



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik

# **Menschenmengen Evakuierung im Bezug auf psychisches und physisches Verhalten unter Verwendung einer Echtzeitphysik aus Videospielen**

## **Masterarbeit**

zur Erlangung des Grades eines Master of Science (M.Sc.)  
im Studiengang Computervisualistik

vorgelegt von  
**Julian Giesen**

Erstgutachter: Prof. Dr. Stefan Müller  
Institut für Computervisualistik, Universität Koblenz, AG Computergrafik

Zweitgutachter: Gerrit Lochmann  
Institut für Computervisualistik, Universität Koblenz, AG Computergrafik

Koblenz, im Januar 2014



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.       

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.       

.....  
(Ort, Datum)

.....  
(Unterschrift)





## Aufgabenstellung für die Masterarbeit

Julian Giesen

(Mat. Nr. 207 200 084)

### **Thema: Menschenmengen Evakuierung im Bezug auf psychisches und physisches Verhalten unter Verwendung einer Echtzeitphysik aus Videospiele**

Evakuierung von Menschenmengen ist ein wichtiges Unterfangen, das vor allem bei der Planung von Massenevents innerhalb eines Gebäudes eine große Rolle spielt. Grafische Simulationen des Personenflusses geben bereits heute Aufschluss über kritische Geländeeigenschaften wie Engpässe, die bei der Evakuierung Gefahren darstellen. Diese Anhaltspunkte können schließlich in die Planung des Events einbezogen werden.

Ansatz dieser Arbeit ist die Verwendung einer Echtzeitphysik und die Entwicklung einer Verhaltens-KI um die Interaktion von Individuen, so genannter Agenten, mit ihrer Umwelt abzubilden. Ziel der Arbeit ist es, anhand der Simulation, Vorhersagen über den Verlauf einer Evakuierung vorzunehmen, die sowohl psychische Verhaltensmuster (wie Stress oder Panik) als auch physische Krafteinwirkung (etwa Sturz oder Erdrücken einer Person) unter chaotischen Zuständen berücksichtigt.

Insbesondere soll die Frage beantwortet werden, in wie weit sich eine Echtzeitphysik, die ursprünglich für die Videospiele-Domäne entwickelt wurde und die Realität nur bis zu einem eingeschränkten Detailgrad abbildet, für derartige Simulationen eignet.

Die inhaltlichen Schwerpunkte der Arbeit sind:

1. Recherche zum Verhalten von Menschen bei einer Evakuierung zur Entwicklung einer Verhaltens-KI
2. Recherche zur Echtzeitphysik aus Videospiele für Menschliche Körper
3. Identifizierung der Anforderungen und Konzipierung eines Prototyps für Menschenmengen Evakuierung mit Echtzeitphysik und Verhaltens-KI
4. Umsetzung eines Prototyps
5. Vergleich des Prototyps mit vorhandenen Evakuierungssimulationen
6. Dokumentation und Bewertung des Ansatzes

Koblenz, 18.07.2013

– Julian Giesen –

– Prof. Dr. Stefan Müller –



## Kurzfassung

Zur Planung von Veranstaltungen werden Menschenmengen-Simulationen eingesetzt, die die Evakuierung der Menschenmenge simulieren, um eventuelle Gefahren einschätzen und vermeiden zu können. Diese Menschenmengen-Simulationen analysieren zum größten Teil nur den Bewegungsfluss einer Menschenmenge. Oft werden physische und psychische Faktoren in diesen Simulationen nicht bedacht und verhindern dadurch, Kollisionen zwischen Agenten abzubilden wie deren Umfallen und das sich verändernde Verhalten auf äußere Einflüsse.

Aus diesem Grund wird in der vorliegenden Arbeit das Ziel verfolgt, das psychische Verhalten und die physischen Krafteinwirkungen der Menschenmenge zu untersuchen. Dazu wurde eine echtzeitfähige Menschenmengen-Simulation entwickelt, die zum einen Verfahren für Videospiele und zum anderen eine Verhaltens-KI für die Agenten der Simulation beinhaltet. Mit Rigid Bodies aus einer Physik-Engine wurde die physikalische Repräsentation der Agenten realisiert, um physische Interaktionen der Agenten untereinander und mit der Umwelt sowie die Fortbewegung der Agenten physikalisch plausibel umzusetzen. Zur Fortbewegung der Agenten wurde zusätzlich das Verfahren von Navigationmeshes und ein geeigneter Algorithmus zur Kollisionsvermeidung realisiert.

Mit der Entwicklung einer Verhaltens-KI wurde ein physisch-psychischer Zustand entworfen, der sich aus einem Stressniveau als psychischer und aus einem Gesundheitswert als physischer Zustand zusammensetzt. Die entwickelte Simulation kann physische Faktoren wie das Gedränge von Agenten oder das Erdrücken eines Agenten sowie die Interaktion der Agenten mit ihrer Umwelt darstellen und psychische Faktoren wie Stress visuell abbilden.

Anhand einer Evaluation von durchgeführten Testszenarien der Simulation wurde untersucht, ob eine Kombination von physischen und psychischen Faktoren in einer Simulation erfolgreich umsetzbar ist und einen Aufschluss über das Verhalten der Agenten in gefährlichen oder stressbehafteten Situationen gibt. Zusätzlich wurde der Nutzen der Komplexität der physikalischen Repräsentation analysiert.

## **Abstract**

Systems to simulate crowd-behavior are used to simulate the evacuation of a crowd in case of an emergency. These systems are limited to the moving-patterns of a crowd and are generally not considering psychological and/or physical conditions. Changing behaviors within the crowd (e.g. by a person falling down) are not considered.

For that reason, this thesis will examine the psychological behavior and the physical impact of a crowd's member on the crowd. In order to do so, this study develops a real-time simulation for a crowd of people, adapted from a system for video games. This system contains a behavior-AI for agents. In order to show physical interaction between the agents and their environment as well as their movements, the physical representation of each agent is realized by using rigid bodies from a physics-engine. The movements of the agents have an additional navigation mesh and an algorithm for collision avoidance.

By developing a behavior-AI a physical and psychological state is reached. This state contains a psychological stress-level as well as a physical condition. The developed simulation is able to show physical impacts such as crowding and crushing of agents, interaction of agents with their environment as well as factors of stress.

By evaluating several tests of the simulation, this thesis examines whether the combination of physical and psychological impacts is implementable successfully. If so, this thesis will be able to give indications of an agent's behavior in dangerous and/or stressful situations as well as a valuation of the complex physical representation.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Zielsetzung . . . . .	3
1.3	Abgrenzung . . . . .	4
1.4	Aufbau der Arbeit . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Verfahren in Videospielen . . . . .	5
2.1.1	Echtzeitphysik . . . . .	5
2.1.2	Künstliche Intelligenz . . . . .	8
2.2	Menschenmengen-Simulation . . . . .	11
2.2.1	Menschenmengen-Verhalten . . . . .	11
2.2.2	Simulationssysteme . . . . .	13
2.2.3	Kollisionsvermeidung . . . . .	15
2.3	Stress in Menschenmengen . . . . .	18
2.4	Zusammenfassung . . . . .	19
<b>3</b>	<b>Verwandte Forschungsansätze</b>	<b>20</b>
3.1	Modellierung von Stress in Menschenmengen-Simulationen . . . . .	20
3.2	Physikalische Interaktion in Multi-Agenten-Simulationen . . . . .	22
3.3	Unterscheidung verwandter Ansätze . . . . .	24
<b>4</b>	<b>Konzept</b>	<b>26</b>
4.1	Ziele und Anforderungen . . . . .	26
4.2	Herangehensweise der Umsetzung verschiedener Prototypen . . . . .	29
4.3	Aufbau und Struktur des ersten Prototypen . . . . .	30
4.4	Aufbau und Struktur des zweiten Prototypen . . . . .	31
4.4.1	Willenskräfte . . . . .	32
4.4.2	Physikalische Kräfte . . . . .	32
4.4.3	Physisch-psychischer Zustand . . . . .	33
4.5	Zusammenfassung . . . . .	34
<b>5</b>	<b>Implementierung</b>	<b>35</b>
5.1	Eingesetzte Software . . . . .	35
5.1.1	Blender . . . . .	35

5.1.2	Unity . . . . .	36
5.2	Umsetzung des ersten Prototypen . . . . .	37
5.3	Umsetzung des zweiten Prototypen . . . . .	40
5.3.1	Navigation . . . . .	42
5.3.2	Kollision . . . . .	53
5.3.3	Physisch-psychischer Zustand . . . . .	54
5.4	Zusammenfassung . . . . .	60
<b>6</b>	<b>Analyse und Diskussion</b>	<b>61</b>
6.1	Testszzenarien . . . . .	61
6.1.1	Szenario 1 - Ausweichen im Tunnel . . . . .	61
6.1.2	Szenario 2 - Gedränge im Tunnel . . . . .	64
6.1.3	Szenario 3 - Erdrücken im Tunnel . . . . .	68
6.1.4	Szenario 4 - Der Loveparade-Tunnel . . . . .	71
6.1.5	Szenario 5 - Der große Platz . . . . .	74
6.1.6	Szenario 6 - Feuer, Gebäudeevakuierung . . . . .	76
6.1.7	Diskussion der Testszzenarien . . . . .	80
6.2	Extrapolation der Prototypen . . . . .	82
6.3	Vergleich der Ergebnisse mit anderen Menschenmengen-Simulationen . . . . .	85
6.4	Zusammenfassung . . . . .	88
<b>7</b>	<b>Fazit und Ausblick</b>	<b>89</b>
7.1	Fazit . . . . .	89
7.2	Ausblick . . . . .	90

# 1 Einleitung

Heutige Standards in Sicherheitsvorkehrungen und Simulationen ermöglichen es, bei einer Evakuierung große Menschenmengen in Gebäuden oder Geländen sicher zu evakuieren. Leider reichen die getroffenen Sicherheitsmaßnahmen nicht in allen Fällen aus, wie zum Beispiel bei der Loveparade 2010 in Duisburg. Bei dieser Tragödie wurden Fehler nachgewiesen, die auf unzureichende Simulation und anschließend unzureichende Sicherheitsvorkehrungen zurückzuführen sind. Das größte Sicherheitsrisiko war, dass das Veranstaltungsgelände nur durch einen Tunnel erreicht und auch wieder verlassen werden konnte (Helbing u. Mukerji, 2012, vgl. S. 4). Nach Helbing (Helbing u. Mukerji, 2012, vgl. S. 4) wurde für die Simulation ein Flow-Modell, welches den Personenfluss der Menschen simuliert, eingesetzt. Die Ergebnisse dieses Modells haben das zuständige Unternehmen dazu veranlasst, den Aufbau des Geländes für die Veranstaltung der Loveparade freizugeben. Durch einen zu hohen Bewegungsfluss der Menschen in Eingangs- und Ausgangsrichtung erstarrte die Bewegung und es war kein Vorwärts- und Zurückgehen mehr möglich. Dieses Problem erhöhte das Stressniveau einiger Menschen, was panisches Verhalten auslöste (Helbing u. Mukerji, 2012, vgl. S. 12) und zu starkem Drängen und Stoßen führte. Trotz der im Vorfeld durchgeführten Sicherheitsvorkehrungen konnte das Unglück nicht verhindert werden. Insgesamt starben 21 Besucher und es gab 541 Verletzte. Um solchen Szenarien entgegenzuwirken, können grafische Simulationen zur Menschenmengen-Evakuierung genutzt werden, die verschiedene Faktoren simulieren. Die Simulation des Personenflusses gibt Aufschlüsse über mögliche Engpässe und andere Gefahren, die bei der Evakuierung eine Rolle spielen. Diese Anhaltspunkte können in die Planung einer Veranstaltung einbezogen werden, um die Sicherheitsvorkehrungen an das Gelände anzupassen. Allerdings lassen sich mit einem solchen Modell nur erschwert dicht beieinander stehende Menschen die ein Gedränge bilden, Zusammenstöße oder das Umfallen von Menschen oder Absperrungen abbilden. Um Szenarien mit solchen Faktoren simulieren zu können, die nicht vordergründig den Bewegungsfluss der Personen simulieren, sondern deren physische und psychische Faktoren in den Mittelpunkt stellen, muss ein neues Simulationsmodell erstellt werden. Dieses Modell, das den Fokus auf physische und psychische Einflüsse setzt, wird in der vorliegenden Arbeit vorgestellt und in unterschiedlichen Testszenarien untersucht.

## 1.1 Motivation

Die Simulation einer Menschenmenge wird in verschiedenen Einsatzfeldern verwendet. Im Bereich der Filmindustrie werden oft Simulationen einer Menge von Menschen oder menschenähnlichen Wesen benötigt, die besonders realistisch aussehen und sich realistisch bewegen und verhalten. Eine Vielzahl von menschenähnlichen Wesen wurde zum Beispiel in dem Film "World War Z" simuliert. Zur Simulation wurde das Simulations- und Visualisierungssystem MASSIVE<sup>1</sup> verwendet, mit dem sich realistische Szenarien mit Menschenmengen abbilden lassen (vgl. Creative Bloq, 2013). Die Anforderungen bei Filmen, eine realistische Simulation zu erzielen, beschränken sich allerdings meist nur auf das, was das Auge sieht. Da die vollständige Kontrolle über die Menschenmenge nötig ist, werden physikalische Eigenschaften und das Verhalten durch Reaktionen auf äußere Einflüsse größtenteils von Hand modelliert. Gerade diese Faktoren, die den Schwerpunkt der vorliegenden Arbeit ausmachen, sind wichtig um eine autonome Menschenmenge zu simulieren.

Die Simulation von Menschenmengen in der Videospieleindustrie greift diese Faktoren auf, um realistische Menschenmengen mit echtzeitfähiger Interaktion darzustellen. Als Beispiel soll die Fußball-Videospielserie FIFA<sup>2</sup> dienen, in der zwei Fußballmannschaften, bestehend aus jeweils 11 Spielern, gegeneinander spielen. Der Benutzer steuert gleichzeitig nur einen Spieler, während alle anderen Spieler durch eine künstliche Intelligenz, kurz KI, gelenkt werden. Electronic Arts entwickelte hierzu seit dem Videospiel FIFA 12 die Pro Player Intelligence, die der KI eine Art eigenes Selbstbewusstsein und die Fähigkeit verleiht, anhand äußerer Einflüsse Entscheidungen zu treffen und individuell zu handeln. Mit der eigens realisierten Physik-Engine, der Player Impact Engine, die ebenfalls seit FIFA 12 zum Einsatz kommt, werden Kollisionen bei jedem Kontakt der Spieler realitätsnah abgebildet. Dazu zählen das Abfangen eines Stoßes, aber ebenso das Umfallen eines Spielers. (vgl. EA SPORTS FOOTBALL WORLD, 2011)

Da in diesem Beispiel erkennbar ist, wie physische Eigenschaften und psychisches Verhalten in Form eines Videospieles dargestellt werden können, ergibt sich die Motivation, mit ähnlichen Verfahren, eine Simulation für Menschenmengen-Evakuierung zu entwickeln. In diesem Zusammenhang sind solche Simulationen wichtig und interessant, damit zum Beispiel Veranstalter bei Planungen von Großveranstaltungen unterstützt

---

<sup>1</sup>MASSIVE: <http://massivesoftware.com/film.html>, letzter Besuch am 10.12.2013

<sup>2</sup>FIFA: <http://www.ea.com/de/fifa>, letzter Besuch am 10.12.2013

werden können. Führende Programme zur Simulation von Veranstaltungen sind beispielsweise Legion<sup>3</sup> oder MassMotion<sup>4</sup>. Legion wird seit 2000 erfolgreich zur Planung der Olympischen Spiele verwendet und führt Simulationen mit individuellen Agenten zweidimensional, aber auch dreidimensional durch. Die Menschenmengen-Simulation gibt Aufschluss über statistische Informationen wie die Wegzeiten, den Bewegungsfluss und das Dichteaufkommen, die zur Veranstaltungsplanung herangezogen werden sollten. Allerdings bildet Legion kein psychisches Verhalten wie Stress oder Emotionen ab. MassMotion ist ein 3D agentenbasiertes Simulationsprogramm und wurde zur Simulation des Bewegungsflusses in der Toronto Union Station eingesetzt. Das Hauptaugenmerk dieses Programms liegt auf der Simulation des Bewegungsflusses und die simulierten Agenten verfügen ebenfalls über kein psychisches Verhalten. Außerdem sind durch beide Programme keine physischen Interaktionen der Agenten untereinander oder mit der Umwelt simulierbar. (vgl. Challenger u. a., 2009)

Das Fehlen von physischen und psychischen Faktoren in Simulationsprogrammen verhindert Kollisionen zwischen Agenten abzubilden wie deren Umfallen und das sich verändernde Verhalten auf äußere Einflüsse. Dadurch lassen sich Szenarien wie die Katastrophe der Loveparade 2010 in Duisburg nicht vollständig abbilden. Durch Verwendung einer Echtzeitphysik und Techniken der Navigation aus Videospiele, kombiniert mit einer Verhaltens-KI basierend auf Stress, soll eine solche Simulation ermöglicht und in der vorliegenden Arbeit prototypisch entwickelt werden. Als Beispiel soll das Szenario der Loveparade 2010 in Duisburg simuliert und ähnliche Phänomene aus dieser Simulation abgelesen werden können. Mit diesen Faktoren in einer Simulation wäre es denkbar, gefährliche Situationen wie Engpässe oder zu wenig Fluchtwege in Gebäuden besser einschätzen zu können und die festgestellten Ergebnisse zur optimierten Planung einer Veranstaltung einzubringen.

## 1.2 Zielsetzung

Das Ziel der vorliegenden Arbeit ist es, mit Hilfe einer agentenbasierten Simulation von Menschenmengen physische Krafteinwirkung und psychische Verhaltensmuster vorherzusagen. Unter Verwendung einer Echtzeitphysik für Videospiele sollen physische Faktoren wie das Gedränge von Menschen oder das Erdrücken einer Person sowie

---

<sup>3</sup>Legion: <http://www.legion.com/>, letzter Besuch am 10.12.2013

<sup>4</sup>MassMotion:

<http://www.oasys-software.com/products/engineering/massmotion.html>,  
letzter Besuch am 10.12.2013

die Interaktion der Agenten mit ihrer Umwelt darstellbar sein. Anhand einer entwickelten Verhaltens-KI, zusammengesetzt aus Navigationselementen und Zuständen, sollen die Fortbewegung und die psychischen Verhaltensmuster der Agenten erkennbar sein. Die Kombination aus Echtzeitphysik und Verhaltens-KI sollen Aufschluss über mögliches Verhalten von Menschen in gefährlichen oder stressbehafteten Situationen geben. Die Realität wird bei der Echtzeitphysik nur bis zu einem bestimmten Detailgrad abgebildet. Deshalb soll der wissenschaftliche Aspekt untersucht werden, inwieweit sich eine Echtzeitphysik der Videospiegelindustrie, in Kombination mit einer Verhaltens-KI, für derartige Simulationen eignet und welcher Mehrwert sich gegenüber anderen Verfahren ergibt.

### **1.3 Abgrenzung**

Die hier erarbeiteten Ergebnisse sind nicht hundertprozentig physikalisch und psychologisch fundiert und sollen als Vorschlag gesehen werden, wie diese Faktoren in einer Simulation von Menschenmengen abgebildet werden können. Für das Verhalten von Menschen in Stresssituationen wurden keine neuen mathematischen Ansätze entwickelt, sondern eine Annäherung des Stresses durch ein Stressniveau erarbeitet. Alle Resultate müssten durch weitere empirische Versuche bestätigt und ergänzt werden, um diese dann zu festigen.

### **1.4 Aufbau der Arbeit**

Die gegenwärtige Arbeit ist wie folgt gegliedert. In Kapitel 2 werden grundlegende Verfahren in Videospielen und elementare Inhalte der Menschenmengen-Simulation sowie ein Verständnis von Stress in Menschenmengen dargelegt. Darauf folgend wird in Kapitel 3 auf verwandte Arbeiten, die einen ähnlichen Ansatz der vorliegenden Arbeit verfolgen, eingegangen und Gleichheiten und Unterschiede erarbeitet. Als Nächstes beschreibt Kapitel 4 das Konzept und die Herangehensweise dieser Arbeit sowie die Ziele und Anforderungen. Kapitel 5 erläutert die Umsetzung des Konzeptes und elaboriert die einzelnen Komponenten der entwickelten Simulation. Die Ergebnisse der Simulation werden anhand verschiedener Testszenarien in Kapitel 6 präsentiert und der entwickelte Ansatz wird bewertet. Abgeschlossen wird die Arbeit mit einem Fazit und einem nachfolgenden Ausblick.

## 2 Grundlagen

Der Ansatz der vorliegenden Arbeit setzt sich aus Verfahren von Videospiele und Menschenmengen-Simulation zusammen. In diesem Kapitel werden die Grundlagen der wichtigsten Verfahren aus beiden Bereichen dargelegt. Die Verfahren in Videospielen unterteilen sich in die einer Echtzeitphysik und die einer künstlichen Intelligenz. Die wichtigsten Elemente einer Echtzeitphysik sind für diese Arbeit Rigid Bodies, Collider und die physikalischen Eigenschaften einer Physik-Engine. Im Abschnitt zur künstlichen Intelligenz wird eine Möglichkeit zur Navigation und Pfadplanung vorgestellt sowie die Entscheidungsfindung für Computer gesteuerte Charaktere. Die Echtzeitphysik beschreibt zentrale Elemente dieser Arbeit, die zur Abbildung von Zusammenstößen und Stürzen der Agenten dienen. Mit einer künstlichen Intelligenz lässt sich ein Verhalten der Agenten in einer Simulation erreichen. Dieses Verhalten wird im Abschnitt zur Menschenmengen-Simulation geschildert. Außerdem werden die unterschiedlichen Simulationssysteme präsentiert. Es wird erläutert, wieso ein mikroskopisches System dem makroskopischen System vorgezogen und in der vorliegenden Arbeit umgesetzt wurde. Zusätzlich werden unterschiedliche Verfahren zur Kollisionsvermeidung vorgestellt, die mitunter in der Arbeit zum Einsatz kommen. Abschließend wird dargestellt, was allgemein unter Stress verstanden wird und wie sich dieser bei Individuen in Menschenmengen verhält.

### 2.1 Verfahren in Videospielen

Zur Erstellung eines Videospiele werden verschiedene Verfahren verwendet. Zum einen wird für aktuellste Spiele häufig eine Physik-Engine verwendet, um physikalisch plausible Bewegungen von Charakteren oder andere physische Interaktionen realitätsnah abzubilden. Desweiteren wird in Spielen, in denen durch den Computer gesteuerte Charaktere benötigt werden, eine künstliche Intelligenz entwickelt. Diese Verfahren der Videospiele werden in der vorliegenden Arbeit zur Simulation von Menschenmengen herangezogen und werden im Folgenden grundlegend erklärt.

#### 2.1.1 Echtzeitphysik

Die Echtzeitphysik nimmt in modernen Videospielen einen zentralen Bereich ein und wird anhand einer Physik-Engine umgesetzt. Diese Physik-Engines existieren als fertige Software Development Kits (SDK) und können in individuellen Projekten verwen-

det werden. In der vorliegenden Arbeit wurden zwei Physik-Engine SDKs, Bullet<sup>5</sup> und PhysX<sup>6</sup>, verwendet (s. Kapitel 5.1.1 u. 5.1.2). Beide Engines legen geringfügig differenzierte Schwerpunkte in ihren SDKs, auf die hier nicht näher eingegangen wird. Viel mehr werden die Hauptbestandteile einer Physik-Engine beleuchtet. Physik-Engines bilden die Physik realitätsnah und möglichst effizient ab. Sie sind als eine Blackbox zu verstehen, in der Einstellungen von Werten vorgenommen werden können, deren Wirkung und Berechnung oberflächlich nicht sichtbar sind. Da die Berechnungen in Echtzeit durchgeführt werden, steht die stabile Performanz unmerkbar über der Abbildung der Realität (vgl. Sullivan, 2009, S. 12). Berechnet werden verschiedene physikalische Größen wie zum Beispiel die Masse, die Kraft, das Trägheitsmoment oder das Drehmoment und werden pro physikalischer Entität bestimmt. Rigid Bodies, Stoff oder Flüssigkeit gehören zu diesen Entitäten. Zur Simulation und Kollisionserkennung werden die Entitäten als Objekte in Form von Primitiven repräsentiert. (vgl. Seugling u. Rolin, 2006, S. 8f)

Im Folgenden werden nicht alle, sondern nur die in dieser Arbeit verwendeten Komponenten der Physik-Engines näher beschrieben. Dazu gehören zum einen physikalische Eigenschaften wie Masse, Kraft, Trägheitsmoment und Drehmoment und zum anderen Rigid Bodies und deren Repräsentation.

## **Collider**

Objekte werden in einer Physik-Engine für einfache Kollisionsberechnungen durch Primitive, wie einer Box, einer Kugel oder einer Kapsel dargestellt. Diese Objekte dienen als Hülle für kompliziert aufgebaute Meshes, die zum Beispiel einen Menschen darstellen. Die Hülle wird auch als Collider oder Bounding Box bezeichnet und gibt Objekten die Eigenschaft miteinander zu kollidieren. Sind Kollisionen mit konkaven Objekten erforderlich, können Objekte auch durch ein Triangle Mesh repräsentiert werden, das sich möglichst genau der Form des zugrundeliegenden Meshes anpasst. Aus Gründen der Performanz können Triangle Meshes nur für fest positionierte Objekte verwendet werden. (vgl. Seugling u. Rolin, 2006, S. 9)

Ein Collider kann auch als Trigger eingesetzt werden. Ein Trigger funktioniert ähnlich wie ein Collider, allerdings werden Kollisionen ignoriert. Collider können sich in und

---

<sup>5</sup>Bullet: <http://bulletphysics.org/wordpress/>, letzter Besuch 11.12.13

<sup>6</sup>PhysX: <https://developer.nvidia.com/physx>, letzter Besuch 11.12.13

durch einen Trigger bewegen. Währenddessen erkennt ein Trigger die Collider innerhalb des Triggers und kann daraufhin eine Aktion durchführen. So kann zum Beispiel in einem Videospiel durch einen Trigger eine Tür geöffnet werden, indem der gesteuerte Charakter durch den Trigger auf dem Boden vor der Tür bewegt wird.

### **Physikalische Eigenschaften**

Physikalische Entitäten werden durch physikalische Eigenschaften bestimmt. Die Masse gibt an, wie schwer ein Körper ist. Die Gewichtseinheit kann individuell bestimmt werden und wird in der vorliegenden Arbeit in Kilogramm festgelegt. Zusammen mit der Beschleunigung, welche die Geschwindigkeit über die Zeit ist, ergibt sich die Kraft, gemessen in Newton. Die benötigte Kraft, um einen Kilogramm genau in einer Sekunde einen Meter zu beschleunigen, entspricht einem Newton. Aus dieser Berechnung lassen sich Kräfte beschreiben, die einen Körper bewegen. Eine dieser Kräfte ist die Gravitation, die einem Körper mit etwa 9.80665 Newton zugefügt wird. Alle berechneten Kräfte werden zur Bewegung des Körpers addiert. Kräfte, Position, Geschwindigkeit und Beschleunigung eines Körpers werden als dreidimensionaler Vektor betrachtet. Anhand der zugefügten Kraft lassen sich eine neue Beschleunigung, die daraus resultierende Geschwindigkeit und eine neue Position für den Körper bestimmen.

Die Rotation eines Körpers wird durch sein Trägheitsmoment beschrieben und definiert einen Widerstand zur möglichen Rotationsbewegung. Das Trägheitsmoment ergibt sich aus der Form des Körpers, dessen Dichte und Masse. Es ist mit dem Massezentrum eines Körpers zu vergleichen und gibt an, wie ein Körper rotiert. Die Kraft zur Rotation ist das Drehmoment, eine angulare Kraft, die durch einen Spinradius zugefügt wird. Diese beschreibt die angulare Geschwindigkeit, die die Achse und Geschwindigkeit der Rotation als dreidimensionalen Vektor beschreibt.

