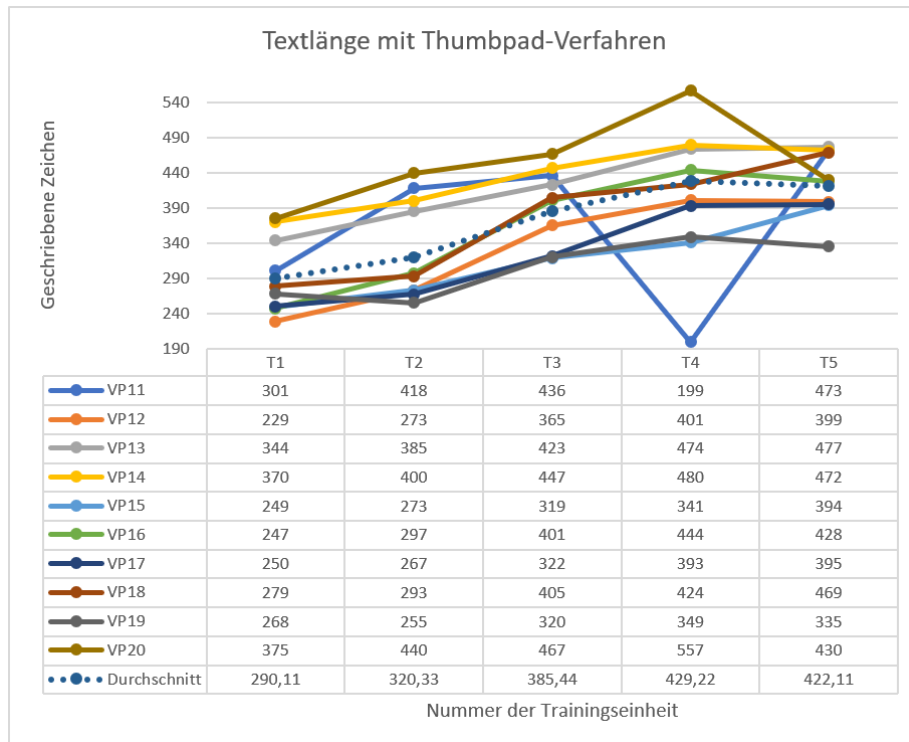


Abbildung A.14: Fehlerquote mit Drumstick-Verfahren

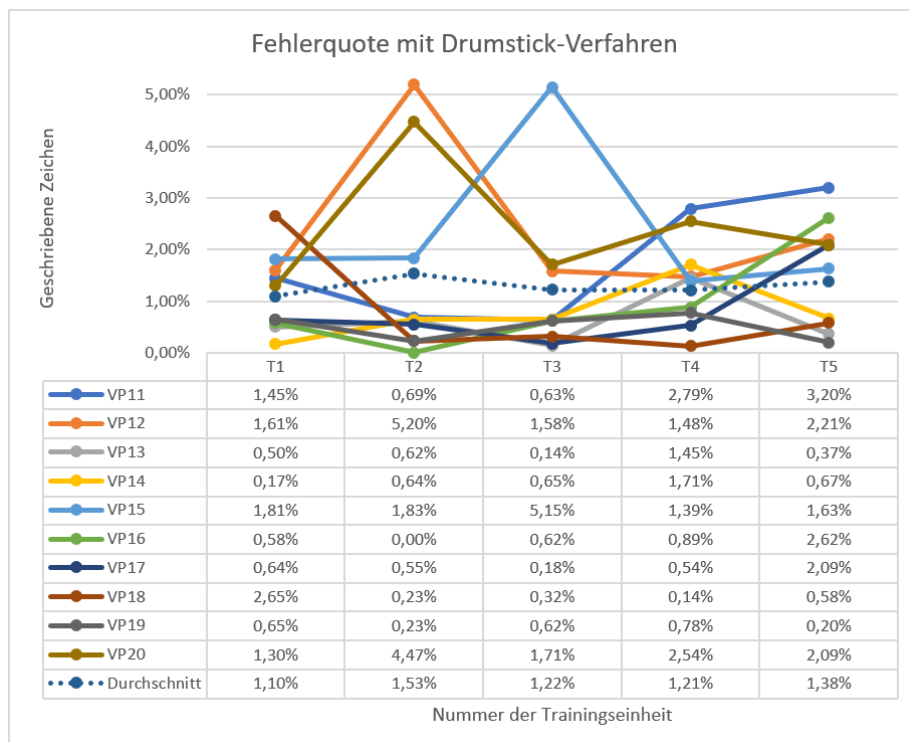


Abbildung A.15: Fehlerquote mit Thumbpad-Tastatur-Verfahren

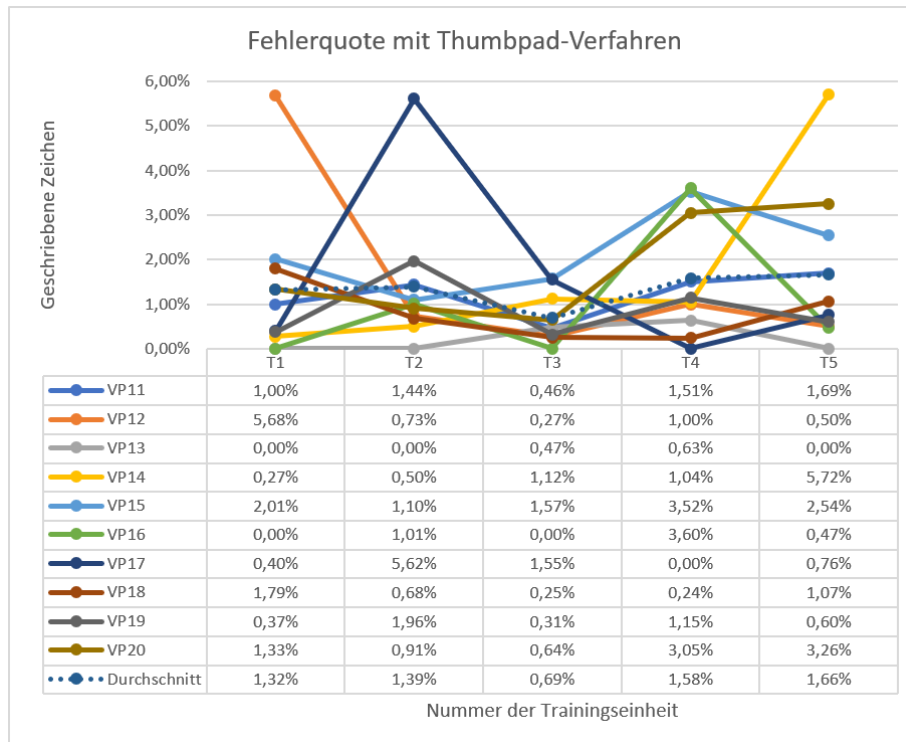


Abbildung A.16: Funktionskurve der Tastenanschläge des Drumstick-Verfahrens

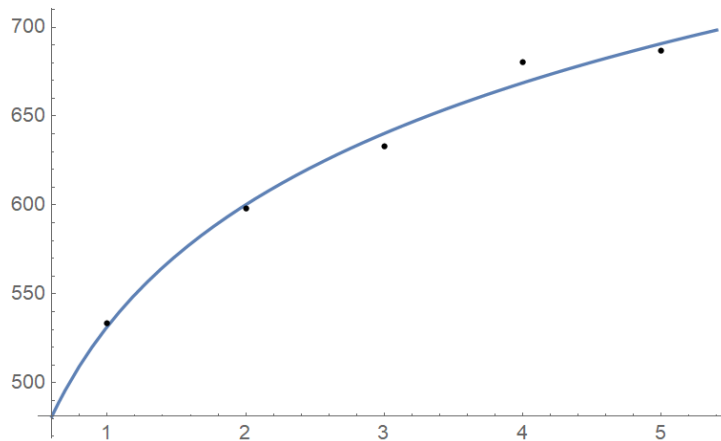


Abbildung A.17: Residuen der Funktionskurve des Drumstick-Verfahrens

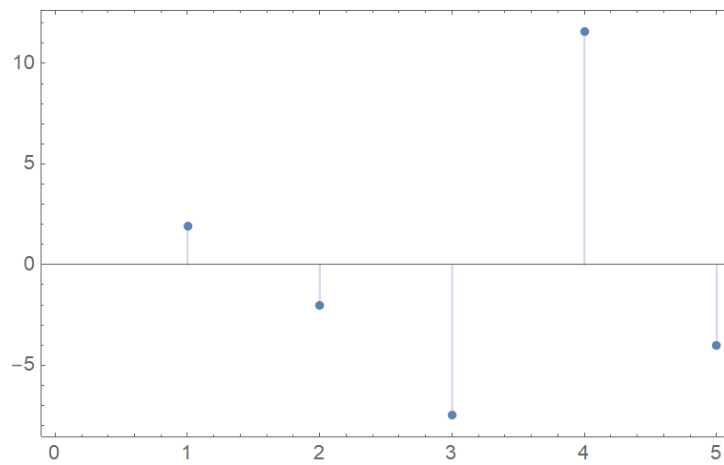


Abbildung A.18: Funktionskurve der Tastenanschläge des Thumbpad-Verfahrens

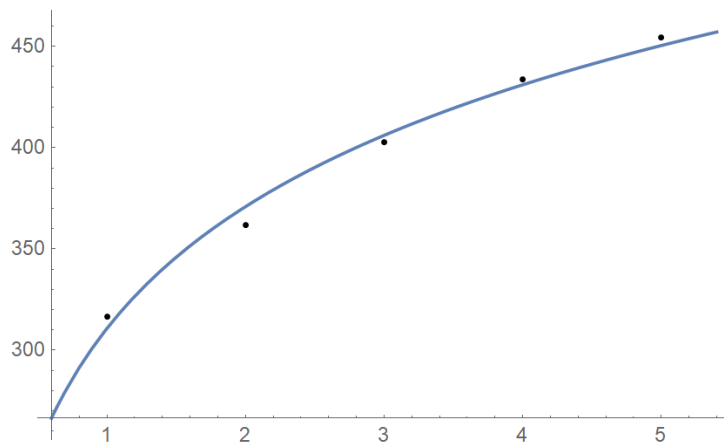
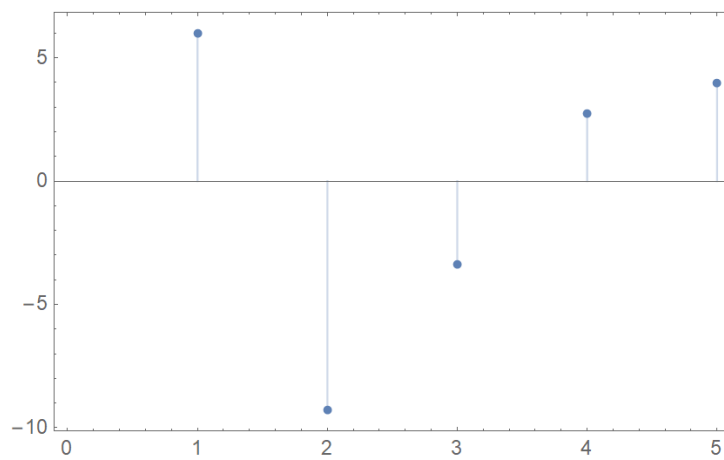


Abbildung A.19: Residuen der Funktionskurve des Thumbpad-Verfahrens



Fragebögen

A.3 Original

A.3.1 Pilotstudie

Abbildung A.20: Fragebogen zur Pilotstudie Seite 1

Fragebogen zur Studie

Bitte beantworten Sie die nachfolgenden Fragen wahrheitsgemäß. Sollten Sie eine Frage nicht beantworten wollen, kennzeichnen Sie dies bitte entsprechend. Ihren Angaben wird ein Pseudonym zugeordnet. Weder ein Rückschluss auf die Antworten einzelner Personen, noch die Zuordnung eines Pseudonyms zu einer Person sind möglich.

Name		Alter	
Geschlecht		Datum	

1. Sind Sie Links- oder Rechtshänder?

2. Wie gewohnt sind sie daran Texte auf Tastaturen zu schreiben?

- Stufe 1: Seltenes Schreiben und/oder sehr wenig Text
- Stufe 2: Unregelmäßiges Schreiben und/oder wenig Text
