



---

# Limbic Chair Anwendung

---

Bachelorarbeit

Studiengang Informatik  
OST - Ostschweizer Fachhochschule  
Campus Rapperswil-Jona

Semester: Frühlingsemester 2024

**Autoren:** Nicolas Gattlen  
Kevin Pfister

**Betreuer/in:** Prof. Dr. Markus Stolze  
**Projektpartner/in:** Limbic Life AG  
**Experte/in:** Markus Flückiger  
**Gegenleser/in:** Prof. Mitra Purandare  
**Version:** 1.0  
**Datum:** 14.6.2024

# Inhaltsverzeichnis

<b>Abstract</b>	<b>iv</b>
<b>I Management Summary</b>	<b>1</b>
0.0.1 Problemstellung . . . . .	2
0.0.2 Lösungsansatz . . . . .	2
0.0.3 Resultate . . . . .	2
0.0.4 Fortsetzung der Arbeit . . . . .	2
<b>II Produkt Dokumentation</b>	<b>3</b>
<b>1 Einführung</b>	<b>4</b>
1.1 Überblick . . . . .	4
1.2 Limbic Chair . . . . .	4
<b>2 Recherche</b>	<b>6</b>
2.1 Wissenschaftliche Literatur . . . . .	6
2.1.1 Sitzlust statt Sitzfrust. Sitzen bei der Arbeit und anderswo . . . . .	6
2.1.2 The biomechanical benefits of active sitting . . . . .	8
2.1.3 Experiment . . . . .	10
<b>3 Voraussetzungen</b>	<b>13</b>
3.1 Bisherige Architektur . . . . .	13
3.2 Aufgabenstellung . . . . .	14
3.3 Use Case Beschreibung . . . . .	14
3.4 Akteure . . . . .	15
3.4.1 Hauptakteur: User . . . . .	15
3.4.2 Supportakteur: Limbic Chair . . . . .	15
3.5 Use Case Diagramm . . . . .	16
3.6 Nicht-funktionale Anforderungen . . . . .	17
<b>4 Architektur</b>	<b>19</b>
4.1 Aktivitätsverlauf . . . . .	19
4.2 Funktion der Software ActiveSit . . . . .	21
4.2.1 Frontend . . . . .	23
4.2.2 Dashboard . . . . .	25
4.2.3 Games . . . . .	28
4.2.4 Backend . . . . .	33
4.2.5 Datenbank . . . . .	35
4.2.6 Limbic Service . . . . .	36
4.2.7 Movementscore . . . . .	38
4.3 Zukunft der Architektur . . . . .	41
4.3.1 Limbic Service Forwarder . . . . .	41
4.3.2 MessagingQueue . . . . .	41

<b>5 Testing</b>	<b>42</b>
5.1 Usability Testing . . . . .	42
5.1.1 Game Testing . . . . .	42
5.1.2 Active Sit . . . . .	43
5.2 Testauswertung . . . . .	43
<b>6 Qualitätsmassnahmen</b>	<b>50</b>
6.1 Organisatorisches . . . . .	50
6.1.1 Definition of Done . . . . .	50
6.2 Tools zur Qualitätskontrolle . . . . .	50
6.2.1 Linter . . . . .	50
6.2.2 Code Coverage . . . . .	50
6.3 CI/CD . . . . .	51
6.3.1 Dokumentation Repository . . . . .	51
6.3.2 Sofware Repository . . . . .	51
6.4 Teststrategie . . . . .	51
6.4.1 Unit Testing . . . . .	51
6.4.2 End User Tests . . . . .	51
<b>III Projekt Dokumentation</b>	<b>52</b>
<b>7 Projektplanung</b>	<b>53</b>
7.1 Organisation und Ressourcen . . . . .	53
7.1.1 Zeitlicher Rahmen . . . . .	53
7.1.2 Kosten . . . . .	53
7.1.3 Personas . . . . .	53
7.1.4 Arbeitsweise . . . . .	53
7.1.5 Meetings . . . . .	53
7.2 Roadmap . . . . .	54
7.2.1 Phasen . . . . .	54
7.2.2 Meilensteine . . . . .	56
7.3 Issue Management . . . . .	57
7.3.1 Issues . . . . .	57
7.3.2 Components . . . . .	57
7.3.3 Jira Kanban Board . . . . .	57
7.4 Risikoanalyse . . . . .	58
7.4.1 Risikomatrix . . . . .	58
7.4.2 Risiken . . . . .	58
7.4.3 Risikominderung . . . . .	59
7.4.4 Risikoverlauf . . . . .	60
<b>8 Zeiterfassung</b>	<b>61</b>
8.1 TimeTracker . . . . .	61
8.2 Auswertung . . . . .	61
8.2.1 Auswertung Gesamtprojekt . . . . .	61
8.2.2 Auswertung pro Phase . . . . .	62
8.2.3 Auswertung pro Person . . . . .	63

<b>IV Appendix</b>	<b>64</b>
8.3 Testauswertung . . . . .	65
8.3.1 Testauswertung Experiment 1 . . . . .	65
8.3.2 Testauswertung Experiment 2 . . . . .	68
8.3.3 Testabdeckung Backend . . . . .	73
8.3.4 Testauswertung Notizen Experiment Zwei . . . . .	74
8.4 Wireframes . . . . .	75
8.4.1 Login Page . . . . .	75
8.4.2 Sign-Up Page . . . . .	75
8.4.3 Statistics Page . . . . .	76
8.4.4 Gaming Page . . . . .	76
8.5 Organisatorisches . . . . .	77
8.5.1 Originalen Aufgabenstellung . . . . .	77
8.5.2 Projektplanung . . . . .	80
<b>Tabellenverzeichnis</b>	<b>80</b>
<b>Abbildungsverzeichnis</b>	<b>81</b>
<b>Literaturverzeichnis</b>	<b>83</b>

# Abstract

**Einleitung:** Der Limbic Chair ist eine innovative Sitzlösung von Limbic Life. Er nutzt Sensorsortechnologie, um durch Bewegungstracking die Interaktion mit dem Benutzer zu verbessern. Diese Bachelorarbeit untersucht die Entwicklung eines fortschrittlichen Softwaresystems, das die Funktionalitäten des Limbic Chairs durch die Nutzung seiner integrierten Sensoren erweitert. Der Stuhl ist mit sechs Sensoren ausgestattet, die verschiedene Beinbewegungen überwachen und aufzeichnen. Insbesondere erfassen die linken und rechten PitchSensoren die vertikalen Bewegungen der Beine, während die linken und rechten Yaw-Sensoren die seitlichen Beinbewegungen verfolgen. Dariüber hinaus messen die Roll-Sensoren auf beiden Seiten die Rotationsbewegungen des Stuhls. Ziel dieses Projekts ist es, diese Sensoreingaben in eine Softwareplattform zu integrieren, die ein erweitertes Benutzererlebnis bietet und den Weg für innovative Anwendungen in Bereichen wie Gesundheit und Gaming ebnet. Der Entwicklungsprozess, die Systemarchitektur und die potenziellen Anwendungen des erweiterten Limbic Chairs werden in dieser Arbeit aufgezeigt.

**Vorgehen / Technologien:** In der heutigen Zeit sitzen viele Menschen aufgrund ihrer beruflichen und privaten Aktivitäten zu lange, was nachweislich negative Auswirkungen auf die Gesundheit hat, wie Rückenprobleme und schlechte Körperhaltung. Diese Bachelorarbeit zielt darauf ab, dieses Problem anzugehen und eine Lösung zu entwickeln, die zu mehr Bewegung im Sitzen anregt.

Eine umfassende Recherche und Analyse wurden durchgeführt, um die gesundheitlichen Auswirkungen von übermäßigem Sitzen zu untersuchen und bestehende Lösungen zur Bewegungsförderung am Arbeitsplatz zu analysieren. Basierend darauf wurde der Movement Score entwickelt, der verschiedene Parameter zur Berechnung der Bewegung des Nutzers heranzieht. Ein Algorithmus berechnet den Score kontinuierlich basierend auf den getrackten Bewegungen.

Zur Erhöhung der Attraktivität des intelligenten Stuhls wurden zwei interaktive Bewegungsspiele entwickelt, die den Stuhl als Controller nutzen. Im Raketen-Spiel steuert der Benutzer eine Rakete, während er im Jump-and-Run-Spiel einen Charakter durch eigene Bewegungen im Stuhl steuert. Diese Spiele sollen zu mehr Bewegung motivieren und gleichzeitig Spass bieten.

**Ergebnis:** Das Usability-Testing zeigte, dass Benutzer sich deutlich mehr bewegten, sobald sie sich des Movement Scores bewusst waren. Der Score stieg schnell an, da Benutzer motiviert waren, ihren Punktestand zu verbessern. Die Nutzung der Bewegungsspiele führte zu gemischten Ergebnissen: Einige spielten bis zu 15 Minuten, andere nur kurz.

Ein weiterer positiver Effekt war das gesteigerte Bewusstsein der Benutzer für ihre täglichen Aktivitäten. Viele versuchten, sich auch im Sitzen mehr zu bewegen. Bei Aufgaben mit hoher Konzentration wurde der Movement Score jedoch häufig ignoriert, während bei leichteren Tätigkeiten wie Meetings eher passive Bewegungen in den Alltag integriert wurden.

## Danksagung

Ein Grosses Dankeschön geht an folgende Institutionen, Programme und Personen:

- Prof. Dr. Markus Stolze für die ausgezeichnete Betreuung und Unterstützung während des Projekts. Er hat uns alle Fragen immer sofort beantwortet und war eine grosse Hilfe während dem Projekt.
- Patrik Künzler und Mark van Raai von der Firma Limbic Life AG für die Unterstützung und Hilfe während des Projekts.
- Alle Studierenden für das Testing unserer Software.
- Der Ostschweizer Fachhochschule für die Ausschreibung dieser Arbeit und die bereitstellung der Server.
- ChatGPT für das korrigieren unserer Texte.

# Teil I

# Management Summary

# Management Summary

## 0.0.1 Problemstellung

Unser Ziel war es, eine Software zu entwickeln, die die Bewegungen im Limbic Chair testet. Der Limbic Chair ist ein ergonomisch gestalteter Stuhl, der dem Benutzer mehr Mobilität als ein herkömmlicher Stuhl bietet. Neben der Mobilität verfügt der Stuhl auch über die Funktion, Daten zu erfassen und an ein externes System zu senden. Ziel des Projekts war es, einen Lösungsansatz zu finden, um die Benutzer des Stuhls zu mehr Bewegung zu animieren.

## 0.0.2 Lösungsansatz

Um die Bewegung des Benutzers zu fördern, wurde beschlossen, einen Algorithmus zu entwickeln, der die momentane Aktivität anhand eines Punktesystems darstellt. Dabei wird der Benutzer bei viel Bewegung mit Punkten belohnt, während bei Bewegungsmangel Punkte abgezogen werden. Zusätzlich wurden zwei Spiele entwickelt, die dem Benutzer die Möglichkeit geben, sich interaktiv zu bewegen. In diesen Spielen wird der Stuhl als Controller verwendet.

## 0.0.3 Resultate

Das Testing ergab, dass der Movementscore auf grosses Interesse stiess. Die Idee, Bewegung anhand von Punkten zu messen, fand viel Anklang, und die Benutzer zeigten einen starken Ehrgeiz, ihren Movementscore hoch zu halten. Die Spiele wurden ebenfalls genutzt, jedoch variierte die Begeisterung je nach Testperson. Nichtsdestotrotz war es uns möglich, die Bewegung der Benutzer durch unsere Software deutlich zu steigern.

## 0.0.4 Fortsetzung der Arbeit

Während der Arbeit stiessen wir auf das Problem, dass eine aktive Verbindung zum Stuhl erforderlich war. Bei einer Fortsetzung des Projekts wäre es sinnvoll, dieses Problem zu beheben. Ein weiteres Problem, das angegangen werden könnte, ist die Optimierung des Movementscores, da dieser bei intensiver Bewegung zu stark ansteigt. Es steht fest, dass diese Arbeit von uns nicht weiter fortgesetzt wird. Dennoch besteht die Möglichkeit, dass die Software in Zukunft erweitert werden kann.

## Teil II

# Produkt Dokumentation

# Kapitel 1: Einführung

## 1.1 Überblick

In dieser Arbeit wird das Sitzen untersucht und die Anwendungsmöglichkeiten des Limbic Chairs erforscht. Im ersten Kapitel erklären wir den Limbic Chair genauer, einschliesslich seiner Funktionalitäten und technischen Möglichkeiten. Im zweiten Kapitel werden wissenschaftliche Berichte zum Sitzen und dessen Status in der Gesellschaft untersucht, ebenso wie Berichte, die ergonomisch designete Stühle analysieren und herausfinden möchten, ob diese zu mehr Bewegung führen.

Im nächsten Kapitel wird basierend auf den Erkenntnissen der Recherche versucht, Voraussetzungen für eine Software zu schaffen, die Bewegung im Sitzen fördert. Auf dieser Grundlage wird die Software dann erstellt. Im vierten Kapitel wird die Architektur der Software besprochen. Wir zeigen, wie die fertige Software aussieht, und beschreiben alle wichtigen Komponenten näher.

In Kapitel 5 werden die Qualitätsmaßnahmen erläutert, wobei wir erklären, wie wir unsere Software strukturiert und organisiert haben. Im nächsten Kapitel geht es um die Projektplanung, in der wir die einzelnen Phasen unserer Arbeit detailliert beschreiben. Im letzten Kapitel wird die Zeiterfassung besprochen.

## 1.2 Limbic Chair

In diesem Kapitel geht es darum, eine kleine Einführung in den Limbic Chair zu geben. Dabei möchten wir die wichtigsten Funktionen des Stuhls erläutern. Dies ist grundlegend, um die Arbeit später gut zu verstehen.

Der Limbic Chair ist ein einzigartiger Stuhl, der sich durch sein ergonomisches Design und seine innovativen Funktionen deutlich von herkömmlichen Stühlen unterscheidet.



Abbildung 1.1: Limbic Chair - Quelle:[5]

Das ergonomische Design des Limbic Chair ermöglicht es dem Stuhl, sich den natürlichen Bewegungen des Körpers anzupassen und so eine dynamische Sitzhaltung zu fördern. Die umfassende Bewegungsfreiheit erlaubt es den Benutzern, sich in verschiedene Richtungen zu bewegen. Durch diese vielfältigen Funktionen ergeben sich zahlreiche Anwendungsmöglichkeiten.

Der Limbic Chair kann beispielsweise in Büroumgebungen eingesetzt werden, um stressbedingte Beschwerden zu verringern. In therapeutischen Settings unterstützt er Bewegungsübungen und kann in der Ergotherapie und Physiotherapie genutzt werden. Auch für den einfachen Heimgebrauch bietet der Stuhl Komfort und gesundheitliche Vorteile, insbesondere für Menschen, die viel Zeit im Sitzen verbringen.



Abbildung 1.2: Limbic Chair der sich der Bewegung anpasst - Quelle: [1]

Neben dem speziellen Design und den Bewegungsfreiheiten ist der Limbic Chair mit 6 Sensoren ausgestattet, die darauf abzielen, die Bewegungen des Nutzers zu erfassen und zu analysieren. Diese Sensoren senden ihre Daten an ein zentrales System zur weiteren Verarbeitung. Zu den Sensoren gehören die Pitch-Sensoren, die in Left Pitch und Right Pitch unterteilt sind und die vertikalen Bewegungen der Beine messen. Zusätzlich gibt es die Yaw-Sensoren, die die seitlichen Bewegungen der Beine verfolgen, sowie die Roll-Sensoren, die die Rotation des Stuhls erfassen. Die Daten der Sensoren werden im Verlauf der Arbeit näher beschrieben, da unsere Software auf den Daten des Mikrocontrollers basiert.

# Kapitel 2: Recherche

## 2.1 Wissenschaftliche Literatur

In diesem Kapitel werden wissenschaftliche Berichte über das Sitzen zusammengefasst und genauer erläutert. Um die Eigenschaften des Limbic Chairs zu verstehen, ist es wichtig die Auswirkungen des Sitzens zu verstehen. In einem ersten Bericht [4] wird erklärt, wie das Sitzen den Menschen beeinflusst und wie normales oder statisches Sitzen schlecht für den Körper ist. Ebenfalls wird beobachtet was dynamisches Sitzen genau ist und inwiefern es besser oder schlechter für den Körper ist. Im zweiten Bericht [6] werden Stühle untersucht, welche das dynamische Sitzen fördern sollen.

### 2.1.1 Sitzlust statt Sitzfrust. Sitzen bei der Arbeit und anderswo

**Einführung** Im Bericht "Sitzlust statt Sitzfrust. Sitzen bei der Arbeit und anderswo" [4] wird das Problem des Sitzens und die körperlichen Auswirkungen davon näher beschrieben. Es wird darüber berichtet, dass Sitzen eigentlich kein Problem für den Körper darstellt. Jedoch ist die Zeitspanne, welche viele Leute sitzend verbringen viel zu hoch und nicht mehr in Relation zum Stehen oder zur Bewegung generell. Denn der Mensch ist für die Bewegung geschaffen, ohne Bewegung leide nicht nur das Herz-Kreislauf System sondern auch der Bewegungs und Stützapparat. Außerdem stehen Krankheiten des Muskel-Skelett Systems an der Spitze der Arbeitsunfähigkeitstage-Statistik.

**Geschichte des Sitzens und Kultureller Einfluss** Damit verstanden werden kann wieso Menschen sitzen, ist es wichtig den Ursprung des Sitzens zu analysieren und dessen Geschichte. Denn die Entwicklung des Sitzens ist mit der menschlichen Evolution und der gesellschaftlichen Veränderung verknüpft. Ursprünglich galt der Mensch als Jäger und Sammler und bewegte sich ständig gehend oder liegend. Der Stuhl galt dabei als Symbol der Sesshaftigkeit und spielte erst viel später eine Rolle. Die Sesshaftigkeit veränderte dann schliesslich nicht nur das menschliche Verhalten, sondern auch die soziale und kulturelle Entwicklung wurde beeinflusst. Die Einführung des Stuhls steht symbolisch für den Übergang von einer nomadischen zu einer sesshaften Lebensweise, wobei der Stuhl anfänglich als Privileg und Machtattribut gesehen wurde, das vorrangig Herrschern und geistlichen Führern vorbehalten war.

Im Laufe der Zeit wurde das Sitzen zur Norm, die durch die industrielle Revolution und die Massenproduktion von Stühlen weiter gefördert wurde. Diese Entwicklung ermöglichte es allen Gesellschaftsschichten, Zugang zu Sitzmöglichkeiten zu erhalten, dadurch wurde das Sitzen zu einer weit verbreiteten und üblichen Tätigkeit. Kulturell gesehen, führte die Einführung des Sitzens zu einer Veränderung der Arbeits- und Lebensweisen. Im Mittelalter beispielsweise wurden Klöster zu Zentren des sitzenden Nachdenkens und Arbeitens, was den kulturellen Stellenwert des Sitzens weiter erhöhte.

In der modernen Gesellschaft hat sich das Sitzen zu einer dominanten Haltung entwickelt, die sowohl im beruflichen als auch im privaten Umfeld herrscht. Dies spiegelt sich in der Anzahl Stunden wider, die Menschen im Sitzen verbringen. Dies gilt für die Arbeit, den Transport und Freizeit. Trotz der negativen gesundheitlichen Auswirkungen, die mit dem langem Sitzen verbunden sind, bleibt diese Körperhaltung gesellschaftlich angesehen und symbolisiert oft Status sowie beruflichen Erfolg. Der kulturelle Einfluss des Sitzens zeigt sich auch darin, wie Arbeitsplätze gestaltet sind und wie soziale Interaktionen strukturiert werden. Der Stuhl, einst ein

einfaches Hilfsmittel, ist somit zu einem zentralen Element der menschlichen Kultur geworden, das nicht nur unsere physische, sondern auch unsere soziale und psychologische Welt prägt.

**Medizinische und biomechanische Aspekte des Sitzens** Im modernen Leben ist das Dauersitzen zum Alltag geworden. Diese Praxis, ist erst seit ca. 150 Jahren massenhaft auf der Nordhalbkugel verbreitet und auf der Südhalbkugel weniger üblich. Diese relativ kurze Zeitspanne liess dem menschlichen Körper kaum Zeit, sich an die Belastungen des Sitzens anzupassen, was zu einer hohen Zahl an Erkrankungen am Muskel-Skelett System führt. Die Statistiken zeigen, dass Beschwerden des Muskel-Skelett Systems führend in der Arbeitsunfähigkeitsstatistik sind, mit Rücken- und Nackenschmerzen als zweithäufigste Beschwerden mit fast 43 Prozent und Schmerzen im Nacken und Schulterbereich mit fast 46 Prozent. Jedoch lässt sich diese Statistik nicht nur auf das Sitzen zurückschliessen, weitere Gründe für diese körperlichen Beschwerden liegen unter anderem an körperlicher Fehlbelastung, ungünstige Körperhaltungen, häufiges und Tragen schwerer Lasten und rückenfeindliche Sportarten.

Die Wirbelsäule, bestehend aus 24 Einzelwirbeln und Bandscheiben, ist für statische, dynamische und schützende Funktionen des Menschen notwendig. Bandscheiben, die als Stoßdämpfer dienen, benötigen regelmässige Bewegung zur Nährstoffversorgung, ohne die sie ihre Elastizität verlieren. Sitzen beansprucht die Wirbelsäule mehr als stehen. So liegt der Druck der auf die Bandscheiben liegt bei 100 Prozent wenn die Person steht. Beim geraden Sitzen liegt sie bereits bei 140 Prozent und beim vorgebeugten Sitzen bei 190 Prozent. Bedeutet so viel dass beim Sitzen nicht nur die Bandscheiben ihre Elastizität durch fehlende Nährstoffversorgung verlieren, Sie werden beim Sitzen auch überbeansprucht. Langfristiges Sitzen in einer gebeugten oder krummen Haltung kann zu Rundrücken, Überdehnung des Bandapparates und Muskelverspannungen führen. Obwohl das aufrechte Sitzen biomechanisch günstiger ist, erfordert es statische Muskelarbeit, die schnell zu Ermüdung führt.

Das Sitzen bietet jedoch nicht nur Nachteile. So liegen die Vorteile des Sitzens in der verbesserten Rumpfstabilität, der Entlastung der Hüftgelenke und der Beine, einen niedrigeren Energieverbrauch und eine stabile Arbeitshaltung.

Die Nachteile des langen Sitzens überwiegen jedoch die Vorteile, da es zur Erschlaffung wichtiger Muskelgruppen führen kann, Fehlbelastungen der Wirbelsäule und Beeinträchtigung innerer Organe.

Um die gesundheitlichen Risiken zu vermindern ist es wichtig, dass man sich beim Sitzen in einem ständigen Haltungswechsel befindet und laufend Bewegung in den Alltag integriert. Im Bezug auf Arbeitsplätze bedeutet dies, dass ein Umdenken in der Gestaltung von Arbeitsplätzen erforderlich sein muss und den Mitarbeiter die negativen Risiken des langen Sitzens erklärt werden müssen.

**Physiologische Aspekte des Sitzens** Wie aus den vorherigen Abschnitte zu entnehmen ist bringt das Sitzen viele Nachteile mit sich. Da das lange Sitzen in vielen Jobs nicht mehr wegzudenken ist, muss versucht werden Alternativen zu schaffen, welche für ein möglichst geringen Gesundheitsrisiko sorgen.

Ein primärer Gedanke wäre, das man einfach richtig Sitzt. Jedoch gibt es ein "richtiges Sitzen" nicht. So wird im Bericht darüber gesprochen, dass die nächste Sitzhaltung immer die beste ist, da sich hier bewegt wird und bei anderen Sitzpositionen auch andere Muskeln beansprucht werden.

Ein weiterer Ansatz der verfolgt werden kann, sind Sitzhaltungen bei dem die Wirbelsäule am wenigsten belastet wird. Sie nimmt also ihre natürliche S-Form an. Dabei soll das Becken leicht nach vorn gekippt sein, um den Brustkorb aufzurichten und die Halswirbelsäule zu strecken,

was zu einer freien Atmung und einer Entlastung der inneren Organe ermöglicht. Um diese Position zu unterstützen, sind Stühle mit einer ausreichend hohen Rückenlehne, einer des Becken stabilisierenden Sitzfläche und Armlehnen für die Entlastung des Schulterbereichs essenziell. Ebenso wichtig ist die Einstellbarkeit der Sitzhöhe, um eine vollständige Auflage der Füsse zu gewährleisten.

Ein weiteres Problem des Sitzens ist die fehlende Bewegung. Durch die konstante Haltung einer bestimmten Sitzposition kommt es zu Verspannungen und schadet dem Körper. Um dieses Problem zu reduzieren ist es wichtig dynamisch zu sitzen. Das bedeutet, dass man trotz der Sitzhaltung in Bewegung bleibt. Regelmässige Veränderungen der Körperposition, unterstützt durch einen anpassungsfähigen Bürostuhl, fördern die Durchblutung, entlasten die Wirbelsäule und verringern das Risiko muskulärer Verspannungen. Wichtig dabei ist, das Sitzen durch Stehen und Bewegung zu unterbrechen, um ein Gleichgewicht zwischen den verschiedenen Haltungen herzustellen und gesundheitlichen Schäden präventiv entgegenzuwirken.

Empfehlungen für ein gesundes Sitzerlebnis und eine gesundheitsfördernde Arbeitsgestaltung umfassen neben der Nutzung ergonomischer Büromöbel auch organisatorische Massnahmen. Dabei sollen Gegenstände welche häufig genutzt werden sich nicht im direkten Greifraum befinden. Dies regt die Bewegung an und zwingt den Menschen zur aktiven Bewegung. Höhenverstellbare Schreibtische erlauben den Wechsel zwischen Sitzen und Stehen und tragen ebenfalls zu einer dynamischeren Arbeitsweise bei.

**Fazit** Die Recherche in diesem Artikel hat gezeigt, dass das Sitzen kulturell tief in unserer Gesellschaft verankert ist. Dabei schaden wir vor allem unserem Körper durch mangelnde Bewegung und belasten insbesondere unsere Wirbelsäule. Aus dem Problem, das dieser Artikel anspricht, muss nun eine Lösung gefunden werden. Ein Lösungsansatz, den man für diese Bachelorarbeit verfolgt, ist die Förderung der Bewegung während des Sitzens. Indem man die Bewegung während des Sitzens fördert, kann dies zu einer gesünderen Lebensweise und einer Entlastung der Folgen des langen Sitzens führen.

### 2.1.2 The biomechanical benefits of active sitting

**Einführung** Diese Studie [6] setzt sich mit dem dynamischen Sitzen auseinander und untersucht innovative ergonomische Lösungen, welche Förderung von Bewegung und die Verringerung von sitzbedingten Beschwerden fördern. Der Grund für diese Studie liegt darin, dass ein Grossteil der kanadischen Bevölkerung etwa 71 Prozent eines 8-Stunden-Arbeitstages in einer statischen Sitzposition verbringt. Dies erhöht das Risiko für muskuloskelettale und unterer Rückenschmerzen. Diese Beschwerden sind häufig ein gekrümmter Rücken oder auf eine nach vorne gelehnte Sitzhaltung zurückzuführen, die die lumbale Krümmung (unterer Rücken) beeinträchtigt und zu niedriger Muskelaktivität führt, wodurch die passive Gewebe um die Wirbelsäule belastet werden.

Als Reaktion auf diese Herausforderung wurden sitz-steh Schreibtische und aktive Stühle entwickelt, um die Bewegung am Arbeitsplatz zu fördern. Aktive Stühle, die in der Forschung untersucht wurden, umfassen ein mehrfach dimensionales Modell, das eine Stabilitätskugel simuliert, einen Sesselstuhl, der eine verbesserte Haltung unterstützt, und einen Stuhl mit geteilter Sitzfläche, der Bewegungen ähnlich dem Gehen ermöglicht. Diese Stühle zielen darauf ab, die statische Muskelaktivität und das empfundene Unbehagen zu reduzieren. Es wird angenommen, dass aktive Stühle im Vergleich zu traditionellen Bürostühlen und Steharbeitsplätzen eine grössere Ganzkörperbewegung und höhere Muskelaktivität in den unteren Extremitäten sowie im Bauchbereich fördern und somit zu weniger empfindenem Unbehagen im ganzen Körper führen.

Die Studie vergleicht die Sitzbiomechanik zwischen aktiven Sitzgelegenheiten, traditionellen Bürositzen und Steharbeitsplätzen und verfolgt die Hypothese, dass aktive Stühle zu einer grösseren Ganzkörperbewegung und geringerem empfundenen Unbehagen führen. Die Ergebnisse deuten darauf hin, dass aktive Sitzlösungen eine vielversprechende Möglichkeit darstellen, die negativen gesundheitlichen Auswirkungen von langem Sitzen am Arbeitsplatz zu mindern.

**Durchführung Experiment** In der Studie wurden 24 Universitätsstudenten (je zur Hälfte männlich und weiblich) rekrutiert, um die Auswirkungen von vier verschiedenen Arbeitsstationen auf die Körperhaltung, Muskelaktivität und das empfundene Unbehagen zu untersuchen. Die Arbeitsstationen umfassten zwei Arten von aktiven Stühlen (einen mit geteilter Sitzfläche und einen mit multiaxialem Design), einen traditionellen Bürostuhl und einen Steharbeitsplatz. Ziel war es, die Wirkung dieser Arbeitsstationen zu vergleichen und zu bewerten, inwieweit sie Bewegung fördern und die mit langem Sitzen verbundenen Gesundheitsrisiken mindern können.

Für die Datenerhebung wurden zwei Methoden angewandt. Zum einen wurden die Muskelaktivitäten mittels Elektromyographie (EMG) gemessen. Dabei werden die Spannungen der Muskeln gemessen und können dabei Muskelreize erkennen. Ebenfalls wurde ein Bewegungserfassungssystem zur Analyse der Körperhaltung verwendet. Diese Methode der Datenerhebung machten Sinn, da vor allem die aktiven Stühle speziell darauf ausgelegt waren, Bewegung zu ermöglichen und die Belastung der Wirbelsäule zu verringern. Dabei förderte einer der Stühle eine Bewegung ähnlich dem Gehen, während der andere die Instabilität eines Gymnastikballs simulierte.

Während des Experiments absolvierten die Teilnehmer zwei Testsessionen, in denen sie verschiedene Aufgaben (Tippen und Websurfen) auf den Arbeitsstationen durchführten. Ebenfalls wurden die Teilnehmer während des Testsessions aufgefordert sich zu bewegen um die Effektivität der aktiven Stühle zu bewerten. Die Erhebung umfasste auch maximale willkürliche Kontraktionen zur Normalisierung der EMG-Ergebnisse, Bewegungserfassung zur kinematischen Analyse sowie Fragebögen zur Erfassung des empfundenen Unbehagens und der Gesamterfahrung mit den Arbeitsstationen.

Die Analyse der gesammelten Daten erfolgte durch den Vergleich von Bewegungsmustern, neuromuskulärer Aktivität und Unbehaglichkeitsniveaus über die verschiedenen Arbeitsstationen hinweg.



**Figure 1.** Four workstations were investigated: the new active chair (AC1), the active multiaxial chair (AC2), the traditional office chair (Control), and the standing desk (Desk).

Abbildung 2.1: Stühle die für das Experiment gebraucht wurden - Quelle: [6]

**Diskussion der Ergebnisse** Die Studie verglich biomechanische Aspekte zweier aktiver Stühle, AC1 und AC2, mit einem Kontrollstuhl und einem Stehpult, um deren Einfluss auf die neuromuskuläre Aktivität, Gelenkwinkel und wahrgenommene Unbehaglichkeit zu bewerten. AC1 förderte durch eine geteilte Sitzfläche Pedal- und Vorwärtsrutschbewegungen, während AC2 eine multidirektionale Bewegungsfreiheit bot, die eine Stabilitätskugel simulierte. Beide aktiven Stühle zielten darauf ab, die Einschränkungen traditioneller Bürostühle zu überwinden, indem sie die Muskeln des unteren Körpers bzw. die Rumpfmuskulatur aktivierten und so die Sitzhaltung verbesserten.

Es wurde festgestellt, dass das Sitzen in AC2 im Vergleich zum Kontrollstuhl eine signifikant höhere neuromuskuläre Aktivität in bestimmten Muskeln erforderte, insbesondere bei und Vorfürück Bewegungen und Bewegungen von einer Seite zur anderen. Keine signifikanten Unterschiede wurden zwischen den vier Arbeitsstationen hinsichtlich der Aktivität der Oberschenkelmuskeln beobachtet, was auf die Effektivität der aktiven Stühle bei der Beinmuskelaktivierung hinweist.

In Bezug auf die Gelenkwinkel ergaben sich keine signifikanten Unterschiede in der Bewegung des Kopfes, der Schultern und der Beine zwischen den aktiven Stühlen und dem Kontrollstuhl, was darauf hindeutet, dass die Teilnehmer unabhängig von der Arbeitsstation ähnliche Körperhaltungen beibehielten. Interessanterweise führte die Nutzung der aktiven Stühle zu einer grösseren Bewegungsfreiheit im Beckenbereich, was auf eine potenzielle Reduzierung der Belastung der Wirbelsäule und eine Verbesserung der Sitzhaltung hinweist.