Für eine realistische physikalische Simulation müssen die weiteren Eigenschaften, statische und dynamische Reibung sowie Restitution mit in Betracht gezogen werden. Restitution beschreibt die Abprallstärke eines Körpers, wenn dieser in einen Zusammenstoß verwickelt ist. Bei einer Kollision bestimmen statische und dynamische Reibung, wie viel Kraft währenddessen verloren geht.

(vgl. McShaffry u. Graham, 2009, S. 527ff, 534)

## **Rigid Bodies**

Rigid Bodies sind starre Körper, die durch die oben genannten physikalischen Eigenschaften beschrieben werden. Ein Rigid Body kann sich nicht verformen, die Massenverteilung ist also an jedem Punkt des Körpers gleich. Die sechs Freiheitsgrade eines Rigid Bodies, drei für die Translation und drei für die Rotation, ermöglichen eine freie Bewegung unter physikalischen Eigenschaften im Raum. Durch Begrenzungen einzelner Freiheitsgrade wird die Bewegung eingeschränkt. Dies kann auch dadurch geschehen, dass zwei Rigid Bodies mit einem Gelenk, auch Joint genannt, verbunden werden. Man unterscheidet zwischen definierten Gelenken wie Scharniere oder Gelenkköpfe und frei justierbaren Gelenken, die beliebige Achsen in der Bewegung beschränken. Rigid Bodies werden in Verbindung mit einem Collider als physikalische Repräsentation in einer Physik-Engine genutzt und versuchen Objekte in einer Welt möglichst realistisch zu beschreiben.

### **2.1.2 Künstliche Intelligenz**

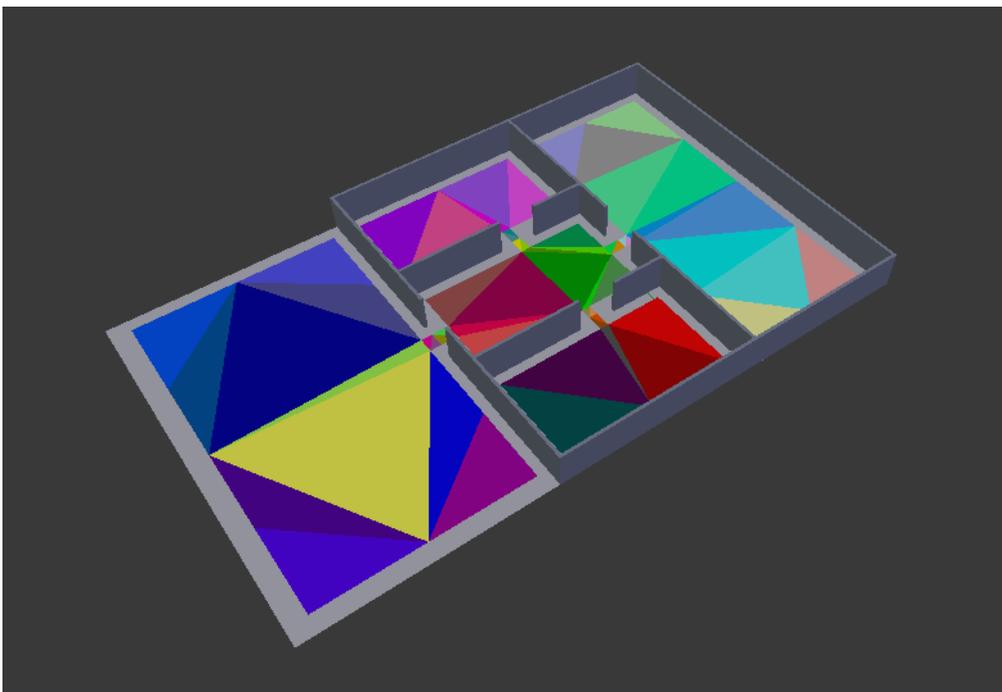
Mit künstlicher Intelligenz wird im Allgemeinen versucht, menschliches oder menschenähnliches Verhalten abzubilden. Der Aufbau einer KI unterscheidet sich leicht in verschiedenen Bereichen. So wird KI neben der Verwendung in Videospielen zum Beispiel auch in der Robotik oder in wissensbasierten Systemen eingesetzt. Im Folgenden werden die grundlegende Elemente, die eine KI in Videospielen ausmachen, näher erläutert. Dazu gehören insbesondere die Steuerung von Agenten sowie deren Verhalten basierend auf Entscheidungen.

### **Pfadfindung und Navigation**

Einer der wichtigsten Punkte der KI von Agenten in Videospielen ist deren präzise Navigation durch eine vorgegebene Welt. Zur Steuerung von Agenten gibt es verschiedene Möglichkeiten. Die allgemeine Fortbewegung in Videospielen wird in der Regel durch Reynolds (2004) Steering Behaviors umgesetzt. Reynolds beschreibt verschiedene Verhalten wie Suchen, Fliehen oder Pfadverfolgung. Suchen lässt einen Agenten ein Ziel ansteuern, indem die Differenz zwischen aktueller Geschwindigkeit und gewünschter Geschwindigkeit berechnet und als Steuerungskraft auf den Agenten angewendet wird. Im Gegensatz hierzu wird bei Fliehen entgegen dem Ziel gesteuert. Bei der Pfadverfolgung versucht ein Agent einen gegebenen Pfad bestehend aus Start- und Endpunkt sowie weiteren Wegpunkten zu traversieren. Die Ziele des Agenten sind die

einzelnen Wegpunkte. Durch Suchen des nächsten Ziels des zu laufenden Pfades erreicht der Agent den Endpunkt. Reynolds Steering Behaviors umfassen noch weitere Verhalten zur Steuerung einzelner Agenten, aber auch zur Steuerung von Gruppen, die weiterführend in Reynolds (2004) dargelegt sind.

Damit ein Agent einen Pfad traversieren kann, muss er zu Anfang einen Weg planen. Zunächst ist für den Agenten das Wissen über begehbare Bereiche notwendig. Hierzu lässt sich die Welt durch ein Navigation Mesh beschreiben. Ein Navigationmesh ist nach Kallmann (2010) als Triangulierung festgelegt (s. Abbildung 2.1). Die einzelnen Dreiecke grenzen Bereiche ab, die ein Agent begehen kann. Dabei werden feste Objekte wie Wände in einer Welt bei der Konstruktion des Navigationmeshes ausgeschlossen und für die Agenten als Hindernisse wahrgenommen. Somit können Wege nur durch die mit dem Navigationmesh abgesteckten Bereiche geplant werden.



**Abbildung 2.1:** Navigationmesh (farblich) als Triangulierung zur Absteckung eines Raumes

Die Planung eines Pfades durch das Navigationmesh ist durch die Verwendung des A\*-Algorithmus möglich. Demyen u. Buro (2006) beschreiben ein Verfahren, mit dem ein optimaler Graph aus einer Triangulierung erzeugt werden kann. Dabei sind die Knoten

des Graphen im Mittelpunkt eines jeden Dreiecks und beschreiben die einzelnen Wegpunkte, die miteinander verbunden sind. Dieser Graph wird benötigt, um mit Hilfe des A\*-Algorithmus den kürzesten Pfad zu berechnen. Anhand eines gegebenen Start- und Endpunktes sucht der Algorithmus schrittweise über die bestehenden Verbindungen des Graphen den kürzesten Pfad zum Ziel. Der kürzeste Weg wird mit einem Schätzwert bestimmt, welcher die Distanz zwischen einem Wegpunkt und dem Endpunkt heranzieht. (vgl. Nareyek, 2004, S. 61), (vgl. McShaffry u. Graham, 2009, S. 627ff, 633ff)

Eine Kombination aus Steering Behaviors und der Verwendung eines Navigation Meshes mit Pfadfindung ermöglicht eine nahezu optimale Steuerung eines Agenten. Damit Agenten nicht untereinander kollidieren, wird zusätzlich ein Verfahren zur dynamischen Kollisionsvermeidung benötigt, das in Kapitel 2.2.3 erläutert wird.

### **Entscheidungsfindung**

Die Entscheidungsfindung beschreibt das Verhalten eines Agenten. Endliche Automaten, die verschiedene Zustände beschreiben, die nicht gleichzeitig, sondern nacheinander erfolgen, werden in Videospielen als einfache if-else-Anweisungen implementiert. Somit führen bestimmte Bedingungen zu entsprechenden Zuständen. (vgl. Nareyek, 2004, S. 61ff), (vgl. McShaffry u. Graham, 2009, S. 642ff)

Ein einfaches Beispiel ist in Abbildung 2.2 zu sehen. Mehrere Agenten bewegen sich in einem Raum und suchen unterschiedliche Ziele. Sobald ein Agent im Sichtfeld eines anderen Agenten ist, wechselt der Agent in den Zustand des Ausweichens. Hat der Agent freie Sicht, sucht er weiter nach seinem Ziel. Sollte ein Feuer im Sichtfeld zu erkennen sein, sucht der Agent ein vor dem Feuer sicheres Ziel und bleibt an dieser Position. An diesem Beispiel ist zu sehen, dass verschiedene Bedingungen unterschiedliche Zustände auslösen können und eine einfache künstliche Intelligenz implementiert werden kann. Das Verfahren der endlichen Automaten lässt sich ebenfalls leicht auf Agenten einer Menschenmenge abbilden, um ein bestimmtes Verhalten zu simulieren. Neben den endlichen Automaten gibt es auch Verfahren wie Fuzzy Logic, mit der eine Kombination von Zuständen möglich ist, sodass mehrere Zustände gleichzeitig zu einem gewissen Prozentsatz je nach Bedingung aktiv sind (vgl. McShaffry u. Graham, 2009, S. 651).

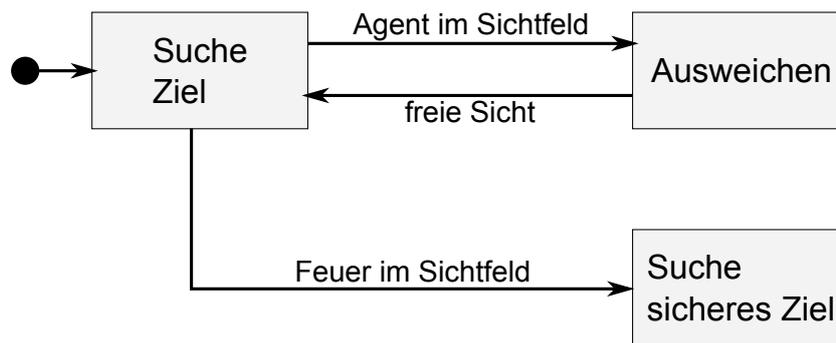


Abbildung 2.2: Beispiel eines einfach verständlichen endlichen Automaten

## 2.2 Menschenmengen-Simulation

Das Thema der Simulation von Menschenmengen bietet viele Gebiete, in denen geforscht wird und die im Nachfolgenden vorgestellt werden. Zum einen gibt es verschiedene Simulationssysteme, die sich in makroskopische und mikroskopische Simulationen unterteilen lassen und den grundlegenden Aufbau einer Simulation beschreiben. Desweiteren werden Verfahren der Kollisionsvermeidung für mikroskopische Simulationssysteme vorgestellt. Zuerst wird jedoch ein generelles Verständnis zu dem Verhalten von Menschenmengen, besonders in Evakuierungssituationen aufgezeigt.

### 2.2.1 Menschenmengen-Verhalten

Das Verhalten von Menschenmengen setzt sich nach Pan u. a. (2005) in der Projektarbeit von Winter (2012) zusammen aus dem Verhalten eines Individuums, der Interaktion zwischen Individuen und dem Verhalten in Gruppen.

#### Verhalten eines Individuums

Das Verhalten der Menschenmenge wird durch das individuelle Verhalten von Personen in unterschiedlichen Situationen bestimmt. Besonders in Situationen, die Stress hervorrufen und zu Panik führen können, wie zum Beispiel bei einer Evakuierung oder in zu engen Gängen in Gebäuden, versucht der Mensch sich instinktiv aus diesen Situationen zu befreien. Diese natürliche Reaktion eines Menschen, die das bewusste Wahrnehmen der Situation einschränkt, führt dazu, dass sich die Individuen in einer Menschenmenge schneller und unkoordinierter fortbewegen und sich gleichzeitig gegenseitig blockieren. Damit wird die Bewegung der gesamten Menschenmenge langsamer und unkontrollierter. Es gibt aber auch Individuen, die sich durch das unkoordinierte

Verhalten der Menschenmenge nicht beeinflussen lassen und den Überblick über die Situation bewahren. Sie können durch ihr rationales Verhalten jedoch nicht das psychisch belastete Verhalten der Menschenmenge ändern, was letztendlich auch bei ihnen zu Stressreaktionen führt. (vgl. Winter, 2012, S. 3f)

### **Interaktion zwischen Individuen**

Wie oben beschrieben, folgen Menschen in stressbehafteten Situationen ihrem Instinkt, wodurch auch ihr soziales Verhalten beeinträchtigt wird. Durch diese Beeinträchtigung kommt es besonders in mit Menschen überfüllten Arealen zu physischen Interaktionen untereinander. Besonders gefährlich kann das dadurch entstehende Umfallen von einzelnen Menschen sein, die zu Tode getrampelt werden können und anderen Menschen das Durchkommen erschweren oder sogar verhindern. Dieses Verhalten kann dazu führen, dass die ganze Menschenmenge ihr Ziel schleppender oder gar nicht erreicht. Individuen, die wie oben beschrieben, eine stressbehaftete Situation besser einschätzen können, helfen unsicheren und gestressteren Menschen, sich an ihnen zu orientieren. Das kann dazu führen, dass eine Menschenmenge ihr Ziel sicherer und schneller erreicht, es kann aber auch zu noch mehr Gedränge und damit zu höherem physischen Kontakt kommen. (vgl. Winter, 2012, S. 4)

### **Verhalten als Gruppe**

Das Verhalten einer Menschenmenge als Gruppe wird maßgeblich durch die einzelnen Individuen und durch die Dichte und Größe der Gruppe als interne Faktoren bestimmt. Diese internen Faktoren stehen in einem Abhängigkeitsverhältnis zueinander. Je größer oder dichter die Menschenmenge ist, desto enger stehen die einzelnen Individuen nebeneinander. Durch eine hohe Dichte der Menschenmenge wird ein einheitlicher Bewegungsfluss erzeugt und Individuen können diesem nicht entgegenwirken, was zu Stößen und Verletzungen führen kann. Neben den internen Faktoren wird die Bewegung der Menschenmenge durch externe Faktoren beeinflusst und verlangsamt. Enge Passagen und Ausgänge sowie versperrte Fluchtwege dienen hier als Beispiele. Das Stagnieren der Bewegung kann das Stressniveau einzelner Individuen deutlich erhöhen, was wiederum das Verhalten und die Bewegung der gesamten Menschenmenge beeinträchtigt. (vgl. Winter, 2012, S. 4)

## 2.2.2 Simulationssysteme

Zur Simulation von Menschenmengen wurden eine Vielzahl verschiedener Simulationssysteme entwickelt. Eine grobe Einteilung kann nach Sullivan (2009) in makroskopische Systeme und mikroskopische Systeme vorgenommen werden.

### Makroskopische Systeme

Makroskopische Systeme modellieren eine Menschenmenge als Ganzes, ohne sich dabei auf einzelne Agenten zu konzentrieren. Mit diesen Systemen lassen sich Phänomene des Bewegungsflusses von Menschen beschreiben. Die Menschenmenge wird vergleichbar mit einer Flüssigkeit oder einem Gas abgebildet. Verschiedene Ansätze wurden zu diesen Systemen entwickelt. Nach Sullivan (2009) entwickelte Helbing (1998) eine der ersten gasförmigen auf Flüssigkeitssimulationen basierende Simulation, die das Verhalten von Fußgängern abbildet. Dazu veränderte Helbing die zugrundeliegende Simulation so, dass Charakteristika von Agenten aufzeigbar waren. Ihm gelang dadurch eine einfache Simulation von dynamischen Menschenmengenbewegungen. Einen weiteren Ansatz entwickelte Hughes (2002), beschrieben von Sullivan (2009), indem er Agenten als ein stetiges Dichtefeld ansieht, das durch eine Potenzialfunktion zu einem Ziel gesteuert wird und das Verhalten von großen Gruppen beschreibt.

Der Vorteil der Simulation von Menschenmengen durch makroskopische Systeme ist die große Anzahl an Agenten, die sich durch ein Gas, bestehend aus einer Vielzahl von Partikeln, abbilden lässt. Mit dem Bewegungsfluss der gesamten Menschenmenge ist auch das Ausweichen von statischen und dynamischen Hindernissen möglich. Der Nachteil, der sich hieraus ergibt, ist das Fehlen der Steuerung von einzelnen Agenten. Agenten werden nicht als Individuen betrachtet, wodurch eine Beschreibung wie in Kapitel 2.2.1 unmöglich ist. Die Betrachtung von Agenten als Individuen ist für diese Arbeit aber essenziell, da nicht das Verhalten der ganzen Menschenmenge, sondern das Verhalten pro Agent simuliert wird, um als Individuum das Verhalten der Menschenmenge zu bestimmen.

### Mikroskopische Systeme

Eigenständige Individuen, die zusammen eine Menschenmenge bilden, lassen sich in mikroskopischen Systemen modellieren. Die Eigenständigkeit eines Individuums ist beschrieben durch sein Verhalten und seinen Entscheidungen bezogen auf seine wahr-

genommene Welt. Die Welt ist dabei so abstrakt gehalten, dass ein Agent sie leicht interpretieren kann. Einen Ansatz zu mikroskopischen Systemen präsentieren Lamarche u. Donikian (2004), beschrieben nach Sullivan (2009), die die geometrische Welt der Agenten in einer topologischen Struktur beschreiben. Dabei kommunizieren die Agenten mit der Welt und anderen Agenten nur in nächster Nachbarschaft und planen ihren Weg, während sie Kollisionen vermeiden. Mit HiDAC (High Density Autonomous Crowds) entwickelten Pelechano u. a. (2007), beschrieben nach Sullivan (2009), eine Simulation, mit der große, dichte Menschenmengen, bestehend aus selbstgesteuerten Agenten, in Bürogebäuden abbildbar sind. Durch zwei verschiedene Module trennt Pelechano u. a. (2007) Navigation, Entscheidungsfindung und Kommunikation der Agenten in das High Level Modul sowie Wahrnehmung und Bewegung der Agenten in das Low Level Modul. Anhand dieser Aufteilung gelingt es ihnen komplexes soziales Verhalten von Menschenmengen zu simulieren wie einen natürlichen entgegengesetzten Bewegungsfluss zweier Gruppen, stoßende und fallende Agenten, die als Hindernisse dienen sowie das Abbilden von Panik. Dazu wird das von Helbing u. Molnar (1995) entwickelte Social Force Model, das es ermöglicht, die Bewegung von Agenten durch soziale Kräfte zu beschreiben, durch psychologische und geometrische Regeln sowie physikalische Kräfte erweitert. Im Gegensatz zu der vorliegenden Arbeit verwenden Pelechano u. a. (2007) keine Physik-Engine.

Der Vorteil, der sich aus der Verwendung eines mikroskopischen Systems ergibt, ist die Repräsentation der Agenten als Individuen. Dadurch, dass Agenten Entscheidungen auf lokaler Ebene treffen, können sie Wege durch komplex aufgebaute Welten planen und daran entlang navigieren. Außerdem verhält sich ein Agent individuell. Nachteilig ist zu erwähnen, dass mikroskopische Systeme sehr komplex und damit sehr rechenintensiv sind. Eine Simulation von großen, dichten Menschenmengen gestaltet sich dadurch aufwendiger und schwieriger als bei makroskopischen Systemen. (vgl. Sullivan, 2009, S. 5f)

Die agentenbasierte Simulation gehört zu den mikroskopischen Systemen. Eine agentenbasierte Simulation lässt sich nach Macal u. North (2011) strukturieren in Agenten mit Attributen und Verhalten, der Interaktion zwischen Agenten und der Umgebung des Agenten. Die Attribute und das Verhalten eines Agenten sollten Selbständigkeit, Modularität, Sozialität und Bedingtheit umfassen. Selbständigkeit bedeutet, dass ein Agent autonom und selbstgerichtet ist und individuell handelt. Modularität beschreibt die Anzahl der Attribute eines Agenten, dessen individuelles Verhalten und die Mög-

lichkeit der Entscheidungsfindung. Die Sozialität umfasst die Interaktion eines Agenten mit anderen Agenten wie das Erkennen von Agenten oder die Kollisionsvermeidung. Modularität bedeutet, dass ein Agent zu jeder gegebenen Zeit einen Zustand hat, der sein Verhalten bestimmt. Ebenfalls interagieren Agenten nur mit einer geringen Anzahl anderer Agenten zu einer gegebenen Zeit in einer für sie limitierten Umgebung. (vgl. Macal u. North, 2011, S. 1458ff)

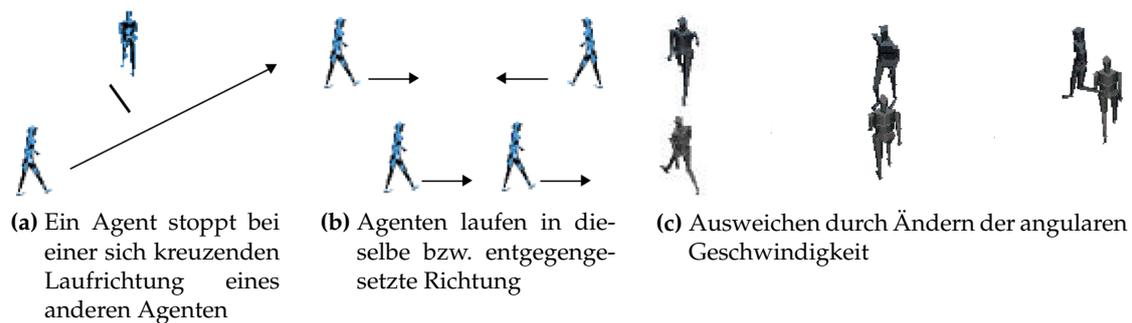
In dieser Arbeit wurde eine agentenbasierte Simulation gewählt, da so ein individuelles psychisches und physisches Verhalten pro Agent realisiert werden kann, um Phänomene wie Stress, Verletzungen oder Umfallen abzubilden.

### **2.2.3 Kollisionsvermeidung**

Ein weiterer wichtiger Teil in der Simulation von Menschenmengen hängt von der Kollisionsvermeidung ab. Um eine realistische Simulation zu erzielen, müssen die Agenten in der Lage sein, sich gegenseitig auszuweichen und die Kollision mit anderen Hindernissen zu vermeiden. Für das grundlegende Verständnis der Kollisionsvermeidung werden im Folgenden zwei unterschiedliche Ansätze, die zum einen in Mustapha u. Bade (2009) zusammengefasst sind, sowie ein Ansatz aus Sullivan (2009) erläutert.

Der erste Ansatz wurde von Musse u. Thalmann (1997) vorgestellt. Diese beschreiben eine einfache, aber erfolgreiche Kollisionsvermeidung auf zwei Arten. Bei der ersten Methode wird der Schnittpunkt zweier Geraden, also die Laufrichtungen der Agenten und die Distanz zwischen den Positionen der Agenten zur Berechnung der neuen Position des Agenten genutzt (s. Abbildung 2.3a). Anhand von festgelegten Prioritätsregeln stoppt ein Agent und lässt den anderen passieren. Sollte es keinen Schnittpunkt geben, wenn die Laufrichtung in dieselbe oder entgegengesetzte Richtung zeigt, wird eine zweite Methode zum Ausweichen verwendet (s. Abbildung 2.3b). Dazu wird die voraussichtliche Kollision von einem Agenten erkannt und dessen Richtung durch seine angulare Geschwindigkeit geändert, wie in Abbildung 2.3c veranschaulicht. Das ist möglich, da der Agent über das Wissen der Positionen der benachbarten Agenten verfügt.

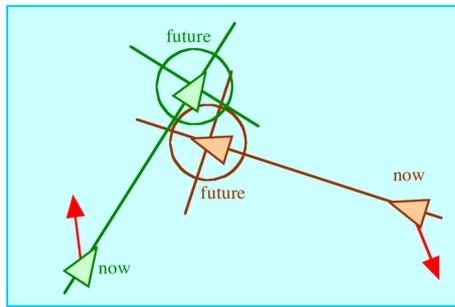
Reynolds (1999, 2004) präsentierte neben den Standard Steering Behaviors (s. Kapitel 2.1.2) eine unausgerichtete Kollisionsvermeidung und das Verhalten von Gruppen, bei denen Agenten untereinander nicht zusammenstoßen. Bei der unausgerichteten Kol-



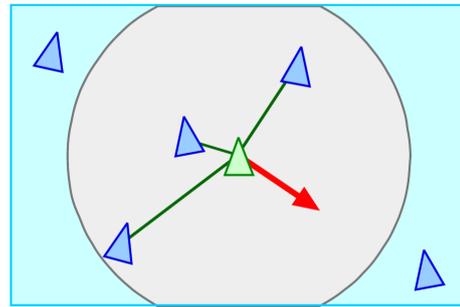
**Abbildung 2.3:** Kollisionsvermeidung beschrieben nach Musse u. Thalmann (1997). Bilderquelle: Musse u. Thalmann (1997)

lisionsvermeidung wird eine potenzielle Kollision zwischen zwei Agenten mit Hilfe der Geschwindigkeit in der Zukunft bestimmt. Sollte eine Kollision vorausgesagt werden, versuchen die Agenten dieser zu entgehen, indem sie langsamer oder schneller werden und ihre Richtung ändern. Der rechte Agent (Orange) in Abbildung 2.4a würde langsamer werden und nach links ausweichen und der linke Agent (Grün) würde schneller werden und ebenfalls nach links ausweichen. Das Verhalten der Gruppe zum Ausweichen unterteilt sich in Separation, Alignment und Kohäsion und ist auch als Flocking bekannt (s. Abbildungen 2.4b,2.4c,2.4d). Bei der Separation wird versucht, sich von zu nahen Agenten wegzubewegen. Alignment führt dazu, dass Agenten in die durchschnittliche Richtung der benachbarten Agenten steuern. Das Bewegen zur durchschnittlichen Position der benachbarten Agenten ist die Kohäsion. Durch diese Verhaltensweisen ist gewährleistet, dass sich ein Agent kollisionsfrei in einer Gruppe bewegt. Dieses Verhalten ist vergleichbar mit dem Verhalten eines Vogel- oder Fischschwarms. (vgl. Reynolds, 1999, 2004)

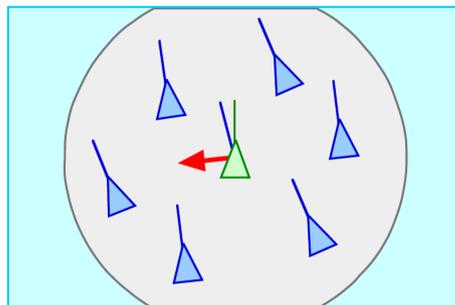
Der in Sullivan (2009) vorgestellte Ansatz zur Kollisionsvermeidung ist eine Zusammensetzung aus verschiedenen Verfahren von Pelechano u. a. (2007) und Lamarche u. Donikian (2004). Agenten haben eine nach Pelechanos HiDAC System festgelegte Reichweite der Wahrnehmung. Dazu wird ein Rechteck in die Richtung, in die sich ein Agent bewegt, projiziert und Agenten, die innerhalb dieses Rechtecks gelangen zur Kollisionsvermeidung herangezogen (s. Abbildung 2.5). Befinden sich Agenten im Bereich des Rechtecks, werden Kräfte zum Ausweichen berechnet. Mit unterschiedlichen Gewichten werden verschiedene Kräfte summiert, um die Richtung der Bewegung zu bestimmen. Zur Bestimmung der neuen Geschwindigkeit eines Agenten werden die gewünschte Richtung, die vorherige Richtung sowie das Ausweichen von Agenten und



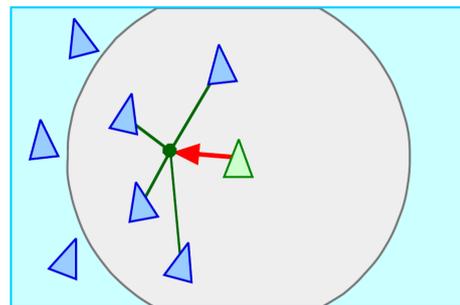
(a) Ungerichtete Kollisionsvermeidung zwischen zwei Agenten



(b) Gruppenverhalten: Separation



(c) Gruppenverhalten: Alignment



(d) Gruppenverhalten: Kohäsion

**Abbildung 2.4:** Kollisionsvermeidung beschrieben nach Reynolds (1999, 2004). Bilderquelle: Reynolds (1999)

Hindernissen berechnet, addiert und normalisiert. Zum Ausweichen wird eine tangentielle Kraft nach der Berechnung von Lamarche u. Donikian (2004) verwendet. Die normalisierte tangentielle Kraft ergibt sich aus der normalisierten Distanz zwischen Agent  $i$  und  $j$ , addiert mit der normalisierten Geschwindigkeit von Agent  $i$  aus Abbildung 2.5. Die Normalisierungen sind notwendig, um die Richtung des resultierenden Vektors nicht zu verändern. Die erhaltene Kraft wird anschließend mit drei Gewichten multipliziert, die sich zum einen auf die Nähe des Agenten beziehen und zum anderen auf die Orientierung der Geschwindigkeiten beider Agenten sowie deren Masse. Je näher ein Agent ist, desto größer ist das erste Gewicht. Die Orientierung zweier Agenten zueinander würde das zweite Gewicht erhöhen. Die beschriebene Berechnung wird für jeden Agenten durchgeführt und führt zu einem erfolgreichen Ausweichen.

