

# Untersuchung des Brain- Computer-Interfaces „NextMind“ als Steuerermethode für medizinische Hilfsmittel

Eingereicht von: Jakob Schmock  
Matrikelnummer: 4825128  
Am: 08.10.2022

1. Gutachter – Prof. Dr. Lutz Prechelt
  - a. Freie Universität Berlin, Fachbereich Informatik
  - b. Leiter der Forschungsgruppe Software Engineering
2. Gutachter – Prof. Dr.-Ing. Volker Roth
  - a. Freie Universität Berlin, Fachbereich Informatik
  - b. Leiter der Forschungsgruppe Secure Identity



# Selbstständigkeitserklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Freitag, 07. Oktober 2022

X

---

Jakob Schmock



## Zusammenfassung

*Viele* Menschen sind körperlich so stark beeinträchtigt, dass ihnen die Bedienung moderner Technik schwerfällt oder diese sogar unmöglich ist.

*In* dieser externen Bachelorarbeit bei *Munevo GmbH* wird das *NextMind*, ein *Brain-Computer-Interface*, getestet, um dessen Nutzen als medizinisches Hilfsmittel für unterstützende Programme zu untersuchen.

*Dafür* wurden zwei Tests entworfen und deren Ergebnisse evaluiert.

*Die* Ergebnisse der Tests zeigen, dass eine Bedienung von Programmen mittels *NextMind* durchaus möglich ist.

*Jedoch* ist das Spektrum der Möglichkeiten begrenzt, da einige Limitationen des Gerätes Schwierigkeiten bei der Umsetzung mit sich bringen. Die Aktivierungszeit von Schaltflächen ist im Vergleich zur Bedienung mit herkömmlichen Methoden, beispielsweise Maus und Tastatur, relativ hoch. Zusätzlich ist das Gerät limitiert auf zehn gleichzeitig nutzbare Bedienelemente und erfordert ein gewisses Maß an Konzentration, was nach längerem Benutzen zu Anstrengung führt.

## Summary

*Many* people are physically impaired to such an extent that they find it difficult or even impossible to operate modern technology.

*In* this external bachelor thesis at *Munevo GmbH*, the *NextMind*, a *brain-computer interface*, is tested to investigate its usefulness as a medical aid for assistive programs.

*For* this purpose, two tests were designed and their results evaluated.

*The* results of the tests show that operation of programs using *NextMind* is quite possible.

*However*, the range of possibilities is limited because some limitations of the device bring difficulties in implementation. The activation time of buttons is relatively high compared to operation with conventional methods, for example, mouse and keyboard. In addition, the device is limited to ten simultaneously usable control elements and requires a certain amount of concentration, which leads to strain after prolonged use.

## Inhalt

I.	Einführung.....	9
1.	Menschen mit Beeinträchtigungen.....	9
2.	Forschungsstand .....	9
3.	Munevo .....	10
II.	Gehirn-Computer-Schnittstelle (BCI) .....	11
1.	Elektroenzephalografie .....	11
2.	BCI gegenüber anderen alternativen Eingabemethoden .....	12
3.	Projektfindung .....	14
III.	NextMind .....	16
1.	Beschreibung des NextMind .....	16
2.	SDK und Unity .....	18
3.	Schwierigkeiten.....	19
IV.	Tests zum Optimieren der Einstellungen.....	21
1.	Hintergrund.....	21
2.	Erster Test: Größe und Aktivierungszeit .....	21
3.	Erster Test: Ergebnisse.....	23
4.	Zweiter Test: Zuversichtlichkeitswert .....	26
5.	Zweiter Test: Ergebnisse .....	29
6.	Fazit.....	33
V.	Ausblick in die Zukunft.....	35
VI.	Literaturverzeichnis .....	36
VII.	Abbildungsverzeichnis .....	38
VIII.	Anhang .....	39





# I. Einführung

## 1. Menschen mit Beeinträchtigungen

Viele Menschen leben mit körperlichen oder geistigen Beeinträchtigungen. Laut dem Statistischen Bundesamt sind beinahe 10% der deutschen Bevölkerung schwerbehindert [1]. Für viele dieser Personen bedeuten diese Beeinträchtigungen einschneidende Auswirkungen auf den Alltag, sei es im Privatleben oder bei der Arbeit. Beispielsweise hindern fehlende Fahrstühle Rollstuhlgebundene Menschen daran, effizient mit öffentlichen Verkehrsmitteln fahren zu können oder Veranstaltungen zu besuchen. Auch die Bedienung vieler technischen Geräte kann körperlich behinderten Menschen schwerfallen oder sogar unmöglich sein. Jedoch wird gerade in der heutigen Zeit die Möglichkeit solche Geräte benutzen zu können beinahe vorausgesetzt.

## 2. Forschungsstand

Es gibt bereits Hilfsmittel, um die Bedienung von Computern und ähnlichem zu ermöglichen oder erleichtern. Ein existierendes Hilfsmittel ist die „*Sip-and-Puff*“ Technologie, bei der der/die Benutzer/in durch eine am Mund gelegene Röhre ein- oder ausatmet und dadurch Befehle an das angeschlossene Gerät übermittelt [2]. Oft wird dies mit einem *Headtracker*, also der Bewegungsverfolgung des Kopfes kombiniert, um einen Mauszeiger nutzen zu können.

Ähnlich wie bei *Sip-and-Puff* wird bei „*Tooth-Click*“ mit einem *Headtracker* gearbeitet. Jedoch wird hier nicht die Atmung, sondern die Vibration im Kiefer, die beim Zusammenklicken der Zähne entsteht, gemessen. Dadurch können verschiedene Befehle ausgelöst werden [3].

Aber der *Headtracker* funktioniert auch allein als Hilfsmittel, denn es können verschiedene Gesten des Kopfes durch befestigte Sensoren oder per Kamera erkannt werden [4]. So ist ein Nicken beispielsweise ein Mausklick oder Tastenanschlag, ein Neigen des Kopfes kann eine Seite umblättern.

Wird an Stelle des Kopfes die Blickrichtung einer Person ermittelt, nennt sich dies *Eye-Tracking*, oder Augensteuerung [5]. Durch eine Kamera wird Infrarotlicht ausgesendet, welches von den Augen reflektiert wird. Diese Reflexion wird wiederum von der Augensteuerung erkannt, dadurch kann gezielt eine Schaltfläche am Computer ausgewählt und durch ein Blinzeln aktiviert werden. Ein weiteres Hilfsmittel ist die Sprachsteuerung [6]. Vor einigen Jahren noch fiel es Geräten schwer, gesprochene Befehle korrekt zu interpretieren. Die Entwicklung ist heutzutage aber weit vorangeschritten und die Algorithmen lassen sogar individuelle, vorher aufgezeichnete Befehle zu.

Eine recht neue, unterstützende Methode sind *Gehirn-Computer Schnittstellen* (englisch: *Brain-Computer-Interface*, kurz *BCI*) [7]. Ein *BCI* misst Gehirnströme und leitet diese an Computerprogramme weiter, die beispielsweise Schaltflächen auslösen können. Es wird unterschieden zwischen invasiven und nicht-invasiven *BCI*. Bei invasiven *BCI* handelt es sich um Implantate, etwa von Firmen wie *Synchron* [8] oder *Neuralink* [9], während die andere Variante auf der Kopfhaut aufliegende, an Hauben befestigte Elektroden benutzt.

### 3. Munevo

Die bayerische Firma *Munevo GmbH*, bei der diese Bachelorarbeit geschrieben wurde, stellt Produkte zur Unterstützung benachteiligter Menschen her [10]. *Munevo Drive* ist ein Programm, mit dessen Hilfe über die Gestensteuerung der *Google Glasses* ein elektrischer Rollstuhl ohne die Benutzung der Hände gefahren werden kann. Dadurch kann Rollstuhlfahrer/innen neue Selbstständigkeit und Freiheit gegeben werden. Es können auch verschiedene Peripheriegeräte, wie ein am Rollstuhl befestigter Roboterarm, mithilfe der Gestensteuerung bedient werden. Ebenso ist es möglich, Komforteinstellungen wie Sitzhöhe oder Winkel der Rückenlehne Händefrei zu verändern. Zusätzlich bietet *Munevo* über die Bluetooth Verbindung der *Google Glasses* die Möglichkeit, Smartphones oder eine Computermaus mit Gestensteuerung zu bedienen.

Die Firma ist daran interessiert, andere Wege zu entwickeln, mit denen Menschen geholfen werden kann. Dazu gehört auch die Untersuchung von *BCI* und deren Potenzial.

## II. Gehirn-Computer-Schnittstelle (BCI)

### 1. Elektroenzephalografie

Wie bereits erwähnt, basiert die Technologie eines *BCI* auf der *Elektroenzephalographie*. Die *Elektroenzephalographie*, kurz *EEG*, ist eine in der Neurologie verwendete Prozedur, um Gehirnströme von Patienten zu messen und grafisch darzustellen [11]. Hierfür werden am Kopf des Patienten Elektroden angebracht, mit deren Hilfe elektrische Impulse im Gehirn gemessen werden. Diese Ströme entstehen durch die Kommunikation der Nervenzellen und lassen sich als Kurve darstellen, wobei deren Spitzen (angegeben in Volt) die Intensität anzeigen und die Frequenz (angegeben in Hertz) die Menge an Aktivität. Ein *EEG* wird mitunter verwendet, um Epilepsie oder Hirnschäden nachzuweisen, die Effektivität einer Narkose bei Operationen zu überwachen oder den Hirntod festzustellen.

Üblicherweise sind die genutzten Elektroden in einer Haube befestigt. Die Prozedur dauert meist zwischen 20 und 30 Minuten, während welchen die Hirnaktivität bei Aktionen wie dem Öffnen der Augen, heftigem Ein- und Ausatmen oder Reizeinfluss durch flackernde Lichter oder ähnlichem gemessen wird.

Das Ergebnis einer *Elektroenzephalografie* sind mehrere Wellenmuster, die jeweils Auskunft über Gehirnaktivität in bestimmten Hirnregionen geben. Hierbei wird unterschieden zwischen [12]:

- Alpha-Wellen (8 bis 13 Hz), die den sogenannten "Grundrhythmus", also die Gehirnaktivität bei wachem Zustand aber geschlossenen Augen beschreiben
- Beta-Wellen (14 bis 30 Hz), welche die Hirnaktivität bei geöffneten Augen, Sinnesreizen und geistiger Aktivität zeigen
- Gamma-Wellen (>30 Hz), die bei Lernprozessen oder erhöhter Aufmerksamkeit auftreten
- Theta-Wellen (4 bis 7 Hz), diese entstehen beispielsweise beim Einschlafen oder starker Müdigkeit
- Delta-Wellen (0,5 bis 3,5 Hz), welche langsam und meist synchron verlaufen und den Tiefschlaf signalisieren

Neben der Nutzung bei medizinischen Untersuchungen können die durch *EEG* gemessenen Gehirnströme auch als Auslöser für Computerprogramme genutzt werden. Hier setzt die Entwicklung von *BCI* an. Diese Geräte fungieren als Schnittstelle zwischen Gehirn und Computer. Oft sind sie für kommerzielle Zwecke entwickelt und werden dementsprechend immer günstiger. Ein *BCI* wird am Kopf der nutzenden Person befestigt, häufig in Form einer Kappe oder eines helmartigen Gebildes, an welchem Elektroden angebracht sind. Zusätzlich enthält ein solches Gerät eine Recheneinheit und eine Möglichkeit, Informationen an ein anderes Gerät zu übermitteln, sei es über USB oder Bluetooth.