- Stufe 3: Regelmäßiges Schreiben und/oder mittellange Texte
- Stufe 4: Häufiges Schreiben und/oder lange Texte
- Stufe 5: Sehr häufiges Schreiben und/oder sehr lange Texte

3. Wie viele Finger benutzen Sie beim Schreiben auf Tastaturen?

4. Wie gewohnt sind sie daran Texte auf Mobilgeräten zu schreiben?

- Stufe 1: Seltenes Schreiben und/oder sehr wenig Text
- Stufe 2: Unregelmäßiges Schreiben und/oder wenig Text
- Stufe 3: Regelmäßiges Schreiben und/oder mittellange Texte
- Stufe 4: Häufiges Schreiben und/oder lange Texte
- Stufe 5: Sehr häufiges Schreiben und/oder sehr lange Texte

5. Wie gewohnt waren Sie vor der Studie an die Benutzung der Virtuellen Realität?

- Stufe 1: Erste Benutzung in Studie
- Stufe 2: Wenige Benutzungen
- Stufe 3: Unregelmäßige wiederholte Benutzung
- Stufe 4: Regelmäßige kurze Benutzung
- Stufe 5: Häufige langzeitige Benutzung

Abbildung A.21: Fragebogen zur Pilotstudie Seite 2

6. Wie gewohnt waren Sie vor der Studie in der Virtuellen Realität Texte zu schreiben?

- Stufe 1: Erstes Schreiben in Studie
- Stufe 2: Seltenes Schreiben und/oder sehr wenige Worte
- Stufe 3: Unregelmäßiges Schreiben und/oder wenige Worte
- Stufe 4: Regelmäßiges Schreiben und/oder mittellange Texte
- Stufe 5: Häufiges Schreiben und/oder lange Texte

7. Wie schnell konnten Sie die Bedienung der Schreibmethoden auf einer Skala von 1 (sehr langsam) bis 10 (sehr schnell) lernen.

