

MicroLabVR – Konzeption und
Implementierung einer VR-Anwendung zur
Visualisierung raumzeitlicher
Mikrobiomdaten

Masterarbeit

Zur Erlangung des Grades
Master of Science (M.Sc.)
im Studiengang

Medical Data Science

**eingereicht von
Simon Burbach**

Erstprüfer: Prof. Dr. rer. nat. Florian Centler
Zweitprüferin: Dr. rer. nat. Tanja Joan Schmidt

**Lebenswissenschaftliche Fakultät
Universität Siegen
2024**

Anfertigung der Arbeit

Die vorliegende Masterarbeit wurde in der Zeit vom 17.06.2024 bis 07.11.2024 als Abschlussarbeit des viersemestrigen Masterstudiengangs Medical Data Science an der Universität Siegen, in der Lebenswissenschaftlichen Fakultät, unter der Leitung von Prof. Dr. Centler, angefertigt.

Gender-Disclaimer

In dieser Arbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

Eidesstattliche Erklärung

Ich versichere, dass ich die schriftliche Ausarbeitung selbständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach (inkl. Übersetzungen) anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle (einschließlich des World Wide Web sowie anderer elektronischer Datensammlungen) deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für angefügte Zeichnungen, bildliche Darstellungen, Skizzen und dergleichen. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken sowie die Verwendung KI-basierter Textgeneratoren als solche kenntlich gemacht habe. Ich nehme zur Kenntnis, dass die nachgewiesene Unterlassung der Herkunftsangabe als versuchte Täuschung gewertet wird.

.....
Ort, Datum

.....
Unterschrift

Zusammenfassung

MicroLabVR ist eine Standalone-Virtual-Reality-Anwendung, die Mikrobiomdatensätze in einer immersiven Umgebung visualisiert und damit einen innovativen Ansatz für die Analyse raumzeitlicher mikrobieller Daten bietet. Die Anwendung überwindet die Einschränkungen herkömmlicher Visualisierungstools, indem sie es Anwendern ermöglicht, Daten zu Populationskonstellationen, Substanzen und metabolischen Flüssen intuitiv und interaktiv zu erforschen. Die immersive Visualisierung erleichtert das Verständnis mikrobieller Interaktionen und Zusammenhänge, die in herkömmlichen Darstellungen oft schwer zu erfassen sind. Ein zentrales Erfolgsmerkmal von MicroLabVR ist die optimierte Benutzerfreundlichkeit, die durch die sorgfältige Umsetzung bewährter User-Interface- und User-Experience-Designprinzipien erreicht wird. Diese Prinzipien verbessern nicht nur die Bedienbarkeit, sondern steigern auch die Effizienz und das Engagement der Benutzer. MicroLabVR ist damit ein erfolgreiches Proof of Concept und bietet großes Potential für zukünftige Entwicklungen in der Mikrobiomforschung.

Abstract

MicroLabVR is a standalone virtual reality application that visualizes microbiome datasets in an immersive environment, providing an innovative approach to the analysis of spatiotemporal microbial data. The application overcomes the limitations of conventional visualization tools by allowing users to explore data on population constellations, compounds and metabolic fluxes intuitively and interactively. The immersive visualization facilitates the understanding of microbial interactions and relationships that are often difficult to grasp in conventional representations. A key success feature of MicroLabVR is its optimized usability, which is achieved through the careful implementation of proven user interface and user experience design principles. These principles not only improve usability, but also increase efficiency and user engagement. MicroLabVR is thus a successful proof of concept and offers great potential for future developments in microbiome research.

Inhaltsverzeichnis

Anfertigung der Arbeit, Gender-Disclaimer und Eidesstattliche Erklärung	I
Abstracts	II
Abbildungsverzeichnis	V
Abkürzungsverzeichnis	VI
1 Motivation	1
1.1 Zielsetzung	1
1.2 Inhalt	2
2 Theoretische Grundlagen	3
2.1 Biologische Grundlagen	3
2.1.1 Das Mikrobiom	3
2.1.2 Bedeutung der Mikrobiomforschung	3
2.1.3 Flux balance analysis	5
2.1.4 Agentbasierte Modellierung	7
2.2 Technische Grundlagen	8
2.2.1 Virtual Reality und ihre Anfänge	8
2.2.2 Immersion	9
2.2.3 Präsenz	9
2.2.4 Cybersickness	10
2.2.5 Meta Quest 3	11
2.2.6 Blender	12
2.2.7 Unity Game Engine	12
3 Aktueller Stand von Forschung und Technik	16
3.1 Software zur Mikrobiommodellierung	16
3.1.1 Restriktionsbasierte Modellierungssoftware	16
3.1.2 Agentbasierte Modellierungssoftware	18
3.2 Mikrobielle Datenvisualisierung	22
3.3 UI- und UX-Design	25
4 Konzeption	31
4.1 Entwicklung eines Konverters in R	31

4.2	Einlesen der Datensätze und Formatprüfungen	32
4.3	Darstellung der Organismenpopulation	32
4.4	Darstellung der Substanzkonzentrationen	33
4.5	Darstellung der metabolischen Flüsse	33
4.6	Steuerung der Datenvisualisierung und Benutzerfreundlichkeit	34
4.7	Steuerung und Fortbewegung	34
4.8	Limittesting	34
4.9	Szenengestaltung	35
4.10	Datenarchitektur der VR-Anwendung	35
5	Implementierung	37
5.1	R-Konverter zur Erstellung passender Datensätze	37
5.1.1	Aufbau des Populationsdatensatzes	38
5.1.2	Aufbau des Substanzdatensatzes	39
5.2	MicroLabVR	39
5.2.1	Datenvisualisierung	39
5.2.2	Szenengestaltung	41
5.2.3	User-Tutorial	42
5.2.4	Datenmanagement über das Hauptmenü	43
5.2.5	Simulationsmenü	45
5.2.6	Interaktionsmöglichkeit mit den Organismen	47
5.3	Aspekte des UI- und UX-Designs	48
5.4	Performance-Optimierung	50
5.5	Probleme und Herausforderungen	52
6	Limit-Testing	54
6.1	Methodik	54
6.2	Ergebnisse	54
6.3	Fazit	56
7	Diskussion	57
8	Fazit und Ausblick	60
	Literatur	62
	Anhang	78

Abbildungsverzeichnis

1	Stöchiometrische Matrix	5
2	CBM Übersicht	7
3	Trackingmethoden	11
4	VR Template	13
5	Unity Transform Component	14
6	MatNet P. aeruginosa Biofilm	19
7	Simulationsgitter in BacArena	19
8	Sihumi Population Entwicklung in BacArena	20
9	Sihumi Ammonium Entwicklung in BacArena	21
10	P. aeruginosa GEM	24
11	Breadcrumb Navigation	26
12	Adobe Color Contrast Analyzer	27
13	Mapping Herdplatte Beispiel	29
14	Desktop-Kontextmenü	30
15	Pseudo-UML-Diagramm	36
16	R-Konverter	37
17	Organismus in Blender	40
18	Mesh einer Substanz	40
19	Flux-Outline-Darstellung	41
20	Szene	42
21	User-Tutorial	43
22	Hauptmenü	44
23	Simulationsmenü	46
24	Schwebeanimation	47
25	Farbschemata	49
26	Occlusion Culling	51
27	Fehlfärbung	53
28	Profiler Meta Quest 3	55

Abkürzungsverzeichnis

Abkürzung	Bedeutung
ABM	agentbasierte Modellierung (engl. agent-based modelling)
CBM	restriktionsbasierte Modellierung (engl. constraint-based modelling)
dFBA	dynamische Fluss-Bilanz-Analyse (engl. dynamic flux balance analysis)
FBA	Fluss-Bilanz-Analyse (engl. flux balance analysis)
FPS	Bilder pro Sekunde (engl. frames per second)
g DW	Gramm Trockengewicht (engl. gram dry weight)
GEM	genomweites Stoffwechselmodell (engl. genome-scale metabolic model)
HDRP	hochauflösende Render-Pipeline (engl. High Definition Render-Pipeline)
HMD	kopfmontiertes Display (engl. Head-Mounted Display)
mM	Millimolar (engl. millimolar)
SIHUMI	vereinfachte Mikrobiota des menschlichen Darms (engl. simplified human intestinal microbiota)
UI	Benutzeroberfläche (engl. User Interface)
URP	universelle Render-Pipeline (engl. Universal Render-Pipeline)
UX	Nutzererlebnis (engl. User Experience)
VR	virtuelle Realität (engl. virtual reality)
WCAG	Web Content Accessibility Guidelines

1 Motivation

Mikrobielle Gemeinschaften sind für das Leben auf der Erde essentiell. Ihre Vielfalt und Komplexität zu entschlüsseln ist entscheidend, um gesundheitliche und ökologische Probleme vorzubeugen [1]. Derzeitige „Omics“-Methoden, wie Metagenomik, Metatranskriptomik und Metaproteomik [2, 3, 4, 5, 6], um deren Aufbau und Funktionen zu untersuchen, sind kompliziert und aufwändig. Mathematische Modelle, welche die Entwicklung von Mikrobiomen berechnen können, haben sich an dieser Stelle als effektiv erwiesen, um komplexe Systeme wie Mikrobiome zu erforschen [7, 8, 9]. Derartige Simulationstools teilen sich vornehmlich in restriktionsbasierte und agentbasierte Modellierungssoftware auf, die entweder Stoffwechselnetzwerke in Gemeinschaften oder gesamte Interaktionsdynamiken simulieren können; manche vereinen sogar beide Ansätze [10]. Bisher lag der Fokus dieser Tools weniger auf der Visualisierung von Informationen. Mikrobiomdaten haben häufig mehr als zwei Dimensionen und lassen sich deshalb schwierig mit einer Desktop-Anwendung abbilden [11]. Darüber hinaus werden viele Informationen getrennt dargestellt. Werkzeuge, die lediglich als Programmbibliotheken verfügbar sind, bieten keine graphische Benutzeroberfläche [10, 12]. Wenn Programmierkenntnisse fehlen, grenzt dies die Usability (dt. Benutzerfreundlichkeit) für viele Wissenschaftler ein [13, 14, 15].

Aus diesem Grund wird Potential in Virtual Reality (VR)-Technologien gesehen, die zum einen Informationen in der dritten Dimension abbilden können und zum anderen Interaktionen mit den dargestellten Einheiten ermöglichen [16, 17]. Derartige Anwendungen sind häufig intuitiver und steigern zudem das Engagement¹ [18, 19].

1.1 Zielsetzung

Ziel dieser Arbeit ist die Konzeption und Entwicklung einer eigenständigen VR-Anwendung als Proof of Concept, genannt MicroLabVR, die raumzeitliche Mikrobiomdatensätze einlesen und visualisieren kann. Diese Anwendung soll auf die bestehenden Mängel in der Visualisierung und Benutzerfreundlichkeit aktueller Tools eingehen. Um geeignete Daten bereitzustellen, wird ein passendes Simulationstool zur Mikrobiommodellierung ausgewählt, das die Erstellung solcher Datensätze ermöglicht. Bei der Entwicklung der Anwendung liegt der Fokus auf der Umsetzung effektiver Methoden zur Visualisierung wissenschaftlicher Daten sowie auf der Einhaltung bewährter Prinzipien des User-Interface (UI)- und User-Experience (UX)-Designs, um die Benutzerfreundlichkeit zu optimieren.

¹Mehr zu Situated Learning: <https://ijiet.org/papers/48-R017.pdf> (letzter Zugriff 22.10.2024)

Es sollen die folgenden Forschungsfragen beantwortet werden:

- Welche Visualisierungsmethoden eignen sich am besten, um die verfügbaren Mikrobiomdaten in einer VR-Umgebung anschaulich darzustellen?
- Welche spezifischen Vorteile bietet eine VR-Anwendung im Vergleich zu herkömmlichen Desktop-Anwendungen für die Darstellung von Mikrobiomdaten?
- Welche zentralen Prinzipien des UI- und UX-Designs sind bei der Entwicklung einer VR-Anwendung zu berücksichtigen?
- Welche technischen Herausforderungen ergeben sich bei der Entwicklung einer solchen Anwendung, und welche Lösungsansätze sind dafür geeignet?

1.2 Inhalt

Kapitel 2 erläutert zunächst die für diese Arbeit relevanten biologischen und technischen Grundlagen. Im Anschluss wird der aktuelle Forschungsstand (siehe Kapitel 3) zu bestehender Software der Mikrobiommodellierung, zu gängigen Methoden der mikrobiellen Datenvisualisierung sowie zu wichtigen Verfahren und Leitlinien des UI- und UX-Designs aufgezeigt. Danach folgt die Konzeption in Kapitel 4. Hier wird beschrieben, wie die mikrobiellen Datensätze erzeugt und welche Methoden der Datenvisualisierung in der VR-Anwendung gewählt werden sollen. Darauf folgt die Implementierung der VR-Anwendung in Kapitel 5 und die damit verbundene Performance-Optimierung. Kapitel 6 beschreibt die Limitationen der Anwendung und welche Bedeutung diese für die Datensatzgröße haben. Anschließend wird die Arbeit mit einer Diskussion in Kapitel 7 fortgesetzt, um die Ergebnisse zu reflektieren. Die Arbeit schließt mit einem Fazit und einem Ausblick zur zukünftigen Forschung innerhalb der Thematik ab (siehe Kapitel 8).

2 Theoretische Grundlagen

Die zum Verständnis notwendigen Grundlagen lassen sich in biologische und technische Aspekte unterteilen. In diesem Zusammenhang werden grundlegende Konzepte behandelt, die für das gesamte Thema relevant sind. Zudem werden wichtige Fachbegriffe definiert, um ein einheitliches Verständnis zu gewährleisten.

2.1 Biologische Grundlagen

Hauptaugenmerk dieser Arbeit ist die Darstellung einer modellierten Gemeinschaft von Mikroorganismen. Diese Visualisierung soll helfen, das zugrundeliegende Mikrobiom zu analysieren und zu verstehen. Was jedoch ist ein Mikrobiom? Welche Rolle spielt es für den Menschen und seine Umwelt? Warum wird sich überhaupt mit ihm befasst und welche elementaren Methoden zur Modellierung gibt es?

2.1.1 Das Mikrobiom

In der Literatur über die Analyse von mikrobiellen Gemeinschaften werden die Begriffe Mikrobiom und Mikrobiota häufig synonym verwendet. Zur Fehlverwendung hat insbesondere das nachlässige Festhalten von Forschungsergebnissen beigetragen. Manche Fachartikel verwenden die Begriffe gleichwertig, obwohl definierende Unterschiede vorliegen [20]. Diese Arbeit folgt stets den nachstehenden Definitionen.

Die Mikrobiota bezeichnet die Gemeinschaft aller lebenden Mikroorganismen, wie Bakterien, Archaeen, Pilze, Algen usw., die das Mikrobiom bilden [20]. Es sei an dieser Stelle anzumerken, dass Meinungsverschiedenheiten darüber herrschen, ob organische Strukturen, wie Viren, auch zur Mikrobiota gehören, da sie keine *lebenden* Mikroorganismen sind [20, 21, 22]. Diese Diskussion erübrigt sich jedoch, sobald das gesamte Mikrobiom adressiert wird. Dieses umfasst nämlich die Gesamtheit aller Genome und Moleküle, mitunter Metabolite der Mikroorganismen sowie des Wirts. Diese Definition bezieht Viren und andere extrazelluläre Strukturen ein, weist sie aber nicht der Mikrobiota zu [22, 23].

2.1.2 Bedeutung der Mikrobiomforschung

Das soeben benannte steigende Forschungsinteresse am Verständnis von Mikrobiomen hat verschiedene Ursachen. In den vergangenen Jahrzehnten konnte die Wichtigkeit von mikrobiellen Gemeinschaften in der Natur, aber auch beim Menschen, zunehmend

festgestellt werden. Denn in allen Ökosystemen, sei es der Boden, das Wasser oder der Mensch, kommen Mikrobiome in unterschiedlichen mikrobiellen Zusammensetzungen vor. Von Bedeutung ist, dass der Wirt durch eine Störung der Gemeinschaft, resultierend in einer Dysbiose, an Erkrankungen und Störungen leiden kann. Dies wiederum stört das Gleichgewicht des betroffenen Ökosystems [1]. Hier spielt das Mikrobiom-Engineering eine tragende Rolle. Diese noch junge und sich entwickelnde Methodik erlaubt es, Mikrobiome zu modifizieren, um deren fehlendes Gleichgewicht wiederherzustellen, was, je nach Anwendungsgebiet, einen positiven Effekt auf die Gesundheit des Wirts haben kann [1]. Die Mikrobiomforschung selbst entwickelte sich aus der Untersuchung von Umweltmikrobiomen. Sie bildet eine vielfältige Grundlage an Einsatzbereichen, die viele Fachgebiete verbindet [22]. So sind beispielsweise für den Landwirtschaftssektor neue Erkenntnisse über Wechselwirkungen zwischen Pflanzenarten und ihrem Mikrobiom essentiell, um die landwirtschaftliche Produktivität zu steigern und mit der weltweit steigenden Nachfrage nach Nahrung mitzuhalten [24].

Ein weiterer Grund für das Forschungsinteresse an Mikrobiomen ist die Erkenntnis, dass die meisten Mikroorganismen wohltätige Aufgaben für den Menschen übernehmen und das Wirt-Mikroben-System gänzlich fördern. Mikrobiome lassen sich je nach Standort dem Darm, dem Mund, den Atemwegen und der Haut zuteilen. Deren Leistungen erstrecken sich von der Biokonversion der von uns aufgenommenen Nahrung und den daraus produzierten Ressourcen, bis hin zu dem Immunschutz vor Erregern [25, 26]. Das Verständnis der elementaren Prozesse dieser symbiotischen Interaktionsnetzwerke und der taxonomischen Zusammensetzung der Mikrobiota ist von großer Bedeutung, um Erkrankungen zu behandeln, die bei mikrobiotischer Dysbiose entstehen. Eine solche Fehlregulation kann zu abweichenden Körperfunktionen oder Krankheiten des kardiovaskulären und respiratorischen Systems oder weiteren schwerwiegenden Erkrankungen, wie Krebs, führen [26].

Eine tragende Rolle spielte dabei auch die Entwicklung von Untersuchungsmethoden, welche nicht auf die Kultivierung von Gemeinschaften angewiesen sind, wie beispielsweise der Hochdurchsatzsequenzierung, auch Next-Generation Sequencing genannt. Auf diese Weise war es erstmals möglich, auch Mikroorganismen zu untersuchen, die im Labor bisher nicht kultiviert werden konnten. Durch Erkenntnisse aus diesen Verfahren können Forscher nun potentiell mehr nicht-kultivierte Mikroben in den Bereich der Laboruntersuchungen bringen und so das Verständnis der mikrobiellen Vielfalt und Ökologie erweitern [27, 28]. Lange Zeit war die Erforschung dieser Themen zudem von der technischen Seite stark eingeschränkt. Im Zuge des stetigen digitalen Fortschritts, resultierend in steigender

Rechenleistung von Computern und der Entwicklung von lauffähigen Algorithmen, stehen der Forschung nun Messtechniken und Möglichkeiten zur mathematischen Modellierung zur Verfügung. Dies ermöglicht es, mikrobielle Populationen zu analysieren, ohne sie über Tage hinweg in einem Labor zu kultivieren [29, 30, 31].

2.1.3 Flux balance analysis

Eine gängige Methode zur Modellierung von Stoffwechselprozessen in Mikrobiomen ist die sogenannte flux balance analysis (FBA) [32]. Sie fällt in die Kategorie der restriktionsbasierten Modellierung (engl. constraint-based modelling, CBM). Mit ihr können Flüsse (engl. fluxes) des zellulären Stoffwechsels modelliert werden, um Zell-Zell-Interaktionen zu untersuchen [33, 34, 35] und Medikamente zu entwickeln, welche in den Metabolismus von Pathogenen eingreifen [36, 37, 38, 39]. Das Prinzip der FBA basiert auf der einfachen Darstellung von biochemischen Netzwerkreaktionen durch eine stöchiometrische Matrix (siehe Abbildung 1), in Verbindung mit Massenangaben zu beteiligten Metaboliten. Zusätzlich können weitere chemische Bedingungen, wie Substratkonzentrationen, einbezogen werden. Neben den einschränkenden Ober- und Untergrenzen für die Aufnahmefflüsse, die von den Konzentrationen abhängen, ist das Anstreben eines maximalen Biomassewachstums eine häufige Voraussetzung in den Modellen. Die Stoffwechselwege eines Organismus richten sich in diesen Fällen so aus, das Ziel zu optimieren [40].

Reaction Network	Stoichiometric Matrix
	R1 R2 R3 R4 R5 R6
R1: 2A + B → C + 3D	
R2: A + 3B → C + E	A
R3: A → 2B + D + E	B
R4: 4B → D + A	C
R5: D + 2B → C + 2E	D
R6: C + 4E → 3B + D	E

-2	-1	-1	1	0	0
-1	-3	2	-4	-2	3
1	1	0	0	1	-1
3	0	1	1	-1	1
0	1	1	0	2	-4

Abbildung 1: Ein Reaktionsnetzwerk mit der repräsentierenden stöchiometrischen Matrix. R_n sind die Reaktionen, Buchstaben A bis E sind die in den Reaktionen beteiligten Metabolite [41]

Die Simulation selbst erfolgt letztendlich unter der Annahme eines stabilen Zustands (engl. steady state) [32, 39, 42]. Dieser beschreibt das stabile Stadium eines Systems, wobei, unter den gegebenen Randbedingungen, die Konzentration der Metabolite konstant bleibt, sich also Produktion und Aufnahme ausgleichen [40]. In steady-state-Modellen werden somit kinetische Parameter von Reaktionen vernachlässigt (siehe Abbildung 2).

Das beschriebene Konzept befasst sich bis jetzt nur mit der Simulation einer Monokultur. Mikrobiome bestehen allerdings aus einer Vielzahl von Taxa. Eine Erweiterung des Modells auf die Community-Ebene erfordert die Berechnung einer Optimierungsfunktion der Flüsse auf mehreren Ebenen (Bi-level optimization). In solchen Fällen erfolgt beispielsweise die Optimierung der Flüsse der einzelnen Spezies und danach der Reaktionen der gesamten Community [43]. Im Vergleich zu echten Bedingungen ist die Simulation im steady state allerdings eher unrealistisch, da sich die Zustände in einem System unter ständigem Wandel befinden. Um die Dynamik darzustellen, wird bei Modellen der dynamischen FBA (dFBA) angenommen, dass das System aus mehreren aufeinanderfolgenden *quasi-stabilen* Zuständen besteht. Bei der dFBA wird pro Zeitschritt ein *quasi-stabiler* Zustand berechnet und die Änderungen zwischen diesen Zuständen modelliert, um die zeitliche Entwicklung des Systems zu simulieren. Änderungen in dem Kontext stellen beispielsweise Metabolitkonzentration und Flussraten dar. So lassen sich Entwicklungen im Substratverbrauch und dessen Produktion sowie das Populationswachstum analysieren (siehe Abbildung 2) [44].

Eine letzte Erweiterung berücksichtigt die räumliche Entwicklung einer mikrobiellen Gemeinschaft. Hierzu wird der Raum diskretisiert dargestellt. Es wird angenommen, dass innerhalb jeder Gitterzelle eine homogene Durchmischung vorliegt. Zwischen den Kompartimenten existieren Diffusionsparameter, die den Austausch von Substanzen untereinander ermöglichen [45]. Dadurch entstehen Flussgradienten, welche die Substratverteilung und -entwicklung im Raum beschreiben (siehe Abbildung 2) [10, 46].

Der Algorithmus durchläuft zwei Schritte: Zunächst wird der Zustand in jeder Gitterzelle aktualisiert, wobei die Aufnahme von Stoffen und die Biomasseproduktion berücksichtigt werden. Im Anschluss werden globale Prozesse berücksichtigt. Es können Stoffe zwischen den Gitterzellen diffundieren und Zellen migrieren. Die Menge an diffundierten Stoffen wird teilweise anhand der bestehenden Gradienten bemessen [45].

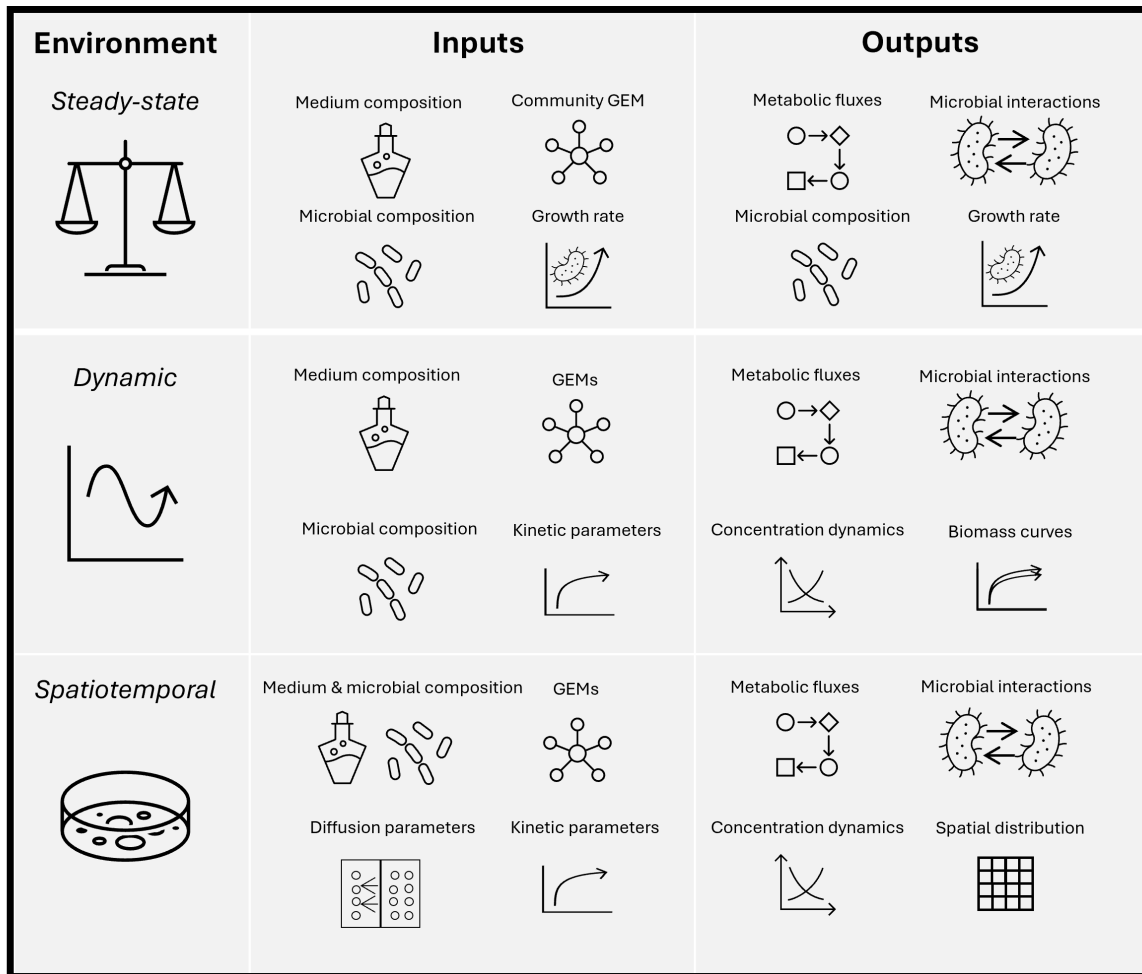


Abbildung 2: Eine Übersicht über die Arten von restriktionsbasierten Modellen (adaptiert von [47])

2.1.4 Agentbasierte Modellierung

Während FBA-Modelle eher der Lösung eines Optimierungsproblems von Populationen und häufig ganzer Gemeinschaften dienen, geht es bei der agentbasierten Modellierung (ABM) um die Simulation einzelner Organismen einer Population. Der Fokus liegt hier nicht auf der Untersuchung von mikrobiellen Stoffwechselprozessen, sondern auf der Analyse des Verhaltens und der Interaktionen der Teilnehmer. Mit der ABM werden Organismen als Individuen berücksichtigt, wodurch Kenntnisse über lokale Zwischendynamiken erzeugt werden können. Diese Individuen, auch Agenten genannt, sind eigene Datenobjekte mit eigenen Zuständen, an die gewisse Eigenschaften, Verhaltensweisen und Interaktionen geknüpft sind [48, 49, 50, 10]. Dies ist ebenso ein Unterschied zur FBA, welche lediglich auf stöchiometrischen Daten basiert. Wie Agenten sich in der

Simulation verhalten, wird von Regeln gelenkt, die dem System zugewiesen werden. Diese Rahmenbedingungen erlauben es, komplexe Dynamiken in einer Population mit den Eigenschaften der einzelnen Organismen zu untersuchen. Neben derartigen Dynamiken können auch raumzeitliche Prozesse wie Gradienten von Substraten analysiert werden [51, 52].