## 2. *BCI* gegenüber anderen alternativen Eingabemethoden

Im Vergleich zu den bereits genannten Hilfsmethoden hat ein *BCI* einige Vorteile. Ein *Sip-and-Puff* System ist nicht besonders hygienisch und muss häufig gereinigt werden [3]. Dazu können Personen, die auf ein Atemgerät angewiesen sind, nur schwer oder gar nicht mit *Sip-and-Puff* arbeiten. Außerdem ist die Anzahl an verschiedenen Befehlen recht limitiert, da nicht viele Methoden des Ein- oder Ausatmens unterschieden werden können.

*Toothclick* hat ein ähnliches Problem, denn auch hier ist es schwer, mehr als eine geringe Anzahl an unterschiedlichen Arten des Zähneklickens zu unterscheiden [3]. Beide dieser Hilfsmittel benutzen außerdem oft einen *Headtracker*, um Maussteuerung zu ermöglichen. Dieser benötigt jedoch ein gewisses Maß an Feinmotorik der Kopf- und Halsmuskulatur, welches nicht für alle Personen gegeben ist. Wird ein *Headtracker* als einzelnes Hilfsmittel genutzt, ist die erforderliche Bewegungsfreiheit des Kopfes oft noch größer.

Ein *Eye-Tracker* benötigt teure Sensoren, darunter eine gute Kamera, die meist auf spezielle Art aufgebaut werden müssen, was den Transport schwierig macht [5]. Dazu kann Sonneneinstrahlung die Infrarotsensorik stören, weshalb das Nutzen außerhalb von Gebäuden nicht immer gut funktioniert.

Bei der Sprachsteuerung hingegen ist die nötige Lautstärke ein klarer Mangel. Einerseits aus Gründen der Privatsphäre, denn beispielsweise sollte eine Nachricht an den eigenen Arzt nicht laut vor Fremden diktiert werden, andererseits kann das Nutzen dieses Hilfsmittels auch die

umgebenden Mitmenschen stören, zum Beispiel wären laute Notizen während einer Vorlesung äußerst unangebracht. Zusätzlich haben selbst moderne Sprachsteuerungen immer noch Schwierigkeiten, Dialekte und Akzente zu verstehen, während manchen Menschen deutliches Sprechen aufgrund von Atemgeräten oder geringer Lungenkapazität schwerfällt.

Während sich invasive *BCI*-Systeme, die operativ in den Kopf gepflanzt werden, wie spannende Science-Fiction anhören, existieren Prototypen schon seit geraumer Zeit [8], [9]. Genau diese nötigen Eingriffe könnte für viele Menschen aber eine abschreckende Vorstellung sein, oder für Laien sogar nach Gedankenlesen oder Gehirnkontrolle klingen. Es wird sicherlich noch geraume Zeit dauern, bis sich ein solches Hilfsmittel durchsetzen kann. Nicht-invasive *BCI* hingegen bieten sich als geeignete Hilfe für Menschen mit starken Beeinträchtigungen an. Es ist nicht viel Bewegungsmöglichkeit nötig, um ein *BCI* effizient zu nutzen, womit auch Schwerbehinderte eigenständig Computer bedienen könnten. Jedoch kann ein solches Gerät auch, ähnlich dem *Headtracker*, als unterstützendes Mittel agieren. Mit einem *Hybrid-BCI* wird eine andere Eingabemethode durch ein *BCI* verbessert, indem mehr Befehle ermöglicht werden, oder auf Ermüdung und Fokusverlust geachtet werden kann. Natürlich ist ein *BCI* nicht frei von Nachteilen. So sind viele der derzeit erhältlichen Geräte ästhetisch nicht sehr ansprechend, was für einige Menschen ein Ausschlusskriterium sein kann. Dazu können dichte Haare die Schwierigkeit des Aufsetzens des Gerätes und die Genauigkeit bei der Nutzung stark beeinflussen. Vor allem ältere Vorschläge von tragbaren *EEG* beinhalten noch gelbeschmierte Elektroden, wobei neuere Modelle oft mit Trockenelektroden verbaut sind. Zusätzlich können auch moderne *BCI*, selbst mit heutigem Forschungsstand, langsam und unpräzise sein, denn ein Gehirn lässt sich nicht so einfach und gezielt steuern wie Muskeln. Sollte der Fokus versehentlich falsch gelegt sein, sei es durch Ablenkung oder Ermüdung, kann das Gerät dies nicht einfach von absichtlichem Fokussieren unterscheiden. Trotz dieser Nachteile scheint die Nutzung von *BCI* separat oder als Hybrid von Interesse.

### 3. Projektfindung

Von *Munevo* wurde ein *BCI* zur Verfügung gestellt, mit dem Auftrag, Projektideen für potenzielle Hilfsmittel zu finden. Um einen Einblick in den Alltag einer Person mit Behinderung und die dort auftretenden Schwierigkeiten zu erhalten, wurde mit einem Mitarbeiter, der an *spinaler Muskelatrophie* erkrankt ist, dessen Tagesablauf untersucht. *Spinale Muskelatrophie* ist eine progressive, neuromuskuläre Krankheit, bei der sich die Muskeln im Körper nicht weiterentwickeln und im schlimmsten Falle verkümmern [13]. Es wurde überlegt, welche Schwierigkeiten mit einer Eingabehilfe überwunden oder erleichtert werden könnten. Da die Problemfindung nur mit einer Person und an einem Krankheitsbild durchgeführt wurde, sind die Ergebnisse natürlich nicht repräsentativ für alle Menschen, die von starken Beeinträchtigungen betroffen sind. Dies war auch nicht das Ziel der Problemfindung, sollte aber dennoch erwähnt werden, um nicht den falschen Eindruck zu erwecken.

Ein häufig auftretendes Problem war das Betätigen von Lichtschaltern. Es gibt bereits viele *Smart-Home* Lösungen, die das Steuern von Heimelektronik ermöglichen [7]. Dafür ist oft Sprachsteuerung oder eine Aktivierung per Smartphone nötig. Wie bereits erwähnt ist es jedoch nicht für alle Menschen möglich, akustisch verstanden zu werden oder ein Smartphone eigenständig zu bedienen. Für diese Menschen könnte hier eine Hybrid-Lösung mit einem *BCI* in Erwägung gezogen werden. Mit einem aufgesetzten *Brain-Computer-Interface* wäre es möglich, den Fokus auf möglicherweise durch Formen und Farben markierte Objekte im Haushalt zu erkennen und diese dann mithilfe des *Smart-Home* Systems zu aktivieren. So könnte auch eine schwerbehinderte Person selbstständig das Licht ausschalten, den Fernseher anstellen oder Jalousie und Heizung bedienen.

Die Bedienung von Computer und Smartphone ist ebenfalls mit Schwierigkeiten verbunden, denn dafür ist es nötig, auf einer Tastatur schreiben, eine Maus bewegen oder auf einem Touchscreen tippen zu können. Es sollte möglich sein, diese Notwendigkeiten mit einer Steuerung durch *BCI* oder einer Hybridlösung gänzlich zu ersetzen oder zumindest zu erleichtern. Eine Bildschirmtastatur, die mittels Fokus bedient wird oder ein Programm, welches die Maus an den Punkt des visuellen Fokus' der nutzenden Person bringt sind vermutlich leichter umzusetzen als

eine Steuerung des Smartphones. Denn der Bildschirm moderner Handys ist im Vergleich zu Computerbildschirmen klein, was eine Ungenauigkeit bei der Bedienung zur Folge haben könnte. Eine Smartphone-Applikation zur Kommunikationsunterstützung könnte wiederum funktionieren. Verschiedene, vorher ausgesuchte Sätze würden sich mit einem *BCI* auswählen lassen, um dann per *Text-to-Speech* laut ausgesprochen zu werden. Die Auswahl wäre dabei auf das Individuum anzupassen, was sprachunfähigen Menschen bei der Kommunikation helfen könnte.

Schwierigkeiten bei der Steuerung eines elektrischen Rollstuhles hingegen sind aufgrund der bereits erwähnten Verzögerung bei der Interpretation von Befehlen vermutlich nicht direkt mit einem *BCI* zu umgehen. Doch bei weniger zeitkritischen Elementen, wie dem Verstellen des Sitzes oder der Fußraste, könnte ein *BCI* durchaus eine Alternative zu der herkömmlichen Bedienung per Knopfdruck sein, denn diese kann Rollstuhlfahrern schwerfallen. Besonders bei kaltem Wetter kann die Kraft in den Fingern verloren gehen.

Einige elektrische Rollstühle verfügen über angeschlossene Roboterarme, um schwache oder bewegungsunfähige Arme oder Hände auszugleichen. Auch hier könnte ein *BCI* hilfreich sein, vor allem bei der Auswahl von vordefinierten Armstellungen. Sicherlich ist auch eine direkte Kontrolle eines solchen Armes möglich, aber vermutlich schwieriger umzusetzen. Ähnliche Geräte, zum Beispiel ein Essensroboter, der auf Knopfdruck aus verschiedenen Schüsseln Essen an den Mund führt, könnten auch von einer Steuerung oder Steuerunterstützung durch ein *BCI* profitieren.

Die komplette Umsetzung einer dieser Projektideen ist jedoch vom Zeitaufwand und Arbeitsumfang nicht als Bachelorarbeit geeignet. Während der Projektfindung stellten sich jedoch einige Fragen zu dem zur Verfügung gestellten *BCI* – dem *NextMind*. Mitunter sollten Funktionsweise des Gerätes, Stärken und Schwächen und Genauigkeit und Geschwindigkeit untersucht werden. Das Beantworten dieser Fragen wurde Inhalt dieser Bachelorarbeit, während sich ein zweiter, bei *Munevo* arbeitender Student für seine Bachelorarbeit mit dem Programmieren eines Prototyps für eine *BCI*-gesteuerte Tastatur beschäftigte. Um die gestellten Fragen zu untersuchen, wird im nächsten Abschnitt zunächst das *NextMind* näher betrachtet.