Die wahrgenommene Unbehaglichkeit variierte je nach Körperbereich und Arbeitsstation. Teilnehmer berichteten über mehr Unbehagen im unteren Beinbereich beim Stehen und eine höhere Unbehaglichkeit im Bereich der Gesäßmuskulatur beim Sitzen auf dem AC1 im Vergleich zu anderen Stationen. Diese Erkenntnisse legen nahe, dass aktive Stühle biomechanische Vorteile bieten können, aber auch Anpassungen erfordern könnten, um spezifische Beschwerden zu minimieren.

Zusammenfassend zeigt die Studie, dass aktive Stühle die neuromuskuläre Aktivität und die Bewegungsfreiheit verbessern können, was zu einer günstigeren Körperhaltung führt. Die Ergebnisse deuten darauf hin, dass langfristige Nutzung von aktiven Stühlen die wahrgenommene Unbehaglichkeit im Vergleich zu traditionellen Bürostühlen reduzieren könnte, obwohl weitere Anpassungen zur Minimierung von Beschwerden in bestimmten Körperbereichen erforderlich sind.

**Fazit** Die Studie zeigt, dass aktive Stühle durchaus einen Einfluss auf aktives Sitzen haben können. So kann durch erhöhte Bewegung die Belastung auf die Wirbelsäule verringert werden und die Bandscheiben können entlastet werden. Im Bezug auf unsere Arbeit zeigt uns dieser Bericht, dass aktive Bewegung mittels dem richtigen Stuhl durchaus gefördert werden kann.

### 2.1.3 Experiment

Im Experiment wollen wir herausfinden, wie stark der Einfluss des Limbic Chairs auf aktives Sitzen ist. Dazu haben wir Testpersonen gebeten, zwei Stunden auf dem Stuhl zu sitzen. Während dieser Zeit konnten die Personen normal weiterarbeiten, mussten aber auf dem Stuhl sitzen bleiben. Um die Aktivität der Personen zu messen, haben wir einen Algorithmus entwickelt, der den aktuellen Aktivitätsstand widerspiegeln soll. Dieser Algorithmus berechnet dann den entsprechenden Movementscore.

Wie der Movementscore genau funktioniert, wird im Kapitel “Architektur“ genauer beschrieben. Für das Experiment ist es wichtig zu wissen, dass der Movement-Score steigt, wenn sich die Testpersonen viel bewegen, und bei Inaktivität sinkt. Damit soll gezeigt werden, zu welchen Zeiten sich die Testpersonen bewegen und ob sie sich während der gesamten Zeit überhaupt bewegen.

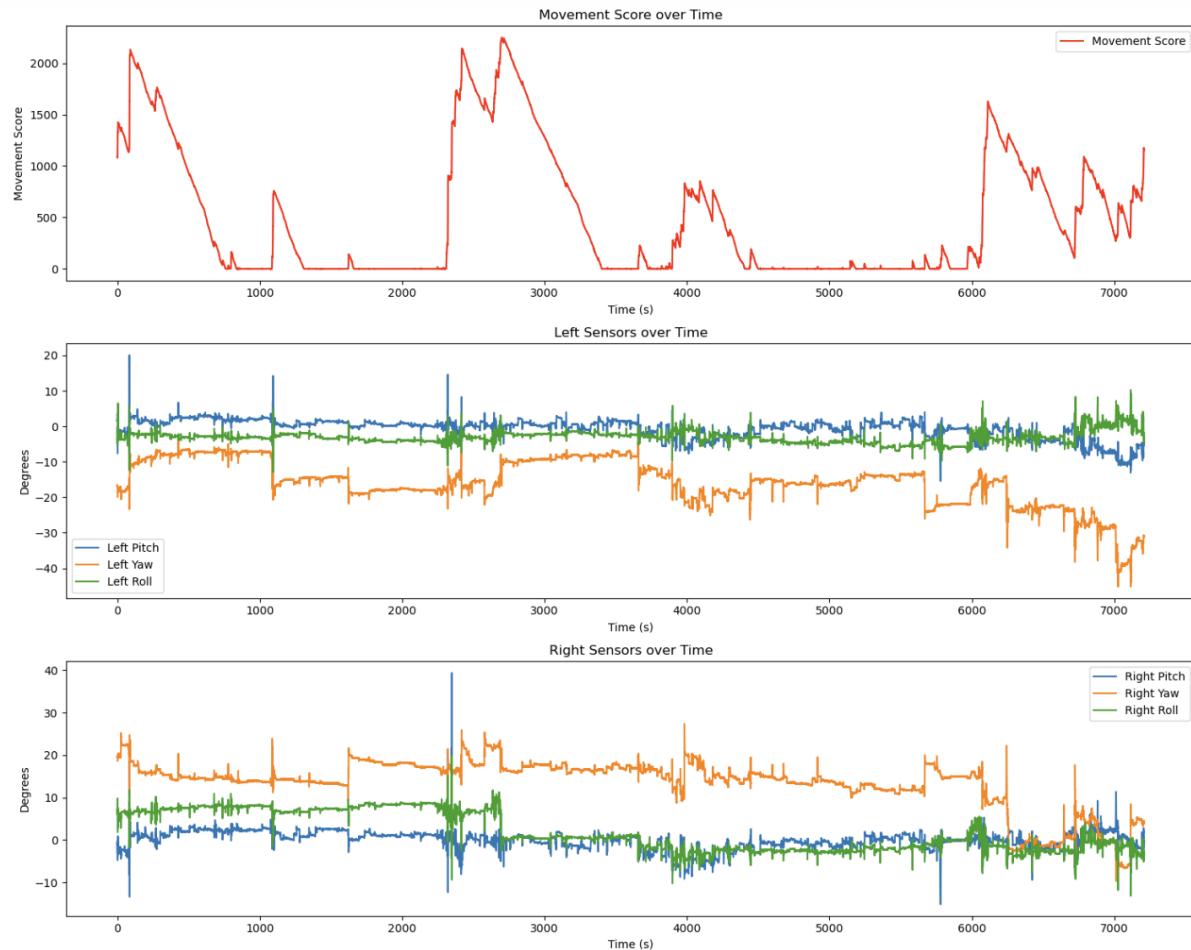


Abbildung 2.2: Auswertung Experiment

In der ersten Grafik ist die Auswertung des Movementscores über die Zeit zu sehen. In der zweiten Grafik zeigen wir die Daten, die mit dem linken Beinhalter erfasst wurden, und in der dritten Grafik jene, die mit dem rechten Beinhalter erfasst wurden. Da die Daten aller Personen ungefähr das Gleiche anzeigen, wollen wir eine Auswertung genauer betrachten. Die Daten anderer Personen sind im Anhang wiederzufinden.

Wenn man die zweite und die dritte Grafik genauer anschaut, sieht man immer wieder kleine Ausschläge nach oben oder nach unten. Diese signalisieren eine Bewegung der Person. Am Anfang der Grafik erkennt man, dass vor allem der Left Pitch nach oben geht. Das bedeutet, dass zu diesem Zeitpunkt eine Bewegung stattgefunden hat, bei der sich das linke Bein vertikal bewegt hat. Diese Bewegung führt dazu, dass sich das rechte Bein automatisch auch bewegt, was zu einem Anstieg im Movementscore führt.

Wenn es keine Ausschläge der Sensoren gibt, bleiben diese ungefähr auf demselben Level. Dies lässt sich auch am Movementscore erkennen, weil dieser entweder sinkt oder auf dem Wert 0

bleibt. Die Movementscore-Grafik zeigt, dass es immer wieder zu Bewegungen kam. Die Bewegungen beinhalten meistens einen der Pitch-Werte. Da eine solche Bewegung nur schwer mit einem normalen Stuhl möglich ist, lässt sich daraus ableiten, dass der Stuhl eine Mobilität bietet, die ein normaler Stuhl nicht ermöglicht.

Allerdings lässt sich auch erkennen, dass der Movementscore oft sinkt und über längere Zeit einen Wert von null annimmt. Eine Vermutung, die sich daraus ableiten lässt, ist die Nichtnutzung der gegebenen Mobilität. Die Daten scheinen zu zeigen, dass zwar Bewegungen stattfinden, die in einem normalen Stuhl nicht möglich sind und dadurch ein aktiveres Sitzen bestätigen. Dennoch ist der Movementscore oft im Nullbereich, was darauf hindeutet, dass die Bewegung nur sehr schwach ist und man keinen grossen Vorteil aus dem Stuhl zieht, wenn man nur die Bewegung im Stuhl betrachtet.

# Kapitel 3: Voraussetzungen

## 3.1 Bisherige Architektur

Da die Verbindung unserer Software auf einer bereits bestehenden Software basiert, muss dies bei der Entwicklung unserer Software berücksichtigt werden. Die bisherige Architektur erfordert, dass sich der User entweder über ein Kabel oder drahtlos mit dem Computer verbündet. In unserem Fall hat die drahtlose Variante nie funktioniert, daher haben wir immer auf eine kabelgebundene Verbindung gesetzt.

Nachdem das Kabel mit dem Computer verbunden wurde, musste eine von Limbic Life entwickelte Software namens Limbic Control Service gestartet werden. Mit dieser Software konnte man sich dann mit dem Stuhl verbinden, und die Daten des Stuhls wurden direkt an die Software weitergeleitet.

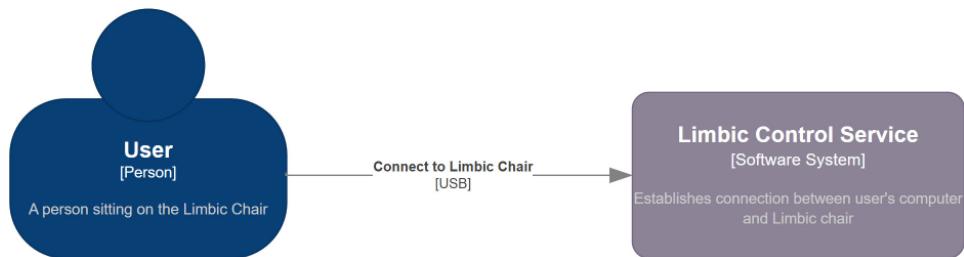


Abbildung 3.1: C4 Diagram Layer 1 - Context Level without ActiveSit

Für unsere Software bedeutete dies, dass wir unsere Daten aus dem Limbic Control Service extrahieren mussten. Dies sorgte jedoch für Probleme, da der Limbic Control Service nur eine Verbindung aufbauen kann, unsere Software jedoch mindestens zwei Verbindungen benötigt. Wie dieses Problem gelöst wurde, wird im Abschnitt Limbic Service beschrieben.

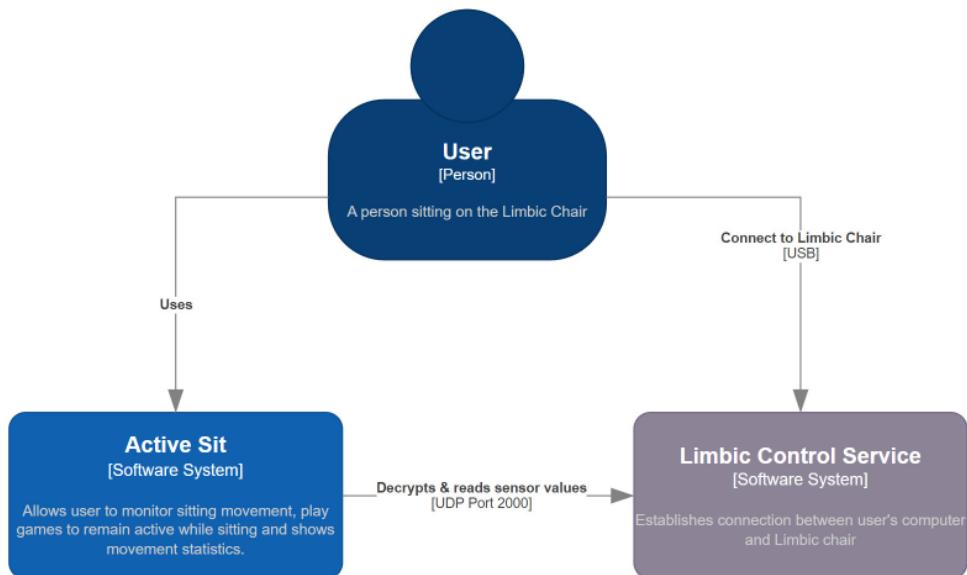


Abbildung 3.2: C4 Diagram Layer 1 - Context Level with ActiveSit

## 3.2 Aufgabenstellung

Die folgende Beschreibung basiert auf Informationen, die aus der originalen Aufgabenstellung entnommen wurden:

Aus der Recherche ist zu entnehmen, dass viele Menschen zu viel Zeit sitzend verbringen. Daher ist es entscheidend, eine gesunde Sitzhaltung zu fördern. Der Limbic Chair bietet die Möglichkeit, gesund zu sitzen und gleichzeitig freie Bewegungen zu ermöglichen, die mit einem normalen Stuhl nicht machbar wären. Obwohl der Limbic Chair bereits einen Fortschritt darstellt, besteht immer noch das Risiko, dass Benutzer vergessen, sich regelmässig zu bewegen.

Nun soll eine Windows-Applikation entwickelt werden, die die Sitzhaltung des Benutzers in Echtzeit überwacht. Zugleich soll die Applikation den Benutzer mit einer Popup-Nachricht regelmässig daran erinnern, sich zu bewegen und ihn auffordern, ein Spiel zu spielen. In diesem Szenario fungiert der Limbic Chair als Controller und kann vom Benutzer zur Steuerung des Spiels genutzt werden.

Da der Benutzer aufgefordert wird, ein Spiel zu spielen, ist es auch möglich, Statistiken anzuzeigen, wie zum Beispiel einen Movementscore, der mittels eines Algorithmus die aktuelle Bewegungsaktivität anzeigt. Dies gibt dem Benutzer eine Übersicht darüber, wie aktiv er über den ganzen Tag hinweg war.

Diese Applikation würde nicht nur das Wohlbefinden und die Gesundheit der Benutzer verbessern, sondern auch das volle Potenzial des Limbic Chairs ausschöpfen.

## 3.3 Use Case Beschreibung

In der heutigen Zeit verbringen Menschen einen Grossteil ihres Tages sitzend, insbesondere auf Stühlen. Diese weit verbreitete Sitzhaltung kann langfristig negative Auswirkungen auf den Rücken und die allgemeine Körperhaltung haben. Unser Körper ist evolutionär nicht darauf ausgelegt, lange Zeit in sitzender Position zu verweilen. Diese Praxis hat sich jedoch mit der Industrialisierung verfestigt und ist heute alltäglich. Um diesem Problem entgegenzuwirken, möchten wir die Menschen dazu animieren, passive Bewegungen in ihren Alltag zu integrieren – und genau hier kommt der Limbic Chair ins Spiel. [4]

Der Limbic Chair ist mit sechs Sensoren ausgestattet, die darauf abzielen, die Bewegungen des Nutzers zu erfassen und zu analysieren. Diese Sensoren senden ihre Daten an ein zentrales System zur weiteren Verarbeitung. Zu den wichtigsten Sensoren gehören die Pitch-Sensoren, die in Left Pitch und Right Pitch unterteilt sind und die vertikalen Bewegungen der Beine messen. Zusätzlich gibt es die Yaw-Sensoren, die die seitlichen Bewegungen der Beine verfolgen, sowie die Roll-Sensoren, die die Rotation des Stuhls erfassen.

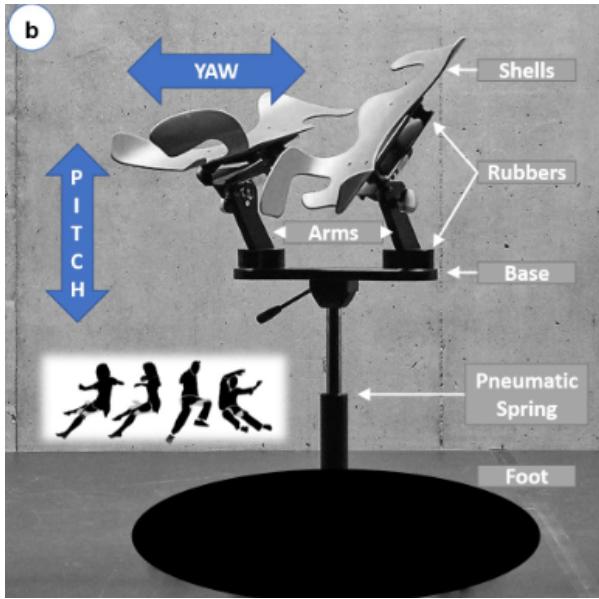


Abbildung 3.3: Limbic Chair mit eingezeichneten Pitch und Yaw Bewegung, ohne Roll Bewegung  
- Quelle: [2]

Die von diesen Sensoren gesammelten Daten bilden die Grundlage für die Entwicklung einer Software, die den Nutzern ermöglicht, ihr Sitzverhalten besser zu verstehen und zu optimieren. Ein wesentlicher Vorteil dieser Sensordaten ist die Fähigkeit, eine umfassende Statistik über die Sitz- und Bewegungsgewohnheiten des Nutzers zu erstellen. Diese Statistikanzeige bietet dem Nutzer Einblicke in sein aktuelles Sitzverhalten und motiviert ihn, Bewegungen in seinen Alltag zu integrieren.

Durch die bewusste Integration von beiläufigen Bewegungen kann der Nutzer seine Bewegung während des Sitzens erhöhen, was letztendlich zu einer verbesserten körperlichen Aktivität führt. Diese erhöhte Körperaktivität kann helfen, die negativen Auswirkungen des langen Sitzens zu minimieren und die allgemeine Gesundheit und das Wohlbefinden zu fördern. Der Limbic Chair und die darauf basierende Software bieten somit eine innovative Lösung, um die Herausforderungen des modernen Sitzalltags zu bewältigen und die Nutzer zu einem aktiveren Lebensstil zu motivieren.

## 3.4 Akteure

### 3.4.1 Hauptakteur: User

Der User ist der Endbenutzer der Software und jemand, der aktiv auf dem Stuhl sitzt und seine Bewegungen erfassen möchte. Sein Ziel ist es, neben seinen beruflichen Tätigkeiten Bewegung in den Alltag zu integrieren. Dabei unterstützt ihn die Software, indem sie ihm Statistiken zu seiner Bewegung aufzeigt und es ihm ermöglicht, sich spielerisch als Ablenkung im Sitzen zu bewegen.

### 3.4.2 Supportakteur: Limbic Chair

Der Limbic Chair wird benötigt, um die Software anzusteuern. Um interessante Daten auswerten zu können, muss der User auf dem Stuhl sitzen und ihn möglichst lange benutzen, um genaue Informationen zu erhalten.

### 3.5 Use Case Diagramm

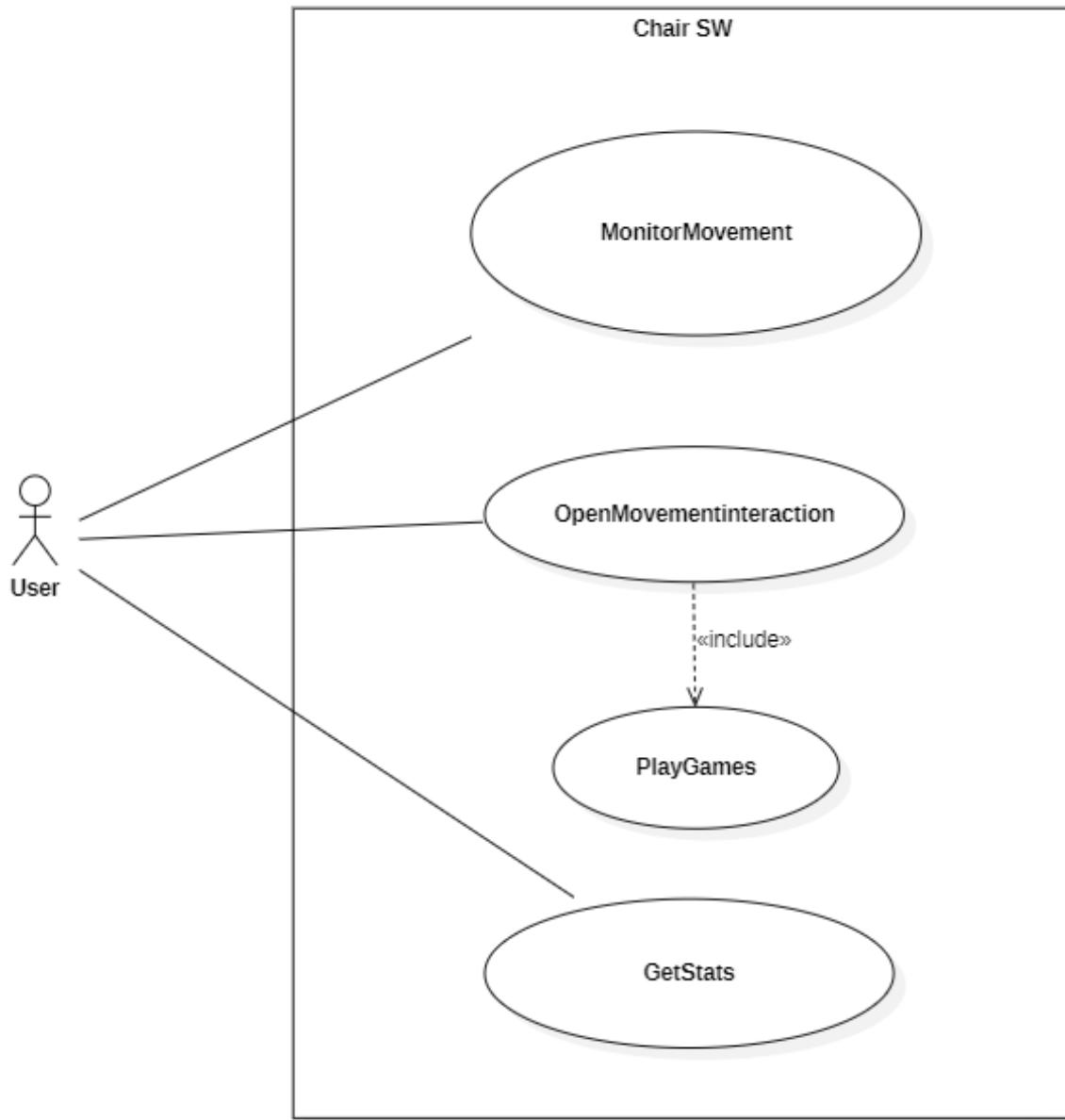


Abbildung 3.4: Use Case Diagramm

**MonitorMovement** Während der User auf dem Stuhl sitzt, werden seine Bewegungen erfasst und die Daten weiterverarbeitet. Diese Daten dienen dann zur Berechnung des Movementscores, der für den User relevant ist.

**GetStats** Der Use Case "GetStats" zielt darauf ab, die erfassten Informationen anzuzeigen. Dazu gehören der Movementscore und die Bewegungen beider Beine.

**OpenMovementinteraction** Dieser Use Case ermöglicht es dem User, externe Software zu starten, die auf die Bewegung mit dem Limbic Chair abzielt. Dazu gehören zwei Spiele, die im Rahmen dieser Arbeit entwickelt wurden.

### 3.6 Nicht-funktionale Anforderungen

In diesem Abschnitt lesen Sie über die Nicht-funktionale Anforderungen. Als Orientierung für unsere Voraussetzungen bedienen wir uns bei "Boehms Software Characteristics Tree" [3], um eine genaue Kategorisierung unserer Voraussetzungen zu machen.

Tabelle 3.1: NFR-1

ID	NFR-1
<b>Thema</b>	Benutzeroberfläche der App sollte intuitiv und leicht verständlich sein.
<b>Kategorie</b>	Benutzerfreundlichkeit
<b>Wichtigkeit</b>	Hoch
<b>Beschreibung</b>	Die Benutzeroberfläche der App sollte intuitiv und leicht verständlich sein, sodass Benutzer ohne vorherige Einweisung die Funktionen der Applikation nutzen können.
<b>Überprüfbarkeit</b>	Usability Tests mit Benutzern, um sicherzustellen, dass sie die App ohne Hilfe verwenden können.
<b>Status</b>	Benutzeroberfläche wurde mit fünf Probanden getestet. Probanden waren immer in der Lage die Benutzeroberfläche selbst zu bedienen. Siehe Kapitel: Testing

Tabelle 3.2: NFR-2

ID	NFR-2
<b>Thema</b>	Push Benachrichtigungen anzeigen
<b>Kategorie</b>	Benutzerfreundlichkeit
<b>Wichtigkeit</b>	Hoch
<b>Beschreibung</b>	Push Benachrichtigungen sollen angezeigt werden wenn der Movementscore zu tief ist. Falls der Movementscore von Tief zu hoch geht soll dies auch eine Push Benachrichtigung auslösen
<b>Überprüfbarkeit</b>	Wenn der Movementscore unter 500 ist, soll die Push Benachrichtigung ausgelöst werden. Nachdem der Score mit einem Wert von unter 500 auf einen Wert von über 1500 geht soll nochmals eine Push Benachrichtigung ausgelöst werden.
<b>Status</b>	Benachrichtigung erscheint immer sobald der Movementscore unter 500 ist. Bei einem Wert von über 1500 wird die Push Benachrichtigung auch immer aktiviert.

Tabelle 3.3: NFR-3

ID	NFR-3
<b>Thema</b>	Berechnung des Movementscore
<b>Kategorie</b>	Verständlichkeit (Understandability)
<b>Wichtigkeit</b>	Hoch
<b>Beschreibung</b>	Basierend der Bewegungen im Stuhl soll eine Movementscore berechnet werden welcher die aktuelle Bewegungsaktivität anzeigt.
<b>Überprüfbarkeit</b>	Der Movementscore soll grösser werden bei Bewegungen im Stuhl. Falls keine Bewegungen im Stuhl getrackt werden können, muss der Score automatisch sinken.
<b>Status</b>	Movementscore steigt in allen Fällen bei Bewegungen. Beim Stillstand des Stuhles ist der Wert immer gesunken. Siehe Kapitel: Movement Score

Tabelle 3.4: NFR-4

ID	NFR-4
<b>Thema</b>	Der Prozess zwischen dem automatischem Öffnen des Clients und dem start des Spiels soll nicht länger als eine Minute dauern
<b>Kategorie</b>	Benutzerfreundlichkeit
<b>Wichtigkeit</b>	Hoch
<b>Beschreibung</b>	Nachdem die Statusbar den roten Bereich erreicht öffnet sich der Client und man das Spiel sollte in weniger als einer Minute gestartet werden können.
<b>Überprüfbarkeit</b>	Der Prozess zwischen Start des Clients und start des Spiels soll nicht länger als eine Minute dauern
<b>Status</b>	Durchschnittszeit bei erstmaligem starten der Spiele über 30 Sekunden aber nie länger als eine Minute. Ansonsten sogar immer schneller als 30 Sekunden.

Tabelle 3.5: NFR-5

ID	NFR-5
<b>Thema</b>	Zugriffbeschränkung und Sichtbarkeit der Benutzerdaten
<b>Kategorie</b>	Security
<b>Wichtigkeit</b>	Hoch
<b>Beschreibung</b>	Die Applikation muss persönliche Daten vom Benutzer sicher verarbeiten und speichern. Jeder Benutzer soll in der Applikation seine eigene Daten einsehen und zugreifen können.
<b>Überprüfbarkeit</b>	Es wird ein automatisierter Test implementiert, der den Anmeldeprozess des Benutzers nachbildet und nach erfolgreichem Login überprüft, ob die Daten den richtigen Benutzer zugeordnet wurde.
<b>Status</b>	Alle Unit-Tests wurden erfüllt. Siehe Kapitel: Unit Testing

# Kapitel 4: Architektur

## 4.1 Aktivitätsverlauf

In diesem Abschnitt wird beschrieben, wie das Sitzen auf dem Stuhl und die Interaktion mit der Software funktioniert. Der Benutzer muss zuerst seinen Stuhl mit dem Computer verbinden. Dabei ist zu beachten, dass der Computer mit Windows betrieben wird. Der Stuhl bietet die Möglichkeit, sich entweder über Kabel oder drahtlos mit dem Computer zu verbinden. Da die drahtlose Funktion nie zuverlässig funktioniert hat, wird in dieser Arbeit der Fokus auf die Verbindung mittels Kabel gelegt.

Anschliessend wird die Software Limbic Control Service gestartet und der Stuhl mit dem Computer verbunden. Dies ist notwendig, damit die Daten der Sensoren konstant an den Computer gesendet werden können. Danach muss der Benutzer auf dem Stuhl Platz nehmen und seine Position gerade ausrichten. In der Software Limbic Control Service drückt er dann auf den Button "Recenter", um später die Daten basierend auf dieser Position zu erhalten. Das bedeutet, dass alle Veränderungen der Sensoren auf Basis der ausgerichteten Position berechnet werden.

Nachdem der Benutzer seinen Stuhl konfiguriert hat, kann die Software ActiveSit gestartet werden. Beim Start stellt ActiveSit eine Verbindung zum Limbic Control Service her, um die Rohdaten der Sensorwerte auszulesen und zu speichern. Diese Daten sind für die Auswertung der Statistiken notwendig. Der Benutzer hat folgende Möglichkeiten, die er frei nutzen kann: Zum einen kann er die Statistiken über die Bewegungen seines linken und rechten Beins einsehen. Zum anderen kann er seinen Movementscore betrachten, der eine berechnete Punktzahl aus seinen bisherigen Bewegungen darstellt.

Neben der Möglichkeit, Statistiken einzusehen, kann der Benutzer den Stuhl aktiv nutzen. Dank der Mobilität des Stuhls kann er sich entweder während seiner gewöhnlichen Beschäftigungen wie Lernen oder Arbeiten bewegen oder sich bewusst für eine aktive Nutzung entscheiden. Falls der Benutzer sich für aktive Bewegung entscheidet, bietet die Software die Möglichkeit, eines der verfügbaren Spiele zu spielen. Bei einem Spiel liegt der Fokus auf aktiver Bewegung, wobei der Benutzer aktiv Bewegungen ausführen muss, um im Spiel voranzukommen. Beim anderen Spiel geht es darum, eine gewisse Balance zu halten und die Feinmotorik zu fördern. Wenn der Benutzer sich entscheidet, die Spiele zu spielen, wird sein Movementscore durch die Aktivität im Spiel erhöht.

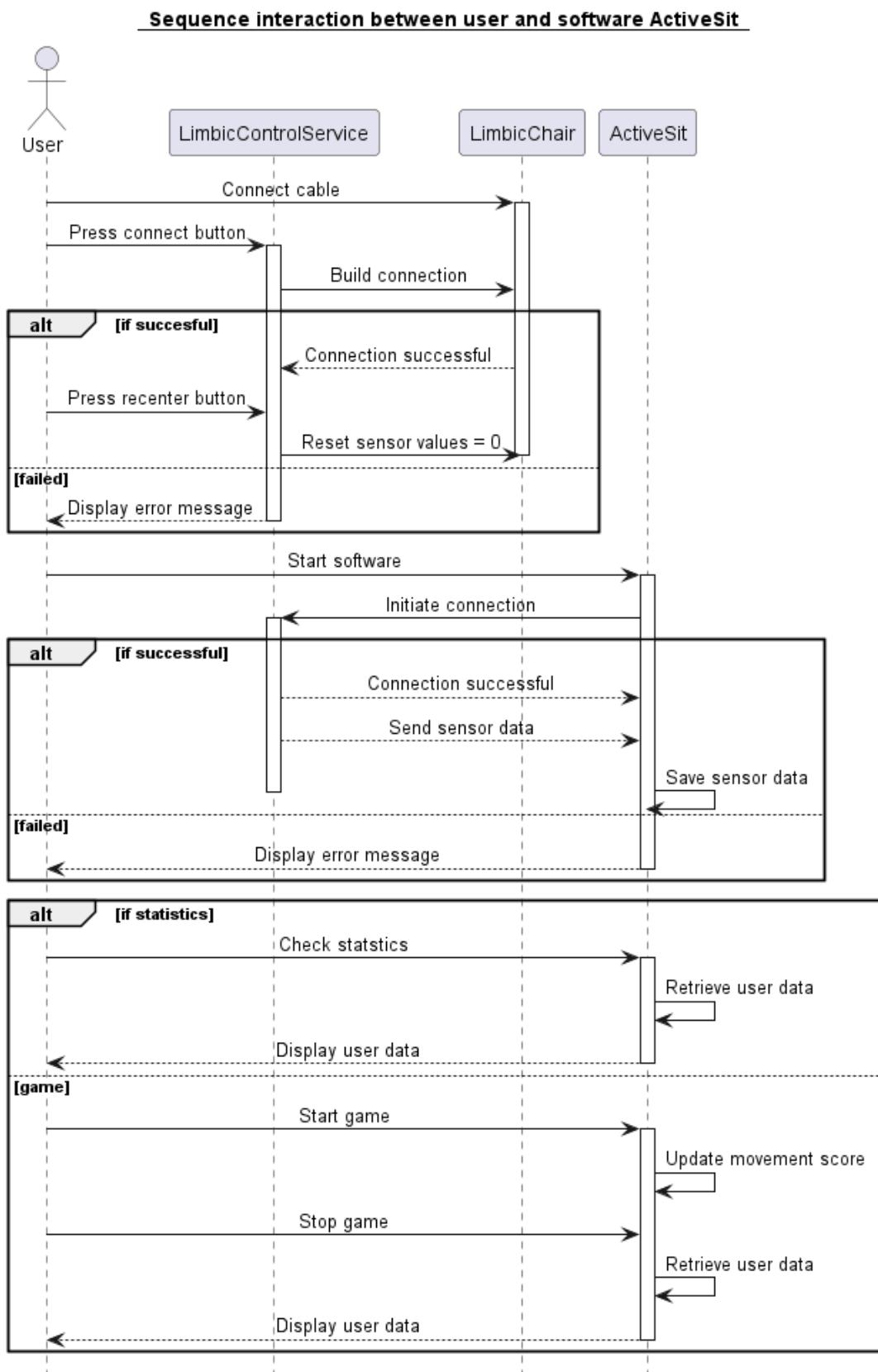


Abbildung 4.1: Sequenz Diagramm - Aktivitätsverlauf

## 4.2 Funktion der Software ActiveSit

Unsere Software besteht aus vier Hauptkomponenten. Die erste Komponente ist das Frontend. Das Frontend ist der zentrale Bestandteil der Software, über den der Benutzer navigieren kann. Es bietet sowohl die Möglichkeit, Statistiken einzusehen, als auch externe Programme wie Spiele zu öffnen.