In der vorliegenden Arbeit wurde ebenfalls ein Ansatz zur Kollisionsvermeidung erarbeitet und umgesetzt. Dieser ist eine Kombination der Arbeiten von Sullivan (2009) und Musse u. Thalmann (1997) und wird in Kapitel 5 näher erläutert.

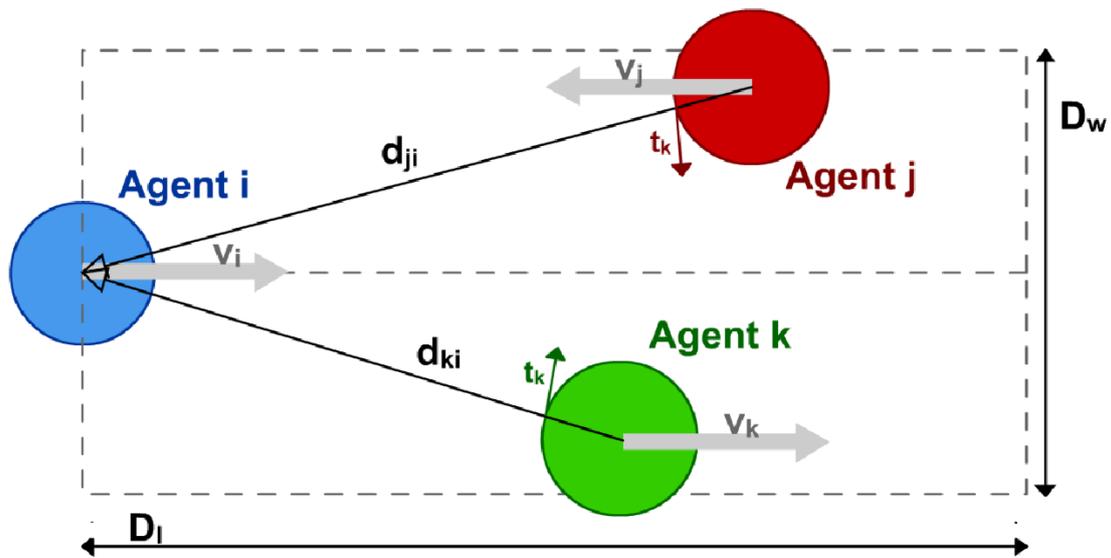


Abbildung 2.5: Wahrnehmungsbereich eines Agenten und Verhältnis zu benachbarten Agenten zur Kollisionsvermeidung beschrieben nach Sullivan (2009). Bildquelle: Sullivan (2009)

### 2.3 Stress in Menschenmengen

Allgemein ist Stress zu definieren als eine Reaktion auf äußere Reize, die den psychischen und physischen Zustand eines Menschen beeinflussen (vgl. Schröder (1996), zit. nach Hartig (2011), S. 6). Hartig (2011) beschreibt auch, dass diese äußeren Reize, sogenannte Stressoren, die Faktoren sind, die höchstwahrscheinlich Stress hervorrufen. Er unterteilt Stressoren nach der Arbeit von Janke (1976) in äußere Stressoren, Leistungsstressoren, soziale Stressoren und Bedingungen, die die eigenen Bedürfnisse einschränken sowie Stressoren wie Konflikte oder Ungewissheit. Für die Simulation von Menschenmengen sind besonders äußere und soziale Stressoren sowie Leistungsstressoren von Interesse. Äußere Stressoren können zum Beispiel bei Evakuierungen durch versperrte Wege aufgrund von Hindernissen oder Feuer in einem Gebäude den Stress der Agenten erhöhen. Auch die Kollision der Agenten untereinander oder mit Hindernissen kann dazu führen, dass der Stress steigt. Der letzte Faktor lässt sich auch als sozialer Stressor einordnen, da ein Zusammenstoßen bei einer hohen Dichte der Menschenmenge wahrscheinlicher ist und eine hohe Dichte den Stress ebenfalls steigern kann. Als Leistungsstressor lässt sich der zeitliche Druck benennen, dem Agenten ausgesetzt sein können, um ihr Ziel rechtzeitig zu erreichen. Diese Stressoren lassen sich in einer Simulation von Menschenmengen integrieren, um die Repräsentation des Stres-

ses von Menschen als Stressniveau abbilden zu können. Die Komplexität des Stresses wurde deshalb in dieser Arbeit versucht zu approximieren, um eine geeignete Darstellung von Stress in einer Simulation zu erzielen.

## **2.4 Zusammenfassung**

In diesem Kapitel wurden die Grundlagen für die in der gegenwärtigen Arbeit entwickelte Simulation vorgestellt. Die Grundlagen umfassten Verfahren aus Videospiele und der Menschenmengen-Simulation sowie ein grundlegendes Verständnis zu Stress. Es wurden Elemente einer Echtzeitphysik beschrieben wie Collider, physikalische Eigenschaften und Rigid Bodies die zur physikalischen Repräsentation eines Agenten von Bedeutung sind. Die Künstliche Intelligenz beschrieb das grundlegende Vorgehen, um bei einem Agenten ein Verhalten umzusetzen. Dazu wurde zusätzlich das Verhalten in Menschenmengen beleuchtet. Außerdem wurde herausgestellt, dass sich zur Umsetzung für die vorliegende Arbeit ein mikroskopisches System eignet. Zur Navigation eines Agenten wurden weiterhin Technik zur Kollisionvermeidung beschrieben, die einem Agenten eine kollisionsfreie Fortbewegung ermöglichen. Mit dem grundlegenden Verständnis zu Stress lässt sich das Verhalten eines Agenten auf psychischer Ebene abbilden. Nachdem nun ein Grundverständnis vermittelt wurde, ist es möglich weitere Ansätze näher zu betrachten. Als nächstes werden verwandte Ansätze zu dem in der vorliegenden Arbeit entwickelten Ansatz vorgestellt und eine anschließende Abgrenzung zu diesen vorgenommen.

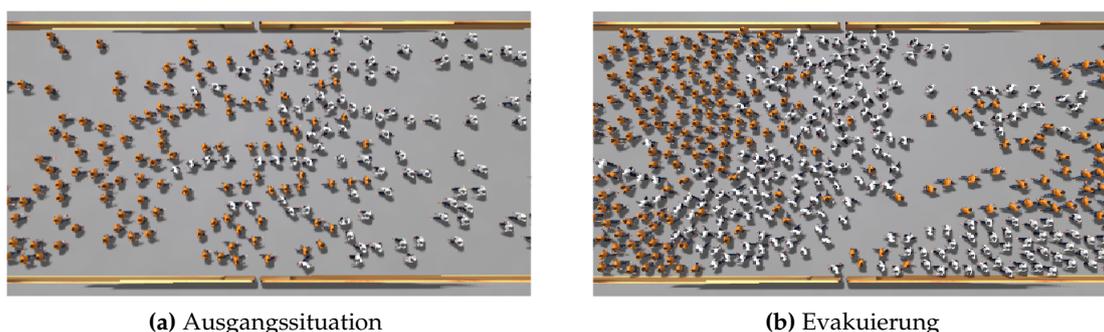
### 3 Verwandte Forschungsansätze

Im Bereich der mikroskopischen Menschenmengensimulation gibt es eine Reihe von verschiedenen Ansätzen. Van Goethem (2012) entwickelte einen Algorithmus, welcher den Bewegungsfluss von individuellen Agenten und Gruppen von Agenten in unterschiedlichen Dichten von Menschenmengen koordiniert, um kleine Menschenmengen in gleicher Form abbilden zu können wie große Menschenmengen. Einen anderen Ansatz verfolgen Rymill u. Dodgson (2005), die mit einer Simulation in Echtzeit das Verhalten von Menschen in einer Menschenmenge mit dem Fokus auf Kollisionsvermeidung simulieren. Oft wird das Hauptaugenmerk auf die Fortbewegung und die Navigation der Agenten gelegt und weniger psychische sowie physische Faktoren und die Interaktion mit der Umwelt untersucht. Kim u. a. (2012) allerdings präsentieren ein Simulationssystem, welches das dynamische Verhalten von Menschenmengen durch Modellierung von Stress zeigt. Einen physikalisch basierten Ansatz stellen Kim u. a. (2013) vor, indem sie einen auf Geschwindigkeit basierenden Kollisionsvermeidungs-Algorithmus mit externen physikalischen Kräften kombinieren. Diese Arbeiten verfolgen jeweils ähnliche Ansätze, die in der vorliegenden Arbeit in einfacher Form zusammengeführt wurden. Als Nächstes werden diese beiden Ansätze eingehend beschrieben und abschließend Parallelen und Unterschiede zu dem Konzept dieser Arbeit zusammengetragen.

#### 3.1 Modellierung von Stress in Menschenmengen-Simulationen

Ein Teil der vorliegenden Arbeit beschreibt die Abbildung von Stress in einer Menschenmengen-Simulation. Eine allgemeine Definition von Stress ist die Reaktion auf die Interaktion zwischen Individuen und der Umwelt (vgl. Kim u. a., 2012, S. 57). Kim u. a. (2012) verfolgen einen Ansatz, Stress möglichst realitätsnah abzubilden und verwenden dazu eine Approximation des Modells des allgemeinen Anpassungssyndroms (AAS) von Hans Selye (vgl. Selye (1956), beschrieben nach Kim u. a., 2012), welches sich im Bereich der Stressforschung etabliert hat. AAS ist ein Muster, das die zeitlich bedingte Reaktion auf Stressoren beschreibt. Man unterscheidet die drei Stadien Alarmbereitschaft, Resistenz und Erschöpfung. Diese Stadien unterteilen die Zunahme von Stress, der im ersten Stadium schnell ansteigt, in dem der Agent durch einen Stressor alarmiert wird. Im zweiten Stadium wird durch eine Stressresistenz versucht, das Stressniveau zu reduzieren. Gelingt dies nicht, wird ein Erschöpfungszustand erreicht, der bis zum Tod führen kann. Da der Stress nach diesem Modell durch situationsbedingte Faktoren zustande kommt und sich Individuen und Menschenmengen

in verschiedenen Situationen unterschiedlich verhalten, wird ebenfalls ein Persönlichkeitswert zur Berechnung des Stresses hinzugenommen. Stressoren und deren Effekte werden anhand von Ergebnissen empirischer Studien modelliert. Sie unterteilen sich in zeitlichen Druck, areale Stressoren, der individuellen Lage betreffende Stressoren und zwischenmenschliche Stressoren. Der erlittene Stress durch die Stressoren wird durch die Stevensche Potenzfunktion (vgl. Stevens (1957), beschrieben nach Kim u. a., 2012) bestimmt, die je nach Stressortyp eine Intensität approximiert. Die Approximation des AAS wird erreicht, indem durch den zugefügten Stress eine Stressreaktion berechnet wird. Zu dieser Berechnung wird der Wertebereich des zugefügten Stresses durch einen Faktor limitiert und die folgende Stressreaktion durch ein Maximum bestimmt. Diese Faktoren beziehen sich auf genau einen Agent und einen Stressor und dienen dazu, dass das Ansteigen des Stresses nicht explosionsartig geschieht. Um diese Berechnungen des Stresses visuell darzustellen, wird eine Multi-Agenten-Simulation verwendet, die das Verhalten eines Agenten individuell steuert und Aggressivität und Impulsivität erhöht.



**Abbildung 3.1:** Agenten bewegen sich durch einen Korridor während ein Alarm ertönt. Bilderquelle: Kim u. a. (2012).

Zur Validierung wurden verschiedene Szenarien durchgeführt. Eines der Szenarien zeigt zwei Gruppen, die sich durch einen Korridor in entgegengesetzte Richtungen bewegen und durch einen Alarm einem arealen Stressor ausgesetzt werden (s. Abbildung 3.1a u. 3.1b). Die Ergebnisse zeigen, dass die sich vorher in einer Spur bewegendenden Agenten nach Auslösen des Alarms unkooperativ verhalten und der Bewegungsfluss verstopft. Ein weiteres Szenario simuliert eine Evakuierung mit tausend Agenten, die ein Bürogebäude wegen des Ausbruchs eines Feuers verlassen müssen (s. Abbildung 3.2 ). Stressoren in diesem Szenario sind das Feuer und zu dichte Menschenmengen. Bei der Evakuierung ist vor allem zu erkennen, dass sich die Agenten schneller be-

wegen und ein eigennütziges Verhalten aufweisen. Die erzielten Ergebnisse zeigen ein realistisches, dynamisches Menschenmengenverhalten, das qualitativ und quantitativ mit denen aus der psychologischen Literatur übereinstimmt. (vgl. Kim u. a., 2012)



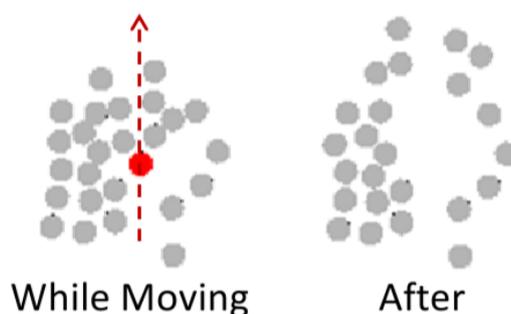
**Abbildung 3.2:** Agenten evakuieren ein Bürogebäude auf Grund von Feuer. Bildquelle: Kim u. a. (2012)

### 3.2 Physikalische Interaktion in Multi-Agenten-Simulationen

Die physikalische Interaktion zwischen Umwelt und Agenten ist ein weiterer wichtiger Teil der vorliegenden Arbeit. Sie lässt sich durch die Effekte des Zusammenstoßens der Agenten untereinander und mit Objekten der Welt sowie dem Umfallen von Agenten oder Objekten beschreiben. Einen Ansatz zur Abbildung von physikalischen Interaktionen verfolgen Kim u. a. (2013), indem sie einen auf Geschwindigkeit basierenden Kollisionsvermeidungs-Algorithmus mit externen physikalischen Kräften verbinden. Ihnen gelingt dabei eine approximierte Simulation der physikalischen Effekte mit plausiblen Verhalten der Agenten in kompakten Umgebungen. Der Ansatz von Kim u. a. (2013) ist so aufgebaut, dass zur Fortbewegung der Agenten eine Berechnung der Geschwindigkeit erfolgt, die sich aus zwei Arten von Constraints bestimmen lässt. ORCA-Constraints beschreiben den Raum der Geschwindigkeit, in dem sich ein Agent ohne Kollisionen mit anderen Agenten oder Objekten bewegt. Kraft-Constraints sind Zwangsbeschränkungen, die sich aus der physikalischen Interaktion zwischen Agenten und Objekten ergeben. Agenten werden als eine zweidimensionale Scheibe dargestellt, um eine Kollisionsvermeidung durch ORCA-Constraints zu erzielen, welche die Geschwindigkeit eines Agenten basierend auf dessen Nachbarn verändert. Die Kraft-Constraints setzen sich aus verschiedenen Kräften zusammen, die von außen auf den Agenten wirken. Ein Agent kann mit anderen Agenten oder Objekten kollidieren. Dabei generiert er eine Stoßkraft, die der kollidierte Agent oder das Objekt erleidet und

kann diese nach Newtonscher Dynamik auf weitere Agenten oder Objekte übertragen. Bei einem Zusammenstoß erfährt ein Agent zusätzlich eine Kollisionskraft, die den Agenten beim Aufprall zurückstößt. Neben diesen beiden impulsiven Kräften werden noch eine Abbrems- und Widerstandskraft beschrieben, die die Bewegung zu den äußeren Kräften anpassen. Diese werden zur Wiederfindung der Balance eines Agenten bestimmt. Zum einen wird durch die Abbremskraft die Schnelligkeit des Agenten ausgenommen der Wirkung externer Kräfte verringert, ohne dessen Richtung zu verändern und zum anderen werden durch die Widerstandskraft die externen Kräfte abgefangen. Der Kraft-Constraint ergibt sich dann als Summe aller Kräfte und wird zur Berechnung der gesamten Geschwindigkeit herangezogen.

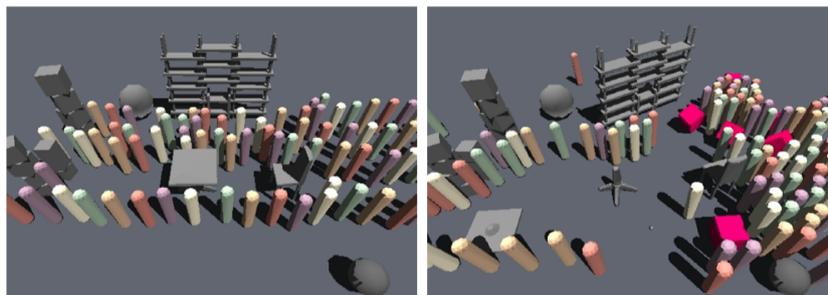
Objekte werden dreidimensional als Rigid Bodies unter Verwendung der Physik-Engine Bullet dargestellt. Die Interaktionen wie Kollisionsvermeidung, Stoßen und ungewollte Kollisionen werden zwischen Agenten und Objekten gezeigt. Da Objekte als Rigid Bodies repräsentiert werden, wird zur Interaktion die Bounding Box (Collider) des Objektes vom dreidimensionalen Raum auf eine zweidimensionale Fläche projiziert.



**Abbildung 3.3:** Ein Agent passiert eine Gruppe von Agenten und drückt diese weg. Bildquelle: Kim u. a. (2013)

Die Ergebnisse des Ansatzes wurden anhand der Durchführung und Analyse verschiedener Szenarien erzielt. Zum einen wird die Demonstration der Kraftausbreitung durch eine Interaktion zwischen Agenten und zum anderen die Verbindung zwischen dynamischen Objekten und den Agenten beschrieben. Zu Ersterem wurde als Beispiel ein Durchlauf-Szenario getestet, bei dem ein Agent durch eine Gruppe von Agenten läuft. Mit physikalischen Constraints ist die Ausbreitung der Kräfte unter den Agenten zu sehen, da im Gegensatz zur Simulation ohne Kräfte-Constraints, die Agenten verstreut zur Seite gedrückt werden (s. Abbildung 3.3). Zu Letzterem dient ein überfülltes Büro-Szenario, bei dem eine Gruppe von Agenten durch ein Bürogelände mit umher-

liegenden und stehenden Objekten navigieren muss (s. Abbildung 3.4). Die Ergebnisse zeigen, dass sich Agenten untereinander und den Objekten ausweichen, aber auch miteinander kollidieren, da der Benutzer zusätzliche Objekte in die Szene einfügen oder werfen kann. Durch die Interaktion des Benutzers präsentiert sich die Echtzeitfähigkeit des Ansatzes. Aus allen Ergebnissen der Untersuchungen ergibt sich ein Ansatz, der interaktives und plausibles physikalisches Verhalten abbilden kann.



**Abbildung 3.4:** Agenten bewegen sich durch einen mit Objekten (Rigid Bodies) gefüllten Raum. Bildquelle: Kim u. a. (2013)

### 3.3 Unterscheidung verwandter Ansätze

Die oben beschriebenen Ansätze schildern zwei aktuelle Verfahren, Menschenmengen auf verschiedene Weisen zu simulieren. Der in Kapitel 3.1 zusammengefasste Ansatz konzentriert sich darauf, ein realistisches Verhalten der Agenten durch die Modellierung von Stress zu erzielen. Im Gegensatz dazu setzt der in Kapitel 3.2 beschriebene Ansatz den Schwerpunkt auf eine plausible physikalische Interaktion der Agenten. Allgemein gesehen simuliert der erste Ansatz psychische, der zweite Ansatz physische Reaktionen auf äußere Einflüsse. Ansonsten sind die Ansätze in der Umsetzung der Navigation der Agenten ähnlich. Sie verfolgen beide einen auf Geschwindigkeit basierenden Ansatz zur Fortbewegung der Agenten.

Beide Ansätze erzielen auf ihre Weise einen deutlichen Mehrwert für eine realistische Simulation von Menschenmengen in unterschiedlichen Situationen. Eine Erweiterung hierzu wäre die Kombination von physischem und psychischem Verhalten. Eine solche Verknüpfung wird in der vorliegenden Arbeit vorgenommen. Allerdings werden nicht die oben genannten Ansätze miteinander kombiniert, sondern ein neuer Ansatz erarbeitet und umgesetzt, mit dem es möglich ist, das psychische und physische Verhalten eines Agenten in einer Menschenmengen-Simulation abzubilden.

Der Unterschied des in dieser Arbeit vorgestellten Ansatzes zum Ansatz in Kapitel 3.1 ist die Modellierung von Stress zur Bestimmung des Verhaltens der Agenten. Der in Kapitel 3.1 vorgestellte Ansatz modelliert Stress nach einer Approximation des allgemeinen Anpassungssyndroms und erzielt damit ein realistisches und fundiertes Verhalten von Agenten in Menschenmengen. In der vorliegenden Arbeit wird eine einfachere Abbildung des Stresses vorgenommen, indem der Stress eines Agenten als Stressniveau angenähert wird, um dadurch das Verhalten der Agenten zu bestimmen.

Zum zweiten verwandten Ansatz lassen sich Gleichheiten und Unterschiede benennen. Ähnlich zum Ansatz der vorliegenden Arbeit wird zur Abbildung der physikalischen Interaktion eine Physik-Engine und Rigid Bodies verwendet. Zudem verfügt der Ansatz aus Kapitel 3.2 wie in der vorliegenden Arbeit über eine Kollisionsvermeidung zwischen Agenten und anderen Objekten sowie deren Kollision. Allerdings werden nur Objekte der Umgebung als Rigid Bodies modelliert. Agenten werden als zweidimensionale Scheibe repräsentiert und deren Bewegung durch Ändern der Geschwindigkeit bestimmt. Auch die Kollision findet nur auf einer zweidimensionalen Ebene statt, da die Rigid Bodies auf diese Ebene projiziert werden. Der Ansatz der vorliegenden Arbeit repräsentiert hingegen auch die Agenten als Rigid Bodies. Die Fortbewegung der Agenten wird durch die Bestimmung einer Kraft anhand der Geschwindigkeit vorgenommen, die der Agent dann zum Fortbewegen nutzt. Somit ergibt sich auch eine Kollision im dreidimensionalen Raum, da Objekte und Agenten durch Rigid Bodies repräsentiert werden.

Zudem grenzt sich dieser Ansatz von den anderen Ansätzen ab, da die Möglichkeit besteht, Verletzungen der Agenten farblich codiert zu visualisieren sowie die Möglichkeit vorhanden ist, dass Agenten stürzen. Das Konzept des Ansatzes dieser Arbeit wird im folgenden Kapitel erläutert und verdeutlicht die unterschiedliche Herangehensweise der verwandten, beschriebenen Ansätze.

## 4 Konzept

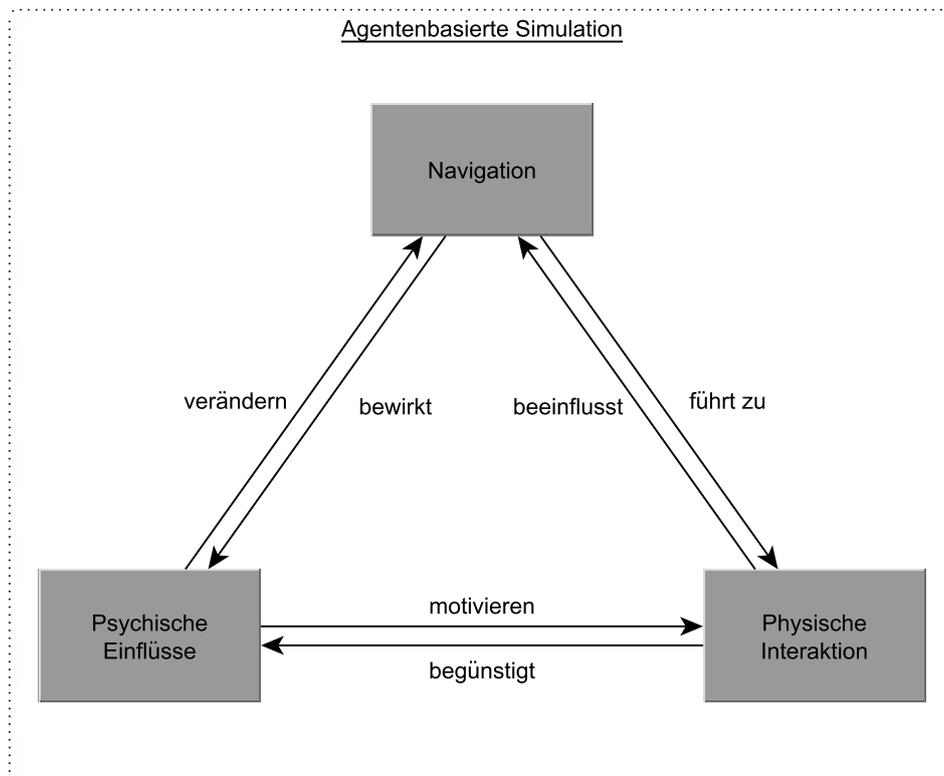
In diesem Kapitel wird das Konzept der vorliegenden Arbeit beschrieben. Zunächst werden die gesetzten Ziele und gestellten Anforderungen einer Menschenmengen-Evakuierung im Bezug auf psychisches und physisches Verhalten unter Verwendung einer Echtzeitphysik aus Videospiele erläutert. Anschließend wird die Herangehensweise an diesen Ansatz verdeutlicht. Es wurden hierzu zwei Prototypen konzipiert, welche zwei verschiedene Herangehensweisen an die Ziele und Anforderungen festlegen. Zuletzt wird der Aufbau und die Struktur dieser beschrieben.