Drumstick-Verfahren:

Touch-Verfahren:

Laser-Verfahren:

Gaze-Verfahren:

8. Welche Vor- & Nachteile sehen Sie an den Verfahren?

Drumstick-Verfahren:

Touch-Verfahren:

Laser-Verfahren:

Gaze-Verfahren:

9. Was könnte man an den Verfahren verbessern?

Drumstick-Verfahren:

Touch-Verfahren:

Laser-Verfahren:

Gaze-Verfahren:

Abbildung A.22: Fragebogen zur Pilotstudie Seite 3

10. Wie hoch würden Sie Ihre Fehlerquote beim Schreiben der Diktate mit den Verfahren schätzen? (0-100%)

Drumstick-Verfahren:

Touch-Verfahren:

Laser-Verfahren:

Gaze-Verfahren:

11. Bitte bewerten Sie Ihre Schreibgeschwindigkeit mit den Verfahren, sowie die Schreibgeschwindigkeit auf einer Tastatur mit einem Wert von 0 (schlecht) bis 10 (gut).

Drumstick-Verfahren:

Touch-Verfahren:

Laser-Verfahren:

Gaze-Verfahren:

Tastatur:

12. Welches Verfahren hat Ihnen am besten gefallen? Wieso?

13. Welches Verfahren hat Ihnen am wenigsten gefallen? Wieso?

14. Auf einer Skala von 0 (gar nicht) bis 10 (sehr), wie sehr hat Sie die Technik bei der Aufgabe beeinträchtigt? (Unschärfe Texte, Unangenehme Bedienung etc.)

15. Welche technischen Aspekte haben Sie bei der Aufgabe beeinträchtigt? (Unschärfe Texte, Unangenehme Bedienung etc.)

16. Wie hoch war die geistige Anforderung der Aufgabe für Sie (Suchen, Denken, Erinnern, etc.)?

Abbildung A.23: Fragebogen zur Pilotstudie Seite 4

17. Wie hoch war die körperliche Anstrengung der Aufgabe für Sie (Bewegen, Drücken, Halten, etc.)?

18. Wie erfolgreich haben Sie Ihrer Meinung nach die Aufgabe bewältigt? Sind Sie zufrieden mit der erreichten Schreibgeschwindigkeit Ihrer bevorzugten Methode?

19. Wie frustriert fühlten Sie sich während der Aufgabe (Unsicherheit, Stress, Verärgerung)?

20. Welches Verfahren (in optimierter Form) würden Sie in Zukunft in der Virtuellen Realität verwenden wollen um Texte zu schreiben?

21. Ist Ihnen in der Verwendung der Verfahren etwas besonders Negatives aufgefallen?

22. Haben Sie sonstige Anmerkungen, Kritik oder Ideen?

A.3.2 Hauptstudie

Abbildung A.24: Fragebogen zur Hauptstudie Seite 1

Fragebogen zur Hauptstudie

Bitte beantworten Sie die nachfolgenden Fragen wahrheitsgemäß. Sollten Sie eine Frage nicht beantworten wollen, kennzeichnen Sie dies bitte entsprechend. Ihren Angaben wird ein Pseudonym zugeordnet. Weder ein Rückschluss auf die Antworten einzelner Personen, noch die Zuordnung eines Pseudonyms zu einer Person sind möglich.

Name		Alter	
Geschlecht		Datum	

1. Haben Sie an der Pilotstudie teilgenommen?

2. Sind Sie Links- oder Rechtshänder?

3. Wie gewohnt sind Sie daran Texte auf Tastaturen zu schreiben?

- Stufe 1: Seltenes Schreiben und/oder sehr wenig Text
- Stufe 2: Unregelmäßiges Schreiben und/oder wenig Text
- Stufe 3: Regelmäßiges Schreiben und/oder mittellange Texte
- Stufe 4: Häufiges Schreiben und/oder lange Texte
- Stufe 5: Sehr häufiges Schreiben und/oder sehr lange Texte

4. Wie viele Finger benutzen Sie beim Schreiben auf Tastaturen?

5. Wie gewohnt sind sie daran Texte auf Mobilgeräten zu schreiben?

- Stufe 1: Seltenes Schreiben und/oder sehr wenig Text
- Stufe 2: Unregelmäßiges Schreiben und/oder wenig Text
- Stufe 3: Regelmäßiges Schreiben und/oder mittellange Texte
- Stufe 4: Häufiges Schreiben und/oder lange Texte
- Stufe 5: Sehr häufiges Schreiben und/oder sehr lange Texte

6. Wie gewohnt waren Sie vor der Studie an die allgemeine Benutzung der Virtuellen Realität?

- Stufe 1: Erste Benutzung in Studie
- Stufe 2: Wenige Benutzungen
- Stufe 3: Unregelmäßige wiederholte Benutzung

Seite 1 von 5

Abbildung A.25: Fragebogen zur Hauptstudie Seite 2

Stufe 4: Regelmäßige kurze Benutzung
 Stufe 5: Häufige langzeitige Benutzung

7. Wie gewohnt waren Sie vor der Studie in der Virtuellen Realität Texte zu schreiben?

Stufe 1: Erstes Schreiben in Studie
 Stufe 2: Seltenes Schreiben (2-5x) und/oder sehr wenige Worte
 Stufe 3: Unregelmäßiges Schreiben (6-10x) und/oder wenige Worte
 Stufe 4: Regelmäßiges Schreiben (11-20x) und/oder mittellange Texte
 Stufe 5: Häufiges Schreiben (20+) und/oder lange Texte

8. Welche Vor- & Nachteile sehen Sie hinsichtlich der langzeitigen Benutzung bei den getesteten Verfahren?

Drumstick-Verfahren:

Thumbpad-Verfahren:

9. Welche Verbesserungen bzw. Veränderungen würden Sie sich an den getesteten Verfahren erhoffen, um sie langfristig in der VR einzusetzen?

Drumstick-Verfahren:

Thumbpad-Verfahren:

10. Welches Verfahren gefällt Ihnen besser um kurze Texte zu verfassen? Wieso?

11. Welches Verfahren gefällt Ihnen besser um lange Texte zu verfassen? Wieso?

Seite 2 von 5

Abbildung A.26: Fragebogen zur Hauptstudie Seite 3

12. Wie hoch im Vergleich zum jeweils vorherigen Tag würden Sie die Verbesserung Ihrer Schreibgeschwindigkeit einschätzen (in %)?

Verfahren:	Dienstag	Mittwoch	Donnerstag	Freitag
Drumstick				
Thumbpad				

13. Wie hoch würden Sie Ihre Fehlerquote (Vertippen) beim Schreiben der Diktate an den einzelnen Tagen schätzen? (0-100%)

Verfahren:	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
Drumstick					
Thumbpad					

14. Wurden Sie durch technische Aspekte bei der Aufgabe beeinträchtigt? (Periphere Unschärfe, Unangenehme Bedienung etc.)

NASA-TLX Fragebogen

Beispiel:

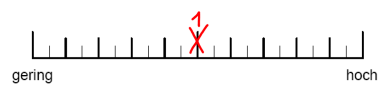


Abbildung A.27: Fragebogen zur Hauptstudie Seite 4

15. **Geistige Anforderung:** Wie viel geistige Anstrengung war bei der Informationsaufnahme und -verarbeitung erforderlich (z.B. Denken, Entscheiden, Rechnen, Erinnern, Hinsehen, Suchen...)? War die Aufgabe leicht oder anspruchsvoll, einfach oder komplex, erforderte sie hohe Genauigkeit oder war sie fehlertolerant?