Die Spiele stellen eine weitere Hauptkomponente der Software dar. Hierbei handelt es sich um zwei externe Programme, die über das Frontend aufgerufen werden. Diese Spiele dienen dazu, sich aktiver im Stuhl zu bewegen und haben einen direkten Einfluss auf die Statistiken.

Ein weiterer wichtiger Bestandteil ist der Limbic Service. Dabei handelt es sich um eine Software, die Daten an mehrere Prozesse weiterleitet. Wie der Limbic Service genau funktioniert, wird im Kapitel 4.2.5 Limbic Service genauer erklärt.

Die vierte und letzte Hauptkomponente ist das Backend. Das Backend besteht aus einer REST-API und einer Datenbank, die für das Speichern der Daten verantwortlich ist. Das Frontend greift auf die REST-API zu, und diese holt die benötigten Daten von der Datenbank, um sie darstellen zu können.

In den folgenden Unterabschnitten wird die Funktionsweise der einzelnen Komponenten genauer beschrieben.

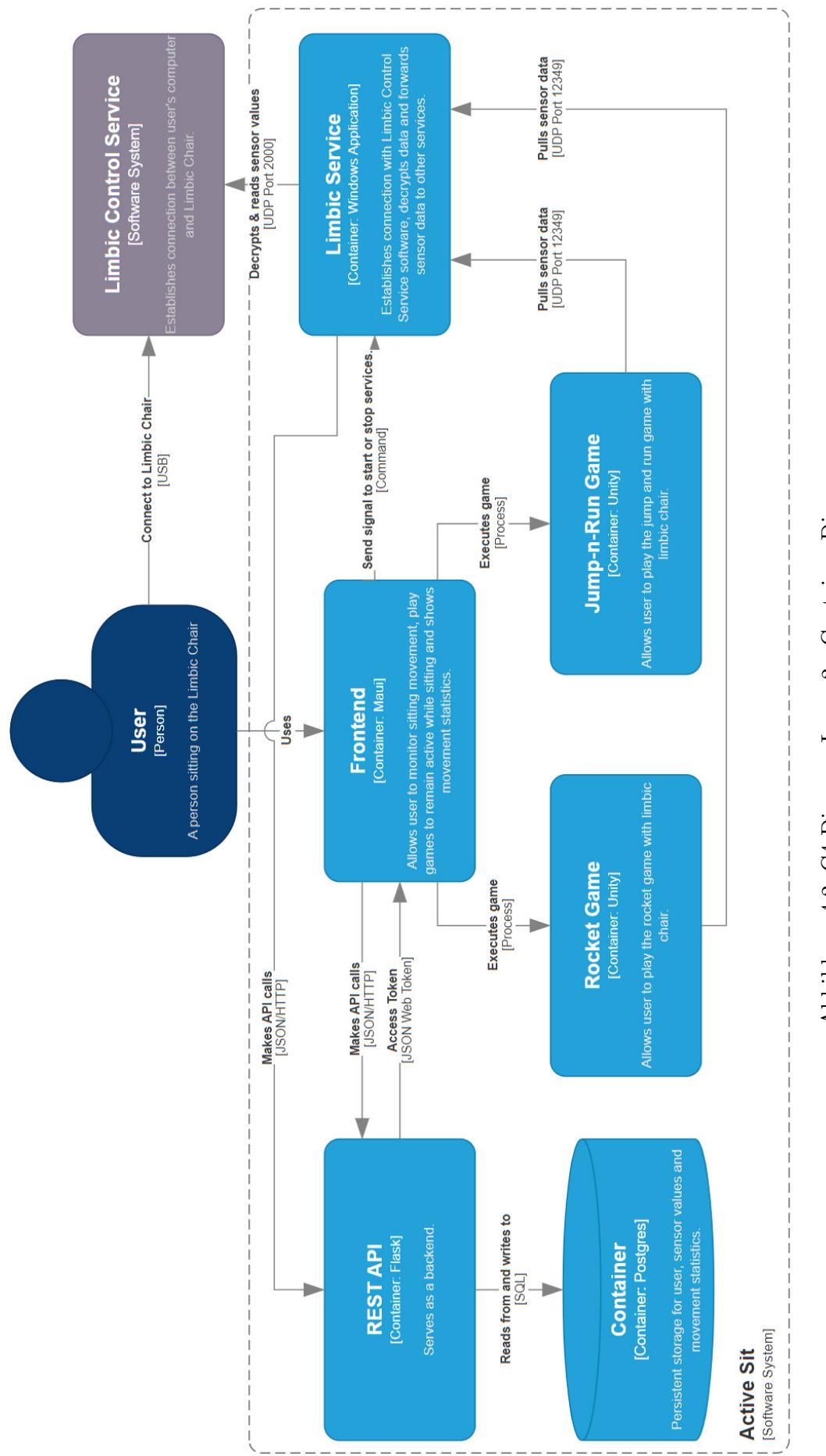


Abbildung 4.2: C4 Diagram Layer 2 - Container Diagramm

### 4.2.1 Frontend

Das Frontend unserer Software ist speziell für Benutzer des Stuhls entwickelt worden. Nachdem sich der Benutzer mit dem Stuhl verbunden hat, kann er die Software ActiveSit starten. Zunächst muss der Benutzer ein Benutzerkonto erstellen, damit er sich anschliessend beim Login-Screen einloggen kann.

Nach dem Einloggen verbindet sich der Benutzer mit dem Limbic Service und empfängt die Daten. Diese aufgezeichneten Daten werden an das Backend weitergeleitet und in der Datenbank abgespeichert. Gleichzeitig erhält der Benutzer ein Zugriffstoken, das ihm erlaubt, andere Seiten zu besuchen. Anschliessend gelangt der Benutzer auf die Statistikseite, auf der er seinen Movementscore einsehen und abrufen sowie die Bewegungsdaten beider Beine betrachten kann. Von dort aus kann er zur Gamingseite navigieren, wo die Möglichkeit besteht, Spiele zu öffnen. Sollte sich ein anderer Benutzer auf den Stuhl setzen, hat der aktuelle Benutzer die Möglichkeit, sich auszuloggen.

Das Frontend ist mit .NET MAUI entwickelt. Es wird das MVVM-Pattern (Model-View-ViewModel) verwendet, welches mit .NET MAUI sehr gut funktioniert und dafür geeignet ist. Auf der nächsten Seite wird gezeigt, wie das Frontend aufgebaut ist und wie die einzelnen Komponenten miteinander verbunden sind:

Tabelle 4.1: Beschreibung Frontend

Komponent	Aufgabe
Views	Beim Start der Applikation muss der Benutzer sich zuerst anmelden. Falls noch kein Benutzerkonto eingerichtet ist, kann dies direkt auf der Authentication Page gemacht werden. Bei einer erfolgreichen Authentifizierung wird der Benutzer zur Statistics Page weitergeleitet. Ab hier kann der Benutzer beliebig zwischen der Statistics Page und der Gaming Page navigieren und sich von beiden Seiten aus ausloggen.
View Models	Für jede Page wird ein ViewModel erstellt. Ein ViewModel bindet die Daten an die Pages, wodurch sichergestellt wird, dass die Daten vom Backend auf den Pages angezeigt werden können. Zusätzlich wird hier auch die Logik implementiert, um die Pages interaktiv für die Benutzer zu machen. Besonders beim Gaming View Model wurde die Logik implementiert, um die Spiele zu starten.
Services	Die Services werden verwendet, um mit dem Backend zu kommunizieren. Sie ermöglichen die Übergabe von Daten im JSON-Format an das Backend, damit diese verarbeitet werden können. Außerdem können die Services Daten vom Backend abrufen, um sie in den Views zu visualisieren.

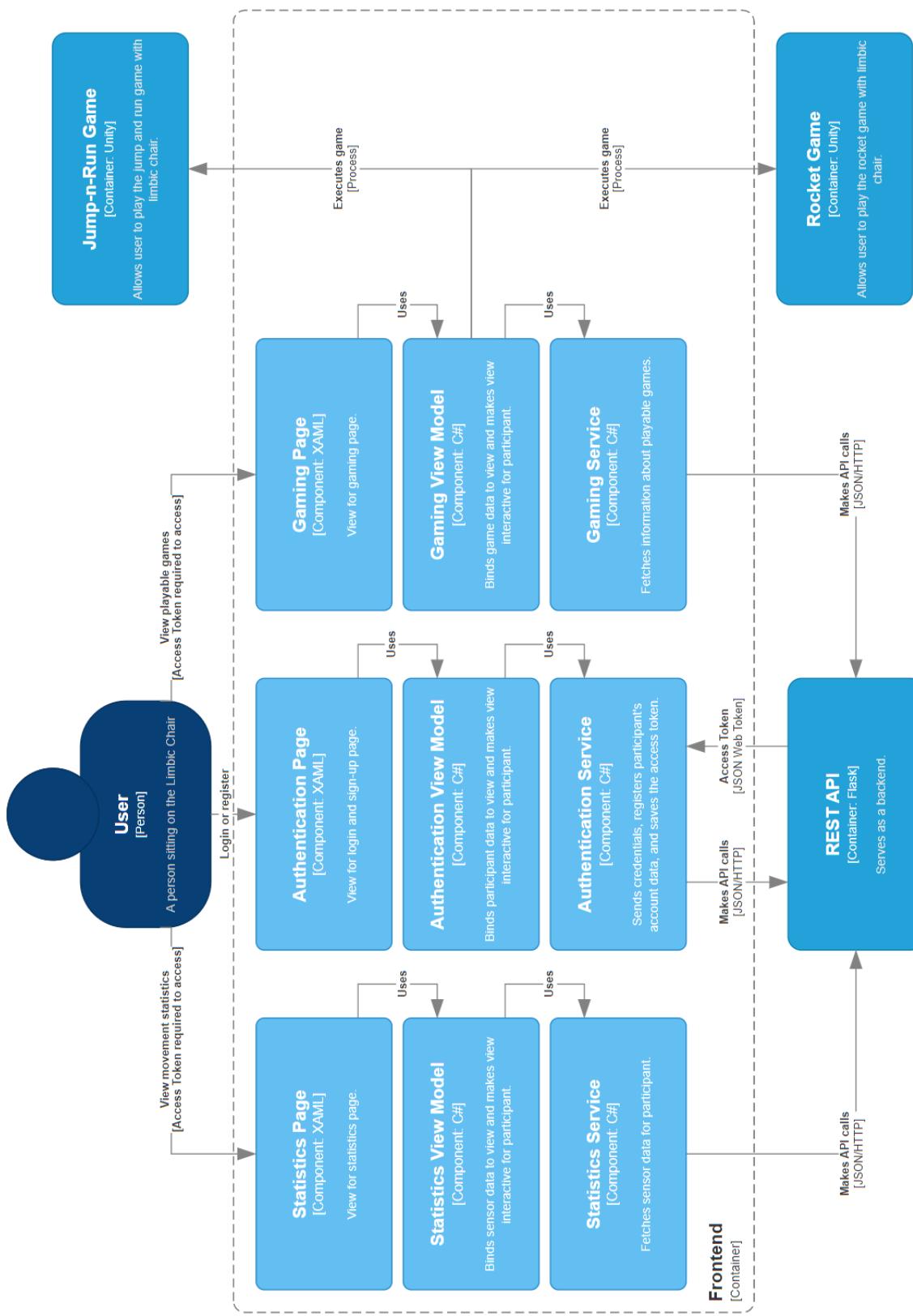


Abbildung 4.3: C4 Diagram Layer 3 - Frontend

### 4.2.2 Dashboard

Für die Darstellung und Auswertung von Daten wird ein Dashboard entwickelt. Mit dem Dashboard geben wir dem Benutzer die Möglichkeit, seine Statistiken einzusehen. Dabei kann er sowohl den Movement Score als auch die Bewegungen der Beine einsehen. Zusätzlich hat er die Möglichkeit, externe Spiele über das Dashboard zu öffnen.

Im Folgenden wird erläutert wie das Frontend von Beginn einer "Sitzungssession" bis zum Schluss eingesetzt wird.

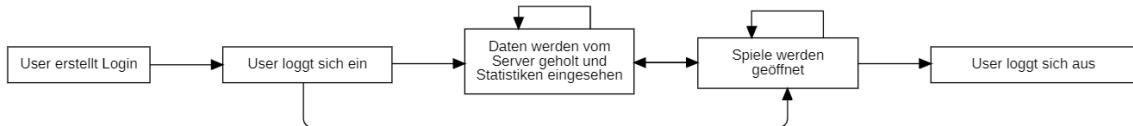


Abbildung 4.4: Ablauf Sitzungsseesion

Tabelle 4.2: Ablauf einer Sitzungsseesion im Frontend

Nummerierung	Beschreibung
1.	Der Benutzer legt über das Frontend einen Account an, indem er ein Passwort und eine E-Mail-Adresse hinterlegt.
2.	Nach der Registrierung kann sich der Benutzer einloggen, und ab diesem Zeitpunkt werden die Daten aufgezeichnet.
3.	Mithilfe eines "Refresh Data" Buttons hat der Benutzer die Möglichkeit, die Daten vom Backend abzurufen und seine Statistiken einzusehen.
4.	Der Benutzer hat die Möglichkeit auf die Gamingseite zu gehen, wo er mittels einem Mausklick die Spiele öffnen kann.
5.	Falls der Benutzer keine Lust mehr hat oder sich eine andere Person auf den Stuhl setzen möchte, kann er sich ausloggen. Ab dem Zeitpunkt des Ausloggens wird auch die Datensammlung und -verarbeitung gestoppt.

### Wireframes

Um eine klare Richtlinie für das Design und die Struktur des Dashboards zu schaffen, werden Wireframes entwickelt. Die Wireframes dienen dazu, dass das Team nicht unstrukturiert mit der Umsetzung beginnt, sondern von Anfang an einen klaren Überblick über die Layout- und Funktionsaspekte des Dashboards hat. Diese sollten als Inspirationsquelle betrachtet werden, die als Leitfaden für das Design dient und auch Raum für kreative Anpassungen und Verbesserungen lässt. Die Wireframes werden im Anhang hinzugefügt.

## Aufbau Dashboard

Das Userflow-Diagramm bietet einen Gesamtüberblick darüber, wie auf die einzelnen Seiten zugegriffen wird. Es visualisiert den Pfad, den ein Benutzer durch die verschiedenen Abschnitte der Webseite nimmt, inklusive der Entscheidungspunkte und Navigationsmöglichkeiten. Die genauen Funktionen oder Aufgaben jeder Seite sowie das Aussehen der Seite werden in den folgenden Abschnitten beschrieben.

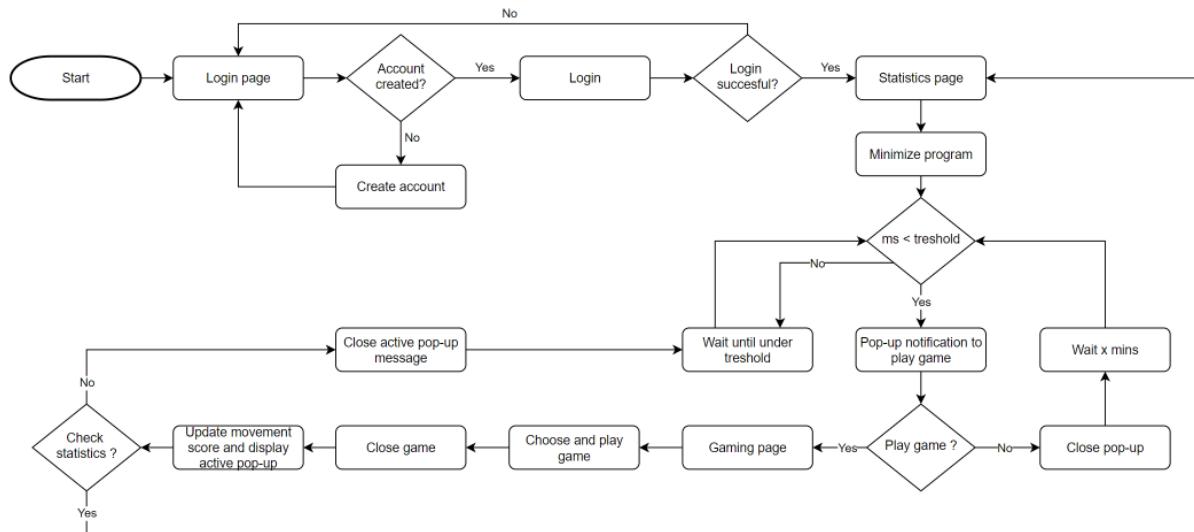


Abbildung 4.5: Userflow Diagramm

## Dashboard Statistics Page

Nach der Login-Seite sieht der Benutzer die Statistik-Seite. Dort kann er den Movementscore einsehen, der seinen aktuellen Bewegungsstand anzeigt. Mit dem "Refresh Data" Button oben hat er die Möglichkeit, die Daten vom Server abzurufen, um seinen Movementscore aktuell zu halten.

Der "Refresh Data" Button auf der Statistik-Seite ist notwendig, weil das Frontend nicht in der Lage ist, Daten in Echtzeit anzuzeigen. Echtzeit-Datenaktualisierungen würden erhebliche Performance-Probleme verursachen. Der Ladevorgang würde mehrere Minuten dauern, was für die Benutzererfahrung nicht akzeptabel wäre. Durch den Refresh-Button kann der Benutzer die Daten manuell vom Server abrufen und so seinen Movementscore aktuell halten. Diese Lösung stellt sicher, dass die Benutzer immer auf dem neuesten Stand sind, ohne dass das Backend durch kontinuierliche Echtzeit-Abfragen überlastet wird. Auf diese Weise bleibt die Anwendung performant und reaktionsschnell.

Links in der Navigationsleiste hat der Benutzer die Möglichkeit, die Spiel-Seite aufzurufen oder sich bei Bedarf auszuloggen.

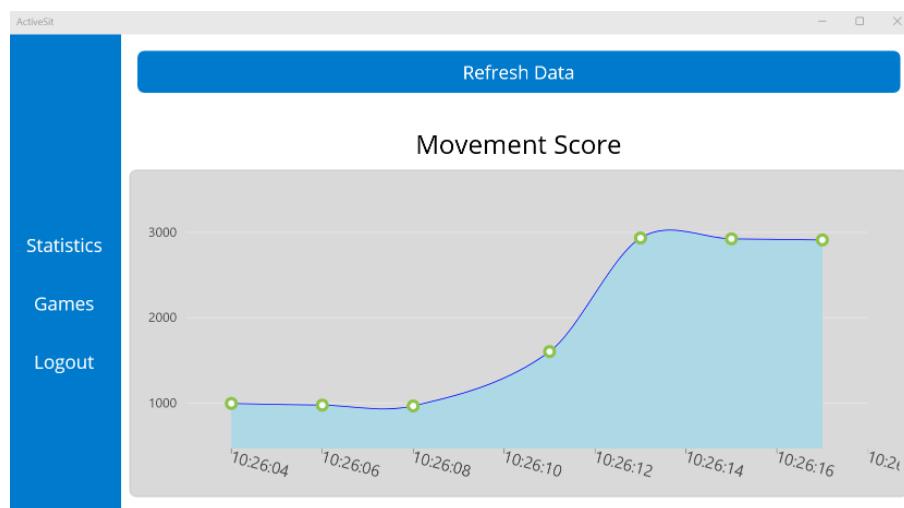


Abbildung 4.6: Dashboard Statistics Page

## Dashboard Gaming Page

Der Benutzer hat die Möglichkeit, über die linke Navigationsleiste zur Spiel-Seite zu navigieren. Dort kann er eines der beiden Spiele starten, indem er auf den Play-Button klickt.

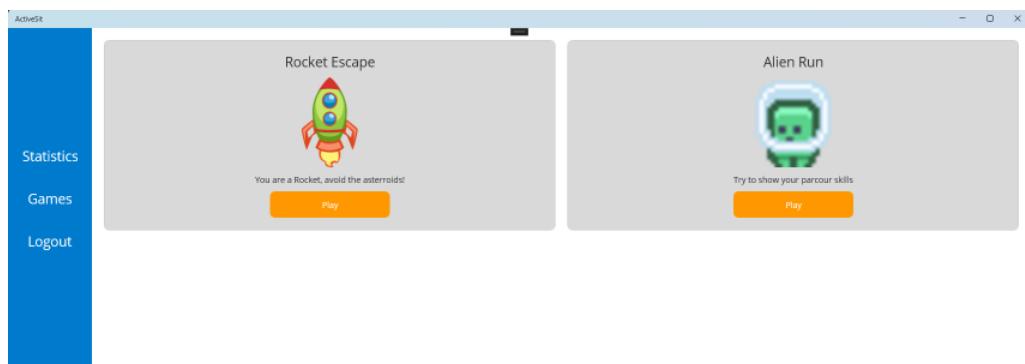


Abbildung 4.7: Dashboard Statistics Page

### 4.2.3 Games

Um dem Benutzer die Möglichkeit zu geben, sich spielerisch aktiv im Stuhl zu bewegen, wurden zwei Spiele entwickelt. Im ersten Spiel, "Rocket Escape", steuert man eine Rakete und versucht, den fliegenden Asteroiden zu entkommen. Das zweite Spiel, "Alien Run", ist ein Jump-and-Run-Spiel, bei dem sich der Alien nur bewegt, wenn man sich im Stuhl bewegt. In den folgenden Abschnitten wird beschrieben, wie der Stuhl in die Spiele eingebunden wird und wie die Integration des Stuhls in den Spielen funktioniert.

#### Unity

Für die Entwicklung der Spiele wurde die Entwicklungsplattform Unity verwendet. Unity ist bekannt für die Erstellung von 2D- und 3D-Spielen sowie interaktiven Erlebnissen. Die Entscheidung für Unity fiel, weil Limbic Life uns bereits Spiele gezeigt hat, die mit dem Stuhl funktionieren und über Unity programmiert wurden.

#### Rocket Escape

Beim Spiel "Rocket Escape" geht es darum, fliegenden Asteroiden auszuweichen. Die Asteroiden fliegen auf die Rakete zu, und der Spieler muss durch Bewegungen im Stuhl die Rakete nach oben oder unten steuern, um ihnen zu entkommen. Für die Steuerung wurde entschieden, dass sich der Spieler nach vorne lehnen muss, um die Rakete nach unten zu bewegen, und nach hinten, um die Rakete nach oben zu steuern.

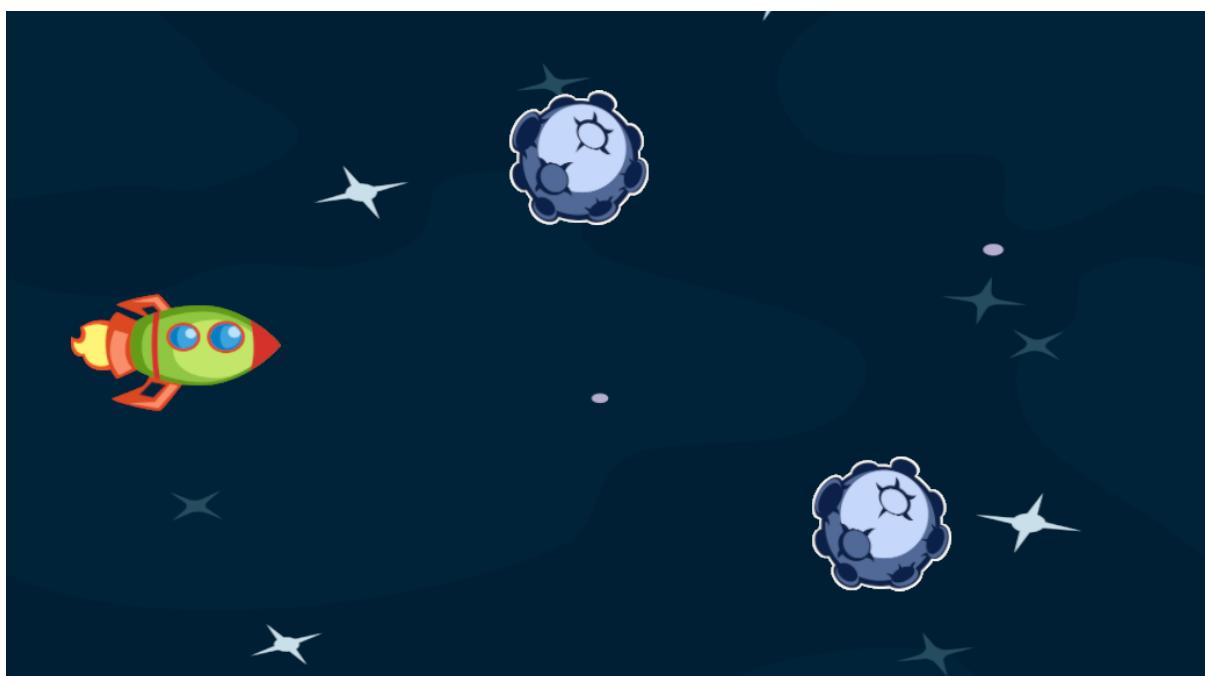


Abbildung 4.8: Rocket Escape

Nachdem das Programm gestartet wurde, versucht das Spiel, eine Verbindung mit dem Limbic Forwarder aufzubauen, um Bewegungsdaten zu erhalten. Der Limbic Forwarder sendet diese Daten dann über UDP-Pakete an das Spiel, das sich zuvor beim Service angemeldet hat. Der ClientReceiver im Spiel übernimmt die Aufgaben, die Registrierungsnachricht zu senden, Daten asynchron zu empfangen und die Bewegungsinformationen des Stuhls zu verarbeiten. Diese Informationen werden anschliessend im Spiel verwendet, um auf die Bewegungen des Nutzers zu reagieren und entsprechende Rückmeldungen zu geben.

```
public void RegisterWithServer()
{
    var registrationMessage = new RegistrationMessage { Type = "Register" };
    string jsonMessage = JsonUtility.ToString(registrationMessage);
    byte[] data = Encoding.UTF8.GetBytes(jsonMessage);

    try
    {
        udpClient.Send(data, data.Length, serverEndPoint);
        BeginReceivingData();
    }
    catch (Exception e)
    {
        Debug.LogError($"Failed to send registration message: {e.Message}");
    }
}
```

Abbildung 4.9: Clientreceiver bei der Anmeldung zum Connector

Zu Beginn jeder Runde werden die Anfangswerte für LeftPitch und RightPitch gespeichert. Diese Anfangswerte werden zusammengezählt, um die Startposition des Nutzers als Mittelpunkt der Rakete im Spiel zu definieren. Spätere Bewegungen werden dann relativ zur Startposition gemessen.

In der Update-Methode werden kontinuierlich die aktuellen Sensordaten vom ClientReceiver abgerufen. Danach werden die relativen Veränderungen der Pitch-Werte zur Startposition berechnet. Diese relativen Pitch-Werte werden innerhalb eines bestimmten Bereichs von 0 bis 20 gehalten. Der Grund dafür ist, extreme Bewegungen des Stuhls zu vermeiden und ein angenehmeres Spielerlebnis zu gewährleisten. Das bedeutet, dass der Nutzer sich nicht vollständig nach unten bewegen muss, um die maximale Bewegung der Rakete zu erreichen. Eine relative Veränderung von 20 Einheiten reicht aus, um die Rakete vollständig zu steuern.

Die Berechnung der Raketenposition erfolgt dann durch die Summe der angepassten LeftPitch- und RightPitch-Werte. Basierend auf dieser Summe wird der y-Wert für die Raketenposition mithilfe einer linearen Formel angepasst:

$$y = (13.5 - 1.325 \cdot xSum)$$

Der Wert 13,5 wurde gewählt, weil dies die höchste Y-Position im Spiel darstellt. Der Koeffizient 1,325 wurde gewählt, um eine gleichmässige Verteilung der Bewegungen auf dem Spielfeld zu gewährleisten. Dadurch verursacht jede Einheit des Pitch-Werts eine proportional gleich grosse Verschiebung im Spiel.

Nach jeder Runde werden die Startwerte für LeftPitch und RightPitch neu berechnet, da die Werte immer relativ zueinander berechnet werden. Durch dieses Steuerungsschema kann der Nutzer die Rakete nach oben oder unten bewegen, indem er sich nach vorne oder hinten lehnt.

Das System interpretiert die Veränderungen in den Pitch-Werten des Stuhls und übersetzt diese in eine vertikale Bewegung der Rakete, wodurch ein intuitives und reaktives Spielerlebnis geschaffen wird.

```
void Update()
{
    if (ClientReceiver.Instance != null)
    {
        sensorData = ClientReceiver.Instance.playerData;

        float relativeLeftPitch = sensorData.LeftPitch - startLeftPitch;
        float relativeRightPitch = sensorData.RightPitch - startRightPitch;

        float leftpitch = Mathf.Clamp(relativeLeftPitch, 0f, 20f);
        float rightpitch = Mathf.Clamp(relativeRightPitch, 0f, 20f);

        float xSum = leftpitch + rightpitch;

        y = (13.5f - 1.325f * xSum);

        playerDirection = new Vector2(0.4f, y);
    }
}
```

Abbildung 4.10: Berechnung der Y Position der Rakete

### Alien Run

Beim Spiel "Alien Run" handelt es sich um ein Jump-and-Run-Spiel, bei dem der Spieler einen Alien von Plattform zu Plattform springen lässt. Das Spiel ist in drei Sektoren unterteilt, die unterschiedliche Herausforderungen bieten.

Im ersten Sektor hat der Spieler die Möglichkeit, das Spiel zu testen und kann dabei nicht verlieren. Wenn der Alien von einer Plattform fällt, landet er auf dem Boden und kann wieder zurückgehen, um erneut auf die Plattform zu springen. Dies bietet eine risikofreie Umgebung, um sich mit den Spielmechaniken vertraut zu machen.

Im zweiten Sektor wird die Herausforderung erhöht. Der Spieler verliert sofort, wenn der Alien von einer Plattform fällt. Die Größe der Plattformen bleibt jedoch gleich wie im ersten Sektor. Dieser Sektor fordert den Spieler heraus, indem er präzises Springen verlangt, ohne die Plattformen schwieriger zu erreichen.

Der dritte Sektor stellt die höchste Schwierigkeitsstufe dar. Hier werden die Plattformen kleiner, und der Spieler verliert sofort, wenn der Alien herunterfällt. Dieser Sektor erfordert präzises Springen und perfekte Kontrolle, um die kleineren Plattformen sicher zu erreichen.

Das Ziel des Spiels ist es, den gesamten Parcours zu bewältigen. Wenn der Spieler alle drei Sektoren erfolgreich durchläuft, hat er das Spiel gewonnen.

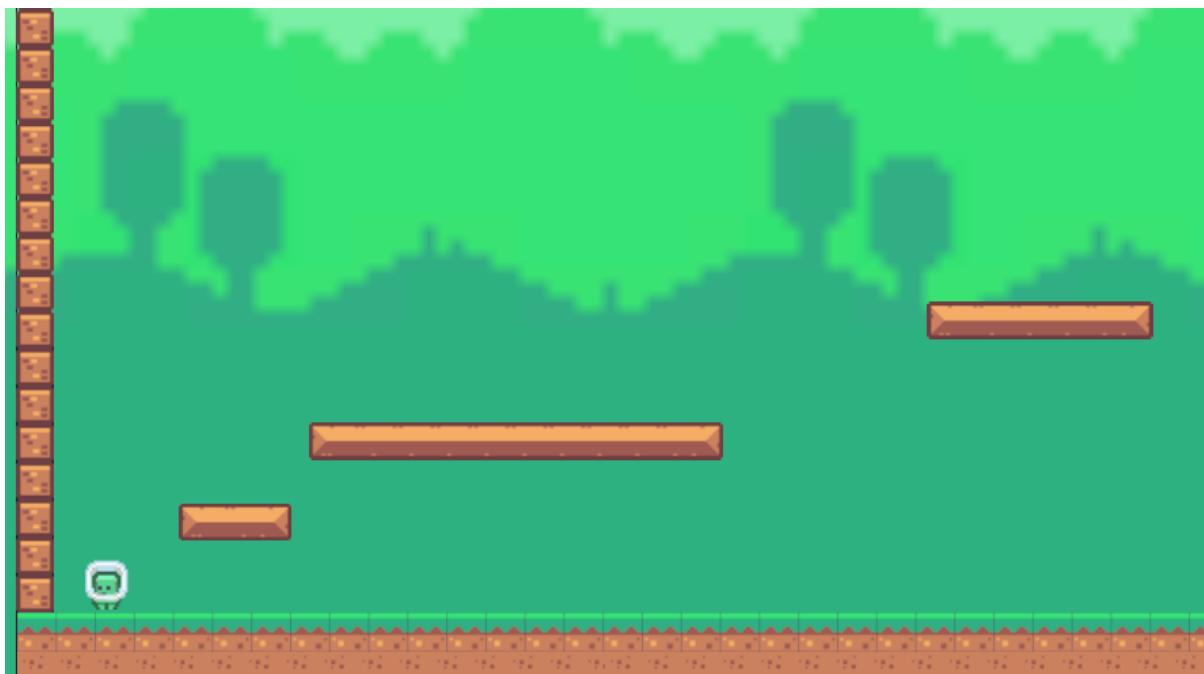


Abbildung 4.11: Spiel Alien Run

Nach dem Starten des Spiels wird erneut eine Verbindung mit dem Limbic Forwarder aufgebaut, um die Daten bezüglich des Stuhls zu erhalten. Dabei wird dasselbe Skript wie beim Spiel Rocket Escape verwendet.