### 4.1 Ziele und Anforderungen

Ziel dieser Arbeit ist es, eine Simulation für Menschenmengen insbesondere für Evakuierungsszenarien zu entwickeln, die neben der Berücksichtigung der Navigation, physische Krafteinwirkungen und psychisches Verhalten einzelner Agenten abbildet. Diese Anforderungen, wie in Abbildung 4.1 gezeigt, beziehen sich auf den Agenten der Simulation, da der Agent als Individuum den größten Teil einer agentenbasierten Menschenmengen-Simulation ausmacht. Zu den Anforderungen an einen Agenten gehören die Navigation, die psychischen Einflüsse und die physische Interaktion. Alle drei Bereiche arbeiten miteinander. Die physische Interaktion begünstigt zum Beispiel durch Zusammenstöße die psychischen Einflüsse und beeinflusst dadurch auch die Navigation. Sie wird durch psychische Einflüsse wie erhöhtem Stress motiviert, welche zum Beispiel durch ein rücksichtsloseres Verhalten die Navigation verändern. Die Navigation bewirkt durch nicht rechtzeitiges Erreichen eines Ziels die psychischen Einflüsse und kann dann zu physischer Interaktion führen.

Zusätzlich wurden weitere Anforderungen an die Simulation gestellt. Zur Umsetzung der Simulation sollte ausschließlich frei verfügbare Software zum Einsatz kommen. Die entwickelten Prototypen sollten außerdem echtzeitfähig sein, um die Simulation in kurzer Zeit ggf. mehrfach durchzuführen und eine Benutzerinteraktion zu gewährleisten. Eine denkbare Interaktion des Benutzers wäre zum Beispiel durch zusätzliche Zäune ein abgestecktes Gelände zu verkleinern oder zu verändern oder bei Bedarf ein Feuer auszulösen.

Als Erstes musste eine Simulationsmethode gewählt werden. Die einzige Methode, die hierfür in Frage kommt, ist die agentenbasierte Simulation der mikroskopischen Systeme. Der Vorteil dieser Simulation ist, dass Agenten als Individuen gesehen werden und



**Abbildung 4.1:** Anforderungen eines Agenten in einer agentenbasierten Menschenmengen-Simulation

für sich selber agieren und interagieren. Eine alternative Simulationsmethode wie die Fluidsimulation zu verwenden ist unbestritten, da den Anforderungen entsprechend eine Interaktion zwischen den einzelnen Agenten möglich sein muss. Außerdem muss jeder Agent ein eigenes Verhalten aufzeigen können. Insgesamt muss der Agent in einer agentenbasierten Simulation den Anforderungen in Abbildung 4.1 gerecht werden.

Dazu muss zunächst eine geeignete physikalische Repräsentation bestimmt werden, um physische Interaktionen abbilden zu können. Hierzu können Rigid Bodies in Verbindung mit Collider verwendet werden, die sich problemlos mit Hilfe einer Physik-Engine umsetzen lassen. Sie ermöglichen es einem Agenten physikalische Eigenschaften zuzuschreiben. Dadurch kann ein Agent durch Kräfte realistisch bewegt werden und mit anderen Agenten und Hindernissen in seiner Umgebung kollidieren. Außerdem ist es möglich den Agenten durch mehrere zusammengesetzte Rigid Bodies und

Collider zu repräsentieren, was eine Beweglichkeit des Körpers mit sich zieht. Rigid Bodies können einer simulierten Schwerkraft ausgesetzt sein, wodurch es möglich ist, das Umfallen der Agenten umzusetzen.

Desweiteren benötigt ein Agent eine adäquate und zielstrebige Navigation, die es ihm ermöglicht sich uneingeschränkt und kollisionsfrei fortzubewegen. Das umschließt das Wissen über die begehbaren Bereiche seiner Umgebung und die Möglichkeit sich in dieser fortzubewegen. Außerdem wird eine Kollisionsvermeidung benötigt, die es dem Agenten erlaubt anderen Agenten und Hindernissen auszuweichen. Um Ersteres zu erhalten, lässt sich ein statisches Navigationmesh verwenden, welches die möglichen passierbaren Stellen innerhalb eines Gebäudes beschreibt. Dieses Verfahren zur Navigation wird standardmäßig in der Videospieldomäne eingesetzt. Dadurch plant ein Agent die kürzesten und realistischen Wege, um Wände und andere feste Hindernisse zu umgehen. Ein statisches Navigationmesh hindert die sich bewegenden Agenten allerdings nicht daran zu kollidieren. Um dies zu verhindern wäre die Verwendung eines dynamisches Navigationmeshes sinnvoll, bei dem alle Agenten als Hindernisse innerhalb des Meshes angesehen werden und das Mesh bei Bewegung der Agenten jedes Mal neu berechnet wird, um einen Weg um andere Agenten herum zu planen. Allerdings stehen für die Umsetzung eines dynamischen Navigationmeshes keine frei zugänglichen Frameworks zur Verfügung und eine eigene Implementierung war hier nicht Teil des Themas und könnte in einer Weiterentwicklung der Simulation umgesetzt werden.

Damit der Agent Kollisionen vermeiden kann, muss ein anderer Ansatz verfolgt werden. Dazu wäre eine Kollisionsvermeidung nötig, die sich aus den Ansätzen von Sullivan (2009) und Musse u. Thalmann (1997), die in Kapitel 2.2.3 vorgestellt wurden, zusammensetzt. Diese Kollisionsvermeidung ermöglicht es dem Agenten, ohne das Wissen über die ganze Umgebung, dynamischen Objekten wie andere Agenten, aber auch statischen Objekten auszuweichen. Damit sich der Körper physikalisch plausibel bewegt, darf er sich nur durch eine Kraft bewegen, bei der die Masse mit in Betracht gezogen wird. Die Agenten bewegen sich also nicht durch die direkte Veränderung der Geschwindigkeit, sondern durch eine Kraft, die in Relation zur Masse ihre Geschwindigkeit verändert. Das ist möglich, da der Agent durch einen Rigid Body repräsentiert wird.

Zuletzt muss es den Agenten möglich sein, auf psychische Einflüsse zu reagieren. Da-

zu muss ein Agent ein Verhalten besitzen, das den psychischen Zustand eines Agenten und die physische Beschaffenheit abbildet. Diese physisch-psychischen Zustände können mit einer Verhaltens-KI beschrieben werden. Sie sollten durch unterschiedliche Faktoren in der Umwelt beeinflusst werden und wiederum den Agenten selbst in seiner Bewegung und Interaktion mit der Welt beeinflussen.

## **4.2 Herangehensweise der Umsetzung verschiedener Prototypen**

Zur Umsetzung der oben beschriebenen Ziele und Anforderungen werden zwei Herangehensweisen verfolgt. Dabei werden zwei Prototypen entwickelt, mit denen jeweils ein Ziel genauer angestrebt wird. Bei dem ersten Prototypen wird der Fokus auf die physikalische Repräsentation des Agenten gelegt. Der Kern des zweiten Prototyps ist eine Menschenmengen-Simulation, die zudem die Navigation und das psychische Verhalten eines Agenten abbilden kann.

Mit dem ersten Prototypen wird das Ziel verfolgt, eine möglichst menschenähnliche, physikalische Repräsentation eines Agenten zu erzielen. Die Repräsentation wird durch mehrere Rigid Bodies beschrieben, die jeweils durch einen Joint miteinander verbunden sind. Dadurch lässt sich ein menschlicher Körper von zwei übereinander angebrachten Rigid Bodies bis hin zu einer so genannten Ragdoll zusammenstellen. Die zusammengesetzten Rigid Bodies werden so miteinander vereint, dass sie selber die Balance halten und sich miteinander bewegen. Mit diesem Prototypen sollte vor allem eine physikalisch plausible Bewegung und Kollision sowie das Umfallen der Agenten gezeigt werden.

Der zweite Prototyp konzentriert sich auf eine Menschenmengen-Simulation, die die Navigation, das psychische Verhalten und die physikalische Repräsentation eines Agenten zusammenführt. Dieser Prototyp dient dazu, ganze Menschenmengen zu simulieren, um die Interaktion mit der Umwelt durch psychische und physische Einflüsse abzubilden. Anders als beim ersten Prototyp, wird eine einfachere physikalische Repräsentation verwendet. Statt mehrerer Rigid Bodies, die durch Joints miteinander verbunden sind, wird vereinfacht nur ein einziger Rigid Body und Collider verwendet, der den Körper des Agenten darstellt. Zur Repräsentation wird eine längliche Kapsel gewählt (siehe Abbildung 5.4), um eine passende Größe eines Menschen zu simulieren. Ein weiterer Vorteil dieser Form ist das ähnliche Verhalten beim Kollidieren oder Umfallen eines Agenten zum Menschen. Nachteilig ist zu erwähnen, dass durch die

Abwesenheit von Joints keine Beweglichkeit im Körper des Agenten vorhanden ist. Neben der physikalischen Repräsentation der Agenten wird das Verhalten der Agenten modelliert. Dieses Verhalten bezieht sich auf physische Interaktionen der Agenten untereinander sowie auf die Anpassung der Navigation und wird durch ein Stressniveau und einen Gesundheitswert eines Agenten gesteuert. Physische Interaktionen sind hier das Stoßen oder das Umfallen der Agenten. Die Navigation schließt eine Pfadplanung und Kollisionsvermeidung mit ein. Im Nachfolgenden wird der Aufbau beider Prototypen erläutert.

### **4.3 Aufbau und Struktur des ersten Prototypen**

Der erste Prototyp konzentriert sich auf die Untersuchung des physikalischen Verhaltens eines einzelnen Agenten, um eine menschenähnliche Repräsentation zu erhalten. Zunächst muss dazu eine geeignete physikalische Repräsentation gewählt werden. Die Repräsentation durch Rigid Bodies ist ideal, da diese mit einer Echtzeitphysik umsetzbar sind und es dem Agenten ermöglichen, sich physikalisch plausibel zu verhalten. Ein Ansatz, der für diesen Prototypen verfolgt wird, ist den Körper durch mehrere Rigid Bodies zu beschreiben. Dabei ist die Anzahl der Rigid Bodies beliebig. Optimal ist eine Ragdoll, die aus so vielen miteinander verbundenen Rigid Bodies besteht, so dass sie die Beweglichkeit eines Menschen möglichst genau abbildet. Die einzelnen Rigid Bodies werden durch Joints miteinander verbunden. Verschiedene Gelenke des Körpers haben dabei verschiedene Einschränkungen in der Beweglichkeit. Deswegen muss ein frei justierbarer Joint gewählt werden, mit dem sich alle Beweglichkeiten individuell einstellen lassen.

Durch eine Kombination aus zwei Skeletten und einem Modell könnte eine Art kontrollierbare Ragdoll erzeugt werden. Das erste Skelett, das Animationsskelett, dient dazu das modellierte Mesh zu bewegen. Das zweite Skelett ist eine Konstruktion aus mehreren Rigid Bodies verbunden durch Joints, das eine physikalische Repräsentation des Körpers darstellt.

Der modellierte Agent ist einer simulierten Gravitation ausgesetzt und es muss eine Balance der Rigid Bodies erreicht werden, da der Agent sonst umfallen würde. Hierzu gibt es zwei Möglichkeiten. Zum einen kann der Agent versuchen durch eigene Bewegung die Balance zu finden, zum anderen könnte der Agent durch eine Art Stütze stabilisiert werden. Durch Bewegung die Balance zu finden müsste durch einen Aus-

gleich der Kräfte geschehen, die auf den Agenten wirken und ihn zu Boden ziehen. Die Stützen oder Controller könnten so gewählt werden, dass sie sich unabhängig von der Physik bewegen und dem Agenten eine konstante, aufrechte Haltung ermöglichen. Sollte der Agent umfallen, kann er sich zu der Position der Controller zurückbewegen.

Diese Balance, bei einem komplex konstruierten Rigid Body Skelett wie einer Ragdoll zu finden, ist sehr aufwendig. Deshalb ist es sinnvoll, zunächst ein vereinfachtes Skelett zu verwenden, das zum Beispiel aus nur zwei Knochen besteht. Mit dieser Vereinfachung können die Ansätze zur Erhaltung der Balance effektiver untersucht werden. Zur Findung der Balance müssen nur zwei Körperteile bewegt werden und um den Körper durch Controller zu stützen müssten auch weniger Stützen verwendet werden. Außerdem bleibt eine Beweglichkeit des Körpers erhalten. Mit dieser lassen sich einfache Zusammenstöße abbilden, da sich die zusammengesetzten Teile des Rigid Body Skeletts bei einer Kollision unterschiedlich bewegen können.

#### **4.4 Aufbau und Struktur des zweiten Prototypen**

Der zweite Prototyp setzt den Fokus darauf, eine prototypische Simulation einer Menschenmenge durchzuführen. Der Prototyp ist in seinem Aufbau in die drei Ebenen Willenskräfte, physikalische Kräfte und psychische Zustände zu unterteilen. Dabei bezieht sich die Navigation auf die Willenskräfte; physische Interaktion und physikalischen Einfluss beschreiben die physikalischen Kräfte und psychische Einflüsse erzeugen einen physisch-psychischen Zustand. Diese Unterteilung bezieht sich pro Agent auf sich und seine Umwelt. Alle drei Ebenen greifen untereinander und ineinander ein, sodass eine Ebene die andere beeinflusst.

Im Gegensatz zum ersten Prototypen, bei dem der Schwerpunkt eine menschenähnliche physikalische Repräsentation ist, wird bei dem zweiten Prototypen die Kombination aus Navigation, physikalischer Repräsentation und physisch-psychischem Zustand untersucht. Dazu wird die physikalische Repräsentation des Agenten zunächst auf nur einen Rigid Body verringert, um eventuelle Probleme der Balance zu vermeiden. Ein Agent sollte sich in der Simulation nur innerhalb eines definierten Raumes bewegen. Der Raum sollte durch einen Boden und durch Wände bestimmt sein.

#### 4.4.1 Willenskräfte

Die Willenskräfte beziehen sich in diesem Zusammenhang auf die Navigation eines Agenten. Die Willenskraft beschreibt den psychischen Zustand etwas aus eigener Verhaltensweise zu erreichen und eventuelle Hindernisse zu überwinden. In der vorliegenden Arbeit lässt sich die Willenskraft eines Agenten auf dessen Navigation abbilden, da das Navigieren sein willentliches Verhalten beschreibt.

Ein Ansatz zur Navigation der Agenten wäre ein Navigationmesh. Dazu könnte für einen Raum ein Navigationmesh erzeugt werden, das einem Agenten als eine Art Karte dient, die ihm hilft zu erkennen, in welchen Bereichen des Raumes er sich bewegen kann. Das generierte Navigationmesh ist eine triangulierte Fläche, durch die sich mit Hilfe des A\*-Algorithmus ein Weg, bestehend aus mehreren Wegpunkten mit Start- und Endpunkt, berechnen lässt. Dieser Weg lässt sich dann von dem Agenten vom Start- zum Endpunkt über die Wegpunkte ablaufen.

Zusätzlich muss es dem Agenten möglich sein, nur einen Teil seiner Umgebung wahrzunehmen. Das bedeutet, dass er nur über einen definierten Sichtbereich verfügen sollte, um nur Hindernisse in nächster Umgebung wahrzunehmen. In diesem Zusammenhang würde sich ein an das menschliche Sichtfeld angenäherter Sichtkegel eignen. Der Agent wird dann über die Objekte und Agenten innerhalb seines Sichtkegels informiert und versucht diesen gegebenenfalls auszuweichen.

#### 4.4.2 Physikalische Kräfte

Für ein möglichst realistisches, echtzeitfähiges und physikalisches Verhalten der Agenten kann eine Physik-Engine verwendet werden, die die Berechnung der Physik übernimmt. Agenten und Objekte werden durch Rigid Bodies mit einem Collider verkörpert. Dadurch sind die Agenten und Objekte den durch die Echtzeitphysik physikalisch wirkenden Kräften ausgesetzt. Um eine innerhalb der erzeugten Welt physikalisch plausible Berechnung für Kollisionen und den wirkenden Kräften zu gewährleisten, muss die Fortbewegung der Agenten durch physikalische Kräfte entstehen. Deshalb sollten die Geschwindigkeit und Richtung zur Fortbewegung eines Agenten berechnet werden. Allerdings müssen diese vorher zu einer Kraft umgerechnet werden, die in Relation zu der Masse des Agenten steht. Durch Anwendung dieser Kraft kann sich der Agent fortbewegen und mit Hilfe des Drehmoments kann die Rotation in Laufrichtung ausgerichtet werden. Damit der Agent wegen der Rotation durch das Drehmoment

nicht ungewollt umfällt, sollte eine Stütze für den Agenten eingesetzt werden. Hierzu könnte statt eines Controllers die Rotation auf die senkrechte Achse beschränkt werden, sodass die anderen Achsen fixiert sind.

Der physische Zustand des Agenten sollte in der entsprechenden Situation die Fixierung der Rotationsachsen lösen, damit es möglich ist, dass der Agent umfällt. Das Umfallen wird ebenfalls durch die Gravitation unterstützt, die ähnlich der realen Welt als physikalische Kraft herrscht. Kollidieren Agenten, sollte die Kraft der Kollision an die kollidierenden Agenten und Objekte übertragen werden. Dadurch können Agenten sich gegenseitig wegstoßen.

#### **4.4.3 Physisch-psychischer Zustand**

Jedem Agenten wird ein physisch-psychischer Zustand zugeteilt. Diese Beschaffenheit lässt sich unterteilen in einen physischen und psychischen Zustand. Der physische Zustand des Agenten wäre als Gesundheit abbildbar. Mit einer Abbildung von Stress ließe sich ein psychischer Zustand beschreiben. In der vorliegenden Arbeit soll die Gesundheit durch einen Gesundheitswert repräsentiert werden, der Stress durch ein Stressniveau. Idealerweise sollten die Zustände während der Simulation erkenntlich sein, um eine Aussage über die Zustände eines Agenten zur Laufzeit machen zu können.

Der Wert der Gesundheit gibt Aufschluss darüber, wie stark ein Agent verletzt ist. Außerdem lässt sich dadurch beschreiben, ob ein Agent seinen Verletzungen erliegt oder in einen Ohnmachtzustand gelangt. Das bedeutet, der Agent fällt um und kann sich nicht mehr fortbewegen. Darüber hinaus wird er durch das Umfallen für andere ein Hindernis.

Das Stressniveau ist eine Orientierung an den möglichen Stress eines Menschen, das die Anspannung und geistige Belastung eines Agenten widerspiegeln soll. Bei einem zu hohen Stressniveau ließe sich ein vergleichbarer Zustand der Panik bei einem Agenten abbilden. In dem Fall ignoriert der Agent benachbarte Agenten und Hindernisse und versucht sich seinen Weg durch Drücken und Stoßen frei zu machen. Das Stressniveau erhöht sich durch verschiedene Stressoren. Das können Ereignisse wie ein ausgebrochenes Feuer sein, aber auch zu viele Menschen, die sich eng beieinander befinden. Eine Stressresistenz soll die unterschiedliche Reaktion von Menschen auf stressbehaftete Situationen beschreiben. Durch diese können die Agenten wiederum unterschiedlich

stark auf die Stressoren reagieren und ein Ansteigen des Stressniveaus teilweise unterdrücken.

#### **4.5 Zusammenfassung**

Das Ziel einer Menschenmengen-Simulation mit dem Fokus auf Navigation, physische Interaktionen und psychischem Verhalten, wurde in diesem Kapitel beschrieben. Dabei wurde die Herangehensweise von zwei unterschiedlichen Prototypen ausführlich dargestellt. Der Aufbau des ersten Prototypen untersucht die physikalische Repräsentation eines Agenten der Simulation, wohingegen der zweite Prototyp den Fokus auf eine Kombination dieser mit der Navigation und dem psychischen Verhalten legt. Dabei entstand eine Unterteilung in Willenskräfte, physikalische Kräfte und einem physisch-psychischen Zustand. Die Konzepte der Prototypen wurden umgesetzt und die Implementierung wird im nächsten Kapitel ausführlich erläutert.

## 5 Implementierung

In diesem Kapitel werden die eingesetzten Frameworks und Bibliotheken vorgestellt und die Funktionen der entwickelten Prototypen auf technischer Ebene detailliert beleuchtet. Zur Umsetzung wurden gezielt Frameworks verwendet, die zur Entwicklung von Videospiele eingesetzt werden, da hierzu frei zugängliche Software existiert und diese durch Integration einer vorhandenen Physik-Engine eine einfache Anbindung und Implementierung ermöglichen konnte. Die Verwendung dieser Software soll außerdem verdeutlichen, dass sich die in dieser Arbeit entwickelte Menschenmengen-Simulation mit bereits vorhandenen Werkzeugen konstruieren lässt.

### 5.1 Eingesetzte Software

Zur Entwicklung der Prototypen kamen zwei unterschiedliche Frameworks zum Einsatz. Der erste Prototyp wurde ausschließlich mit Blender<sup>7</sup> umgesetzt, wohingegen der zweite Prototyp mit Unity<sup>8</sup> unter Verwendung der zuvor erstellten Modelle in Blender umgesetzt wurde. Im Folgenden werden beide Frameworks vorgestellt und deren Vorteile erläutert.

#### 5.1.1 Blender

Blender ist eine freie Open-Source-Software zur Erstellung von 3D-Modellen, Animationen, Simulationen und zur Erzeugung von Videospiele. Es ist ein Gemeinschaftsprojekt, das unter der GNU General Public License (GPL) lizenziert ist. Im Mittelpunkt der Software steht die Modellierung von 3D-Objekten und deren Animation. Alle Modelle, die in beiden Prototypen genutzt werden, wurden eigenhändig in Blender modelliert. Neben der Hauptfunktion der Modellierung bietet Blender die Möglichkeit Videospiele zu kreieren. Hierzu verfügt das Programm über eine eigens integrierte Game-Engine. Mit Hilfe einer Python API ist es möglich eine eigene Logik für das zu entwickelnde Spiel zu erzeugen. Außerdem bietet die Game-Engine den Zugriff auf die Physik-Engine Bullet. Ein Vorteil, der sich hier ergibt, ist die direkte Verwendung der modellierten Objekte innerhalb der Game-Engine. Da zur Umsetzung des Konzepts eine Simulation unter Verwendung von Videospiele-Komponenten implementiert werden sollte, wurde der erste Prototyp in Blender erstellt. Hierbei wurde vor allem die zugehörige Physik-Engine herangezogen, um eine Simulation eines Agenten durchzuführen.

---

<sup>7</sup>Blender: <http://www.blender.org/>, letzter Besuch am 8.12.2013

<sup>8</sup>Unity3D: <http://unity3d.com/>, letzter Besuch am 8.12.2013

Anhand der Python API wurden die zusammengesetzten Rigid Bodies angesteuert, um deren Fortbewegung zu implementieren. Desweiteren verfügt die Blender Game-Engine über die Möglichkeit Navigationmeshes zu generieren. Die in Blender generierten Navigationmeshes, wurden zum einen testweise im ersten Prototypen verwendet und umsichtig im zweiten Prototypen eingesetzt.

### 5.1.2 Unity

Unity ist eine proprietäre Software, die zur Nutzung eine freie Lizenz zur Verfügung stellt, welche in dieser Arbeit verwendet wurde. Die freie Version von Unity reicht für die Anforderungen an die in dieser Arbeit entwickelte Simulation aus. Unity ist eine Game-Engine, die mit verschiedenen Features zur Entwicklung von Videospiele ausgestattet ist. Dazu gehören zum Beispiel die Physik-Engine PhysX, Crowd-Simulation und Animation Handling. Unity eignet sich demnach nicht nur zur Entwicklung von Videospiele, sondern auch zur Entwicklung von Simulationen. In dieser Arbeit wurde die integrierte Physik-Engine verwendet, die verfügbare Crowd-Simulation wurde auf Grund der Restriktionen zur proprietären Version außer Acht gelassen und eine eigene Implementierung zur Navigation in Verbindung mit einem Unity Addon zur Pfadfindung umgesetzt. Das verwendete Addon ist das A\* Pathfinding Project<sup>9</sup>, dass es ebenfalls in einer freien und proprietären Version gibt. Zur Nutzung von Navigationmeshes mit Pfadfindung wurde die freie Version des Addons verwendet und durch eigene Navigation und Kollisionsvermeidung erweitert.

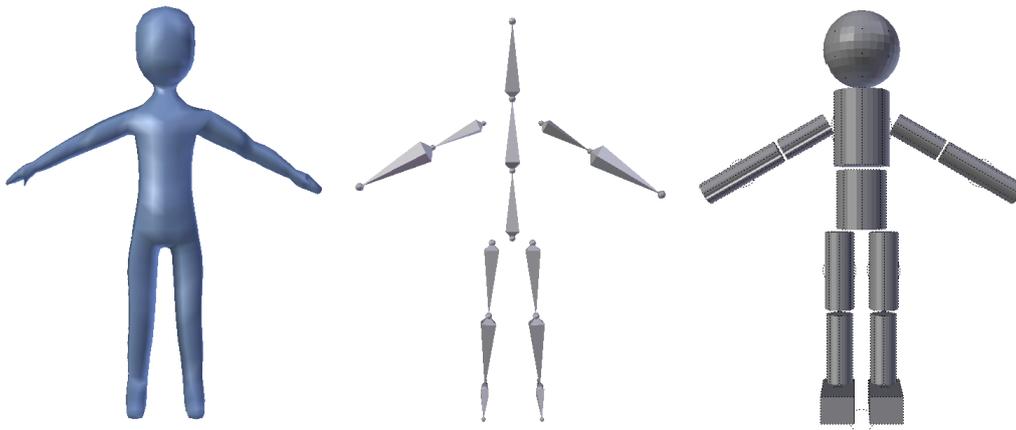
Unity wurde zur Umsetzung des zweiten Prototypen genutzt, da Komponenten wie die Pfadfindung schneller einsetzbar waren. In der Software können sogenannte Szenen erstellt werden. Hierzu wurden die in Blender erzeugten Modelle in Unity integriert und entsprechende Szenarien erstellt. Unity zeichnet sich durch komponentenbasierte Entwicklung aus. Ein Objekt in der Szene hat dabei mehrere Komponenten. Ein Agent in der Simulation hat zum Beispiel eine Rigid Body Komponente, die den Agenten als Rigid Body beschreibt und eine weitere Komponente, die die Navigation des Agenten übernimmt. Komponenten sind Skripte, die in C# implementiert sind. Besonders das Skript zur Navigation beschreibt die wichtigsten Funktionen eines Agenten in der in dieser Arbeit entwickelten Simulation.

---

<sup>9</sup>A\* Pathfinding Project: <http://arongranberg.com/astar/>, letzter Besuch am 9.12.2013

## 5.2 Umsetzung des ersten Prototypen

Zur Umsetzung des ersten Prototypen wurden Blender und die Blender Game Engine verwendet. In ersten Versuchen wurde ein menschenähnlicher Agent modelliert. Dazu wurde ein modelliertes Mesh mit zwei Skeletten kombiniert. Das Animations skelett bewegt das Modell und das aus Rigid Bodies bestehende Skelett bewegt das Animations skelett. Abbildung 5.1 zeigt das Modell und die beiden Skelette.



**Abbildung 5.1:** Das Modell des Agenten aus den ersten Versuchen (links) mit Skelett zur Steuerung des Modells (mitte) und das Skelett bestehend aus Rigid Bodies zur Steuerung des ersten Skeletts (rechts)

Es wurden 13 Knochen für beide Skelette verwendet. Das Skelett bestand aus einem Knochen für den Kopf, zwei für den Körper, jeweils zwei für die Arme und jeweils drei für die Beine. Der Knochen des Kopfes wurde als Kugel repräsentiert, alle anderen Knochen hatten die Form eines Zylinders für eine angenäherte Form des Körpers und die Füße die Form eines Quaders für eine ideale Stabilität. Die einzelnen Rigid Bodies wurden durch frei justierbare Joints miteinander verbunden, die für die entsprechende Bewegung eines Körperteils angepasst und begrenzt wurden. Der Torso wurde mit dem Abdomen verbunden, der Kopf mit dem Torso, die Unterarme jeweils mit den Oberarmen, diese wiederum mit dem Torso. Die Füße wurden mit den Waden und diese mit den Oberschenkeln verbunden, die dann am Abdomen befestigt waren. Diese Konstruktion eines Agenten ergab eine funktionierende, menschenähnliche Ragdoll. Allerdings wurde schnell ersichtlich, dass das Kontrollieren und Balancieren

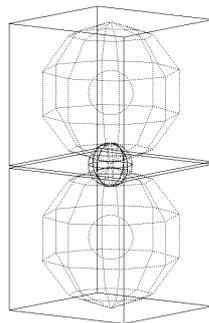
dieser Ragdoll unter dem Einfluss von physikalischen Kräften Schwierigkeiten mit sich brachte. Deshalb wurde zunächst eine vereinfachte Variante der Skelette mit weniger Knochen verwendet.