16. **Körperliche Anforderung:** Wie viel körperliche Aktivität war erforderlich (z.B. Ziehen, Drücken, Drehen, Steuern, Aktivieren...)? War die Aufgabe leicht oder schwer, einfach oder anstrengend, erholsam oder mühselig?



17. **Zeitliche Anforderung:** Wie viel Zeitdruck empfanden Sie hinsichtlich der Häufigkeit oder dem Takt, mit dem Aufgaben oder Aufgabenelemente auftraten? War die Abfolge langsam und geruhsam oder schnell und hektisch?



18. **Leistung:** Wie erfolgreich haben Sie Ihrer Meinung nach die vom Versuchsleiter (oder Ihnen selbst) gesetzten Ziele erreicht? Wie zufrieden waren Sie mit Ihrer Leistung bei der Verfolgung dieser Ziele?



19. **Anstrengung:** Wie hart mussten Sie arbeiten, um Ihren Grad an Aufgabenerfüllung zu erreichen?



Abbildung A.28: Fragebogen zur Hauptstudie Seite 5

20. Frustration: Wie unsicher, entmutigt, irritiert, gestresst und verärgert (versus sicher, bestätigt, zufrieden, entspannt und zufrieden mit sich selbst) fühlten Sie sich während der Aufgabe?



21. Haben Sie sonstige Anmerkungen, Kritik oder Ideen?

A.4 Allgemeine Fragen

Tabelle A.1: Frage zur Gewohnheit des Schreibens auf Tastaturen

Wie gewohnt sind Sie daran Texte auf Tastaturen zu schreiben?						
	Seltenes Schreiben und / oder sehr wenig Text	Unregelmäßiges Schreiben und / oder wenig Text	Regelmäßiges Schreiben und / oder mittellange Texte	Häufiges Schreiben und / oder lange Texte	Sehr häufiges Schreiben und / oder sehr lange Texte	∅
Pilotstudie	1	0	3	1	3	3,625
Hauptstudie	0	0	3	2	5	4,2

Tabelle A.2: Frage zur Gewohnheit des Schreibens auf Mobilgeräten

Wie gewohnt sind Sie daran Texte auf Mobilgeräten zu schreiben?						
	Seltenes Schreiben und / oder sehr wenig Text	Unregelmäßiges Schreiben und / oder wenig Text	Regelmäßiges Schreiben und / oder mittellange Texte	Häufiges Schreiben und / oder lange Texte	Sehr häufiges Schreiben und / oder sehr lange Texte	∅
Pilotstudie	1	2	0	3	2	3,375
Hauptstudie	0	2	3	2	3	3,6

Tabelle A.3: Frage zur Gewohnheit an die virtuelle Realität

Wie gewohnt waren Sie vor der Studie an die Benutzung der VR?						
	Erste Benutzung in Studie	Wenige Benutzungen (2-5)	Wiederholte Benutzung (6-10)	Regelmäßige kurze Benutzung	Häufige langzeitige Benutzung	∅
Pilotstudie	1	2	3	1	1	2,875
Hauptstudie	0	3	5	0	2	3,1

Tabelle A.4: Frage zur Gewohnheit des Schreibens in der Virtuellen Realität

Wie gewohnt waren Sie vor der Studie in der VR Texte zu schreiben?					
	Erstes Schreiben in Studie	Seltenes Schreiben und /oder sehr wenige Worte	Unregelmäßiges Schreiben und / oder wenige Worte	Regelmäßiges Schreiben und / oder mittellange Texte	Häufiges Schreiben und / oder lange Texte
Pilotstudie	5	3	0	0	0
Hauptstudie	5	2	3	0	0

A.5 Pilotstudie

Tabelle A.5: Frage zum Erlernen der Bedienung in der Pilotstudie

Wie schnell konnten Sie die Bedienung der Schreibmethoden auf einer Skala von 1 (sehr langsam) bis 10 (sehr schnell) lernen?									
	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	∅
Thumbpad-Verfahren									
Tastatur	7	5	8	10	6	8	7	10	7,625
Kreise	5	4	7	9	5	7	5	9	6,375
Gesten	2	1	3	5	1	2	3	4	2,625
Cluster	6	3	5	7	2	5	3	5	4,5
Controller-Verfahren									
Drumstick	8	9	7	9	8	9	9	-	8,429
Touch	9	8	9	9	10	9	10	-	9,143
Laser	7	10	6	8	6	8	7	-	7,429
Gaze	5	6	4	8	4	7	4	-	5,429

Tabelle A.6: Frage zu Vor- & Nachteilen der Thumbpad-Verfahren in der Pilotstudie

Welche Vor- & Nachteile sehen Sie an den Thumbpad-Verfahren?	
Allgemein	4 · (+) Im Sitzen verwendbar
Tastatur	3 · (+) Position der Buchstaben bereits bekannt (+) Übersichtlich und alt-vertraut 2 · (+) Sehr gewohnt und simpel (+) Gewohntes Tastaturlayout (+) Tasten auswählbar, wie Computertastatur (–) Zweiteilung der Tastatur gewöhnungsbedürftig (–) Alles sehr klein (–) Strenge Teilung in rechts und links
Kreise	(+) Relativ unkompliziert (–) Etwas friemelig (–) Vorwiegende Nutzung der rechten Hand (–) Anordnung der Zeichen gewöhnungsbedürftig (+) Nur einmal Auswahl und Button-Press am Touchpad für einen Buchstaben (im Vergleich zu Thumbpad-Cluster) (–) Cursortasten neben wichtigen Funktionstasten (–) Zu schnelles Verrutschen
Gesten	(–) Sehr schwierig fehlerfrei einzugeben (–) Habe oft versehentlich das falsche Zeichen genommen (–) Hohe Fehlerrate durch unbeabsichtigte Bewegungen (–) Es kam vor, dass das Thumbpad zu klein war, wenn man nicht in der Mitte angefangen hat (–) Häufige Falschwahl (–) Sehr fehleranfällig (+) Einhändige Bedienung möglich (–) Steile Lernkurve (–) Extrem schwierig zu lernen (–) Lange Eingewöhnungszeit (+) Schnell sobald eingewöhnt (+) Hohes Potential sobald 2-händig bedient
Cluster	2 · (–) Gute rechts-links Koordination nötig (–) Erfordert zu Beginn viele Augenbewegungen zwischen den Controllern. (+) Alphabetische Anordnung (+) Sehr viele Buttons belegbar (–) Schwierig zu lernen (–) Diagonale Richtung schwer auszuwählen

Tabelle A.7: Frage zu Vor- & Nachteilen der Controller-basierten Verfahren in der Pilotstudie