### III. NextMind

#### 1. Beschreibung des NextMind

Das *NextMind* wird von einer gleichnamigen Firma aus Frankreich entwickelt. Ähnlich wie andere *BCIs* ist das *NextMind* ein tragbares, nicht-invasives Gerät mit eingebauten Elektroden zur Messung und Aufzeichnung von Spannungsschwankungen der Neuronen des menschlichen Gehirns [14]. Es unterscheidet sich von ähnlichen Geräten hauptsächlich dadurch, dass es nur am oberen Hinterkopf Messungen vornimmt - dem visuellen Cortex des Gehirns. Der visuelle Cortex ist für die Verarbeitung von visuellen Stimuli verantwortlich, also den Informationen, die das Auge an das Gehirn liefert. Dadurch ist das *NextMind* im Vergleich zu anderen *BCI* klein und kann mit einem Stirnband oder einer passenden Kappe am Kopf befestigt werden. Das Gerät wiegt ca. 60 Gramm und hat eine Akkulaufzeit von acht Stunden. Durch den Fokus auf visuelle Stimuli wird zwar Größe und Gewicht geringgehalten, jedoch gehen natürlich Messungen an anderen Stellen des Gehirns verloren, die bei anderen *BCI* ermittelt werden können.



Abbildung 1 - Das NextMind mit Stirnband



Abbildung 2 - Die neun Elektroden des NextMinds

Ein anderes, kommerziell erwerbliches Beispiel ist das *OpenBCI*, bei dem der Fokus aber auf der Wissenschaft liegt [15]. Hier können die Elektroden variabel befestigt werden, wodurch Messungen der Gehirnströme an ausgewählten Bereichen des Gehirns möglich sind. Im



Gegensatz zum *NextMind* sind hier auch alle erschlossenen Daten zugänglich, was die Nutzung für Laien deutlich schwieriger gestaltet. Das *NextMind* erlaubt keinen Einblick auf die Rohdaten der Elektroden, es wird ausschließlich als Eingabegerät genutzt. Es besitzt neun Elektroden zum Messen der Gehirnströme, welche nach dem Aufsetzen oder starkem Bewegen des Gerätes erneut kalibriert werden müssen. Die Kalibrierung dauert etwa 45 Sekunden und beinhaltet das Fokussieren eines grauen Kreises mit einem blinkenden, weißen Muster.

Ist das Gerät kalibriert, kann die nutzende Person ein Programm starten, welches mithilfe des *NextMind* gesteuert wird. In solchen Programmen wird mit standardmäßig hellgrauen Schaltflächen gearbeitet, über welche ein sogenannter "*NeuroTag*" gelegt wird. *NeuroTags* sind die bereits bei der Kalibrierung erwähnten, blinkenden, weißen Muster. Das *NextMind* hat bei der Kalibrierung die beim Fokus auf jene Muster entstehenden Gehirnsignale gespeichert. Nun kann mithilfe von maschinellem Lernen erkannt werden, wenn der Nutzer eine Schaltfläche, die mit solch einem *NeuroTag* belegt ist, anschaut und daraufhin ein Signal gesendet werden. Das Programm empfängt diese Signale und simuliert dann das Klicken oder Halten eines Mauszeigers auf der angeschauten Schaltfläche. Komplexere Funktionen als Klicken oder Halten sind zurzeit noch nicht möglich.



Abbildung 3 - Das NextMind in Benutzung (Marcel Glab)

## 2. SDK und Unity

Die Programme, die mit dem *NextMind* steuerbar sind, werden in der Spiele-Engine *Unity* programmiert, wodurch der Fokus der Hersteller des Gerätes auf Spiele und Unterhaltung klar wird [16]. Jedoch sind auch Projekte außerhalb dieser Domäne umsetzbar. Die primäre Programmiersprache für Skripte in *Unity* ist *C#*. Ein *Unity*-Programm wird visuell mehrmals in einer Sekunde dargestellt, oft mit einem Minimum von 30 Bildern pro Sekunde. Dadurch können Animationen flüssig abgespielt werden. Ein einzelnes Bild wird *Frame* genannt, die Anzahl an *Frames* pro Sekunde ist die *Framerate*. Zu jedem *Frame* wird überprüft, ob ein Skript ausgeführt werden muss. *Unity* ist in der Lage, Programme für Windows, Mac und auch Smartphones zu realisieren. Die Entwickler des *NextMind* haben ein *Software Development Kit (SDK)* zur Verfügung gestellt, mit welchem das Erstellen von Schaltflächen mit *NeuroTags* und das Hinzufügen von Funktionalität erleichtert wird. Ein sogenanntes *Sprite* ist eine optisch veränderbare Fläche, die von *Unity* bereitgestellt wird. In einer zweidimensionalen Umgebung sind die einfachsten *Sprites* Kreise oder Vierecke, im dreidimensionalen Raum dementsprechend Kugeln und Würfel. Diesem kann ein *NeuroTag* als Komponente hinzugefügt werden, wodurch das *Sprite* im späteren Programm per *NextMind* fokussiert werden kann. Nun wird dem *Sprite* ein Skript zugeordnet, welches bei Auslösen des *NeuroTags* ausgeführt wird. Bei einem Schachprogramm könnte das die Bewegung eines Bauern sein, bei einer *BCI*-gesteuerten Tastatur das Schreiben eines Buchstabens.

Das *SDK* erlaubt es die Größe des *NeuroTag*-Musters zu ändern, um es auf verschiedenen Bildschirmgrößen jeweils sichtbar zu machen. Es enthält auch Funktionen, um nach verfügbaren *NextMind*-Geräten zu suchen und sich mit ihnen zu verbinden oder die Verbindung zu trennen. Zusätzlich gibt es verschiedene Varianten das Auslösen eines *NeuroTags* zu werten, indem unterschiedliche *Events* verknüpft werden. Beispielsweise kann ein *Event* ausgelöst werden, wenn der *NeuroTag* gerade in diesem *Frame* aktiv geworden ist. Ebenso ist es möglich, ein *Event* durch kontinuierlichen Fokus bis zum Fokusverlust immer wieder zu aktivieren, oder das *Event* erst dann auszulösen, wenn die nutzende Person den Fokus auf den *NeuroTag* stoppt.

Die Anzahl der gleichzeitig aktivierten *NeuroTags* beschränkt sich auf zehn Stück. Das bedeutet, dass zwar mehr *Tags* existieren können, aber nur maximal zehn davon das blinkende Aktivierungsmuster zeigen können. Dies ist eine starke Einschränkung für die Entwicklung von Programmen. Vermutlich ist diese Limitation bedingt durch die Schwierigkeit, bei hoher Anzahl an *Tags* korrekt den Ausgewählten von den anderen zu unterscheiden.

### 3. Schwierigkeiten

Während des Projektes kristallisierten sich einige Probleme und Schwierigkeiten heraus. Im März des Jahres 2022 wurde das *NextMind* von der amerikanischen Firma *Snap Inc.* aufgekauft [17]. Vermutlich wurden aus diesem Grund alle Kommunikationsversuche ignoriert. Weder wurde auf E-Mails geantwortet, noch ließen sich die Mitarbeiter/innen von *NextMind* auf ein Gespräch über andere Wege, wie *Discord*, ein.

Zusätzlich ist die Dokumentation des *SDK* nicht besonders ausführlich, ebenso gibt es nur eine geringe Anzahl von Anleitungsvideos, die nicht weit in die Tiefe greifen. Auch von der Firma unabhängige Projekte sind schwer zu finden und eine Community scheint nicht zu existieren. Dies hat die Arbeit deutlich erschwert, denn es existierten keinerlei inoffizielle Bibliotheken zum Importieren oder Projekte, bei denen in dieser Arbeit auftretende Probleme bereits gelöst wurden. Zusammen führten diese Schwierigkeiten zu einem insgesamt höheren Arbeitsaufwand, da viele Probleme mittels wiederholten Ausprobierens in einer neuen Arbeitsumgebung gelöst werden mussten.

Die Beschränkung auf zehn *NeuroTags* erwies sich als größtes Hindernis. Für viele der vorgeschlagenen Projekte wären mehr als zehn nutzbare Schaltflächen zumindest von Vorteil, wenn nicht sogar nötig. Viele Benutzeroberflächen und unterstützende Programme haben eine hohe Anzahl an Bedienelementen, wodurch für einige der Schaltflächen mehrere Aktivierungen von Nöten sind. Dies kann lange Zeit in Anspruch nehmen, was die Benutzerfreundlichkeit leiden lässt. Um dieses Hindernis zu umgehen, wurden mehrere Ideen ausformuliert. Gomide et al. [18] haben Texte über die Benutzeroberfläche und Funktionsweise einer virtuellen Tastatur zusammengetragen und die dort gefundenen Ideen bewertet, um eine eigene Tastatur zu

entwickeln. Die dort genannten Ideen haben einige der nachfolgenden Beispiele inspiriert, auch wenn für andere Nutzung als eine Bildschirmtastatur abstrahiert wurde.

Mit einer 10-Kreuz-10 Matrix könnten 100 verschiedene Schaltflächen mit jeweils nur zwei Aktivierungen bedient werden, indem zuerst per *NeuroTag* eine Reihe, danach eine Spalte ausgewählt wird. Eine solche Matrix könnte jedoch viel Platz einnehmen und unübersichtlich sein.

Eine hierarchische Struktur könnte bis zu zehn Auswahlmöglichkeiten bieten, die jeweils zehn weitere Schaltflächen öffnen. Auf jeder Ebene könnten diese Schaltflächen entweder eine Funktion auslösen oder auf die nächste Ebene verweisen. So könnten die wichtigsten, oft benutzten Befehle mit einem oder zwei Schritten erreicht werden, während seltener gebrauchte Funktionen tiefer in der Hierarchie liegen. Visuell könnte diese Methode sehr unterschiedliche Formen annehmen, beispielsweise wie von Gomide et al. [18] beschrieben.

Möglicherweise könnte das Konzept einer hierarchischen Struktur umgekehrt werden und damit als Bedienhilfe für den kompletten Bildschirm dienen. Dafür würde jedes klickbare Objekt auf dem Bildschirm mit neun anderen, nahegelegenen Objekten zu Zehnergruppen zusammengefügt. Jeweils zehn dieser Gruppen könnten erneut zusammengefügt werden, bis weniger als zehn Gruppen entstehen. Diese würden per *NeuroTag* aktivierbar sein, bei der Aktivierung einer Gruppe werden die zehn Untergruppen aktiviert, bis die klickbaren Objekte erreichbar sind. Diese Methode könnte aber sehr viele Aktivierungen benötigen, bis ein Ergebnis erzielt wird, wobei schon die dritte Ebene 1000 verschiedene Auswahlmöglichkeiten besitzt.

Eine weitere Überlegung war, mehrere Gruppen von je zehn *NeuroTags* zyklusartig nur für Sekundenbruchteile zu aktivieren. Wird einer der *NeuroTags* fokussiert, erhöht dies einen Wert, der ab einer Schwelle die zugehörige Schaltfläche auslöst. In der Theorie sollte dadurch zwar keine Zeit gespart werden, aber es könnte auf Kontextwechsel verzichtet werden. Vermutlich würde diese Methode aber mehr Konzentration abverlangen, da das Aktivieren einer einzigen Schaltfläche längeres Fokussieren benötigt.