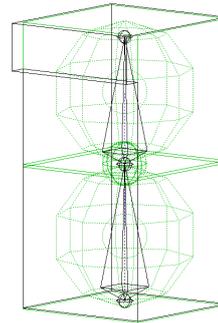
(a) Modell



(b) Animationskelett



(c) Rigid Bodies



(d) Agent zusammengesetzt

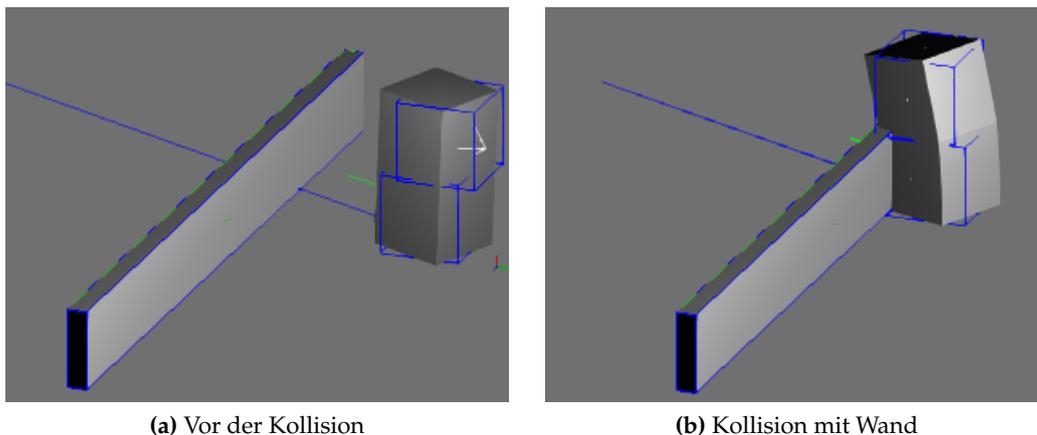
**Abbildung 5.2:** Agent des ersten Prototypen

Diese Variante bestand aus drei übereinander gesetzten Knochen, siehe Abbildung 5.2c. Abbildung 5.2 zeigt außerdem das Modell (5.2a), das Animationskelett (5.2b) und eine Zusammensetzung der drei Teile (5.2d) des Agenten. Das Rigid Body-Skelett ist wie das Skelett der vorher beschriebenen Ragdoll aufgebaut. Als Form der Rigid Bodies wurden Kugeln verwendet für eine realistischere Beugung und um eine Erhöhungen der Umgebung kollisionsfrei zu überwinden. Allerdings gibt es nur einen Oberkörper- und einen Unterkörperteil, der durch ein Zwischenteil mit jeweils einem Joint verbunden

ist. Dieses Zwischenteil dient dazu, ein menschenähnlicheres Gelenk für eine realistischere Bewegung zu erzeugen. Dadurch ist eine Beugung des Oberkörpers möglich, um Zusammenstöße abzufangen, umzufallen oder eine realistischere Fortbewegung durchzuführen. In Abbildung 5.3 ist ein Beispiel zu sehen, wie der Agent mit einer niedrigen Wand kollidiert und der Oberkörper über die Wand neigt.

Die Bewegung geschieht durch eine physikalische Kraft, mit der der obere Teil des Agenten in eine Richtung bewegt wird und sich der untere Teil hinterher bewegt, um die Balance zu halten. Für eine bessere Balance wurde ein ähnlicher Ansatz wie bei Wu u. Popović (2010) und Muico u. a. (2011) verwendet, bei denen zusätzliche, unsichtbare Stützen, auch Controller genannt, die nicht von der Physik betroffen sind, zur Balance und stabilen Bewegung verwendet werden. Dieses Verfahren wurde ansatzweise in vereinfachter Form getestet, wodurch aber Probleme entstanden, sodass die Rigid Bodies nicht mehr vollständig unter physikalischem Einfluss standen. Außerdem sind die von Wu u. Popović (2010) und Muico u. a. (2011) beschriebenen Verfahren zu aufwendig, um sie zur Lösung des Problems heranzuziehen, wodurch die vorliegende Arbeit ihren Fokus verlieren würde. Bei der vereinfachten Variante des Verfahrens wird für die sich bewegenden Teile des Rigid Body Skeletts jeweils ein Controller in die Mitte des Körpers gesetzt, der nicht von der herrschenden Physik betroffen ist. Diese Controller bewegen sich mit dem Rigid Body Skelett, aber bleiben in der senkrechten Achse fest. Die Rigid Body Knochen bewegen sich dann wieder durch die physikalische Kraft zu den Controllern zurück und versuchen deren Position zu halten, um eine bessere Balance zu gewährleisten. Der Nachteil, welcher sich hierbei herausstellte, war der, dass sich das Rigid Body Skelett nicht mehr physikalisch plausibel verhalten hat und zum Beispiel ein Umfallen des Agenten erschwerte. Der Versuch, eine Balance ohne diese Controller zu erzielen, führte dahin, dass die Bewegung zu instabil wurde. Der Agent schaffte es nicht, durchgängig die Balance zu halten und fiel nach einiger Zeit um.

Die Implementierung des ersten Prototypen gliedert sich hierzu in zwei Komponenten. Die erste Komponente erzeugt das Rigid Body Skelett und die zweite Komponente dient zur Steuerung des Rigid Body Skeletts. Die Konstruktion des Rigid Body Skeletts basiert auf einem zuvor modellierten Skeletts, das zur Animation benötigt wurde. Das Animationsskelett bewegt und verformt einen modellierten Agenten. An den Knochen dieses Skeletts werden die Rigid Body Knochen und deren Collider angebracht. Diese steuern dann wiederum das Animationsskelett, um den Agenten anhand physikalischer Kräfte zu bewegen. Beide Skelette bestehen aus zwei Knochen, das Rigid Body



**Abbildung 5.3:** Agent des ersten Prototypen kollidiert mit niedriger Wand

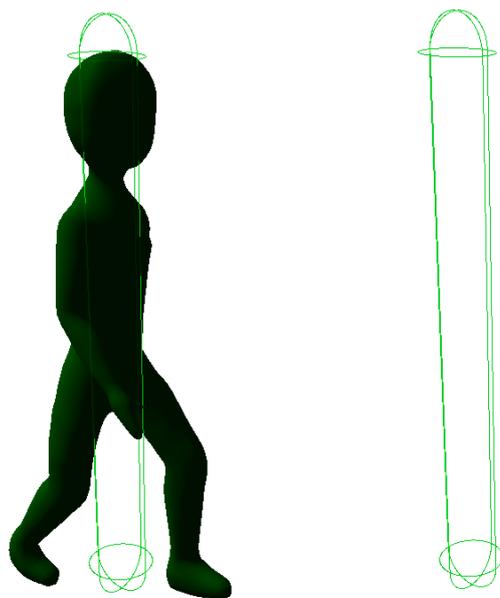
Skelett hat einen zusätzlichen dritten Knochen, der die Funktion eines Gelenks übernimmt. Die beiden Knochen beschreiben den Ober- und Unterkörper, sodass das Gelenk zwischen den beiden Knochen eine realistischere Bewegung des Oberkörpers ermöglicht. Die Komponente zur Fortbewegung dient dazu, eine Balance zwischen Ober- und Unterkörper zu erzielen. Dazu bewegt sich zuerst der Oberkörper durch eine initiale physikalische Kraft nach vorne, die im nächsten Zug durch den unteren Teil des Körpers ausgeglichen wird. Dadurch soll eine natürliche Bewegung entstehen, da sich beim Laufen zuerst der Oberkörper in Bewegung setzt und erst kurz danach der untere Teil die Fortbewegung durchführt.

Allerdings ergaben die Untersuchungen, dass die durch Joints verbundenen Rigid Bodies schwierig zu balancieren sind und der prototypische Agent zu instabil für eine Simulation mit mehreren Agenten ist. Allerdings wurde mit dem zweiten Prototypen zunächst eine vereinfachte Variante der physikalischen Repräsentation verwendet. Es konnte eine Simulation erarbeitet werden, die dieses physische Verhalten zusammen mit der Agentennavigation und den psychischen Reaktionen vereint.

### 5.3 Umsetzung des zweiten Prototypen

Die Umsetzung des zweiten Prototypen beinhaltet die Verwendung von Blender und Unity. Blender wurde zur Modellierung der Räume, Hindernisse und Agenten verwendet. Diese Modelle wurden dann in Unity verwendet, um eine Simulation einer Menschenmenge zu realisieren. Die Simulation ist eine agentenbasierte Simulation, deshalb

beziehen sich die folgenden Teile auf die Eigenschaften eines Agenten. Der Aufbau eines Agenten umfasst verschiedene Bestandteile. Sein Verhalten lässt sich in drei Elemente unterteilen, die Navigation, die Kollision und den physisch-psychischen Zustand. Diese Elemente beeinflussen sich gegenseitig und werden im Folgenden weiter unten beschrieben.



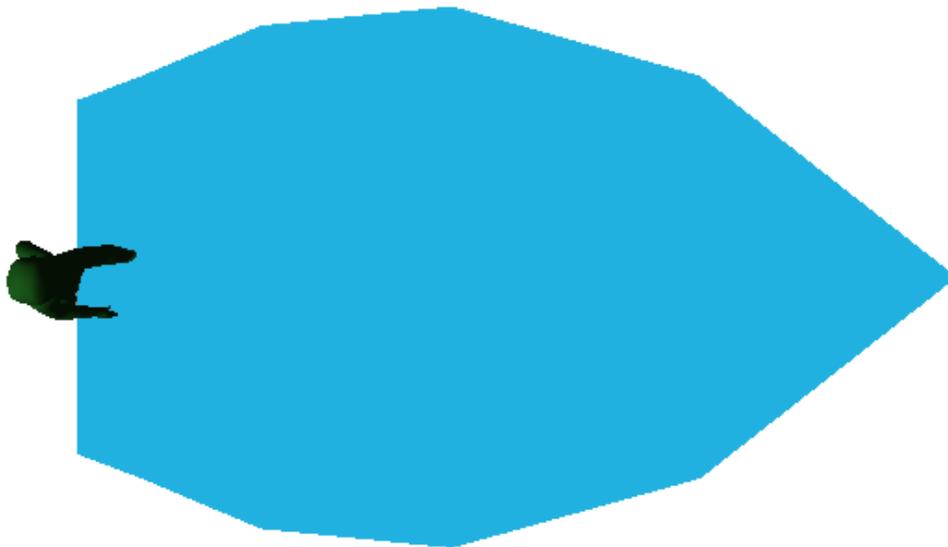
**Abbildung 5.4:** Modell und physikalische Repräsentation eines Agenten des zweiten Prototyps, links Agent mit Kapsel, rechts Kapsel allein

Desweiteren wird ein Agent visuell durch ein dreidimensionales Mesh und physikalisch durch einen Rigid Body und Collider repräsentiert wie in Abbildung 5.4 zu sehen. Ein Agent bewegt sich innerhalb eines definierten Raumes. Dieser Raum ist durch einen Boden und eventuelle Wände bestimmt. Diese Abgrenzungen besitzen Collider, mit denen ein Agent kollidieren kann, sodass er auf dem Boden laufen und an Wände stoßen kann, beziehungsweise sich nicht wie ein Geist durch die Wände bewegt.

Ein zusätzlicher Bestandteil ist der Sichtbereich des Agenten, der mit Hilfe eines Sichtkegels, welcher ein Trigger ist, umgesetzt wurde und zur Kollisionsvermeidung eingesetzt wird. Dieser Sichtkegel, wie in Abbildung 5.5 zu sehen, ist ein Mesh, das als Trigger funktioniert. Objekte und andere Agenten, die diesen Trigger betreten, werden von dem Agenten gesehen. Dieser kann dann den erkannten Hindernissen ge-

benenfalls ausweichen. Die Form des Sichtkegels wurde zunächst dem Sichtfeld eines Menschen nachempfunden. Da der Agent dadurch aber zu viele Hindernisse wahrnimmt und nicht gezielt ausweichen kann, wurde die Form so verändert, dass sie einer Glockenform ähnelt. Mit dieser Form des Sichtkegels ist es dem Agenten möglich sich zielgerichtet fortzubewegen.

Es muss den Agenten außerdem möglich sein, auf psychische Einflüsse zu reagieren. Dazu wurde eine Verhaltens-KI entwickelt, die den psychischen Zustand eines Agenten als Stressniveau und die physische Beschaffenheit als Gesundheit abbildet. Das Stressniveau und die Gesundheit werden durch unterschiedliche Gegebenheiten in der Umwelt beeinflusst und beeinflussen wiederum den Agenten selbst in seiner Bewegung und Interaktion mit der Welt.

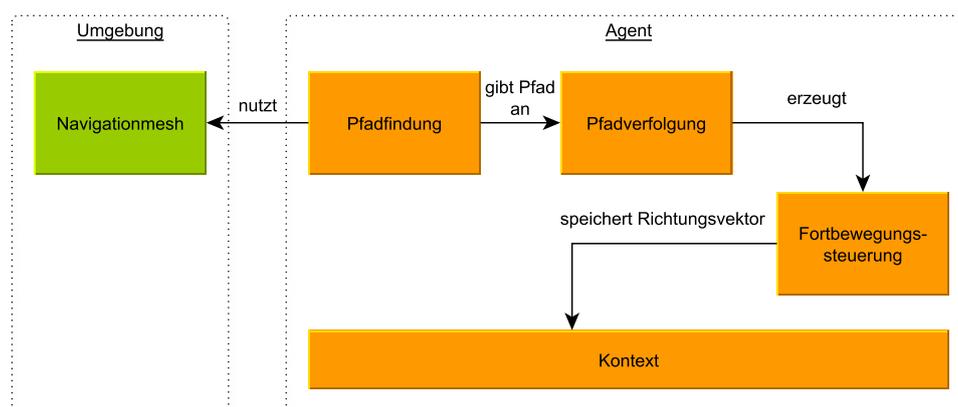


**Abbildung 5.5:** Das Sichtfeld eines Agenten als zweidimensionale Glockenform

### 5.3.1 Navigation

Die Navigation umfasst verschiedene Komponenten, wie in Abbildung 5.6 gezeigt, welche zusammen das Fortbewegungsverhalten eines einzelnen Agenten beschreiben. Ein Agent verfügt standardmäßig über drei Komponenten zur Fortbewegung und eine Schnittstelle der Umgebung, um sich in dieser gezielt zu bewegen. Diese drei Kom-

ponenten verständigen sich zum einen untereinander und zum anderen mit der Umgebung. Die Navigation läuft insgesamt folgendermaßen ab. Zunächst plant ein Agent mit Hilfe der Pfadfindungs-Komponente anhand gegebenem Start- und Endpunkt einen Pfad. Die Pfadverfolgungs-Komponente verwendet den geplanten Pfad, solange keine potenzielle Kollision eintritt, um diesen abzulaufen. Mit Hilfe des Steering Behaviors Suchen wird die Richtung zum nächsten Punkt bestimmt und im Kontext gespeichert. Diese Richtung wird dann zur Berechnung der Kraft für die Fortbewegung verwendet.



**Abbildung 5.6:** Standard Navigationskomponenten eines Agenten und seine Schnittstelle zur Umgebung

## Pfadfindung

Für die Pfadfindung wurde ein Addon Framework eingesetzt, das es ermöglicht, anhand eines gegebenen Navigationmeshes einen Pfad für einen Agenten zu planen. Das Navigationmesh wurde mit Blender generiert und dem Addon Framework zur Verfügung gestellt. Es gibt einem Agenten die Information über begehbare Bereiche in seiner Umgebung. Durch die Pfadfindungs-Komponente des Agenten, stammend aus dem Addon Framework, kann dieser einen Pfad innerhalb des begehbaren Bereichs planen. Zur Planung des Pfades wird ein Start- und Endpunkt benötigt. Dazu wird für den Startpunkt die aktuelle Position des Agenten genutzt und als Endpunkt ein für den Agenten definiertes Ziel bestimmt. Mit diesen Punkten wird durch den A\*-Algorithmus ein Pfad bestehend aus Wegpunkten erzeugt. Diese Wegpunkte befinden sich optimalerweise im Mittelpunkt eines Dreiecks des Navigationmeshes. Durch weitere Optimierungen, um den kürzesten Weg zu planen, verschieben sich die Wegpunkte aus der Mitte des Dreiecks zu einer geeigneten Position. Der geplante Pfad bietet

dem Agenten einige Vorteile. Der Agent kennt zu jeder Zeit den Weg zu seinem Ziel und muss erst bei Erreichung seines Ziels einen neuen Pfad berechnen. Außerdem ermöglicht der geplante Pfad eine kollisionsfreie Navigation durch die Umgebung, allerdings nur mit statischen Objekten. Der so erhaltene Pfad wird anschließend an die Pfadverfolgungs-Komponente weitergegeben.

### **Pfadverfolgung**

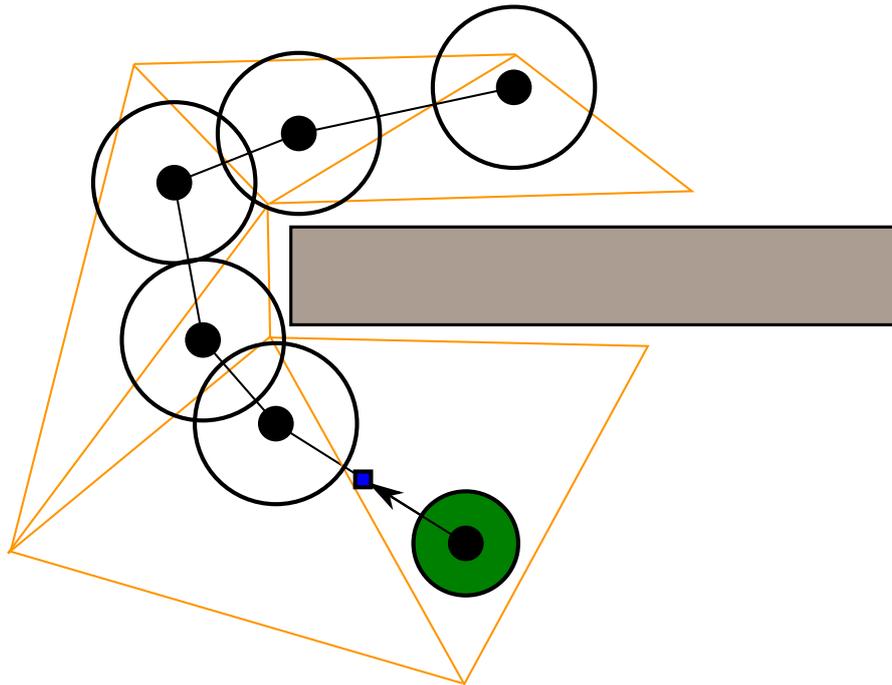
Die Pfadverfolgungs-Komponente dient dem Agenten zum Entlanggehen eines gegebenen Pfades. Den Pfad erhält der Agent durch die Pfadfindungs-Komponente. Reynolds (2004) präsentierte ein Pfadverfolgungsverhalten, bei dem ein Agent einen Pfad aus Linien exakt entlang geht. Diese Präzision wird allerdings in dieser Arbeit nicht benötigt, da eine losgelöstere und realistischere Fortbewegung erzielt werden sollte. Deshalb wird eine Vereinfachung der Pfadverfolgung von Reynolds (2004) verwendet, bei der keine Linien, sondern gegebene Wegpunkte vom Agenten abgelaufen werden. Diese Vereinfachung der Pfadverfolgung wurde aus einem Artikel<sup>10</sup> und dem zur Pfadplanung verwendeten Addon Framework zusammengetragen. Der Agent läuft bei der Pfadverfolgung Wegpunkt für Wegpunkt ab, bis er sein Ziel erreicht. Für eine genauere Fortbewegung und um eventuelle Kollisionen durch Engpässe zu vermeiden, wird statt des nächsten Wegpunktes zwischen der aktuellen Position des Agenten und dem Wegpunkt interpoliert. Der interpolierte Punkt wird dann als nächster Wegpunkt gewählt. Erreicht ein Agent einen Wegpunkt, wird der nächste im Pfad bestimmt und wieder interpoliert, bis das Ziel erreicht ist. Damit der Agent nicht exakt den Wegpunkt passieren muss, hat jeder Wegpunkt einen Radius, den der Agent durchqueren muss, um diesen erreicht zu haben. Dadurch wird dann ein natürlicheres Ablaufen des Pfades erzielt. Abbildung 5.7 zeigt ein Beispiel einer Pfadverfolgung. Zur Erreichung eines Wegpunktes wird das Steering Behavior Suchen verwendet, das im folgenden Abschnitt erläutert wird.

### **Steering Behaviors und Fortbewegungskraft**

Die Steering Behavior Komponente beinhaltet verschiedene Verhalten zur Fortbewegung. Außerdem wird in diesem Abschnitt die allgemeine Fortbewegung eines Agenten erklärt. Es wurden für einen Agenten mehrere Steering Behaviors implementiert.

---

<sup>10</sup>Understanding Steering Behaviors - Path Following: <http://gamedevelopment.tutsplus.com/tutorials/understanding-steering-behaviors-path-following--gamedev-8769>, letzter Besuch am 20.12.2013



**Abbildung 5.7:** Beispiel einer Pfadverfolgung mit interpoliertem Wegpunkt (Blau) eines Agenten (Grün) um eine Wand (Braun) herum, in der durch ein Navigationsnetz (Orange) beschriebenen Umgebung

Dazu zählen das Suchen, das Fliehen, das Umherwandern, das Verfolgen, das Entgehen sowie das Gruppenverhalten Separation, Alignment und Kohäsion. Zum Einsatz kommen in der entwickelten Simulation allerdings nur das Suchen und das Fliehen, da zur weiteren Berechnung der Fortbewegung nur die Richtung der Funktionen benötigt wird. Die Berechnungen werden mit dreidimensionalen Vektoren durchgeführt. Das Suchen berechnet einen Richtungsvektor zu einem vorgegebenen Ziel wie im Pseudocode 5.1 gezeigt. Dazu wird ein Geschwindigkeitsvektor `desiredVelocity` von der Position des Agenten `agentPosition` zum Ziel `destination` ermittelt, normalisiert und mit einem maximalen Geschwindigkeitsfaktor `maxVelocity` skaliert. Sollte das Ziel näher als in einem vorgegebenen Radius `slowdownRadius` sein, wird der Vektor ebenfalls mit dem Quotient zwischen der Distanz zum Ziel `distance` und `slowdownRadius` verkleinert. Diese Skalierung dient dazu, die Geschwindigkeit des Agenten kurz vor dem Erreichen des Ziels zu limitieren. Die Differenz von `desiredVelocity` und des aktuellen Geschwindigkeitsvektors `agentVelocity` ergibt den Vektor, der die Richtung zum Ziel des Agenten bestimmt. Das Suchverhalten wurde, wie oben erwähnt, nur zur Bestimmung der Richtung für die Wegpunkte bei

der Pfadverfolgung verwendet.

Das Fliehen läuft in ähnlicher Weise ab. Hierbei wird der entgegengesetzte Geschwindigkeitsvektor bestimmt und nur mit der maximalen Geschwindigkeit limitiert. Der Richtungsvektor, der von der zu fliehenden Position abgewandt ist, ergibt sich wiederum aus der Differenz der berechneten und der aktuellen Geschwindigkeit. Das Fliehverhalten wird zur Flucht vor einem Feuer verwendet und zwar nur, wenn sich das Feuer im Sichtfeld des Agenten befindet.

```
Data : Anzustuerndes Ziel
Result : Richtungsvektor zum Ziel
Function seek( Vector3 destination )
    Vector3 desiredVelocity = destination - agentPosition;
    float distance = desiredVelocity.magnitude;
    desiredVelocity.Normalized();
    if ( distance <= slowdownRadius ) then
        | desiredVelocity *= maxVelocity * ( distance / slowdownRadius );
    else
        | desiredVelocity *= maxVelocity;
    end
    return desiredVelocity - agentVelocity
End
```

**Pseudocode 5.1** : Steering Behavior: Suchen

Die Fortbewegung des Agenten geschieht mit einer Kraft, die mit Hilfe des durch die Pfadverfolgung und Kollisionsvermeidung errechneten Vektors `steeringForce` erzielt wird (s. Pseudocode 5.2). Solange keine potenzielle Kollision stattfinden kann, wird der Vektor der Pfadverfolgung verwendet, ansonsten wird der Vektor der Kollisionsvermeidung genutzt. Nachdem die Vektoren berechnet wurden, werden sie durch einen maximalen Wert `maxForce` limitiert. Anschließend wird die aktuelle Geschwindigkeit des Agenten `agentVelocity` mit dem Vektor `steeringForce` addiert und mit einem Schnelligkeitwert `maxSpeed` begrenzt. Die Limitationen der Vektoren dienen dazu, eine zu schnelle Fortbewegung zu vermeiden. Bevor der Vektor als Kraft zur Fortbewegung genutzt werden kann, muss dieser mit der Masse des Agenten `agentMass` multipliziert werden.

Der an dieser Stelle bestehende Vektor `steeringForce` wird zunächst, wie in Pseudocode 5.3 gezeigt, zur Rotation des Agenten genutzt. Die Rotation wird mittels Dreh-

```

Data : Vector3 steeringForce
Result : Fortbewegender und rotierter Agent
Function Kontext:
    steeringForce += collisionDetection();
    if ( !isCollision ) then
        | steeringForce += pathFollowing();
    end
    steeringForce = truncate( steeringForce, maxForce );
    steeringForce = truncate( agentVelocity + steeringForce, maxSpeed );
    steeringForce *= agentMass;
    torqueTowards(steeringForce);
    applyForce(steeringForce);
End

```

**Pseudocode 5.2** : Berechnung der Fortbewegung eines Agenten

moment bestimmt. Für eine stabile Rotation durch das Drehmoment und für eine ausgeglichene Balance ist bei den laufenden Agenten nur die senkrechte Achse rotierbar. Die anderen beiden Achsen sind fixiert, sodass der Agent während der Fortbewegung nicht umfallen kann. Als Erstes wird die Lage des Ziels überprüft und nur rotiert, wenn das Ziel nicht direkt vor dem Agenten ist. Ist es direkt hinter dem Agenten, wird mit der höchsten Geschwindigkeit ( $\pi$  bzw.  $180^\circ$ ) an der senkrechten Achse rotiert. Ansonsten wird die zu rotierende Achse durch das Kreuzprodukt zwischen der Blickrichtung des Agenten `agentForwardAxis` und des Ziels `target` ermittelt. Der Winkel `angle` ergibt sich aus dem Arcussinus der Länge der Rotationsachse `rotationAxis`, ist aber mindestens 1 rad. Ist das Ziel hinter dem Agenten, wird `angle` mit der Differenz zu  $\pi$  angepasst. Anschließend wird die gewünschte Schnelligkeit `desiredSpeed` zur Rotation bestimmt, indem `angle`, skaliert mit einem maximalen Drehmomentwert `maxTorque`, im Verhältnis zum Trägheitsmoment `agentInertia` radiziert wird. Die Rotationsachse `rotationAxis`, limitiert durch die Schnelligkeit `desiredSpeed`, ergibt ein angulare Geschwindigkeit, von welcher die aktuelle angulare Geschwindigkeit `agentAngularVelocity` subtrahiert wird. Diese Geschwindigkeit `deltaVelocity`, multipliziert mit `agentInertia`, ergibt das Drehmoment, das limitiert durch `maxTorque`, den Agenten physikalisch plausibel rotiert.