Die Steuerung von Alien Run basiert auf den Eingaben der Tasten "A" und "D", mit denen sich der Spieler nach links beziehungsweise rechts bewegen kann. Das Spiel wartet auf diese Eingaben und überprüft dann, ob eine Bewegung stattfindet. Die Update-Methode wird ungefähr 60 Mal pro Sekunde aufgerufen, wodurch eine hohe Reaktionsgeschwindigkeit gewährleistet wird.

Wenn ein Tasteneingang erkannt wird, überprüft das Spiel die Bewegungsdaten des Spielers. Dazu wird die Summe der Pitch- und Yaw-Werte aus dem aktuellen und dem vorherigen Methodenaufruf verglichen. Wenn die Differenz dieser Summen grösser oder gleich 1 ist, wird der Charakter entsprechend bewegt. Andernfalls wird der Tasteneingang zwar registriert, aber der Charakter bleibt an seiner Position. Diese Methode stellt sicher, dass der Charakter nur bei einer tatsächlichen Bewegung des Spielers reagiert, was die Präzision der Steuerung erhöht und ein realistisches Spielerlebnis ermöglicht.

```
if (Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.D))
{
    float leftpitch = sensorData.LeftPitch;
    float rightpitch = sensorData.RightPitch;
    float leftyaw = sensorData.LeftYaw;
    float rightyaw = sensorData.RightYaw;
    float currentXSum = leftpitch + rightpitch + leftyaw + rightyaw;

    if (Mathf.Abs(currentXSum - lastXSum) >= 1)
    {
        body.velocity = new Vector2(horizontalInput * speed, body.velocity.y);
        lastXSum = currentXSum;
    }
}
```

Abbildung 4.12: Codeausschnitt für die Bewegung des Charakters

#### 4.2.4 Backend

Im Backend wird Flask eingesetzt. Flask wurde als Backend-Technologie ausgewählt, weil es minimalistisch und leichtgewichtig ist. Das REST-API muss hauptsächlich Authentifizierungsaufgaben durchführen, Sensordaten speichern und Informationen über die Spiele liefern. Das Backend der Anwendung folgt dem Model-View-Controller-Konzept, jedoch werden keine Views benötigt, da die Views bereits im Frontend erstellt sind. Auf der nächsten Seite ist das C4-Diagramm zu sehen, das den Aufbau des Backends zeigt.

Im Diagramm werden drei Controller verwendet: der Authentication Controller, der Gaming Controller und der Statistics Controller. Die Controller nehmen die Anfragen vom Client entgegen, bearbeiten diese und liefern Daten zurück, um sie im GUI darzustellen. In der folgenden Tabelle werden die Aufgaben der Controller erklärt:

Tabelle 4.3: Beschreibung Backend Controllers

Controller	Aufgabe
Authentication Controller	Der Authentication Controller ist für die Authentifizierung der Benutzer zuständig sowie für das Anlegen von Benutzerkonten. Bei einer erfolgreichen Authentifizierung wird ein JSON Web Token (JWT) erstellt, das benötigt wird, um sich im GUI zu navigieren und Daten abzurufen. Dadurch wird sichergestellt, dass niemand unbefugt auf die Daten im Backend zugreifen kann. Der Controller sendet außerdem dem Limbic Service den Befehl, seine Prozesse zu starten, um die Sensordaten auszulesen, zu speichern und den Movement Score zu berechnen. Sobald der Benutzer die Applikation stoppt oder sich ausloggt, werden alle Prozesse im Limbic Service terminiert.
Gaming Controller	Der Gaming Controller zeigt alle verfügbaren Spiele, die direkt im GUI gestartet werden können. Beim Starten eines Spiels wird die entsprechende ausführbare Datei ausgeführt, wodurch das Spiel geöffnet wird.
Statistics Controller	Der Statistics Controller ist für die Darstellung des Movementscores verantwortlich und zeigt den Verlauf der Aktivitäten des Benutzers. Zusätzlich werden die Sensordaten des rechten und linken Stuhls angezeigt, um den Ausschlag der Winkel zu visualisieren, in denen sich der Benutzer mit dem Stuhl bewegt hat. Dies bietet einen guten Überblick darüber, wann der Benutzer aktiv oder statisch gesessen hat.

Ergänzend wurden Service-Komponenten für jede Tabelle in der Datenbank implementiert, die eine Vielzahl von CRUD-Operationen ausführen können. Diese Architektur erleichtert die Verwaltung des Codes und stellt sicher, dass das System einfach gewartet und erweitert werden kann.

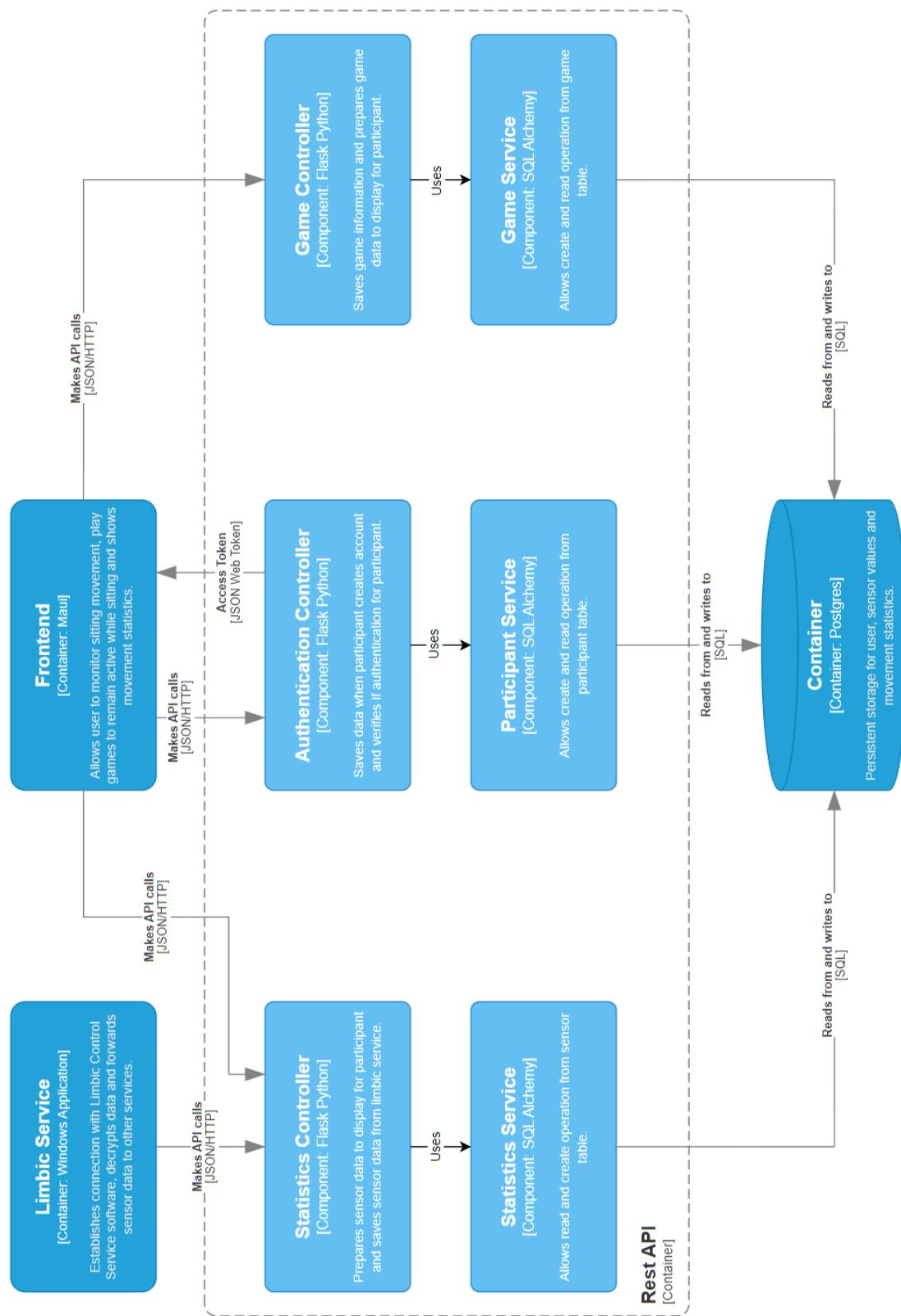


Abbildung 4.13: C4 Diagram Layer 3 - Backend

#### 4.2.5 Datenbank

Als Datenbank wird eine PostgreSQL-Datenbank verwendet. Im ERD-Diagramm werden die Tabellen aufgelistet, die in der Datenbank für das ActiveSit-System vorhanden sind. Das System ist entscheidend für die Speicherung der Sensordaten, was notwendig ist, damit die Resultate der Sensoren im Dashboard angezeigt werden können. Während die Datenbank nur die Pfade der Spiele speichert, befinden sich die Spiele auf dem C-Laufwerk unter C:\Program Files\<Game\_Folder>\<Game\_Name>.exe.

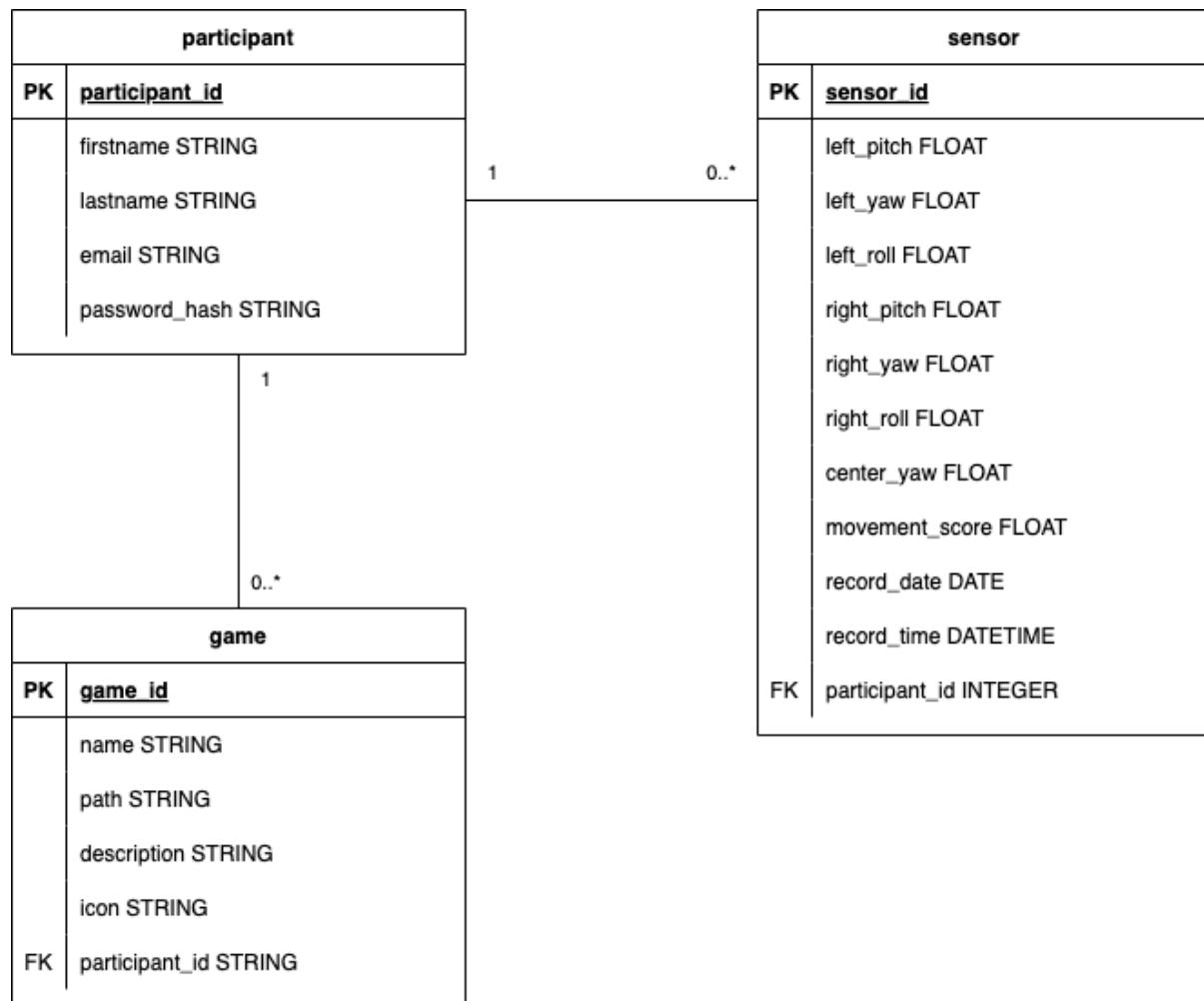


Abbildung 4.14: ERD Diagramm

#### 4.2.6 Limbic Service

Der Limbic Service kontrolliert alle Prozesse, um die Daten von der Limbic Chair Software zu steuern. Auf der nächsten Seite werden die Komponenten des Limbic Services im C4-Diagramm gezeigt und wie sie miteinander verbunden sind. In der folgenden Tabelle werden die einzelnen Aufgaben des Services erklärt:

Tabelle 4.4: Beschreibung Limbic Services

Service	Aufgabe
IPCSERVER	Der IPC-Server hört auf Befehle, die er vom Frontend erhält. Sobald sich ein Benutzer erfolgreich authentifiziert hat, sendet das Frontend dem IPC-Server den Befehl, alle Prozesse zu starten. Die Reihenfolge ist im C4-Diagramm nummeriert und kann dort nachgelesen werden. Sobald sich der Benutzer ausloggt, sendet der Client den Befehl, alle Prozesse zu beenden. Auch hier ist die Reihenfolge gleich wie beim Starten der Prozesse.
Limbic Service Connector	Der Limbic Service Connector verbindet sich mit der externen Limbic Control Service Software. Die Sensordaten in der Limbic Chair Software sind verschlüsselt, daher versucht der Limbic Service Connector zuerst, die Daten zu entschlüsseln. Wenn dieser Schritt erfolgreich ist, können die Rohwerte ausgelesen werden. Anschliessend werden die Rohdaten an den Limbic Forwarder weitergeleitet. Dies ist die wichtigste Komponente in der Limbic Service Applikation. Wenn die Verbindung zwischen der Limbic Chair Software und dem Limbic Service Connector ausfällt, erhalten die anderen Anwendungen die Sensordaten nicht mehr, und die gesamte Applikation würde somit nicht mehr funktionieren. Daher sind die anderen Komponenten sowie Anwendungen vom Limbic Service Connector abhängig.
Limbic Service Forwarder	Da die Limbic Control Service Software nicht in der Lage ist, mehrere Verbindungen gleichzeitig zu verarbeiten, weil die Daten nur von einer Anwendung abgehört werden können, wurde als Zwischenlösung ein Forwarder implementiert. Dieser stellt die Sensordaten für andere Dienste zur Verfügung und ermöglicht, dass mehrere Verbindungen parallel laufen können und gleichzeitig Daten pullen können. Auf diese Weise können verschiedene Anwendungen die benötigten Daten flexibel und effizient abrufen, was mit der ursprünglichen Limbic Control Service Software bis jetzt nicht möglich ist.
Movement Score Service	Der Movement Score Service berechnet für den Benutzer den Movement Score und speichert die Sensordaten sowie den Movement Score in der Datenbank. Es wurde so eingestellt, dass nur alle zwei Sekunden ein Eintrag in der Datenbank erstellt wird. Wenn jede Millisekunde ein Eintrag gespeichert würde, könnte der API-Server durch die Anfragen überlastet werden, was zu Performanceproblemen führen könnte. Außerdem macht es keinen Sinn, jede Millisekunde Daten zu speichern, da auch bei einem Intervall von zwei Sekunden die Genauigkeit noch gewährleistet ist. Zusätzlich ist dieser Service auch dafür verantwortlich, die Benutzer daran zu erinnern, sich zu bewegen, oder eine Meldung zu geben, wenn der Movement Score über dem Schwellwert liegt.

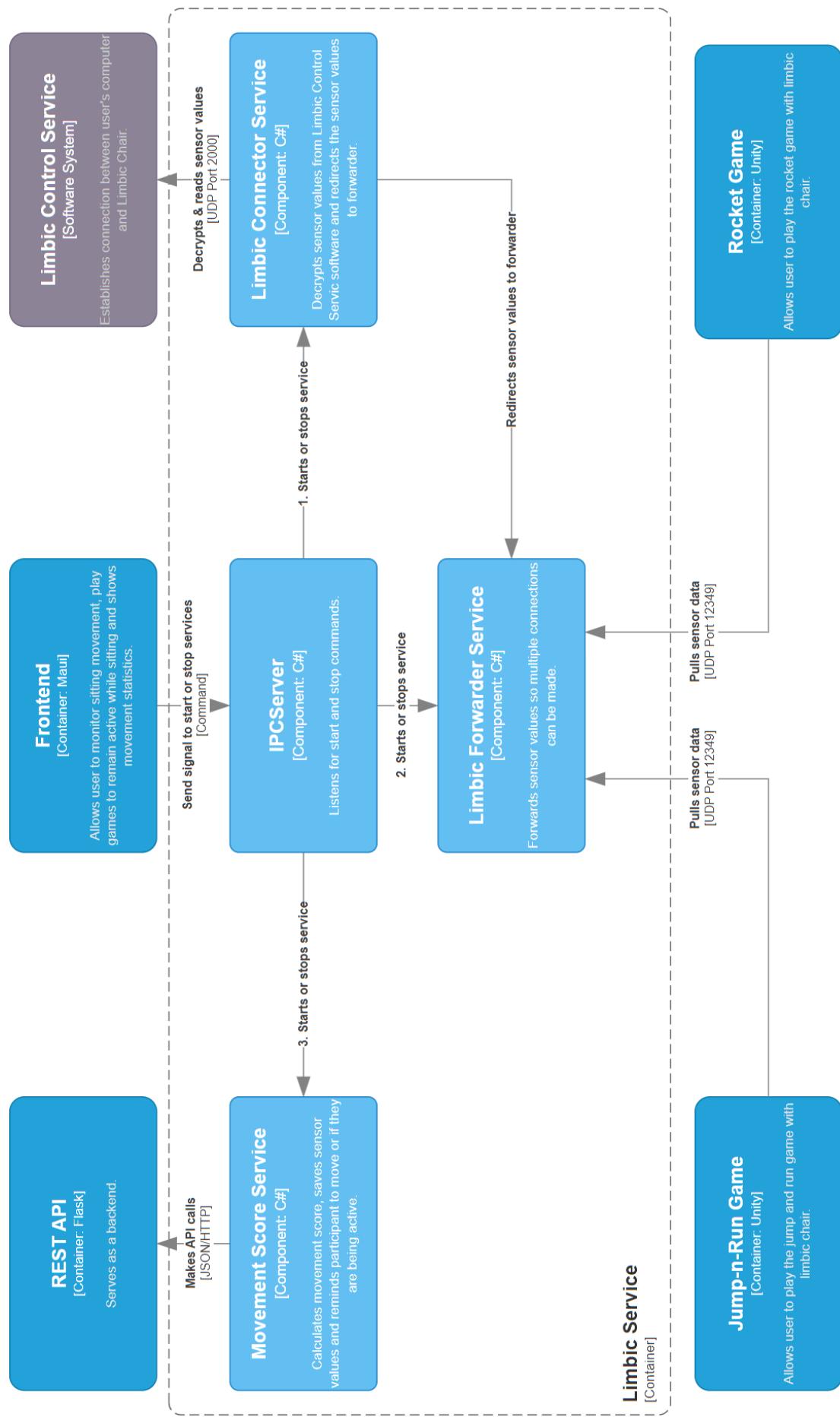


Abbildung 4.15: C4 Diagram Layer 3 - Limbic Service

#### 4.2.7 Movementscore

Im vorherigen Abschnitt wurde erklärt, wie der Movementscore mit dem Backend kommuniziert. In diesem Abschnitt wird die Funktionsweise des Movementscores behandelt und erläutert, wie die Berechnung zustande kommt.

Der Movementscore wird basierend auf den Bewegungen des Benutzers berechnet. Dabei werden die Pitch- und Yaw-Werte des Stuhls verwendet, da für diese Bewegungen aktive Anstrengung erforderlich ist. Die Roll-Werte werden ignoriert, da angenommen wird, dass für diese Bewegungen keine Anstrengung nötig ist.

Der Algorithmus zur Berechnung des Movementscores funktioniert wie folgt: Zu Beginn wird der Movementscore auf einen Wert von 1000 gesetzt. Die Methode iteriert über alle aktuellen Sensordaten und berechnet die Differenz zwischen den aktuellen und den letzten bekannten Werten für jeden Sensor. Diese Differenz wird als die Anzahl der Punkte betrachtet, die zur Berechnung des Movementscore verwendet werden.

Bevor diese Punkte jedoch dem Movementscore zugeordnet werden, werden die Differenzen mit einem Multiplikator versehen. Dadurch soll der Nutzer für "gute Bewegungen" besser belohnt werden. Laut Patrick Künzler, dem Ersteller des Stuhls, ist die beste Bewegung diejenige, bei der ein Bein nach unten geht, während das andere Bein nach oben gehalten wird. Diese Bewegung erhält den höchsten Multiplikator und die Punkte werden hierbei vervierfacht. Die nächstbeste Bewegung ist die, bei der beide Beine gleichzeitig nach unten oder nach oben gehen, dabei werden die Punkte verdoppelt. Alle anderen Bewegungen erhalten den Multiplikator 1. Nachdem diese Werte mit dem Multiplikator berechnet wurden, werden Sie dem Movementscore dazuberechnet.

```
double sensorDiff = Math.Abs(currentSensorValues[sensor] - lastSensorValues[sensor]);
double multiplier = 1;

if (sensor == "Lpitch" || sensor == "Rpitch")
{
    bool LpitchIncreasing = currentSensorValues["Lpitch"] > lastSensorValues["Lpitch"];
    bool RpitchIncreasing = currentSensorValues["Rpitch"] > lastSensorValues["Rpitch"];

    if (LpitchIncreasing && !RpitchIncreasing)
    {
        multiplier = 4;
    }
    else if (!LpitchIncreasing && RpitchIncreasing)
    {
        multiplier = 4;
    }
    else if (LpitchIncreasing == RpitchIncreasing)
    {
        multiplier = 2;
    }
}
```

Abbildung 4.16: Berechnung der Sensordaten mit der Bestimmung des Multipliers

Der Score soll jedoch nicht nur steigen, sondern keine Bewegung soll auch bestraft werden. Dazu wurde eine Decayfunction hinzugefügt welche folgendermassen aussieht.

$$\text{Movementscore} \times 0.99^{\frac{1}{3600 \cdot 4}}$$

Ursprünglich war die Funktion mit anderen Parametern geplant, da wir einen bestimmten Prozentsatz der Abnahme erreichen wollten, den wir nach 15 Minuten messen konnten. Dies erklärt die Multiplikation von 3600 (Sekunden in einer Stunde) mit 4, um die Abnahme über eine Viertelstunde (15 Minuten) zu berechnen.

Da der Movement Score jedoch beim Testen sehr schnell sank, wurde der Prozentsatz auf 0,99 angepasst, um eine langsamere und gleichmässigere Abnahme zu erzielen. Bei der Erstellung der Formel gingen wir davon aus, dass die Berechnung einmal pro Sekunde erfolgt. Zu diesem Zeitpunkt wurde jedoch noch nicht berücksichtigt, dass die Berechnung im Programm tatsächlich mehrmals pro Sekunde durchgeführt wird.

Die  $0.99^{\frac{1}{3600 \cdot 4}}$  in der Formel steht dafür, dass wir  $0.99^{\frac{1}{3600 \cdot 4}}$  Prozent des Movement Scores erhalten wollen, exponentiell reduziert über die Zeit. Diese Funktion scheint zwar zu stimmen, wenn man davon ausgeht, dass jede Iteration über eine Sekunde läuft. Wir wollten aber einen tieferen Wert und jede Iteration dauert viel kürzer als eine Sekunde. Das bedeutet, dass unsere Parameter letztlich nur durch Trial and Error ausgewählt wurden. Wir gingen beim testen des Movementscores von einem anfänglichen Movement Score von 2000 aus und stoppten die Zeit, wie lange es dauerte, bei keiner Bewegung wieder auf einen tiefen Wert von 500 zu kommen. Dies dauerte ungefähr eine Viertelstunde.

Wir testeten die Formel mit verschiedenen Parametern so lange, bis wir diese gewünschte Zeitspanne erreichten. Durch diese Anpassungen und Tests konnten wir sicherstellen, dass der Movement Score über eine realistische Zeitspanne abnimmt, was zu einer besseren Nutzererfahrung führte.

Die schlussendliche Formel für die Berechnung des Movementscores sieht nach jeder Iteration folgendermassen aus:

$$\text{TotalMovementScore} = \text{TotalMovementScore} \times \text{DecayRate} + \text{Difference} \times \text{Multiplier}$$

Dabei multiplizieren wir zu Beginn den Movementscore aus der vorherigen Iteration mit der DecayRate. Anschliessend wird die Differenz der Sensorwerte aus der letzten Iteration mit dem Multiplier addiert.

```
if (!lastSensorValues.ContainsKey(sensor))
{
    lastSensorValues[sensor] = currentSensorValues[sensor];
}

double sensorDiff = Math.Abs(currentSensorValues[sensor] - lastSensorValues[sensor]);
double multiplier = 1;

if (sensor == "Lpitch" || sensor == "Rpitch")
{
    bool LpitchIncreasing = currentSensorValues["Lpitch"] > lastSensorValues["Lpitch"];
    bool RpitchIncreasing = currentSensorValues["Rpitch"] > lastSensorValues["Rpitch"];

    if (LpitchIncreasing && !RpitchIncreasing)
    {
        multiplier = 4;
    }
    else if (!LpitchIncreasing && RpitchIncreasing)
    {
        multiplier = 4;
    }
    else if (LpitchIncreasing == RpitchIncreasing)
    {
        multiplier = 2;
    }
}

if (sensorDiff < movementTreshold)
{
    movementScore -= sensorDiff * multiplier;
}
else
{
    movementScore += sensorDiff * multiplier;
}

lastSensorValues[sensor] = currentSensorValues[sensor];
```

Abbildung 4.17: Berechnung des Movementscores

## 4.3 Zukunft der Architektur

### 4.3.1 Limbic Service Forwarder

Momentan kann die externe Limbic Control Service Software der Limbic Life AG nicht gleichzeitig mehrere Verbindungen aufbauen. Wenn bereits eine Verbindung zur Limbic Control Service Software von einer anderen Anwendung blockiert wird, erhalten andere Anwendungen keine Sensorwerte mehr. Limbic Life wurde über dieses Problem informiert und hat in einer Besprechung mitgeteilt, dass diese Funktionalität noch nicht in ihrer Software implementiert ist. Sie werden jedoch sicherstellen, dass ihre Software zukünftig in der Lage sein wird, mehrere Verbindungen gleichzeitig aufzubauen, sodass auch andere Anwendungen die Sensorwerte empfangen können. Sobald diese Funktionalität implementiert ist, wird der Limbic Forwarder nicht mehr benötigt und kann somit ausser Betrieb genommen werden.

### 4.3.2 MessagingQueue

Aktuell haben wir das Problem, dass im Backend nicht alle Sensordaten ankommen oder teilweise Verbindungsprobleme auftreten, wodurch keine Daten mehr gespeichert oder empfangen werden. Unser Backend arbeitet synchron und nicht asynchron. Daten werden alle zwei Sekunden an das Backend gesendet. Das Problem könnte durch die Implementierung einer MessageQueue gelöst werden. Eine MessageQueue würde eingehende Daten zwischenspeichern und sicherstellen, dass sie in der richtigen Reihenfolge und ohne Datenverlust verarbeitet werden, selbst wenn es temporäre Verbindungsprobleme gibt. Dies würde die Zuverlässigkeit und Robustheit unseres Systems erheblich verbessern und sicherstellen, dass keine Daten verloren gehen.

# Kapitel 5: Testing

## 5.1 Usability Testing

In diesem Abschnitt wird das Usability Testing ausgewertet. Dabei werden sowohl der Movementscore unserer Software als auch die Spiele getestet. Es wird untersucht, ob die Software bewegungsfördernd ist. Das Usability Testing basiert auf den Notizen, welche während dem Testing gemacht wurden und sich im Anhang befinden.

### 5.1.1 Game Testing

**Gaming** Die Spiele wurden getestet, indem man sie für die Nutzer startete und diese dann einfach spielen liess. Dabei sollten die Nutzer laut denken und alles aussprechen, was ihnen auffiel. Falls keine Kommentare von ihnen kamen, wurden Fragen zur Steuerung des Spiels, zur Bedienung (insbesondere, ob das Spiel selbstständig bedient werden kann) und zu allgemeinen Verbesserungsvorschlägen gestellt.

**Rocket Escape** In einem ersten Testlauf sollte herausgefunden werden, wie sich das Spiel spielen lässt und ob es angenehm zu spielen ist. Das Spiel wurde im ersten Durchlauf mit sechs Personen getestet. Zu Beginn war das Spiel so programmiert, dass die Bewegung der Rakete nur drei Bewegungen kannte. Es wurden Schwellenwerte eingeführt, welche bestimmte Bereiche markierten: Niedrige Werte entsprachen einer konstanten Bewegung nach unten, mittlere Werte keiner Bewegung und hohe Werte einer konstanten Bewegung nach oben. Dies schien zwar gut zu funktionieren, jedoch kam mehrfach das Feedback, dass man eine relative Steuerung zur Höhe der Rakete haben wollte. Das bedeutete, dass jeder Neigungswert einer bestimmten Höhe der Rakete zugewiesen werden sollte, um eine bessere Kontrolle zu ermöglichen. Abgesehen von diesem Hauptkriterium gab es keine weiteren Mängel. Das Spiel war für alle Spieler sehr intuitiv und sie konnten es selbstständig bedienen.

Im zweiten Testdurchlauf wurde die neue Steuerung mit vier anderen Probanden getestet. Die neue Steuerung kam bei den Probanden gut an. Es gab häufig das Feedback, dass eine gewisse Spannung im Rumpfbereich erforderlich sei, um die Kontrolle über die Rakete zu behalten. Dies sehen wir als positives Feedback und als unerwartete Förderung der Muskulatur. Da es in dieser Arbeit nicht um den medizinischen Einfluss geht, kann bezüglich der Rumpfförderung nur eingeschränkt gesagt werden, ob dies zutrifft und ob die neue Steuerung tatsächlich einen Einfluss hat. Auch im zweiten Testdurchlauf konnte die Applikation von allen Nutzern selbstständig bedient werden.

**Alien Run** Im Spiel Alien Run wurden ebenfalls zwei Testdurchläufe mit der gleichen Anzahl an Probanden durchgeführt. Während des ersten Tests gab es viel Feedback zum Spiel. In Bezug auf die Steuerung wurde kritisiert, dass die Bewegung nur zählt, wenn man eine vertikale Bewegung mit dem Bein macht. Hier wurde gewünscht, dass auch die horizontale Bewegung einbezogen wird. Hinsichtlich des Spiels wurde bemängelt, dass es am Anfang sehr frustrierend wirkt, weil man oft stirbt. Um diesem Problem entgegenzukommen, wurde entschieden, im ersten Teil des Levels einen Boden zu integrieren, damit man nicht gleich stirbt, wenn man den Sprung auf die nächste Plattform nicht schafft.

Ein weiteres Problem, das während des Usability-Testings festgestellt wurde, war, dass der Alien manchmal nicht starb, wenn er von der Plattform fiel, sich aber gleichzeitig auch nicht weiter bewegen konnte. Dies war ein Bug im Spiel, der dann behoben wurde. In Bezug auf die

Bedienung des Spiels war diese wieder intuitiv. Ein Verbesserungsvorschlag war, dass es nervig sei, dass man mit der Maus immer auf "Restart" klicken muss, wenn man stirbt. Dies war besonders ärgerlich, da man teilweise sehr schnell starb. Das Problem wurde behoben, indem der Restart-Button aus dem Spiel entfernt wurde und das Spiel automatisch neu startet, wenn man stirbt.

Im zweiten Durchlauf kam das Spiel besser an. Die neue Steuerung wurde nicht mehr kritisiert, und es gab generell keine Kritikpunkte mehr in Bezug auf das Spiel. Auffällig war jedoch, dass kein Spieler das Level beenden konnte. Jede Person starb davor, was jedoch nie zu Frustration führte. Für einen nächsten Anlauf wäre es eine Option, das Spiel leichter zu gestalten.