2.2 Technische Grundlagen

Die Umsetzung der Mikrobiomvisualisierung erfolgt mittels einer VR-Applikation. Während diese Technologie in vielerlei Hinsicht aus der Unterhaltungsbranche durch Videospiele bekannt ist, findet sie auch Beachtung in wissenschaftlichen Kontexten.

2.2.1 Virtual Reality und ihre Anfänge

VR bezieht sich auf die Simulation und zeitgleiche Wahrnehmung einer scheinbaren Realität und ihrer physischen Eigenschaften, welche in Echtzeit auf einem Computer generiert wird. Diese simulierte Wirklichkeit ist für den Träger eines Head-Mounted-Displays (HMD) interaktiv. Während einer derartigen Simulation ist der Anwender, je nach Setting, gänzlich visuell, ggf. auch auditiv von seiner realen Umgebung getrennt [53].

Erste Konzeptauffassungen von VR soll Stanley G. Weinbaums in seiner Science-Fiction Kurzgeschichte „Pygmalion’s Spectacles“ 1935 verfasst haben, wie manche Quellen berichten [54, 55]. Konkrete Entwicklungen waren erst zwei Jahrzehnte später mit Morton Heiligs „Sensorama“ zu finden. Das Sensorama war eine Art Kino-Kabinett für eine Person. Es zeigte kleine Kurzfilme und adressierte dabei alle fünf Sinne. Umgesetzt wurde diese Apparatur mit stereophonischen Lautsprechern und einem dreidimensionalen (3D)-Display für das Stereosehen. Zudem hatte das Sensorama eine vibrierende Sitzfläche und einen Ventilator sowie Gefäße, welche Gerüche versprühten [54, 56].

Der amerikanische Elektrotechniker und Informatiker Ivan Sutherland gilt als der Erfinder des ersten funktionstüchtigen HMDs, welches die digitale mit der realen Welt verknüpft. Er entwickelte es Ende der 1960er Jahre am Massachusetts Institute of Technology [57, 58]. Es war das erste HMD mit Head-Tracking. Allerdings war es so schwer, dass es von einer Montur an der Decke getragen werden musste. Deshalb hatte es auch den Namen „Sword of Damocles“ (dt. Damoklesschwert) [54]. Aus heutiger Sicht ist Sutherlands Erfindung in die Augmented Reality einzuordnen, da es das Blickfeld des Nutzers nur erweitert und nicht ersetzt.

Nach Weinbaums Kurzgeschichte trat der Begriff „Virtual Reality“ das erste Mal überhaupt 1982 in dem Science-Fiction-Roman „The Judas Mandala“ von Damien Broderick auf [59, 60]. In der Fachliteratur wird jedoch Jaron Lanier als Präger des Begriffs und Pionier der Technik genannt [53, 61, 62]. Als Pionierleistung in der Entwicklung der VR gilt das von ihm Ende der 1980er Jahre entwickelte EyePhone. Dieses HMD verfügt über zwei Liquid Crystal Display (LCD)-Monitore und ermöglicht es dem Träger, die virtuelle Umgebung stereoskopisch zu erleben [63]. Durch Eingabegeräte wie DataGloves und DataSuits kann der Träger mit dieser Umgebung interagieren. Laniers Innovationen waren wegweisend für spätere VR-HMDs [62].

2.2.2 Immersion

Das Ziel der VR ist eine vollständige Immersion des Nutzers. Immersion beschreibt den Effekt, virtuelle Reize als real wahrzunehmen, wodurch Stimuli aus der echten Welt ausgeblendet werden. Dadurch versinken die Nutzer vollständig in der Simulation [53, 64]. Dieser Effekt beschränkt sich allerdings nicht nur auf das Sehen und Hören, sondern auf alle Sinnesreize. Die übermittelten Signale der menschlichen Sinne beeinflussen zudem die Wahrnehmungen im Gehirn in unterschiedlichem Maße, was sich auf die Immersion auswirkt. Dabei haben das Sehen (83 %) und das Hören (11 %) den größten Einfluss auf die wahrgenommenen Sinneseindrücke. Anschließend folgen der Geruchssinn (3,5 %), der Tastsinn (1,5 %) und zuletzt die Signale, die durch den Geschmackssinn (1,0 %) entstehen [65]².

2.2.3 Präsenz

Mit Präsenz wird im Kontext der VR das maximale Gefühl von Immersion beschrieben. Der Nutzer empfindet sich in einem Stadium der vollständigen Integration in die Simulation und nimmt sie als die echte Realität wahr. Jedoch ist das Gefühl der Präsenz von mehreren Faktoren abhängig. So kann das verwendete VR-Equipment durch die Echtheit der übertragenen Sinnesreize überzeugen, die Simulation selbst jedoch das Erlebnis zunichtemachen. Die Forscher Lombard und Ditton [66] haben dabei folgende Kategorien festgelegt:

1. *Presence as social richness* beschreibt, wie gut soziale Interaktionen umgesetzt werden.

²Braem 2004, S. 192, Linxweiler 2004, S. 49 sowie Schubert/Hehn 2004, S. 1248 zitiert in [65]

2. *Presence as realism* beschreibt, wie realistisch Objekte und andere Komponenten der Realität dargestellt werden.
3. *Presence as transportation* beschreibt das Gefühl des Nutzers, an einen virtuellen Ort transportiert zu werden, wie es sich anfühlt, dass ein anderer Ort zu ihm gebracht wird, oder dass ihn weitere Nutzer an einen Ort begleiten.
4. *Presence as immersion* beschreibt, wie erfolgreich die verwendete Technik (HMDs, Controller, Audiokopfhörer etc.) die sensorischen Eindrücke übermittelt.
5. *Presence as social actor within medium* beschreibt, ob die Nutzer auf menschenähnliche Personen und Avatare reagieren, als wären sie reale soziale Akteure.
6. *Presence as medium as social actor* beschreibt, welche sozialen Reaktionen die Nutzer auf übermittelte Nachrichten dieser Medien reagieren; ähnlich dem bekannten „Turing-Test“ von Alan Turing [67].

Thornson et al. schreiben passend dazu: „Die Präsenz ist somit ein psychologisches Phänomen, das im menschlichen Geist stattfindet und nicht in der spezifischen Technologie. Die Technologie ist lediglich ein Mittel, um diesen Geisteszustand zu erreichen“ [68, Übers. d. Verf., S. 62].

2.2.4 Cybersickness

Cybersickness ist eine Form des Unwohlseins, welche manchen Nutzern von VR-Simulation widerfährt. Dabei empfinden sie Symptome wie Kopfschmerzen, Schwindel, Gleichgewichtsstörungen, Desorientierung, Übelkeit, Erbrechen, Belastung der Augen oder Schwitzen [69]. In der Regel verschwinden die Symptome nach Absetzen des VR-Headsets wieder, können aber auch Stunden [70] bis Tage [71] andauern.

Es bestehen mehrere Theorien, wie diese Krankheit hervorgerufen wird. Die am weitesten akzeptierte Theorie ist die „Sensory Conflict Theory“ [72]. Sie benennt die Ursache darin, dass ein Widerspruch zwischen dem Empfinden des Vestibularorgans und des visuellen Inputs herrscht. Eine Person, die in der VR-Simulation bewegt wird, nimmt die Bewegung nur visuell wahr, das Vestibularorgan verspürt jedoch keine physische Bewegung – ein Konflikt entsteht [69]. Entwickler können Cybersickness zu einem gewissen Grad vorbeugen, indem ihre Anwendungen flüssig, mit vielen Bildern pro Sekunde (FPS) laufen, und dynamische Bewegungen der Umgebung und des Nutzers reduziert werden. Auf diese Weise kann verhindert werden, dass eine Verzögerung zwischen Bewegung und Bild herrscht, welche andernfalls den ungewollten Effekt hervorrufen könnte [73].

2.2.5 Meta Quest 3

Um eine VR-Anwendung zu testen und zu debuggen (dt. fehlerbereinigen), muss während der Entwicklung ein geeignetes HMD verwendet werden. Zur Umsetzung dieser Arbeit wird die Meta Quest 3 in der 512 GB Speicherplatz-Variante von der Firma Meta benutzt. Sie besitzt zwei Pancake-Linsen mit je einer Displayauflösung von 2.064×2.208 Pixeln. Das Sichtfeld beträgt horizontal 110 Grad und vertikal 96 Grad. Sie verfügt über vier Aktualisierungsraten: 72 Hz, 80 Hz, 90 Hz und 120 Hz. Angetrieben wird sie von einem Snapdragon XR2 Gen 2-Prozessor der Firma Qualcomm [74, 75].

Die Meta Quest 3 ist eine Standalone-Brille und macht vom Inside-Out-Tracking Gebrauch. Derartige HMDs besitzen eingebaute Kameras, welche die Umgebung aufnehmen und anhand dieser Daten ihre Position und Ausrichtung erfassen (siehe Abbildung 3a). Dieses Verfahren ist anders als Brillen mit Outside-In-Tracking. Hier tracken im Raum fest montierte Basisstationen die Infrarotsignale der Brille und Controller (siehe Abbildung 3b). Dadurch wird der Simulationsraum von den Basisstationen begrenzt. Dieses Trackingverfahren wird in der Regel von PC-VR-Brillen genutzt, dessen Kabelverbindungen zum PC erneut die Bewegungsfreiheit einschränken [75, 76]. Deshalb wurde sich bewusst für die Nutzung der derzeitigen State-of-the-Art Standalone-Brille entschieden.

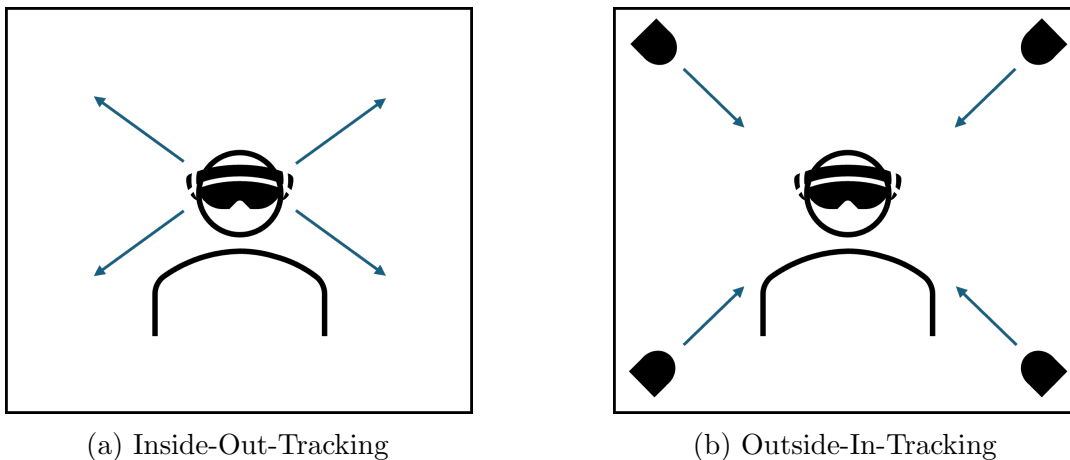


Abbildung 3: Vergleich der beiden Trackingmethoden: Inside-Out-Tracking vs. Outside-In-Tracking (eigene Darstellung)

Die Meta Quest 3 verfügt über einen Sicherheitsmechanismus, in dem die Größe und Position des Simulationsraums vordefiniert wird. Überschreitet der Nutzer diese festgelegte Grenze, schaltet die Brille den VR-Modus ab und wechselt zum „See-Through-Modus“, sodass die reale Umgebung gezeigt wird. Diese Methode soll Zusammenstöße mit Objekten verhindern, um so Hardware- und insbesondere menschlichen Schaden zu vermeiden.

2.2.6 Blender

Blender ist eine open source 3D-Graphiksoftware zur Modellierung von 3D-Modellen und zur Erstellung von Texturen, Animationen und ganzen Simulationen. Die Software wird häufig im Bereich der VR-Entwicklung angewandt, um benutzerdefinierte Modelle zu erstellen, welche in verschiedene Dateiformate exportiert und in Unity oder in anderen Engines importiert werden können [77].

2.2.7 Unity Game Engine

Die Unity Game Engine ist eine Echtzeit-Entwicklungsplattform für zweidimensionale (2D)- und 3D-Anwendungen [78]. Sie ermöglicht die Entwicklung von Anwendungen für Plattformen wie Windows, iOS, Linux, Android und auch VR-Geräte. Unity bietet eine Graphik-Engine, welche das Rendering von 2D- und 3D-Graphiken in Echtzeit erlaubt. Die Engine enthält vorgefertigte *Render Pipelines*, die, je nach Belieben und Anforderungen an die Anwendung, ausgewählt werden können und sich auf die Performance und Echtheit der Graphik auswirken [79]. In Unity ist ebenfalls eine Physik-Engine integriert, welche beispielsweise Kollisionen und Bewegungen von Objekten realistisch umsetzt [80]. *Scripts* (dt. Skripte) werden in Unity genutzt, um das Verhalten von Objekten und Regeln in der Simulation zu steuern [81]. Sie werden in der Programmiersprache C# geschrieben. Der *Unity Asset Store* ermöglicht dem Entwickler, eine Vielzahl vorgefertigter *Assets* zu integrieren. Dazu gehören 2D- oder 3D-Modelle, Texturen, Sounds, Add-Ons und Templates [82].

Unity Scenes

Scenes (dt. Szenen) in Unity sind die Umgebungen, in denen sich der Nutzer während der Simulation befindet. Sie enthalten alle Bausteine und *Assets* der Umgebung [83]. Eine Unity-Anwendung kann aus einer oder mehreren *Scenes* bestehen, die untereinander verknüpft sind. Standardmäßig enthält eine neue *Scene* lediglich eine Kamera und eine Lichtquelle. Weitere *GameObjects* (siehe Kapitel 2.2.7) können je nach Anforderungen hinzugefügt werden. Unity bietet auch *Template-Scenes* für Projekte an. Als VR-Template (siehe Abbildung 4) liefert Unity beispielsweise ein Tutorial Framework mit vorinstallierter *Universal Render Pipeline* (URP), welche als performante *Render Pipeline* für VR-Anwendungen gilt sowie Software-Packages wie dem *XR Plug-in Management* (siehe Kapitel 2.2.7) und dem *XR Interaction Toolkit* (siehe Kapitel 2.2.7) [83].

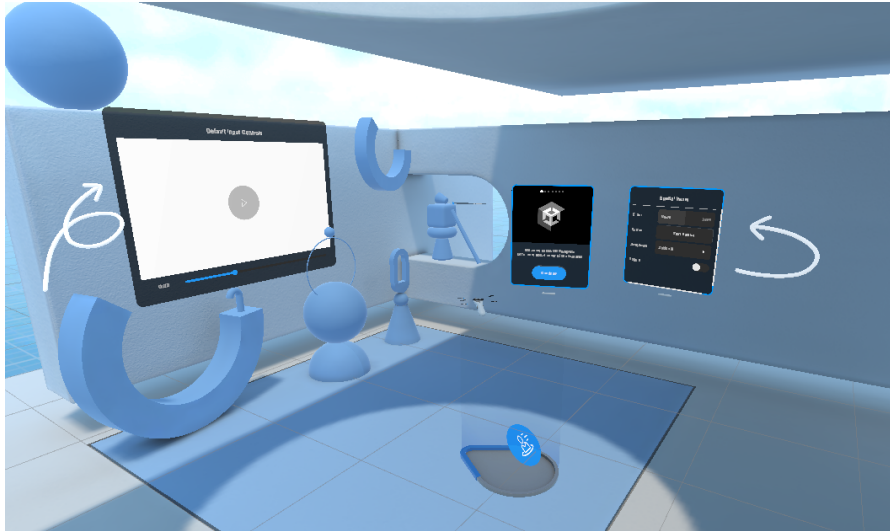


Abbildung 4: Unity VR-Template mit *GameObjects* und UI-Elementen, mit konfigurierten Interaktionsmöglichkeiten (eigene Darstellung)

Unity GameObjects

Alle Objekte, die in einer *Scene* vorkommen, sind Instanzen der Klasse *GameObject* [84]. Die Datenstruktur ermöglicht die Verwaltung dieser Objekte über *Components* (dt. Komponenten, siehe Kapitel 2.2.7). *Components* können beispielsweise das Aussehen und das Verhalten von *GameObjects* steuern. Entwickler können durch Methoden der Klasse *GameObject* die *GameObjects* der *Scene* manipulieren und bearbeiten, indem sie ihren Status ändern, dessen *Components* und Werte anpassen und sie miteinander kommunizieren lassen [84].

Unity Components

Funktionelle Eigenschaften von *GameObjects* werden in Unity von *Components* verwaltet [85]. Ein *GameObject* kann mehrere *Components* haben. Die Funktionen von *Scripts* werden an *GameObjects* übertragen, indem sie als *Components* hinzugefügt werden. Weitere *Components* können z. B. Audioquellen, *Materials* (dt. Materialien) und ein sogenannter *Rigid Body* (dt. starrer Körper) sein, wenn das *GameObject* von der Physik-Engine beeinflusst werden soll.

Eine wichtige *Component* ist der *Transform*. Hiervon kann ein *GameObject* nur eins besitzen. Der *Transform* regelt die Position, Rotation und Skalierung der Objektinstanz (siehe Abbildung 5). Er unterscheidet sich insofern von anderen *Components*, als dass ein *GameObject* nicht mehrere *Transforms* besitzen und er auch nicht entfernt werden kann.

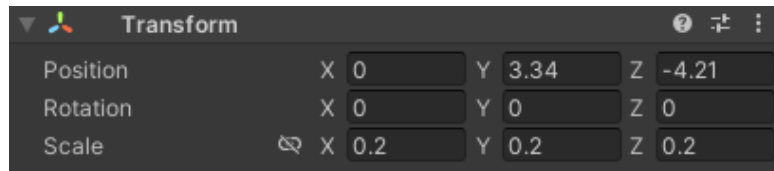


Abbildung 5: Die Transform *Component* eines *GameObjects* (eigene Darstellung)

XR Plug-in Management

Das XR-Plugin-Management ist ein Softwarepaket von Unity, welches das Unity-Projekt für die Integration und Nutzung von VR-Hardware befähigt. Es liefert dem Entwickler zudem eingebaute UI-Elemente mit Schnittstellen zu VR-Controller-Eingaben [86]. Für die Meta Quest 3 sind zusätzlich die Installationen der Meta XR Unity Package Manager Pakete aus dem *Asset Store* nötig, welche das veraltete Oculus Integration Paket ablösen.

XR Interaction Toolkit

Für Interaktionen mit Benutzeroberflächen und *GameObjects* kann das XR Interaction Toolkit genutzt werden [87]. Dieses Plug-in liefert ein Framework für Controllereingaben von VR-Geräten, die an mehrere Hersteller angepasst sind. Über konfigurierte *Scripts*, die den *GameObjects* als *Component* hinzugefügt werden können, werden dem VR-Nutzer das Auswählen und Greifen der 3D-Objekte, verbunden mit vibrotaktilen Feedback, ermöglicht. Das Add-on bietet ein einfaches Setup der VR-Spielerkomponente, dem *XR Rig*. Controller bekommen hierbei einen *Ray Caster* zugewiesen, welcher Interaktionen mit 3D-Objekten und UI-Elementen indiziert [87].

Unity User Interfaces

In Unity besehen UIs aus mehreren *GameObjects*. Den Grundbaustein bildet ein *Canvas* (dt. Leinwand), welches alle weiteren UI-Komponenten als hierarchische Child-Objects enthält [88]. Je nach Zielsetzung können verschiedene Elemente in dem Bereich des *Canvas* platziert werden. Diese Elemente unterteilen sich in visuelle Komponenten und interaktive Komponenten. Textfelder und Bilder gehören zu den visuellen Komponenten. Sie eignen sich dazu, andere UI-Elemente zu beschreiben, ihnen Symbole zuzuschreiben, Informationen zur Anwendung zu liefern, oder den Hintergrund der Benutzeroberfläche anzupassen [89]. Interaktive Komponenten sind *Buttons* (dt. Knöpfe), welche Methoden von *Scripts* aufrufen können, die das UI und Funktionen der Simulation steuern. *Toggles* (dt. Kippschalter), besser bekannt als Checkboxes, können nur die binären Werte „ist

an“ (*true*) oder „ist aus“ (*false*) annehmen. In *Scripts* können auf diese Weise Aktionen durchgeführt werden, die auf der Wertänderung der *Toggles* basieren. *Slider* (dt. Schieber) nehmen einen numerischen Wert zwischen einem definierten Minimum und Maximum an. Hierbei kann ausgewählt werden, ob der aktuelle Wert kontinuierlich oder in diskreten Abständen gewählt werden kann. Ebenso wie beim *Toggle* kann auf Wertänderungen reagiert werden. Auf ähnliche Weise funktionieren *Dropdowns* (dt. Auswahllisten), deren Wert sich auf eine Liste von Optionen beschränkt. Diese erläuterten UI-Komponenten können innerhalb einer *Scroll View* (dt. Bildlaufansicht) positioniert werden, um die Elemente auf einer begrenzten Fläche verfügbar zu machen [90].

3 Aktueller Stand von Forschung und Technik

Die Forschung an mikrobiellen Populationen in den vergangenen Jahrzehnten hat die Bedeutung von Mikrobiomen für die Umwelt und die Gesundheit des Menschen gezeigt [10]. Das menschliche Mikrobiom ist ein lebenswichtiger Teil unseres Organismus und verantwortlich für den Erhalt unserer Gesundheit. Abnormale Zusammensetzungen werden mit verschiedenen Erkrankungen des Menschen assoziiert. So kann eine Störung des Mikrobioms zu Übergewicht, Krebs, kardiovaskulären Erkrankungen, Parkinson oder auch Depressionen führen [91].

Bereits im neunzehnten Jahrhundert war den Mikrobiologen Robert Koch und Louis Pasteur bekannt, dass verschiedene Spezies für infektiöse Krankheiten ursächlich sind. Der Fokus auf die gesundheitlichen Auswirkungen von Systemen verschiedener Mikroben und ihrem Zusammenspiel in einer Gemeinschaft wurde erst Anfang der 2000er gelegt [92].

Derzeitige Methoden, um Kompositionen von Mikroorganismen und deren Funktion zu untersuchen, sind unter anderem die Metagenomik [2, 3], die funktionelle Metagenomik [93], die Metatranskriptomik [4, 5], die Metaproteomik [6] und die Metabolomik [7, 94]. Viele dieser Methoden erlauben nur eingeschränkte Kenntnisse über die zugrundeliegenden Funktionen des betrachteten Mikrobioms. Insbesondere individuelle und gemeinschaftliche Stoffwechselfunktionen der Spezies und Wechselwirkungen untereinander werden selten ermittelt. An dieser Stelle haben mathematische Modelle ihr Potential erwiesen, um komplexe Systeme, wie Mikrobiome, zu untersuchen [7, 8, 9].

3.1 Software zur Mikrobiommodellierung

3.1.1 Restriktionsbasierte Modellierungssoftware

Modelle im Kontext der CBM haben sich bewährt, Stoffwechselnetzwerke von mikrobiellen Gemeinschaften zu simulieren. Sie bieten die Möglichkeit, Interaktionen zwischen beteiligten Mikroorganismen und dem Wirt zu untersuchen und Reaktionswege zu verstehen [95].

Genomweite Stoffwechselmodelle (GEMs) bilden die Grundlage für restriktionsbasierte Modelle. Sie beinhalten alle biochemischen Reaktionen eines Organismus und dienen als eine mathematische Repräsentation dessen Stoffwechselnetzwerks. Die stöchiometrische Matrix, wie sie in Kapitel 2.1.3 zur FBA erläutert wurde, bildet die wichtigste Komponente eines solchen GEMs. Diese mathematische Darstellung hält die stöchiometrischen Beziehungen zwischen den Metaboliten und den biochemischen Reaktionen fest [7].

Während sich manche Modelle auf die Optimierung einer Spezies fokussieren, erweitert MICOM (kurz für: Microbial Community) die Herangehensweise der Stoffwechsellmodellierung auf die Gemeinschaftsebene [96]. Das Modell berücksichtigt unter anderem Abundanzen, um den Verbrauch von Metaboliten präziser darzustellen. Denn die logische Schlussfolgerung ist, je abundanter eine Spezies, desto größer ist die Aufnahme von präsenten Stoffen und die Abgabe von Endprodukten. Für eine bessere Vorhersage wurden des Weiteren realistische Informationen über die Ernährung einbezogen. Die Optimierung erfolgt, wie in Kapitel 2.1.3 erläutert, auf zwei Wegen. MICOM optimiert nicht nur das Wachstum einzelner Mikroben, sondern das kollektive Wachstum der Gemeinschaft. In dieser Berechnung wird die Teilung von Ressourcen untereinander sowie der Austausch von Endprodukten berücksichtigt (Cross-Feeding) [96].

Diener et al. [96] testeten ihr Modell an Daten von 186 Testpersonen unter der Annahme einer typisch westlichen Ernährung. Die Kohorte bestand aus Diabetespatienten und einer Kontrollgruppe. Das Ergebnis war, dass Bakterien entlang der Gruppen starke Variationen im Wachstum aufwiesen. Zwischenmikrobielle Interaktionen waren spezifisch für die jeweiligen Gemeinschaften. Es überwogen kompetitive Wechselbeziehungen, was auf Konkurrenz um begrenzte Ressourcen hinweist. Das Modell zeigte zudem, dass Darmspezies, welche an der Aufrechterhaltung der Gesundheit beteiligt waren, eine dominierende Menge an Interaktionen aufwiesen. Zudem wurde unter den Diabetikern die erwartete geringere Produktion von kurzkettigen Fettsäuren festgestellt.

Zwar ermöglicht die FBA die Vorhersage metabolischer Flüsse und Wachstumsraten innerhalb einer mikrobiellen Gemeinschaft unter der Annahme eines steady-states, jedoch erlaubt dies nicht die Betrachtung der dynamischen Veränderungen, die üblicherweise in einem Mikrobiom vorkommen. COMETS (kurz für: computation of microbial ecosystems in time and space) hat die dFBA (siehe Kapitel 2.1.3) als Grundlage, wodurch sich zwar die metabolischen Abläufe innerhalb der Zelle im steady-state befinden, jedoch die Abundanzen der Organismen und das Aufkommen der Metabolite dynamisch sind. Auf diese Weise können zeitliche Veränderungen in der Population beobachtet und studiert werden [97].

Zu den Funktionen von COMETS gehört in erster Linie die Simulation des Bakterienwachstums. Diese Funktion wurde gezeigt, indem die Entwickler den bekannten Versuch von Varma und Palsson [98] zur Untersuchung von *Escherichia coli* nachbildeten. COMETS erlaubt ebenfalls die Simulation in Chemostat, wodurch gleichbleibende Bedingungen gewährleistet werden. Ein weiteres Feature ist, dass die Eigenschaften der Umgebung geändert werden können, wie z. B. die Berücksichtigung des Tag-Nacht-Zyklus. Dies

bietet sich besonders an, um Organismen zu untersuchen, welche Photosynthese betreiben [99]. Weitere Prozeduren bieten z. B. die Simulation mit extrazellulären Reaktionen und die Simulation von evolutionären Prozessen [97].

Eine Limitation von COMETS ist, dass es die Einschränkungen der FBA, wie die fehlende Genregulation, übernimmt. Dadurch ist es nicht möglich, Änderungen an der Genexpression dynamisch vorzunehmen. Ebenfalls können die fix zu wählenden Zeitschritte numerische Fehler verursachen. Zur Minimierung der Fehleranfälligkeit sollten die Zeitschritte daher klein genug gewählt werden. Eine Simulation dauert typischerweise 30 Minuten bis wenige Stunden. Werden die Parameter jedoch zu klein gewählt, können die Simulationszeiten mehrere Tage betragen [97].