Nachdem der Agent sich in Richtung seines Ziels gedreht hat, wird der zuvor berechnete Vektor `steeringForce` als Kraft zur Bewegung des Agenten genutzt. Die Kraft wird nur auf zweidimensionaler Ebene angewandt, da der Agent keine höher gelege-

nen Bereiche oder Stockwerke in Gebäuden begehen kann. Die Kraft sowie das Drehmoment ist eine kontinuierliche Kraft  $F$ , die, nach der Gleichung (5.1), auf den Rigid Body als Masse  $m$ , multipliziert mit der Distanz  $d$ , im Verhältnis zum Quadrat der Zeit  $t^2$  angewendet wird.

$$F = m * d/t^2 \quad (5.1)$$

Die Berechnung findet intern in der Physik-Engine statt. Daraufhin bewegt sich der Agent dann physikalisch plausibel fort. Die Agenten bewegen sich also nicht durch die direkte Veränderung der Geschwindigkeit, sondern durch eine Kraft, die in Relation zur Masse ihre Geschwindigkeit verändert.

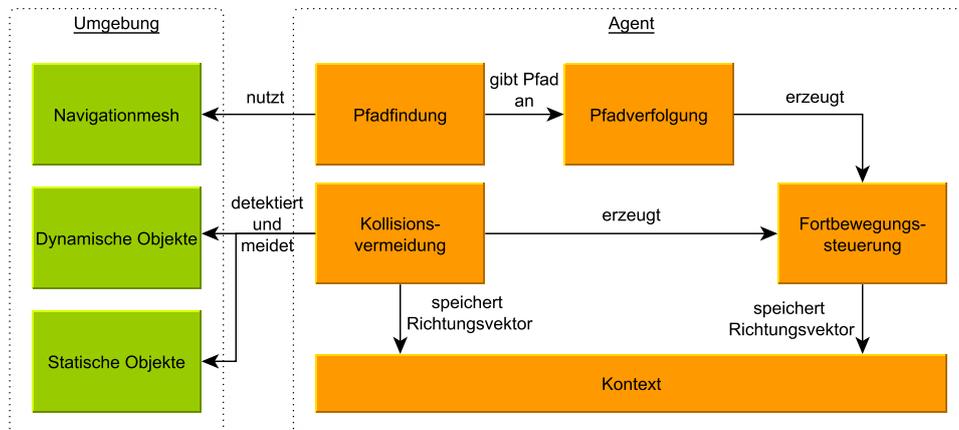
```

Data : Richtung in die rotiert wird
Result : Rotation in gegebene Richtung
Function torqueTowards( Vector3 target )
    float dot = dotProduct( agentForwardAxis, target );
    if ( 1 - dot > ε ) then
        Vector3 rotationAxis;
        float angle;
        if ( dot + 1 < ε ) then
            rotationAxis = upAxis;
            angle = π;
        else
            rotationAxis = crossProduct( agentForwardAxis, target );
            angle = asin( min( 1, rotationAxis.magnitude ) );
            rotationAxis.Normalize();
            if ( dot < 0 ) then
                angle = π - angle;
            end
        end
    end
    float desiredSpeed = sqrt( angle * maxTorque / agentInertia );
    Vector3 deltaVelocity = ( rotationAxis * desiredSpeed ) -
    agentAngularVelocity;
    if ( deltaVelocity.magnitude > ε ) then
        applyTorque( truncate( deltaVelocity * agentInertia, maxTorque ) );
    end
end
End

```

**Pseudocode 5.3** : Berechnung der Rotation eines Agenten

## Kollisionsvermeidung



**Abbildung 5.8:** Kollisionsvermeidungskomponente eines Agenten und dessen Schnittstelle zur Umgebung

Mit der Kollisionsvermeidung lässt sich eine weitere Komponente beschreiben, die das Fortbewegungsverhalten der Agenten steuert und eine neue Richtung für den Agenten bestimmt, die wieder im Kontext gespeichert wird. Diese wird, wie in Abbildung 5.8 zu sehen, für statische und dynamische Objekte berechnet. Sobald ein Objekt im Sichtkegel des Agenten ist, wird die Kollisionsvermeidung der Pfadverfolgung priorisiert. Eine Ausnahme dieser Priorisierung entsteht bei einem zu hohem Stressniveau des Agenten, was in Kapitel 5.3.3 näher erläutert wird. Die statische Kollisionsvermeidung wird in der hier entwickelten Simulation verwendet, um einen Abstand zu Wänden zu halten und diesen auszuweichen. Anhand der dynamischen Kollisionvermeidung kann ein Agent anderen Agenten, Hindernissen und Feuer ausweichen und umgehen. Zunächst wird entschieden, mit welchem Objekt eine Kollisionsvermeidung durchgeführt wird. Dazu werden die Objekte im Sichtkegel ausgewertet. Ein Agent nimmt gleichzeitig nur höchstens fünf Objekte innerhalb des Sichtfeldes wahr. Das bietet eine bessere Performance, da weniger Abfragen mit benachbarten Agenten durchgeführt werden müssen. Außerdem wird aus denen sich im Sichtfeld befindlichen Objekten das Bedrohlichste selektiert, welches das naheste Objekt ist. Für dieses Objekt wird dann die entsprechende Kollisionsvermeidung gewählt.

Die statische Kollisionsvermeidung ist ein einfaches Ausweichen ortsgebundener Objekte, indem der Agent versucht einen Abstand zu diesen Objekten zu halten (s. Pseu-

```

Data : Das auszuweichende Objekt
Result : Richtungsvektor zum Ausweichen
Function staticCollisionAvoidance( Object collisionObject )
    if ( distanceToSteerTarget < distanceToCollisionObject ) then
        | return
    end
    /* gibt den nächsten Punkt zum Kollisionsobjekt von der
       Position des Agenten */
    Vector3 closest = collisionObject.getClosestPointOnBound(agentPosition);
    Vector3 direction = agentPosition - closest;
    if ( direction.magnitude <= viewDistance ) then
        | return direction.Normalized()
    end
End

```

**Pseudocode 5.4** : Kollisionsvermeidung mit statischen Objekten

docode 5.4). Als Erstes wird der nächste Punkt `closest` von der Position des Agenten `agentPosition` zum Objekt `collisionObject` ermittelt. Anschließend wird die Strecke `direction` zwischen den Punkten bestimmt, wobei der erhaltene Vektor vom Objektpunkt zur Agentenposition zeigt. Sollte die Distanz dieser Strecke kleiner sein als ein geeigneter Schwellwert `viewDistance`, wird der normalisierte Vektor zum Ausweichen verwendet und, wie weiter oben beschrieben, als neue Richtung zur Fortbewegung genutzt.

Zur dynamischen Kollisionsvermeidung wurden die Ansätze von Sullivan (2009) und Musse u. Thalmann (1997), wie in Kapitel 2.2.3 umschrieben, leicht abgeändert und zusammengeführt verwendet. Ähnlich wie im Ansatz von Musse u. Thalmann (1997) wird zur Abfrage der Kollision zwischen zwei sich bewegendenden Agenten der Schnittpunkt von den Richtungen der Geschwindigkeiten der Agenten ermittelt. Diese Methode erinnert auch an Reynolds (2004) Ansatz der unausgerichteten Kollisionsvermeidung, bei dem eine potenzielle Kollision in der Zukunft vorhergesagt wird, um das Verhalten des Agenten zu ändern. Die Ermittlung des Schnittpunktes beschreibt in dieser Arbeit keine potenzielle Kollision, sondern eine Kreuzung der Wege zweier Agenten. Lediglich die Richtungen der Geschwindigkeiten werden in der Zukunft betrachtet. Der Pseudocode 5.5 zeigt die Implementierung des Ansatzes. Zur Berechnung des Schnittpunktes wird wie in Abbildung 5.9 zu sehen, der Schnittpunkt der Geschwindigkeiten ermittelt, indem zunächst ein Faktor `intersectionScale`, nach Gleichung (5.2) bestimmt wird.

```

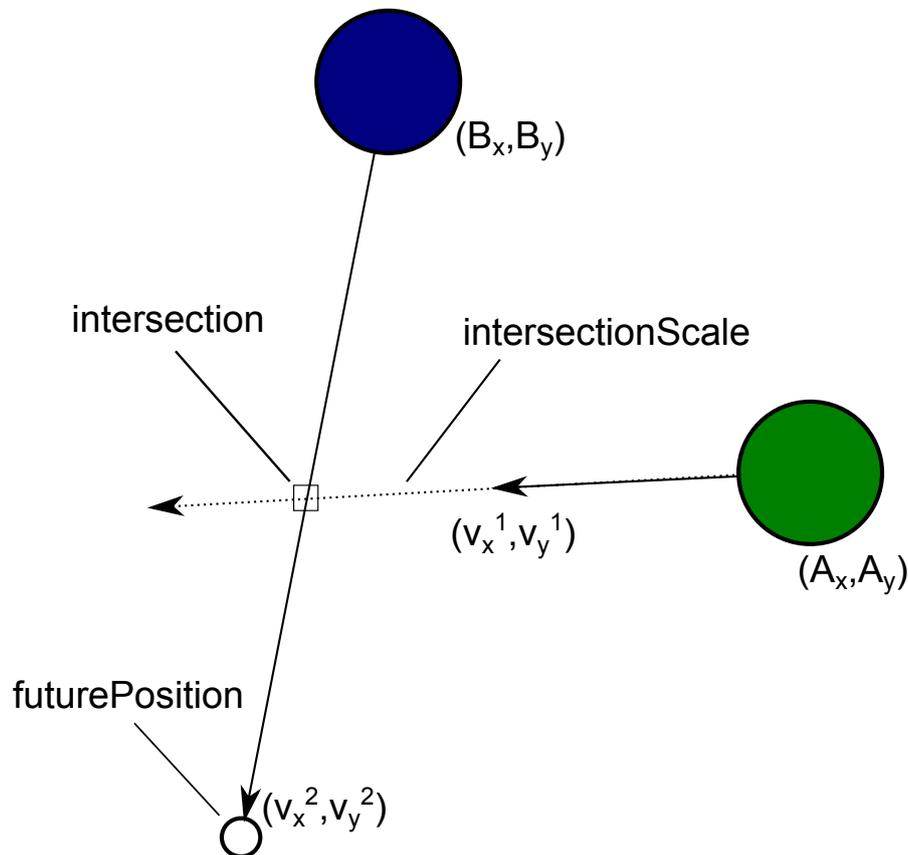
Data : Das auszuweichende Objekt
Result : Richtungsvektor zum Ausweichen
Function dynamicCollisionAvoidance( Object collisionObject )
    Vector3 direction;
    if (  $v_x^1 * v_y^2 - v_x^2 * v_y^1$  ) then
        float intersectionScale = calculateIntersectionScale();
        Vector3 intersection = agentPosition + intersectionScale * agentVelocity;
        float timeScale = (intersection - agentPosition).magnitude /
            agentVelocity.magnitude;
        Vector3 futurePosition = collisionObjectPosition + collisionObjectVelocity
            * timeScale;
        direction = agentPosition - futurePosition;
        if ( direction.magnitude > viewDistance ) then
            | return
        end
    else
        | direction = agentPosition - collisionObjectPosition;
    end
    Vector3 tangent = direction.normalized + agentVelocity.normalized;
    float distanceWeight = (viewDistance - direction.magnitude) / viewDistance;
    float orientationWeight = dotProduct( agentVelocity,
        -1 * collisionObjectVelocity ) / 2.0 + 1.5;
    if ( something is between Agent and Object to avoid ) then
        | return
    end
    if ( distanceToSteerTarget < distanceToCollisionObject ) then
        | return
    end
    Vector3 avoidanceVector = tangent * distanceWeight * orientationWeight *
        collisionObjectMass;
    return avoidanceVector.Normalize()
End

```

**Pseudocode 5.5** : Kollisionsvermeidung mit dynamischen Objekten wie Agenten

$$intersectionScale = \frac{(-A_x * v_y^2 + A_y * v_x^2 + B_x * v_y^2 - B_y * v_x^2)}{(v_y^2 * v_x^1 - v_x^2 * v_y^1)} \quad (5.2)$$

Der Schnittpunkt ist dann die Position des Agenten `agentPosition`, addiert mit der durch den Faktor `intersectionScale` skalierten Geschwindigkeit des Agenten `agentVelocity`. Als Nächstes wird die Position in der Zukunft des vermeintlich kollidierenden Hindernisses `futurePosition` ermittelt. Dazu wird ein weiterer Faktor `timeScale` benötigt, der sich aus der Länge der Differenz des Schnittpunktes `intersection` und `agentPosition`, dividiert durch die Länge von `agentVelocity` ergibt. Die `futurePosition` ist dann die Position des zu kollidierenden Objekts `collisionObjectPosition`, addiert mit der durch `timeScale` skalierten Geschwindigkeit des Objektes `collisionObjectVelocity`.



**Abbildung 5.9:** Ermittlung des Schnittpunktes der Geschwindigkeiten zweier Agenten und der Position des zu kollidierenden Agenten in der Zukunft

Für die weitere Berechnung wird ein Vektor `direction` von `futurePosition` in Richtung `agentPosition` bestimmt. Sollte der Fall auftreten, dass die Geschwindigkeitsvektoren der Agenten parallel verlaufen, genau entgegengesetzt oder aufeinander zu zeigen, existiert kein Schnittpunkt. In diesem Fall wird der Vektor `direction` von `agentPosition` zu `collisionObjectPosition` festgelegt.

Sobald die Richtung bestimmt wurde, wird die weitere Berechnung, wie in (Sullivan, 2009, S. 29ff) beschrieben, durchgeführt. Dieser Ansatz beschreibt eine Kollisionsvermeidung für dynamische Objekte, bei der die Agenten einen bestimmten Auffassungsbereich ihrer Umgebung besitzen. Zum Ausweichen wird für die Agenten im Auffassungsbereich eine Ausweichkraft berechnet, welche durch den Zusammenschluss der Verfahren von Pelechano u. a. (2007) und Lamarche u. Donikian (2004) bestimmt wird. Der Unterschied zu Sullivans Ansatz ist der verwendete Auffassungsbereich der Agenten. Sullivan verwendet ein Rechteck damit ein Agent andere Agenten erkennt. In der vorliegenden Arbeit hingegen wird ein Sichtkegel benutzt, der sich der Sicht eines Menschen genauer annähert.

Die zuvor berechnete Richtung `direction` ergibt zusammen mit der aktuellen Geschwindigkeit des Agenten `agentVelocity` eine Tangente `tangent`. Diese wird mit der Distanz `distanceWeight` und der Orientierung `orientationWeight` der Agenten sowie der Masse des ausweichenden Agenten `collisionObjectMass` gewichtet und anschließend normiert. Das ergibt den Vektor `avoidanceVector` zum Ausweichen, der anschließend zur Fortbewegung verwendet wird.

Befindet sich ein Feuer in dem Sichtbereich eines Agenten, wird die dynamische Kollisionsvermeidung mit dem Fliehverhalten und der Pfadverfolgung zusammengerechnet, um ein zielstrebigeres und schnelleres Entkommen vom Feuer zu erwirken.

### 5.3.2 Kollision

Die Kollision wird größtenteils von der Physik-Engine übernommen. Alle Objekte in der Welt, die mit einem Collider ausgestattet sind, können miteinander kollidieren. Der Collider steht unter dem Einfluss der Physik-Engine und macht ein Objekt kollisionsfähig. Objekte, die einen Collider besitzen, sind die Agenten, die Hindernisse und die Umgebung selbst wie zum Beispiel die Wände eines Raumes. Dynamische Hindernisse und alle Agenten haben zusätzlich einen Rigid Body, der bei Zusammenstößen physi-

kalisch reagiert und die realistische Bewegung durch Kraft ermöglicht. Stoßen zwei Objekte mit einem Collider und einem Rigid Body zusammen, entsteht bei den kollidierenden Objekten ein Rückstoß. Dieser Rückstoß wird durch eine Kraft beschrieben, die das eine Objekt auf das andere Objekt überträgt.

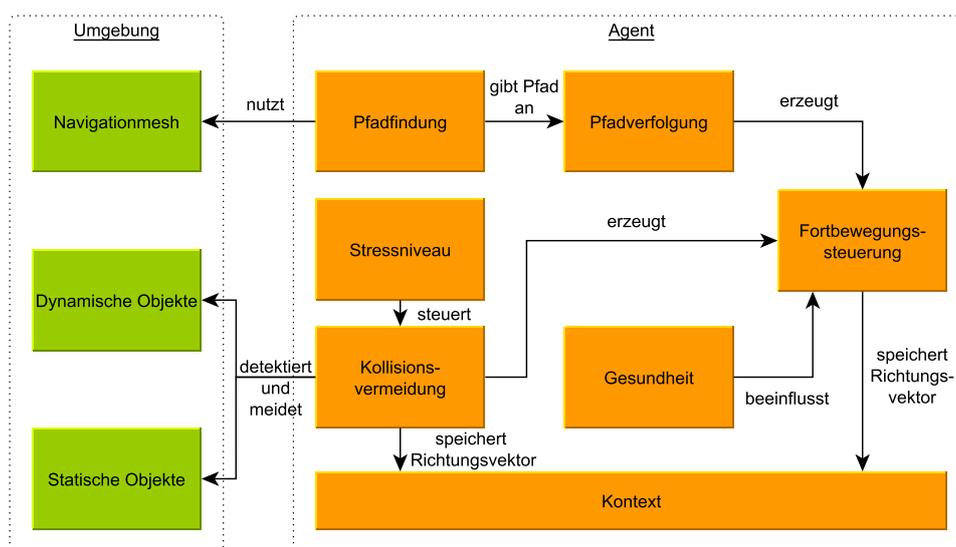
Die Agenten sind während der Navigation stabilisiert durch eine Limitation der Rotation auf der senkrechten Achse. Dadurch wird der Rückstoß bei einem Zusammenstoß deutlich gemindert. Damit die Agenten weiterhin realistisch kollidieren, wurden die Stoß- und Kollisionskraft der Newtonschen Dynamik aus Kim u. a. (2013) übernommen, um die Kräfte bei der Kollision zu verstärken. Das ermöglicht den Agenten bei einer Kollision andere Agenten wegzudrücken, aber gleichzeitig einen Rückstoß durch den Aufprall zu erleiden. Die Limitation der Achsen des Agenten wird bei einem Gesundheitswert von Null (s. Kapitel 5.3.3) aufgehoben, sodass der Agent wieder allen Kräften ausgesetzt ist und sich nicht mehr kontrollieren kann. Das führt dazu, dass der Agent umkippt und sich nicht mehr eigenständig bewegen kann. Er wird damit zum Hindernis für anderen Agenten.

Bei einer Kollision eines Agenten mit Hindernissen kann der Agent diese durch die ermittelte Kraft wegdrücken. Eine Kollision ist in der entwickelten Simulation außerdem mit dem physisch-psychischen Zustand verbunden. Kollidiert ein Agent, erhöht sich zum einen sein Stressniveau und zum anderen sinkt sein Gesundheitswert. Die Erhöhung des Stressniveaus signalisiert dem Agenten die Kollision und bedingt dessen Stress. Das Sinken des Gesundheitswertes soll mögliche Verletzungen beschreiben, die bei einem Zusammenstoß entstehen.

### **5.3.3 Physisch-psychischer Zustand**

Jeder Agent hat einen physisch-psychischen Zustand. Dieser Zustand wird in der entwickelten Simulation anhand zweier Eigenschaften des Agenten abgebildet. Ein Gesundheitswert beschreibt den physischen Zustand wie Verletzungen oder das Ableben eines Agenten. Das Stressniveau ist der Versuch, den psychischen Zustand des Agenten darzustellen und beschreibt, wie gestresst ein Agent zur gegebenen Zeit ist. Mit einer Veränderung dieser Werte durch äußere Einflüsse verändern diese Eigenschaften das Verhalten des Agenten. Durch das Stressniveau und die Gesundheit können die Kollisionsvermeidung und die Fortbewegung beeinflusst werden. In Abbildung 5.10 ist zu sehen, dass das Stressniveau die Kollisionsvermeidung steuern kann und die Ge-

gesundheit die Fortbewegung beeinflusst. Ist das Stressniveau zum Beispiel zu hoch wird der Agent rücksichtslos und ignoriert die Vermeidung von Kollisionen. Die Gesundheit beeinflusst den Agenten insoweit, dass bei einem niedrigsten Gesundheitswert die Fortbewegung des Agenten nicht mehr möglich ist, weil er umfällt.

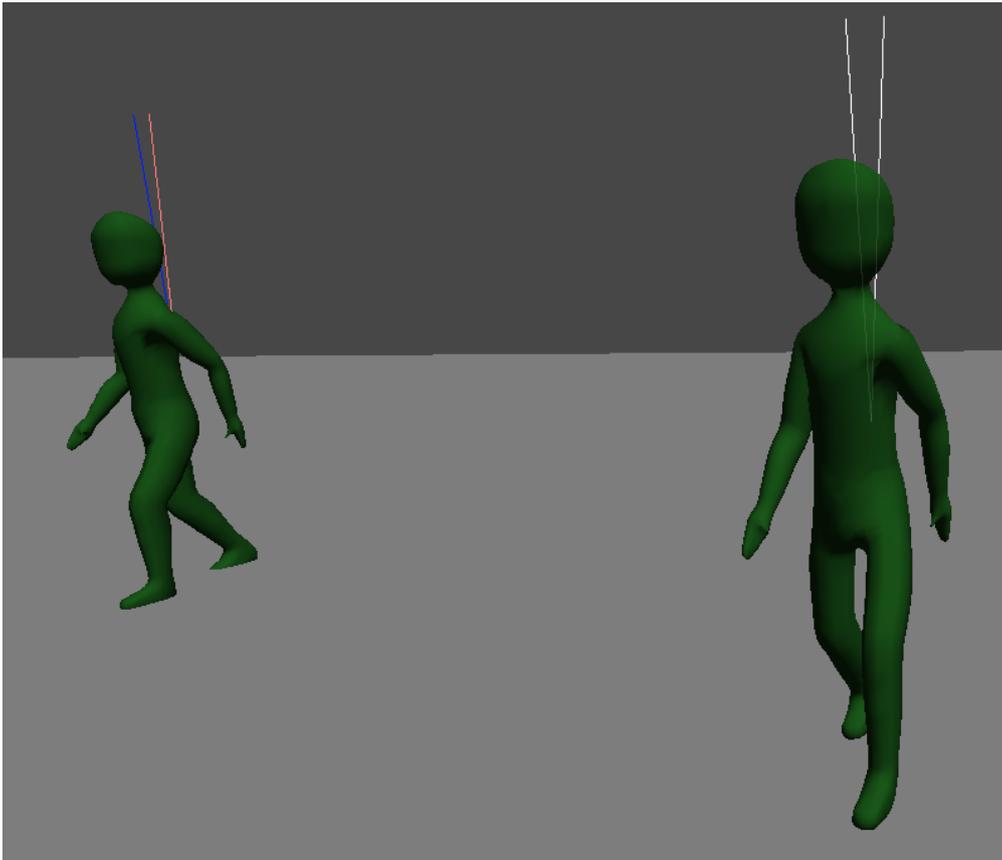


**Abbildung 5.10:** Stressniveau und Gesundheit eines Agenten und dessen Auswirkung auf das Fortbewegungsverhalten

Zur Ablese der aktuellen Zustände eines Agenten sind die Werte durch farblich codierte Stäbe, die an den Rücken des Agenten angebracht sind, visualisiert. Die Gesundheit wird in Rot dargestellt und das Stressniveau in Blau (s. Abbildung 5.11). Ein Agent hat eine Gesundheit im Wert von insgesamt 100 und ein Stressniveau im Wertebereich von 0 bis 1. Die Farbe der Stäbe wird kräftiger, je niedriger der Wert der Gesundheit und je höher er bei dem Stressniveau ist. Nachfolgend werden die Eigenschaften des Agenten detailliert beschrieben.

### Gesundheitswert

Der Gesundheitswert steht repräsentativ für die Gesundheit eines Agenten. Der Wertebereich reicht von 0 bis 100. Ein Gesundheitswert von 100 bedeutet, dass der Agent hundertprozentig gesund ist. Solange der Wert über 0 ist, kann der Agent navigieren. Im Pseudocode 5.6 sind die Bedingungen gelistet, die den Gesundheitswert `agentHealth` beeinflussen und wie sich dieser verändert.



**Abbildung 5.11:** Visualisierung der psychischen Zustände eines Agenten, links: Hohes Stressniveau und erhöhter Gesundheitswert, rechts: Beide Werte nicht erhöht

Wie oben beschrieben, ändert sich der Gesundheitswert bei Kollisionen. Kollidiert ein Agent mit einem anderen Agenten, wird der Gesundheitswert stärker verringert als bei einer Kollision mit einem Hindernis, da ein Agent einen gezielteren Stoß ausüben kann. Wird ein Agent von vier oder mehr Agenten gleichzeitig gedrückt, verringert sich der Gesundheitswert des Agenten sehr stark über die Zeit. Dieser Druck von mehreren Seiten soll die Enge in zu überfüllten Räumen beschreiben, bei denen Menschen schnell erdrückt werden können. Fällt der Wert auf 0, wird die Fixierung der Rotationsachsen gelöst, wodurch der Agent umfällt und zu einem Hindernis für andere Agenten wird. Er erfährt keine äußeren Einflüsse mehr, sondern ist den äußeren Kräften erlegen. Auch das Stressniveau spielt bei einem Gesundheitswert von 0 keine Rolle mehr. Liegt der Gesundheitswert zwischen 0 und 100, kann dieser als Verletzungsgrad eines Agenten interpretiert werden.

Der Gesundheitswert wird durch einen farblich codierten Stab am Rücken eines Agenten repräsentiert. Ein weißer Stab bedeutet volle Gesundheit. Ist der Stab rot, liegt der Gesundheitswert bei 0. Ein Wert dazwischen wird durch eine Interpolation der Farben von Weiß nach Rot dargestellt.

```
Data : float agentHealth = 100
Result : Verringerter Gesundheitswert
if ( collision with Agent ) then
  | agentHealth -= 2;
end
if ( collision with Obstacle ) then
  | agentHealth -= 0.1;
end
if ( lasting collision with Agent && Agents >= 4 ) then
  | agentHealth -= 25 * deltaTime;
end
if ( agentHealth < 0 ) then
  | Agent falls down;
else
  | Agent is able to steer;
end
```

**Pseudocode 5.6** : Auswirkungen und Berechnungen des Gesundheitswertes

### Stressniveau

Das Stressniveau beschreibt den Stress eines Agenten. Der Wertebereich des Stressniveaus reicht von 0 bis 1. Zu Beginn einer Simulation sind die Agenten nicht gestresst und haben ein Stressniveau von 0. Das Stressniveau ändert sich durch die Wirkung von unterschiedlichen Stressoren. Zusätzlich zum Stressniveau besitzt jeder Agent eine Stressresistenz, die die Berechnung des Stressniveaus beeinflusst. Die Stressresistenz bestimmt, wie schnell das Stressniveau anwächst, und soll eine Variabilität der Reaktion auf Stressoren unter den Agenten beschreiben, da auch Menschen unterschiedlich auf Stressoren reagieren.

Ein Agent besitzt eine Stressresistenz `agentStressResistance` zwischen 0.7 und 0.9, die zu Beginn der Simulation zufällig festgelegt wird und sich während der Simulation nicht verändert. Die `agentStressResistance` wird bei der Berechnung des

Stressniveaus `agentStressLevel` mit dem Wert eines Stressors multipliziert. Der Pseudocode 5.7 zeigt die verschiedenen Stressoren und welchen Einfluss sie auf das Stressniveau des Agenten haben.