### 5.1.2 Active Sit

Die Software Active Sit wurde nur einmal von fünf Probanden getestet. Beim Testvorgang wurde den Nutzern erklärt, dass die von ihnen erzeugten Daten an den Server übermittelt werden. Sie hatten die Möglichkeit, ihre Statistiken einzusehen und Spiele direkt aus der Software heraus zu starten. Den Nutzern stand es frei zu entscheiden, ob sie die Spiele testen wollten oder nicht, sowie in welchem Umfang sie sich mit der Software beschäftigen möchten. Es wurde den Nutzern ausdrücklich mitgeteilt, dass sie ihre Arbeit wie gewohnt fortsetzen können und die Software nach eigenem Ermessen verwenden dürfen. Zudem wurden sie aufgefordert, die Software eine Stunde lang laufen zu lassen, um ausreichend Daten zu sammeln.

Bezüglich der Benutzeroberfläche bestand das Problem, dass beim Einloggen die Gross- und Kleinschreibung beachtet werden musste. Beim Registrieren musste daher darauf geachtet werden, wie die Gross- und Kleinschreibung gesetzt war. Dies führte zu Verwirrung auf dem Login-Bildschirm, und bei zwei von fünf Personen musste beim Einloggen Hilfe geleistet werden. Nach dem Login gelangten die Nutzer zur Statistikseite, und die Bedienung war unproblematisch. Alle Nutzer haben auf den Refresh-Button geklickt und konnten ihre Daten ohne Schwierigkeiten einsehen. Die Gaming-Seite konnte ebenfalls von allen geöffnet werden, und alle Nutzer konnten die Spiele eigenständig starten. Auch das Ausloggen war für niemanden ein Problem, und alle Nutzer konnten sich selbstständig ausloggen.

Bei der Betrachtung der Softwarefunktionalität traten während des Testings Probleme auf. Nutzer konnten die Spiele nicht spielen, wenn eine instabile Verbindung zum Server bestand. Obwohl der Movementscore weiterhin berechnet wurde, konnten die Spiele keine Daten vom Limbic Service empfangen, wodurch sie bei schlechter Verbindung nicht gestartet werden konnten.

In Bezug auf die nächste Iteration könnte die Software verbessert werden, indem Nutzer bei der Eingabe ihrer E-Mail-Adresse nicht mehr auf Gross- und Kleinschreibung achten müssen. Insgesamt betrachten wir das User Interface jedoch als Erfolg in Bezug auf die Benutzerfreundlichkeit.

## 5.2 Testauswertung

In diesem Abschnitt soll die Effektivität der Software bewertet werden. Dazu haben wir den Test wie folgt durchgeführt:

Zuerst liessen wir den Nutzer eine Stunde lang auf dem Stuhl sitzen. Während dieser Zeit sollte der Nutzer normal weiterarbeiten und sich im Stuhl so bewegen, wie es ihm gerade gefällt. Im Hintergrund lief eine Software, die uns von Limbic Life zur Verfügung gestellt wurde und die die Daten erfassende und in eine CSV-Datei schrieb. Dem Nutzer war es während dieser Zeit

nicht erlaubt, aufzustehen oder seine aktuellen Bewegungen einzusehen. Während dieser ersten Stunde lief ausschliesslich die Datenerfassungssoftware von Limbic Life, während unsere eigene Software Active Sit sowie die Spiele nicht verwendet wurden. Dies sollte testen, wie oft sich der Nutzer im Allgemeinen auf dem Stuhl bewegte.

In der zweiten Stunde wurde dem Nutzer unsere Software vorgestellt. Dabei wurden sowohl das Usability-Testing als auch die Effektivität der Bewegungen gleichzeitig getestet. Dem Nutzer wurde das Konzept des Movementscores erklärt, jedoch nicht, wie der Algorithmus funktioniert. Ebenso wurde nicht erläutert, wie die Einsicht in die Daten funktioniert, um den Usability-Test nicht zu beeinträchtigen. Anschliessend wurde die Software wieder eine Stunde lang verwendet. Der Nutzer hatte während dieser Zeit die Möglichkeit, die Spiele zu starten, um seinen Movementscore zu erhöhen. Es oblag jedem Nutzer selbst zu entscheiden, wie lange er die Spiele testen wollte.

Zur Auswertung der Daten wird der Movementscore als Leitfaden verwendet. Die Sensordaten werden dabei ignoriert, da sie während der Aufnahme mit Active Sit nur alle 2 Sekunden an den Server gesendet werden, während die Daten ohne Active Sit kontinuierlich erfasst werden. Dies kann zu Verfälschungen der Ergebnisse führen. Im Gegensatz dazu wird der Movementscore in beiden Tests kontinuierlich neu berechnet.

Da das Aufrufen der Spiele aufgrund von Netzwerkproblemen nicht immer zuverlässig funktionierte, werden in diesem Kapitel nur jene Spiele genauer analysiert, die ohne Probleme gestartet werden konnten. Die Ergebnisse der anderen Spiele zeigen ähnliche Muster, werden aber aufgrund der unvollständigen Funktionsweise nicht detailliert behandelt. Die entsprechenden Daten sind dennoch im Anhang verfügbar. Dieses Problem trat nur bei einem Probanden auf.

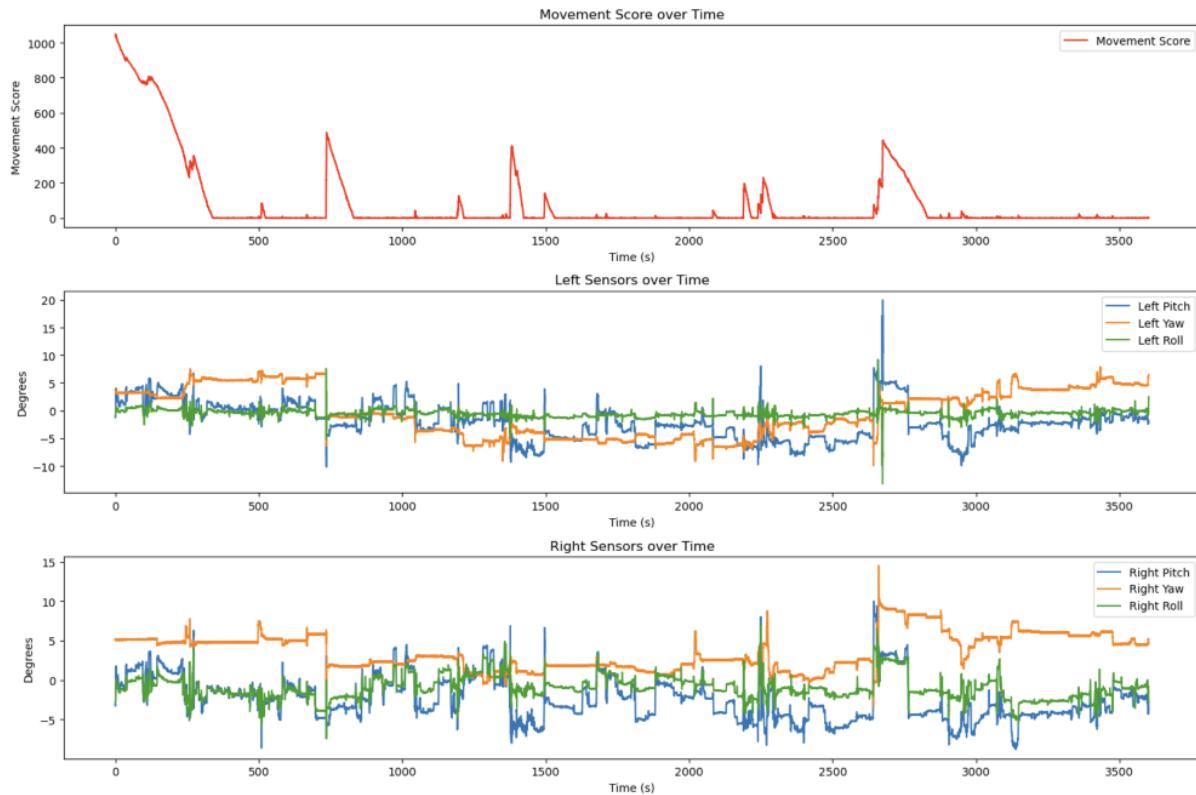


Abbildung 5.1: Auswertung Experiment Proband 1 ohne ActiveSit

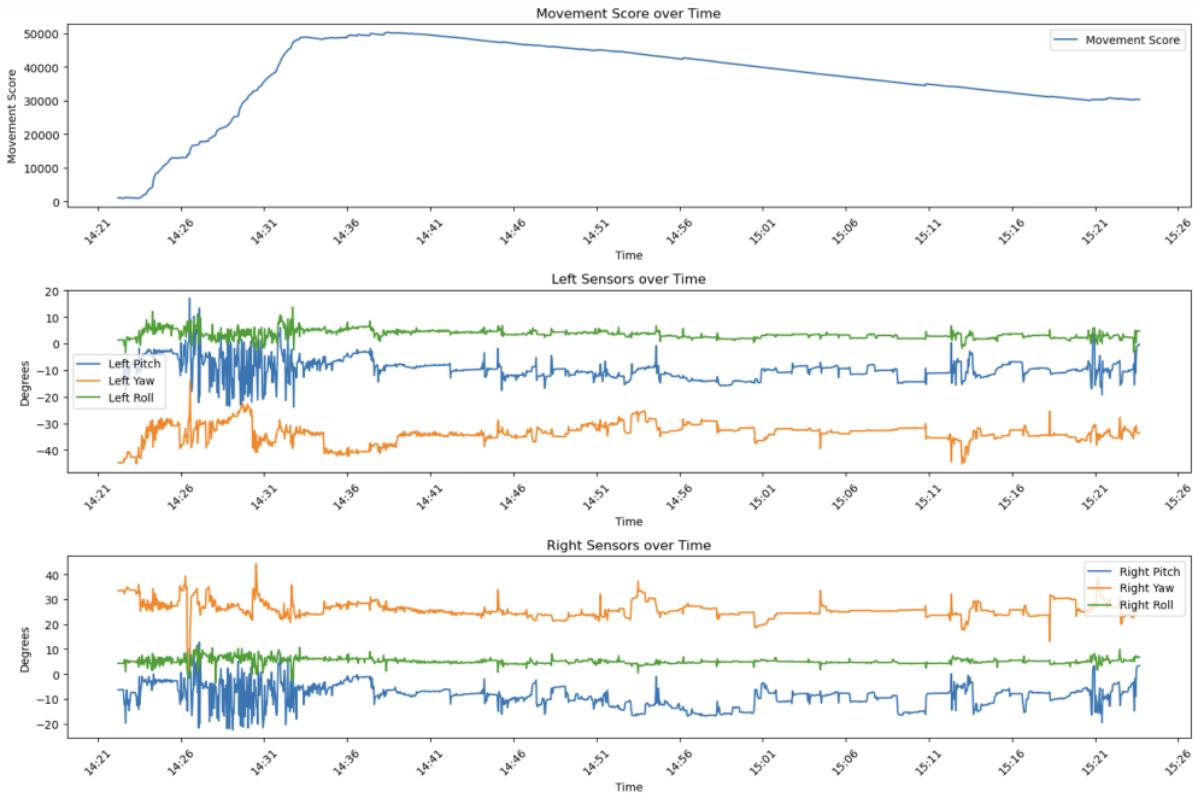


Abbildung 5.2: Auswertung Experiment Proband 1 mit ActiveSit

Der Movementscore des ersten Probanden erreichte im Test ohne ActiveSit (Abbildung 5.1) zwar einen Maximalwert von 1000, dieser ist jedoch als Startwert zu ignorieren und daher irrelevant. Während der ersten Stunde wurde der tatsächliche Maximalwert etwas über 400 erreicht, was dreimal während der Stunde vorkam. Ein weiterer Wert lag knapp bei 200. Diese Ergebnisse deuten darauf hin, dass der Proband während der ersten Stunde nur geringfügige Bewegungen ausführte. Dies könnte darauf zurückzuführen sein, dass er während dieser Zeit arbeitete und kein Bedürfnis verspürte, sich im Stuhl zu bewegen.

Beim Test von ActiveSit lässt sich in den ersten Minuten ein steiler Anstieg des Movementscores auf über 50.000 beobachten. Dies fällt in den Zeitraum, in dem der Proband die Spiele testete. Anschliessend zeigt sich ein kontinuierlicher Abfall des Movementscores. Nach dem Testen der Spiele begann der Proband, Musik zu hören und bewegte dabei sein Bein. Diese Bewegung war jedoch nicht ausreichend, um den Movementscore zu erhöhen, und führte stattdessen zu einem weiteren Absinken nach Beendigung des Softwaretests.

Im Feedback des Probanden wurde die Mobilität des Stuhls positiv bewertet. Insbesondere wurde die Möglichkeit, die Beine freier zu bewegen im Vergleich zu einem gewöhnlichen Stuhl, als angenehm empfunden. Das Interesse an dem Spiel war hoch, da es eine neue Art des Spielens darstellte. Jedoch zeigte sich, dass trotz der Mobilität des Stuhls kein Interesse daran bestand, sich während der Arbeit aktiv zu bewegen.



Abbildung 5.3: Auswertung Experiment Proband 2 ohne ActiveSit

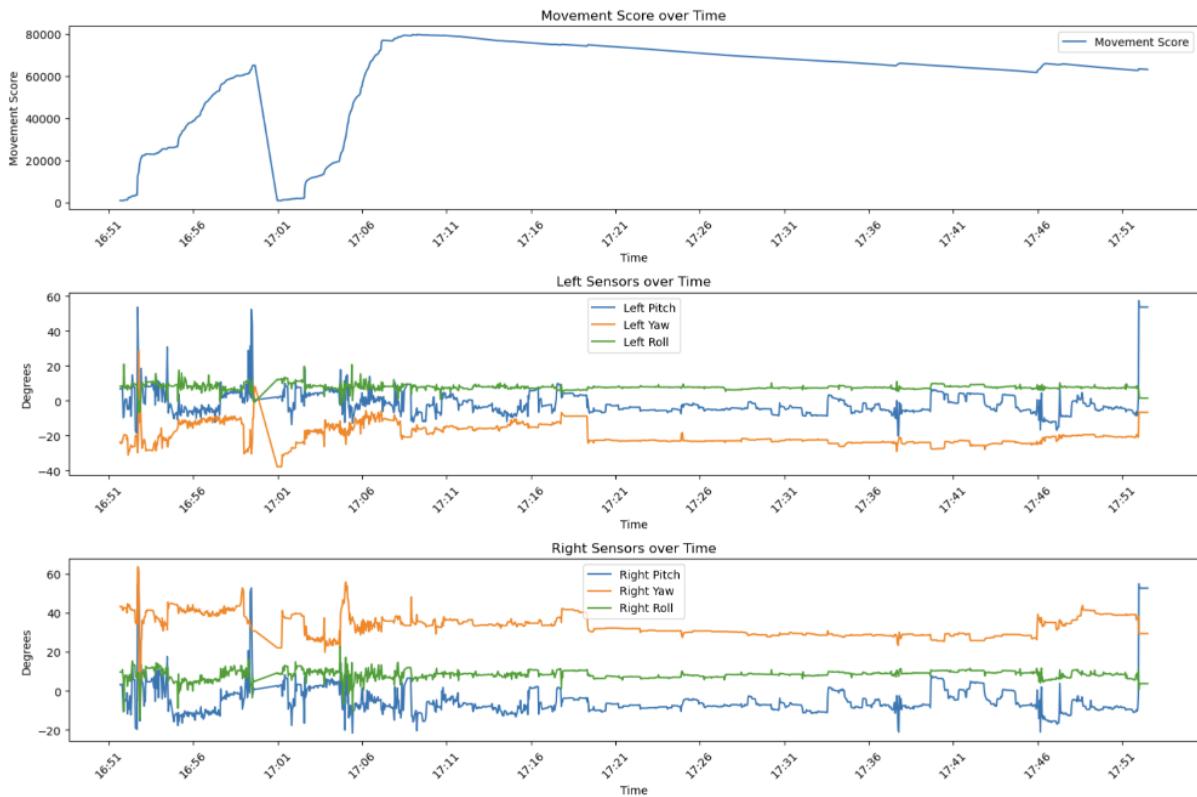


Abbildung 5.4: Auswertung Experiment Proband 2 mit ActiveSit

Beim zweiten Probanden zeigt die Messung ohne ActiveSit (Abbildung 5.3), dass der Movementscore die meiste Zeit bei 0 liegt. Erst in den letzten paar Minuten ist eine leichte Bewegung zu erkennen, die zu einem Höchstwert von etwa 500 führt. Da 500 ein sehr niedriger Wert ist, lässt sich schlussfolgern, dass im Grossen und Ganzen keine nennenswerten Bewegungen stattgefunden haben.

Beim Test mit ActiveSit lässt sich beobachten, dass der Movementscore zu Beginn der Softwareanwendung stark ansteigt. In dieser Phase versuchte der Proband, den Movementscore zu testen und herauszufinden, wie er ihn erhöhen kann. Es war eine deutliche Begeisterung erkennbar, den Movementscore hoch zu halten und einen möglichst hohen Wert zu erreichen. Anschliessend wurde der Wert auf 1000 zurückgesetzt, da der Movementscore bei jedem Ab- und Wiederanmelden auf 1000 zurückgesetzt wird. Der Proband loggte sich aus, um die Spiele zu testen. Aufgrund von Netzwerkproblemen konnten die Spiele zunächst nicht gestartet werden. Nachdem er sich erneut angemeldet und die Software neu gestartet hatte, funktionierten die Spiele und der Movementscore stieg erneut stark an. Nach der Spieldauer arbeitete der Proband konzentriert weiter. Um 17:46 Uhr zeigt sich ein erneuter Anstieg des Movementscores. Dies lässt sich darauf zurückführen, dass die Arbeitsaufgaben zu diesem Zeitpunkt weniger anspruchsvoll waren und der Proband versuchte, leichte Bewegungen in seinen Arbeitsalltag zu integrieren.

Im Feedback des Probanden äusserte er, dass er nach 40 Minuten mit Rückenschmerzen kämpfen hatte. Nach der ersten Stunde des Tests wurde eine Pause eingelegt, da das Sitzen als anstrengend empfunden wurde. Zudem beschrieb er anfängliches Unbehagen, da es sich nicht um einen "normalen Stuhl" handelte. Der Proband berichtete ausserdem, dass er bei starker Konzentration dazu neigte, die Bewegung mit dem Stuhl zu vernachlässigen, jedoch das Programm dennoch ein Bewusstsein für Bewegung schuf.

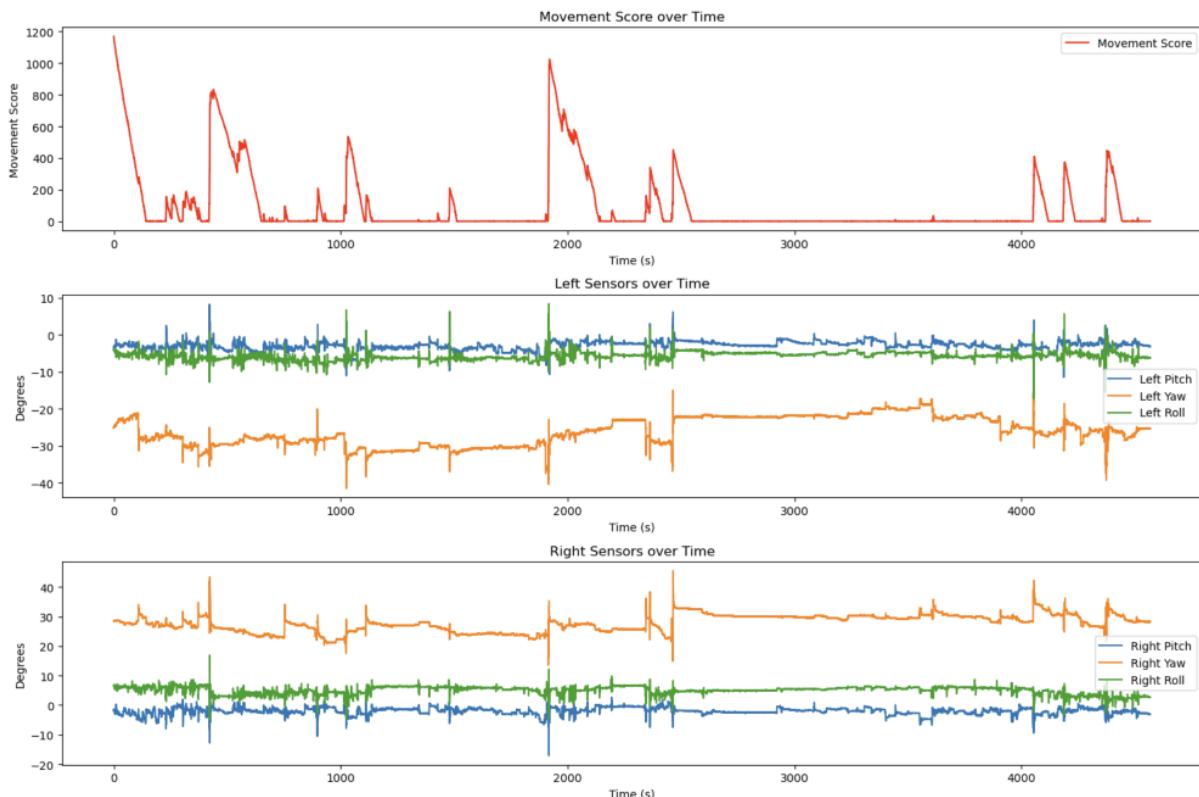


Abbildung 5.5: Auswertung Experiment Proband 3 ohne ActiveSit

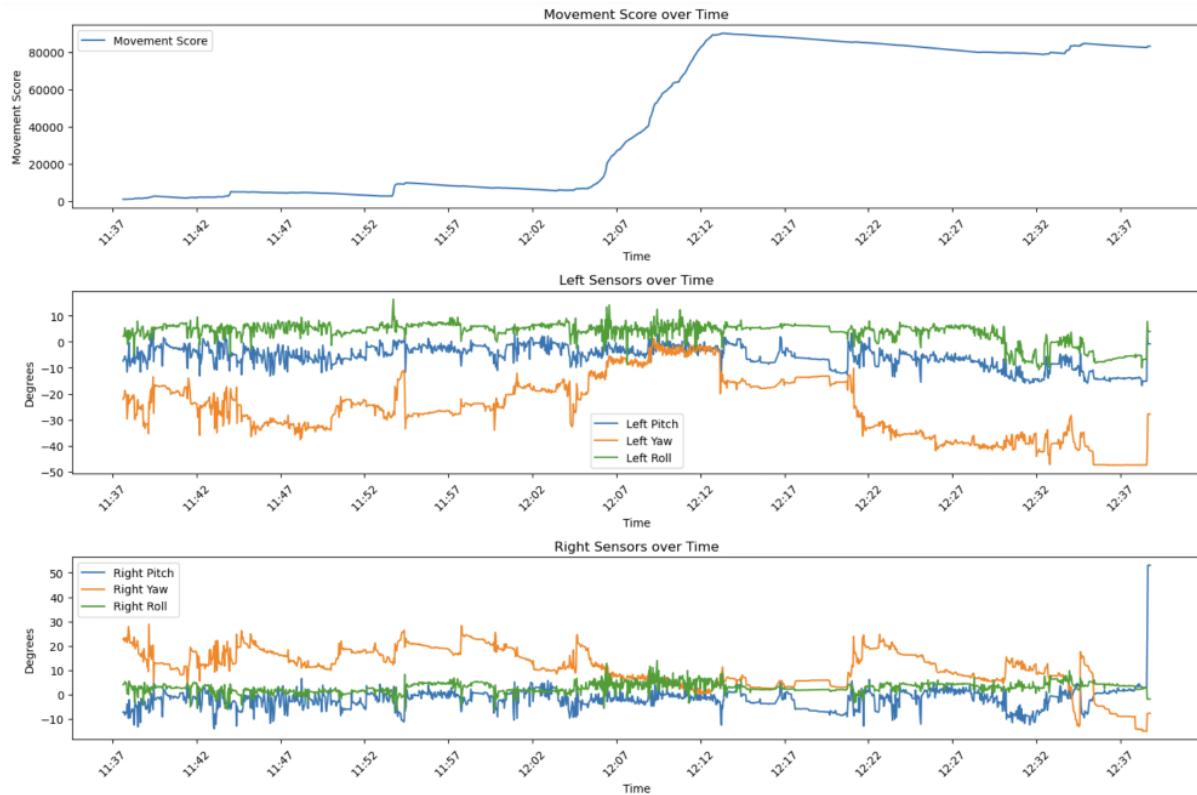


Abbildung 5.6: Auswertung Experiment Proband 3 mit ActiveSit

Beim dritten Probanden lässt sich beim Testen ohne ActiveSit beobachten, dass der Movementscore zweimal hohe Werte erreicht: einmal 800 und einmal 1000. Zudem sind im Vergleich zu den vorherigen Probanden häufiger kurze Bewegungen zu erkennen, die den Movementscore erhöhen. Die Phasen mit einem Movementscore von 0 sind kürzer. Da der Wert jedoch auch hier nie über 1000 steigt, lässt sich weiterhin schliessen, dass die Bewegung insgesamt nicht ausreichend war.

Beim Testen mit ActiveSit lässt sich erkennen, dass es zu Beginn der Software einen kontinuierlichen Anstieg des Movementscores gibt. Dies ist darauf zurückzuführen, dass der Nutzer sah, wie seine Bewegungen die Statistik beeinflussten, und versuchte, eine konstante Bewegung aufrechtzuerhalten. Diese konsequente Bewegung wurde dann durch den Movementscore belohnt, der kontinuierlich anstieg. Der starke Anstieg des Movementscores zeigt die Phasen, in denen die Spiele getestet wurden. Das Absinken des Movementscores danach kann darauf zurückgeführt werden, dass wieder konzentrierte Arbeit verrichtet wurde, bei der die Bewegung vernachlässigt wurde. Da der Maximalwert des Movementscores über 80.000 liegt, lässt sich sagen, dass über den Verlauf der Stunde sicherlich ausreichend Bewegung stattfand.

Im Feedback wurde erwähnt, dass die Statistik dazu motivierte, sich aktiver zu bewegen. Es bestand ein Ehrgeiz, den Movementscore auf einem konstanten Niveau zu halten oder sogar zu erhöhen. Obwohl die Spiele getestet wurden, war die Haltung gegenüber den Spielen eher neutral.

**Fazit** Das Testen der ActiveSit-Software hat gezeigt, dass durchaus Potenzial besteht, die Bewegung durch eine anzeigenende Statistik und interaktive Spiele zu fördern. Die Statistikanzeige mittels Movementscore sorgte für grosse Begeisterung und spiegelte sich entsprechend im Movementscore wider. Der Movementscore war im Test mit ActiveSit immer höher als im Test ohne ActiveSit. Allerdings muss beachtet werden, dass gleichzeitig mit dem Test der Beweglichkeit auch ein Usability-Test stattfand. Das bedeutet, dass alle Teilnehmer die Software zum ersten Mal sahen und testen wollten, was sie genau kann. Dies könnte die Ergebnisse insofern verfälschen, als der Fokus fast zu stark auf die Software gerichtet war und nicht auf die eigentliche Arbeit, die die Probanden erledigen sollten.

Ein weiterer Kritikpunkt, der während des Testens auftrat, betraf den Movementscore. Im Test ohne ActiveSit funktionierte der Score zuverlässig und befand sich stets im erwarteten Bereich. Beim Test mit ActiveSit stiegen die Werte jedoch stark an und sanken kaum. Als nächster Schritt wäre es interessant, andere Algorithmen zu testen, um einen weniger starken Unterschied festzustellen. Personen, die sich häufig im Stuhl bewegen, sollten dennoch weiterhin angemessen belohnt werden. Hier müsste ein besser ausbalancierter Algorithmus gefunden oder die Parameter entsprechend angepasst werden.

# Kapitel 6: Qualitätsmassnahmen

## 6.1 Organisatorisches

### 6.1.1 Definition of Done

Ein Feature oder eine User Story gilt als "done", wenn folgende Kriterien erfüllt sind:

- Der Code wurde geschrieben und alle Änderungen wurden im Git dokumentiert.
- Der Code wurde von einem anderen Entwickler überprüft und Feedbacks wurden eingepflegt.
- Neue Tests wurden geschrieben, welche die Funktionalität des Features sinnvoll überprüfen.
- Alle Tests wurden erfolgreich ausgeführt und die Codeabdeckung ist angemessen.
- Alle notwendigen Dokumentationen wurden aktualisiert.

## 6.2 Tools zur Qualitätskontrolle

### 6.2.1 Linter

Im Produkt Repository werden die Programmiersprachen Python für das Backend, C# für das Frontend und für die Spielentwicklung verwendet. Aufgrund der zwei verschiedenen Programmiersprachen, wird Mega-Linter verwendet. Mega-Linter bietet mehrere Linter an und kann in einem Durchlauf alle Dateien unabhängig von der verwendeten Programmiersprache analysieren.

Für Python wird Flake8 verwendet, welches einheitliche Codingstandards sicherstellt und Probleme im Code frühzeitig erkennen kann.

Für C# wird der Linter Roslynator verwendet, der Fehlerprüfung und Verbesserungsvorschläge. Als Formatter wird CSharfier eingesetzt, der ebenfalls für Clean Code sorgt. Beide Tools sind in Visual Studio verfügbar, was zusätzliche Sicherheit bietet.

Nach jedem Merge Request und Scan wird ein Bericht erstellt. Der zeigt, welche Linter eingesetzt wurden, wie viele Dateien analysiert und welche Fehler gefunden wurden.

### 6.2.2 Code Coverage

Für die Testabdeckung im Backendbereich kommt das Tool Pytest zum Einsatz, um die Anwendung gründlich zu testen. In der CI/CD-Pipeline wird ein Code-Coverage-Bericht erstellt. Dieser Bericht gibt einen klaren Überblick darüber, welche Teile des Codes durch die Tests abgedeckt werden und welche nicht. Er wird automatisch generiert und als Artefakt in Gitlab gespeichert. Die Berichte werden im HTML-Format zur Verfügung gestellt und können im Browser angezeigt und analysiert werden. Der Bericht ist im Anhang zu finden.

## 6.3 CI/CD

### 6.3.1 Dokumentation Repository

Das Dokumentations-Repository verfügt über keine CI/CD-Pipeline, da die Dokumentation über Overleaf erstellt wird. Mithilfe von Overleaf kann die Dokumentation kollaborativ bearbeitet werden. Zudem ist Overleaf mit unserem GitHub Dokumentations Repository synchronisiert. Falls die Overleaf-Website nicht verfügbar ist, kann auf die Dokumentation über GitHub zugegriffen und lokal daran gearbeitet werden.

### 6.3.2 Software Repository

Das Software-Repository ist in zwei Branches aufgeteilt. Der Main-Branch wird für stabile Releases der Applikation verwendet, während der Staging-Branch für tägliche Änderungen genutzt wird. Neue Funktionen werden über Feature-Branches implementiert, die mit "feature/" gekennzeichnet sind. Die daraus resultierenden Merge Requests werden von anderen Teammitgliedern überprüft. Die CI/CD Pipeline dockerisiert eine PostgreSQL-Datenbank und einen Flask-Server, der als REST API fungiert. Die Codebasis wird in der CI/CD-Pipeline zunächst mit Mega-Linter überprüft. Anschliessend werden die Backend Unit Tests ausgeführt.

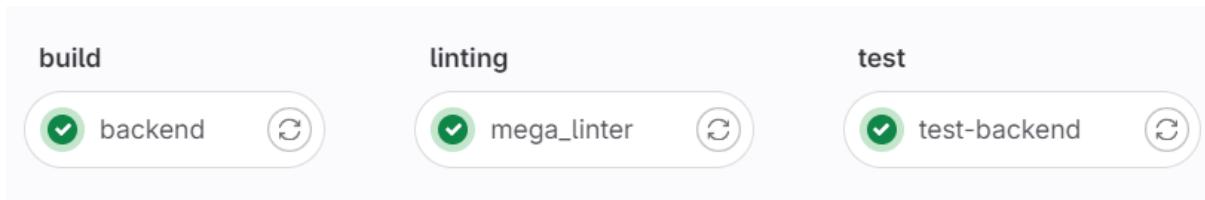


Abbildung 6.1: CI/CD Pipeline

## 6.4 Teststrategie

### 6.4.1 Unit Testing

Die Unit Tests für das Flask Backend werden entweder automatisch in einer GitLab CI/CD Pipeline ausgeführt oder manuell über ein Makefile. Das Ziel dabei ist, für alle Controller entsprechende Tests zu entwickeln, um nicht nur die korrekte Speicherung der Daten zu gewährleisten, sondern auch sicherzustellen, dass diese Daten richtig im Frontend dargestellt werden.

### 6.4.2 End User Tests

Um die Spiele und ActiveSit zu testen wurde beschlossen End User Tests durchzuführen. Dabei findet für ActiveSit ein Usability Tests statt. Die Spiele werden fortlaufend getestet und zum Schluss findet auch bei den Spielen gemeinsam mit ActiveSit ein Usability testing statt.

**End User Tests Gaming** Die Spiele werden so getestet dass man einen Prototyp erstellt und sobald es eine spielbare Version des Spieles gibt wird diese Probanden gegeben um das Spiel zu testen. Dabei sollen die Spieler ihr Feedback geben und sagen was Sie am Spiel stört. Ziel des fortlaufenden Testings ist eine angenehmes Spielerlebnis zu garantieren.