Welche Vor- & Nachteile sehen Sie an den Controller-basierten Verfahren?	
Allgemein	(+) Layout und Wortvorhersage wie beim Handy altvertraut
Drumstick	(+) Zum Sitzen besser geeignet als Touch (+) Beliebige Sticklänge (-) Unangenehme Handhaltung (+) Höhere Reichweite (-) Lange Texte anstrengend (+) Intuitiv (+) Schnelle Reaktionszeit (+) Gut bedienbar (-) Ungewohnte Verlängerung der Hände 2 · (+) Macht Spaß (+) Geht schnell
Touch	3 · (+) Intuitiv und einfach (-) Unangenehm im Sitzen (+) Sehr gewohnt, fast wie Tastatur mit 2 Fingern (-) Lange Texte anstrengend (+) Schnelle Reaktionszeit (+) Gut bedienbar (+) Geht schnell
Laser	(+) Schnell und intuitiv (-) Schnell Fehler durch Tracking-Ruckler (-) Genauigkeit und lange Texte anstrengend (+) Schnelle Reaktionszeit (+) Gut bedienbar (+) Tastatur kann weiter weg sein (-) Es muss genau gezielt werden (+) Gut, dass man den Trigger nicht durchdrücken muss (-) Etwas langsamer, da man Buchstaben auswählt und bestätigt
Gaze	(+) Handfrei (-) Unintuitiv (-) Diktattext nicht im Fokus / Nicht möglich gleichzeitig zu lesen und zu schreiben 3 · (-) Dauert sehr lange und ist anstrengend (-) Sehr gewöhnungsbedürftig 3 · (-) Fehleranfällig 2 · (-) Anstrengend für den Nacken (-) Kein Umschauen möglich (-) Potential der Controller wird nicht genutzt (-) Stressig, wenn man suchen muss

Tabelle A.8: Frage zu Verbesserungsvorschlägen in der Pilotstudie

Was könnte man an den Verfahren verbessern?	
Allgemein	-NEO2-Layout bei tastaturähnlichen Verfahren
Thumbpad-Tastatur	-Trennung der Tastatur auf zwei Controller irritierend
Thumbpad-Kreise	-2 · Radius des Kreises evtl. kleiner -Alternative Belegung als Vorschau anzeigen -Evtl. Buchstabenverteilung ähnlich wie bei Tastaturen -Anordnung an Buchstaben-Touple der Sprache anpassen. Abwechselnde links-rechts Verwendung. -Zeichen entsprechend ihrer Häufigkeit o.Ä. anordnen -Funktionstasten weiter von anderen Tasten entfernen
Thumbpad-Gesten	-Felder gleichmäßig gedreht anordnen. So musste man sich immer auf verschiedene Orientierungen einstellen. -Deutlich machen, welches Zeichen ausgewählt ist -Kürzere Wege um Buchstaben zu schreiben, erhöht allerdings die Fehleranfälligkeit -Anfangsposition bereits in einer der Gruppen und die Anordnung drehen -Zeichen entsprechend ihrer Häufigkeit o.Ä. anordnen
Thumbpad-Cluster	-Mit links zu navigieren schwieriger für rechtshändige Personen, aber kein Plan was man tun könnte -Vorschau des alternativen Layouts anzeigen -Zwischen Layouts durch Drücken des linken Thumbpads wechseln -Alphabetisch, d.h. Buchstaben künftige Quadrupel der Sprache -Zeichen entsprechend ihrer Häufigkeit o.Ä. anordnen
Drumstick-Verfahren	-2 · Länge der Sticks frei wählen können
Touch-Verfahren	-Länge variabel machen -Den weißen Zylinder etwas kleiner -Spitzes Ende des Colliders
Laser-Verfahren	-Tracking verbessern -Leichter Tiefpass-Filter auf die Position -Durchdrücken des Triggers könnte Großbuchstaben schreiben
Gaze-Verfahren	-Thresholding vergrößern um unabsichtliches Abbrechen zu verringern -Tasten wie Shift/Space/etc. am Controller -Vielleicht mit Eyetracking besser? -Aktivierung durch Grip

Tabelle A.9: Frage zur positiven Verfahrensauswahl in der Pilotstudie

Welches Verfahren hat Ihnen am besten gefallen? Wieso?	
Thumbpad-Verfahren	
5 · Tastatur	-Am einfachsten -4 · Gewohnt -Alle Buchstaben simpel erreichbar -Schnellstes Schreiben
2 · Kreise	-Weil es am wenigsten Gewöhnung brauchte -Schnell, großes Potential
1 · Gesten	-Anfangs schwierig aber hohes Potential
Controller-Verfahren	
1 · Touch oder Drumstick	Es hat am wenigsten Gewöhnung gebraucht & war dadurch nicht sehr anstrengend
1 · Laser	Einfachste Bedienung
3 · Touch	-2 · sehr intuitiv -Ähnlich zu Tastatur
2 · Drumstick	-Weil man sich so schnell vorkam im Vergleich zu den anderen

Tabelle A.10: Frage zur negativen Verfahrensauswahl in der Pilotstudie

Welches Verfahren hat Ihnen am wenigsten gefallen? Wieso?	
Thumbpad-Verfahren	
6 · Gesten	-3 · Hohe Fehlerquote -Nicht genug Platz auf dem Thumbpad -Benötigt exakte Bewegungen -Schlechtes Error-Cancelling -Zu komplex für Anfänger
-1 · Kreise	Zu schneller Wechsel zwischen den Buchstaben
-1 · Cluster	Anspruchsvolle Koordination, sowie zwei Button-Presses notwendig
Controller-Verfahren	
7 · Gaze:	-3 · Gewöhnungsbedürftig -Belastung der Nackenmuskulatur -Anstrengend und langsam -Der Ladekreis macht Druck -Schwer gleichzeitig zu lesen und zu schreiben

Tabelle A.11: Frage zur Fehlerquote in der Pilotstudie

Wie hoch würden Sie Ihre Fehlerquote beim Schreiben der Diktate mit den Verfahren schätzen?									
	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	Ø
TP-Tastatur	25%	10%	10%	5%	15%	5%	10%	5%	10,625%
TP-Kreise	20%	20%	15%	10%	15%	10%	60%	15%	20,625%
TP-Gesten	50%	10%	25%	20%	20%	20%	20%	20%	23,125%
TP-Cluster	30%	10%	20%	10%	15%	20%	10%	25%	17,5%
Drumstick	30%	10%	15%	15%	10%	5%	20%	-	15%
Touch	40%	10%	10%	10%	5%	10%	5%	-	12,857%
Laser	45%	10%	10%	10%	15%	15%	40%	-	20,714%
Gaze	80%	90%	5%	10%	10%	20%	35%	-	35,714%

TP = Thumbpad

Tabelle A.12: Frage zur Schreibgeschwindigkeit in der Pilotstudie

Bitte bewerten Sie Ihre Schreibgeschwindigkeit mit den nachfolgenden Verfahren von 0 (schlecht) bis 10 (gut).									
	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	Ø
TP-Tastatur	4	5	8	9	3	6	5	9	6,125
TP-Kreise	4	3	7	7	1	5	2	5	4,25
TP-Gesten	0	1	3	4	0	2	4	4	2,25
TP-Cluster	2	2	5	4	1	4	3	3	3
Drumstick	7	9	7	8	3	8	8	-	7,143
Touch	7	9	9	9	4	8	8,5	-	7,786
Laser	6	9	6	7	3	7	9	-	6,714
Gaze	2	5	4	6	2	6	2	-	3,857
Tastatur	8	8	10	10	10	7	10	10	9,125