Weiterhin sollte untersucht werden, welche Größe mit *NeuroTags* versehenen Schaltflächen benötigen, um ausreichend gut zu funktionieren. Dafür wurden im folgenden Abschnitt Tests entworfen und durchgeführt.

## IV. Tests zum Optimieren der Einstellungen

### 1. Hintergrund

Laut den Entwicklern des *NextMinds* funktionieren *NeuroTags* schlechter auf kleinen Schaltflächen. Eine Bedienhilfe sollte jedoch nicht zu viel Platz auf dem Bildschirm einnehmen, da sonst andere anzuzeigende Programme nicht mehr oder kaum noch komfortabel nutzbar sind. Vor allem bei einem kleinen Bildschirm, beispielsweise dem eines Smartphones, sind kleine Schaltflächen von Vorteil, um mehr Funktionen einbauen zu können. Es ist also wichtig, die Größe der einzelnen Schaltflächen zu minimieren, um die Anwendung so klein wie möglich gestalten zu können, aber gleichzeitig müssen die *NeuroTags* zuverlässig aktivierbar sein. Außerdem muss darauf geachtet werden, dass eine Schaltfläche in kurzer Zeit aktiviert werden kann, da sonst eine frustrierende Benutzererfahrung entsteht.

### 2. Erster Test: Größe und Aktivierungszeit

Der erste Test wurde entworfen, um eine mögliche Korrelation zwischen der Größe von Schaltflächen und der nötigen Zeit, diese zu aktivieren, zu finden. Der Test beginnt mit einem weißen Programmfenster, in dem ein mit einem *NeuroTag* versehenes, graues Quadrat sichtbar ist. Dieses Viereck hat eine Größe von 512px mal 512px, im Graph (Abbildungen 6 bis 9) beschrieben als 200%. Wird der *NeuroTag* erfolgreich betätigt, verschwindet das Viereck und es erscheint ein neues Viereck an einer zufälligen Stelle im Fenster. Hierfür wurde eine *Unity*-eigene Methode für zufälliges kreieren von Spielobjekten genutzt, damit soll eine Programmumgebung simuliert werden, in der verschiedene Schaltflächen über den Bildschirm verteilt auftreten. Wird der *NeuroTag* des neuen Vierecks aktiviert, wiederholt sich die Prozedur, bis für die Aktivierung mehr als fünf Sekunden vergehen. Eine derart lange Aktivierungszeit lässt darauf schließen, dass das Viereck zu klein ist, um erfolgreich betätigt zu werden, denn die neu erscheinenden Vierecke sind immer eine genaue Kopie des vorherigen, mit Ausnahme der Größe. Nach zehn Quadraten ist das Elfte immer 20% kleiner als das Vorangehende. Die Zeit zwischen jeder Aktivierung wird

gemessen, um nach mehreren Durchläufen eine durchschnittliche Zeit für alle verschiedenen Größen von Vierecken bestimmen zu können. Um herauszufinden, ob sich unterschiedliche Entfernungen zum Bildschirm auf die Aktivierungszeit auswirken, wird der Test für Entfernungen von 50 Centimetern, einem, anderthalb und zwei Metern durchgeführt.

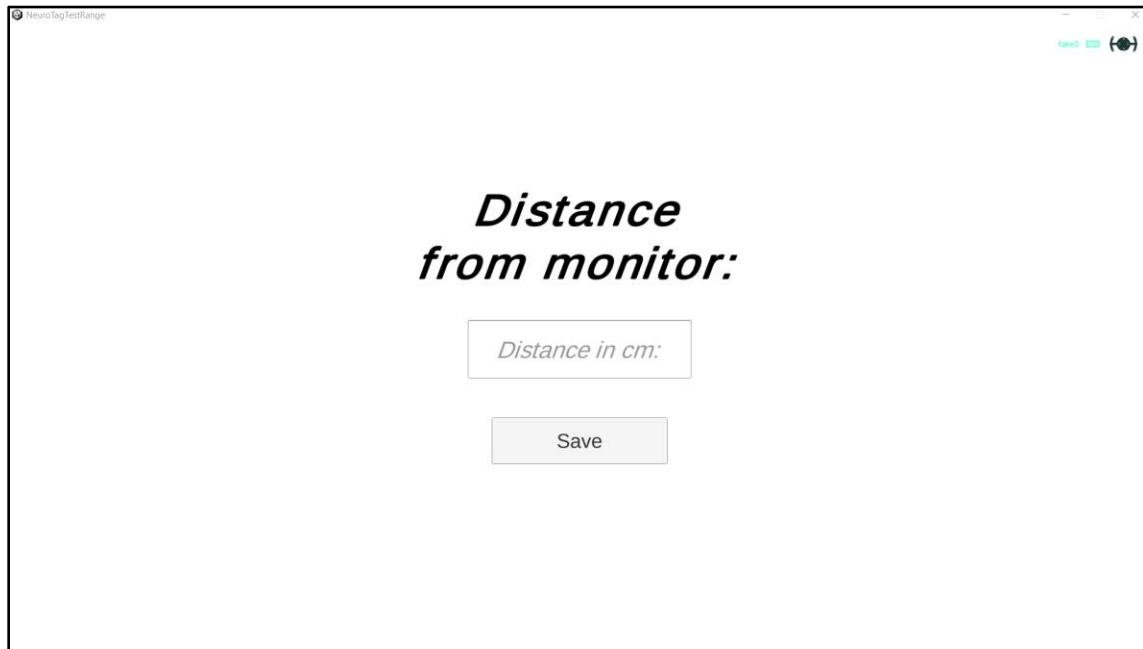


Abbildung 4 - Test zu Größe und Aktivierungszeit, Entfernungsangabe



Abbildung 5 - Test zu Größe und Aktivierungszeit, auszuwählendes Quadrat

Die Ergebnisse wurden in den Abbildungen 6 bis 9 dargestellt, jede Abbildung zeigt die Ergebnisse für eine der verschiedenen Entfernungen. Auf der Y-Achse ist die durchschnittliche Zeit in Millisekunden zu sehen, auf der X-Achse die unterschiedlichen Größenwerte der Quadrate. Letztere sind angegeben in Prozent, wobei 100% einer Größe von 256px mal 256px entspricht. Der Test wurde von einer Person durchgeführt. Die Ergebnisse wurden in Excel verarbeitet und als Graphen dargestellt.

### 3. Erster Test: Ergebnisse

Die Erwartung lag bei einem Anstieg der Aktivierungszeit bei Verkleinerung der Aktivierungsfläche. Zusätzlich wurde vermutet, dass eine steigende Entfernung zum Bildschirm auch eine längere Aktivierungszeit bedeutet. Grundlage dieser Vermutungen ist die Aussage der *NextMind*-Hersteller, dass kleinere Flächen schlechter für *NeuroTags* geeignet sind.

Entgegen den Erwartungen zeigt der erste Graph (Abbildung 6) ein anderes Ergebnis, denn die Aktivierungszeit scheint mit kleinerer Fläche sogar zu sinken, die Zeitwerte bei Größen unter 100% sind geringer als die Werte bei 100% aufwärts. Der maximale Zeitunterschied liegt bei 628 Millisekunden, zwischen 200% und 40% Schaltflächengröße.

Zusammen mit den zwei folgenden Graphen (Abbildungen 7 und 8) betrachtet, scheint die Aktivierungszeit aber eher willkürlich zu fallen und steigen. Die Werte bei einem Meter Bildschirmabstand (Abbildung 7) bewegen sich außerdem nur zwischen 3734ms (100%) und 3391ms (180%), einer Spannweite von 343ms. Der größte Zeitunterschied bei anderthalb Metern Abstand (Abbildung 8) ist mit 311ms ähnlich gering, lediglich Maximum (3971ms, 200%) und Minimum (3660, 80%) sind jeweils höher.

Tabelle 1 - Durchschnittswerte der Testergebnisse zur Korrelation von Größe und Aktivierungszeit

	50cm	100cm	150cm	200cm
200%	3954ms	3592ms	3971ms	3901ms
180%	3488ms	3391ms	3856ms	3782ms
160%	3634ms	3624ms	3851ms	3915ms
140%	3773ms	3505ms	3775ms	3910ms
120%	3536ms	3516ms	3872ms	N/A
100%	3547ms	3734ms	3863ms	N/A
80%	3446ms	3699ms	3660ms	N/A
60%	3351ms	3572ms	3800ms	N/A
40%	3326ms	3642ms	3936ms	N/A
20%	3409ms	3426ms	3793ms	N/A

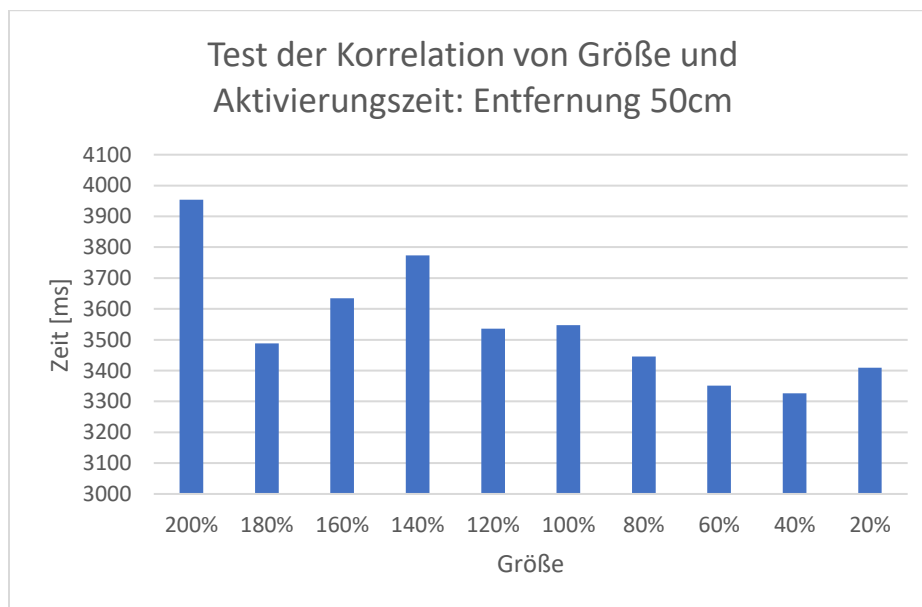


Abbildung 6 - Test zur Korrelation von Größe und Aktivierungszeit: 50cm Entfernung