## **Teil III**

# **Projekt Dokumentation**

# Kapitel 7: Projektplanung

## 7.1 Organisation und Ressourcen

Hier werden die Ressourcen, welche uns für dieses Projekt zur Verfügung stehen, beschrieben. Dazu gehört der Zeitliche Rahmen, die Kosten und die Personas.

### 7.1.1 Zeitlicher Rahmen

Das Projekt begann am 19. Februar 2024 und dauert bis Freitag, den 14. Juni 2024. Während dieser Zeitspanne finden Projektreviews mit dem Betreuer Prof. Dr. Markus Stolze statt. Bei den Reviews werden die Fortschritte vorgestellt und das weitere Vorgehen wird besprochen. Der Aufwand des Projektes soll dabei 21,2 Stunden pro Woche und Person betragen.

### 7.1.2 Kosten

Da es sich hierbei um ein Schulprojekt handelt, entstehen weder Personal- noch Ressourcenkosten. Die benötigte Soft-/Hardware wird sowohl von den Teilnehmenden als auch von der Schule zur Verfügung gestellt.

### 7.1.3 Personas

In diesem Abschnitt werden die Mitarbeiter des Projektes vorgestellt:

**Name:** Kevin Pfister

**Studium:** Informatikstudium im 8. Semester

**Arbeitserfahrung:** Hostpoint Customer Care Support, Quereinsteiger

**Name:** Nicolas Gattlen

**Studium:** Informatikstudium im 8. Semester

**Arbeitserfahrung:** Lehre in der Systemtechnik. Mehrere Jahre Arbeitserfahrung in der Systemtechnik + Cybersecurity

### 7.1.4 Arbeitsweise

Das Projekt wird nach der agilen Arbeitsweise durchgeführt. Aufgrund der unterschiedlichen Stundenpläne kann kein klar definiertes agiles Framework eingehalten werden. SCRUM+ wird jedoch als Orientierungshilfe verwendet.

### 7.1.5 Meetings

Es finden wöchentlich zwei Meetings statt:

- Betreuungsmeeting jeweils am Montag um 15:00 Uhr
- Teammeetings jeweils Dienstag um 13:00 - 14:00 Uhr

In Teammeetings werden die getätigten Arbeiten, aufgetretenen Probleme und anstehenden Aufgaben besprochen. Die Aufgaben werden untereinander gleichmäßig aufgeteilt. Das Meeting mit dem Betreuer dient dazu, den Projektfortschritt zu zeigen sowie das weitere Vorgehen zu besprechen. Nur Meetings mit dem Betreuer werden protokolliert und können im Abschnitt Meeting Protokoll eingesehen werden.

## 7.2 Roadmap

Für die Gesamtplanung des Projekts wird der Rational Unified Process (RUP) verwendet. Das Projekt wird in vier Phasen unterteilt, wobei jede Phase mehrere Iterationen (Sprints) haben kann. Eine Iteration dauert eine Woche. Eine detaillierte Roadmap inklusiv der definierten Arbeitspaktete pro Phase ist im Anhang abgelegt.



Abbildung 7.1: Roadmap

Tabelle 7.1: Daten und Dauer der einzelnen Phasen

Phase	Startdatum	Enddatum	Dauer (Tage)
<b>Inception</b>	19.02.2024	03.03.2024	14
<b>Elaboration</b>	04.03.2024	07.04.2024	35
<b>Construction</b>	08.04.2024	30.05.2024	53
<b>Transition</b>	31.05.2024	14.06.2024	15

### 7.2.1 Phasen

In diesem Abschnitt werden die einzelnen Phasen beschrieben. Dabei werden die Hauptaufgaben der jeweiligen Phase definiert. Zum Schluss jeder Phase werden Meilensteine gesetzt und kontrolliert. Diese dienen dazu, den aktuellen Fortschritt zu überprüfen und einen Überblick darüber zu bekommen, wo wir stehen.

#### Inception

In der Inception-Phase sollen alle organisatorischen Aspekte so weit bearbeitet werden, dass die Arbeit danach ohne organisatorische Unterbrechungen fortgesetzt werden kann. Zudem ist geplant, sich in der ersten Woche mit der Firma Limbic Life AG zu treffen, um die Funktionen des Limbic Chairs kennenzulernen und Ideen für das Projekt zu sammeln. Ein weiteres Ziel ist es, alle benötigten Softwareprogramme auf unseren Laptops zu installieren, damit beim Start des Projekts keine Zeit verloren geht. In der zweiten Woche geht es darum, die Aufgabenstellung zu definieren, eine grobe Planung des Projekts zu erstellen, wichtige Meilensteine festzulegen und mögliche Risiken aufzulisten.

#### Elaboration

In einem ersten Schritt soll nach geeigneter wissenschaftlicher Literatur gesucht werden, die Informationen über gesundes, aktives und statisches Sitzen mit ergonomischen Stühlen vergleicht. Das Ziel in dieser Phase ist es, geeignete Technologien und Tools zu finden, die sowohl mit dem Limbic Chair als auch mit Windows kompatibel sind. Zugleich sollen Wireframes für die Benutzeroberfläche der Applikation erstellt werden, zusammen mit einem User Flow Diagramm. Dieses Diagramm beschreibt den Aufbau der Benutzeroberfläche und die einzelnen Funktionen der Applikation. Es dient als Orientierungshilfe für die Implementierung in der Konstruktionsphase. Zusätzlich soll ein Experiment durchgeführt werden, bei dem die Bewegungsdaten des

Benutzers auf dem Limbic Chair gesammelt werden, um zu erkennen, wie aktives und statisches Sitzen unterschieden werden kann.

### **Construction**

Während dieser Phase werden die Use Cases umgesetzt und die zwei Spiele entwickelt. Zudem wird ein Minimum Viable Product (MVP) präsentiert, um zu demonstrieren, dass die Use Cases und Spiele erfolgreich implementiert wurden. In der letzten Woche dieser Phase werden Endbenutzertests durchgeführt, bei denen die Benutzer die Applikation mit dem Limbic Chair testen und Feedback geben können. Auch während der gesamten Phase werden die Benutzer kontinuierlich gebeten, die Software und Spiele weiter auszuprobieren. Ihr Feedback wird genutzt, um nachträglich Verbesserungsvorschläge umzusetzen.

### **Transition**

Die Transition-Phase erstreckt sich über zwei Wochen, wobei die erste Woche als Puffer dient, der im Falle eines unerwarteten Bedarfs während der Konstruktionsphase genutzt werden kann. In der letzten Woche liegt der Fokus auf der abschliessenden Dokumentation, um sicherzustellen, dass sie für die Abgabe bereit ist.

### 7.2.2 Meilensteine

Im Projekt werden die folgenden sechs Meilensteine definiert. Zudem werden in der untenstehenden Tabelle alle Artefakte/Ziele aufgeführt, die bis zum Enddatum des entsprechenden Meilensteins abgearbeitet werden.

Tabelle 7.2: Meilensteine

Meilenstein	Quality Gates	Termin
M1 - Projektplanung	<ul style="list-style-type: none"> <li>Es müssen mindestens 3 funktionale Anforderungen definiert und abgenommen werden.</li> <li>Bis zum 03.03.2024 muss ein Use Case Diagramm erstellt und beschrieben sein.</li> <li>Es müssen mindestens 3 Risiken aufgelistet und eine Risikoanalyse durchgeführt werden, die bis zum 03.03.2024 dokumentiert sein muss.</li> </ul>	03.03.2024
M2 - Architektur	<ul style="list-style-type: none"> <li>Bis zum 11.03.2024 muss ein Domain Model erstellt und dokumentiert sein.</li> <li>Bis zum 17.03.2024 müssen mindestens 3 Benutzer auf dem Limbic Chair sitzen und die Bewegungsdaten ausgewertet sein.</li> <li>Bis zum 24.04.2024 muss ein Algorithmus gefunden werden um die Bewegung des Benutzers zu monitoren.</li> <li>Bis zum 31.03.2024 muss für die Benutzeroberfläche Wireframes und ein User Flow Diagramm erstellt sein.</li> </ul>	31.03.2024
M3 - Prototyp	<ul style="list-style-type: none"> <li>Bis zum 07.04.2024 muss eine Entwicklungsumgebung eingerichtet sein, damit in der Konstruktionsphase sofort entwickelt werden kann.</li> </ul>	07.04.2024
M4 - MVP	<ul style="list-style-type: none"> <li>Bis zum 23.05.2024 müssen alle Use-Cases sowie die technischen Lösungsansätze dokumentiert sein.</li> </ul>	23.05.2024
M5 - Endbenutzertests	<ul style="list-style-type: none"> <li>Es werden mit fünf Personen Endbenutzertests durchgeführt.</li> </ul>	30.05.2024
M6 - Abgabe	<ul style="list-style-type: none"> <li>Bis zum 14.06.2024 müssen die Dokumentation sowie der Softwarecode abgeschlossen und abgegeben sein.</li> </ul>	14.06.2024

## 7.3 Issue Management

Als Issue-Management-Tool wird Jira verwendet, welches einfach zu bedienen ist und bei dem sich alle Funktionen und Methoden auf einer Plattform befinden. Außerdem soll das Projekt eine definierte Struktur aufweisen.

### 7.3.1 Issues

Im Jira gibt es verschiedene Typen von Issues, darunter Story, Epic, Task und Subtask. Im Folgenden wird beschrieben, wie diese Typen in diesem Projekt verwendet werden.

Tabelle 7.3: Beschreibung der verschiedenen Typen von Issues

Issuearten	Beschreibung
Epics	Epics werden als Phasen des Gesamtprojekts dargestellt.
Story	Stories werden verwendet, sobald sich eine Anforderung oder eine Funktion auf eine spezifische Benutzeraktion oder ein Benutzerziel fokussiert.
Task	Tasks werden verwendet, wenn es sich um allgemeine Aufgaben handelt.
Subtask	Falls eine Story oder Task weiter unterteilt werden muss, kann ein Subtask dafür erstellt werden.

Die Issues in Jira werden hierarchisch strukturiert. In diesem Projekt werden Stories und Tasks einem Epic zugeordnet. Subtasks hingegen werden ausschließlich Stories und Tasks zugewiesen.

### 7.3.2 Components

Jeder Issue wird einem Component zugeordnet, der dazu dient, Aufgaben zu gruppieren. Mit anderen Worten kann ein Component als Tag betrachtet werden. Nachfolgend sind die definierten Components aufgeführt. Die Liste wird regelmäßig aktualisiert. Zusätzlich werden die Components für die Zeitauswertung benötigt.

### 7.3.3 Jira Kanban Board

Das Jira Kanban Board dient als Hilfestellung, um den Zustand und den Fortschritt eines Issues / Tasks zu überwachen, sowie auch die Erstellung von Tasks jeder Iteration vom Teammeetings.

## 7.4 Risikoanalyse

### 7.4.1 Risikomatrix

Tabelle 7.4: Risikomatrix

Auswirkung Wahrscheinlichkeit	Niedrig - 1	Mittel - 2	Hoch - 3	Sehr hoch - 4	Kritisch - 5
Sehr Wahrscheinlich - 5	(leer)	(leer)	R4	(leer)	(leer)
Wahrscheinlich - 4	(leer)	R2, R4	R1	(leer)	(leer)
Möglich - 3	(leer)	(leer)	(leer)	(leer)	R5
Unwahrscheinlich - 2	(leer)	(leer)	(leer)	(leer)	(leer)
Unmöglich - 1	(leer)	(leer)	R3	(leer)	(leer)

### 7.4.2 Risiken

In dieser Tabelle werden alle möglichen Risiken die im Projekt auftreten können aufgelistet.

Tabelle 7.5: Risiken

Nr.	Risiko	Wahrscheinlichkeit	Auswirkung
R1	Probleme beim Verbinden des Stuhles mit dem Computer	4	3
R2	Erstellung eines Interaktiven Spiels	4	2
R3	Testing der Software bei End-User	1	3
R4	Vereinbarung der zu benutzenden Technologien	4	2
R5	Störung bei Versuch des Aufbau von mehreren Verbindungen	3	5

### 7.4.3 Risikominderung

In diesem Abschnitt wird beschrieben, wie die oben genannten Risiken vermindert werden können.

#### **R1: Probleme beim Verbinden des Stuhles mit dem Computer (Behoben: 04.03.204)**

Ein Risiko welches besteht ist die Verbindung mit dem Computer. Stand jetzt sind wir nur in der Lage den Stuhl mit Kabel zu verbinden. Jedoch ist auch dies nicht immer erfolgreich. Theoretisch besteht ebenfalls die Möglichkeit, den Stuhl Wireless zu verbinden, dies gelang uns jedoch nie und endete immer mit einem Fehler.

Das Risiko wird vermindert indem wir im stetigen Austausch mit der Firma Limbic Chair sind und Sie uns Tipps geben wie das Verbinden mit dem Stuhl sauber funktioniert.

#### **R2: Erstellen eines interaktiven Spiels (Behoben: 13.05.2024)**

Da wir beide noch nie ein Spiel erstellt haben, fällt dieses Risiko unter einer grossen Wahrscheinlichkeit. Die Auswirkungen sind jedoch nicht so gross, da das Spiel nicht komplex sein muss und wir genügend Zeit einplanen können, um uns einzulesen.

Die Massnahmen die helfen sollen das Risiko unter Kontrolle zu halten ist die Komplexität des Spiels tief zu halten und genug Zeit für die Entwicklung einzuplanen.

#### **R3: Testing der Software beim End User (Behoben: 27.05.2024)**

Ein Risiko könnte beim Testen bestehen, da wir zehn Personen finden müssen, die unseren Stuhl testen wollen. Da dieser Stuhl ein einzigartiges Design hat und das Sitzerlebnis speziell ist, denken wir, dass es ein geringes Risiko ist, nicht genug Personen dafür zu finden. Die Auswirkungen wären jedoch hoch, wenn wir nicht genügend Tester finden, da wir dann kein Feedback zu unserer Software hätten.

#### **R4: Vereinbarung der zu benutzenden Technologien (Behoben: 29.04.2023)**

Da noch nie eine Software in dieser Richtung gebaut wurde, stellt sich die Wahl der zu wählenden Technologie eher schwierig heraus und tritt mit grosser Wahrscheinlichkeit auf.

Um dem Risiko entgegenzukommen ist die Elaboration Phase dazu gedacht, sich über Technologien und Architektur Gedanken zu machen. Diese kann sich zwar während dem Projekt noch ändern, jedoch soll es als Anhaltspunkt für den Beginn der Construction Phase einen Überblick geben.

#### **R5: Störung bei Versuch des Aufbau von mehreren Verbindungen (Teilweise behoben)**

(Stand: 8. Mai 2024) In der Entwicklungsphase ist das Problem aufgetreten, dass die Limbic Controller Software nicht in der Lage ist, gleichzeitig mehrere Verbindungen aufzubauen. Dies kann dazu führen, dass, wenn bereits eine Verbindung zum Limbic Controller Software von einer anderen Anwendung blockiert wird, andere Anwendungen keine Sensorwerte mehr erhalten. Als Zwischenlösung werden die Sensorwerte lokal auf dem Client an einen anderen Port weitergeleitet. Dadurch können andere Anwendungen auf diese zugreifen und die Sensorwerte empfangen. Das Problem konnte bis zum Ende der Arbeit nicht komplett behoben werden. Der Grund dafür ist die Abhängigkeit vom Netzwerk. Dennoch ist es aktuell möglich, mehrere Verbindungen mit dem Stuhl herzustellen.

#### 7.4.4 Risikoverlauf

Alle zwei Wochen wurden die Risiken neu evaluiert und der Risk Score aktualisiert. Die Formel zur Berechnung des Risk Scores lautet: Risk Score = Wahrscheinlichkeit \* Auswirkung.

Im untenstehenden Diagramm ist der Verlauf der Risiken zu sehen. Es zeigt, welche Risiken an welchem Datum abgeschlossen wurden und an welchen Tagen neue Risiken hinzugekommen sind.

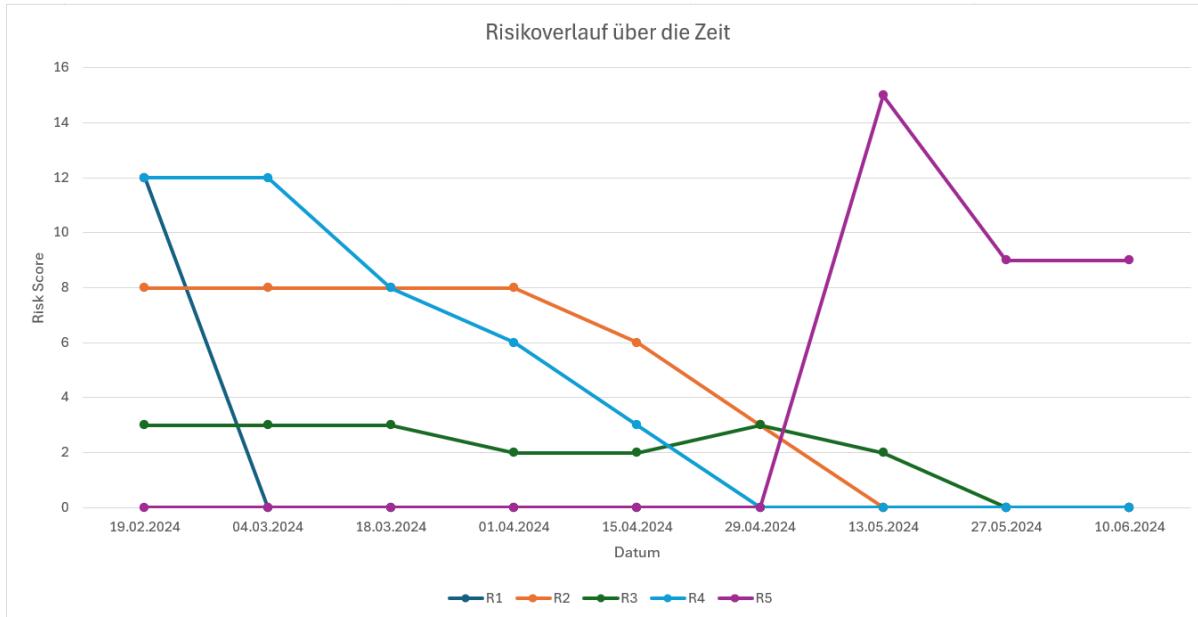


Abbildung 7.2: Risikoverlauf

# Kapitel 8: Zeiterfassung

## 8.1 TimeTracker

Die Zeiterfassung wird durch eine Third Party App namens TimeTracker verwaltet, die mit Jira kompatibel ist. Anschliessend werden die Daten von Jira exportiert und mit einem Python Skript wird ein grafischer Zeitreport erstellt. Jeder Zeitreport zeigt den Ist- und Sollzustand.

## 8.2 Auswertung

Die Zeiterfassung wird vor den Betreuungsmeetings aktualisiert und jeweils als Screenshot in diesem Kapitel hinzugefügt.