3.1.2 Agentbasierte Modellierungssoftware

Wie bis jetzt gezeigt wurde, hat sich die CBM bewährt, Stoffwechselprozesse mikrobieller Gemeinschaften zu untersuchen und zu verstehen. Dabei spielt insbesondere die FBA eine tragende Rolle. Anwendungsfälle der ABM beziehen sich eher auf komplexe Interaktionsdynamiken zwischen Mikroorganismen, da sie die Aktivitäten einzelner Individuen der Population erfassen [10].

MatNet kombiniert die beiden Modellierungsmethoden CBM und ABM, wodurch sich lokale Interaktionen von Mikroorganismen in einer Gemeinschaft simulieren lassen. Dabei beschränkt sich das Tool auf Monokulturen. MatNet ist eine Erweiterung von MATLAB-NetLogo und bringt neue Funktionen für den Datenaustausch zwischen NetLogo und MATLAB [12].

Am Beispiel des Pathogens *Pseudomonas aeruginosa* zeigten Biggs und Papin [12] die Wichtigkeit einer solchen Softwareerweiterung. Sie reproduzierten die erforschte Struktur des Biofilms von *P. aeruginosa*, welche durch die eingeschränkte Diffusion von Sauerstoff entsteht. Abbildung 6 zeigt die Entwicklung und den Aufbau des Biofilms. Sie verdeutlicht den Abfall von Sauerstoff, je tiefer in die Schleimschicht eingedrungen wird. Nur an der sauerstoffreichen, blau gefärbten Oberfläche, hätten Zellen die Möglichkeit, Proteinbiosynthese zu betreiben.

Bauer et al. [10] haben sich ebenfalls der Aufgabe gewidmet, beide Ansätze zu verknüpfen und das R-Paket BacArena³ veröffentlicht. Das Paket erweitert den Ansatz von MatNet, polykulturelle Gemeinschaften zu modellieren.

³<https://cran.r-project.org/src/contrib/Archive/BacArena> (letzter Zugriff 01.06.2024)

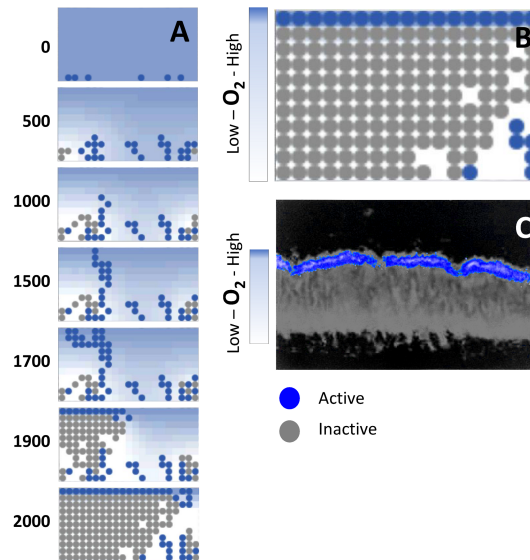


Abbildung 6: (A) Wachstum des Biofilms (Zeitpunkte entsprechen 5-Minuten-Intervallen) (B) Zeitpunkt 2000 des Biofilmmodells (C) In-vitro-Querschnitt des *P. aeruginosa*-Biofilms (Biggs und Papin modifizierten die Abbildung von Xu et al. [100]) [12]

Der Simulationsbereich von BacArena ist ein zweidimensionales Gitter, auf dem die einzelnen Mikroorganismen verteilt werden (siehe Abbildung 7). Während der Simulation kann sich die Verteilung der Organismen zwischen den zeitlichen Abschnitten ändern, indem sie migrieren, sterben, oder sich in einer benachbarten Gitterzelle duplizieren. Auf diese Weise werden zeitliche Dynamiken des Mikrobioms dargestellt [10].

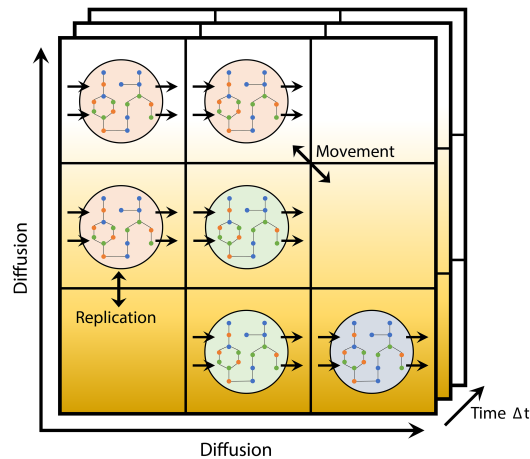


Abbildung 7: Die einzelnen Organismen sind auf einem 2D-Gitter verteilt [10]

Der Umgebung sind Substanzkonzentrationen zugeordnet. Diese können in einem Stoffwechselprozess, je nach Zustand des Organismus, aufgenommen werden. Organismen können auch selbst Biomasse produzieren. Der Stoffwechsel wird für jeden Agenten in der Simulation mittels der FBA und dem zugehörigen GEM berechnet, somit auch die Menge an produzierter Biomasse. Das Paket erlaubt den Import etablierter oder benutzerdefinierter GEMs. Die Voraussetzung ist, dass die Modelle auf der Systems Biology Markup Language basieren. Die Biomasse wird aus der Konzentration der Metabolite in der Umgebung und der Position des Organismus im Simulationsbereich berechnet. Die Metabolitkonzentration ändert sich an der betroffenen Stelle und berechnet sich aus der Differenz von aufgenommenen und abgegebenen Metaboliten. Dadurch entsteht pro Agent ein eigenes Stoffwechselprofil, ein sogenannter „metabolischer Phänotyp“. Auf diese Weise lässt sich auf die gegenseitige Versorgung mit Metaboliten zwischen den Individuen zurückgreifen [10].

Alle Simulationsschritte können visualisiert werden, um sie mit bereits erhobenen Daten aus Experimenten zu vergleichen. Abbildung 8 zeigt beispielsweise die Entwicklung von acht Darmbakterien, welche acht Stunden lang mit einer westlichen Diät gewachsen sind. Jedes Symbol stellt ein Taxon dar. Passend dazu zeigt Abbildung 9 die Entwicklung der Konzentration von Ammonium.

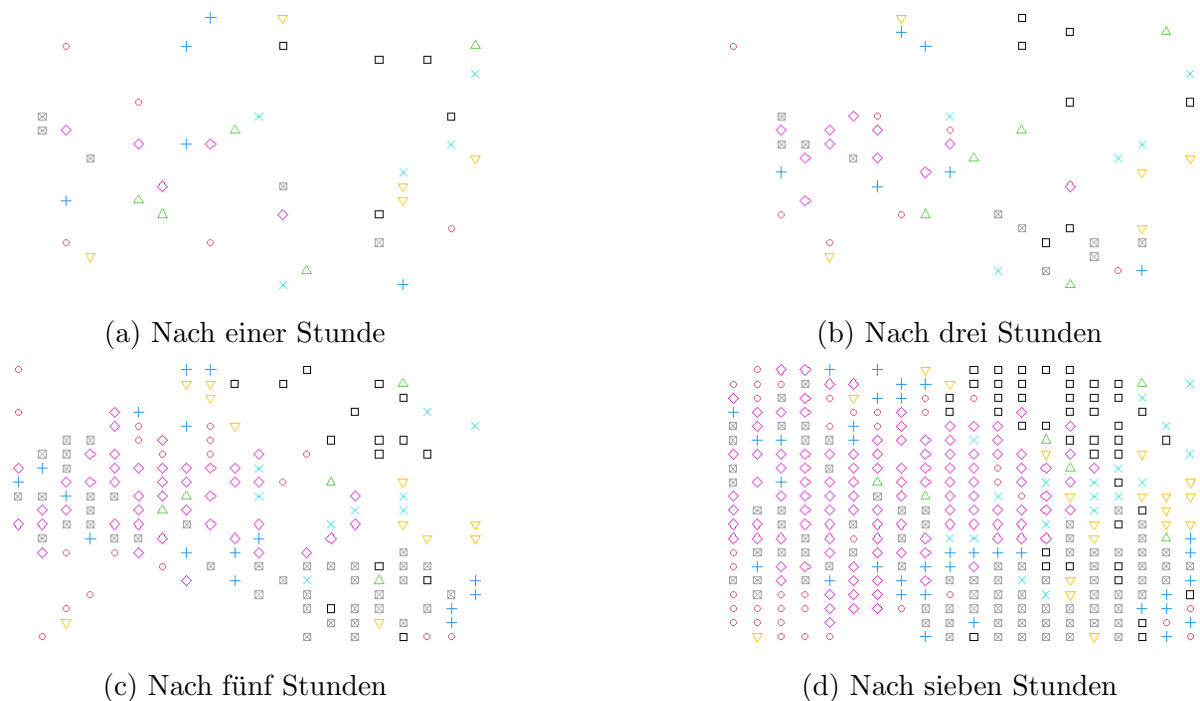


Abbildung 8: Die Entwicklung von acht Darmbakterien, die acht Stunden lang mit einer westlichen Diät wachsen (eigene Darstellung, erstellt mit BacArena)



(a) Nach einer Stunde



(b) Nach drei Stunden



(c) Nach fünf Stunden



(d) Nach sieben Stunden

Abbildung 9: Die Entwicklung der Ammoniumkonzentration im selben Simulationsbereich. Je dunkler die Farbe, desto höher die Konzentration (eigene Darstellung, erstellt mit BacArena)

Die Entwickler führen an, dass BacArena im Vergleich zu MatNet die Simulation einer Gemeinschaft erlaubt, welche aus einer Vielzahl von Arten besteht und sich somit nicht nur auf eine Spezies begrenzt. Zusätzlich besteht die Funktion, Kinetiken für Austauschreaktionen nach Monod zu setzen [101, 102]. Ein weiteres Merkmal ist, dass bei den Simulationen Phänotypen berücksichtigt werden, was die Heterogenität der mikrobiellen Gemeinschaften berücksichtigt. Die Laufzeit verhält sich nach Herstellerangaben linear und bleibt ab einer Anzahl von 50 Spezies konstant. Unterstützt wird die Rechenzeit durch eine parallele Verarbeitung der Daten. Im Gegensatz zu MatNet, bietet BacArena als R-Paket kein graphisches UI [10]. Alle Befehle und Funktionen von BacArena sind dokumentiert ⁴.

Bauer et al. [10] berichteten im selben Paper zur Entwicklung von BacArena über dessen Anwendung zur Modellierung der Biofilmbildung von *P. aeruginosa*. Sie zeigten die differenzierte Anordnung von Individuen mit unterschiedlichen metabolischen Phänotypen, resultierend aus der zugrundeliegenden Nährstoffverteilung. Die Phänotypen verteilten sich in Areale, abhängig vom Aufkommen von Glukose, Sauerstoff, Azetat, Succinat und

⁴<https://rdrr.io/cran/BacArena/man/> (letzter Zugriff 04.07.2024)

Kohlenstoffdioxid. Das Aufkommen der Phänotypen erfolgt in allen Simulationsschritten und die Populationsgröße korrelierte mit höheren Stoffkonzentrationen. Im selben Zuge zeigten sie auch Cross-Feeding zwischen manchen Organismen. Genauer fanden sie heraus, dass Endprodukte des Fermentationsprozess mancher Phänotypen zugunsten von anderen Mitgliedern verbraucht werden und dadurch dessen Wachstum fördern [10].

In einer weiteren Simulation untersuchten sie das Verhalten von sieben Bakterienarten, welche eine vereinfachte menschliche Darmmikrobiota darstellen sollten. Die Simulation führte eine ungleichmäßige Verteilung von Schleimglykanen auf. Diese bewirkte, dass bestimmte Bereiche entstanden, die besonders gut für Bakterien geeignet waren, die diese Glykane abbauen konnten. Diese Bakterien fanden dort eine optimale Umgebung, in der sie wachsen und sich vermehren konnten. Ihre Untersuchungen deuten folglich darauf hin, dass Gradienten, welche durch Stoffwechselprozesse entstehen, eine wesentliche Bedeutung für die Zusammensetzung der Mikrobiota des Darms haben. Bauer et al. schlossen daraus: „Dies hat wichtige Auswirkungen für das Verständnis der Schleimschicht und legt nahe, dass diätetische oder metabolische Behandlungen im Falle einer gestörten Schleimhautmikrobiota relevanter sein könnten als Immunsuppressiva“ [10, Übers. d. Verf., S. 10].

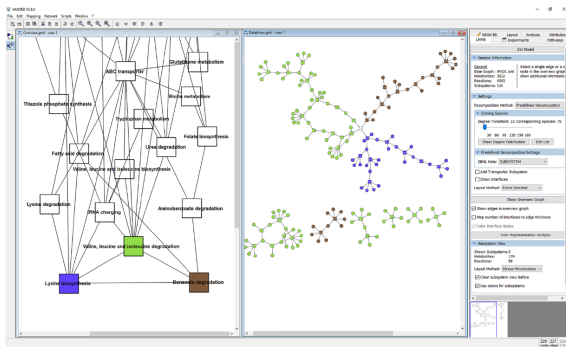
3.2 Mikrobielle Datenvisualisierung

Um Daten besonders in Zeiten von „Big Data“ interpretieren zu können, ist es wichtig, sie adäquat darzustellen. Die richtige Visualisierung von Daten ist der Schlüsselweg, um die Wahrnehmung des Betrachters zu verbessern [16]. Gerade räumliche Daten, welche den Aufbau eines Mikrobioms beschreiben, können auf einem Computerbildschirm nur zweidimensional angezeigt werden. Ein Diagramm, welches den Darstellungsraum erweitert, ist die sogenannte Heatmap (siehe Abbildung 9). Heatmaps haben den Vorteil, dass sie nicht nur eine Flächenverteilung repräsentieren können, sondern die Färbung der Datenpunkte einen Intensitätswert indiziert. Nachteilig ist dabei, dass Relationen zwischen den Intensitäten häufig nicht präzise interpretiert werden können.

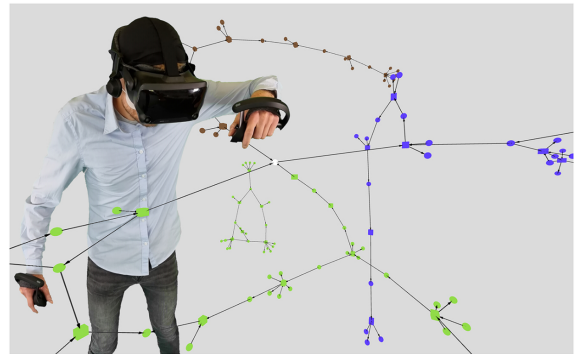
Moore et al. [103] adressierten diese typische Restriktion von Heatmaps. Sie schrieben in ihrem Paper: „Die dritte Dimension bietet zusätzliche Informationsebenen, die mit einer herkömmlichen 2D-Heatmap nicht visualisiert werden können“ [103, Übers. d. Verf., S. 154]. Deshalb entwickelten sie eine Software, welche die Analyse von mehrdimensionalen Daten über das menschliche Mikrobiom ermöglicht. Sie integrierten in ihre Anwendung auch die Option einer Stereoansicht, welche eine intensivere Tiefenwahrnehmung mit 3D-fähigen Monitoren erlaubt.

Die Unterscheidung zwischen Korrelation und Kausalität ist eine zentrale Herausforderung bei der Analyse von Daten. Wie Garg et al. [104] betonen, ist diese Differenzierung entscheidend, um Krankheiten zu verstehen und adäquate Behandlungen zu schaffen. Wie sie am Beispiel der Mukoviszidose veranschaulichten, kann sich die mikrobiotische Verteilung in Organen auf die Schwere der Infektionserkrankung auswirken. Verschiedene Omics-Methoden können in solchen Fällen verwendet werden, um die Struktur des Mikrobioms sowie dessen Wechselwirkung mit dem Wirt zu erforschen. Diese Techniken nutzten sie unter anderem zur Verbindung räumlicher Mikrobiomkarten und anatomischen Modellen einer Lunge. Das Mikrobiom projizierten sie unter der Verwendung verschiedener Softwareprogramme auf das Computertomographiebild eines Patienten im Endstadium der Krankheit. Ihre Resultate erwiesen, dass die Wirkung von pharmazeutischen Behandlungen und das Auftreten von Pathogenen innerhalb der Lunge variieren. Ebenso trifft dies auf die Folgen von diversen Umweltbelastungen zu. Sie vermuten, dass derartige Differenzen ursächlich für die Schwere der Erkrankung innerhalb der Lunge sein können [104]. Mit ihrer Arbeit veranschaulichten Garg et al., wie Datenvisualisierung helfen kann, Interaktionen zwischen Organismen und dem Wirt zu verstehen.

Anwendungen wie diese bieten den Vorteil der Visualisierung in 3D, wodurch räumliche Bezüge besser hergestellt werden können. Was diesen Desktop-Anwendungen jedoch fernbleibt, ist die Immersion (siehe Kapitel 2.2.2). VR-Simulationen erzeugen den Effekt der Immersion und erlauben dem Nutzer eine bessere Wahrnehmung der Datenräume. Diese kann beim Menschen zu einem besseren Verständnis der Daten und dessen Beziehungen führen [16]. Eine bessere Interpretation von räumlichen Bezügen ist besonders bei komplexen bimolekularen Netzwerkdaten, wie GEMs, wichtig. Denn die Datensätze gewinnen mit weiteren Einblicken in biologische Prozesse, wie durch beispielsweise Hochdurchsatzsequenzierungsmethoden, an Größe und erschweren somit dessen Auswertung und Analyse. Aichele et al. [17] entwickelten deshalb eine hybride Anwendung, welche sowohl eine Desktopumgebung, als auch eine VR-Schnittstelle bietet (siehe Abbildung 10). Auf diese Weise wollten sie die Vorteile beider Werkzeuge nutzen. Sie wollten zum einen eine bessere 3D-Darstellung bieten und zum anderen den Anwender durch kürzere VR-Nutzung vor Cybersickness bewahren. Die Desktopumgebung ist mit biochemischen Datenbanken verknüpft und erlaubt die Erstellung von Abbildungen. Netzwerkstrukturen von Stoffwechselmodellen eines Organismus können in der VR-Umgebung interaktiv visualisiert werden. Die Darstellung kann z. B. hierarchisch manipuliert und skaliert werden, sodass die Daten aus verschiedenen Perspektiven betrachtet werden können, um das räumliche Gedächtnis besser anzusprechen [17].



(a) Desktopumgebung



(b) VR-Umgebung

Abbildung 10: Die Darstellung des GEMs von *P. aeruginosa* [17]

Die Limitationen von Desktopanwendungen wurden ebenfalls in einer Studie von Pirch et al. [11] adressiert. Sie beriefen sich insbesondere auf die Komplexität der Netzwerke, welche auf Computermonitoren schnell wie unübersichtliche Cluster aussehen, wodurch die Auswertung nahezu unmöglich ist. VRNetzer wurde entwickelt, um die Visualisierung und Analyse solcher Netzwerke zu verbessern, da VR-Technologien eine bessere Tiefenwahrnehmung ermöglichen. Denn so können selbst Knoten, die auf einer 2D-Ebene verdeckt werden, in einer 3D-Ansicht visuell auseinandergehalten werden. Ihre Plattform stützt sich dabei auf die typische Darstellung mit Knoten und verbindenden Kanten, welche einfach zu interpretieren sind. Die Inhalte selbst sind erweiterbar und können benutzerdefiniert angepasst werden. So können externe Datenbanken mit vielfältigen Datentypen importiert werden und die Benutzeroberfläche nach eigenem Belieben modifiziert werden [11].

Als Proof of Concept zeigten sie, wie VRNetzer zur Erkundung molekularer Netzwerke Anwendung finden kann, um Verbindungen zwischen Genen und diversen Krankheiten zu untersuchen. Pirch et al. argumentierten, dass Zusammenhänge zwischen molekularen Daten häufig nicht in einer Region erkannt werden können, sondern sich ggf. über größere Cluster hinweg erstrecken. Dazu ist die Untersuchung des gesamten Netzwerks nötig. Die Auswertung derartiger Muster kann von bestehenden Software-Tools kaum übernommen werden, sodass ebenfalls spezifisches Expertenwissen nötig ist, um relevante Anomalien in einem Sachverhalt zu interpretieren, welches in VRNetzer mit Datenquellen und Berechnungsmethoden kombiniert wird [11].

3.3 UI- und UX-Design

Wenn es um die Entwicklung einer Softwareanwendung geht, findet der Entwurf auf zwei Ebenen statt – im Backend und im Frontend. Das Backend beschreibt die tatsächlichen Funktionen und Hintergrundprozesse, welche von einem Programm ausgeführt werden, die unsichtbar für den Nutzer sind. Das Frontend hingegen bezieht sich auf den Entwurf von Benutzeroberflächen mit verschiedenen visuellen Elementen und Graphiken, die vom Anwender wahrgenommen werden und über die er das Programm steuern kann [105]. In diesem Schritt der Entwicklung spielt das UI- und UX-Design eine tragende Rolle [106]. Der Begriff „UX“ wurde vom amerikanischen Professor Donald Norman geprägt und bezieht sich auf die Erfahrungen, die ein Nutzer mit digitalen oder analogen Systemen macht [107]. Diese Erfahrungen umfassen das visuelle Design, die Darbietung von Schnittstellen und physische Wechselwirkungen mit dem System. Ein weiterer Punkt ist, wie Instruktionen für den Umgang mit dem System dargestellt werden. UX ist ebenfalls in den ISO-Normen vertreten [107, 108]. Dort berücksichtigen sie die „[...] Emotionen, Überzeugungen, Vorlieben, Wahrnehmungen, physischen und psychologischen Reaktionen, Verhaltensweisen und Leistungen, die vor, während und nach der Nutzung eines Systems auftreten“ [107, Übers. d. Verf., S. 2].

Im Anwendungsfall kann die UX vielseitig beeinflusst werden. Dabei gibt es gewisse Komponenten, die mehr Einfluss auf das kognitive Erlebnis eines Nutzers gegenüber einem Computerprogramm haben. Wenn ein Nutzer eine beliebige Softwareanwendung startet, sei es auf dem PC, oder einer VR-Brille, wird er i. d. R. mit einem UI begrüßt. Eine Benutzeroberfläche ist folglich die erste Instanz, welche schon initial die Haltung des Nutzers gegenüber dem Programm formt [107, 109]. Deshalb sei bei dem Entwurf eines UIs Vorsicht und eine gründliche Konzeptionierung geboten, um den Nutzer nicht „abzuschrecken“. Denn eine verwirrende oder unzureichende Benutzeroberfläche für den Gebrauch einer Anwendung ist eine Verletzung der Erwartungskonformität [110] und kann den Nutzer dazu bringen, sie zu verlassen [111, 112, 113]. Der Vorfall würde in die Kategorie des schlechten UX-Designs fallen. Dieser Effekt kann auf verschiedene Weisen verursacht werden. Beispielsweise wird dem Benutzer nicht verdeutlicht, welchem Zweck eine Anwendung dient, wodurch er keinen Sinn erkennt, diese weiterzuverwenden [112, 113]. Usern sollte deshalb verdeutlicht werden, was die Aufgaben und Funktionen eines Programms sind, oder was die nächsten Schritte sein werden, um die Aufgabenangemessenheit nach ISO 9241-110 [110] zu erfüllen. Eine Möglichkeit wäre eine Art Willkommensnachricht, besonders, wenn es um die Verwendung eines Chatbots oder von kommunikativen Systemen geht. Transparenz ist in solchen Fällen besonders wichtig,

da Nutzer auf diese Weise offener sind [114, 115]. Ein kurzes Tutorial als Übersicht über die Funktionen des Programms können dem Nutzer einen angenehmen Einstieg in die Arbeit ermöglichen und tragen zur Selbstbeschreibungsfähigkeit bei. Eine Website oder Anwendung wird so erlernbar gestaltet. Andernfalls ist es wichtig, eine sorgfältige Dokumentation als Nachschlagewerk beispielsweise im WWW zu bieten [110].

Eine unübersichtliche Gestaltung, durch verschachtelte oder hierarchisch verzweigte Ansichten, kann ebenfalls dazu führen, dass der Nutzer sich in der Anwendung nicht zurechtfindet [112]. Websites mit vielen verschachtelten Seiten nutzen zur Übersicht und Steuerbarkeit die sogenannte Breadcrumb Navigation (dt. Brotkrümelnavigation). Sie kann dabei helfen, einen Überblick über die hierarchischen Strukturen zu verschaffen und erlaubt eine einfache Navigation zu vorherigen Seiten (siehe Abbildung 11) [110, 116, 117].

Universität zu Köln  > Albertus Magnus Center  > Promotion (R1)  > Promotion an der Universität zu Köln 

Abbildung 11: Ein Beispiel für die Breadcrumb Navigation auf der Website der Universität zu Köln (eigene Darstellung)

Ein weiteres Problem kann sein, dass Texte und Informationen nicht nutzerfreundlich dargestellt werden, sodass der Sinn des Programms gar nicht erst erkannt wird. Hierbei ist es wichtig, dass Texte in ausreichender Größe dargestellt werden und die Farbwahl des Hintergrunds und der Schrift angemessen ausfällt. Verschiedene Farbkombinationen führen dazu, dass die Texte schwer zu erfassen sind und das Lesen anstrengend oder gar unmöglich machen [113, 118]. Derartige Regeln sind in den Web Content Accessibility Guidelines (WCAG) festgelegt [119, 120]. Sie schreiben ein barrierefreies Design vor, was insbesondere bei Menschen mit Behinderungen und Kindern von Bedeutung ist. Tools wie der Adobe Color Contrast Analyzer⁵ können assistieren, um passende Farbkombinationen für einen möglichst hohen Kontrast herzustellen. Das Kontrastverhältnis kann hierbei zwischen 1:1 (kein Kontrast: Text und Hintergrund haben die gleiche Farbe) bis zu 21:1 (maximaler Kontrast: schwarzer/weißer Text auf weißem/schwarzem Hintergrund) liegen. Wie sich das Kontrastverhältnis berechnet, zeigt Formel (1) [121]:

$$\text{Kontrastverhältnis} = (L1 + 0.05)/(L2 + 0.05) \quad (1)$$

⁵<https://color.adobe.com/de/create/color-contrast-analyzer> (letzter Zugriff 10.08.2024)

Den Zähler bildet die relative Helligkeit der Bildpunkte von helleren Farben, auch Luminanz genannt (L1). Der Nenner ist die relative Luminanz der dunkleren Farben (L2). Die Formel basiert nach Angaben der WCAGs auf den ISO 9241-3 [122] und ANSI-HFES-100-1988 [123] Standards.

Die Anforderungen an Textgrößen und Farbkontraste sind von Konformitätsstufen abhängig. Diese sind dreistufig eingeteilt: A, AA und AAA [124]. Dabei ist Level A die niedrigste und AAA die höchste Stufe. Je nach Grad der Barrierefreiheit wird einer Webseite eines der Level zugeschrieben. Die Level sind hierarchisch geordnet und beinhalten jeweils die Konformität der darunterliegenden Stufe. Als minimales Kontrastverhältnis für graphische Oberflächen schreiben die WCAG 2.0 ein Verhältnis von 4.5:1 (Level AA) und 7:1 (Level AAA) für normalgroße Texte (Schriftgröße bis 17 Punkte und maximal 13 Punkte und fett) vor. Sind die Texte groß (Schriftgröße ab 18 Punkte und minimal 14 Punkte und fett), so ist ein Verhältnis von 3:1 (Level AA) und 4.5:1 (Level AAA) ausreichend. Ein Beispiel ist in Abbildung 12 zu sehen. Diese Richtlinien gelten allerdings nicht für Texte und Bilder, die zwar Teil eines UIs sind, aber nur zur Zierde existieren und keinen informativen Beitrag leisten (z. B. das Logo eines Unternehmens). Für derartige UI-Elemente existieren keine Vorgaben [121].