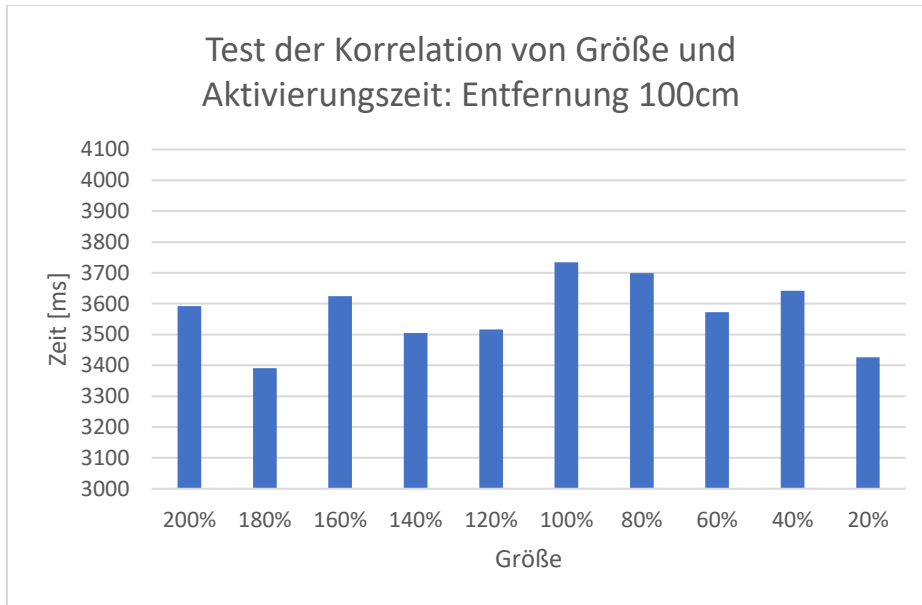


Abbildung 7 - Test zur Korrelation von Größe und Aktivierungszeit: 100cm Entfernung

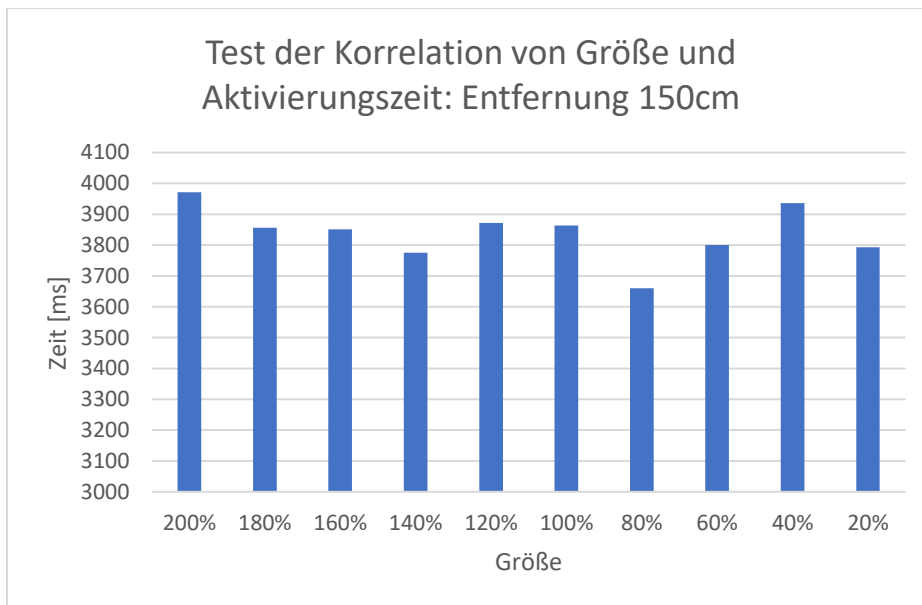


Abbildung 8 - Test zur Korrelation von Größe und Aktivierungszeit: 150cm Entfernung

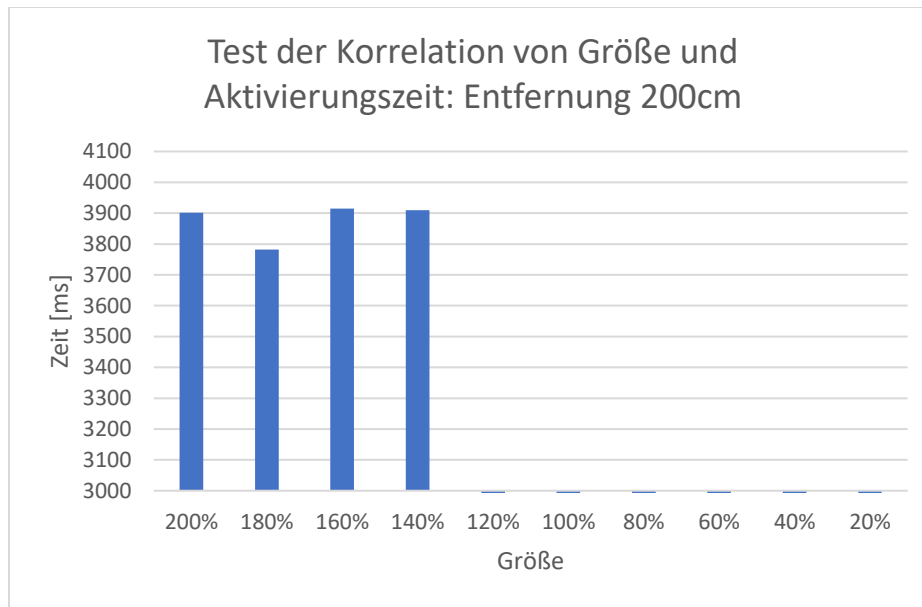


Abbildung 9 - Test zur Korrelation von Größe und Aktivierungszeit: 200cm Entfernung

Der vierte Graph (Abbildung 9) zeigt nicht nur eine höhere Aktivierungszeit von 3.8 bis 3.9 Sekunden, sondern enthält auch keine Werte ab einer Schaltflächengröße unter 140%. Es war nicht möglich, auf zwei Meter Entfernung kleinere Schaltflächen zu aktivieren.

Es scheint also, als hätte die Größe der aktivierbaren Schaltflächen hauptsächlich bei geringer Entfernung zum Bildschirm einen Einfluss auf die Aktivierungszeit, interessanterweise aber genau entgegengesetzt der Erwartung. Eine mögliche Erklärung ist die Anzahl der erscheinenden Flächen. Da keine anderen Quadrate außer dem zu Aktivierenden auftauchen, kann sich das Gehirn vielleicht leichter auf jene Form fokussieren, auch bei den kleinsten Quadraten. Zusätzlich kann gesagt werden, dass eine größere Entfernung zum Bildschirm negative Auswirkungen auf die Aktivierungszeit hat oder das Aktivieren von *NeuroTags* sogar beinahe unmöglich macht.

#### 4. Zweiter Test: Zuversichtlichkeitswert

*NeuroTags* besitzen einen „*Confidence Value*“ (deutsch: Zuversichtlichkeitswert). Dieser zeigt an, wie hoch die Wahrscheinlichkeit ist, dass das Gerät den jeweiligen *NeuroTag* korrekt als den gerade betrachteten identifiziert hat. Die Zuversichtlichkeit liegt zwischen den Werten „0“ und „1“, wobei eine „1“ die maximale Zuversichtlichkeit beschreibt. Standardmäßig wird das

*onTriggered* Event eines *NeuroTags* ausgelöst, wenn die Zuversichtlichkeit „1“ erreicht. Durch das Zugreifen auf den Zuversichtlichkeitswert kann eine Schaltfläche also auch schon bei geringerer Zuversichtlichkeit ausgelöst und so die Empfindlichkeit eines *NeuroTags* verändert werden. Natürlich kann sich hierbei die Gefahr erhöhen, einen falschen *NeuroTag* auszuwählen. Es wurde ein zweiter Test entwickelt, um herauszufinden, bei welchem Zuversichtlichkeitswert die Aktivierungszeit möglichst gering ist, während auf die Fehlerquote geachtet wird.

In diesem Test sind neun quadratische Schaltflächen in einer drei-Kreuz-drei Matrix angeordnet. Eine zufällig gewählte Schaltfläche wird grün umrandet und soll möglichst schnell per *NextMind* ausgelöst werden. Die Anzahl aller Erfolge wird in einer Variable gespeichert, ebenso die Zeit seit der letzten Aktivierung. Anschließend wird eine andere Schaltfläche per Zufall gewählt und die grüne Umrandung wechselt zum neuen Ziel. Wird eine der acht unmarkierten Schaltflächen versehentlich aktiviert, wird dies als Misserfolg in einer Variable gezählt. Es lassen sich die Größe der Schaltflächen sowie die *Confidence Value* anpassen, um verschiedene Konfigurationen testen zu können. Ein Testdurchlauf wurde mit derselben Größe und Confidence ausgeführt, wobei mindestens 50 Erfolge als Ziel gesetzt waren. Die Anordnung in einer Matrix sollte dafür dienen, eine Nutzeroberfläche mit mehreren, nah aneinander gelegenen Schaltflächen zu simulieren. Die Entfernung zum Bildschirm wurde möglichst gleich gehalten.

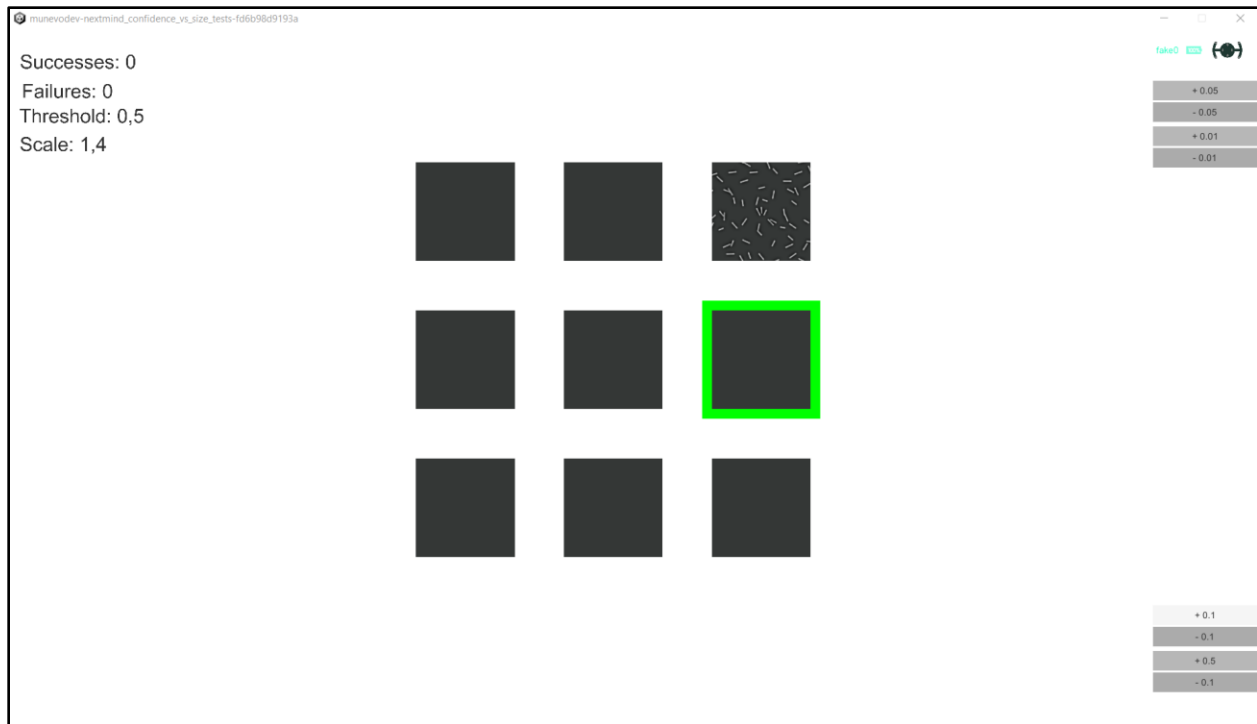


Abbildung 10 - Zuversichtlichkeitstest, Durchlauf mit durch Maus simulierter Eingabe