### 8.2.1 Auswertung Gesamtprojekt

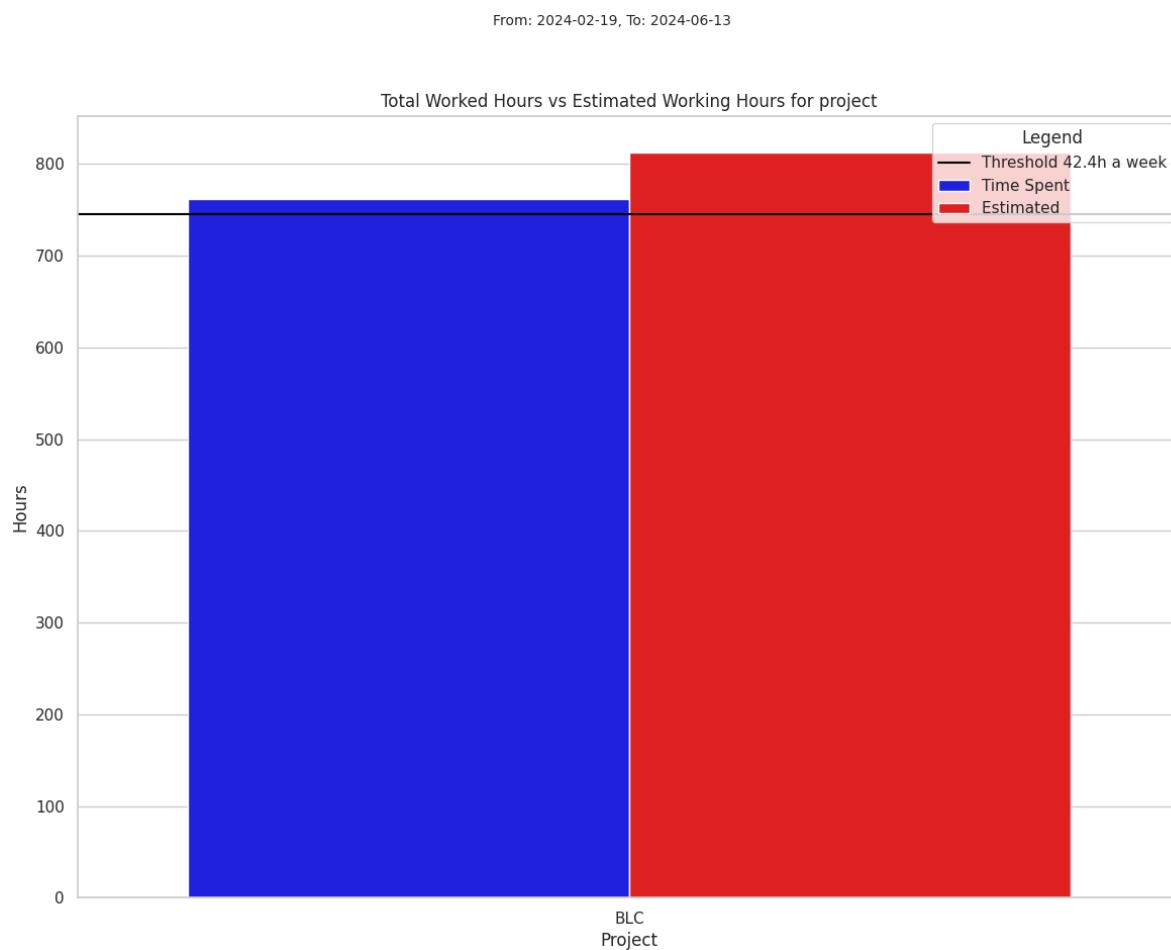


Abbildung 8.1: Auswertung Gesamtprojekt

### 8.2.2 Auswertung pro Phase

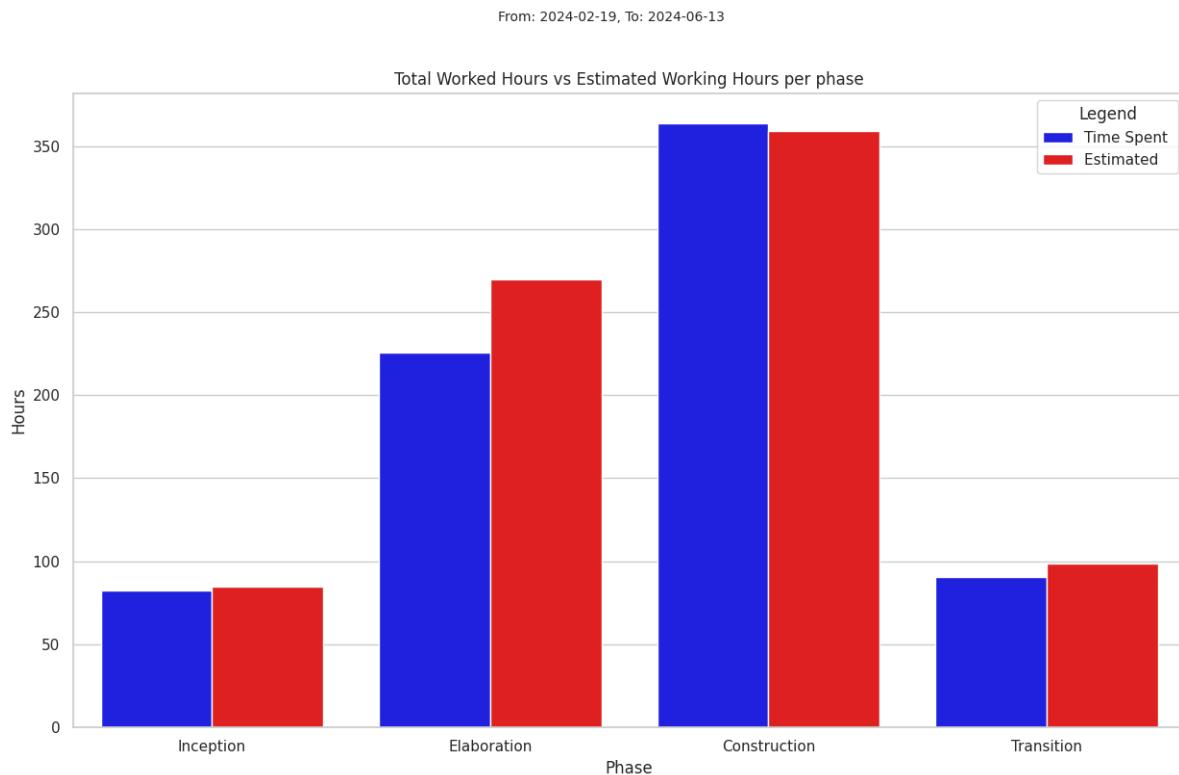


Abbildung 8.2: Auswertung pro Phase

### 8.2.3 Auswertung pro Person

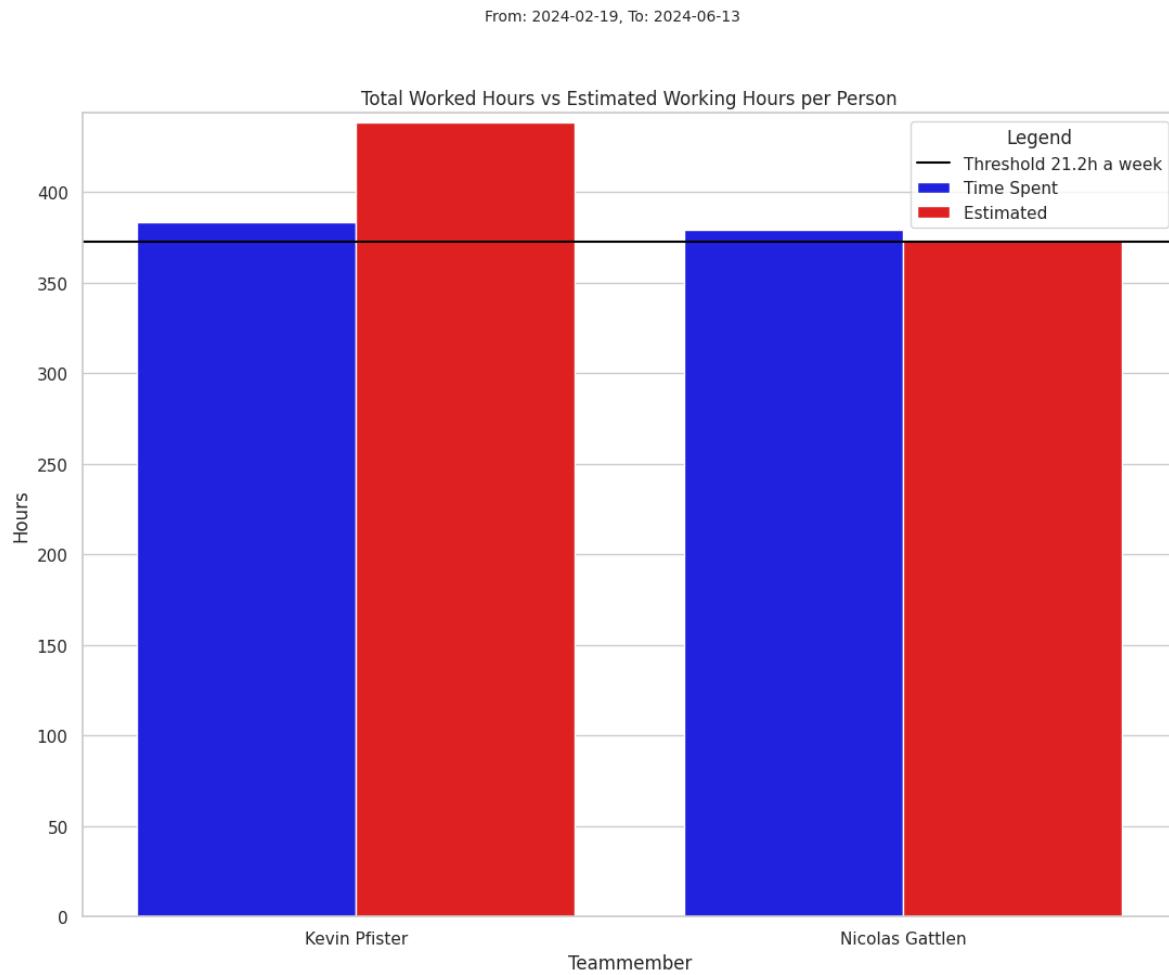


Abbildung 8.3: Auswertung pro Person

## **Teil IV**

## **Appendix**

## 8.3 Testauswertung

### 8.3.1 Testauswertung Experiment 1

#### Auswertung Nico

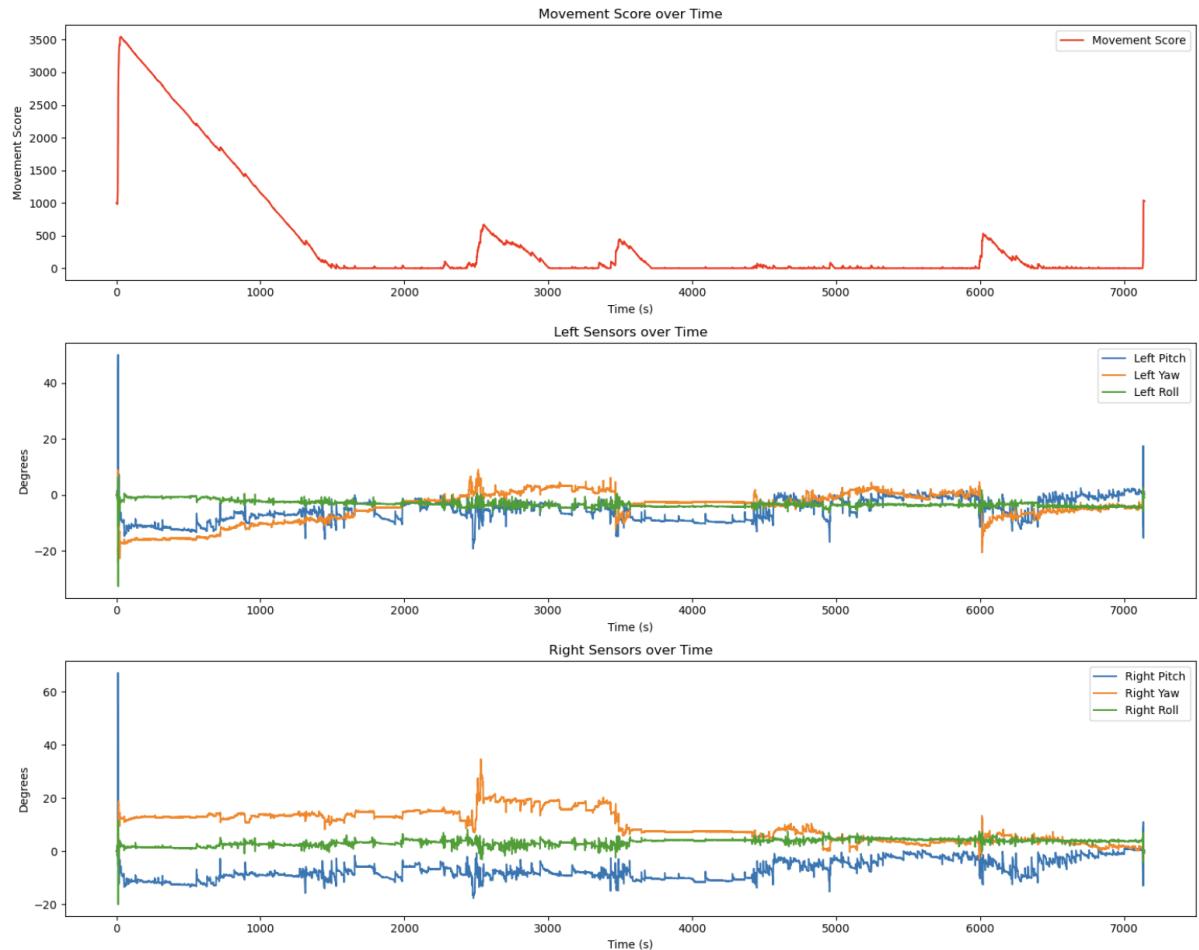


Abbildung 8.4: Auswertung Experiment One - Nico

## Auswertung Bruno

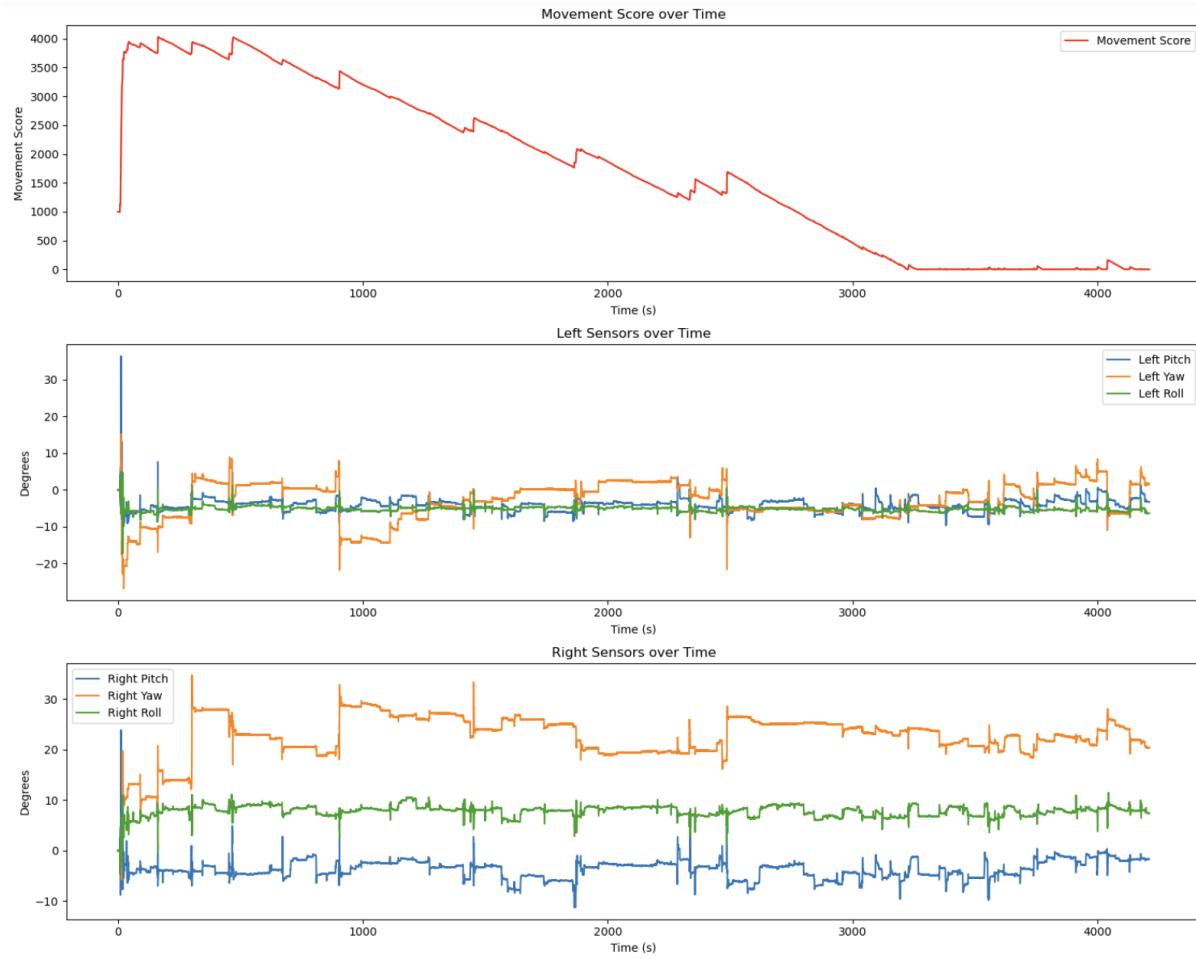


Abbildung 8.5: Auswertung Experiment One - Bruno

## Auswertung Dennis

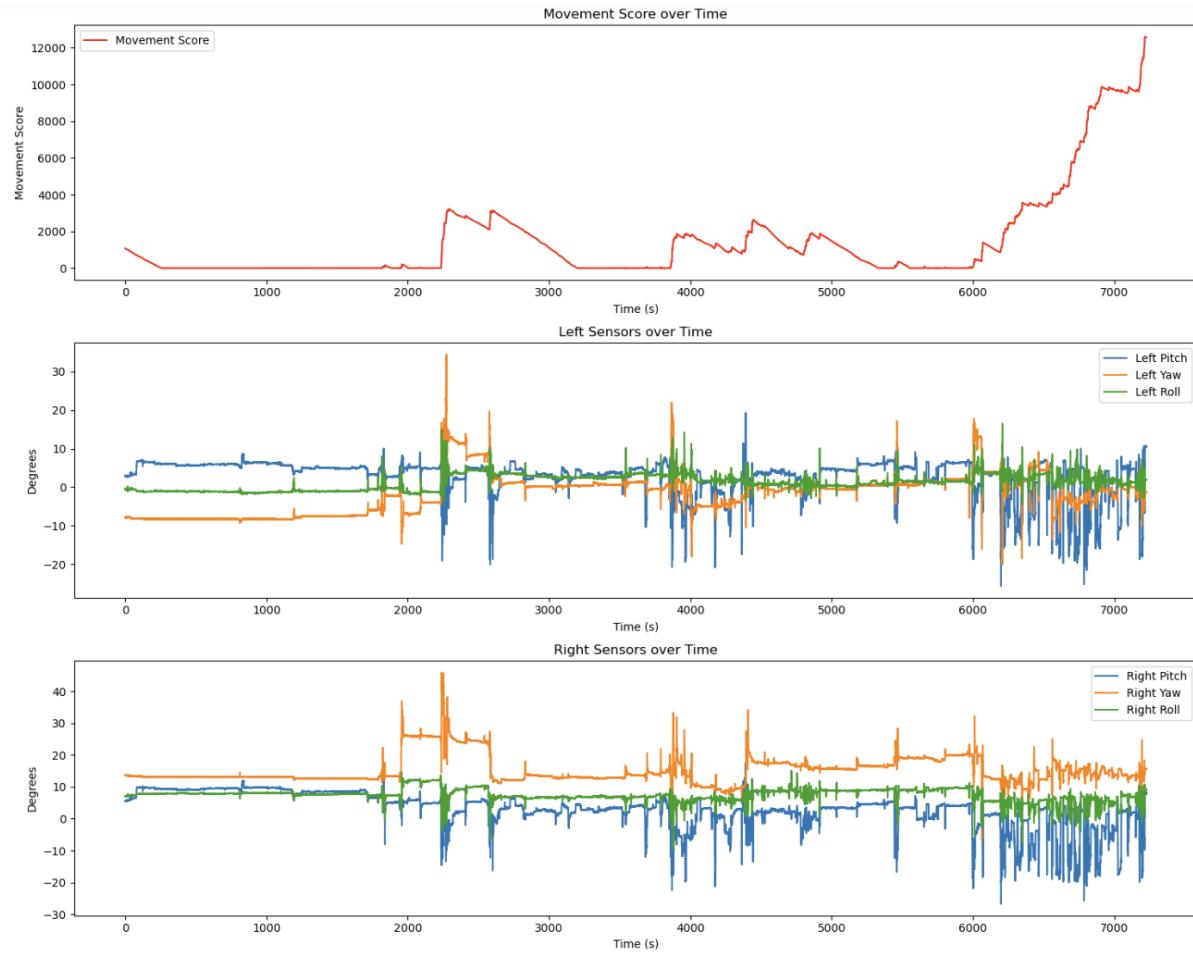


Abbildung 8.6: Auswertung Experiment One - Dennis

### 8.3.2 Testauswertung Experiment 2

#### Auswertung Basil

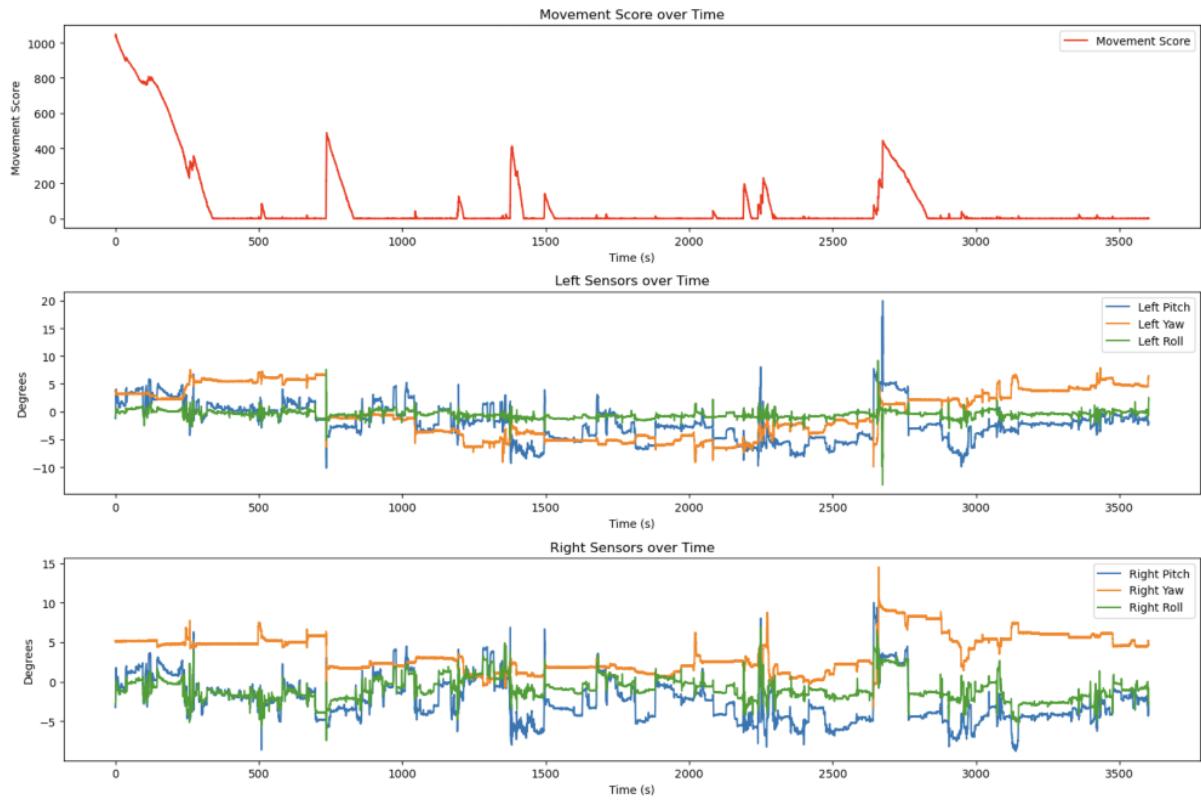


Abbildung 8.7: Auswertung Experiment Two - Basil ohne ActiveSit

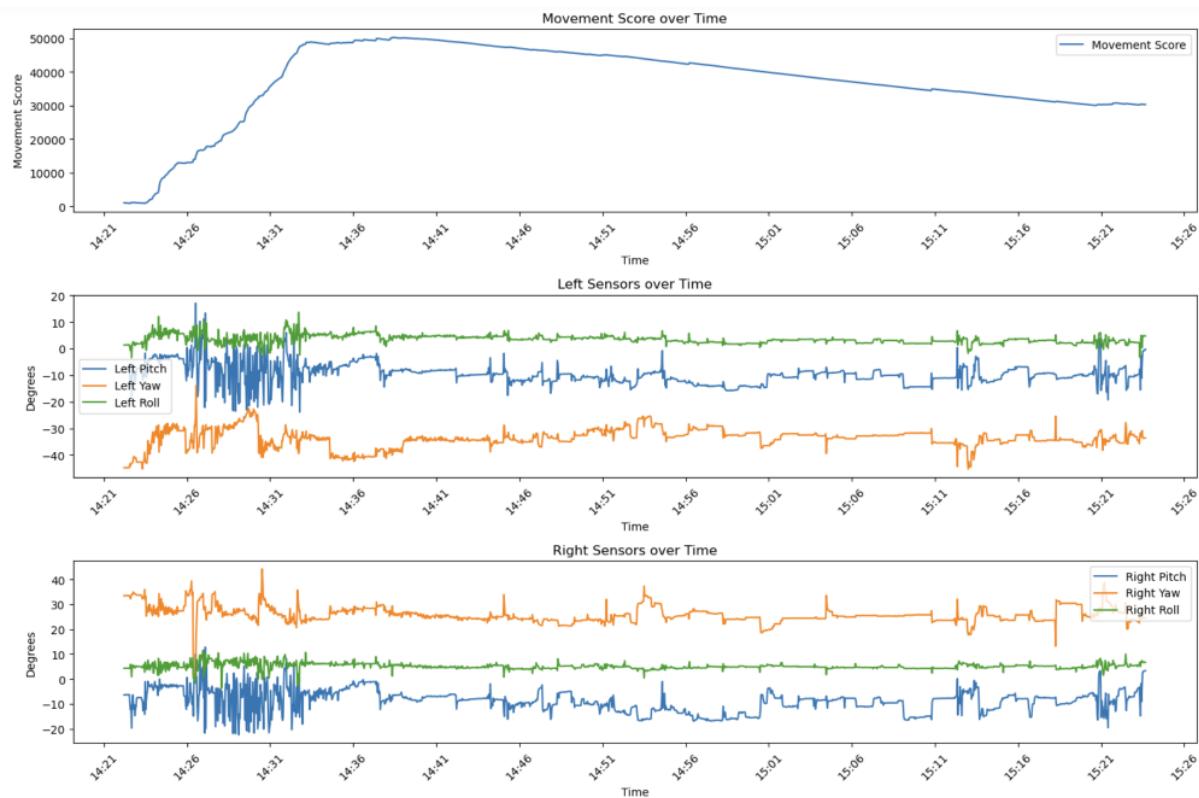


Abbildung 8.8: Auswertung Experiment Two - Basil mit ActiveSit

### Auswertung Florian

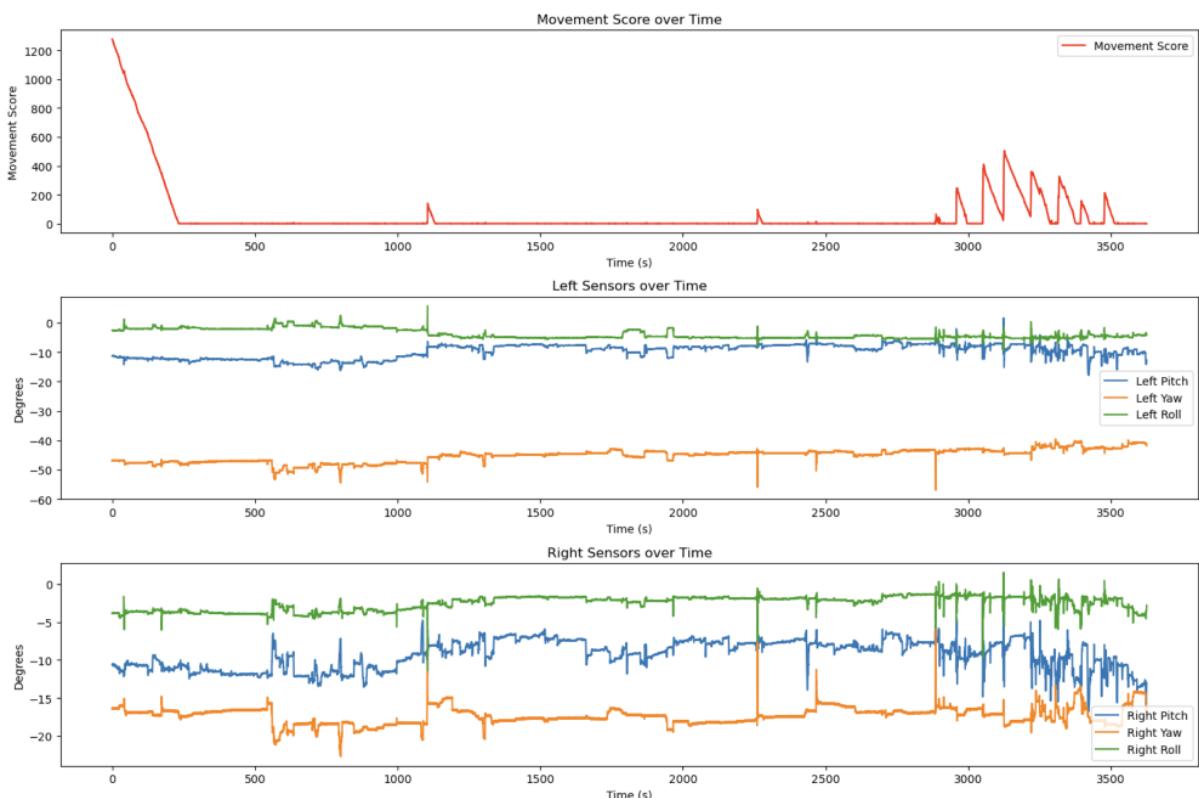


Abbildung 8.9: Auswertung Experiment Two - Florian ohne ActiveSit

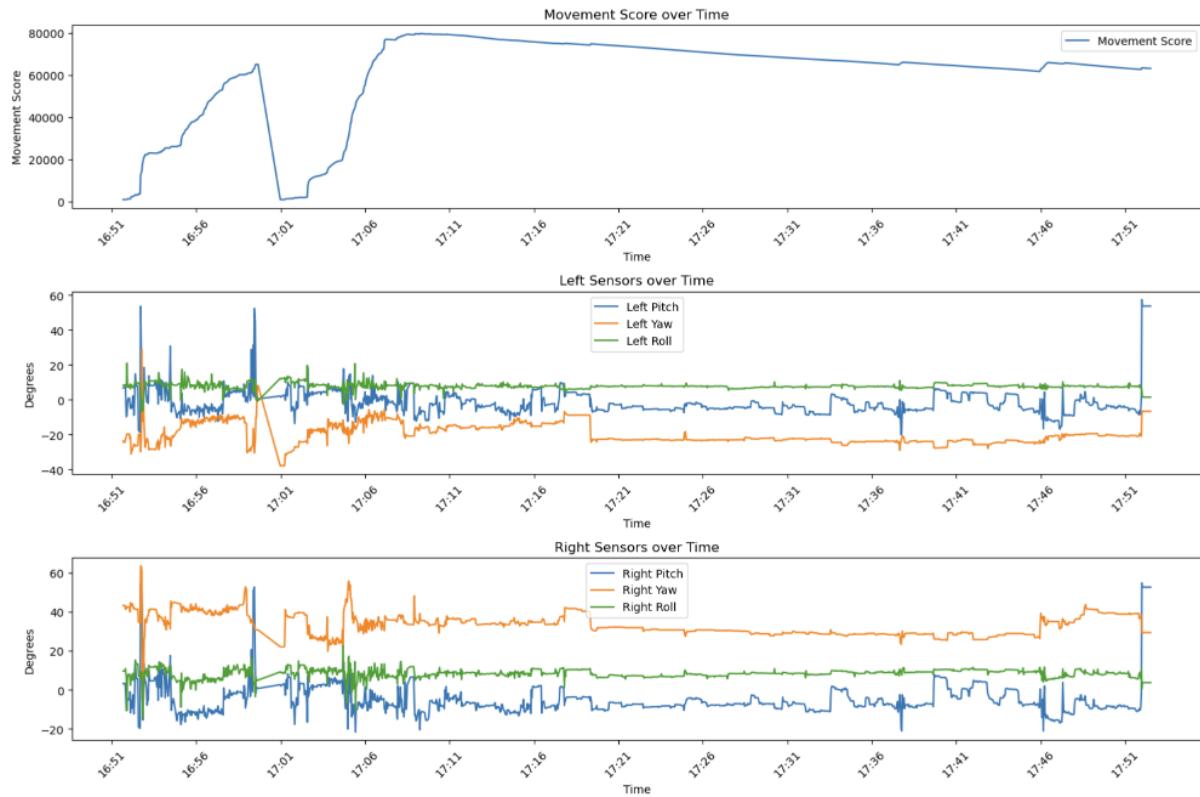


Abbildung 8.10: Auswertung Experiment Two - Florian mit ActiveSit

### Auswertung Joel

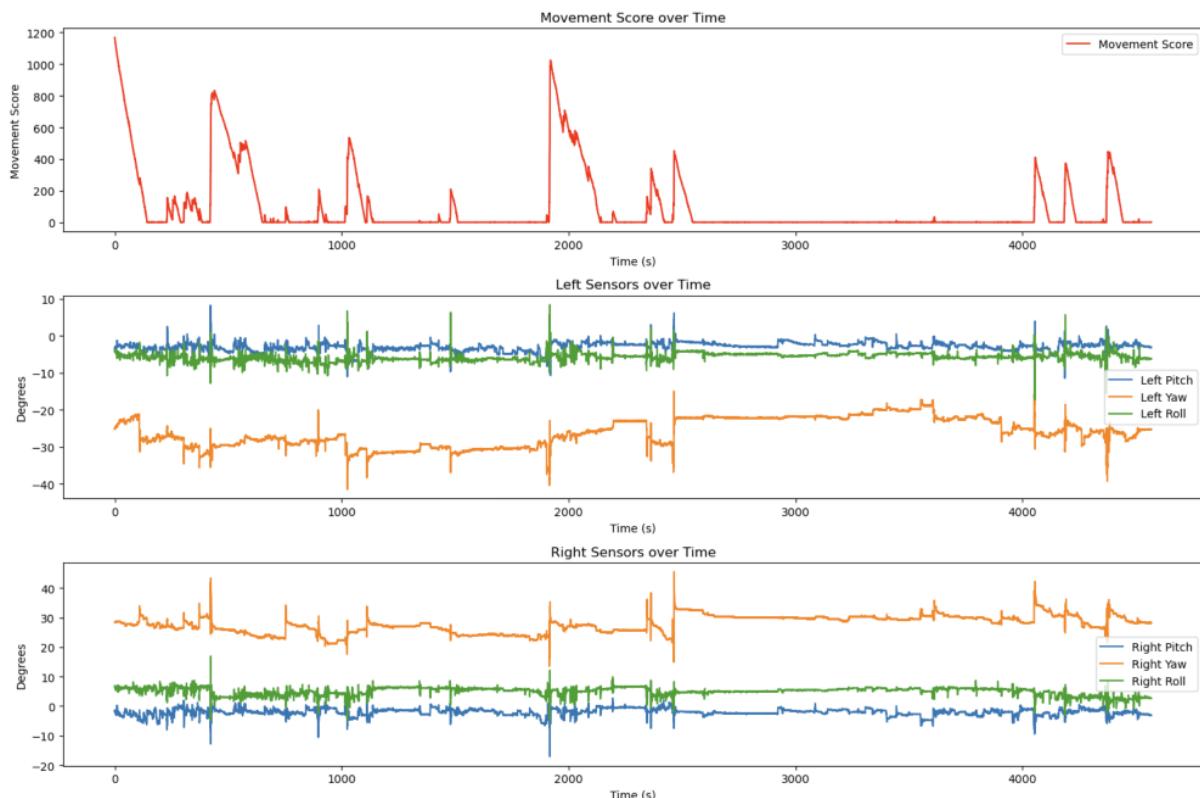


Abbildung 8.11: Auswertung Experiment Two - Joel ohne ActiveSit

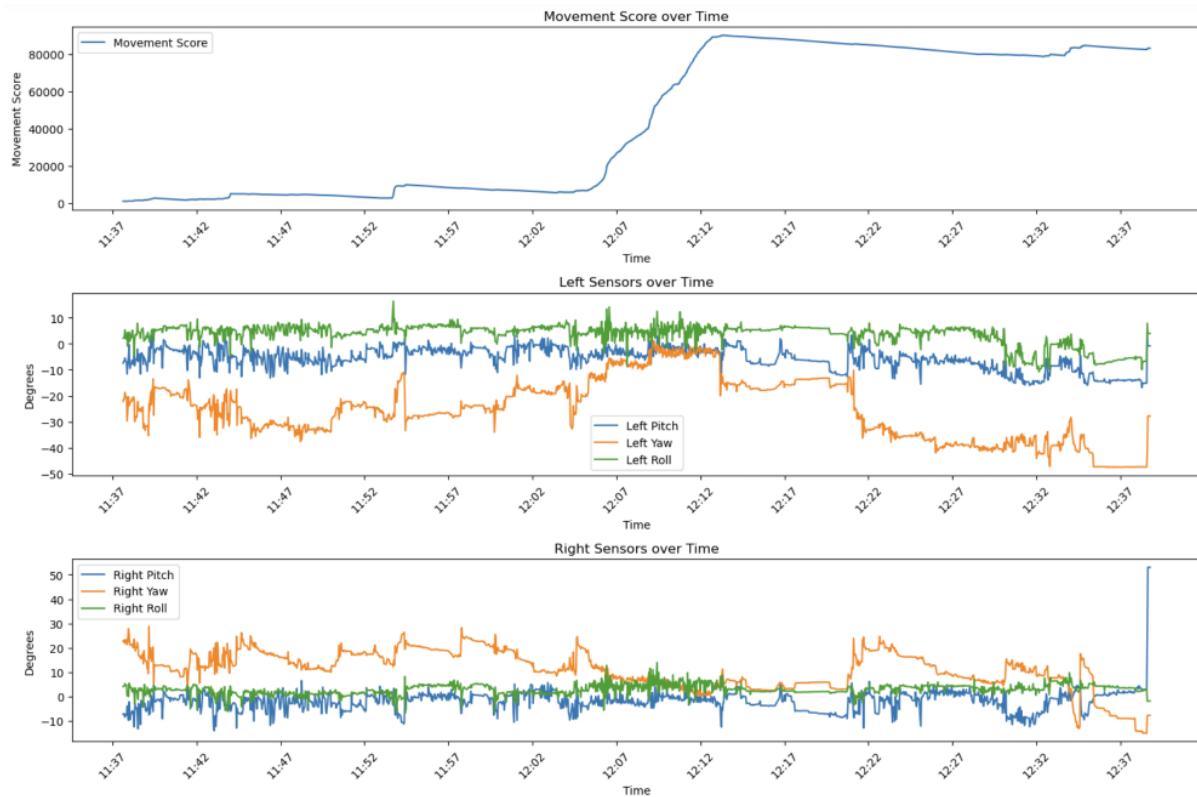


Abbildung 8.12: Auswertung Experiment Two - Joel mit ActiveSit

### Auswertung Joshua

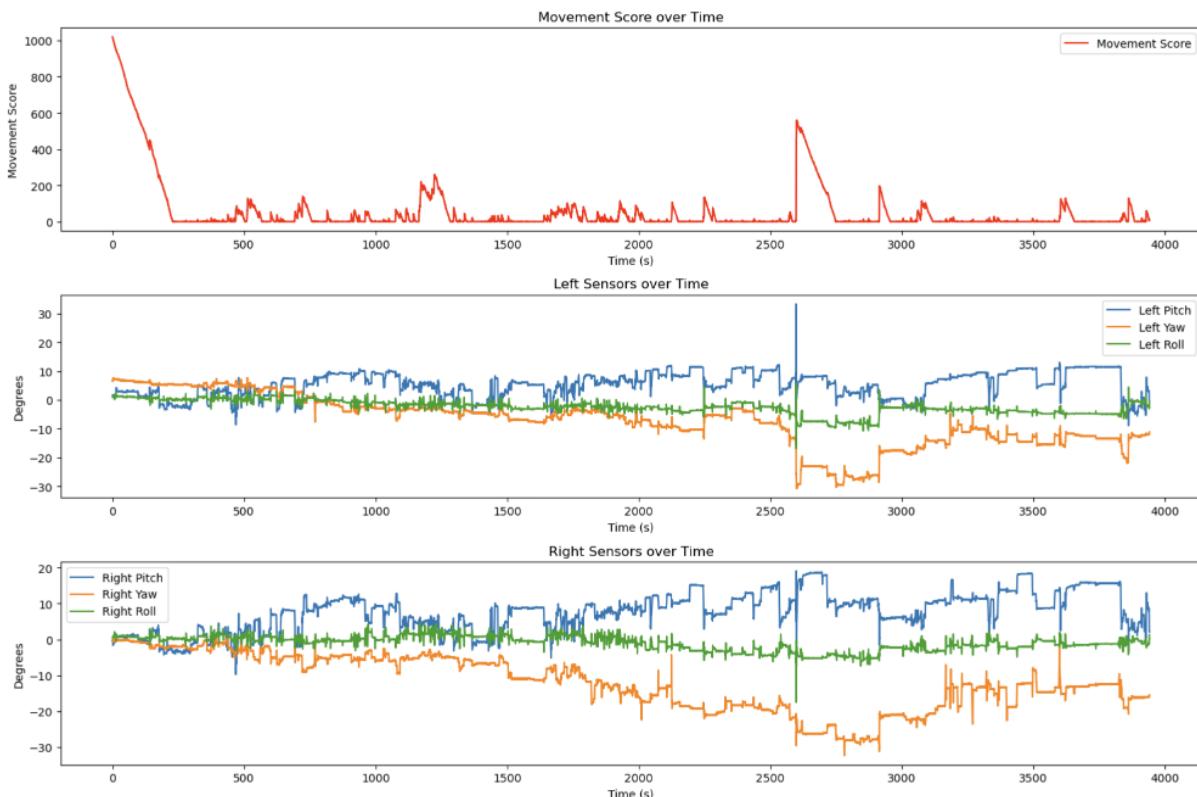


Abbildung 8.13: Auswertung Experiment Two - Joshua ohne ActiveSit

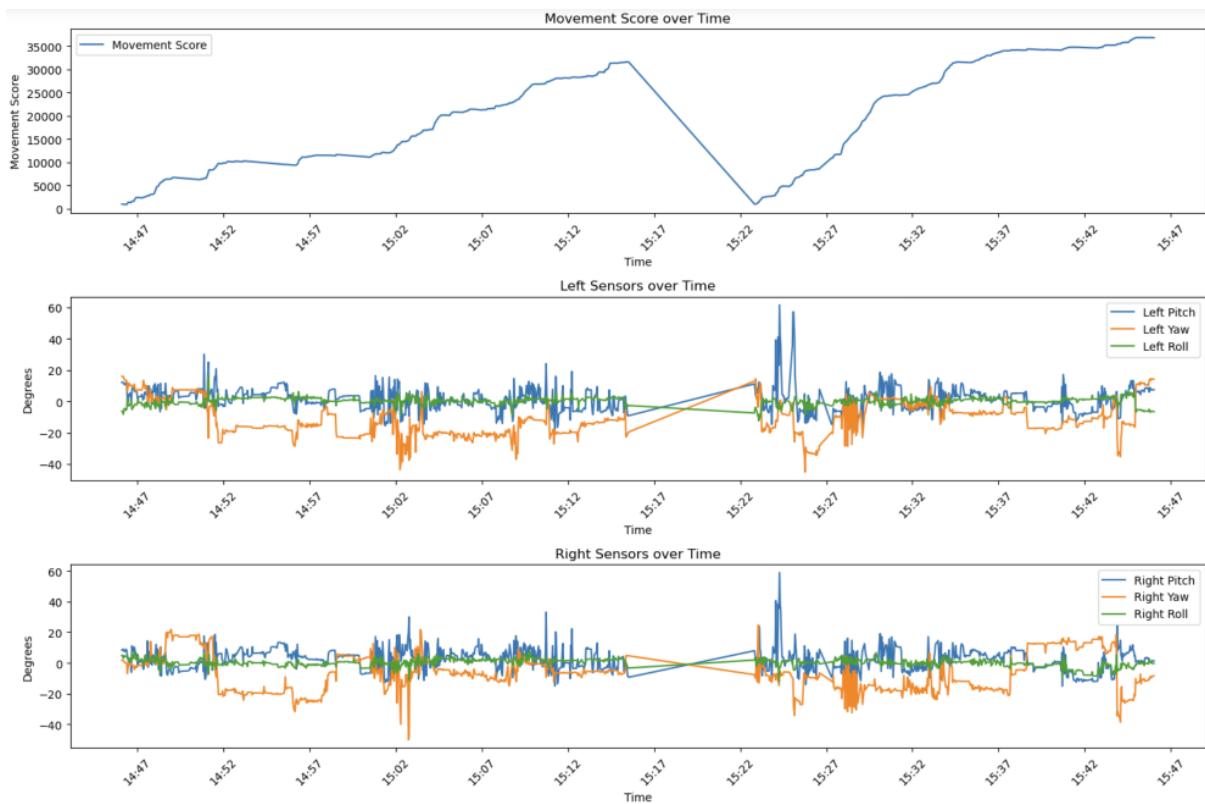


Abbildung 8.14: Auswertung Experiment Two - Joshua mit ActiveSit

### Auswertung Marco

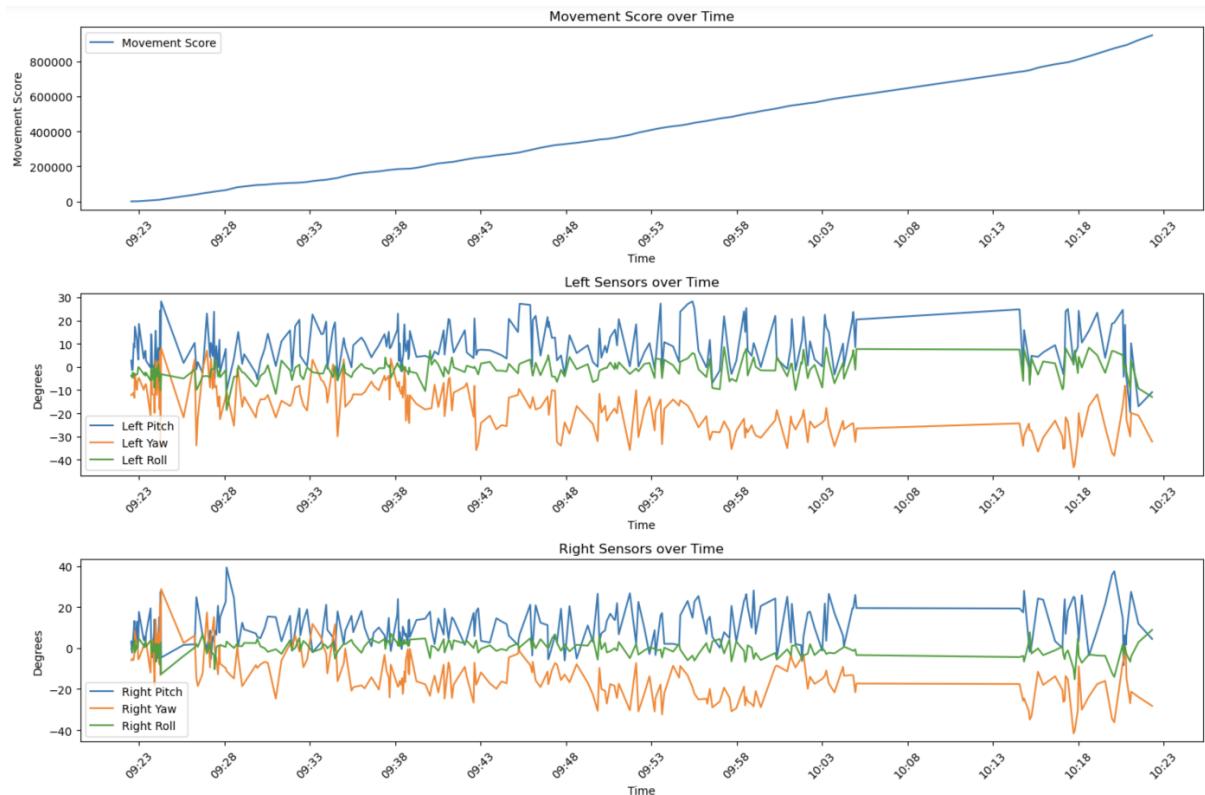


Abbildung 8.15: Auswertung Experiment Two - Marco mit ActiveSit

### 8.3.3 Testabdeckung Backend

Coverage report: 99%

*coverage.py v7.4.4, created at 2024-06-11 16:46 +0000*

Module	statements	missing	excluded	coverage
app.py	25	1	0	96%
controllers/__init__.py	0	0	0	100%
controllers/authentication_controller.py	32	0	0	100%
controllers/game_controller.py	20	0	0	100%
controllers/statistics_controller.py	34	0	0	100%
extensions.py	4	0	0	100%
models/__init__.py	0	0	0	100%
models/game.py	11	1	0	91%
models/participant.py	12	1	0	92%
models/sensor.py	17	1	0	94%
routes/__init__.py	0	0	0	100%
routes/authentication_bp.py	9	0	0	100%
routes/game_bp.py	12	0	0	100%
routes/statistics_bp.py	11	0	0	100%
services/__init__.py	0	0	0	100%
services/game_service.py	10	0	0	100%
services/participant_service.py	10	0	0	100%
services/sensor_service.py	13	0	0	100%
test/__init__.py	0	0	0	100%
test/conftest.py	30	0	0	100%
test/test_authentication.py	37	0	0	100%
test/test_game.py	22	0	0	100%
test/test_statistics.py	31	0	0	100%
<b>Total</b>	<b>340</b>	<b>4</b>	<b>0</b>	<b>99%</b>

Abbildung 8.16: Code Coverage Backend

### 8.3.4 Testauswertung Notizen Experiment Zwei

Protokoll Testing:

Stuhl Feedback:

Körperlich: Florian: Ab 40 Minuten Schmerzen im Rücken. Beine. Komisches Anfangsfehlgefühl. Keine körperlich. Testete Movementscore gleich zu Beginn. Bei starkem Fokus wurde Bewegung vernachlässigt.