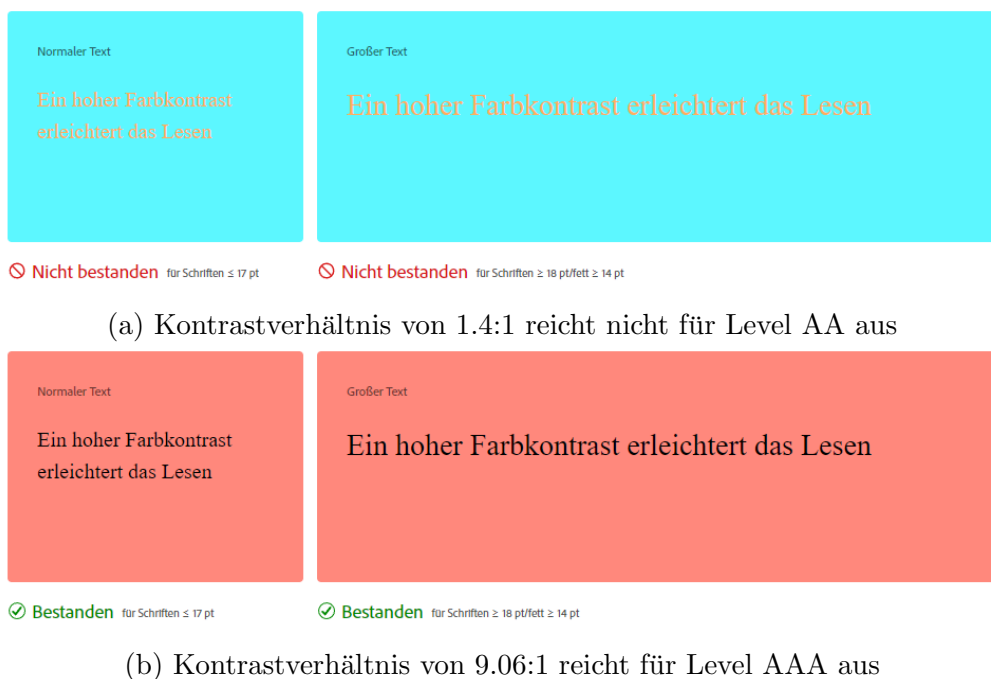


Abbildung 12: Der Satz „Ein hoher Farbkontrast erleichtert das Lesen“ wird hier mit zwei unterschiedlichen Kontrastverhältnissen über den Color Contrast Analyzer von Adobe dargestellt (eigene Darstellung)

Generell gilt daher, den Entwurf eines UIs an die Adressatengruppe anzupassen. Besonders für Kinder und physisch oder psychisch beeinträchtigte Personen, sollte ein UI simpel und einfach erlernbar gestaltet werden [125, 126]. Im Falle von visuell beeinträchtigten Nutzern ist besonders wichtig, auf die Farbschemata zu achten und ggf. einen Farbenblindenmodus einzubauen [127]. Diese genannten Beispiele verdeutlichen, dass Usability ausschlaggebend für eine gute UX ist [111, 112, 113, 128].

Usability kann durch verschiedene weitere Faktoren geprägt werden. Hat eine Website lange Ladezeiten, so leidet die Erfahrung des Nutzers erheblich. Ist das Importieren von Datensätzen beispielsweise Teil einer Software, ist eine Statusanzeige wichtig, um dem Nutzer direktes Feedback über Prozesse zu geben. Andernfalls könnte er meinen, die Anwendung wäre abgestürzt, wenn der Ladevorgang länger dauert [129, 130, 131].

Im Kontext von UIs bezieht sich Usability auch auf die eindeutige Darstellung von Komponenten, um den Nutzer ein unkompliziertes Erlebnis zu beschern. Aspekte, die hier eine Rolle spielen, wären z. B. die Verständlichkeit von Inhalten. Verständlichkeit bezieht sich darauf, dass beispielsweise die Navigation intuitiv erfolgen kann und Anweisungen zur Anwendung klar und deutlich formuliert werden. Die Verständlichkeit und Erwartungskonformität kann dadurch gefördert werden, wenn Interaktionselemente konsistent dargestellt werden. *Buttons* sollten z. B. nicht grundlos Form, Farbe und Beschriftung wechseln [113, 132, 133, 134].

In der Literatur zu UX-Design taucht der Begriff „Affordance“ (dt. Nutzbarkeitshinweis) häufig auf. Affordance beschreibt die Aktionen, die mit einem UI-Element ausgeführt werden können. Ein *Button* kann gedrückt werden, ein *Slider* entlang einer Achse verschoben und ein *Dropdown* ausgeklappt werden. „Signifier“ (dt. Kennzeichnung) stehen mit Affordances in Verbindung und sind ihre visuellen Elemente, welche den Nutzer auf Aktionen hindeuten [113, 135, 136]. Ein unterstrichener Text kann bedeuten, dass er klickbar ist, wie es von Hyperlinks bekannt ist. Ein Text, welcher mit einer Box umrandet ist und in einer 3D-Optik heraussteht, deutet darauf hin, dass auch er angeklickt werden kann [113, 137].

Ein weiteres Designprinzip ist das „Mapping“ (dt. Zuordnung). Dieses Prinzip bezieht sich auf die Anordnung von Steuerelementen und den Steuereinheiten. Konkret heißt es, dass beide so angeordnet werden, dass ihre Relation für den Nutzer leicht erkennbar ist. Ein Beispiel zeigt Abbildung 13. Hier werden die Regler der Herdplatten im selben Muster angeordnet wie die Herdplatten selbst. Der User erkennt auf diese Weise sofort, welche Herdplatte über welchen Regler angeschaltet wird [113, 138].

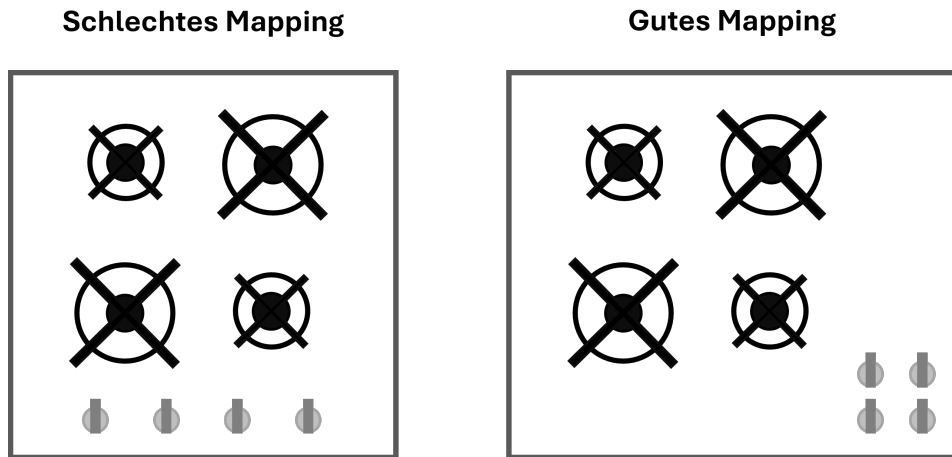


Abbildung 13: Durch die Anordnung der Regler ist die Bedienung des Herds intuitiv verständlich (eigene Darstellung in Anlehnung an [113, 138])

Bezüglich der Skalierung und Anordnung eines *Buttons* gilt: Je kleiner der *Button* ist, desto länger wird ein Nutzer benötigen, um ihn anzuklicken. Ebenso beeinflusst die Distanz zwischen dem Cursor und dem *Button* die Zeit, die benötigt wird, um ihn zu erreichen. Diesen Zusammenhang beschreibt Fitts' Law. Fitts formulierte diese Beziehung erstmals, als er das Verhältnis zwischen Dauer von Reaktionen und der Menge an zu verarbeitenden Informationen untersuchte [139]. Dabei stellte er eine direkte Proportionalität zwischen ihnen fest. Das bedeutet, dass unabhängig von der Größe der Bewegung und Präzision des Ziels, die Zeit (T) für eine Reaktion durch das Verhältnis von Distanz (D) zum Ziel und der Größe bzw. Breite (W) des Ziels beeinflusst wird (siehe Gleichungen (2) und (3)) [139, 140]:

$$T = a + b \cdot ID \quad (2)$$

$$ID = \log_2 \left(1 + \frac{D}{W} \right) \quad (3)$$

Daher ist die Reaktionsgeschwindigkeit bei variierenden Bedingungen größtenteils konstant, solange das Verhältnis von Distanz zu Zielgröße konstant bleibt. Beeinflussende Faktoren sind zum einen die Zeit, um ein Interaktionsgerät (z. B. Computermaus) zu bewegen oder zu stoppen (a), zum anderen Hardwarespezifikationen, wie die Geschwindigkeit (b) des Geräts. Fitts' Law kann seither als Vorhersage für die Interaktionsdauer mit *Buttons* benutzt werden. Über die Gleichung lässt sich die Interaktionszeit berechnen, bevor ein UI überhaupt implementiert wurde.

Aus Fitts' Law lassen sich manche Vorschläge zum UI-Design ableiten. Bei Desktop-Anwendungen ist empfohlen, Ecken und Kanten des Bildschirms zu verwenden, da die Breite solcher *Buttons* unendlich groß ist und so die Zeit gegen null geht [141]. Um die Distanz zwischen Cursor und Ziel zu minimieren, empfiehlt sich die Position des Cursors als Ausgangspunkt zu nehmen. Ein Beispiel wäre das Kontextmenü auf dem Windowsdesktop (siehe Abbildung 14).

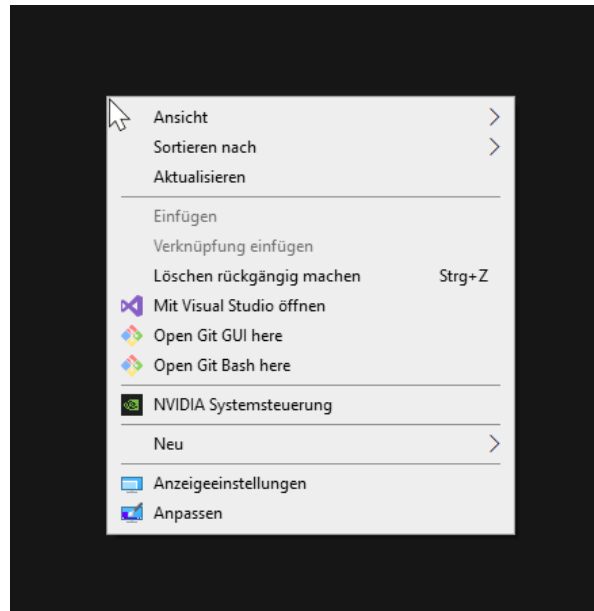


Abbildung 14: Das Kontextmenü erscheint bei einem Rechtsklick auf den Windows-Desktop immer beim Cursor (eigene Darstellung)

Da sich diese Arbeit mit der Entwicklung einer VR-Anwendung befasst, ist es eine angemessene Frage, ob Fitts' Law auch hier seine Gültigkeit hat. Zu diesem Zweck haben Clark et al. [142] eine Zeigeaufgabe in VR implementiert, um die Erweiterung von Fitts' Law auf die VR zu überprüfen [142]. Ihre Untersuchungen ergaben, dass die Zielgrößen, der radiale Abstand und der Neigungswinkel zum Ziel, die Zeit signifikant beeinflussten. Sie stellten außerdem fest, dass die Einflussgröße des Neigungswinkels mit der Änderung der Zielgröße variiert. Sie deuteten das Ergebnis so, dass die Tiefenwahrnehmung von der Zielgröße beeinflusst wird.

4 Konzeption

Bei der Konzeptionierung dieser Arbeit wurden die Bereiche eindeutig definiert, welche vor der Entwicklung der VR-Anwendung bearbeitet werden müssen und welche Darstellungsmethoden in die Anwendung integriert werden sollen. Diese Bereiche gliedern sich wie folgt:

1. Entwicklung eines Konverters in R, welcher einen Simulationsdatensatz in .csv-Dateien exportiert (vor der Implementierung)
2. Einlesen der Datensätze und Formatprüfungen (VR-Anwendung)
3. Darstellung der Organismenpopulation (VR-Anwendung)
4. Darstellung der Substanzkonzentrationen (VR-Anwendung)
5. Darstellung der metabolischen Flüsse (VR-Anwendung)
6. Steuerung der Datenvisualisierung (VR-Anwendung)
7. Testen von Limitation der Anwendung, die mit der Größe von Datensätzen zusammenhängen (nach der Implementierung)
8. Szenengestaltung (optional, VR-Anwendung)

4.1 Entwicklung eines Konverters in R

Die BacArena-Simulation⁶ soll zunächst klein sein, aber ausreichend viele Daten enthalten, da sie für die Entwicklung der VR-Anwendung als Demo-Datensatz dient. Für die Inhalte gelten die Einschränkungen, dass sie Informationen über die Population beinhalten sollen, wie z. B. die Spezies eines Organismus und dessen Position im Simulationsraum. Ebenfalls sollen Konzentrationswerte von Substanzen der Umgebung enthalten sein sowie deren Verwendung im metabolischen Austausch. Damit die Größe der Datensätze den erforderlichen Rahmen nicht überschreitet, soll die Simulation nicht mehr als zehn Zeitschritte, Spezies und Substanzen umfassen. Wie der Dokumentation zu entnehmen ist, enthält BacArena bereits den simulierten Testdatensatz „simplified human intestinal microbiota“ (dt. vereinfachte Mikrobiota des menschlichen Darms, SIHUMI). Es empfiehlt sich daher zu prüfen, ob SIHUMI den genannten Anforderungen entspricht.

⁶<https://rdrr.io/cran/BacArena/man/> (letzter Zugriff 25.09.2024)

Es ist zu vermuten, dass die geforderten Dateninhalte nicht mit einem einzigen Befehl produziert werden können. Deshalb müssen sie in einheitliche Darstellungsformate gebracht und zusammengeführt werden. Je nachdem, wie die Daten vorliegen, bietet es sich an, für jede Darstellungsvariante (Population, Substanzen und metabolische Flüsse) einen eigenen Datensatz zu erstellen, oder sie in einer gemeinsamen .csv-Datei zu kombinieren. In diesem Zug ist es ebenfalls ratsam, dass die Dateien einer Naming-Convention folgen, welche das Einlesen in die VR-Anwendung erleichtert.

4.2 Einlesen der Datensätze und Formatprüfungen

Nach dem Start der VR-Anwendung müssen die Datensätze eingelesen werden. Dazu eignet sich am besten ein UI, welches den Import eines Demo-Datensatzes starten kann. Hierbei sollen die Daten direkt aufgeteilt und in für Unity verwendbare Datentypen konvertiert werden. Beispielsweise wäre es sinnvoll, wenn aus den Populationsdaten direkt Instanzen der Organismen-Klasse erstellt werden und die Objekte alle zugehörigen Attribute bekommen. Dasselbe gilt für Substanzen und metabolische Flüsse.

Zusätzlich soll dem Nutzer die Möglichkeit gegeben werden, eigene Datensätze einzulesen. Da die Anwendung im Standalone-Modus verwendet werden soll, muss eine Verbindung zum lokalen Dateisystem der Brille hergestellt werden. Entweder soll die Auswahl über einen Dateibrowser erfolgen, oder es wird ein Ordner im Dateisystem definiert, in dem alle Datensätze hinterlegt werden. Dieser Ordner soll beim Start gänzlich importiert werden.

Bei diesem Prozess sollen die Datensätze Formatprüfungen unterzogen werden. Auf diese Weise wird gewährleistet, dass die benutzerdefinierten Dateien verwendbare Daten enthalten und an den Importalgorithmus weitergegeben werden können. Von Bedeutung wäre hier beispielsweise, dass die Spalten einheitliche Datentypen aufweisen und die Dimensionen der Simulationsfläche übereinstimmen.

4.3 Darstellung der Organismenpopulation

Welche Methode verwendet wird, um die Organismen darzustellen, ist von den Daten abhängig, die BacArena produziert. Grob können die Anforderungen aber schon wie folgt beschrieben werden: Jeder einzelne Organismus soll als eigene Einheit dargestellt werden. In Unity könnte dies über *GameObjects* erfolgen. Die *GameObjects* sollen in derselben räumlichen Beziehung angeordnet werden, wie in der BacArena-Simulation. Dazu müssen die exportierten Datensätze jeden Organismus mit seinen x- und y-Koordinaten enthalten.

Für die *GameObjects* selbst soll ein 3D-Modell eines Bakteriums in Blender (siehe Kapitel 2.2.6) erstellt werden. Das Modell soll die Form einer Kapsel haben, welche ggf. durch Pili und das Flagellum ergänzt werden. Abhängig von der Spezies eines Organismus sollen die darstellenden *GameObjects* auch differenzierbare *Materials* haben, um sie unterscheiden zu können.

Wenn die Datensätze mehr Informationen über die Organismen enthalten als die Raumkoordinaten und die Speziesbezeichnung, soll ein UI erstellt werden, welches diese Informationen abbildet. Aufgerufen kann das UI beispielsweise durch Anklicken eines Organismus in der *Scene*.

4.4 Darstellung der Substanzkonzentrationen

Wie im State-of-the-Art-Kapitel 3.2 aufgegriffen, eignen sich Heatmaps gut zur Visualisierung von Matrizen. Da VR-Anwendungen aber eine dritte Dimension einbeziehen, soll in MicroLabVR zusätzlich eine 3D-Heatmap angeboten werden, um die Substanzkonzentrationen darzustellen. In Anbetracht der vorgestellten Visualisierungsmethode soll immer nur eine Substanz angezeigt werden, da sich die Heatmaps ansonsten überlappen. Um verschiedene Intensitäten besser unterscheiden zu können, soll der Farbverlauf an die Höhe des Konzentrationswerts angepasst werden. Eine Skala soll zur Unterstützung bei der Interpretation der Daten das Spektrum von Minimal- bis Maximalwert zeigen.

4.5 Darstellung der metabolischen Flüsse

Metabolische Flüsse darzustellen, ist potentiell etwas komplizierter. An dieser Stelle spielt die Form der Daten von BacArena eine tragende Rolle. Es kann sein, dass die Produktion und Aufnahme von Metaboliten lediglich durch numerische Werte dargeboten werden. In diesem Fall reicht die Darstellung, ob ein Organismus einen Metabolit aufnimmt oder abgibt, ggf. gepaart mit der Geschwindigkeit, aus. Wenn allerdings der Austausch von Metaboliten zwischen zwei Spezies enthalten ist, wäre es angeraten, auch dieses Zusammenspiel darzustellen. Denkbar wäre hier die Visualisierung mittels Verbindungslinien oder der Markierung von Bereichen, in denen der Austausch stattfindet. Eine höhere Flussgeschwindigkeit könnte in diesen Fällen mit dickeren Linien oder kräftigerer Färbung der Cluster abgebildet werden.

Aus derartigen Austauschreaktionen lassen sich Flussgradienten ableiten. Eine gemäße Darstellungsmethode wäre die Angabe der Flussrichtung mit Pfeilen. Andernfalls könnte ein Farbverlauf entlang des Gradienten genutzt werden.

4.6 Steuerung der Datenvisualisierung und Benutzerfreundlichkeit

Anders als in BacArena, soll dem Nutzer ein UI geboten werden, über welches sich die Simulation steuern lässt. Um alle Stadien zu erkunden, soll der Benutzer durch alle Zeitpunkte iterieren können. Alle Komponenten der Simulation müssen, basierend auf dem gewählten Zeitpunkt, ihren Zustand ändern. Parallel soll jede der drei Hauptkomponenten (Population, Substanzen und metabolische Flüsse) unabhängig von den anderen beiden gesteuert werden können. So soll dem Anwender erlaubt werden, sich nur auf die wesentlichen Daten zu konzentrieren.

Da in dieser Arbeit auch ein Augenmerk auf das UI- und UX-Design gelegt wird, soll bei der Datenvisualisierung die Barrierefreiheit in einem angemessenen Grad berücksichtigt werden. Es sollen deshalb Farbkontraste geändert werden können und Texte ausreichend großgeschrieben werden. Damit der User die Anwendung eigenständig erforschen kann, soll er von einem kurzen Tutorial angeleitet werden.

4.7 Steuerung und Fortbewegung

Die Fortbewegung in MicroLabVR wird über den Joystick des Controllers möglich sein, da der verfügbare Platz im realen Raum häufig nicht ausreicht, um sich physisch zu bewegen. Auf diese Weise soll zusätzlich das Verletzungsrisiko minimiert werden. Um weitere Distanzen zurückzulegen, oder der Cybersickness entgegenzuwirken, wird auch die Teleportation im Raum erlaubt sein.

4.8 Limittesting

VR-Anwendungen sind in der Regel anfälliger für Lags (dt. Verzögerungen) als reguläre Computerprogramme, da sie auf schwächerer Hardware, im Vergleich zu einem Desktop-PC mit Graphikprozessor, laufen und graphisches Rendern in Echtzeit nötig ist. Diese Einschränkung kann die Komplexität und Größe der Datensätze stark herabsetzen und zeitgleich den Komfort für den User verschlechtern, indem es potentiell zu Cybersickness führt. Deshalb sollen nach der Implementierung die Limits der Anwendung erforscht werden. Dazu sollen mehrere Datensätze verschiedener Größen erstellt werden, um die Maximalgröße herauszufinden, welche auf der Meta Quest 3 flüssig mit ungefähr 70 FPS läuft. Getestet werden soll MicroLabVR sowohl in der Standalone-Version als auch in der PC-VR-Version, um zu analysieren, wie stark die Leistung beider Varianten differenziert.

Die Tests sollen vergleichbare Werte produzieren. Daher empfiehlt es sich, folgende Messwerte tabellarisch festzuhalten:

- Anzahl erzeugter Instanzen in Unity (beispielsweise GameObjects)
- Benötigte Zeit zum Einlesen der Daten (hängt sich die Anwendung dabei auf?)
- Benötigte Zeit zum Starten der Simulation (hängt sich die Anwendung dabei auf?)
- Anzahl der FPS

4.9 Szenengestaltung

Nicht erforderlich, aber praktisch, wäre eine angemessene Szenengestaltung. Die VR-Anwendung soll in einem Setting stattfinden, welches zu dem Thema dieser Arbeit passt. Ein Labor scheint geeignet, um die dargestellten Inhalte in einen passenden Kontext zu setzen. Der Fokus der *Scene* soll auf einer überdimensionalen Petrischale liegen, die in der Mitte des Raums auf einem Tisch steht. Diese Petrischale soll die Organismen tragen. Damit der Nutzer sich eine Übersicht verschaffen kann, soll es eine weitere Etage geben, die eine Vogelperspektive auf die Petrischale erlaubt.

4.10 Datenarchitektur der VR-Anwendung

Bei der Entwicklung einer Software ist es zu Beginn wichtig, die Programmstruktur zu konzipieren. Auf diese Weise können die Aufgabenbereiche definiert und die Entwicklung geregelt angegangen werden. Anderen Entwicklern kann es helfen, die Relationen zwischen Klassen und Objekten nachzuvollziehen, um spätere Modifikationen sorgfältig vorzunehmen.

Da Nutzer über UIs die VR-Anwendung steuern sollen, muss eine UI-Klasse erstellt werden, welche den Zugriff auf alle UI-Elemente, wie *Buttons*, *Slider* etc. hat. Sie soll auf alle weiteren UI-Klassen (*MainUI*, *SimulationUI*, *TutorialUI*) zugreifen können, welche die Funktionen hinter den Steuerelementen verwalten. Dabei soll das Tutorial von der Klasse *TutorialUI* kontrolliert werden. Die Klassen *MainUI* und *SimulationUI* steuern den Import der Datensätze und beeinflussen den Ablauf der Simulation. Deshalb müssen sie mit dem *FileManager* kommunizieren. Der *FileManager* stellt das Herzstück der Anwendung dar. Er soll alle Aufgaben übernehmen, die mit der Datenverwaltung zusammenhängen. Somit ist er für das Einlesen der Datensätze, dessen Überprüfung und die Konvertierung in verwendbare Datenformate zuständig. In enger Beziehung

zum *FileManager* steht der *EntityManager*. Dieser erzeugt aus den Simulationsdaten Entitäten (Organismen und Substanzen), die als Darstellungsobjekte der Daten fungieren. Hierzu soll er Objekte der Klassen *Microorganism* und *Substance* instanziiieren, welche alle relevanten Attribute enthalten. Das Pseudo-UML-Diagramm für MicroLabVR sieht wie folgt aus:

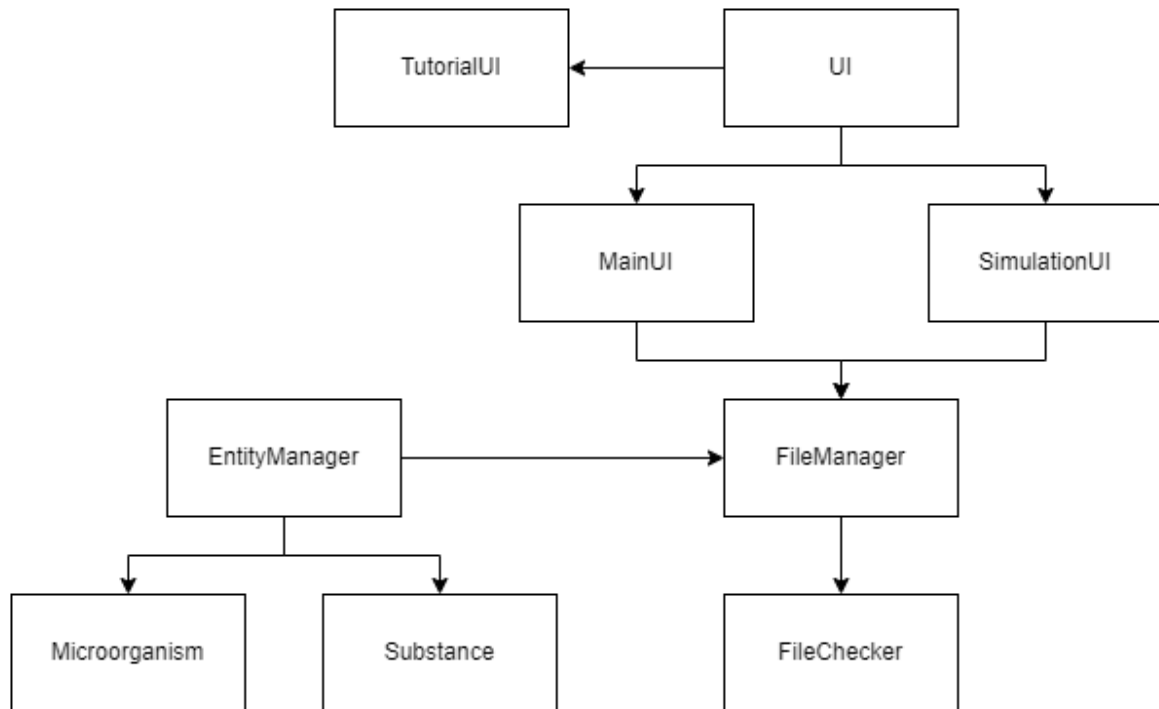


Abbildung 15: Die Datenarchitektur von MicroLabVR. Ein gerichteter Pfeil zeigt an, dass die Klasse, von der er ausgeht, auf die Zielklasse zugreifen kann (eigene Darstellung)

5 Implementierung

Die Implementierung lässt sich in zwei Abschnitte einteilen. Zunächst wurde ein Konverter in R entwickelt und im Anschluss die VR-Anwendung MicroLabVR. Währenddessen wurden UI- und UX-Aspekte in einem angemessenen Rahmen berücksichtigt.

5.1 R-Konverter zur Erstellung passender Datensätze

Der Konverter soll dem Nutzer dienen, Datensätze zu produzieren, welche direkt in MicroLabVR eingelesen werden können. Er nutzt zur Veranschaulichung die Demo-Simulation SIHUMI ⁷. Hierbei handelt es sich um einen Testdatensatz von acht Bakterien (*Anaerostipes caccae*, *Bacteroides thetaiotaomicron*, *Bifidobacterium longum*, *Blautia producta*, *Clostridium butyricum*, *Clostridium ramosum*, *Escherichia coli*, *Lactobacillus plantarum*), welche über einen Zeitraum von acht Stunden auf einer westlichen Diät gewachsen sind. Die Anzahl an Substanzen wurde auf die sechs variabelsten beschränkt, um die Dimensionen der Datensätze kleinzuhalten.