Die Zuversichtlichkeitswerte 0.9 (aufgerundet zu 1), 0.6, 0.5, 0.4 und 0.3 wurden je mit den Größen 200%, 150%, 100% und 50% getestet, wobei 100% 256 Pixel mal 256 Pixel entsprechen. Die Ergebnisse wurden in Abbildung 11, 12 und 13 dargestellt. In den Balkendiagrammen wird die durchschnittliche Zeit in Millisekunden auf der linken Y-Achse (blau) für die verschiedenen Größen und Zuversichtlichkeitswerte (X-Achse) beschrieben. Die rechte Y-Achse (rot) beschreibt die Genauigkeit, errechnet aus der Anzahl der Erfolge geteilt durch die Anzahl der Gesamtversuche in Prozent. Der Test wurde von zwei Personen durchgeführt. Die Ergebnisse wurden mit einem simplen *Python*-Skript in Graphen dargestellt.

## 5. Zweiter Test: Ergebnisse

Zunächst wurde der Test mit einer Maus simuliert. Dafür gibt es eine Einstellung, bereitgestellt durch das *SDK*, womit durch Klicken und Halten mit dem Mauszeiger ein *NeuroTag* aktiviert werden kann. Die so erstellten Werte (Abbildung 11) sollen als eine Vergleichsbasis für die Testergebnisse fungieren. Erwartet war genau das erhaltene Ergebnis, eine 100-prozentige Präzisionsrate bei sehr kurzer durchschnittlicher Aktivierungszeit. Die Größe der Vierecke ändert bei der Simulation durch Maus wenig bis gar nichts, da die klickbaren Flächen genug Platz für schnelles und präzises Klicken bieten.

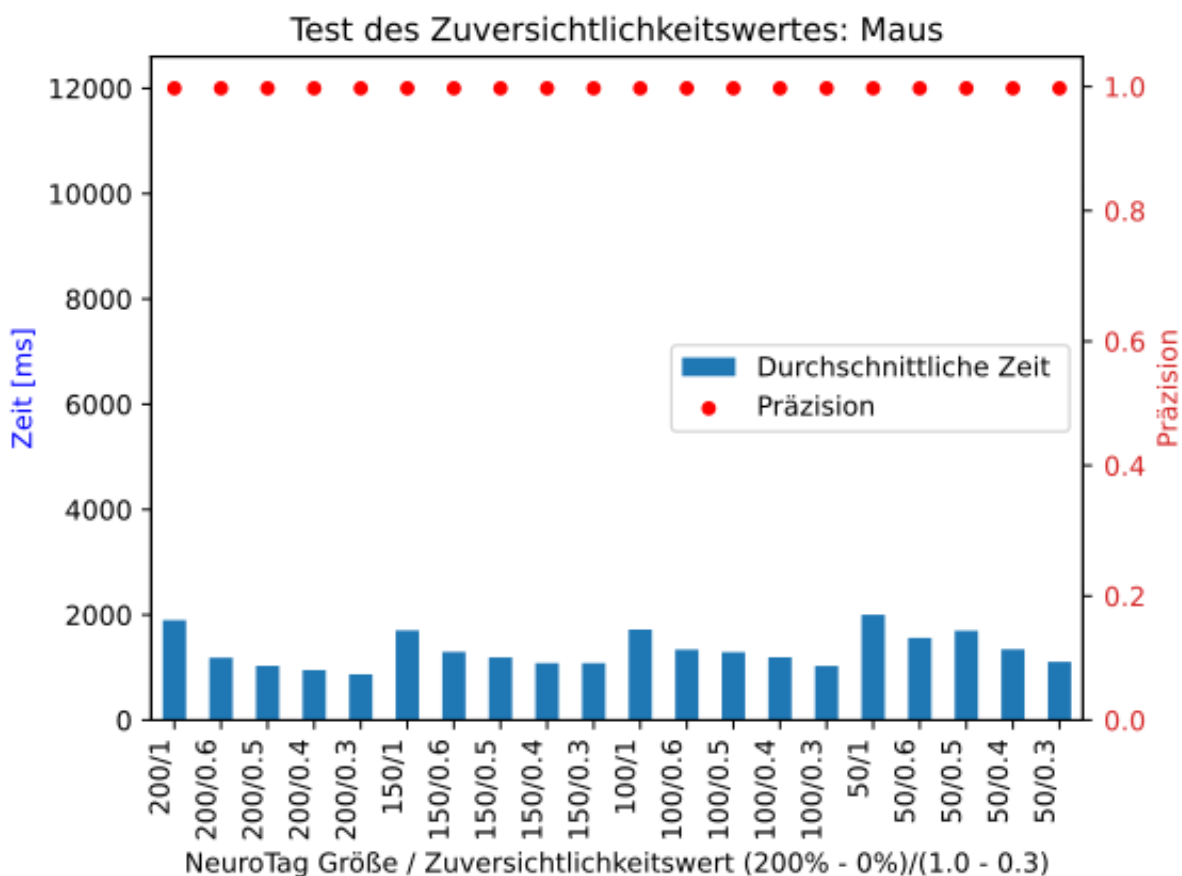


Abbildung 11 - Graph der Testergebnisse (Zuversichtlichkeitstest, Maus)

Für die tatsächlichen Tests war eine Zunahme der Zeit bei kleiner werdenden Schaltflächen aber innerhalb einer Größenkategorie eine Abnahme der Zeit korrespondierend mit dem Verringern des Zuversichtlichkeitswertes erwartet. Zudem wurde eine geringere Genauigkeit bei schnellerer Aktivierungszeit vorausgesehen. Die Annahme war, dass sich kleinere Flächen schlechter fokussieren lassen, geringere Zuversichtlichkeitswerte zu kürzerer Aktivierungszeit führen und beides die Präzision negativ beeinflusst.

Die Ergebnisse von Testperson 1 (Abbildung 12) zeigen einen geringeren Anstieg in der Aktivierungszeit bei kleineren Schaltflächen als erwartet. Allein die 50-prozentige Schaltflächengröße weist eine deutlich höhere Durchschnittsaktivierungszeit auf. Bei niedrigerem Zuversichtlichkeitswert ist die durchschnittliche Zeit in einer Größenkategorie jedoch nicht wie erwartet immer kürzer. Nur bei 50-prozentiger Größe sieht der Graph fast wie erwartet aus, die anderen Größenkategorien sind aber jeweils wesentlich ausgewogener als vorhergesehen. Die Präzisionswerte liegen bei höherem Zuversichtlichkeitswert wie erwartet näher an 100%, überraschend ist aber die Präzision bei 200/1 und 100/1 niedriger als die darauffolgenden Werte. Testperson 1 erzielte die besten durchschnittlichen Aktivierungszeiten bei einem Zuversichtlichkeitswert von 0.5, mit Ausnahme der Schaltflächengröße von 200%. Auch die Präzision war bei diesem Wert äußerst hoch.

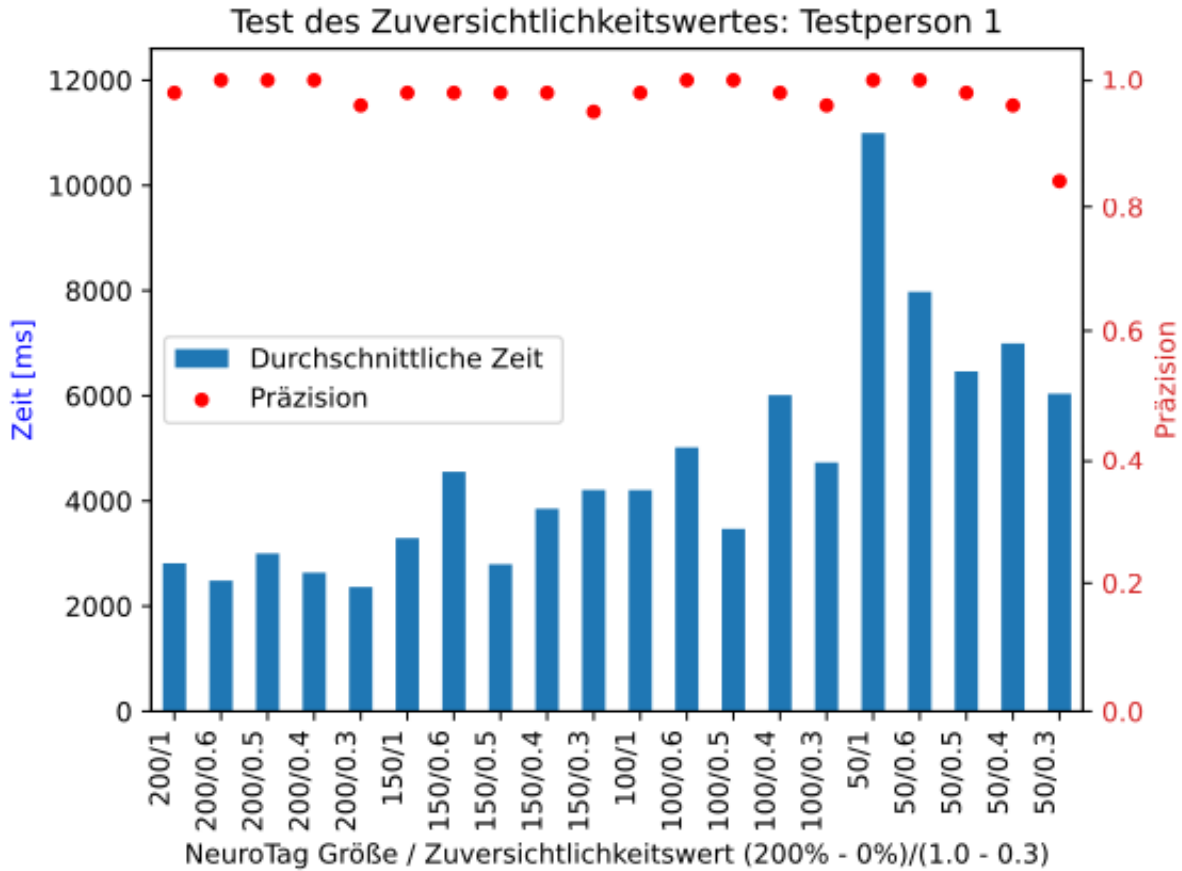


Abbildung 12 - Graph der Testergebnisse (Zuversichtlichkeitstest, Testperson 1)