TP = Thumbpad

Tabelle A.13: Frage zur technischen Beeinträchtigung in der Pilotstudie

Aus einer Skala von 0 (gar nicht) bis 10 (sehr), wie sehr hat Sie die Technik bei der Aufgabe beeinträchtigt?									
Verfahrensgruppe	VP1	VP2	VP3	VP4	VP5	VP6	VP7	VP8	Ø
Thumbpad	7	3	2	4	0	9	1	1	3,375
Controller	7	1	4	2	3	0	3	-	2,857

Tabelle A.14: Frage zu Gründen der technischen Beeinträchtigung in der Pilotstudie

Welche technischen Aspekte haben Sie bei der Aufgabe beeinträchtigt? (Unschärfe Texte, Tracking-Probleme, etc.)
Thumbpad-Verfahren
<ul style="list-style-type: none"> -3 · Unschärfe Texte -Controller zu groß für Hände -Unschärfe im peripheren Bereich -Tracking -Griptasten sehr unangenehm -FOV
Controller-Verfahren
<ul style="list-style-type: none"> -Unschärfer Text bei Gaze -Druck der VR-Brille -Brille ruckelt bei schnellen Bewegungen -Unschärfe im peripheren Bereich -Teilweise hat es gelaggt (wenig FPS) -3 · Tracking -Unschärfe Texte

Tabelle A.15: Frage zur geistigen Anforderung in der Pilotstudie

Wie hoch war die geistige Anforderung der Aufgabe für Sie (Suchen, Denken, Erinnern, etc.)?
Thumbpad-Verfahren
<ul style="list-style-type: none"> -Relativ hoch, wegen schnellem Umstellen auf verschiedene Schreib-Techniken -Sehr hoch -Häufiges Suchen der Buchstaben -2 · Gering -2 · Hoch -Abhängig vom Verfahren nahezu gar nicht bis hoch
Controller-Verfahren
<ul style="list-style-type: none"> -Viel suchen & passend ausrichten bei Gaze. Sonst eher normale Anforderung ähnlich normalem Abtippen -2 · Mittel -2 · Sehr gering, Verfahren sehr intuitiv -Gering -Normal, höher bei Gaze

Tabelle A.16: Frage zur körperlichen Anstrengung in der Pilotstudie

Wie hoch war die körperliche Anstrengung der Aufgabe für Sie (Bewegen, Drücken, Halten, etc.)?
Thumbpad-Verfahren
<ul style="list-style-type: none"> -3 · Gering -Hoch -Kaum Bewegung notwendig -2 · Sehr gering -Griptasten unangenehm zu drücken
Controller-Verfahren
<ul style="list-style-type: none"> -Gaze hoch, sonst gering. Kopf in komischen Positionen. -2 · Niedrig -Das Stehen kann anstrengend werden -Recht anstrengend -Sehr gering -Griptasten unangenehm zu betätigen

Tabelle A.17: Frage zur Erfolgseinschätzung in der Pilotstudie

Wie erfolgreich haben Sie Ihrer Meinung nach die Aufgabe bewältigt? Sind Sie zufrieden mit der erreichten Schreibgeschwindigkeit Ihrer bevorzugten Methode?
Thumbpad-Verfahren
<ul style="list-style-type: none"> -Nicht viele Fehler, aber sehr langsam geschrieben -Aufgabe wenig erfolgreich bewältigt, Schreibgeschwindigkeit zu langsam -Kann wesentlich besser sein -Schlecht, viel zu langsam -8/10 -Nicht so erfolgreich. Höhere Geschwindigkeiten gewohnt -Viel zu langsam, müsste mehr üben -Zufrieden, mit der Tastatur konnte ich deutlich mehr Text schreiben
Controller-Verfahren
<ul style="list-style-type: none"> -Gut & zufrieden -Gute Bewältigung, zufrieden mit Geschwindigkeit -Für eine virtuelle Tastatur sehr zufrieden -8/10 -War ganz ok -Das Gaze-Verfahren empfand ich als zu langsam -Zu langsam aufgrund mangelnder Erfahrung

Tabelle A.18: Frage zur Frustration in der Pilotstudie

Wie frustriert fühlten Sie sich während der Aufgabe (Unsicherheit, Stress, Verärgerung)?
Thumbpad-Verfahren
<ul style="list-style-type: none"> -Fast gar nicht -Teilweise sehr (Gesten), ansonsten leicht. Unsicher bzgl. Handling. -Manchmal ärgerlich die falsche Taste gedrückt zu haben -2/10 -Sehr -2 · Gar nicht: Es war spannend und ungewohnt. -Nur bei Kreisen geärgert, da ich beim Bearbeiten vertippe -Abhängig vom Verfahren, gar nicht bis leicht verärgert
Controller-Verfahren
<ul style="list-style-type: none"> -Stress bei Gaze, sonst gar nicht frustriert -Gar nicht frustriert -Relativ sicher -0/10 -Wenig bis stark (Gaze) -Gar nicht

Tabelle A.19: Frage zur Präferenz für die Zukunft in der Pilotstudie

Welches Verfahren (in optimierter Form) würden Sie in Zukunft in der Virtuellen Realität verwenden wollen um Texte zu schreiben?
Thumbpad-Verfahren
<ul style="list-style-type: none"> -6 · Thumbpad-Tastatur -Evtl. die Kreise -Thumbpad-Kreise -Gesten
Controller-Verfahren
<ul style="list-style-type: none"> -4 · Drumstick oder Touch -Laser -2 · Touch

Tabelle A.20: Frage zu negativen Auffälligkeiten in der Pilotstudie

Ist Ihnen bei der Verwendung der Verfahren etwas besonders Negatives aufgefallen?
-10 · Nein -Bei den Thumbpad-Verfahren sollte man nicht für das Schreiben eines Buchstabens zwei Operationen ausführen müssen -Steile Lernkurve bei Thumbpad-Verfahren -2 · Verwendung der Grip-Tasten manchmal schwierig -Grip-Tasten für meine Finger falsch positioniert

Tabelle A.21: Frage zu sonstigen Anmerkungen in der Pilotstudie

Haben Sie sonstige Anmerkungen, Kritik oder Ideen?
-12 · Keine -Autovervollständigung suboptimal -Das Vibrationsfeedback ist sehr hilfreich -Anpassung des Controllers, Buttons unter dem Touchpad oder zweiter Trigger

A.6 Hauptstudie

Tabelle A.22: Frage zu Vor- & Nachteilen der Drumstick-Verfahren in der Hauptstudie

Welche Vor- & Nachteile sehen Sie hinsichtlich der langzeitigen Benutzung bei den getesteten Verfahren?	
Drumstick-Verfahren	<ul style="list-style-type: none"> (+) Einfache Handhabung (+) Koordination verbessert sich 2 · (+) Schnell & vertraute Tastatur (-) 2 Sticks, statt bis zu 10 Finger (-) Teils sehr viel Bewegung notwendig (+) Sehr intuitiv zu bedienen -Teils unergonomische Haltung → durch Kippen der Tastatur entfallen 2 · (-) Auf Dauer anstrengend, da Stehen nötig (-) Wird nicht schneller als das Schreiben auf der Tastatur (-) Handbewegungen anstrengender als Fingerbewegung (-) Auf Dauer anstrengend (-) Ermüdung der Arme (-) Evtl. zu hohe Belastung der Sehnen (+) Arbeit im Stehen positiv für den Rücken (-) Versehentliches Abschicken des Textes leicht möglich (-) Man muss sich an die verlängerten Arme gewöhnen (+) Länge des Sticks einstellbar 2 · (-) Auf Dauer ermüdend (+) Spaß (+) Schreiben geht recht schnell → Wird immer besser (+) Intuitiv, wie eine Tastatur