Der Nutzer hat in dieser Proof of Concept-Anwendung mit Kommentaren gekennzeichnete Stellen im Code, die er durch eigene Angaben ausfüllen soll. Diese beschränken sich auf den Speicherort der Datensätze und das Simulationsmodell, dessen Daten exportiert werden.

```
##### Declare your file directory here #####
folder_path <- "C:/Users/username/UniSiegen/MDS/Masterarbeit/Datasets"

##### Load your model here #####
data("sihumi_test")      # Loads the already simulated data.
```

Abbildung 16: Der Header des Konverters: Hier können der Dateipfad gesetzt und das Modell geladen werden (eigene Darstellung)

Der Quellcode nutzt verschiedene Befehle von BacArena, um Zugriff auf Simulationsdaten zu erlangen. Vier Funktionen spielen eine tragende Rolle. Die konzipierten Informationen über Organismen, Substanzen und metabolische Flüsse können allein über die Methoden *evalArena* (oder *simlist* kombiniert mit *extractMed*) und *exchangeslist* produziert werden. Die Methode *evalArena* plottet alle raumzeitlichen Simulationsschritte der Population und Substanzen. Diese Plots waren bereits in Kapitel 3.1.2 zu sehen (siehe Abbildungen 8 und 9). Über einen Boolean-Parameter liefert sie alle Daten, welche die Plots erzeugen.

⁷https://rdrr.io/cran/BacArena/man/sihumi_test.html (letzter Zugriff 16.10.24)

Bezüglich der Populationsdaten bedeutet das: Zeitpunkt, Biomasse, Genotyp, Phänotyp, x- und y-Koordinaten für jeden Organismus. Zu jeder Substanz wird eine Konzentrationsmatrix pro Zeitpunkt zurückgegeben.

Austauschgeschwindigkeiten von Metaboliten zu jedem Zeitpunkt lassen sich mit *exchangelist* zurückgeben. Allerdings war es nicht möglich, die Werte mit den Organismen zusammenzuführen, da das Mapping der Flussdaten von BacArena fehlerhaft ist. Aus diesem Grund wurden im späteren Verlauf zufällige Fluxwerte zwischen -50 und 50 gewählt, um einen verwendbaren Datensatz zu erzeugen.

Der finalisierte Konverter erzeugt zwei Datensätze: den Populationsdatensatz „population_dataset.csv“ und den Substanzdatensatz „substance_dataset.csv“. Er berücksichtigt bereits vorhandene Dateien mit demselben Namen und nummeriert die neu generierten Datensätze fortlaufend, um die Bestehenden nicht zu überschreiben.

5.1.1 Aufbau des Populationsdatensatzes

Die erste Spalte enthält das Label des Datensatzes: „Population“. Dies gilt als Sicherheitsmechanismus, sodass bei der späteren Weiterverarbeitung geprüft werden kann, ob es sich um den richtigen Datensatz handelt. Die nächsten Spalten enthalten den Zeitpunkt, zu welchem der Eintrag gehört und die Koordinaten, an welcher x- und y-Stelle sich der Organismus befindet. Danach folgt die Biomasse des Mikroorganismus sowie dessen numerischer Genotyp und Phänotyp. In der achten Spalte steht der vollständige Name des Organismus, der dem Genotypen zugeordnet wird. Der Name setzt sich aus mehreren Fragmenten zusammen, die von Unterstrichen getrennt werden: *Gattung_Spezies_Stammbezeichnung*. Die finalen sechs Spalten bilden die Flussangaben zu den sechs zugelassenen Metaboliten.

Ein Eintrag des Datensatzes kann wie folgt aussehen:

Population, 1, 19, 13, 0.487656, 1, 6, "Anaerostipes_caccae_dsm_14662", 20.300157, -40.844730, 4.426094, 3.289703, -2.599154, -30.084884

In diesem Eintrag handelt es sich um das Bakterium *Anaerostipes caccae*, mit der Stammbezeichnung DSM 14662 aus der Datenbank „Deutsche Sammlung von Mikroorganismen und Zellkulturen“⁸. Es befindet sich an Position (19, 13) im Simulationsbereich, hat eine Biomasse von 0.487656 Gramm Trockengewicht (g DW), gehört zum Genotyp mit der Identifikationsnummer (ID) 1 und zum Phänotyp mit der ID 6.

⁸<https://www.dsmz.de/> (letzter Zugriff 16.10.2024)

5.1.2 Aufbau des Substanzdatensatzes

Anders als beim Populationsdatensatz setzt sich ein verwendbarer Eintrag des Substanzdatensatzes aus mehreren Zeilen zusammen. Die Konzentrationsmatrizen werden zeilenweise aufgespalten. Das bedeutet konkret, dass zunächst das Label „Substance“ voransteht. Den nächsten Eintrag machen der Substanzname und der Zeitstempel. Danach folgt die Zeilenzahl der Konzentrationsmatrix, die zu der benannten Substanz und dem erwähnten Zeitpunkt gehört. Danach folgen, abhängig von der x-Dimension des Simulationsbereichs, alle Konzentrationswerte einer Matrixzeile. Eine 4×5 Matrix sieht im Datensatz wie folgt aus:

Eine Zeile

Substance, "EX_nh4(e)", 2, 3, 0.7338, 0.6507, 0.3002, 0.7220, 0.3619, 0.4340

Vier Zeilen

Substance, "EX_nh4(e)", 2, 1, 0.1926, 0.8305, 0.6199, 0.2139, 0.7053, 0.9031

Substance, "EX_nh4(e)", 2, 2, 0.6287, 0.0318, 0.9742, 0.3920, 0.1163, 0.7891

Substance, "EX_nh4(e)", 2, 3, 0.7338, 0.6507, 0.3002, 0.7220, 0.3619, 0.4340

Substance, "EX_nh4(e)", 2, 4, 0.4132, 0.2514, 0.8605, 0.0945, 0.5730, 0.2114

Die Einträge sind wie folgt zu lesen: Die Konzentration von Ammonium ist zum Zeitpunkt 2 an dem Punkt (1, 1) im Simulationsbereich 0.4132 Millimolar (mM) und an dem Punkt (4, 3) 0.3920 mM.

5.2 MicroLabVR

Als Grundlage für das Projekt wurde das VR-Template von Unity gewählt. Dieses enthält bereits einige Konfigurationen für VR-Anwendungen, wie das XR Plug-in Management (siehe Kapitel 2.2.7) und das XR Interaction Toolkit (siehe Kapitel 2.2.7). Nützlich sind ebenfalls UI-Templates, welche beliebig modifiziert werden können.

5.2.1 Datenvisualisierung

Angesichts der produzierten Datensätze wurde sich dazu entschieden, die Organismen durch einfache *GameObjects* und die Substanzen als detaillierte *Meshes*⁹ in der Petrischale darzustellen. Zunächst wurden einfache Würfel als Platzhalter für die Organismen herangezogen. Später wurde jedoch gemäß der Konzeption (siehe Kapitel 4.3) ein 3D-Modell

⁹<https://docs.unity3d.com/ScriptReference/Mesh.html> (letzter Zugriff 16.10.24)

eines Bakteriums in Blender modelliert (siehe Abbildung 17). Das Modell repräsentiert nur die Kapsel eines Bakteriums. Auf die Pili und die Flagellen wurde verzichtet, um die Performance nicht weiter zu beeinträchtigen. In Unity erhalten die *Prefabs*¹⁰, je nach Spezies, ein andersfarbiges *Material*.

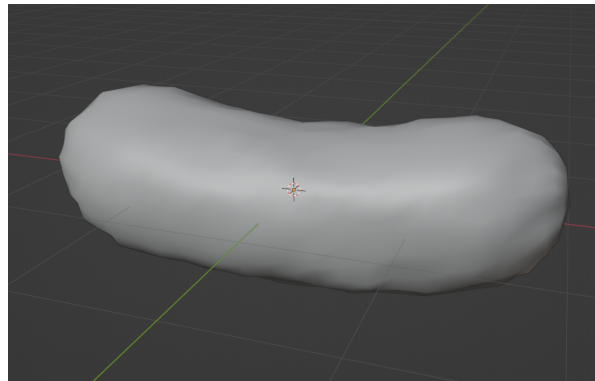
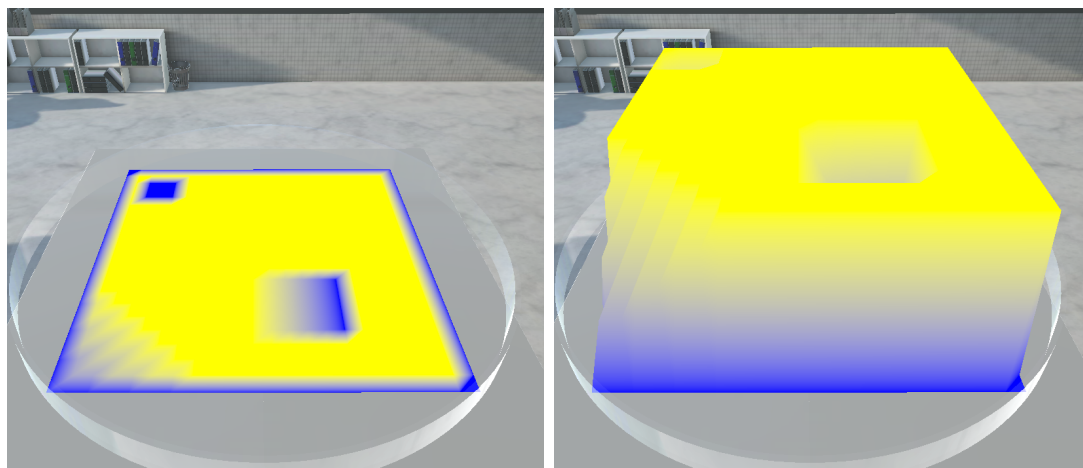


Abbildung 17: Das modellierte 3D-Modell eines Organismus in Blender (eigene Darstellung)

Die Substanzkonzentrationen werden durch *Meshes* repräsentiert, die zur Laufzeit erzeugt werden. Ein *Shader* färbt die Stelle abhängig von der Konzentration ein, sodass eine 2D-Heatmap entsteht. Für die 3D-Darstellung werden die Knoten räumlich angehoben, um so Hügel und Täler zu erzeugen.



(a) 2D

(b) 3D

Abbildung 18: Das Mesh zur Darstellung der Substanzkonzentration wird zur Laufzeit geladen. Blau: niedrige Konzentration, gelb: hohe Konzentration (eigene Darstellung)

¹⁰<https://docs.unity3d.com/Manual/Prefabs.html> (letzter Zugriff 16.10.24)

Ob ein Organismus einen Metaboliten produziert oder konsumiert, wird mithilfe von „Outline“-*Materials* visualisiert. Die 3D-Modelle werden entweder grün (Produktion) oder rot (Aufnahme) umrandet (siehe Abbildung 19).

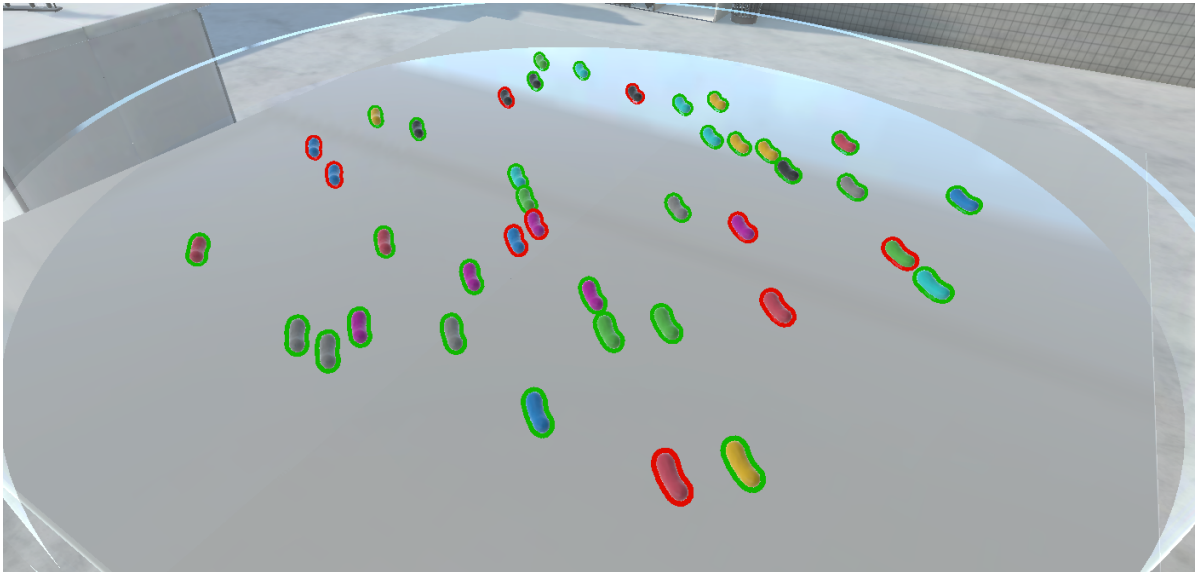


Abbildung 19: Organismen mit einer grünen Umrandung produzieren die Substanz, rot umrandete konsumieren sie (eigene Darstellung)

5.2.2 Szenengestaltung

MicroLabVR besteht aus einer einzigen *Scene*, deren Inneneinrichtung an ein Labor angelehnt ist. In der Mitte des Raums stehen zwei Tische, welche eine überdimensionale Petrischale tragen. Sie wurde eigenständig in Blender modelliert und mit einem transparenten *Material* versehen. Daneben sind zwei Arbeitsplatten mit Erlenmeyerkolben, Reagenzgläsern und weiteren Gefäßen. Auf der rechten Seite des Raums stehen Tische und Stühle mit Arbeitsmaterial sowie Stauregale mit Ordnern. Auf der linken Seite befindet sich ein Schreibtisch mit PC und Bürostuhl. Diese benannten 3D-Modelle wurden dem Unreal Engine¹¹ Projekt „Modular Hospital v.2“¹² entnommen und dienen als Dekoration der Szene. Sie erlauben somit keine weitere Interaktion.

Der User startet auf einem Balkon im Innenraum. Der Startpunkt ermöglicht die Übersicht über den gesamten Raum. Dadurch befindet sich der Nutzer in einer Vogelperspektive zu der Petrischale und kann das gesamte Mikrobiom überblicken (siehe Abbildung 20).

¹¹<https://www.unrealengine.com/> (letzter Zugriff 04.10.2024)

¹²<https://www.unrealengine.com/marketplace/en-US/product/modular-hospital/> (letzter Zugriff 04.10.2024)

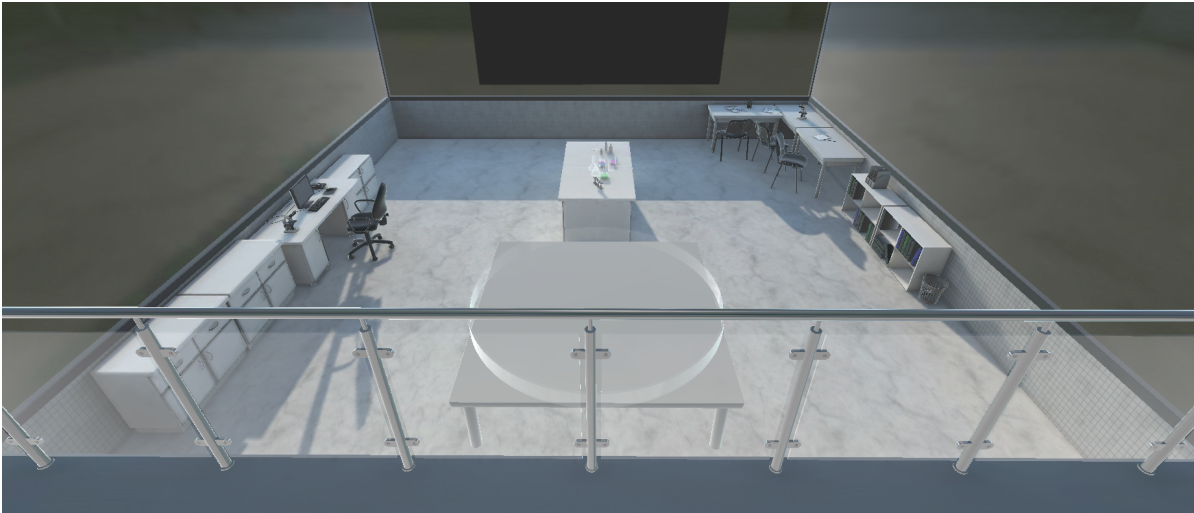


Abbildung 20: Die *Scene* aus der Perspektive, in der der Nutzer startet (eigene Darstellung)

In der Szene kann sich über den Joystick des linken Controllers bewegt werden. Sollte der Nutzer anfällig für Cybersickness sein, kann er sich auch im Raum teleportieren, oder physisch bewegen, sollte er genug Platz haben.

5.2.3 User-Tutorial

Basierend auf einem UI-Template der verfügbaren *Sample-Scene* wurde ein Tutorial mit sieben Slides (dt. Folien) erstellt (siehe Abbildung 21). Dieses befindet sich vor dem Nutzer, links neben dem Hauptmenü im Raum. Durch alle Slides kann über einen *Button* an der Unterseite des UIs iteriert werden. Betätigt der Nutzer den *Button* zum Schluss erneut, fängt das Tutorial von vorne an.

Zunächst zeigt das Menü ein Bild der Anwendung mit einem kurzen Begrüßungstext. Es folgt die Anleitung, wie ein Demodatensatz importiert und die Simulation gestartet werden kann. Auf Slide 3 wird gezeigt, wie mit dem *Slider* des Simulationsmenüs zwischen verschiedenen Zeitpunkten gewechselt werden kann. Danach erscheint die Übersicht der *Toggles*, welche die Anzeige der Substanzkonzentrationen steuern. Slide 5 informiert den Nutzer, dass er zwischen einer 2D-Heatmap und einer 3D-Ansicht der Konzentrationen wechseln kann. Nachfolgend steht ein Hinweis über die verschiedene Farbschemata der Konzentrationsdarstellung. Die letzte Slide erklärt die Funktion der Flux-*Toggles*.

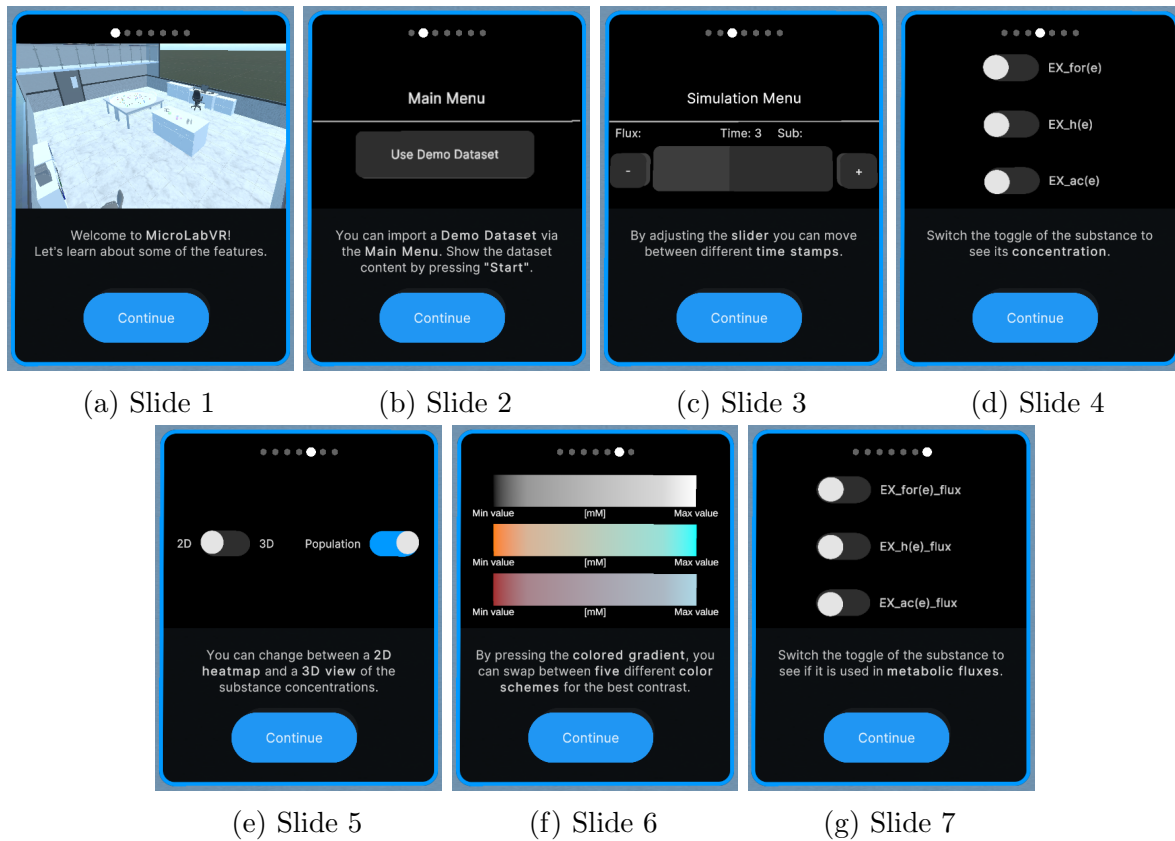


Abbildung 21: Alle Slides des User-Tutorials (eigene Darstellung)

5.2.4 Datenmanagement über das Hauptmenü

Sobald der User MicroLabVR startet, erscheint vor ihm ein Hauptmenü (siehe Abbildung 22). Der Nutzer muss nun über den *Button* mit der Aufschrift „Use Demo Dataset“ die hinterlegten Demodatensätze aus dem Ressourcen-Ordner der Anwendung laden. Dabei werden der Populations- und Substanzdatensatz jeweils in eine zweidimensionale Liste konvertiert. Die Datensätze können in diesem Format für weitere Backend-Prozesse praktisch verwendet werden. Das Einlesen der Datensätze erfolgt mittels sogenannter *Coroutinen*¹³. *Coroutinen* sind Methoden in Unity, die Prozesse über mehrere Frames (dt. Bildwechsel) strecken. Diese Vorgehensweise erlaubt, dass andere Funktionen während des Durchlaufs der *Coroutine* aufgerufen werden können. Dadurch konnte eine Fortschrittsanzeige implementiert werden, welche in jedem Frame aktualisiert wird (siehe Abbildung 22).

¹³<https://docs.unity3d.com/Manual/Coroutines.html> (letzter Zugriff 04.10.2024)

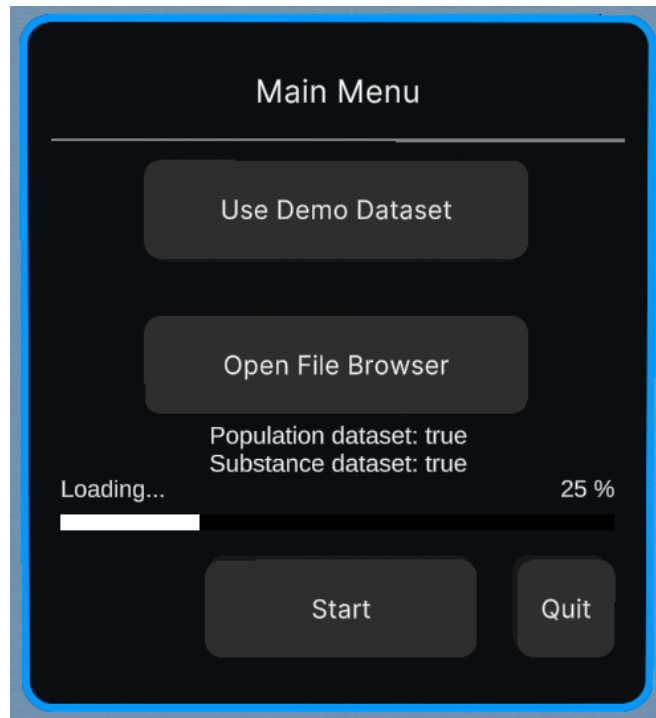


Abbildung 22: Das Hauptmenü von MicroLabVR (eigene Darstellung)

Wenn die Datensätze erfolgreich konvertiert wurden, findet ein Datentyp-Check jedes Zeileneintrags statt, welcher den Regeln der Unterkapitel 5.1.1 und 5.1.2 folgt. Beispielsweise müssen die Datensätze in der ersten Spalte korrekt gelabelt sein und die Spalten zwei bis vier des Populationsdatensatzes dürfen nur Integer enthalten. Wichtig ist außerdem, dass die Dimensionen der Konzentrationsmatrix im Substanzdatensatz mit der Größe der Simulationsfläche übereinstimmen und die Koordinaten der Organismen in dem definierten Bereich liegen. Bestehen Regelverstöße, werden je nach Kontext dem Nutzer Fehlermeldungen angezeigt:

- **Wenn ein Eintrag in den Datensätzen nicht das passende Format hat:** „Format of *Dateipfad* is invalid. Please check line *Zeile*, column *Spalte*. Invalid entry: *Fehleintrag*. Should be of type: *korrektes Format!*“
- **Wenn der Populationsdatensatz mehr als 14 Spalten hat (da auf sechs Flüsse derzeit beschränkt):** „Population dataset has *Anzahl Spalten* instead of 14 columns!“
- **Wenn die Dauer der Simulation in beiden Datensätzen nicht übereinstimmt:** „The simulation times *Zeiten* of your datasets don't match!“

- **Wenn die Dimensionen der Simulationsbereiche in beiden Datensätzen nicht übereinstimmen:** „The simulation dimensions of x *x-Dimensionen* or y *y-Dimensionen* don't match!“
- **Wenn der numerische Genotyp nicht eindeutig zugeordnet werden kann:** „Genotype does not match a name in line *Zeile* of population dataset!“

Diese Sicherheitsmaßnahme gewährleistet, dass keine Laufzeitfehler beim Zugriff auf die Daten auftreten, wenn benutzerdefinierte Datensätze importiert werden.

Im Anschluss an die Kontrolle werden mithilfe dieser Daten Instanzen der Klassen *Microorganism* und *Substance* erzeugt. Dabei erhalten die Membervariablen der beiden Klassen die gelesenen Werte.

Abhängig davon, ob die Formate mit den Vorgaben übereinstimmen und die Instanzen erzeugt wurden, zeigt eine Checkliste an, ob der Import erfolgreich war oder nicht. Auf dem UI werden die Namen der Datensätze gepaart mit einem Boolean gelistet, welcher den Importstatus zeigt (true: erfolgreich importiert, false: Fehler aufgetreten).

Wenn die gewünschten Datensätze erfolgreich importiert wurden, kann die Simulation über den „Start“-*Button* geladen werden. Genau wie der Datensatzimport erfolgt das Setup (dt. Vorbereitung) der Szene über eine *Coroutine*, sodass der prozentuale Ladevorgang dargestellt werden kann.

Während des Setups werden unter anderem wichtige Schlüsselwerte aus den Datensätzen gelesen. Dazu gehören die Simulationsdauer, die Dimensionen des Simulationsbereichs, die verbreiteten Substanzen und die metabolischen Flüsse. Anhand dieser Informationen wird das Simulationsmenü gestaltet.

5.2.5 Simulationsmenü

Das Simulationsmenü (siehe Abbildung 23) setzt sich zum Teil aus einem *Slider* zusammen, welcher die Iteration zwischen allen Zeitpunkten ermöglicht. Seine Konfiguration erfolgt mittels der extrahierten Simulationsdauer. Neben dem *Slider* befinden sich zwei *Buttons*, auf denen ein „+“ und ein „-“ zu sehen sind. Der User kann zwischen den Zeitschritten nun entweder durch Bewegen des *Sliders* mit dem Controller oder durch Klicken auf die Plus- und Minus-*Buttons*, wechseln. Für Situationen, in denen ein Blick aufs Menü hinderlich ist, können andernfalls der X- und der Y-Knopf des linken Controllers gedrückt werden. Wenn der *Slider* bewegt wird, aktualisiert sich das Mikrobiom in der Petrischale. Alle Organismen werden an einer weit entfernten Stelle in der Szene gepoolt (dt. gesammelt). Daraufhin werden diejenigen Organismen wieder in der Petrischale platziert, die benötigt

werden. Wurden bisher zu wenige *GameObjects* erzeugt, werden die übrigen instanziiert und ebenfalls in die Petrischale gesetzt. Auf diese Weise wird verhindert, dass sofort zu jedem Organismus ein 3D-Modell erzeugt wird, um MicroLabVR performanter zu machen.