Die Ergebnisse von Testperson 2 (Abbildung 13) weichen stärker von den Erwartungen ab. Nur die 50-prozentige Schaltflächengröße zeigt einen deutlichen Anstieg in der Aktivierungszeit, die anderen drei Größenkategorien sind in dieser Hinsicht jedoch beinahe gleich. Die Präzision nimmt zwar mit kleineren Flächen und niedrigerem Zuversichtlichkeitswert ab, jedoch sind die Präzisionswerte wesentlich mehr gestreut. Die größte Überraschung bringt aber die Aktivierungszeit innerhalb einer Größenkategorie. Statt wie erwartet abzunehmen, nimmt bei Testperson 2 die Zeit bei niedrigerem Zuversichtlichkeitswert zu (Größen 100% und 50%) oder bleibt beinahe stetig (Größen 200% und 150%). Anders als bei Testperson 1 scheint für Testperson 2 eine Zuversichtlichkeit von 0.4 die besten Ergebnisse erzielt zu haben. Die große Schwankung bei 50-prozentiger Größe könnten sich mit der deutlich niedrigeren Präzision erklären lassen. Durch kleine Schaltflächen könnte Testperson 2 öfter die falsche Fläche

fokussiert haben, was bei niedrigeren Zuversichtlichkeitswerten schneller zu Fehlern führt. Damit dauert das korrekte Aktivieren länger, denn jedes fehlerhafte Auslösen kostet Zeit.

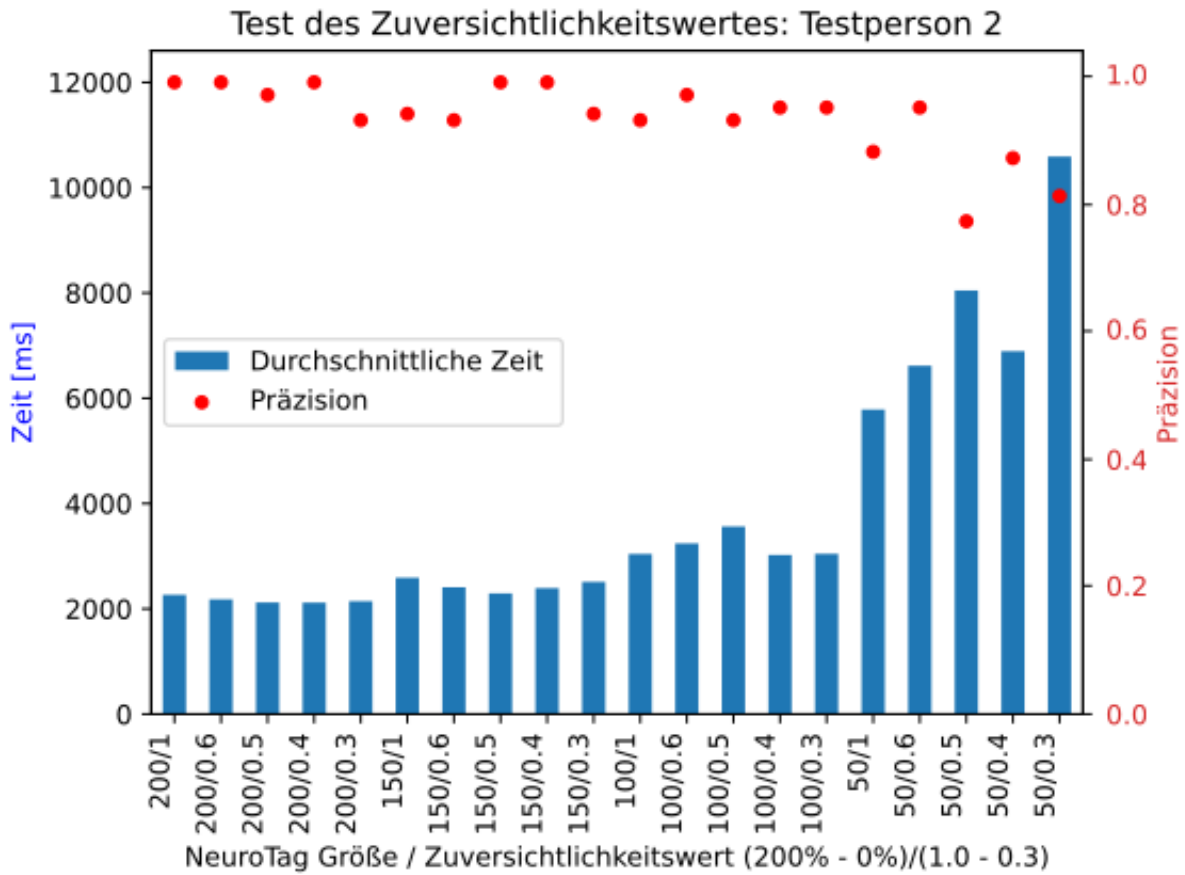


Abbildung 13 - Graph der Testergebnisse (Zuversichtlichkeitstest, Testperson 2)



## 6. Fazit

Diese Ergebnisse sind ausschließlich mit dem *NextMind* produziert worden und dementsprechend nicht repräsentativ für andere *BCI*. Es ist möglich, aber nicht garantiert, dass die Erkenntnisse bei der Nutzung anderer *BCI* helfen können. Insgesamt zeigen die Tests, dass die Größe der Schaltflächen vor allem dann eine Rolle spielt, wenn mehrere Schaltflächen dicht aneinander liegen, oder wenn die nutzende Person sich weit vom Bildschirm entfernt befindet. Die Präzision des *NextMind* ist dem zweiten Test zufolge in den meisten Fällen hoch. Nur bei den kleinsten Schaltflächen mit niedrigen Zuversichtlichkeitswerten wurden genug Fehler gemacht, um die Fehlerquote als problematisch anzusehen. Dementsprechend ist eine Schaltflächengröße von 256px mal 256px als Richtlinie zu empfehlen, zusammen mit einem Zuversichtlichkeitswert von 0.4 oder höher. Damit sollte die Fehlerquote möglichst geringgehalten werden.

Da die Aktivierungszeit selbst bei geringem Zuversichtlichkeitswert bei zwei Sekunden liegt, ist die Steuerung von zeitkritischen Programmen oder Geräten durch das *NextMind* gefährlich. Dazu kommt ein weiteres Problem: der Fokus. Mein Arbeitskollege und ich bemerkten beim Durchführen der Tests, dass das Fokussieren auf die Schaltflächen einen Großteil der Konzentration in Anspruch nimmt. Dementsprechend wäre es nicht ratsam, einen elektrischen Rollstuhl allein mit dem *NextMind* zu bedienen, denn eine Verzögerung der Bremse von mehreren Sekunden und Ablenkung von einer roten Ampel kann im Straßenverkehr fatal enden. Ebenso wäre es schwierig, eine Bedienoberfläche zu gestalten, die das gleichzeitige Fokussieren auf Schaltflächen und Fahrweg erlaubt. Zusätzlich könnten in einer Umgebung mit vielen visuellen Stimuli ähnliche Gehirnströme wie beim Fokussieren eines *NeuroTags* entstehen, und so versehentlich Funktionen aktiviert werden.

Das Programmieren von Bedienungshilfen, bei denen klare Befehle und Funktionen definiert sind, ist mit dem *NextMind* nach einer Einarbeitung in *Unity* und das *SDK* vermutlich recht einfach. Zum Beispiel ein SmartHome System, bei dem Licht, Musikanlage und Fernseher per visueller Schaltfläche am Computer mit dem *NextMind* an- oder ausgeschaltet werden können, wäre durchaus denkbar. Es ist aber wesentlich schwieriger, eine generelle Eingabehilfe „für alles“

zu entwickeln. Moderne Computer besitzen sehr viele klickbare Objekte, so dass es nicht realistisch ist, diese mit dem *BCI* in akzeptabler Zeitspanne aktivieren zu können.

Für eine Hybrid-Lösung sollte sich das *NextMind* hingegen gut eignen. Durch niedriges Gewicht und Größe stört das Gerät wenig bei der Nutzung und kann auch für längere Zeit getragen werden, um beispielsweise mit einem Headtracker zusammen die Bedienung eines Computers zu ermöglichen.

Neben Größe und Empfindlichkeit der *NeuroTags* gibt es einige andere Faktoren, die möglicherweise interessant für weitere Tests sein könnten. Der Einfluss von verschiedenen Farben auf die Effektivität der Schaltflächen machte sich während der Auswertung der Arbeit von Marcel Glab bemerkbar [19]. Dort schienen sich verschiedene Farben unterschiedlich gut für *NeuroTags* zu eignen. Ein Test, bei dem Farben auch mit unterschiedlicher Deckkraft gegeneinander untersucht werden, könnte interessante Ergebnisse hervorbringen. Ebenso wäre eine Untersuchung mit variierender Anzahl und verändertem Abstand zwischen Schaltflächen, die gleichzeitig im Blickfeld liegen womöglich nicht verkehrt.

## V. Ausblick in die Zukunft

Es stellt sich die Frage, welche Änderungen das *NextMind* oder ähnliche Geräte benötigen, um für ein breiteres Spektrum von Aufgaben nützlich zu sein. Der technische Fortschritt sollte für schnelleres und präziseres Erkennen von Gehirnströmen sorgen. Spezifisch das *NextMind* wird hoffentlich durch den Kauf von *Snap Inc.* weiter verbessert und kann dann, wenn das Gerät wieder auf den Markt gebracht wird, möglicherweise mehr als zehn *NeuroTags* zur gleichen Zeit unterstützen. Ein großer Fortschritt wäre eine Unabhängigkeit von der Nutzung eines Musters. Dadurch könnte es reichen, beispielsweise einen blauen Kreis als Auslöser anzuschauen, wodurch das *NextMind* auch ohne Bildschirm funktionieren würde.

Zusätzlich wären auch weitere Tests sehr hilfreich. Mehr Testdaten könnten die gefundenen Ergebnisse bestätigen. Auch vergleichende Versuche mit anderen *BCI*-Systemen dürften wichtige Erkenntnisse liefern, denn andere Geräte haben jeweils eigene Vor- und Nachteile gegenüber dem *NextMind*. So könnte sich herausstellen, dass ein anderes *BCI* besser geeignet ist, um als Hilfsmittel für medizinische Programme zu dienen.