Tabelle A.23: Frage zu Vor- & Nachteilen der Thumbpad-Verfahren in der Hauptstudie

Welche Vor- & Nachteile sehen Sie hinsichtlich der langzeitigen Benutzung bei den getesteten Verfahren?	
Thumbpad-Verfahren	<ul style="list-style-type: none"> (-) Etwas schwerer zu erlernen, nicht intuitiv (-) Langsamer 2 · (+) Körperlich nicht anstrengend (+) Benötigt weniger Konzentration (-) Anstrengender für die Hände (-) Erste Verwendung sehr anstrengend (+) Könnte effizient werden → ist effizient geworden (-) Anfangs mittig liegende Zeichen oft mit falschem Controller versucht (+) Im Sitzen möglich (+) Theoretisch blindes Schreiben möglich (-) Fingerspitzengefühl nötig, etwas friemelig (-) Ungenauer als Drumsticks (-) Limitiert durch die kleine Fläche des Touchpads (-) Präzise Bewegungen nötig (-) Ungenauigkeit der nicht-dominanten Hand (-) Wortvorschläge außerhalb des üblichen Sichtfeldes (-) Ungenauigkeit in der Peripherie (-) Belastung des Daumengelenks (-) Belastung der Daumensehnen und -muskeln (+) Bequem zu benutzen, auch im Sitzen (+) Sehr schnelles Schreiben möglich, Lernkurve (-) Großbuchstaben durch Gedrückthalten verursacht Fehler (+) Im Sitzen, ohne körperliche Anstrengung möglich

Tabelle A.24: Frage zu Verbesserungsvorschlägen in der Hauptstudie

Welche Verbesserungen bzw. Veränderungen würden Sie sich an den getesteten Verfahren erhoffen, um sie langfristig in der VR einzusetzen?	
Drumstick-Verfahren	<ul style="list-style-type: none"> -2 · Zur Großschreibung sollte das Drücken eines beliebigen Triggers möglich sein (nicht nur der des jeweiligen Sticks) -Vielleicht Enter etwas deutlicher hervorheben -Auch nach „:“ und Ähnlichem automatisch ein Leerzeichen -Durch Halten der Taste mehrfach den gleichen Buchstaben schreiben -Ergonomischere Tastaturposition → vorhanden -Shift-Taste an originale Position -Leichtere Controller -Textvorschläge präserter anzeigen -Je länger der Stick, desto größer sollte die Kollision sein -Bessere Wortvorhersagen für Fachwörter -Kombination mit anderen Verfahren, z.B. Gaze -Controller-Button für Leerzeichen, da häufig benutzt
Thumbpad-Verfahren	<ul style="list-style-type: none"> -Grip-Taste schwer zu erreichen -Das Umschalten auf Umlaute und Zahlen könnte nach Loslassen des Triggers automatisch wieder zurückgehen -Tasten am Rand besser erreichbar machen → Kein Problem mehr -Textvorschläge präserter anzeigen -Bedienung nur mit dominanter Hand -Buchstaben nur im zentralen Bereich des Pads (oder anpassbar) -Vorschläge rechts und links -Leertaste links kaum genutzt, Buttons könnten umverteilt werden -Bessere Wortvorhersagen für Fachwörter -Custom Key mappings -Controller-Button für Großbuchstaben

Tabelle A.25: Frage zur Verfahrensauswahl für kurze Texte in der Hauptstudie

Welches Verfahren gefällt Ihnen besser um kurze Texte zu verfassen? Wieso?	
8 · Drumstick	-4 · Spaßfaktor -Hände nicht so verkrampft -7 · Geht schnell -3 · Intuitiver -Zur Not auch mit einer Hand
2 · Thumbpad	-Fehlerquote geringer -Weniger störend in immersiven Settings

Tabelle A.26: Frage zur Verfahrensauswahl für lange Texte in der Hauptstudie

Welches Verfahren gefällt Ihnen besser um lange Texte zu verfassen? Wieso?	
7 · Drumstick	-3 · Spaßfaktor -Hände nicht so verkrampft -6 · Geht schnell -3 · Intuitiver -Genauer
3 · Thumbpad	-Fehlerquote geringer -Bequeme Haltung, Arme können abgelegt werden -Weniger anstrengend

Tabelle A.27: Frage zur technischen Beeinträchtigung in der Hauptstudie

Wurden Sie durch technische Aspekte bei der Aufgabe beeinträchtigt? (Periphere Unschärfe, Unangenehme Bedienung, etc.)
-Periphere Unschärfe -Akku der Controller zwischenzeitig alle -2 · Etwas unscharf -Zu großes Thumbpad -2 · Keine -Wortvorschläge bei langen Wörtern sehr klein -Nicht wesentlich -Selten kleine Tracking-Probleme

Tabelle A.28: Frage zur Verbesserung der Geschwindigkeit in der Hauptstudie

Wie hoch im Vergleich zur jeweils vorherigen Trainingseinheit würden Sie die Verbesserung Ihrer Schreibgeschwindigkeit einschätzen?								
	Drumstick				Thumbpad			
	T2	T3	T4	T5	T2	T3	T4	T5
VP11	7%	5%	3%	0%	10%	5%	0%	2%
VP12	1%	0%	5%	0%	5%	5%	5%	5%
VP13	1%	1%	1%	1%	20%	16%	10%	6%
VP14	5%	1%	1%	1%	10%	5%	3%	1%
VP15	0%	5%	10%	15%	15%	15%	0%	5%
VP16	50%	-10%	30%	-5%	20%	-10%	15%	0%
VP17	-10%	20%	3%	7%	0%	7%	2%	5%
VP18	20%	25%	15%	0%	15%	18%	17%	0%
VP19	20%	20%	15%	5%	5%	10%	5%	0%
VP20	10%	5%	10%	0%	5%	0%	20%	0%

T = Trainingseinheit

Tabelle A.29: Frage zur Fehlerquote in der Hauptstudie

Wie hoch würden Sie Ihre Fehlerquote beim Schreiben der Diktate bei den einzelnen Trainingseinheiten schätzen?										
	Drumstick					Thumbpad				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
VP11	5-7%	7%	5%	3%	3%	5%	5%	5%	3%	3%
VP12	25%	25%	20%	15%	15%	19%	25%	25%	15%	15%
VP13	5%	4%	3%	3%	3%	20%	15%	10%	8%	6%
VP14	5%	10%	10%	5%	5%	35%	25%	10%	5%	5%
VP15	5%	10%	10%	15%	5%	15%	15%	10%	10%	10%
VP16	3%	2%	5%	1%	4%	2%	3%	5%	3%	2%
VP17	3%	7%	2%	4%	3%	8%	7%	4%	6%	2%
VP18	8%	8%	7%	6%	5%	10%	9%	9%	7%	6%
VP19	10%	5%	3%	2%	1%	8%	5%	3%	2%	1%
VP20	10%	5%	5%	6%	5%	5%	5%	5%	5%	5%
∅	8%	8,3%	7%	6%	4,9%	12,7%	11,4%	8,6%	6,4%	5,5%