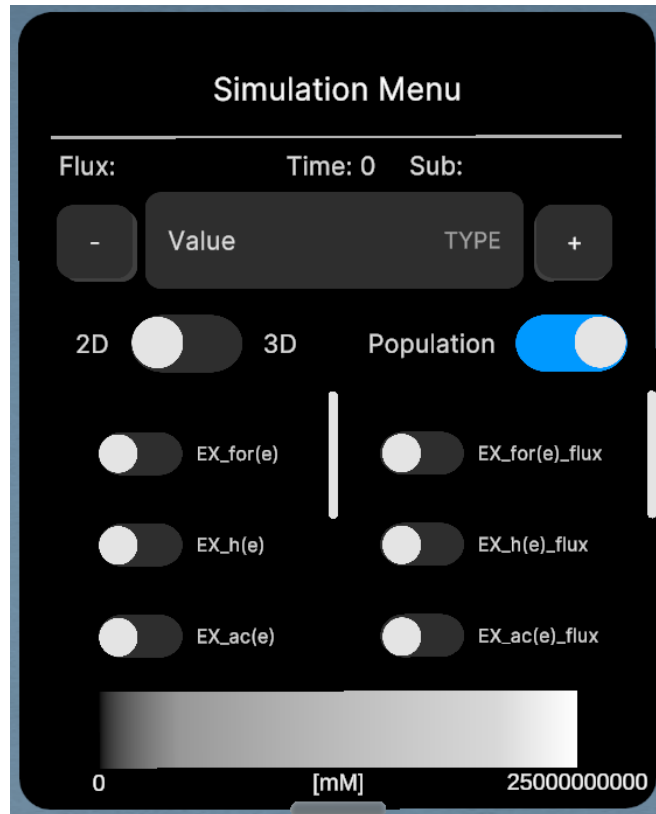


Abbildung 23: Das Simulationsmenü von MicroLabVR (eigene Darstellung)

Unter dem Slider sind ausgeschaltete *Toggles* zu allen verfügbaren Substanzen aufgelistet. Der Nutzer kann einen *Toggle* auswählen und sich die Konzentrationsverteilung anzeigen lassen. Dabei wird, ähnlich zu den Organismen, das *Mesh* der Substanz in der Petrischale positioniert; alle anderen verweilen unsichtbar außerhalb des Labors. Schaltet der User einen anderen *Toggle* an, stellt sich der vorherige wieder aus und die neue Substanzverteilung wird gezeigt. Oberhalb der Liste befindet sich ein weiterer *Toggle*, der standardmäßig auf „2D“ steht. Mit ihm kann die Dimension gewechselt werden, in welcher die Substanzkonzentration dargestellt werden soll. Um diesen Effekt zu erreichen, wurden nach dem Start der Simulationen zu jeder Substanz zwei *Mesches* erzeugt – in 2D und in 3D. Je nach gewählter Dimension wird das passende *Mesh* im Labor angezeigt. Analog funktionieren die Flux-*Toggles*. Anstatt jedoch *GameObjects* in der Szene zu platzieren, geben sie den Organismen das passende *Material*, um die Produktion oder

Aufnahme von Metaboliten zu kennzeichnen. Hierbei werden die alten *Materials* zwischengespeichert, um nach dem Ausschalten aller *Toggles* wieder zum ursprünglichen Aussehen zurückzukehren.

Am unteren Rand des Simulationsmenüs zeigt eine Farbskala mit Verlauf die Zuordnung von Konzentrationswerten an, wobei links der minimale und rechts der maximale Wert steht. Die Extremwerte werden zur Laufzeit aus dem Substanzdatensatz extrahiert und im UI gesetzt. Die Skala selbst ist ein *Button*, welcher, wenn betätigt, den Farbverlauf zwischen fünf Schemata wechselt.

5.2.6 Interaktionsmöglichkeit mit den Organismen

Der Nutzer kann mit jedem Organismus in der *Scene* interagieren. Hovort (dt. schwebt) er mit dem Controller über eines der 3D-Modelle, wird es schwarz umrandet. Drückt der User nun den Trigger, beginnt eine Schwebanimation (siehe Abbildung 24), um den Organismus hervorzuheben. Alle Informationen über den Organismus werden auf einer Informationstafel mit einer Mikroskopaufnahme angezeigt. Die Informationen setzen sich aus den Attributen des *Microorganism*-Objekts zusammen. Ein Skript, welches auf jedem *GameObject* liegt, liefert den Zugriff auf dessen Attribute und leitet sie weiter, um einen Informationstext zu gestalten. Klickt der Nutzer an eine andere Stelle im Raum, hört die Animation auf und die Informationstafel zeigt wieder den Standardtext: „Select an organism for information“ an.

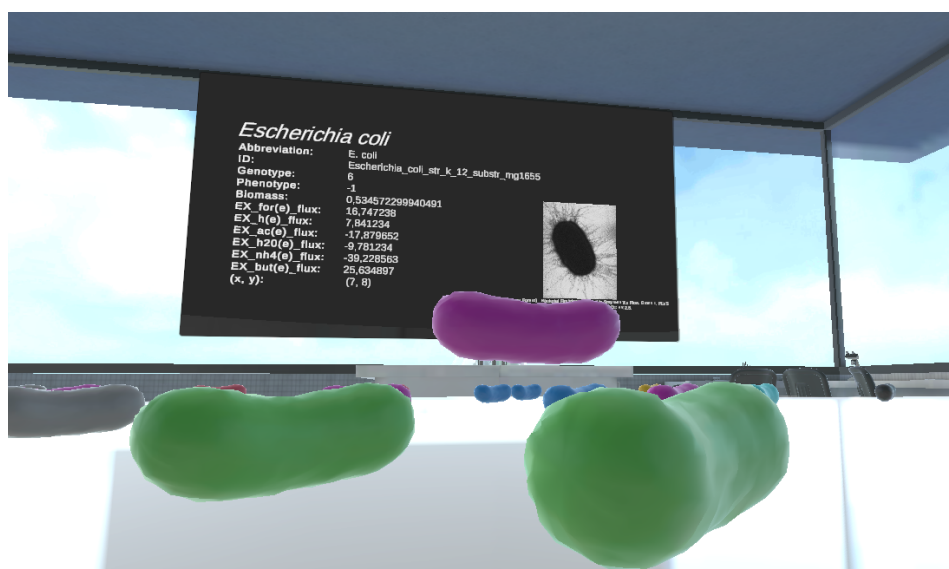


Abbildung 24: Der ausgewählte, violette Organismus ist ein *E. Coli* Bakterium (eigene Darstellung)

5.3 Aspekte des UI- und UX-Designs

In dieser Arbeit wurde die Notwendigkeit von gutem UI- und UX-Design ausgiebig erläutert (siehe Kapitel 3.3). Daher war es von Bedeutung, dass auch MicroLabVR viele Aspekte berücksichtigt, um eine gute Nutzererfahrung zu gewährleisten. Zunächst war es wichtig, dass der Nutzer eine verständliche Einführung bekommt, wie MicroLabVR funktioniert, damit die Anwendung seine Erwartungskonformität erfüllt. Daher wurde das User-Tutorial entwickelt, welches den Anwender Schritt für Schritt durch die Funktionen der UIs führt. Zugleich trägt das Tutorial zur Selbstbeschreibungsfähigkeit bei.

Bei dem Design der UIs wurde besonders darauf geachtet, dass die Farbkontraste der Texte intensiv sind. Wie bereits angemerkt, wurden die UIs des VR-Template von Unity verwendet. Um den Kontrast zum Text zu erhöhen, wurde die Transparenz des Hintergrunds entfernt. Letztendlich liegen die Kontrastverhältnisse der Menüs bei 6,1:1 (Weiß #FFFFFF auf Grau #626262) und 21:1 (Weiß #FFFFFF auf Schwarz #000000). Lediglich der *Button* des User-Tutorials hatte zunächst ein Verhältnis von 3,13:1 (Weiß #FFFFFF auf Blau #2096F3). Deshalb wurde die Farbe des *Buttons* auf ein dunkleres Blau (#2075B9) geändert, sodass ein Verhältnis von 4,88:1 erzielt wurde. Somit kommen alle UIs in MicroLabVR auf die Konformitätsstufe AA, da alle Kontrastverhältnisse über 4,5:1, aber unter 7:1 liegen. Wenn manche Nutzer dennoch Probleme haben, die Texte zu lesen, oder die Schaltflächen mit dem Controller zu bedienen, können sie alle Menüs größer skalieren. Diese Funktion war bereits in den UI-Template vorhanden.

Stetiges Feedback ist für den User notwendig, wenn Hintergrundprozesse laufen, die er angesteuert hat. Im Kontext der VR-Anwendung konzentriert sich diese Empfehlung auf die Statusanzeige des Hauptmenüs. Diese projiziert den Importstatus der Datensätze und danach den Setup-Status der *Scene* und informiert den Nutzer, sobald die Simulation gestartet werden kann. Diese Methode kann verhindern, dass ein User ungeduldig wird, oder gar die Anwendung beendet.

Visuelles und vibrotaktiler Feedback kann die Selbstbeschreibungsfähigkeit einer Anwendung fördern. In MicroLabVR wird visuelles Feedback genutzt, um zu verdeutlichen, welcher Organismus mit dem Controller ausgewählt wurde. Zunächst wird er beim Hovern schwarz umrandet und nach dem Anklicken mit einer Animation hervorgehoben. Bei beiden Aktionen gibt der Controller einen kurzzeitigen Vibrationsimpuls ab. Anhand dieser Methoden können Unsicherheiten vermieden werden, da ansonsten nicht zurückverfolgt werden kann, welcher Organismus ausgewählt wurde. Die Anwendung macht dem Nutzer zudem klar, in welchem Status sie sich befindet. Diese Form der Selbstbeschreibungsfähigkeit verdeutlicht, dass die Objekte interaktiv sind.

Da MicroLabVR als Werkzeug genutzt werden soll, um bioinformatische Daten zu analysieren, sollte Ablenkung vermieden werden. Daher wurde sich bewusst dazu entschieden, dass mit der Dekoration in der *Scene* nicht weiter interagiert werden kann. Ebenfalls soll das Blickfeld des anwendenden Wissenschaftlers nicht vom Simulationsmenü behindert werden. Aus diesem Grund wurde das Feature der UIs beibehalten, dass sie mit dem Controller gegriffen und im Raum verschoben werden können. Außerdem sollte es möglich sein, das Menü gänzlich zu verstecken. Der B-Knopf des rechten Controllers schaltet daher die Sichtbarkeit des Simulationsmenüs um. So muss während des Analyseprozesses der Blick nicht von der Petrischale abgewendet werden. Mit derselben Intention wurden die X- und Y-Knöpfe des linken Controllers mit der Funktion verknüpft, durch die Zeitschritte zu iterieren. Ein Blick aufs Simulationsmenü ist deshalb ebenfalls nicht notwendig. Wenn es um UI- und UX-Design geht, spielt Barrierefreiheit eine tragende Rolle. Ein Aspekt wurden bereits durch die Wahl der Farbkontraste berücksichtigt. Als es um die Wahl der Farbschemata für die Konzentrationsintensitäten ging, war es schwierig, die passende Farbkombination zu finden, da viele Sehschwächen berücksichtigt werden müssen. Final wurde sich dazu entschieden, fünf Schemata anzubieten, sodass der Nutzer den für sich besten Kontrast wählen kann.



(a) Schwarz (#000000) zu Weiß (#FFFFFF)



(b) Blau (#0000FF) zu Gelb (#FFFF00)



(c) Orange (#FF7F00) zu Cyan (#00FFFF)



(d) Violett (#7E1E9C) zu Gelb (#FFFF00)



(e) Braun (#A52A2A) zu Hellblau (#ADD8E6)

Abbildung 25: Alle fünf Farbschemata der Substanzverteilung (eigene Darstellung)

Manche Propositionen von Fitts' Law konnten ebenfalls integriert werden. Um die Distanz zwischen User und Simulationsmenü zu minimieren, wurde das Feature implementiert, dass bei der Betätigung des A-Knopfes des rechten Controllers das Menü in das Blickfeld des Users springt. Diese Eigenschaft folgt der Empfehlung, dass der Cursor (in diesem Fall: das Zentrum des Blickfelds) der Ursprung von Bedienelementen sein soll (siehe Kapitel 3.3), wodurch der Index of Difficulty gegen null geht (siehe Formel (3)). Die Skalierbarkeit der Menüs beeinflusst zusätzlich die Größe aller *Buttons* und *Trigger*. Folglich reduziert sich die nötige Zeit, die ein Nutzer braucht, um zu ihnen zu navigieren. Zur Vorbeugung der besagten Cybersickness wurde die Möglichkeit zur Teleportation in der *Scene* integriert. Dadurch sind die Nutzer nicht darauf angewiesen, sich per Joystick fortzubewegen. Andernfalls könnte bei anfälligen Personen die vorbeiziehende Umgebung die benannten Symptome verursachen. Von besonderer Bedeutung war zudem, dass das Standalone-Programm flüssig auf der Quest 3 läuft. Methoden der Performance-Optimierung sollten die FPS maximieren.

5.4 Performance-Optimierung

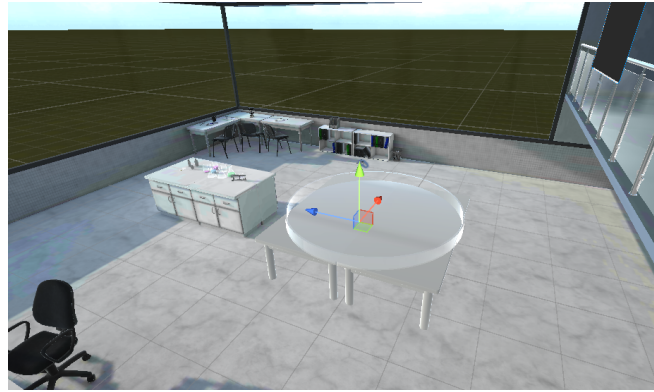
Wenn Standalone-VR-Anwendungen entwickelt werden, ist es immer eine Herausforderung, den Rechenaufwand zur Laufzeit des Programms so klein wie möglich zu halten. Unity bietet deshalb selbst Methoden an, um die Performance zu erhöhen. MicroLabVR verwendet daher die URP¹⁴, statt der High Definition Render-Pipeline (HDRP)¹⁵. Denn für eine VR-Anwendung in diesem wissenschaftlichen Kontext sind hochauflösende und realitätsnahe Texturen und Effekte nicht notwendig. Wenn die Szenengestaltung realistische *Materials* bedurfte, wurden öffentlich verfügbare Physically Based Rendering-*Materials* von der Website „Free PBR Materials“¹⁶ verwendet. Auf diese Weise konnte Objekten prägnantere Texturen gegeben werden, ohne viel Rechenleistung zu fordern. Da der Nutzer immer nur einen Teil der *Scene* sieht, wurde ebenfalls das Occlusion Culling¹⁷ in MicroLabVR integriert. Occlusion Culling ist eine Methode, die verhindert, dass Objekte gerendert werden, die außerhalb des Sichtfelds liegen (siehe Abbildung 26). Der Nutzen von Occlusion Culling beläuft sich jedoch vermutlich auf ein Minimum, da er mit der Größe der *Scene* skaliert; die Laborszene ist verhältnismäßig klein.

¹⁴<https://unity.com/de/srp/universal-render-pipeline> (letzter Zugriff 17.10.24)

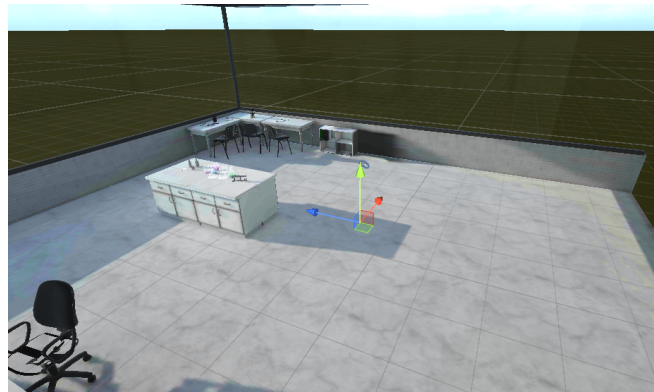
¹⁵<https://unity.com/de/srp/high-definition-render-pipeline> (letzter Zugriff 17.10.24)

¹⁶<https://freepbr.com/> (letzter Zugriff 17.10.24)

¹⁷<https://docs.unity3d.com/Manual/OcclusionCulling.html> (letzter Zugriff 17.10.24)



(a) Occlusion Culling ist nicht aktiviert



(b) Occlusion Culling ist aktiviert

Abbildung 26: Der Effekt von Occlusion Culling. Die Blickrichtung des Users geht entlang des blauen Pfeils (eigene Darstellung)

Die Lichtmodi aller Lichtquellen der *Scene* wurden auf „baked“¹⁸ gesetzt, was bedeutet, dass die Beleuchtung der Szene manuell vorab berechnet wird. Dieser Prozess kann vor der Finalisierung der Anwendung einige Minuten bis Stunden in Anspruch nehmen. Zur Laufzeit werden so allerdings die Echtzeitberechnungen des Lichts gespart.

Weitere kleinere Verbesserungen, die vorgenommen wurden, sind zum einen, dass die Organismen keine Schatten werfen und zum anderen, dass „GPU Instancing“¹⁹ bei allen Organismus-*Materials* aktiviert wurde. Letzteres ist ein Verfahren, um die Menge an Draw Calls (dt. Aufrufen) an die GPU zu reduzieren. Dabei werden mehrfach vorhandene Instanzen eines *Mesh* mit demselben *Material* in einem Render-Aufruf verarbeitet. Diese Methode ist im Fall von MicroLabVR besonders nützlich, da mehrere tausend Organismen erzeugt werden.

¹⁸<https://docs.unity3d.com/2017.2/Documentation/Manual/LightMode-Baked.html> (letzter Zugriff 17.10.24)

¹⁹<https://docs.unity3d.com/Manual/GPUInstancing.html> (letzter Zugriff 17.10.24)

5.5 Probleme und Herausforderungen

Nicht alle Anforderungen der Konzeption konnten zeitlich umgesetzt werden und bei manchen kamen Probleme auf. Eines der Probleme war das bereits erwähnte fehlerhafte Mapping zwischen den Organismen und den metabolischen Flüssen in BacArena. Deshalb erstellt der Konverter nur den Populationsdatensatz, ohne die Flüsse einzufügen. Um einen vollständigen Datensatz zu erstellen, wurden alle sechs Spalten der Flussdaten mit zufälligen Werten gefüllt.

Ebenfalls wurde das metabolische Zusammenspiel zwischen den Organismen nicht visualisiert. Der Grund dafür ist, dass zwischen den Zeitschritten nicht nachverfolgt werden kann, welche Organismen an welchen Ort in der Petrischale wandern oder absterben, da sie nicht mit einer eindeutigen ID gekennzeichnet sind. Aus diesem Grund kann auch der Austausch zwischen den Organismen nicht nachverfolgt werden. Diese Art der Nachverfolgung ist ohnehin in der Realität untypisch, weil symbiotische Beziehungen eher zwischen ganzen Spezies bestehen als zwischen einzelnen Organismen. Da dieses Feature fehlt, wurden auch keine Flussgradienten visualisiert. Die Berechnung einer allgemeinen Flussrichtung erfordert einen komplexen Algorithmus, welcher nicht entwickelt werden konnte.

Ziel der Anwendung war zudem, dass benutzerdefinierte Datensätze geladen werden können. Die für diese Eigenschaft nötigen Funktionen wurden ebenfalls implementiert und sollten ursprünglich vom „Open File Browser“-*Button* ausgeführt werden. Da jedoch der Zugriff auf das lokale Dateisystem der Quest 3 verweigert wird, können keine Datensätze ausgewählt werden. Um dieses Problem zu beheben, wurde die Manifest-Datei der Anwendung editiert, sodass einmalig eine Eingabeaufforderung beim Start von Micro-LabVR erschien. Diese Anzeige fragte, ob Zugriff auf die lokalen Dateien erlaubt werden soll. Nach der Bestätigung bestand dennoch kein Zugriff und die Eingabeaufforderung erschien nie wieder. Das Problem konnte final nicht gelöst werden, weshalb die Anwendung nur den Import der beiden Demodatensätze erlaubt.

Wenn in Zukunft das Laden benutzerdefinierter Datensätze möglich ist, muss ebenfalls die Skalierung der Organismen korrigiert werden. Da alle Bakterien in der Petrischale platziert werden, muss der Skalierungsfaktor der 3D-Modelle neu berechnet werden, um sie korrekt auszufüllen. Die zuständige Methode ist derzeit etwas ungenau, weshalb je nach gewählter Größe des Simulationsbereichs das Mikrobiom nicht zentriert in der Schale positioniert ist. Die korrekte Position haben alle Organismen in der Petrischale nur bei einer Fläche von 20×20 .

In Testläufen ist zudem aufgefallen, dass der Reset des ursprünglichen *Materials*, nachdem über einen Organismus gehovert und zuvor die Flux-Anzeige ausgeschaltet wurde, fehlerhaft ist. Anstatt, dass das schwarze Outline-*Material* verschwindet, behält der Organismus die Färbung, passend zu seinem metabolischen Verhalten bei und ist als Einziger umrandet (siehe Abbildung 27). Die Ursache konnte nicht identifiziert werden.

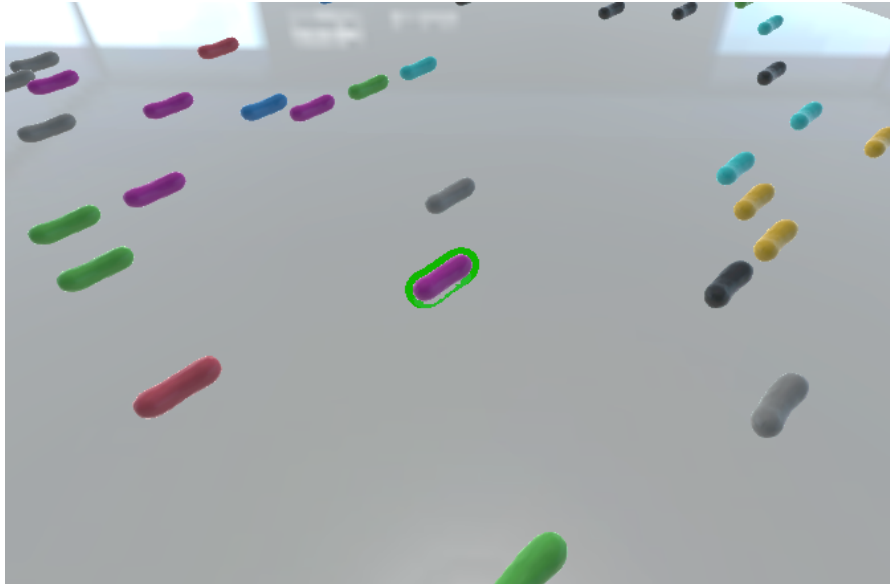


Abbildung 27: Das violette Bakterium in der Mitte ist als einziges fehlerhaft grün markiert (eigene Darstellung)

6 Limit-Testing

Interessant im Hinblick auf die potentielle Möglichkeit, benutzerdefinierte Datensätze zu importieren, ist, welche Größe diese Datensätze annehmen dürfen, damit MicroLabVR performant läuft. Relevant ist in diesem Kontext nur der Populationsdatensatz, da anhand seiner Daten eine Vielzahl von *GameObjects* instanziiert werden. Aus diesem Grund sollen die Limitationen der Anwendung geprüft werden.

6.1 Methodik

MicroLabVR ist als Standalone-Anwendung gedacht. Deshalb ist ein Performance-Vergleich zu einer PC-VR-Version aufschlussreich. Es wurde dazu die gebaute finale Version von MicroLabVR auf der Meta Quest 3, mit der gebauten PC-VR-Version getestet. Der PC, der für die Performance-Tests verwendet wurde, hat folgende technische Spezifikationen: Eine 12th Gen Intel® Core™ i7-12700KF CPU, eine NVIDIA GeForce RTX 4070 Ti 12 G GPU, 32 GB DDR4 RAM, eine 1 TB SSD, den AZZA Blizzard 240 mm CPU-Kühler, das Gigabyte B760M DS3H DDR4 Motherboard und ein 1000 Watt be quiet! Straight Power 11 Platin Netzteil. Die technischen Spezifikationen der Quest 3 wurden bereits in Kapitel 2.2.5 aufgeführt.

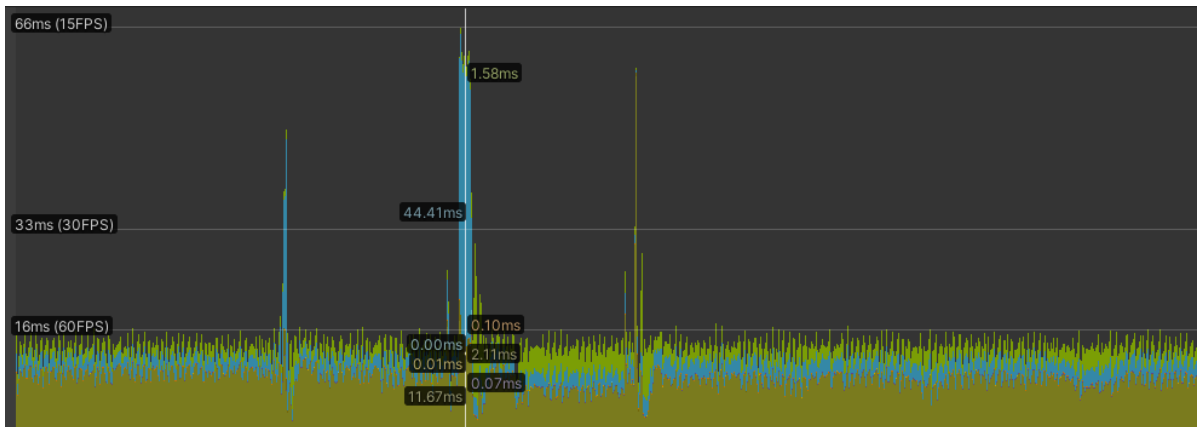
Um die Limitationen zu testen, wurden Datensätze verschiedener Größen erstellt. Daher wurde ein R-Skript geschrieben, welches den Populations- und den Substanzdatensatz mit zufälligen Werten erzeugt. Während der Tests wurden verschiedene Werte gemessen. Zunächst wurde die Zeilenzahl der Populationsdatensätze betrachtet und die daraus resultierende Anzahl an *GameObjects*. Danach wurde gemessen, wie lange der Import der Datensätze und das Setup der *Scene* dauert. Dabei wurde darauf geachtet, wie sich die Anwendung verhält: Läuft sie flüssig weiter? Stottert das Bild? Kommt es zu einem Standbild? Stürzt sie sogar ab? Die wichtigsten Parameter waren jedoch die FPS während der Simulation. Anhand dieser Messdaten, die über den Unity-Profiler²⁰ entnommen wurden, konnte letztendlich eine Empfehlung für die Datensatzgröße gegeben werden.

6.2 Ergebnisse

Es wurden final drei Größenvarianten von Datensätzen erstellt, mit 1389 (D_{Small}), 10600 (D_{Medium}) und 48000 (D_{Large}) Zeilen (siehe Tabelle 1). Die Performance aller Datensätze wurde mit dem Unity-Profiler in einem Developer-Build der Anwendung überwacht. Seine

²⁰<https://docs.unity3d.com/Manual/Profiler.html> (letzter Zugriff 20.10.2024)

x-Achse zeigt jeden Frame und die y-Achse die Dauer des Frames in Millisekunden (ms) (siehe Abbildung 28). Bei jedem Test wurde derselbe Ablauf gewählt: Erst wurde der Demodatensatz importiert, die Simulation gestartet, die maximale Anzahl an *GameObjects* erzeugt und daraufhin sich in der Anwendung umgeschaut. Dieser Prozess wurde zunächst auf der Quest 3 ausgeführt und danach auf dem PC. Wie erwartet sinkt die Performance von MicroLabVR mit zunehmender Datensatzgröße auf beiden Geräten (siehe Abbildung 28).



(a) Meta Quest 3: D_{Small}



(b) Meta Quest 3: D_{Large}

Abbildung 28: Die Historie des Profilers, während MicroLabVR D_{Small} vs. D_{Large} im Standalone-Modus verwendet wird (eigene Darstellung)

Ebenso wie erwartet, läuft die Anwendung flüssiger über den PC. Auffällig war dabei, dass die Zeit zum Vorbereiten der Simulation (nach Betätigen des „Start“-Buttons) hinreichend länger dauert als der Datensatzimport (siehe Tabelle 1).

Tabelle 1: Die Performance der Anwendung in Abhängigkeit der Größe des Populationsdatensatzes: Dimension des Simulationsbereichs (Dim) Anzahl instanzierter *GameObjects* (n), Zeit zum Import der Datensätze über die Quest 3 (t_1^Q) und über den Computer (t_1^C), Zeit zum Instanzieren der *GameObjects* über die Quest 3 (t_2^Q) und über den Computer t_2^C . FPS während die Simulation auf der Quest 3 (FPS^Q) und auf dem PC (FPS^P) läuft.