## VI. Literaturverzeichnis

- [1] „Behinderte Menschen - Statistisches Bundesamt“.  
[https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/\\_inhalt.html](https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/_inhalt.html) (zugegriffen Sep. 23, 2022).
- [2] J. C. da Silva Junior und W. Germanovix, „Development of a Sip-and-Puff Interface for Communication and Control of Devices“, *IFMBE Proc*, Bd. 75, S. 1137–1146, 2020, doi: 10.1007/978-3-030-30648-9\_148/COVER.
- [3] T. Simpson, C. Broughton, M. J. A. Gauthier, und A. Prochazka, „Tooth-click control of a hands-free computer interface“, *IEEE Trans Biomed Eng*, Bd. 55, Nr. 8, S. 2050–2056, Aug. 2008, doi: 10.1109/TBME.2008.921161.
- [4] Y. Fu und T. S. Huang, „HMouse: Head tracking driven virtual computer mouse“, *Proceedings - IEEE Workshop on Applications of Computer Vision, WACV 2007*, 2007, doi: 10.1109/WACV.2007.29.
- [5] K. S. Arslan, „EYE TRACKING FOR INTERACTIVE ACCESSIBILITY: A USABILITY ANALYSIS OF COMMUNICATION INTERFACES“, 2020.
- [6] J. Zhu, X. Gao, Y. Yang, H. Li, Z. Ai, und X. Cui, „Developing a voice control system for ZigBee-based home automation networks“, *Proceedings - 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2010*, S. 737–741, 2010, doi: 10.1109/ICNIDC.2010.5657880.
- [7] R. Corralejo Palacios, R. Hornero Sánchez, D. Álvarez González, und L. Martín González, „Improving the accessibility at home: implementation of a domotic application using a p300-based brain computer interface system“, *Journal of Accessibility and Design for All*, Bd. 2, Nr. 1, S. 1–14, 2012, Zugegriffen: Okt. 05, 2022. [Online]. Available: <https://upcommons.upc.edu/handle/2099/12200>
- [8] „The Science | Synchron“. <https://synchron.com/medicine> (zugegriffen Okt. 06, 2022).
- [9] „Home - Neuralink“. <https://neuralink.com/> (zugegriffen Okt. 06, 2022).

- [10] „Geschichte der Kopfsteuerung munevo DRIVE - munevo“.  
<https://munevo.com/geschichte-der-kopfsteuerung-munevo-drive> (zugegriffen Okt. 05, 2022).
- [11] M. X. Cohen, „Where Does EEG Come From and What Does It Mean?“, *Trends Neurosci*, Bd. 40, Nr. 4, S. 208–218, Apr. 2017, doi: 10.1016/J.TINS.2017.02.004.
- [12] „Was passiert bei einer Elektroenzephalografie (EEG)?“  
<https://www.gesundheitsinformation.de/was-passiert-bei-einer-elektroenzephalografie-eeg.html> (zugegriffen Okt. 07, 2022).
- [13] M. R. Lunn und C. H. Wang, „Spinal muscular atrophy“, *The Lancet*, Bd. 371, Nr. 9630, S. 2120–2133, Juni 2008, doi: 10.1016/S0140-6736(08)60921-6.
- [14] „NextMind“. <https://www.next-mind.com/> (zugegriffen Okt. 07, 2022).
- [15] „OpenBCI | Home“. <https://openbci.com/> (zugegriffen Okt. 07, 2022).
- [16] „Unity Echtzeit-Entwicklungsplattform | 3D-, 2D-VR- & AR-Engine“. <https://unity.com/de> (zugegriffen Okt. 07, 2022).
- [17] „Welcome NextMind! - Snap Newsroom“. <https://newsroom.snap.com/de-DE/welcome-nextmind/> (zugegriffen Okt. 05, 2022).
- [18] R. de S. Gomide, L. F. B. Loja, R. P. Lemos, E. L. Flôres, F. R. Melo, und R. A. G. Teixeira, „A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques“, *Research on Biomedical Engineering*, Bd. 32, Nr. 2, S. 176–198, Mai 2016, doi: 10.1590/2446-4740.01715.
- [19] M. Glab, „Umsetzung und Vergleich verschiedener BCI gesteuerter Bildschirmstaturen“, Freie Universität Berlin.

## VII. Abbildungsverzeichnis

Abbildung 1 - Das NextMind mit Stirnband .....	16
Abbildung 2 - Die neun Elektroden des NextMinds.....	16
Abbildung 3 - Das NextMind in Benutzung (Marcel Glab).....	17
Abbildung 4 - Test zu Größe und Aktivierungszeit, Entfernungsangabe .....	22
Abbildung 5 - Test zu Größe und Aktivierungszeit, auszuwählendes Quadrat .....	22
Abbildung 6 - Test zur Korrelation von Größe und Aktivierungszeit: 50cm Entfernung .....	24
Abbildung 7 - Test zur Korrelation von Größe und Aktivierungszeit: 100cm Entfernung .....	25
Abbildung 8 - Test zur Korrelation von Größe und Aktivierungszeit: 150cm Entfernung .....	25
Abbildung 9 - Test zur Korrelation von Größe und Aktivierungszeit: 200cm Entfernung.....	26
Abbildung 10 - Zuversichtlichkeitstest, Durchlauf mit durch Maus simulierter Eingabe .....	28
Abbildung 11 - Graph der Testergebnisse (Zuversichtlichkeitstest, Maus).....	29
Abbildung 12 - Graph der Testergebnisse (Zuversichtlichkeitstest, Testperson 1).....	31
Abbildung 13 - Graph der Testergebnisse (Zuversichtlichkeitstest, Testperson 2).....	32

## VIII. Anhang

### Rohdaten – Test 1

#### 50cm Abstand – Zeit in ms

200%	180%	160%	140%	120%	100%	80%	60%	40%	20%
4005	3145	3693	4054	3078	3033	4071	3495	2864	3317
4012	3021	4214	3921	2911	4088	3151	3202	4200	2853
4151	4222	3571	3852	3876	4205	4079	2895	2975	3282
4068	3095	4011	3802	3809	3314	3336	3177	3086	3849
3987	3598	3508	3563	3463	3435	3454	3828	3935	3661
3834	3152	2951	4004	3643	3774	3392	3203	2886	4042
4246	3244	3790	3128	3175	2900	2820	3108	3181	3624
4123	3989	3552	3990	4057	3054	3342	3530	3097	3182
3261	3650	3520	4244	3452	3831	3940	3290	3964	3675
3875	4040	3303	3617	4107	3307	3375	3555	2952	3062
3932	3209	3863	3331	3329	4080	2949	3575	3447	2956
-----									
<b>3.954</b>	<b>3.488</b>	<b>3.634</b>	<b>3.773</b>	<b>3.536</b>	<b>3.547</b>	<b>3.446</b>	<b>3.351</b>	<b>3.326</b>	<b>3.409</b>

#### 100cm Abstand – Zeit in ms

200%	180%	160%	140%	120%	100%	80%	60%	40%	20%
4010	3416	3972	3063	3273	3546	3711	3440	3216	3287
3504	3630	3881	4195	3920	3657	3517	3610	3848	4240
3669	2923	2927	3349	2970	3406	3670	2820	3973	3256
4003	3118	3528	3265	2793	4063	4202	3782	3809	2873
3606	3448	3378	3374	3425	4117	3784	3507	3480	3210
3186	3728	4034	3580	3965	4226	3371	3930	3433	2941
3046	3135	3692	3995	4117	3696	2818	3842	3600	4002
3202	3404	2996	3282	2975	3612	4160	3289	4211	3340
3767	3269	3421	3553	3376	3393	4147	4041	3829	3661
3811	3703	4056	4106	3917	3248	3173	4033	2869	2830
3709	3527	3975	2792	3946	4111	4138	2994	3799	4048
-----									
<b>3.592</b>	<b>3.391</b>	<b>3.624</b>	<b>3.505</b>	<b>3.516</b>	<b>3.734</b>	<b>3.699</b>	<b>3.572</b>	<b>3.642</b>	<b>3.426</b>

150cm Abstand – Zeit in ms

200%	180%	160%	140%	120%	100%	80%	60%	40%	20%
3823	3819	3568	3801	3387	4118	3883	3755	4268	3753
4025	3484	3442	3764	3499	3829	3632	3921	3437	3883
4011	3884	3989	3459	4165	4234	3549	3643	4085	4102
4106	4110	3434	3815	3661	3521	3787	4004	3724	3402
4100	3911	4086	3749	3596	4152	3622	3519	3967	3638
4231	3706	4059	3514	4264	4207	3684	3851	3985	3536
4100	3828	4027	4028	3948	3796	3672	4142	3726	4077
3924	4060	3866	4033	4093	3516	3562	3509	4239	3849
3569	3871	4152	3498	3939	3430	3727	3976	3852	3633
3632	3549	3506	3998	3807	3625	3551	3991	4119	4140
4161	4195	4236	3868	4230	4069	3594	3484	3889	3711
-----									
<b>3.971</b>	<b>3.856</b>	<b>3.851</b>	<b>3.775</b>	<b>3.872</b>	<b>3.863</b>	<b>3.660</b>	<b>3.800</b>	<b>3.936</b>	<b>3.793</b>

200cm Abstand – Zeit in ms

200%	180%	160%	140%	120%	100%	80%	60%	40%	20%
3933	3457	4242	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3768	3996	3703	4042	N/A	N/A	N/A	N/A	N/A	N/A
3699	3984	3814	4258	N/A	N/A	N/A	N/A	N/A	N/A
3886	3609	4221	3966	N/A	N/A	N/A	N/A	N/A	N/A
3930	4135	4165	3908	N/A	N/A	N/A	N/A	N/A	N/A
3435	3589	3779	3951	N/A	N/A	N/A	N/A	N/A	N/A
3983	3810	3491	3933	N/A	N/A	N/A	N/A	N/A	N/A
4170	3555	3741	3785	N/A	N/A	N/A	N/A	N/A	N/A
4171	3510	3787	3676	N/A	N/A	N/A	N/A	N/A	N/A
3839	3997	3943	3564	N/A	N/A	N/A	N/A	N/A	N/A
4095	3965	4174	4016	N/A	N/A	N/A	N/A	N/A	N/A
-----									
<b>3.901</b>	<b>3.782</b>	<b>3.915</b>	<b>3.910</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>

Rohdaten – Test 2

Aufgrund der großen Datenmenge wurden diese als .txt im git-Repo hochgeladen.

Repository mit Code:

[https://git.imp.fu-berlin.de/kaiby/bachelorarbeit\\_test\\_neurotag\\_zuversichtlichkeit/-/tree/4fe831770a47421f517224c2016e280866d56e9c](https://git.imp.fu-berlin.de/kaiby/bachelorarbeit_test_neurotag_zuversichtlichkeit/-/tree/4fe831770a47421f517224c2016e280866d56e9c)

[https://git.imp.fu-berlin.de/kaiby/bachelorarbeit\\_test\\_nextmind\\_groesse/-/tree/ceb3cfe3a34ab5a52ceff00a9e4b854456739a2](https://git.imp.fu-berlin.de/kaiby/bachelorarbeit_test_nextmind_groesse/-/tree/ceb3cfe3a34ab5a52ceff00a9e4b854456739a2)