Ein Stressor liegt ebenfalls im Wertebereich von 0 bis 1. Verschiedene Stressoren modifizieren das Stressniveau dabei unterschiedlich stark. Der höchste Anstieg des Stressniveaus geschieht durch das Sehen von Feuer. Befindet sich ein Feuer im Sichtkegel eines Agenten, erhöht sich das Stressniveau über die Zeit maximal. Ein weiterer Stressor ist die Verspätung eines Agenten. Zum einen wird die Distanz zum Ziel gemessen und zum anderen die Geschwindigkeit des Agenten überprüft. Verringert sich die Distanz zum Ziel nicht über die Zeit, erhöht sich das Stressniveau, ansonsten fällt der Wert des Stressniveaus wieder. Wenn der Agent langsamer als gewollt ist, erhöht sich ebenfalls das Stressniveau. Außerdem wird das Stressniveau durch eine zu hohe Dichte der Agenten größer. Sind vier oder mehr Agenten im Sichtkegel des Agenten, steigt das Stressniveau über die Zeit.

Agenten befinden sich außerdem in einem Zustand, der die Wahrnehmung eines Feuers beschreibt und ebenfalls durch einen Wert zwischen 0 und 1 abgebildet wird. Sieht ein Agent ein Feuer, wird dieser Wert sofort maximal. Wenn andere Agenten einen Agenten registrieren, der das Feuer gesehen hat, erhöht sich deren Wahrnehmung des Feuers. Ist dieser Wert bei den Agenten ebenfalls maximal, ist den Agenten bewusst, dass ein Feuer ausgebrochen ist. Außerdem erhöht sich das Stressniveau, wenn ein Agent, der ein Feuer gesehen hat, in dem Sichtkegel des Agenten ist. Das Stressniveau setzt sich dann standardmäßig auf 0.5.

Desweiteren wirken sich Kollisionen auf das Stressniveau aus. Kollidiert ein Agent mit einem anderen Agenten oder mit Hindernissen, erhöht sich das Stressniveau. Erhöht sich das Stressniveau auf über 0.97, ändert sich das Verhalten eines Agenten. Dieses Verhalten soll einer panischen Reaktion ähneln. Der Agent wird einerseits schneller und andererseits ignoriert er die Kollisionsvermeidung. Wenn die Kollisionsvermeidung ignoriert wird, kommt es zwischen den Agenten häufiger zu Kollisionen und ein Gedränge mit Drücken und Stoßen kann entstehen.

Das Stressniveau eines Agenten wird ebenfalls durch einen farblich codierten Stab, der am Rücken des Agenten befestigt ist, dargestellt. Die Farbe dieses Stabes reicht von Weiß bis Blau. Weiß bedeutet, dass der Agent ein Stressniveau von 0 hat und Blau zeigt

ein maximales Stressniveau von 1. Dazwischen wird die Farbe wie beim Gesundheitswert interpoliert, um das aktuelle Stressniveau eines Agenten ablesen zu können.

```
Data : float agentStressLevel = 0
Result : Neues Stressniveau
Function stressCalculation()
    if ( Neighbor of Agent has seen fire ) then
        | agentStressLevel += 1 * deltaTime * agentStressResistance;
    end
    if ( distanceToTarget >= distanceToOldTarget ) then
        | agentStressLevel += 0.05 * deltaTime * agentStressResistance;
    else
        | agentStressLevel -= 0.01 * deltaTime;
    end
    if ( agentVelocity < desiredVelocity ) then
        | agentStressLevel += 0.05 * deltaTime * agentStressResistance;
    end
    if ( neighborOfAgent >= 4 ) then
        | agentStressLevel += 0.05 * deltaTime * agentStressResistance;
    else
        | agentStressLevel -= 0.1 * deltaTime;
    end
    if ( agentStressLevel < 0.5 && agentHasSeenFire ) then
        | agentStressLevel = 0.5;
    end
    if ( Agent collides with other Agents or Obstacles ) then
        | agentStressLevel += 0.05 * deltaTime * agentStressResistance;
    end
    if ( agentStressLevel > 0.97 ) then
        | Agent gets faster and ignores collision avoidance;
    end
End
```

**Pseudocode 5.7 :** Auswirkungen und Berechnungen des Stressniveaus

## 5.4 Zusammenfassung

In diesem Kapitel wurde die Implementierung des Ansatzes der vorliegenden Arbeit beschrieben. Diese gliederte sich in die Umsetzung eines ersten und eines zweiten Prototyps. Mit dem ersten Prototypen wurde eine physikalische Repräsentation eines Agenten durch mehrere Rigid Bodies erforscht und inwieweit sich eine Balance dieser erzielen lässt. Der zweite Prototyp zeigte die Implementierung einer Menschenmengen-Simulation mit dem Fokus auf der Umsetzung der Agenten. Dazu wurden Techniken zur Navigation wie der Pfadfindung, der Pfadverfolgung, der Fortbewegungskraft und der Kollisionsvermeidung präsentiert. Außerdem wurde die Kollisionsfähigkeit der Agenten beschrieben. Der physisch-psychische Zustand wurde aufgeteilt in einen Gesundheitswert, der den physischen und in ein Stressniveau, das den psychischen Zustand eines Agenten abbildet. Anschließend wurde die entwickelte Simulation im nächsten Kapitel anhand von Testszenarien evaluiert und durch eine Extrapolation der physikalischen Repräsentation und dem Vergleich der Simulation mit anderen Ansätzen weiter bewertet.

## **6 Analyse und Diskussion**

Die Analyse und Bewertung der in der vorliegenden Arbeit entwickelten Simulation findet anhand von unterschiedlichen Testszenerarien statt. Diese durchgeführten Szenarien sollen Aufschluss über die Besonderheiten und den Nutzen der Echtzeitphysik und der Verhaltens-KI geben. Außerdem werden die untersuchten Agenten beider Prototypen dieser Arbeit in Relation zueinander gesetzt und eine Extrapolation derselben durchgeführt. Abschließend werden die erzielten Ergebnisse mit den Ergebnissen anderer Ansätze einer Menschenmengen-Simulation verglichen.

### **6.1 Testszenerarien**

Zur Analyse der entwickelten Simulation wurden sechs verschiedene Testszenerarien aufgestellt und durchgeführt. Jedes Szenario umfasst den Testaufbau, die Erwartungen an das Szenario, die gemachten Beobachtungen und eine anschließende Bewertung. Es wurden für jedes Szenario fünf Simulationen durchgeführt, dokumentiert und ausgewertet. Den Simulationen wurden pseudozufällige Werte übergeben wie die Position der Agenten und deren Stressresistenz. Die Anzahl der Agenten variiert von 30 bis 60 Agenten, aufgeteilt in 1 bis 3 Gruppen pro Simulation. Vorab ist zu sagen, dass die Anzahl der Agenten gering gehalten wurde, da die Bilder pro Sekunde bei mehr als 60 Agenten auf unter 25 gefallen sind. Dieser Nachteil entsteht, da zum einen eine große Menge an physikalischen Objekten simuliert wird und zum anderen die physikalischen Berechnungen in einer konstanten Schleife bestimmt werden, die öfter als einmal pro Frame aufgerufen wird. Aus diesem Grund wurde für die geringe Anzahl an Agenten die Umgebung verkleinert, um in bestimmten Situationen eine Überfüllung herbeizuführen. Außerdem ändern sich die Zustände und das Verhalten der Agenten deutlich schneller als bei einem Menschen in der Realität. Mit den sechs Szenarien werden die Phänomene des Ausweichens, des Drückens und Stoßens, des Erdrückens und des Stressniveaus der Agenten untersucht.

#### **6.1.1 Szenario 1 - Ausweichen im Tunnel**

##### **Testaufbau**

Das erste Szenario zeigt eine Simulation in einem Tunnel. Der Tunnel ist durch Wände auf beiden Seiten abgegrenzt und hat jeweils einen Ein- beziehungsweise Ausgang links und rechts. Insgesamt sind 60 Agenten, gleichmäßig aufgeteilt in zwei Gruppen,

an der Simulation beteiligt. Eine Gruppe (Grün) kommt vom linken Eingang und versucht den Ausgang auf der anderen Seite zu erreichen. Die andere Gruppe (Rot) kommt von rechts und versucht den Ausgang auf der linken Seite zu erreichen. Beide Gruppe laufen sich also frontal entgegen. Die Szene ist so aufgebaut, dass die Agenten ein Ziel vor dem Ausgang ansteuern. Der Aufbau ist in Abbildung 6.1 zu sehen.

### **Erwartung**

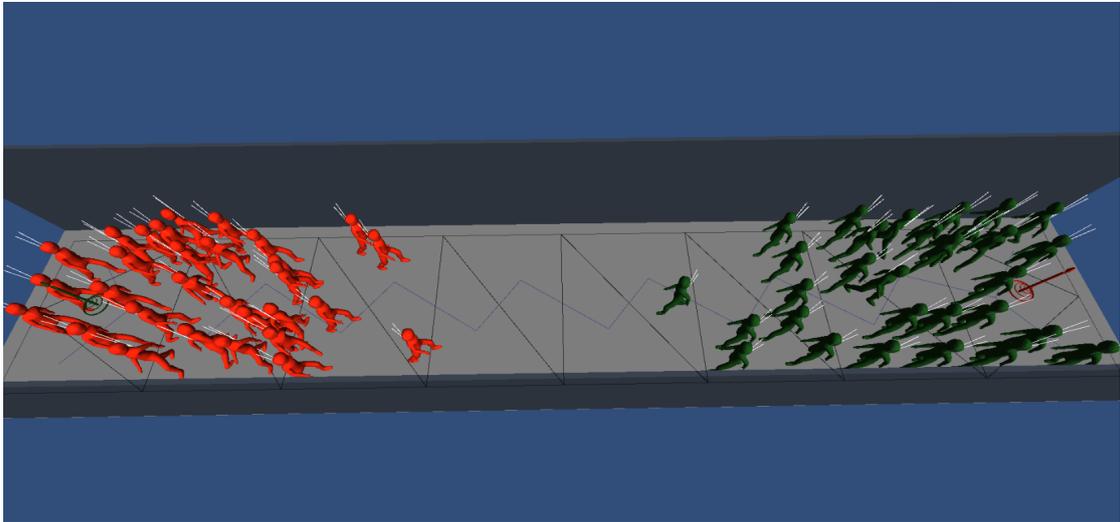
Bei diesem Aufbau der Szene sollte jede Gruppe ihr Ziel erreichen. Gruppe Rot und Grün sollten jeweils zum Ausgang auf der anderen Seite gelangen. Die Agenten sollten einen frontalen Zusammenstoß vermeiden und sich möglichst erfolgreich ausweichen. In diesem Szenario sollte es zu keinem Umkippen der Agenten kommen. Außerdem sollten die Agenten gar nicht bis leicht gestresst sein.

### **Beobachtung**

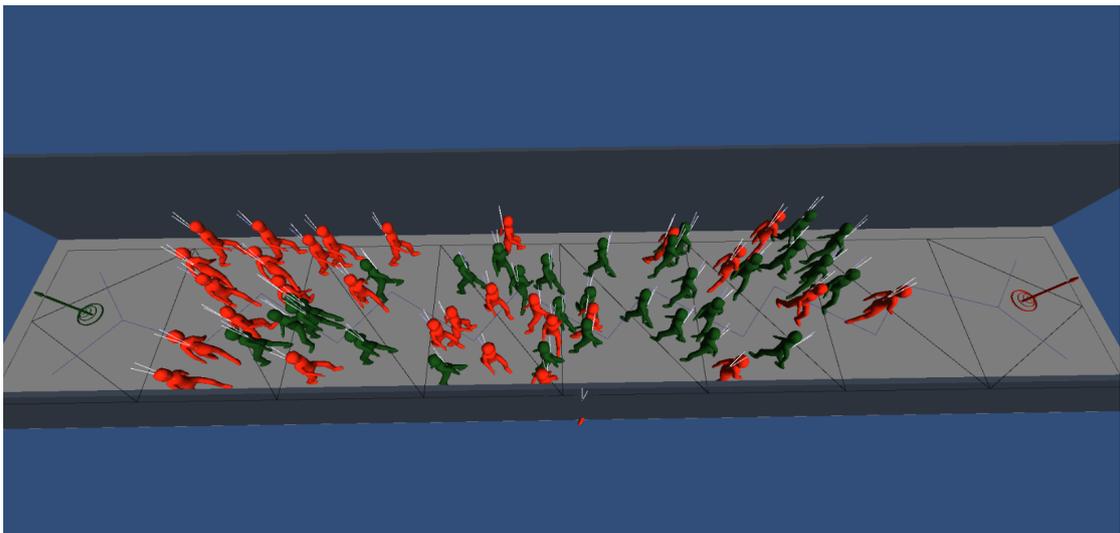
Die Beobachtungen zum ersten Szenario ergaben, dass alle Durchläufe ähnlich verlaufen sind. Jede Gruppe konnte in den fünf Simulationen ihr Ziel erreichen und gelangte auf die andere Seite des Tunnels. Die Agenten weichen sich erfolgreich aus, wobei es in manchen Situationen zu leichten Kollisionen kommt. Bewegt sich ein Agent sehr schnell fort, kommt es dazu, dass dieser erst spät ausweicht. Im vierten Durchlauf ist ein frontaler Zusammenstoß zu beobachten, welcher sich später aber wieder auflöst. Das Stressniveau der Agenten steigt ein wenig an, wenn die Gruppen aufeinander treffen und die Agenten sich auszuweichen versuchen. Die Agenten sammeln sich am Ausgang, wo sich der Stress deutlich erhöht, da sie versuchen in die Mitte des Ziels zu gelangen. Der Gesundheitswert verringert sich bei den Agenten erst am Ende der Simulation, während sie versuchen das Ziel in der Mitte zu erreichen (s. Abbildung 6.2). Außerdem fallen auch vereinzelt Agenten um, was besonders im ersten Durchlauf zu sehen ist.

### **Bewertung**

Da alle Agenten ihr Ziel erreichen und ein deutliches Ausweichen zu erkennen ist, lässt sich das Fortbewegungsverhalten der Agenten positiv bewerten. Die leichten Kollisionen unter den Agenten sind bei einer höheren Dichte einer Menschenmenge durchaus möglich. Sie sind zum Beispiel zu vergleichen mit Menschen, die mit den Schultern aneinander stoßen, vor allem wenn einer der Beteiligten sich zu schnell fortbewegt.



(a) Ausgangssituation

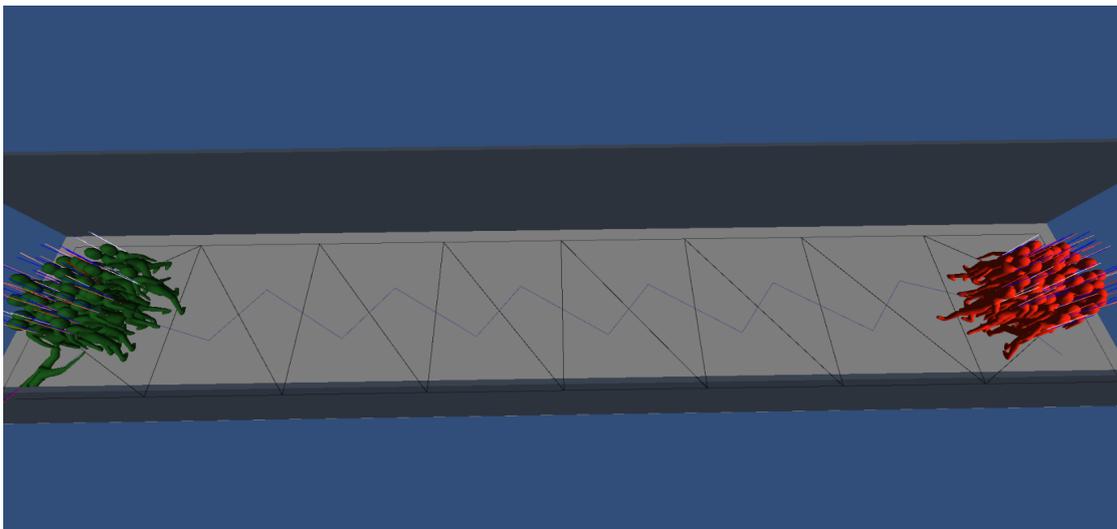


(b) Agenten weichen sich aus

**Abbildung 6.1:** Szenario 1: Zwei Gruppen von Agenten versuchen ein Ziel auf der anderen Seite eines Tunnels zu erreichen, während sie sich gegenseitig ausweichen

Ein frontaler Zusammenstoß ist in der Realität eher selten aufzufinden und ist in den fünf Durchläufen auch nur einmal zu beobachten. Dennoch kann ein solches Phänomen auftreten, wenn zum Beispiel in einer dichten Menschenmenge zwei Menschen in Gedanken sind und nicht nach vorne schauen.

Die leichte Erhöhung des Stressniveaus während des Ausweichens ist dadurch bedingt, dass eine große Gruppe von Agenten, aus der Sicht eines Agenten, direkt auf diesen zukommt. Zudem erhöht das leichte Aneinanderstoßen das Stressniveau zusätzlich. Die Erhöhung des Stressniveaus am Ende der Simulation ergibt sich aus der hohen Anzahl von Agenten und des nicht vollständigem Erreichen des Ziels. Dadurch steigt das Stressniveau so hoch, dass die Agenten in einen Panik ähnlichen Zustand geraten und ohne Rücksicht versuchen das Ziel zu erreichen. Das führt dazu, dass die Agenten drängen und sich schupsen, wodurch der Gesundheitswert fällt und sie sich gegenseitig verletzen. Einige werden dadurch teilweise erdrückt und fallen um.



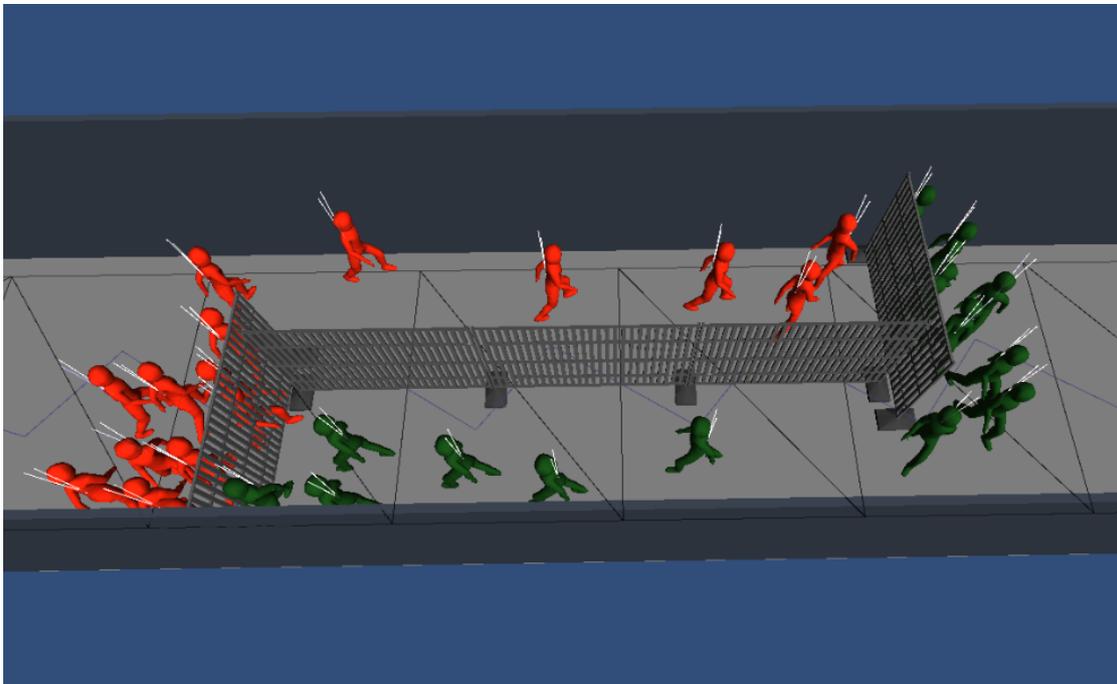
**Abbildung 6.2:** Szenario 1: Beide Gruppen von Agenten haben ihr Ziel erreicht und drängeln und drücken sich.

### 6.1.2 Szenario 2 - Gedränge im Tunnel

#### Testaufbau

Das zweite Szenario spielt sich im gleichen Tunnel wie im ersten Szenario ab. Hierbei versuchen wiederum zwei Gruppen jeweils das Ziel auf der anderen Seite zu erreichen.

Der Unterschied zum ersten Szenario ist die Anzahl der Agenten und der weitere Aufbau der Szene. Insgesamt sind 30 Agenten beteiligt, die wiederum gleichmäßig auf die beiden Gruppen Rot und Grün aufgeteilt sind. Zusätzlich befinden sich im Tunnel Absperrzäune als Hindernisse (s. Abbildung 6.3), die das Durchkommen für die Agenten erschweren. Ein Absperrzaun ist ein Rigid Body mit Collider, der frei im Raum bewegbar ist. Die Agenten können mit diesem zusammenstoßen und ihn wegschieben. In der Mitte sind die Absperrzäune mit Rigid Body Joints verbunden, aber sie können aus der Verankerung durch genügend Kraft gelöst werden. Das Ziel der Agenten ändert sich mit der Zeit nicht, weswegen sie auf jeden Fall versuchen ihr vorgegebenes Ziel zu erreichen.



**Abbildung 6.3:** Szenario 2: Zwei Gruppen von Agenten versuchen sich einen Weg durch einen eingezäunten Tunnel zu erarbeiten.

### **Erwartung**

Zu erwarten ist, dass die Agenten auf Grund der Hindernisse ihr Ziel mit Verzögerung erreichen. Sie sollten versuchen den Absperrzäunen und sich untereinander auszuweichen. Das Stressniveau der Agenten wird sich erhöhen, da in der vorgegebenen Zeit das

Ziel nicht erreicht werden kann, weil es versperrt ist. Das sollte dazu führen, dass die Agenten die Absperrung wegdrücken und sich somit den Weg frei machen. Es könnte passieren, dass Agenten wegen zu viel Druck durch andere Agenten umfallen. Alle Agenten, die nicht umfallen, sollten das Ziel auf der anderen Seite erreichen.

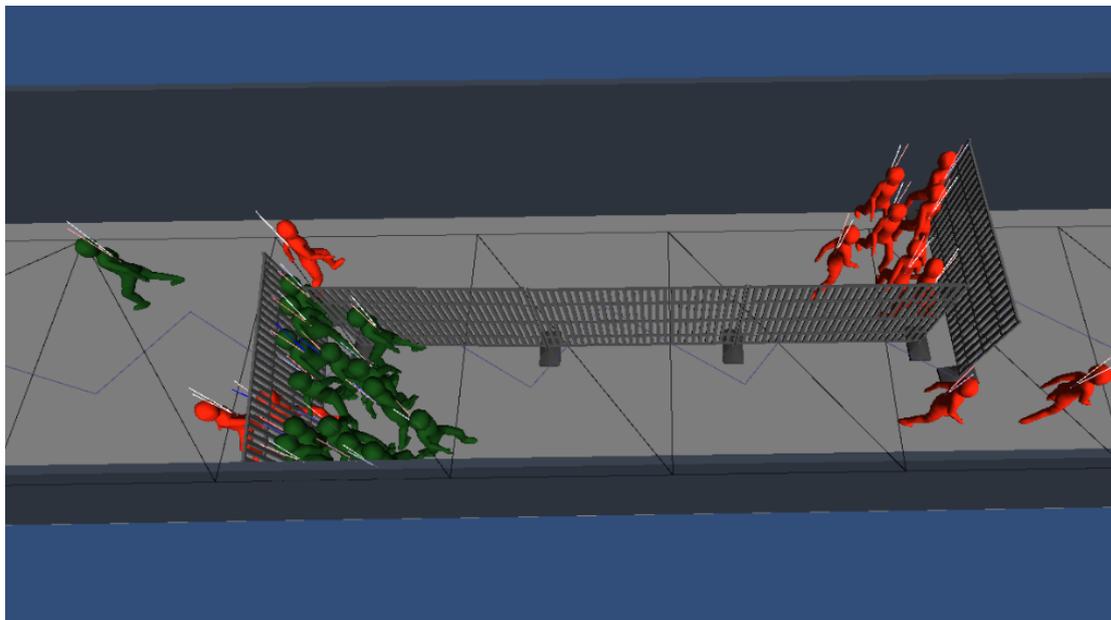
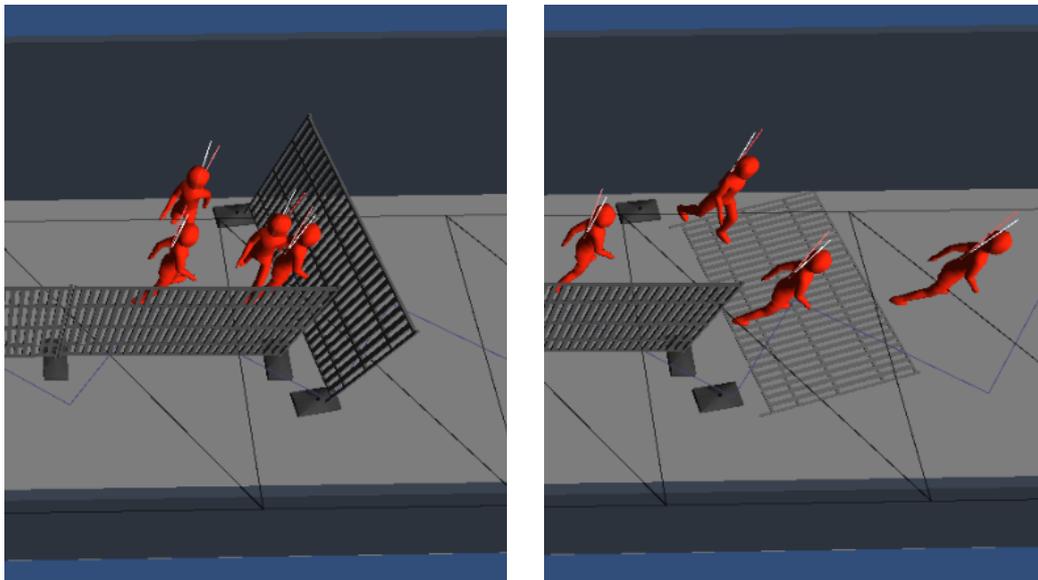


Abbildung 6.4: Szenario 2: Agenten schieben die Absperrzäune beiseite

### Beobachtung

Die Beobachtungen der fünf Durchläufe haben ergeben, dass nur ein Teil der Agenten das Ziel erreicht. Einige Agenten fallen um, andere erkennen den durch andere freigemachten Weg nicht. Die Agenten bewegen sich von ihrem Startpunkt auf die Zäune zu und weichen diesen eine Zeit lang aus. Mit der Zeit erhöht sich das Stressniveau der Agenten vor den Zäunen. Hat das Stressniveau ein Maximum erreicht, ignorieren die Agenten das Ausweichen und beginnen damit, den Zaun wegzudrücken. Das wird dadurch erschwert, dass Agenten der anderen Gruppe von der anderen Seite ebenfalls gegen den Zaun drücken. Die Agenten schaffen es trotzdem, sich den Weg freizudrücken, indem der Zaun leicht zur Seite geschoben wird, und gelangen an ihr Ziel (s. Abbildung 6.4). Im zweiten Durchlauf ist sogar zu beobachten, dass ein Zaun komplett umgestoßen und aus der Verankerung gerissen wird, wie in Abbildung 6.5 zu sehen. Dadurch haben die Agenten der roten Gruppe den kompletten Weg frei zum

Ziel. Sobald der Weg frei ist, sinkt das Stressniveau der Agenten wieder. Während des Drückens der Absperrzäune fällt der Gesundheitswert der Agenten nur leicht. Vereinzelt fallen Agenten um, da sie von mehreren Seiten von anderen Agenten angestoßen und gedrückt werden.