Dim	$Zeilen$	n	t_1^Q	t_1^P	t_2^Q	t_2^P	FPS^Q	FPS^P
20×20	1389	392	0,12 s	0,06 s	0,59 s	0,3 s	71,43	76,92
50×50	10600	2500	0,2 s	0,15 s	1,3 s	0,45 s	33,33	76,92
100×100	48000	10000	2,2 s	0,9 s	162,7 s	72,9 s	11,21	71,42

Aus den gemessenen Zeiten der Frames konnten die FPS während der rechenintensiven Intervalle mit folgender Formel berechnet werden:

$$FPS = \frac{1 \text{ Sekunde}}{\text{Durchschnittsdauer der Frames}} \quad (4)$$

Für die Entscheidung über einen Grenzwert der Datensatzgröße müssen mehrere ausschlaggebende Aspekte betrachtet werden: die Ladezeit der Datensätze und der Simulation sowie die Höhe der FPS. Während der Limit-Tests war folgenreich, dass D_{Large} eine zu lange Ladezeit der Simulation hatte. In dieser Zeit hatte der Nutzer ein Standbild, welches nicht tolerierbar ist. Zwar sind die FPS im PC-VR-Modus über 70, im Standalone-Modus allerdings bei gerade einmal 11,21. Nur D_{Small} performte durchweg zufriedenstellend, da D_{Medium} in der Standalone-Anwendung auch nur 33,33 FPS erreichen konnte. Für manche Nutzer mag dies ausreichend sein, sollte aber nicht der Standard für VR-Anwendungen sein.

6.3 Fazit

Schlussendlich ist zu sagen, dass alle Datensätze im PC-VR-Modus verwendet werden können, wenn während des Imports und des Setups der Simulation ein flüssiger Ladebildschirm angezeigt wird. Als Standalone-Anwendung erfüllt nur D_{Small} alle Performance-Anforderungen. Während beider Ladephasen sollte dem Nutzer eine Ladeanzeige präsentiert werden, um FPS-Einbrüche und potentielle Standbilder unbemerkt zu lassen.

7 Diskussion

MicroLabVR umfasst alle konzipierten Features. Alle Organismen werden in ihrer richtigen Konstellation als interaktionsfähige *GameObjects* dargestellt, auch wenn sie in manchen Fällen nicht korrekt zentriert sind. Über ein UI lassen sich die Substanzverteilungen und Flüsse anzeigen. Verbesserungspotential hat insbesondere die metabolische Flussdarstellung. Sie wird lediglich mit einer Umrandung der Organismen abgebildet. Informativer wäre eine konkrete Visualisierung von Austauschreaktionen. Dazu wären jedoch mehr Daten von BacArena und ein komplexer Algorithmus nötig, welcher den Austausch zwischen ganzen Spezies berechnet.

Die Anwendung liefert entscheidende Vorteile im Vergleich zu einer Desktopanwendung, wie BacArena. Zunächst erlaubt MicroLabVR eine einfache und intuitive Navigation und Steuerung der Funktionen. Ein User-Tutorial unterstützt die Selbstbeschreibungsfähigkeit und Erlernbarkeit des Programms. Über den *Slider* kann der Nutzer einfacher durch alle Zeitpunkte iterieren und muss sich nicht auf graphische Plots verlassen. Potentiell kann so die Entwicklung und sogar die Bewegung mancher Organismen verfolgt werden, auch wenn sich das nicht sicher bestätigen lässt. Nachteilig ist dabei, dass nicht alle Zeitpunkte gleichzeitig angezeigt werden können. MicroLabVR eignet sich daher als zusätzliches Hilfsmittel, welches mikrobielle Daten aus einer anderen Perspektive darstellt.

Ein wesentlicher Vorteil der VR ist, dass sie die Informationsdarstellung in der dritten Dimension ermöglicht. 3D-Daten auf 2D-Oberflächen zu projizieren, kann die Interpretierbarkeit einschränken. Dieser Vorteil trifft nicht unbedingt zum aktuellen Zeitpunkt bei MicroLabVR zu, da nur eine Schicht eines Mikrobioms dargestellt werden kann. Der Grund dafür liegt in der Wahl des Simulations-Tools. Das Framework, um 3D-Daten zu importieren und darzustellen, in MicroLabVR zu implementieren, ist perspektivisch realisierbar. Es sei jedoch zunächst abzuwarten, ob der Bedarf besteht. Außerdem wäre zu diskutieren, wie die 3D-Visualisierung ausgereift und adäquat implementiert werden soll.

Sinnvoll an der VR-Anwendung erscheint besonders die 3D-Heatmap-Darstellung, da sie direkten Nutzen von einer räumlichen Umgebung macht. Der User bekommt eine bessere Darstellung der Intensitätswerte, indem das Größenverhältnis der Werte besser dargestellt wird. Zudem lässt sich die Population über die 2D-Heatmap legen, wodurch in manchen Fällen ggf. eine Migration in Richtung einer Nahrungsquelle zu erkennen ist. Neben dem besseren Verständnis räumlicher Beziehungen, profitieren Nutzer von der Immersion, indem ein realistisches Lernerlebnis geschaffen wird. Das sogenannte „Situating

Learning“ (dt. situatives Lernen)²¹ kann das Engagement von Nutzern und die Effektivität des Lernens erhöhen. Daher kann darüber nachgedacht werden, ob MicroLabVR auch Potential in der Lehre von Studierenden hat. Ob Wissenschaftler diese theoretischen Vorteile ebenso empfinden, muss noch erprobt werden.

Vorschriften des UX-Designs konnten zum Teil umgesetzt werden. Es wurde stetig darauf geachtet, dass eine barrierefreie Anwendung entsteht. Farbkontrastverhältnisse liegen minimal bei 4,88:1. Um auf ein Level AAA zu kommen, könnte die Farbe des Buttons des User-Tutorials dunkler gesetzt werden. Nützlich für den User sind ebenfalls kleinere Features, wie die Umrandung eines Organismus gepaart mit vibrotaktilen Feedback, wenn mit dem Controller auf ihn gezeigt wird sowie die Schwebanimation, wenn er angeklickt wird. Diese Formen des Feedbacks könnte noch durch das Abspielen von Tönen, die zur Aktion passen, erweitert werden. Nur die Bedienelemente der UIs hatten dieses Feature bereits implementiert. Eine weitere Verbesserung wäre, das User-Tutorial auf eine gesamte Führung durch die *Scene* zu erweitern. Da MicroLabVR allerdings nur aus einer *Scene* besteht, scheint dies weniger sinnvoll. Alternativ wäre eine Audio-Führung eine praktische Verbesserung.

Ein Mangel der UX in MicroLabVR könnte in der Farbwahl der Bakterien und der Anzeige der Flüsse gesehen werden. Für Betroffene einer anomalen Trichromasie (Rot-Grün-Schwäche) wäre sie ein Problem, da rote und grüne *Materials* benutzt wurden. Für sie könnte kaum unterscheidbar sein, ob ein Organismus eine Substanz konsumiert oder produziert. Lediglich bei der Darstellung der Substanzkonzentration konnte eine Farbauswahl implementiert werden.

Mit zunehmender Datensatzgröße sinkt sowohl die Performance der Anwendung, als auch die Größe der Organismen. Schon bei einem Simulationsbereich, der 100×100 Bakterien umfasst, kann es nach Fitts' Law schwierig werden, die *GameObjects* anzuklicken. Daher müsste eine Untergrenze für die Organismengröße festgelegt werden, um die Steuerbarkeit zu gewährleisten. Diese Änderung würde auch bewirken, dass die Petrischale und damit die gesamte *Scene* skalierbar werden müssen. Während der Entwicklung von MicroLabVR traten verschiedene technische Probleme auf. Zunächst war angedacht, dass MicroLabVR auch benutzerdefinierte Datensätze einlesen kann, um den Nutzen der Anwendung zu erweitern. Wie bereits in Kapitel 5.5 beschrieben, erlaubt die Quest 3 keinen Zugriff auf das Dateisystem, weshalb die dort hinterlegten Datensätze nicht eingelesen werden konnten. Eine Alternativlösung könnte die Verwendung des Storage Access Frameworks²²

²¹Mehr zu Situated Learning: <https://ijiet.org/papers/48-R017.pdf> (letzter Zugriff 22.10.2024)

²²<https://developer.android.com/guide/topics/providers/document-provider?hl=de> (letzter Zugriff 24.10.2024)

für Android-Entwickler sein. Auf diese Weise könnten ggf. spezifische freigegebene Verzeichnisse („Downloads“ oder „Dokumente“) verfügbar gemacht werden. Eine umfassende Lösung wäre die Anbindung eines Cloud-Speichers, auf dem alle Datensätze hinterlegt sind. Herausfordernd wäre, eine passende Schnittstelle zwischen der Anwendung und dem Speicher zu entwickeln. Abseits davon wäre der Dateizugriff über den PC einfacher. Es existieren einige *Assets*, die in Betracht gezogen werden können, um einen Datei-Browser zu integrieren. Der Standalone File Browser²³ wurde in MicroLabVR auch zu Beginn eingebunden, da es eine Standalone-Anwendung werden sollte, aber aufgrund von technischen Problemen wieder verworfen. Das Framework hierzu existiert noch.

Ein weiteres Problem war, dass das Mapping der Flussdaten von BacArena inkorrekt ist. Der entwickelte Konverter erzeugt daher nur Datensätze, die keine metabolischen Flüsse aufweisen. Die VR-Anwendung ist aus diesem Grund auch nur auf sechs Substanzen und Flüsse beschränkt. Würden Dateien mit mehr Substanzen eingelesen werden, fängt der Datenformat-Checker sie ab.

²³<https://github.com/gkngkc/UnityStandaloneFileBrowser/tree/master?tab=readme-ov-file>
(letzter Zugriff 22.10.2024)

8 Fazit und Ausblick

MicroLabVR ist eine in Unity entwickelte Standalone-VR-Anwendung, spezialisiert auf die Meta Quest 3, welche raumzeitliche Daten über Mikrobiome einlesen und darstellen kann, um mangelnde Datenvisualisierung anderer Tools entgegenzuwirken. Die Anwendung unterstützt Datensätze, die einem definierten Format folgen. Passende Datensätze können mittels eines R-Konverters erzeugt und eingelesen werden. Der Konverter ist auf dem agentenbasierten Simulationstool BacArena aufgebaut, das in R als Bibliothek verfügbar ist. Über den Konverter lassen sich zwei Datensätze erzeugen, welche Daten über die Gemeinschafts-, Substanz- und metabolische Flussverteilung enthalten. Diese Daten werden auf unterschiedliche Weisen dargestellt. Organismen sind einfache, interaktive *GameObjects*, Substanzen sind 2D- oder 3D-*Meshes*, und Fluxes werden als farbliche Markierungen visualisiert. Das Programm verfügt über ein User-Tutorial, ein Hauptmenü, welches den Datenimport regelt sowie ein Simulationsmenü, das den Simulationsverlauf steuert. Diese UIs folgen stets bewährten UI- und UX-Leitlinien und erzeugen einen gewissen Grad an Barrierefreiheit.

Im Hinblick auf funktionelle Erweiterungen von MicroLabVR wäre eine Expansion auf 3D-Schichten interessant. So könnte das volle Potential der VR ausgeschöpft werden. Problematisch würde nur die Darstellung der Substanzen werden. Denn nun wäre nicht nur eine Konzentration einer Fläche abzubilden, sondern eines Volumens. Spannend wäre ebenfalls, wenn die Mikrobiome auf einen Ursprungsort gemappt werden könnten, wie beispielsweise auf die Lunge, oder den Darm. Bei größeren Datensätzen könnten so regionale Unterschiede erkennbar werden.

Eine Visualisierungsmethode, die etwas zu kurz kommt, ist die Darstellung der Flussverteilung. Die Organismen werden lediglich farblich markiert, um den Konsum oder die Produktion einer Substanz zu indizieren. Informativer wäre die Illustration von Flussrichtungen und -intensitäten. Auf diese Weise könnten ganze Gradienten dargestellt werden, was die Datenauswertung verbessern könnte.

Ein erster Performancetest ergab, dass die Datensätze zunächst nur sehr klein (weniger als insgesamt 10000 instantiierte Organismen) sein können. MicroLabVR verfolgt bereits einen Ansatz für das asynchrone Laden der Datensätze über *Coroutinen*. Dennoch kommt es zu FPS-Einbrüchen während der Ladezeit. Dieser Mangel suggeriert, dass die Coroutine-Logik noch nicht ausgereift ist und optimiert werden muss. Eine alternative selbst implementierte Lösung zum asynchronen Laden (unter der Verwendung vorgefertigter *Assets*), gepaart mit einem angemessenen Datenstreaming von BacArena

selbst, würde die Leistung noch stärker heben. Anhand dieser Fortschritte könnte die Performance und die UX verbessert werden. Zwar ist ein Ladebalken integriert, um dem Nutzer Feedback über den Status zu geben, dennoch wäre eine zusätzliche Möglichkeit, den Nutzer mit einem Ladebildschirm gänzlich vom Geschehen zu isolieren.

Dieser Grundgedanke würde zwar die Ladeperformance verbessern, aber nicht die Frame-rate während der Simulation. Ein Verbesserungsansatz ist, das entworfene 3D-Modell der Organismen zu optimieren. Derzeit verfügt es über 832 Polygone. Bei 10.000 *GameObjects* wären das 8.320.000 Polygone, die gerendert werden müssen. Die Anzahl könnte bereits mit einfachen Optimierungsmethoden in Blender reduziert werden.

Neben der Behebung der in Kapitel 5.5 beschriebenen technischen Probleme wäre ein Vorschlag, die Usability zu verbessern, indem strenger auf die Farbwahl der Organismen geachtet wird. Gerade wenn perspektivisch betrachtet größere Datensätze eingelesen werden sollen, muss die Farbzweisung einem Algorithmus folgen, der angesichts der Speziesanzahl die bestmögliche Farbkombination erzeugt. Verschiedene Shader und Texturen könnten die Unterscheidung zusätzlich vereinfachen.

Sobald BacArena das Mapping der Flüsse zu den Organismen behoben hat, kann auch der R-Konverter angemessen erweitert werden. Derzeit muss der Anwender außerdem eigenständig im R-Konverter den Speicherort und das Modell setzen. Für den idealen Nutzen sollte jedoch eine einfache nutzbare Schnittstelle erschaffen werden, die den direkten Datenaustausch mit MicroLabVR ermöglicht und keine Programmierkenntnisse fordert. Somit würden Nutzer der VR-Anwendung mit einer noch besseren UX belohnt. MicroLabVR dient als Proof of Concept und bestätigt die Machbarkeit sowie die grundlegende Funktionalität der erweiterten Visualisierung von mikrobiellen Daten. Ob weitere Wissenschaftler die diskutierten Vorteile der Anwendung vertreten, muss sich zunächst noch erweisen. Daher wären kontrollierte Experteninterviews zweckhaft, um den Nutzen einer solchen Anwendung in der Praxis und ggf. in der Lehre weiter zu erforschen.

Literatur

- [1] Jee Loon Foo u. a. “Microbiome engineering: Current applications and its future”. In: *Biotechnology Journal* 12.3 (Jan. 2017). ISSN: 1860-7314. DOI: 10.1002/biot.201600099. URL: <http://dx.doi.org/10.1002/biot.201600099>.
- [2] John C. Wooley, Adam Godzik und Iddo Friedberg. “A Primer on Metagenomics”. In: *PLoS Computational Biology* 6.2 (Feb. 2010). Hrsg. von Philip E. Bourne, e1000667. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000667. URL: <http://dx.doi.org/10.1371/journal.pcbi.1000667>.
- [3] Lu Zhang u. a. “Advances in Metagenomics and Its Application in Environmental Microorganisms”. In: *Frontiers in Microbiology* 12 (Dez. 2021). ISSN: 1664-302X. DOI: 10.3389/fmicb.2021.766364. URL: <http://dx.doi.org/10.3389/fmicb.2021.766364>.
- [4] Mary Ann Moran u. a. “Sizing up metatranscriptomics”. In: *The ISME Journal* 7.2 (Aug. 2012), S. 237–243. ISSN: 1751-7370. DOI: 10.1038/ismej.2012.94. URL: <http://dx.doi.org/10.1038/ismej.2012.94>.
- [5] Teija Ojala, Esko Kankuri und Matti Kankainen. “Understanding human health through metatranscriptomics”. In: *Trends in Molecular Medicine* 29.5 (Mai 2023), S. 376–389. ISSN: 1471-4914. DOI: 10.1016/j.molmed.2023.02.002. URL: <http://dx.doi.org/10.1016/j.molmed.2023.02.002>.
- [6] Yuqiu Wang u. a. “Metaproteomics: A strategy to study the taxonomy and functionality of the gut microbiota”. In: *Journal of Proteomics* 219 (Mai 2020), S. 103737. ISSN: 1874-3919. DOI: 10.1016/j.jprot.2020.103737. URL: <http://dx.doi.org/10.1016/j.jprot.2020.103737>.
- [7] Manish Kumar u. a. “Modelling approaches for studying the microbiome”. In: *Nature Microbiology* 4.8 (Juli 2019), S. 1253–1267. ISSN: 2058-5276. DOI: 10.1038/s41564-019-0491-9. URL: <http://dx.doi.org/10.1038/s41564-019-0491-9>.
- [8] Brian Fabien. *Analytical System Dynamics: Modeling and Simulation*. Springer US, 2009. ISBN: 9780387856056. DOI: 10.1007/978-0-387-85605-6. URL: <http://dx.doi.org/10.1007/978-0-387-85605-6>.
- [9] Claire J. Tomlin und Jeffrey D. Axelrod. “Biology by numbers: mathematical modelling in developmental biology”. In: *Nature Reviews Genetics* 8.5 (Mai 2007), S. 331–340. ISSN: 1471-0064. DOI: 10.1038/nrg2098. URL: <http://dx.doi.org/10.1038/nrg2098>.

- [10] Eugen Bauer u. a. “BacArena: Individual-based metabolic modeling of heterogeneous microbes in complex communities”. In: *PLoS Computational Biology* 13.5 (Mai 2017). Hrsg. von Costas D. Maranas, e1005544. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005544. URL: <http://dx.doi.org/10.1371/journal.pcbi.1005544>.
- [11] Sebastian Pirch u. a. “The VRNetzer platform enables interactive network analysis in Virtual Reality”. In: *Nature Communications* 12.1 (Apr. 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-021-22570-w. URL: <http://dx.doi.org/10.1038/s41467-021-22570-w>.
- [12] Matthew B. Biggs und Jason A. Papin. “Novel Multiscale Modeling Tool Applied to Pseudomonas aeruginosa Biofilm Formation”. In: *PLoS ONE* 8.10 (Okt. 2013). Hrsg. von Michael Hensel, e78011. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0078011. URL: <http://dx.doi.org/10.1371/journal.pone.0078011>.
- [13] Greg Wilson u. a. “Best Practices for Scientific Computing”. In: *PLoS Biology* 12.1 (Jan. 2014). Hrsg. von Jonathan A. Eisen, e1001745. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1001745. URL: <http://dx.doi.org/10.1371/journal.pbio.1001745>.
- [14] Serghei Mangul u. a. “Challenges and recommendations to improve the installability and archival stability of omics computational tools”. In: *PLoS Biology* 17.6 (Juni 2019), e3000333. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.3000333. URL: <http://dx.doi.org/10.1371/journal.pbio.3000333>.
- [15] Lindsay Barone, Jason Williams und David Micklos. “Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators”. In: *PLoS Computational Biology* 13.10 (Okt. 2017). Hrsg. von Francis Ouellette, e1005755. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005755. URL: <http://dx.doi.org/10.1371/journal.pcbi.1005755>.
- [16] Ciro Donalek u. a. “Immersive and collaborative data visualization using virtual reality platforms”. In: *2014 IEEE International Conference on Big Data (Big Data)*. IEEE, Okt. 2014. DOI: 10.1109/bigdata.2014.7004282. URL: <http://dx.doi.org/10.1109/BigData.2014.7004282>.
- [17] Michael Aichem u. a. “Towards a hybrid user interface for the visual exploration of large biomolecular networks using virtual reality”. In: *Journal of Integrative Bioinformatics* 19.4 (Okt. 2022). ISSN: 1613-4516. DOI: 10.1515/jib-2022-0034. URL: <http://dx.doi.org/10.1515/jib-2022-0034>.

- [18] Devon Allcoat und Adrian von Mühlenen. “Learning in virtual reality: Effects on performance, emotion and engagement”. In: *Research in Learning Technology* 26.0 (Nov. 2018). ISSN: 2156-7077. DOI: 10.25304/rlt.v26.2140. URL: <http://dx.doi.org/10.25304/rlt.v26.2140>.
- [19] Brendan Mouatt u. a. “The Use of Virtual Reality to Influence Motivation, Affect, Enjoyment, and Engagement During Exercise: A Scoping Review”. In: *Frontiers in Virtual Reality* 1 (Dez. 2020). ISSN: 2673-4192. DOI: 10.3389/frvir.2020.564664. URL: <http://dx.doi.org/10.3389/frvir.2020.564664>.
- [20] Julian R. Marchesi und Jacques Ravel. “The vocabulary of microbiome research: a proposal”. In: *Microbiome* 3.1 (Juli 2015). ISSN: 2049-2618. DOI: 10.1186/s40168-015-0094-5. URL: <http://dx.doi.org/10.1186/s40168-015-0094-5>.
- [21] John Dupré und Maureen A. O’Malley. “Varieties of Living Things: Life at the Intersection of Lineage and Metabolism”. In: *History, Philosophy and Theory of the Life Sciences*. Springer Netherlands, 2013, S. 311–343. ISBN: 9789400724457. DOI: 10.1007/978-94-007-2445-7_13. URL: http://dx.doi.org/10.1007/978-94-007-2445-7_13.
- [22] Gabriele Berg u. a. “Microbiome definition re-visited: old concepts and new challenges”. In: *Microbiome* 8.1 (Juni 2020). ISSN: 2049-2618. DOI: 10.1186/s40168-020-00875-0. URL: <http://dx.doi.org/10.1186/s40168-020-00875-0>.
- [23] J. Whipps, K. Lewis und R. Cooke. “Mycoparasitism and plant disease control”. In: *Burge M, editor. Fungi Biol Control Syst*. Manchester University Press, 1988, S. 161–187.
- [24] Brajesh K. Singh u. a. “Crop microbiome and sustainable agriculture”. In: *Nature Reviews Microbiology* 18.11 (Okt. 2020), S. 601–602. ISSN: 1740-1534. DOI: 10.1038/s41579-020-00446-y. URL: <http://dx.doi.org/10.1038/s41579-020-00446-y>.
- [25] Vincent B Young. “The role of the microbiome in human health and disease: an introduction for clinicians”. In: *BMJ* (März 2017), j831. ISSN: 1756-1833. DOI: 10.1136/bmj.j831. URL: <http://dx.doi.org/10.1136/bmj.j831>.
- [26] Kaijian Hou u. a. “Microbiota in health and diseases”. In: *Signal Transduction and Targeted Therapy* 7.1 (Apr. 2022). ISSN: 2059-3635. DOI: 10.1038/s41392-022-00974-4. URL: <http://dx.doi.org/10.1038/s41392-022-00974-4>.

- [27] Thomas Müller und Silke Ruppel. “Progress in cultivation-independent phyllosphere microbiology”. In: *FEMS Microbiology Ecology* 87.1 (Sep. 2013), S. 2–17. ISSN: 0168-6496. DOI: 10.1111/1574-6941.12198. URL: <http://dx.doi.org/10.1111/1574-6941.12198>.
- [28] Sijia Liu u. a. “Opportunities and challenges of using metagenomic data to bring uncultured microbes into cultivation”. In: *Microbiome* 10.1 (Mai 2022). ISSN: 2049-2618. DOI: 10.1186/s40168-022-01272-5. URL: <http://dx.doi.org/10.1186/s40168-022-01272-5>.
- [29] Vanni Bucci und Joao B. Xavier. “Towards Predictive Models of the Human Gut Microbiome”. In: *Journal of Molecular Biology* 426.23 (Nov. 2014), S. 3907–3916. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2014.03.017. URL: <http://dx.doi.org/10.1016/j.jmb.2014.03.017>.
- [30] Chrysanthi-Maria Moysidou und Róisín M. Owens. “Advances in modelling the human microbiome–gut–brain axis in vitro”. In: *Biochemical Society Transactions* 49.1 (Feb. 2021), S. 187–201. ISSN: 1470-8752. DOI: 10.1042/bst20200338. URL: <http://dx.doi.org/10.1042/BST20200338>.
- [31] Sherli Koshy-Chenthittayil u. a. “Agent Based Models of Polymicrobial Biofilms and the Microbiome—A Review”. In: *Microorganisms* 9.2 (Feb. 2021), S. 417. ISSN: 2076-2607. DOI: 10.3390/microorganisms9020417. URL: <http://dx.doi.org/10.3390/microorganisms9020417>.
- [32] Jeffrey D Orth, Ines Thiele und Bernhard Ø Palsson. “What is flux balance analysis?” In: *Nature Biotechnology* 28.3 (März 2010), S. 245–248. ISSN: 1546-1696. DOI: 10.1038/nbt.1614. URL: <http://dx.doi.org/10.1038/nbt.1614>.
- [33] Kai Zhuang u. a. “Genome-scale dynamic modeling of the competition between *Rhodospirillum rubrum* and *Geobacter* in anoxic subsurface environments”. In: *The ISME Journal* 5.2 (Juli 2010), S. 305–316. ISSN: 1751-7370. DOI: 10.1038/ismej.2010.117. URL: <http://dx.doi.org/10.1038/ismej.2010.117>.
- [34] Sergey Stoliar u. a. “Metabolic modeling of a mutualistic microbial community”. In: *Molecular Systems Biology* 3.1 (Jan. 2007). ISSN: 1744-4292. DOI: 10.1038/msb4100131. URL: <http://dx.doi.org/10.1038/msb4100131>.
- [35] Edwin H Wintermute und Pamela A Silver. “Emergent cooperation in microbial metabolism”. In: *Molecular Systems Biology* 6.1 (Jan. 2010). ISSN: 1744-4292. DOI: 10.1038/msb.2010.66. URL: <http://dx.doi.org/10.1038/msb.2010.66>.

- [36] Dong-Yup Lee u. a. “In silico genome-scale modeling and analysis for identifying anti-tubercular drug targets”. In: *Drug Development Research* 72.2 (Dez. 2010), S. 121–129. ISSN: 1098-2299. DOI: 10.1002/ddr.20408. URL: <http://dx.doi.org/10.1002/ddr.20408>.
- [37] Karthik Raman, Preethi Rajagopalan und Nagasuma Chandra. “Flux Balance Analysis of Mycolic Acid Pathway: Targets for Anti-Tubercular Drugs”. In: *PLoS Computational Biology* 1.5 (Okt. 2005). Hrsg. von Daniel Segre, e46. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.0010046. URL: <http://dx.doi.org/10.1371/journal.pcbi.0010046>.
- [38] Deok-Sun Lee u. a. “Comparative Genome-Scale Metabolic Reconstruction and Flux Balance Analysis of Multiple *Staphylococcus aureus* Genomes Identify Novel Antimicrobial Drug Targets”. In: *Journal of Bacteriology* 191.12 (Juni 2009), S. 4015–4024. ISSN: 1098-5530. DOI: 10.1128/jb.01743-08. URL: <http://dx.doi.org/10.1128/jb.01743-08>.
- [39] M. Lakshmanan u. a. “Software applications for flux balance analysis”. In: *Briefings in Bioinformatics* 15.1 (Nov. 2012), S. 108–122. ISSN: 1477-4054. DOI: 10.1093/bib/bbs069. URL: <http://dx.doi.org/10.1093/bib/bbs069>.
- [40] Kenneth J Kauffman, Purusharth Prakash und Jeremy S Edwards. “Advances in flux balance analysis”. In: *Current Opinion in Biotechnology* 14.5 (Okt. 2003), S. 491–496. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2003.08.001. URL: <http://dx.doi.org/10.1016/j.copbio.2003.08.001>.
- [41] Shalini Singh u. a. “Flux-based classification of reactions reveals a functional bow-tie organization of complex metabolic networks”. In: *Physical Review E* 87.5 (Mai 2013). ISSN: 1550-2376. DOI: 10.1103/physreve.87.052708. URL: <http://dx.doi.org/10.1103/PhysRevE.87.052708>.
- [42] Matthew A. Oberhardt, Arvind K. Chavali und Jason A. Papin. “Flux Balance Analysis: Interrogating Genome-Scale Metabolic Networks”. In: *Systems Biology*. Humana Press, 2009, S. 61–80. ISBN: 9781597455251. DOI: 10.1007/978-1-59745-525-1_3. URL: http://dx.doi.org/10.1007/978-1-59745-525-1_3.
- [43] Ali R. Zomorodi und Costas D. Maranas. “OptCom: A Multi-Level Optimization Framework for the Metabolic Modeling and Analysis of Microbial Communities”. In: *PLoS Computational Biology* 8.2 (Feb. 2012). Hrsg. von Christopher V. Rao, e1002363. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002363. URL: <http://dx.doi.org/10.1371/journal.pcbi.1002363>.