Basil: Gar keine Schmerzen

Mit Sw: Movementscore wollte hoch gebracht werden. Spiele wurden genutzt aber nicht so lange.

Bei starkem Fokus ging der Movementscore nach unten.

Joel: Nach Bewegung wurde sich mehr geachtet. Auch bei höherem Fokus. Haltung zu spielen eher neutral. Movementscore sorgte für Begeisterung

Basil: Spiele wurden sofort gezeigt und wurde sofort gespielt. Movementscore war hoch. Bewegung war neben den Spielen eher tief. Während dem Musik hören wurde sich bewegt. Mobilität geschätzt. Aber nicht gebraucht.

Joshua: Spiele funktionierten nicht. Begeisterung für den Movementscore. Kein körperlichen Beschwerden.

Ana: Nichts anstrengend körper keine beschwerden. Am anfang wurde sich viel bewegt aufgrund der Spiele ca 12 min gespielt. Statistik nicht angeschaut nach dem Spiel. Es wurde nicht speziell bewegt danach. Keine aktive bewegung.

GUI: Intuitiv keine Probleme. End Game beim Spiel war Unklar.

## 8.4 Wireframes

### 8.4.1 Login Page

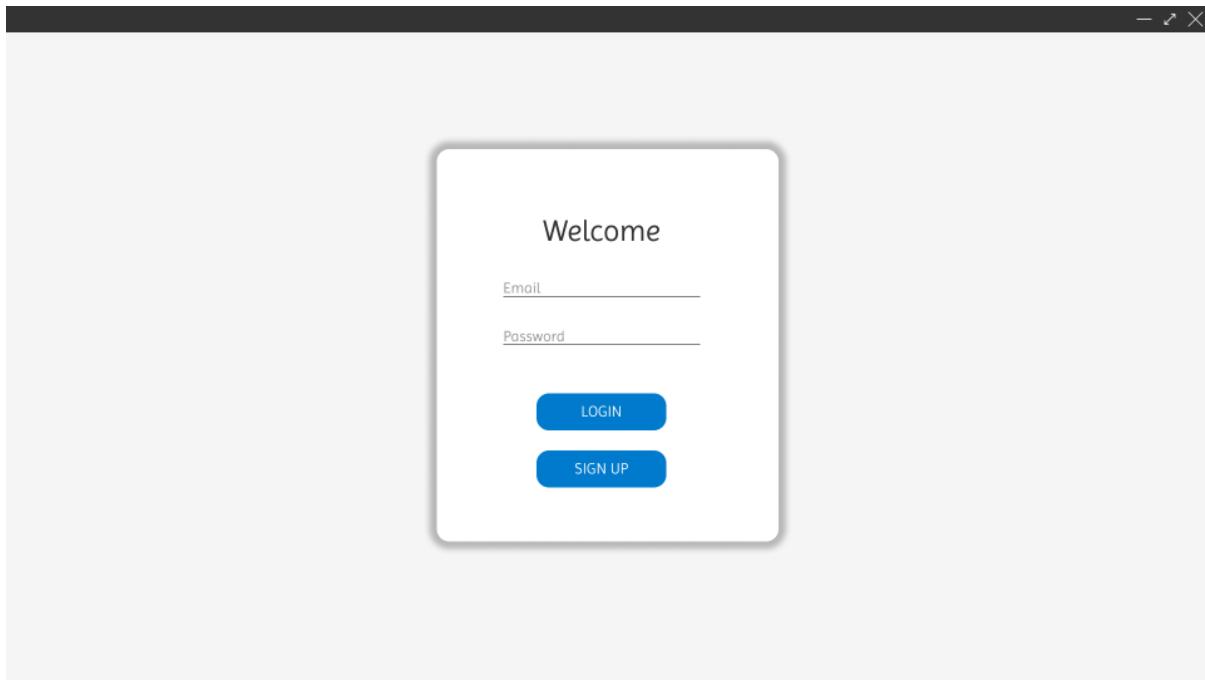


Abbildung 8.17: Wireframe - Login Page

### 8.4.2 Sign-Up Page

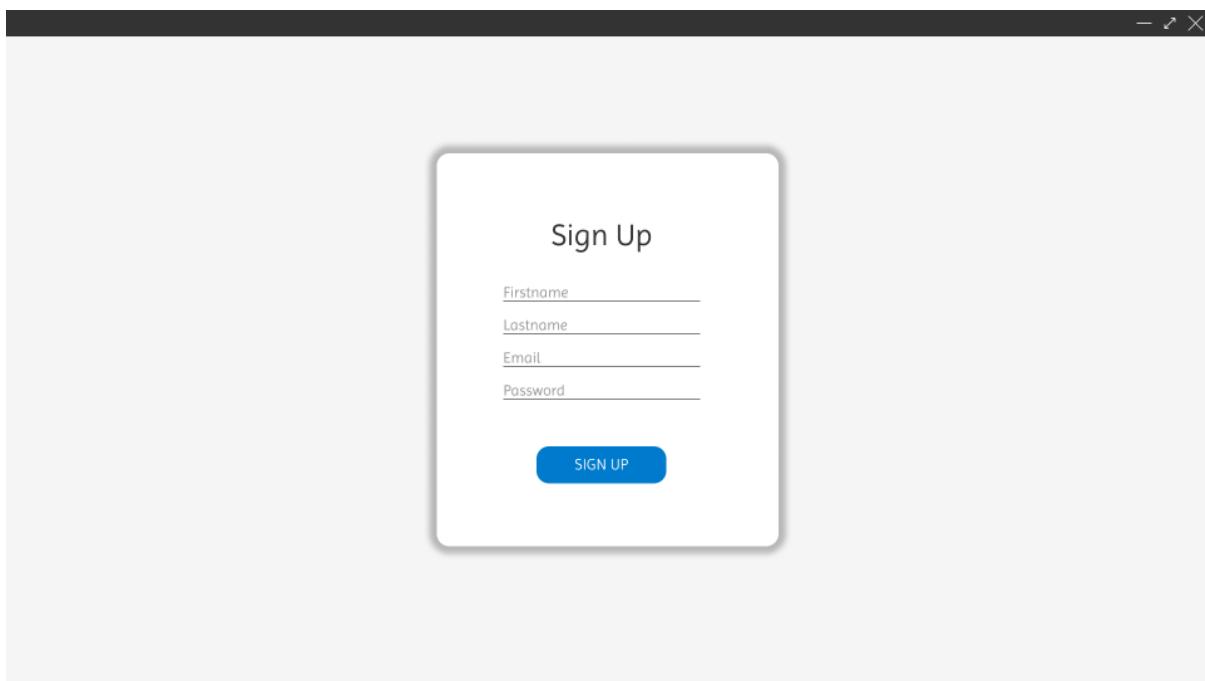


Abbildung 8.18: Wireframe - Sign-Up Page

#### 8.4.3 Statistics Page

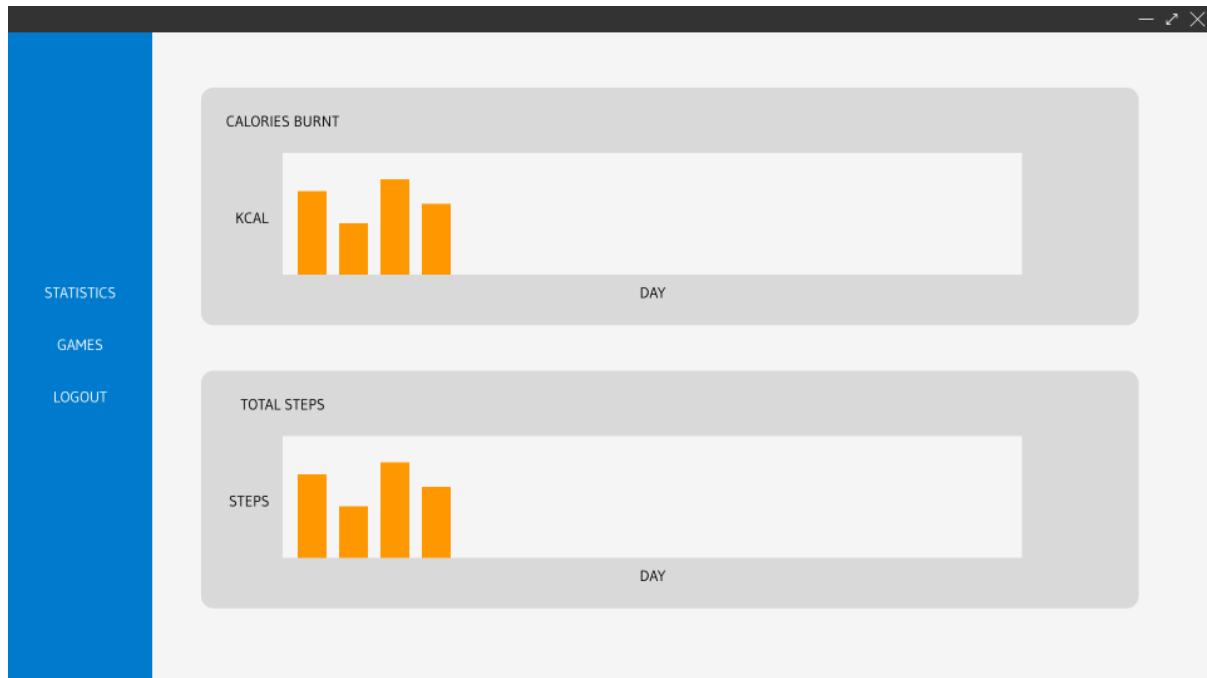


Abbildung 8.19: Wireframe - Statistics Page

#### 8.4.4 Gaming Page

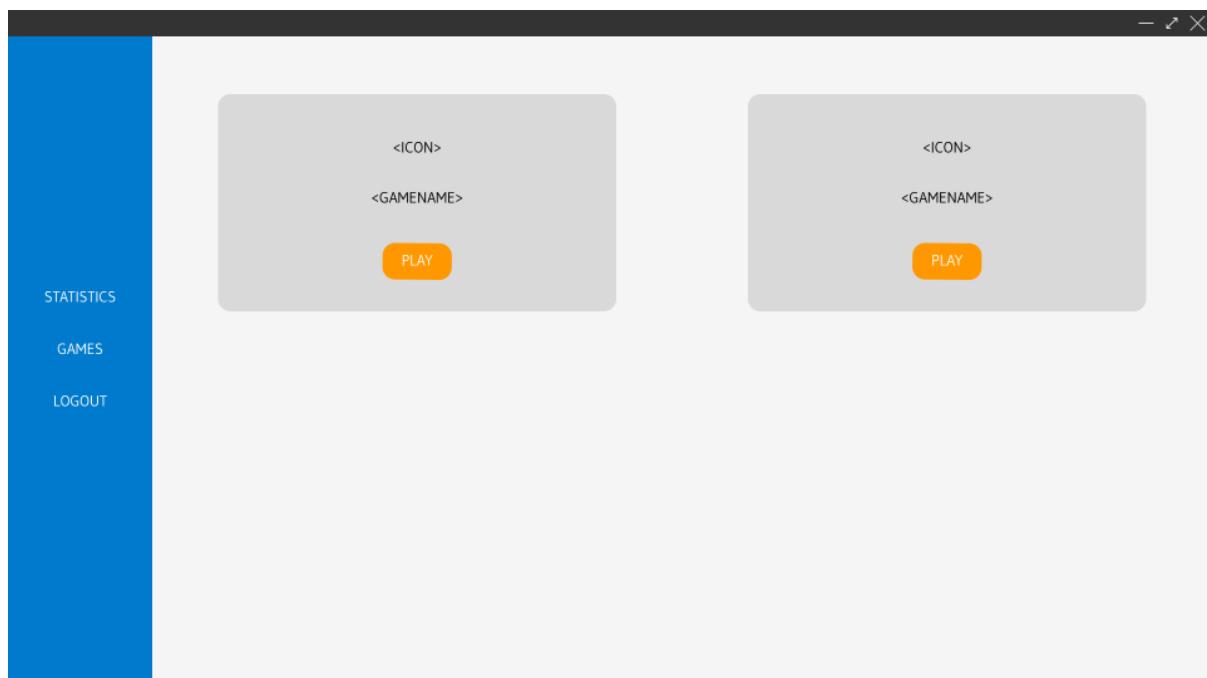


Abbildung 8.20: Wireframe - Gaming Page

## 8.5 Organisatorisches

### 8.5.1 Originalen Aufgabenstellung

---

Aufgabenstellung Bachelorarbeit  
FS24 Informatik OST  
«Limbic Chair Anwendung»

---

#### **1. Betreuer**

Prof. Dr. Markus Stolze, OST, IFS, [markus.stolze@ost.ch](mailto:markus.stolze@ost.ch)

#### **2. Praxispartner**

Limbic Life AG  
Dr. med. Patrik Künzler  
Müllerstrasse 25  
CH-8004 Zürich  
Schweiz

#### **3. Studierende**

Kevin Pfister  
Nicolas Gattlen

#### **4. Ausgangslage**

In der heutigen Arbeitswelt, in der ein Grossteil der Menschen einen erheblichen Teil ihres Tages im Sitzen verbringt, ist die Bedeutung einer ergonomischen Sitzhaltung und der Prävention von sitzbedingten Gesundheitsproblemen unumstritten. Der Limbic Chair, entwickelt von Limbic Life, stellt in diesem Zusammenhang eine innovative Lösung dar. Durch sein einzigartiges Design und die Integration von fünf Sensoren, die die Sitzposition in Echtzeit erfassen, hebt sich der Stuhl von traditionellen Bürostühlen ab. Diese Sensoren bieten die Möglichkeit, detaillierte Daten über das Sitzverhalten der Nutzer zu sammeln. Die aktuelle Softwarelösung ermöglicht bereits eine grundlegende Auswertung dieser Daten unter Windows, jedoch ohne spezifische Ausrichtung auf die Förderung einer gesunden Sitzhaltung.

Die Bedeutung einer aktiven Sitzhaltung, bei der regelmässige Bewegung und Haltungswechsel integriert sind, wird durch zahlreiche Studien unterstrichen. Diese Form des Sitzens kann nicht nur muskuloskelettalen Beschwerden vorbeugen, sondern auch die allgemeine Gesundheit und das Wohlbefinden steigern. Trotz des Potenzials des Limbic Chairs, ein gesundes Sitzverhalten zu unterstützen, fehlt es derzeit an einer spezialisierten Softwarelösung, die Nutzer aktiv dazu anleitet, ihre Sitzgewohnheiten zu verbessern.

#### **5. Ziel der Arbeit**

Die zentrale Herausforderung dieser Bachelorarbeit liegt in der Entwicklung einer Software, die die einzigartigen Eigenschaften des Limbic Chairs nutzt, um die Nutzer zu einem gesünderen Sitzverhalten zu motivieren. Die Software soll durch interaktive Visualisierungen, Erinnerungen und gezielte Übungen dazu beitragen, dass Nutzer eine aktiver und damit gesündere Sitzhaltung einnehmen. Ein kritischer erster Schritt in diesem Prozess ist die Analyse, inwiefern sich gesundes (aktives) von ungesundem (statischem) Sitzen anhand der vom Limbic Chair erfassten Sensordaten in Echtzeit unterscheiden lässt.

Um dieses Ziel zu erreichen, soll die Arbeit folgende Schritte umfassen:

**(1) Literaturrecherche:** Im initialen Literaturstudium sollen die theoretischen Grundlagen von ergonomischem Sitzen und den Einfluss von Sitzgewohnheiten auf die Gesundheit geklärt werden.

Besonderes Augenmerk soll hierbei auf der Identifikation von Parametern, die aktives Sitzen charakterisieren liegen.

**(2) Datensammlung:** Die Sammlung von Sensordaten verschiedener Sitzhaltungen und -aktivitäten unter Einsatz des Limbic Chairs. Diese Daten bilden die Grundlage für die nachfolgende Analyse.

**(3) Datenanalyse:** Die erfassten Daten sollen sowohl manuell als auch automatisiert analysiert werden, um Muster zu identifizieren, die aktives bzw. statisches Sitzen charakterisieren. Diese Analyse soll Aufschluss darüber geben, welche Datenpunkte und Muster für die Entwicklung der Software genutzt werden können.

**(4) Konzeptentwicklung:** Auf Basis der gewonnenen Erkenntnisse soll ein Konzept für die Software entwickelt werden, das festlegt, wie Nutzer durch die Software zu einem gesünderen Sitzverhalten angeleitet werden können. Dies umfasst die Gestaltung der Benutzeroberfläche, die Art der Visualisierungen, die Formulierung von Erinnerungen und die Auswahl bzw. Entwicklung geeigneter Übungen sowie deren Unterstützung mittels Software.

Die Entwicklung dieser Software soll nicht nur das Potenzial des Limbic Chairs voll ausschöpfen, sondern auch beispielhaft demonstrieren, dass durch den Einsatz von Limbic Chairs ein bedeutenden Beitrag zur Förderung der Gesundheit am Arbeitsplatz I einem Betrieb geleistet werden kann. Durch die Kombination aus ergonomischem Design und technologischer Innovation bietet diese Arbeit die Möglichkeit, einen neuen Standard für gesundes Sitzen im Büroalltag zu setzen.

## 6. Auftrag

Die folgenden Teilaufgaben sind zu bearbeiten (soweit sinnvoll)

- Analyse: Erhebung und Dokumentation der Anforderungen
  - Beschreibung der Ausgangssituation. Beschreibung des Status Quo aus technischer Sicht und aus Nutzersicht (wer sind Nutzer, was sind deren aktuelle Probleme)
  - Dokumentation der «Architecture Constraints»
  - Funktionale Anforderungen
  - Nicht-funktionale Anforderungen
- Entwurf: Entwicklung Architektur und Vorschlag UX
  - Analyse sinnvoller Architektur und (soweit sinnvoll) unterstützender Libraries und Frameworks
  - Beschreibung der Nutzer und Rollen
  - Prototyping UX (Paper, Scenario Walkthrough, ...)
  - Definition eines MVP + Hypothesen zu Ausbauschritten
  - Demonstration Architektur-Prototyp und validiertem UX-Vorschlag
- Implementation und Test (mit Referenz auf FA/NFA)
  - Dokumentation des Erreichungsgrads der FA/NFA
  - Beschreibung weiter Limitation
  - Empfehlungen und Beschreibungen zu weiteren Ausbauschritten mit Aufwandschätzung, inklusive von absehbaren Weiterentwicklungen.
- Sinnvolle Dokumentation des Projektes und der prototypischen Implementation
  - Notwendige Dokumentation entsprechend «Leitfaden»
  - Developer-Dokumentation
  - Projekt-Dokumentation inkl. Reflexion zur Nutzung von (KI-)Werkzeugen zur Text-Generierung.

## **7. Rahmenbedingungen**

Die von den Studierenden erstelle Software wird von den Studierenden mit einer Open-Source Lizenz versehen. Der Zeitpunkt und die Modalitäten der Verfügbarmachung des Codes wird zwischen den Studierenden und dem Betreuer abgesprochen. OST und die Studierenden können den Code auch vor der Veröffentlichung durch die Studierenden in Projekten einsetzen.

## **8. Schlussbestimmungen**

Im Weiteren gelten die Bedingungen zu den Informatik Bachelor- und Studienarbeiten entsprechend dem aktuellen «Leitfaden für Bachelor- und Studienarbeiten» des Bachelorstudiengangs Informatik an der OST

Das Kriterien-Raster zur Bewertung wird den Studierenden zusammen mit dieser Aufgabenstellung verteilt. Allfällige Anpassungen müssen bis Mitte der Arbeit einvernehmlich beschlossen werden.

Prof. Dr. Markus Stolze  
(Dokument ohne Unterschrift)

## 8.5.2 Projektplanung

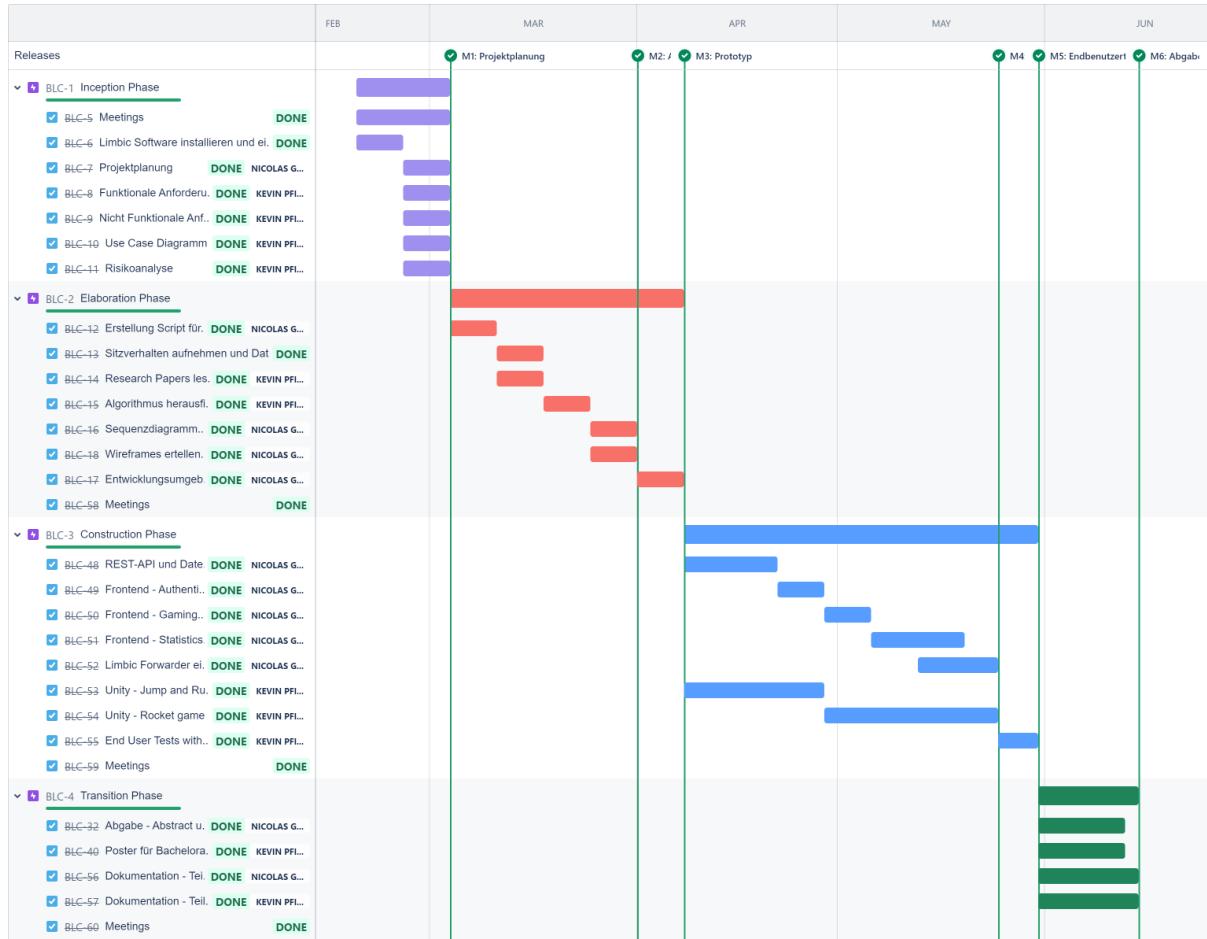


Abbildung 8.21: Projektplanung

# Tabellenverzeichnis

3.1	NFR-1 . . . . .	17
3.2	NFR-2 . . . . .	17
3.3	NFR-3 . . . . .	18
3.4	NFR-4 . . . . .	18
3.5	NFR-5 . . . . .	18
4.1	Beschreibung Frontend . . . . .	23
4.2	Ablauf einer Sitzungssession im Frontend . . . . .	25
4.3	Beschreibung Backend Controllers . . . . .	33
4.4	Beschreibung Limbic Services . . . . .	36
7.1	Daten und Dauer der einzelnen Phasen . . . . .	54
7.2	Meilensteine . . . . .	56
7.3	Beschreibung der verschiedenen Typen von Issues . . . . .	57
7.4	Risikomatrix . . . . .	58
7.5	Risiken . . . . .	58

# Abbildungsverzeichnis

1.1	Limbic Chair - Quelle:[5] . . . . .	4
1.2	Limbic Chair der sich der Bewegung anpasst - Quelle: [1] . . . . .	5
2.1	Stühle die für das Experiment gebraucht wurden - Quelle: [6] . . . . .	9
2.2	Auswertung Experiment . . . . .	11
3.1	C4 Diagram Layer 1 - Context Level without ActiveSit . . . . .	13
3.2	C4 Diagram Layer 1 - Context Level with ActiveSit . . . . .	13
3.3	Limbic Chair mit eingezeichneten Pitch und Yaw Bewegung, ohne Roll Bewegung - Quelle: [2] . . . . .	15
3.4	Use Case Diagramm . . . . .	16
4.1	Sequenz Diagramm - Aktivitätsverlauf . . . . .	20
4.2	C4 Diagram Layer 2 - Container Diagramm . . . . .	22
4.3	C4 Diagram Layer 3 - Frontend . . . . .	24
4.4	Ablauf Sitzungssession . . . . .	25
4.5	Userflow Diagramm . . . . .	26
4.6	Dashboard Statistics Page . . . . .	27
4.7	Dashboard Statistics Page . . . . .	27
4.8	Rocket Escape . . . . .	28
4.9	Clientreceiver bei der Anmeldung zum Connector . . . . .	29
4.10	Berechnung der Y Position der Rakete . . . . .	30
4.11	Spiel Alien Run . . . . .	31
4.12	Codeausschnitt für die Bewegung des Charakters . . . . .	32
4.13	C4 Diagram Layer 3 - Backend . . . . .	34
4.14	ERD Diagramm . . . . .	35
4.15	C4 Diagram Layer 3 - Limbic Service . . . . .	37
4.16	Berechnung der Sensordaten mit der Bestimmung des Multipliers . . . . .	38
4.17	Berechnung des Movementscores . . . . .	40
5.1	Auswertung Experiment Proband 1 ohne ActiveSit . . . . .	44
5.2	Auswertung Experiment Proband 1 mit ActiveSit . . . . .	45
5.3	Auswertung Experiment Proband 2 ohne ActiveSit . . . . .	46
5.4	Auswertung Experiment Proband 2 mit ActiveSit . . . . .	46
5.5	Auswertung Experiment Proband 3 ohne ActiveSit . . . . .	47
5.6	Auswertung Experiment Proband 3 mit ActiveSit . . . . .	48
6.1	CI/CD Pipeline . . . . .	51
7.1	Roadmap . . . . .	54
7.2	Risikoverlauf . . . . .	60
8.1	Auswertung Gesamtprojekt . . . . .	61
8.2	Auswertung pro Phase . . . . .	62
8.3	Auswertung pro Person . . . . .	63
8.4	Auswertung Experiment One - Nico . . . . .	65
8.5	Auswertung Experiment One - Bruno . . . . .	66
8.6	Auswertung Experiment One - Dennis . . . . .	67
8.7	Auswertung Experiment Two - Basil ohne ActiveSit . . . . .	68

8.8	Auswertung Experiment Two - Basil mit ActiveSit . . . . .	69
8.9	Auswertung Experiment Two - Florian ohne ActiveSit . . . . .	69
8.10	Auswertung Experiment Two - Florian mit ActiveSit . . . . .	70
8.11	Auswertung Experiment Two - Joel ohne ActiveSit . . . . .	70
8.12	Auswertung Experiment Two - Joel mit ActiveSit . . . . .	71
8.13	Auswertung Experiment Two - Joshua ohne ActiveSit . . . . .	71
8.14	Auswertung Experiment Two - Joshua mit ActiveSit . . . . .	72
8.15	Auswertung Experiment Two - Marco mit ActiveSit . . . . .	72
8.16	Code Coverage Backend . . . . .	73
8.17	Wireframe - Login Page . . . . .	75
8.18	Wireframe - Sign-Up Page . . . . .	75
8.19	Wireframe - Statistics Page . . . . .	76
8.20	Wireframe - Gaming Page . . . . .	76
8.21	Projektplanung . . . . .	80

# Literatur

- [1] Alba, M. (2024). The Chair That Transforms How You Work [Abbildung]. Engineering.com. [Abrufdatum: 9. Juni 2024]. <https://www.engineering.com/story/the-chair-that-transforms-how-you-work>
- [2] Bektaş, K., Thrash, T., Van Raaij, M. A., Künzler, P., & Hahnloser, R. (2021). The systematic evaluation of an embodied control interface for virtual reality [Abbildung]. ETH Zürich. [Abrufdatum: 10. Juni 2024]. <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/521194/3/journal.pone.0259977.pdf>
- [3] Berander, P., Damm, L.-O., Eriksson, J., Gorschek, T., Henningsson, K., Jönsson, P., Kågström, S., Milicic, D., Mårtensson, F., Rönkkö, K., & Tomaszewski, P. (2005). Software quality attributes and trade-offs [Abrufdatum: 10. März 2024]. [https://www.researchgate.net/publication/238700270\\_Software\\_quality\\_attributes\\_and\\_trade-offs\\_Authors](https://www.researchgate.net/publication/238700270_Software_quality_attributes_and_trade-offs_Authors)
- [4] Bundesanstalt für Arbeitsschutz und Arbeitsmedizin. (2011). Sitzlust statt Sitzfrust. Sitzen bei der Arbeit und anderswo [Broschüre]. <https://www.baua.de/DE/Angebote/Publikationen/Praxis/A31.html>
- [5] ErgoPoint. (2024). Limbiclife Limbic Chair Custom Carbonschalen [Abbildung]. Sitz. [Abrufdatum: 9. Juni 2024]. <https://sitz.ch/produkt/limbiclife-limbic-chair-custom-carbonschalen/>
- [6] Léger, M. C., Dion, C., Albert, W. J., & Cardoso, M. R. (2022). The biomechanical benefits of active sitting. *Ergonomics*, 66(8), 1072–1089. <https://doi.org/10.1080/00140139.2022.2132298>