T = Trainingseinheit

A.6.1 NASA-TLX Fragebogen

Tabelle A.30: Frage NASA-TLX Geistige Anforderung in der Hauptstudie

Geistige Anforderung: Wie viel geistige Anstrengung war bei der Informationsaufnahme und -verarbeitung erforderlich (z.B. Denken, Entscheiden, Rechnen, Erinnern, Hinsehen, Suchen...)? War die Aufgabe leicht oder anspruchsvoll, einfach oder komplex, erforderte sie hohe Genauigkeit oder war sie fehlertolerant?						
	T1	T2	T3	T4	T5	Ø
VP11	16	17	17	17	17	84%
VP12	13	13	13	7	2	48%
VP13	16	15	13	8	4	56%
VP14	10	10	10	10	10	50%
VP15	13	13	13	13	13	65%
VP16	15	15	17	12	12	71%
VP17	15	15	15	15	15	75%
VP18	16	14	12	8	8	58%
VP19	8	6	6	5	4	29%
VP20	15	12	12	7	7	53%
Ø	68,5%	65%	64%	51%	46%	58,9%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

Tabelle A.31: Frage NASA-TLX Körperliche Anforderung in der Hauptstudie

Körperliche Anforderung: Wie viel körperliche Aktivität war erforderlich (z.B. Ziehen, Drücken, Drehen, Steuern, Aktivieren...)? War die Aufgabe leicht oder schwer, einfach oder anstrengend, erholsam oder mühselig?						
	T1	T2	T3	T4	T5	Ø
VP11	15	13	16	16	16	76%
VP12	11	11	17	8	5	52%
VP13	0	0	0	0	0	0%
VP14	11	11	11	11	10	54%
VP15	14	14	14	14	14	70%
VP16	4	4	7	5	5	25%
VP17	13	11	14	13	13	64%
VP18	14	14	12	10	10	60%
VP19	10	8	7	6	6	37%
VP20	15	13	8	6	5	47%
Ø	53,5%	49,5%	53%	44,5%	42%	48,5%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

Tabelle A.32: Frage NASA-TLX Zeitliche Anforderung in der Hauptstudie

Zeitliche Anforderung: Wie viel Zeitdruck empfanden Sie hinsichtlich der Häufigkeit oder dem Takt, mit dem Aufgaben oder Aufgabenelemente auftraten? War die Abfolge langsam und geruhsam oder schnell und hektisch?						
	T1	T2	T3	T4	T5	Ø
VP11	8	8	8	8	8	40%
VP12	8	8	8	8	8	40%
VP13	10	10	10	10	10	50%
VP14	2	2	2	2	2	10%
VP15	12	12	12	12	12	60%
VP16	7	7	10	5	5	34%
VP17	4	5	4	4	4	21%
VP18	14	14	14	14	14	70%
VP19	4	6	5	5	4	24%
VP20	10	10	8	10	10	48%
Ø	39,5%	41%	40,5%	39%	38,5%	39,7%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

Tabelle A.33: Frage NASA-TLX Leistung in der Hauptstudie¹

Leistung: Wie erfolgreich haben Sie Ihrer Meinung nach die vom Versuchsleiter (oder Ihnen selbst) gesetzten Ziele erreicht? Wie zufrieden waren Sie mit Ihrer Leistung bei der Verfolgung dieser Ziele?						
	T1	T2	T3	T4	T5	Ø
VP11	11	14	15	10	15	65%
VP12	14	14	3	16	16	63%
VP13	7	7	7	7	7	35%
VP14	11	8	9	13	14	55%
VP15	16	16	16	16	16	80%
VP16	12	12	8	14	14	60%
VP17	9	7	11	10	11	48%
VP18	12	14	12	16	16	70%
VP19	12	13	13	14	15	67%
VP20	7	10	10	14	14	55%
Ø	55,5%	57,5%	52%	65%	69%	59,8%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

¹Die Skala wurde gegenüber NASA (2018) invertiert, d.h. höher ist besser.

Tabelle A.34: Frage NASA-TLX Anstrengung in der Hauptstudie

Anstrengung: Wie hart mussten Sie arbeiten, um Ihren Grad an Aufgabenerfüllung zu erreichen?						
	T1	T2	T3	T4	T5	Ø
VP11	12	14	14	15	15	70%
VP12	8	8	17	17	6	56%
VP13	13	13	13	13	13	65%
VP14	11	10	10	8	7	46%
VP15	14	14	14	14	14	70%
VP16	6	6	10	8	8	38%
VP17	8	8	6	6	6	34%
VP18	14	14	14	14	14	70%
VP19	14	12	11	10	10	57%
VP20	16	14	14	16	16	76%
Ø	58%	56,5%	61,5%	60,5%	54,5%	58,2%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

Tabelle A.35: Frage NASA-TLX Frustration in der Hauptstudie

Frustration: Wie unsicher, entmutigt, irritiert, gestresst und verärgert (versus sicher, bestätigt, zufrieden, entspannt und zufrieden mit sich selbst) fühlten Sie sich während der Aufgabe?						
	T1	T2	T3	T4	T5	Ø
VP11	4	3	3	6	5	21%
VP12	17	17	19	3	1	57%
VP13	14	10	7	4	1	36%
VP14	4	7	6	4	3	24%
VP15	14	14	14	14	14	70%
VP16	2	2	10	3	3	20%
VP17	5	6	4	5	4	24%
VP18	8	6	6	4	6	30%
VP19	6	6	6	5	4	27%
VP20	7	7	7	3	3	27%
Ø	40,5%	39%	41%	25,5%	22%	33,6%

T = Trainingseinheit; Angaben auf einer Skala von 0 bis 20

Tabelle A.36: Frage zu sonstigen Anmerkungen in der Hauptstudie

Haben Sie sonstige Anmerkungen, Kritik oder Ideen?
-7 · Keine
-War cool :)
-Das Drumstick-Verfahren wäre für Menschen mit beeinträchtigten Händen ein gutes Verfahren zum Schreiben
-Die Verbesserung der Vorschläge gegenüber der Pilotstudie war deutlich bemerkbar

Versicherung

Der Unterzeichnende versichert, dass er die vorliegende schriftliche Arbeit selbstständig verfasst und keine anderen als die von ihm/ihr angegebenen Hilfsmittel benutzt hat. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, wurden in jedem Fall unter Angabe der Quellen kenntlich gemacht. Dies gilt auch für beigegebene Zeichnungen, bildliche Darstellungen, Skizzen und dergleichen. Dem Unterzeichnenden ist bewusst, dass jedes Zuwiderhandeln (Einreichen einer Arbeit, die wörtlich oder nahezu wörtlich, ganz oder zu Teilen aus einer Arbeit oder mehreren Arbeiten [publiziert im Internet, in Zeitschriften, Monographien etc.] anderer übernommen ist) als Täuschungsversuch (siehe §18 MPO) gelten kann, der die Bewertung der Arbeit mit „nicht ausreichend“ zur Folge hat.

Bielefeld, 16. Oktober 2018

Dominik Stallmann