- [44] Michael A. Henson und Timothy J. Hanly. “Dynamic flux balance analysis for synthetic microbial communities”. In: *IET Systems Biology* 8.5 (Okt. 2014), S. 214–229. ISSN: 1751-8857. DOI: 10.1049/iet-syb.2013.0021. URL: <http://dx.doi.org/10.1049/iet-syb.2013.0021>.
- [45] William R. Harcombe u. a. “Metabolic Resource Allocation in Individual Microbes Determines Ecosystem Interactions and Spatial Dynamics”. In: *Cell Reports* 7.4 (Mai 2014), S. 1104–1115. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2014.03.070. URL: <http://dx.doi.org/10.1016/j.celrep.2014.03.070>.
- [46] Benedict Borer und Dani Or. “Spatiotemporal metabolic modeling of bacterial life in complex habitats”. In: *Current Opinion in Biotechnology* 67 (Feb. 2021), S. 65–71. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2021.01.004. URL: <http://dx.doi.org/10.1016/j.copbio.2021.01.004>.
- [47] William T. Scott u. a. “A structured evaluation of genome-scale constraint-based modeling tools for microbial consortia”. In: *PLOS Computational Biology* 19.8 (Aug. 2023). Hrsg. von Tunahan Cakir, e1011363. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1011363. URL: <http://dx.doi.org/10.1371/journal.pcbi.1011363>.
- [48] John H. Holland. “Complex Adaptive Systems”. In: *Daedalus* 121.1 (1992), S. 17–30. URL: <http://www.jstor.org/stable/20025416> (besucht am 02.07.2024).
- [49] Volker Grimm u. a. “Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology”. In: *Science* 310.5750 (Nov. 2005), S. 987–991. ISSN: 1095-9203. DOI: 10.1126/science.1116681. URL: <http://dx.doi.org/10.1126/science.1116681>.
- [50] H. Kaul und Y. Ventikos. “Investigating biocomplexity through the agent-based paradigm”. In: *Briefings in Bioinformatics* 16.1 (Nov. 2013), S. 137–152. ISSN: 1477-4054. DOI: 10.1093/bib/bbt077. URL: <http://dx.doi.org/10.1093/bib/bbt077>.
- [51] Gary An u. a. “Agent-based models in translational systems biology”. In: *WIREs Systems Biology and Medicine* 1.2 (Sep. 2009), S. 159–171. ISSN: 1939-005X. DOI: 10.1002/wsbm.45. URL: <http://dx.doi.org/10.1002/wsbm.45>.
- [52] Tatiana Shashkova u. a. “Agent Based Modeling of Human Gut Microbiome Interactions and Perturbations”. In: *PLOS ONE* 11.2 (Feb. 2016). Hrsg. von

- Brenda A Wilson, e0148386. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0148386. URL: <http://dx.doi.org/10.1371/journal.pone.0148386>.
- [53] J.M. Zheng, K.W. Chan und I. Gibson. “Virtual reality”. In: *IEEE Potentials* 17.2 (1998), S. 20–23. ISSN: 0278-6648. DOI: 10.1109/45.666641. URL: <http://dx.doi.org/10.1109/45.666641>.
- [54] Sergo Martirosov und Pavel Kopecek. “Virtual Reality and its Influence on Training and Education - Literature Review”. In: *Proceedings of the 28th International DAAAM Symposium 2017*. DAAAM International Vienna, 2017, S. 0708–0717. DOI: 10.2507/28th.daaam.proceedings.100. URL: <http://dx.doi.org/10.2507/28th.daaam.proceedings.100>.
- [55] Griffin Kao u. a. “The Rise and Fall of Virtual Reality: What It Takes to Create Technological Disruption”. In: *Turning Silicon into Gold*. Apress, 2020, S. 133–139. ISBN: 9781484256299. DOI: 10.1007/978-1-4842-5629-9_19. URL: http://dx.doi.org/10.1007/978-1-4842-5629-9_19.
- [56] Morton L. Heilig. “Sensorama Simulator”. Utility US3050870A. Filed: January 10, 1961. Aug. 1962. URL: <https://patents.google.com/patent/US3050870A/en>.
- [57] Ivan E. Sutherland. “The ultimate display”. In: *Proceedings of the IFIP Congress*. 1965, S. 506–508.
- [58] Ivan E. Sutherland. “A head-mounted three dimensional display”. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*. AFIPS '68 (Fall, part I). ACM Press, 1968, S. 757–764. DOI: 10.1145/1476589.1476686. URL: <http://dx.doi.org/10.1145/1476589.1476686>.
- [59] Damien Broderick. *The Judas mandala*. en. London, England: Gateway, 1982.
- [60] Stefan Bittmann. “Virtual Reality (VR) in Pediatrics: Innovative Perspectives With Special Reference To Clinical Applications and Pediatric Rehabilitation”. In: *Journal of Regenerative Biology and Medicine* (Juni 2021). ISSN: 2582-385X. DOI: 10.37191/maps-ci-2582-385x-3(3)-070. URL: [http://dx.doi.org/10.37191/Maps-ci-2582-385X-3\(3\)-070](http://dx.doi.org/10.37191/Maps-ci-2582-385X-3(3)-070).
- [61] C. Machover und S.E. Tice. “Virtual reality”. In: *IEEE Computer Graphics and Applications* 14.1 (Jan. 1994), S. 15–16. ISSN: 0272-1716. DOI: 10.1109/38.250913. URL: <http://dx.doi.org/10.1109/38.250913>.

- [62] Brian Spear. “Virtual reality: patent review”. In: *World Patent Information* 24.2 (Juni 2002), S. 103–109. ISSN: 0172-2190. DOI: 10.1016/S0172-2190(02)00002-9. URL: [http://dx.doi.org/10.1016/S0172-2190\(02\)00002-9](http://dx.doi.org/10.1016/S0172-2190(02)00002-9).
- [63] Chuck Blanchard u. a. “Reality built for two: a virtual reality tool”. In: *Proceedings of the 1990 symposium on Interactive 3D graphics - SI3D '90*. SI3D '90. ACM Press, 1990. DOI: 10.1145/91385.91409. URL: <http://dx.doi.org/10.1145/91385.91409>.
- [64] Charlene Jennett u. a. “Measuring and defining the experience of immersion in games”. In: *International Journal of Human-Computer Studies* 66.9 (Sep. 2008), S. 641–661. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2008.04.004. URL: <http://dx.doi.org/10.1016/j.ijhcs.2008.04.004>.
- [65] K. Kilian. “Akustik als klangvolles Element multisensueller Markenkommunikation”. In: *Audio-Branding*, S. 214–227. URL: <https://www.nomos-elibrary.de/10.5771/9783845213927-214.pdf> (besucht am 02.07.2024).
- [66] Matthew Lombard und Theresa Ditton. “At the Heart of It All: The Concept of Presence”. In: *Journal of Computer-Mediated Communication* 3.2 (Sep. 1997). ISSN: 1083-6101. DOI: 10.1111/j.1083-6101.1997.tb00072.x. URL: <http://dx.doi.org/10.1111/j.1083-6101.1997.tb00072.x>.
- [67] Alan M. Turing. “Computing Machinery and Intelligence”. In: *Parsing the Turing Test*. Springer Netherlands, Nov. 2007, S. 23–65. ISBN: 9781402067105. DOI: 10.1007/978-1-4020-6710-5_3. URL: http://dx.doi.org/10.1007/978-1-4020-6710-5_3.
- [68] Carol A. Thornson, Brian F. Goldiez und Huy Le. “Predicting presence: Constructing the Tendency toward Presence Inventory”. In: *International Journal of Human-Computer Studies* 67.1 (Jan. 2009), S. 62–78. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2008.08.006. URL: <http://dx.doi.org/10.1016/j.ijhcs.2008.08.006>.
- [69] Joseph J. LaViola. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32.1 (Jan. 2000), S. 47–56. ISSN: 0736-6906. DOI: 10.1145/333329.333344. URL: <http://dx.doi.org/10.1145/333329.333344>.
- [70] R. S. Kellog, C. Castore und R. Coward. “Psychophysiological Effects of Training in a Full Vision Simulator”. In: *Annual Scientific Meeting of the Aerospace Medical Association*. 1980.

- [71] D. W. Gower und J. E. Fowlkes. *Simulator Sickness in the UH-60 (Black Hawk) Flight Simulator*. USAARL Report 89-20. AD-A214 434. U.S. Army Aeromedical Research Laboratory, Sep. 1989.
- [72] J. T. Reason und J. J. Brand. *Motion Sickness*. London: Academic Press, 1975.
- [73] Chen Zhang. “Investigation on Motion Sickness in Virtual Reality Environment from the Perspective of User Experience”. In: *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*. Bd. 34. IEEE, Sep. 2020, S. 393–396. DOI: 10.1109/iciscae51034.2020.9236907. URL: <http://dx.doi.org/10.1109/ICISCAE51034.2020.9236907>.
- [74] Meta. *Meta Quest 3 — Technische Spezifikationen*. Techn. Ber. Meta, 2024. URL: <https://www.meta.com/de/quest/quest-3/#specs> (besucht am 08.09.2024).
- [75] Meta. *Get Started Developing for the Meta Horizon Platform*. Techn. Ber. Meta, 2024. URL: <https://developers.meta.com/horizon/resources/quest/> (besucht am 21.09.2024).
- [76] Paul Zikas u. a. “A Mobile, AR Inside-Out Positional Tracking Algorithm, (MARIOPOT), Suitable for Modern, Affordable Cardboard-Style VR HMDs”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, S. 257–268. ISBN: 9783319484969. DOI: 10.1007/978-3-319-48496-9_21. URL: http://dx.doi.org/10.1007/978-3-319-48496-9_21.
- [77] Blender. *blender.org — Home of the Blender project — Free and Open 3D Creation Software*. 2024. URL: <https://www.blender.org/> (besucht am 07.08.2024).
- [78] Unity Technologies. *Unity*. 2024. URL: <https://unity.com/de> (besucht am 06.08.2024).
- [79] Unity Technologies. *Graphics*. 2024. URL: <https://docs.unity3d.com/Manual/Graphics.html> (besucht am 06.08.2024).
- [80] Unity Technologies. *Physics*. 2024. URL: <https://docs.unity3d.com/Manual/PhysicsSection.html> (besucht am 06.08.2024).
- [81] Unity Technologies. *Scripting*. 2024. URL: <https://docs.unity3d.com/Manual/ScriptingSection.html> (besucht am 06.08.2024).
- [82] Unity Technologies. *Unity Asset Store*. 2024. URL: <https://assetstore.unity.com/> (besucht am 06.08.2024).

- [83] Unity Technologies. *Scenes*. 2024. URL: <https://docs.unity3d.com/Manual/CreatingScenes.html> (besucht am 06.08.2024).
- [84] Unity Technologies. *GameObject*. 2024. URL: <https://docs.unity3d.com/Manual/class-GameObject.html> (besucht am 06.08.2024).
- [85] Unity Technologies. *Introduction to components*. 2024. URL: <https://docs.unity3d.com/Manual/Components.html> (besucht am 06.08.2024).
- [86] Unity Technologies. *About the XR Plug-in Management package*. 2024. URL: <https://docs.unity3d.com/Packages/com.unity.xr.management@4.4/manual/index.html> (besucht am 07.08.2024).
- [87] Unity Technologies. *XR Interaction Toolkit*. 2024. URL: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html> (besucht am 07.08.2024).
- [88] Unity Technologies. *Canvas*. 2024. URL: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html> (besucht am 07.08.2024).
- [89] Unity Technologies. *Visual Components*. 2024. URL: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UIVisualComponents.html> (besucht am 07.08.2024).
- [90] Unity Technologies. *Interactive Components*. 2024. URL: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UIInteractionComponents.html> (besucht am 07.08.2024).
- [91] Catherine A. Lozupone u. a. “Diversity, stability and resilience of the human gut microbiota”. In: *Nature* 489.7415 (Sep. 2012), S. 220–230. ISSN: 1476-4687. DOI: 10.1038/nature11550. URL: <http://dx.doi.org/10.1038/nature11550>.
- [92] Jack A. Gilbert u. a. “Microbiome-wide association studies link dynamic microbial consortia to disease”. In: *Nature* 535.7610 (Juli 2016), S. 94–103. ISSN: 1476-4687. DOI: 10.1038/nature18850. URL: <http://dx.doi.org/10.1038/nature18850>.
- [93] Salvador Mirete, Verónica Morgante und José Eduardo González-Pastor. “Functional metagenomics of extreme environments”. In: *Current Opinion in Biotechnology* 38 (Apr. 2016), S. 143–149. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2016.01.017. URL: <http://dx.doi.org/10.1016/j.copbio.2016.01.017>.

- [94] Xiaojing Liu und Jason W. Locasale. “Metabolomics: A Primer”. In: *Trends in Biochemical Sciences* 42.4 (Apr. 2017), S. 274–284. ISSN: 0968-0004. DOI: 10.1016/j.tibs.2017.01.004. URL: <http://dx.doi.org/10.1016/j.tibs.2017.01.004>.
- [95] Ines Thiele und Bernhard Ø Palsson. “A protocol for generating a high-quality genome-scale metabolic reconstruction”. In: *Nature Protocols* 5.1 (Jan. 2010), S. 93–121. ISSN: 1750-2799. DOI: 10.1038/nprot.2009.203. URL: <http://dx.doi.org/10.1038/nprot.2009.203>.
- [96] Christian Diener, Sean M. Gibbons und Osbaldo Resendis-Antonio. “MICOM: Metagenome-Scale Modeling To Infer Metabolic Interactions in the Gut Microbiota”. In: *mSystems* 5.1 (Feb. 2020). Hrsg. von Nicholas Chia. ISSN: 2379-5077. DOI: 10.1128/msystems.00606-19. URL: <http://dx.doi.org/10.1128/msystems.00606-19>.
- [97] Ilija Dukovski u. a. “A metabolic modeling platform for the computation of microbial ecosystems in time and space (COMETS)”. In: *Nature Protocols* 16.11 (Okt. 2021), S. 5030–5082. ISSN: 1750-2799. DOI: 10.1038/s41596-021-00593-3. URL: <http://dx.doi.org/10.1038/s41596-021-00593-3>.
- [98] A Varma und B O Palsson. “Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110”. In: *Applied and Environmental Microbiology* 60.10 (Okt. 1994), S. 3724–3731. ISSN: 1098-5336. DOI: 10.1128/aem.60.10.3724-3731.1994. URL: <http://dx.doi.org/10.1128/aem.60.10.3724-3731.1994>.
- [99] Shany Ofaim u. a. “Dynamic Allocation of Carbon Storage and Nutrient-Dependent Exudation in a Revised Genome-Scale Model of Prochlorococcus”. In: *Frontiers in Genetics* 12 (Feb. 2021). ISSN: 1664-8021. DOI: 10.3389/fgene.2021.586293. URL: <http://dx.doi.org/10.3389/fgene.2021.586293>.
- [100] Karen D. Xu u. a. “Spatial Physiological Heterogeneity in Pseudomonas aeruginosa Biofilm Is Determined by Oxygen Availability”. In: *Applied and Environmental Microbiology* 64.10 (Okt. 1998), S. 4035–4039. ISSN: 1098-5336. DOI: 10.1128/aem.64.10.4035-4039.1998. URL: <http://dx.doi.org/10.1128/AEM.64.10.4035-4039.1998>.

- [101] J. H. T. Luong. “Generalization of monod kinetics for analysis of growth data with substrate inhibition”. In: *Biotechnology and Bioengineering* 29.2 (Feb. 1987), S. 242–248. ISSN: 1097-0290. DOI: 10.1002/bit.260290215. URL: <http://dx.doi.org/10.1002/bit.260290215>.
- [102] Francisco Pinto u. a. “Modeling Metabolic Interactions in a Consortium of the Infant Gut Microbiome”. In: *Frontiers in Microbiology* 8 (Dez. 2017). ISSN: 1664-302X. DOI: 10.3389/fmicb.2017.02507. URL: <http://dx.doi.org/10.3389/fmicb.2017.02507>.
- [103] Jason H. Moore u. a. “Human microbiome visualization using 3d technology”. In: *Biocomputing 2011*. World Scientific, Nov. 2010, S. 154–164. ISBN: 9789814335058. DOI: 10.1142/9789814335058_0017. URL: http://dx.doi.org/10.1142/9789814335058_0017.
- [104] Neha Garg u. a. “Three-Dimensional Microbiome and Metabolome Cartography of a Diseased Human Lung”. In: *Cell Host & Microbe* 22.5 (Nov. 2017), 705–716.e4. ISSN: 1931-3128. DOI: 10.1016/j.chom.2017.10.001. URL: <http://dx.doi.org/10.1016/j.chom.2017.10.001>.
- [105] Hanin M. Abdullah und Ahmed M. Zeki. “Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example”. In: *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*. IEEE, Dez. 2014. DOI: 10.1109/acsat.2014.22. URL: <http://dx.doi.org/10.1109/ACSAT.2014.22>.
- [106] A Nurpalah u. a. “Effect of UI/UX Designer on Front End”. In: *International Journal of Research and Applied Technology* 1.2 (Dez. 2021), S. 335–341. ISSN: 2807-7229. DOI: 10.34010/injuratech.v1i2.6759. URL: <http://dx.doi.org/10.34010/injuratech.v1i2.6759>.
- [107] Erlinda Muslim u. a. “User interface redesign of e-commerce platform mobile application (Kudo) through user experience evaluation to increase user attraction”. In: *IOP Conference Series: Materials Science and Engineering* 508 (Mai 2019), S. 012113. ISSN: 1757-899X. DOI: 10.1088/1757-899x/508/1/012113. URL: <http://dx.doi.org/10.1088/1757-899x/508/1/012113>.
- [108] Nigel Bevan. “What is the difference between the purpose of usability and user experience evaluation methods”. In: *Proceedings of the Workshop UXEM 9*. Jan. 2009, S. 1–4.

- [109] Thamsanqa Keith Miya und Irene Govender. “UX/UI design of online learning platforms and their impact on learning: A review”. In: *International Journal of Research in Business and Social Science (2147- 4478)* 11.10 (Dez. 2022), S. 316–327. ISSN: 2147-4478. DOI: 10.20525/ijrbs.v11i10.2236. URL: <http://dx.doi.org/10.20525/ijrbs.v11i10.2236>.
- [110] ISO 9241-110:2020. *Ergonomics of human-system interaction - Part 110: Interaction principles*. Genf, 2020.
- [111] Thomas Mejtoft u. a. “User Experience Design and Digital Nudging in a Decision Making Process”. In: *Humanizing Technology for a Sustainable Society*. University of Maribor Press, 2019, S. 427–442. DOI: 10.18690/978-961-286-280-0.23. URL: <http://dx.doi.org/10.18690/978-961-286-280-0.23>.
- [112] Jakob Nielsen. *Usability 101: Introduction to usability*. Techn. Ber. Nielsen-Norman Group, 2012. URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (besucht am 09.08.2024).
- [113] Donald Norman. *The Design of Everyday Things (Revised and expanded edition)*. New York: Basic Books, 2013.
- [114] Jonathan Vitale u. a. “Be More Transparent and Users Will Like You: A Robot Privacy and User Experience Design Experiment”. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’18. ACM, Feb. 2018. DOI: 10.1145/3171221.3171269. URL: <http://dx.doi.org/10.1145/3171221.3171269>.
- [115] Marvin Heuer u. a. “Rethinking Interaction with Conversational Agents: How to Create a Positive User Experience Utilizing Dialog Patterns”. In: *Design, User Experience, and Usability*. Springer Nature Switzerland, 2023, S. 283–301. ISBN: 9783031357084. DOI: 10.1007/978-3-031-35708-4_22. URL: http://dx.doi.org/10.1007/978-3-031-35708-4_22.
- [116] Page Laubheimer. *Breadcrumbs: 11 Design Guidelines for Desktop and Mobile*. Techn. Ber. Nielsen-Norman Group, 2018. URL: <https://www.nngroup.com/articles/breadcrumbs/> (besucht am 10.08.2024).
- [117] J. Blustein, I. Ahmed und K. Instone. “An Evaluation of Look-ahead Breadcrumbs for the WWW”. In: *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*. New York, NY: ACM, 2005, S. 202–204. URL: <https://dl.acm.org/doi/pdf/10.1145/1083356.1083394> (besucht am 05.09.2024).

- [118] Katie Sherwin. *Low-Contrast Text Is Not the Answer*. Techn. Ber. Nielsen-Norman Group, 2015. URL: <https://www.nngroup.com/articles/low-contrast/> (besucht am 10.08.2024).
- [119] Shawn Lawton Henry. *WCAG 2 Overview*. Techn. Ber. World Wide Web Consortiums (W3C), 2024. URL: <https://www.w3.org/WAI/standards-guidelines/wcag/> (besucht am 10.08.2024).
- [120] Eric Eggert und Shadi Abou-Zahra. *How to Meet WCAG (Quick Reference)*. Techn. Ber. World Wide Web Consortiums (W3C), 2024. URL: <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.1> (besucht am 12.08.2024).
- [121] World Wide Web Consortium. *Contrast (Minimum): Understanding SC 1.4.3*. Techn. Ber. World Wide Web Consortiums (W3C), 2023. URL: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html> (besucht am 12.08.2024).
- [122] ISO 9241-3:2000. *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 3: Visual display requirements*. Amendment 1. Genf, 2000.
- [123] ANSI/HFS 100-1988. *American National Standard for Human Factors Engineering of Visual Display Terminal Workstations*. New York, 1988.
- [124] World Wide Web Consortium. *Web Content Accessibility Guidelines (WCAG) 2 Level AA Conformance*. Techn. Ber. World Wide Web Consortiums (W3C), 2020. URL: <https://www.w3.org/WAI/WCAG2AA-Conformance> (besucht am 12.08.2024).
- [125] Ayesha Amin, Mohammad Shidujaman und Bo Wang. “Improving HCI on Cognition for Children with Intelligent UI/UX”. In: *Culture and Computing*. Springer Nature Switzerland, 2023, S. 355–366. ISBN: 9783031347320. DOI: 10.1007/978-3-031-34732-0_27. URL: http://dx.doi.org/10.1007/978-3-031-34732-0_27.
- [126] Joohwan Park u. a. “1C3-2 A Literature Survey on UX Design Properties and Principles Of Smart Device Design for the Disabled:: From the Perspective of the Value/Quality of Human Life”. In: *The Japanese Journal of Ergonomics* 51.Supplement (2015), S422–S425. ISSN: 1884-2844. DOI: 10.5100/jje.51.s422. URL: <http://dx.doi.org/10.5100/jje.51.S422>.

- [127] Hristo Hristov u. a. “A Model for Designing Accessible Color and Contrast for Users with Visual Deficiency and Color Blindness”. In: *2022 IEEE 11th International Conference on Intelligent Systems (IS)*. IEEE, Okt. 2022. DOI: 10.1109/is57118.2022.10019637. URL: <http://dx.doi.org/10.1109/IS57118.2022.10019637>.
- [128] Jakob Nielsen und Donald Norman. *The Definition of User Experience (UX)*. Techn. Ber. Nielsen-Norman Group, 1998. URL: <https://www.nngroup.com/articles/definition-user-experience/> (besucht am 10.08.2024).
- [129] S. Gopinath u. a. “Usability and accessibility analysis of selected government websites in Sri Lanka”. In: *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, Mai 2016. DOI: 10.1109/tenconspring.2016.7519439. URL: <http://dx.doi.org/10.1109/TENCONSpring.2016.7519439>.
- [130] Jakob Nielsen. *Response Times: The 3 Important Limits*. Techn. Ber. Nielsen-Norman Group, 1993. URL: <https://www.nngroup.com/articles/response-times-3-important-limits/> (besucht am 12.08.2024).
- [131] Anping Cheng u. a. “Exploring the Long-Term User Experience of an Interactive Loading Screen Using UX Curve and QUIS”. In: *International Journal of Human-Computer Interaction* (Mai 2023), S. 1–12. ISSN: 1532-7590. DOI: 10.1080/10447318.2023.2212509. URL: <http://dx.doi.org/10.1080/10447318.2023.2212509>.
- [132] Jakob Nielsen. *10 Usability Heuristics for User Interface Design*. Techn. Ber. Nielsen-Norman Group, 1994. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (besucht am 12.08.2024).
- [133] Rachel Krause. *Maintain Consistency and Adhere to Standards (Usability Heuristic #4)*. Techn. Ber. Nielsen-Norman Group, 2021. URL: <https://www.nngroup.com/articles/consistency-and-standards/> (besucht am 12.08.2024).
- [134] Wenjie Li u. a. “Design Factors to Improve the Consistency and Sustainable User Experience of Responsive Interface Design”. In: *Sustainability* 14.15 (Juli 2022), S. 9131. ISSN: 2071-1050. DOI: 10.3390/su14159131. URL: <http://dx.doi.org/10.3390/su14159131>.
- [135] Francesco Pucillo und Gaetano Cascini. “A framework for user experience, needs and affordances”. In: *Design Studies* 35.2 (März 2014), S. 160–179. ISSN: 0142-694X. DOI: 10.1016/j.destud.2013.10.001. URL: <http://dx.doi.org/10.1016/j.destud.2013.10.001>.

- [136] Jakob Nielsen und Page Laubheimer. *Top 10 Application-Design Mistakes*. Techn. Ber. Nielsen-Norman Group, 2019. URL: <https://www.nngroup.com/articles/top-10-application-design-mistakes/> (besucht am 12.08.2024).
- [137] Kate Moran. *Flat Design: Its Origins, Its Problems, and Why Flat 2.0 Is Better for Users*. Techn. Ber. Nielsen-Norman Group, 2015. URL: <https://www.nngroup.com/articles/flat-design/> (besucht am 12.08.2024).
- [138] Tom Connor. *Don Norman and the design of everyday things*. Techn. Ber. Medium, Aug. 2020. URL: <https://medium.com/10x-curiosity/don-norman-and-the-design-of-everyday-things-d187e6a81fd8> (besucht am 13.08.2024).
- [139] Paul M. Fitts. “The information capacity of the human motor system in controlling the amplitude of movement.” In: *Journal of Experimental Psychology* 47.6 (1954), S. 381–391. ISSN: 0022-1015. DOI: 10.1037/h0055392. URL: <http://dx.doi.org/10.1037/h0055392>.
- [140] I. Scott MacKenzie. “Fitts’ Law as a Research and Design Tool in Human-Computer Interaction”. In: *Human-Computer Interaction* 7.1 (März 1992), S. 91–139. ISSN: 1532-7051. DOI: 10.1207/s15327051hci0701_3. URL: http://dx.doi.org/10.1207/s15327051hci0701_3.
- [141] Michael J. McGuffin und Ravin Balakrishnan. “Fitts’ law and expanding targets: Experimental studies and designs for user interfaces”. In: *ACM Transactions on Computer-Human Interaction* 12.4 (Dez. 2005), S. 388–422. ISSN: 1557-7325. DOI: 10.1145/1121112.1121115. URL: <http://dx.doi.org/10.1145/1121112.1121115>.
- [142] Logan D. Clark, Aakash B. Bhagat und Sara L. Riggs. “Extending Fitts’ law in three-dimensional virtual environments with current low-cost virtual reality technology”. In: *International Journal of Human-Computer Studies* 139 (Juli 2020), S. 102413. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2020.102413. URL: <http://dx.doi.org/10.1016/j.ijhcs.2020.102413>.