(a) Umfallender Zaun

(b) Agenten laufen über den Zaun zum Ziel

**Abbildung 6.5:** Szenario 2: Agenten stoßen einen Absperrzaun um

### **Bewertung**

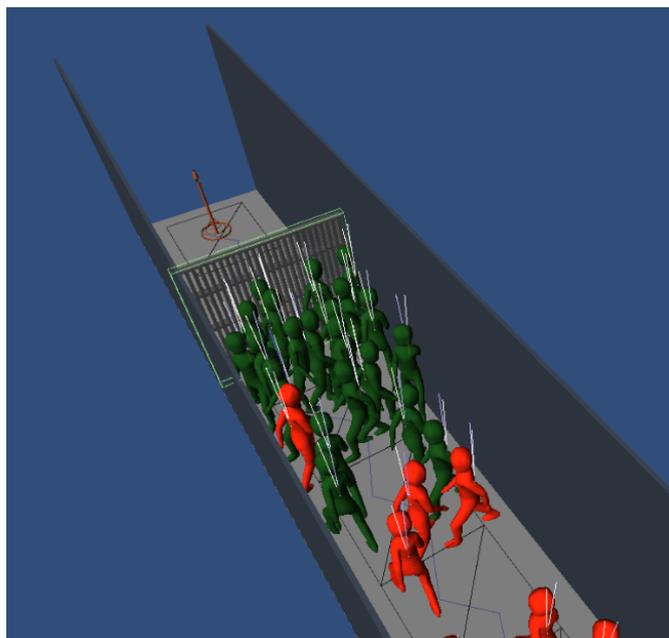
Die Agenten verhalten sich während der Durchläufe zunächst erwartungsgemäß, indem sie den Zäunen ausweichen, da sie versuchen auf die andere Seite zu gelangen. Mit der Zeit ändert sich das Verhalten, da die Agenten unbedingt ihr Ziel erreichen wollen, diesem aber nicht näher kommen. Man kann davon ausgehen, dass der Tunnel zum Eingang jeder Gruppe ebenfalls versperrt ist und ein Durchkommen der Gruppen jeweils nur auf der anderen Seite möglich ist. Das Stressniveau der Agenten erhöht sich kontinuierlich und deutlich bis zum Maximum, wodurch die Agenten in ein Panik ähnliches Verhalten geraten. Das führt dazu, dass sie anfangen die Absperrzäune wegzudrücken, um mit aller Kraft das Ziel zu erreichen. Dabei kann es passieren, dass sie sich Verletzungen zuziehen, die in dieser Situation aber für die Agenten unbedeutender sind als das Ziel zu erreichen. Sobald sie dem Ziel wieder näher kommen, sinkt das Stressniveau, da der Weg zum Ziel wieder frei ist. Beim Versuch, die Zäune zur

Seite zu schieben, schubsen und drücken sich die Agenten gegenseitig, was dazu führt, dass Agenten sich verletzen oder sogar ohnmächtig werden und umfallen. Das Schieben und das Umfallen des Absperrzauns zeigt, wie ein Agent als Rigid Body mit einem Hindernis als Rigid Body interagiert. Die Agenten können eine Kraft auf die Absperrzäune auswirken, um diese zur Seite zu bewegen und sogar zum Umfallen zu bringen.

### 6.1.3 Szenario 3 - Erdrücken im Tunnel

#### Testaufbau

Dieses Szenario zeigt wieder einen Tunnel, der durch Wände begrenzt ist und jeweils einen Ein- und einen Ausgang hat. Dieser Tunnel ist allerdings um die Hälfte schmaler als der Tunnel aus den ersten beiden Szenarien. Es wurden in dieser Szene 44 Agenten gleichmäßig auf 2 Gruppen aufgeteilt. Beide Gruppen versuchen von derselben Seite das Ziel auf der anderen Seite zu erreichen. Allerdings ist der Tunnel durch ein fest verschlossenes Tor versperrt, das die Agenten nicht öffnen können. Dadurch ist das Erreichen des Ziels für die Agenten unmöglich (s. Abbildung 6.6).



**Abbildung 6.6:** Szenario 3: Agenten versuchen vergeblich ein verschlossenes Tor zu passieren und sammeln sich davor

Ein solches Szenario ist vergleichbar mit Katastrophen in Fußballstadien, wie sie sich 1988 in Katmandu und 1968 in Buenos Aires ereigneten. Dabei sind, wie in Sueddeutsche.de (2012) beschrieben, Fußballfans in Katmandu wegen eines Hagelsturms und in Buenos Aires wegen einer brennenden Zeitung zu den Ausgängen geflüchtet, die zu dieser Zeit leider verschlossen waren. In Buenos Aires kamen 74 Menschen ums Leben und 150 Personen wurden verletzt, in Katmandu starben laut Pressemitteilungen 93 Menschen. Wie solche Ereignisse in Zukunft bereits im Vorfeld vermieden werden können, soll durch die entwickelte Simulation dieses Szenarios aufgezeigt werden.

### **Erwartung**

Die Agenten werden in den Simulationen dieses Szenarios ihr Ziel niemals erreichen, da der Weg absichtlich versperrt ist. Zunächst sollten sich die Agenten vor dem verschlossenen Tor sammeln und versuchen mit der Zeit das Tor aufzudrücken. Agenten, die von weiter hinten kommen, sollten mit der Zeit damit beginnen, ebenfalls zu drängen und die Agenten weiter vorne zu drücken. Als Folge werden die Agenten direkt vor dem Tor so stark bedrängt, dass sie erdrückt werden und anschließend umfallen.

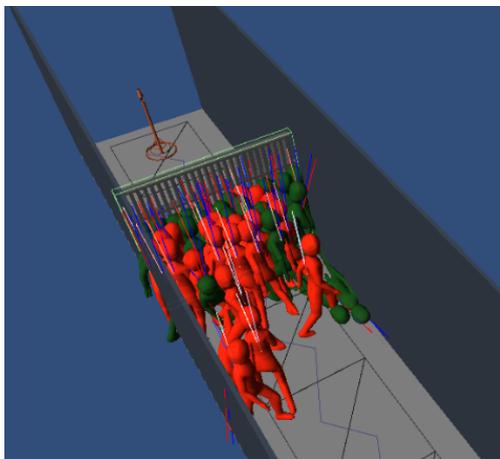
### **Beobachtung**

Wie erwartet, erreichen die Agenten ihr Ziel nicht, da das Tor zwischen den Agenten und dem Ziel nicht zu öffnen ist. Allerdings versuchen die Agenten mit der Zeit, ähnlich wie in Szenario 2 (s. Kapitel 6.1.2), das Hindernis umzustößen. Auf Grund des fest verankerten Tors gelingt es ihnen aber nicht. Der Versuch des Umstoßens ereignet sich erst bei maximalem Stressniveau. Das Stressniveau steigt bei den Agenten sehr schnell an. Die Agenten vorne drücken gegen das Tor und die Agenten weiter hinten drücken mit der Zeit gegen die Agenten im vorderen Bereich. Dadurch fallen einige Agenten weiter vorne um.

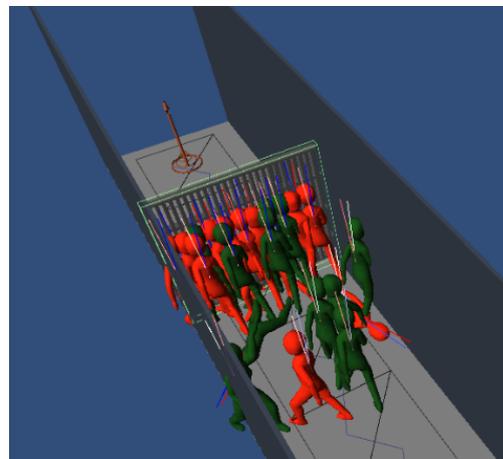
Der fünfte Durchlauf wurde mit nur 22 Agenten durchgeführt. Abgesehen davon lassen sich die gleichen Phänomene beobachten wie in den anderen Simulationen des Szenarios. Im fünften Durchlauf fallen jedoch proportional weniger Agenten um, dafür ist ein stark verringerter Gesundheitswert bei den übrigen Agenten zu erfassen.

## Bewertung

Die Agenten versuchen in dieser Situation in Sicherheit zu gelangen. Der einzige Weg um geschützt zu sein, befindet sich allerdings hinter dem versperrten Tor. Die vorderen Agenten werden gestresster, da der Weg versperrt ist und sie ihr Ziel nicht erreichen. Sie geraten in einen Panik ähnlichen Zustand und versuchen das verschlossene Tor aufzudrücken. Bei den hinteren Agenten erhöht sich das Stressniveau, weil vor ihnen eine große Menge von Agenten ist, die sich nicht weiter Richtung Ziel fortbewegt. Übertragen auf die Realität würden die Agenten wahrscheinlich nicht wissen, warum es vorne nicht weitergeht und versuchen deshalb, weiter voranzukommen, indem sie gegen die Agenten vor ihnen drücken. Das führt dazu, dass der Druck vorne zu hoch ist und einige Agenten erdrückt werden, weswegen diese dann wahrscheinlich tot oder ohnmächtig umfallen (s. Abbildung 6.7). Auch bei der Simulation mit weniger Agenten zeigt sich ein ähnliches Verhalten der Agenten, jedoch gibt es hierbei weniger umgefallene Agenten, aber dafür mehr Verletzte. Dieses Szenario bildet die Katastrophen in Fußballstadien relativ ähnlich ab. Eine große Menschenmenge versucht durch einen versperrten Ausgang in Sicherheit zu gelangen, aber erreicht diesen nicht. Die hinten stehenden Menschen erzeugen einen so hohen Druck, dass die Menschen vorne gegen das Tor gedrückt werden und sterben.



(a) Simulation mit 44 Agenten



(b) Simulation mit 22 Agenten

**Abbildung 6.7:** Szenario 3: Agenten fallen im vorderen Bereich um, da Agenten von hinten gegen die Menge drücken

#### **6.1.4 Szenario 4 - Der Loveparade-Tunnel**

##### **Testaufbau**

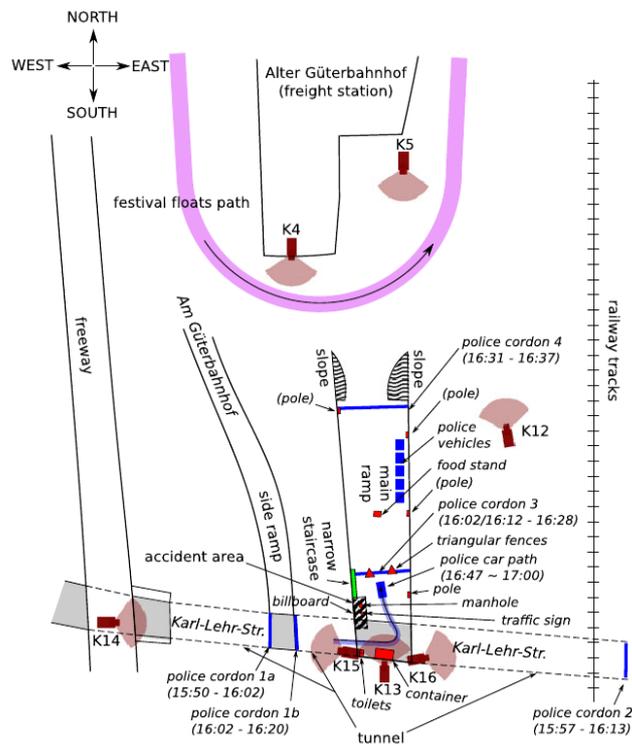
Das vierte Szenario ist ein Vergleich zur Tragödie der Loveparade 2010 in Duisburg, wie in Kapitel 1 beschrieben. Es beschreibt einen Tunnel, der sich, wie in Abbildung 6.8 gezeigt, in drei Ein- beziehungsweise Ausgänge unterteilt. Die Menschen versuchten bei der Veranstaltung zum einen auf das Gelände zu gelangen und zum anderen das Gelände zu verlassen. Dies gelang ihnen nur durch den aufgezeigten Tunnel. In den fünf Simulationen zu diesem Szenario gibt es drei Gruppen mit je 20 Agenten. Aus jeder der drei Richtungen kommt eine Gruppe, deren Ziele sich auf die jeweils anderen Ein- und Ausgänge aufteilen. Die violette Gruppe versucht teilweise nach links und teilweise nach rechts zu kommen. Da das Szenario mit nur 60 Agenten durchgeführt wurde, ist der Tunnel deutlich schmaler modelliert worden, um ein Verhältnis zu einer größeren Menschenmenge zu erzielen. Zudem sind die Gruppen zu Beginn der Simulation sehr nah in der Mitte, um ein eventuelles Gedränge besser abzubilden.

##### **Erwartung**

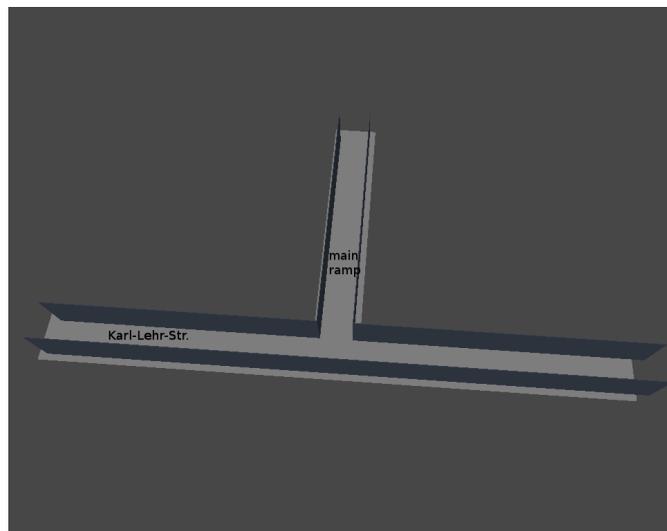
Zu erwarten ist, dass es zu einer Anhäufung der Agenten in der Mitte kommt und das Durchkommen zum Ziel erschwert wird. Die Agenten sollten größtenteils ihr Ziel nicht erreichen, da es zu einem Stoppen der Bewegung kommt und der Druck der Agenten andere Agenten zum Umfallen bringen kann. Umgefallene Agenten erschweren oder versperren dann anderen Agenten das weitere Durchkommen. Der Gesundheitswert der Agenten sollte sich insgesamt stark verringern und das Stressniveau sollte bei vielen Agenten das Maximum erreichen.

##### **Beobachtung**

Die fünf Simulationen ergaben, dass sich die Agenten wenig ausgewichen sind und es zu vielen Kollisionen kam. Besonders ist eine Anhäufung der Agenten in der Mitte des Tunnels, wie in Abbildung 6.9 gezeigt, zu beobachten. Es ist deutlich zu erkennen, dass das Stressniveau der Agenten sehr hoch ist und sich kaum verringert. Einige Agenten finden einen Weg durch die Menge und erreichen ihr Ziel. Im ersten Durchlauf fallen keine Agenten in der Mitte um und es gibt nur wenige Agenten, bei denen der Gesundheitswert gesunken ist. Die anderen Durchläufe zeigen, dass der Gesundheitswert bei vielen Agenten abfällt und einige Agenten vor allem in der Mitte umfallen. Die um-



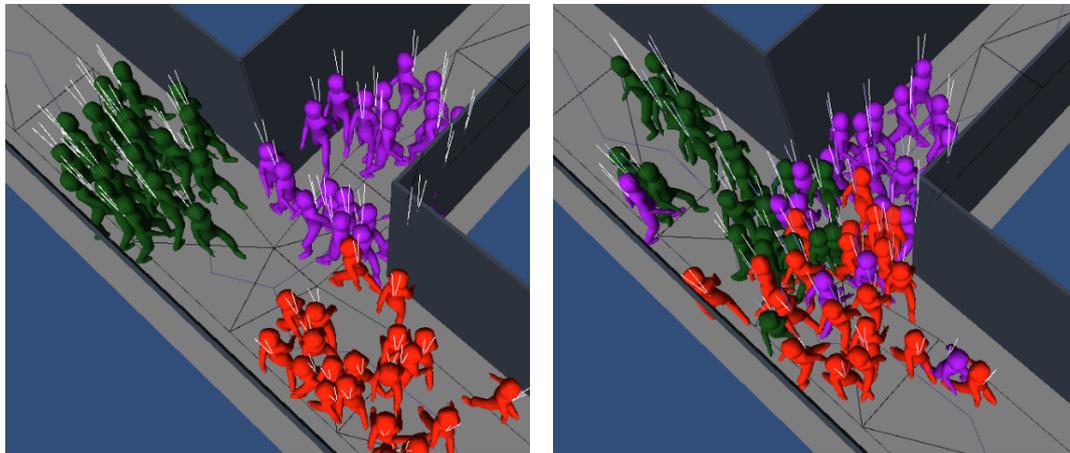
(a) Grundriss des realen Szenarios der Loveparade 2010. Bildquelle: Helbing u. Mukerji (2012)



(b) Aufbau des simulierten Szenarios

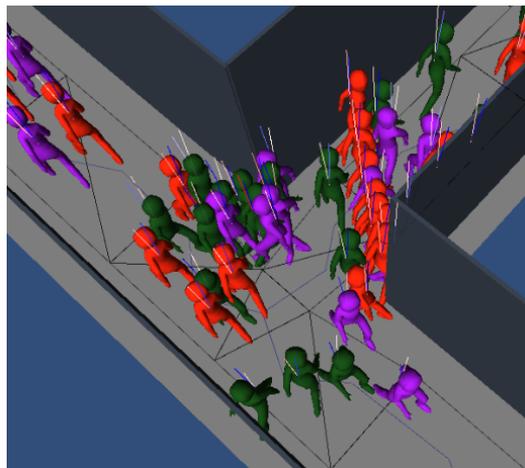
Abbildung 6.8: Szenario 4: Vergleich zwischen dem Grundriss der Loveparade 2010 und dem Aufbau des durchgeführten Testszenarios

gefallenen Agenten hindern zum Beispiel im vierten Durchlauf die anderen Agenten daran, ihr Ziel zu erreichen.



(a) Ausgangssituation

(b) Ansammlung und Gedränge



(c) Auflösung des Gedränges

**Abbildung 6.9:** Szenario 4: Große Ansammlung und Gedränge von Agenten in der Mitte der Kreuzung

### Bewertung

Die Agenten wurden absichtlich beim Start der Simulation näher zur Mitte hin platziert, um eine hohe Dichte der Menschenmenge zu erzwingen. Da viele Agenten sich aufeinander zubewegen und jeder Agent eine große Menge von Agenten vor sich sieht, erhöht sich deren Stressniveau sehr schnell und sehr stark. Das führt dazu, dass

ein Panik ähnlicher Zustand ausgelöst wird, bei dem das Ausweichen der Agenten größtenteils ignoriert wird. Dadurch kommt es zu mehr Kollisionen und Gedränge in den Simulationen und eine Anhäufung der Agenten in der Mitte entsteht. Durch die vielen Zusammenstöße und anhaltenden Kollisionen sinkt der Gesundheitswert der Agenten, was wiederum als Verletzungen dieser interpretiert werden kann. Der Druck, der bei dem Gedränge entsteht und auf die Agenten wirkt, ist teilweise so stark, dass einige Agenten ohnmächtig werden und umfallen. Da die Simulationen nur mit 60 Agenten durchgeführt wurden, sind das Umfallen der Agenten und das Stoppen des Bewegungsflusses nicht sehr deutlich zu erkennen. Außerdem löst sich die Anhäufung mit der Zeit auf und viele Agenten erreichen ihr Ziel. Dennoch ist zu erkennen, dass es schon bei einer kleinen Menge von Agenten zu großen Einbußen in der Koordination, der Gelassenheit und der Gesundheit der Agenten kommt.

### **6.1.5 Szenario 5 - Der große Platz**

#### **Testaufbau**

In diesem Szenario soll das Fortbewegungsverhalten der Agenten untersucht werden. Es gibt 60 individuelle Agenten in einer großen Gruppe, die sich auf einer großen Fläche frei umher bewegen (s. Abbildung 6.10). Dabei hat jeder Agent ein eigenes privates Ziel, das er versucht zu erreichen. Sobald der Agent sein Ziel erreicht hat, wartet dieser und sucht sich ein zufälliges neues Ziel. Dieses Szenario ist vergleichbar mit Fußgängern, die auf einem großen Platz herumlaufen und sich zum Beispiel Sehenswürdigkeiten anschauen oder nach etwas suchen.

#### **Erwartung**

In den fünf Simulationen zu diesem Szenario sollte ein gleichmäßiger Bewegungsfluss der Agenten erkennbar sein. Dabei sollten die fortbewegenden Agenten sich konsequent ausweichen und wartende Agenten ebenfalls umgehen. Das Stressniveau der Agenten sollte sich in diesem Szenario deutlich gering halten, da die Agenten selten und wenigen Stressoren ausgesetzt sind. Auch der Gesundheitswert sollte sich nicht verändern, da Kollisionen konsequent vermieden werden. Entsprechend werden auch keine Agenten umfallen.

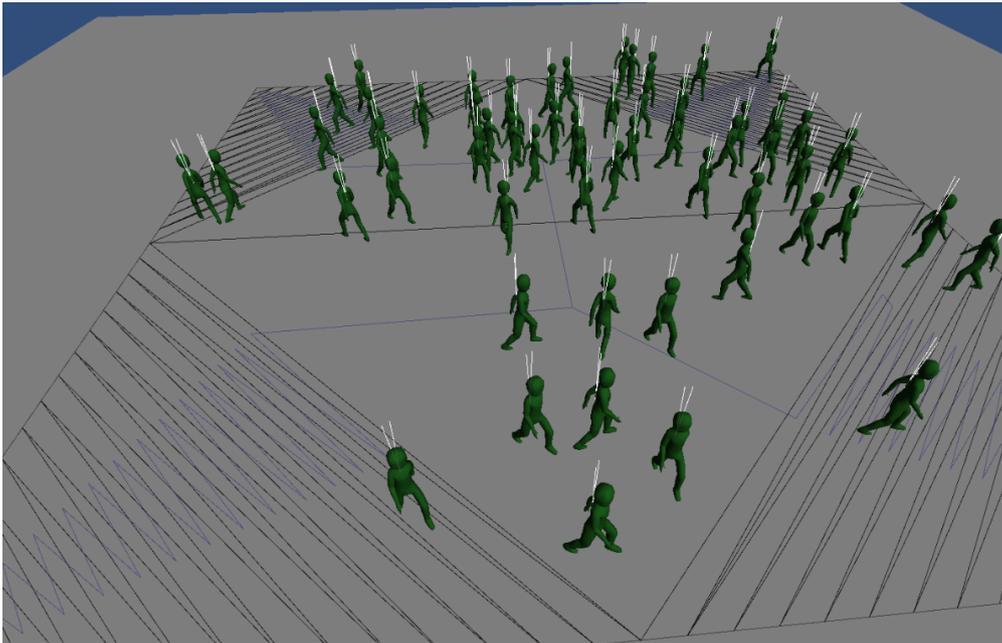


Abbildung 6.10: Szenario 5: Agenten bewegen sich wie Fußgänger auf einem Platz

### Beobachtung

In allen fünf Simulationen ist ein deutlich gleichmäßiger Bewegungsfluss zu erkennen. Erreicht ein Agent sein Ziel, bleibt dieser für eine kurze Zeit stehen, bis er ein neues Ziel gefunden hat. Die Agenten weichen sich konsequent aus und stoßen nur in seltenen Fällen leicht zusammen, wenn sich eine Anhäufung von Agenten ergibt. Das Stressniveau der Agenten erhöht sich nur leicht in Situationen, in denen sich Anhäufungen von Agenten bilden, aber sinkt kurz danach wieder auf ein Minimum. Es erreicht in diesem Szenario niemals das maximale Niveau. Da die Agenten sich konsequent ausweichen, kommt es zu keiner merklichen Veränderung des Gesundheitswertes, weswegen auch kein Agent umfällt.

### Bewertung

Es ist ein ähnliches Fortbewegungsverhalten wie bei Fußgängern zu erkennen. Die Agenten bewegen sich zufällig umher und vermeiden dabei Kollisionen mit anderen Agenten. Ein leichtes Zusammenstoßen bei einer Anhäufung von Agenten kann wieder als Streifen der Schultern gesehen werden, das in solchen Situationen vorkommen kann. Das leichte Steigen des Stressniveaus, wenn ein Agent auf eine große Ansammlung von Agenten trifft, zeigt, dass der Agent mit der Situation kurz überfordert ist und

einen Weg durch oder um die Agenten herum planen muss. Der Agent gerät hierbei jedoch nicht in einen panischen Zustand, sondern erfährt nur eine kurze Überforderung. Kurz danach sinkt sein Stressniveau auch wieder bis der Agent sich wieder ganz beruhigt. Die Agenten verletzen sich in diesem Szenario offensichtlich nicht, da es zu keinen starken Zusammenstößen kommt und räumliche Engpässe nicht vorhanden sind. Das Szenario zeigt auch, dass das Ausweichen der Agenten zuverlässig auf großen Plätzen funktioniert.

### **6.1.6 Szenario 6 - Feuer, Gebäudeevakuierung**

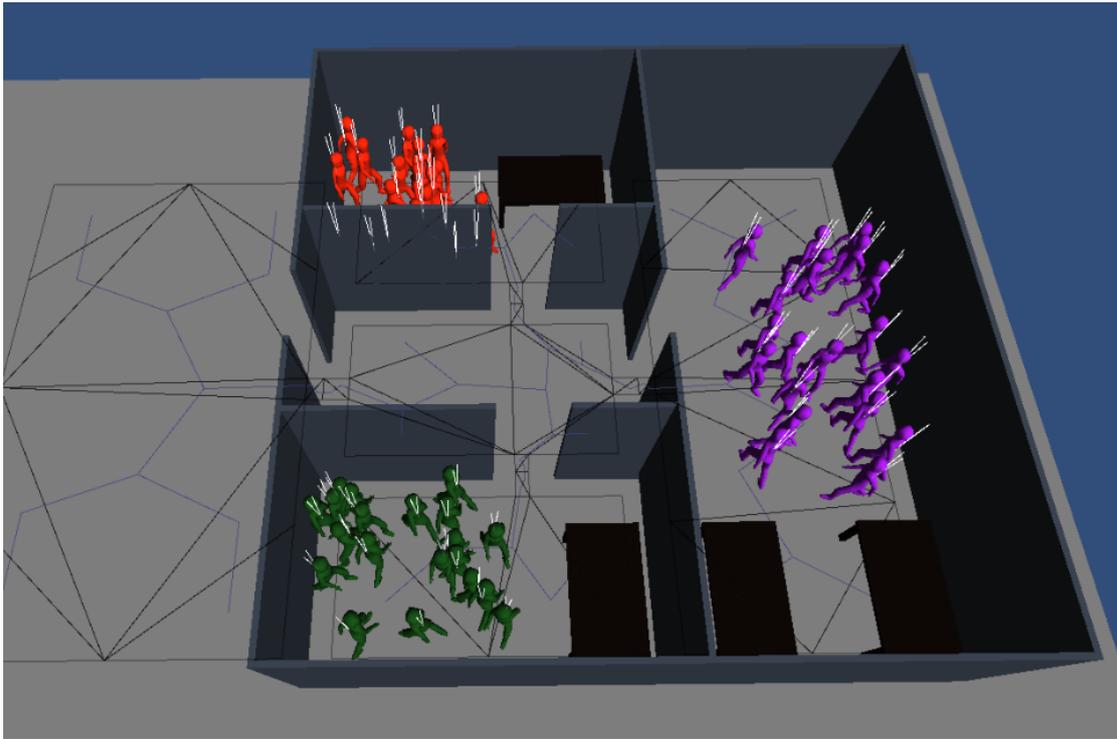
#### **Testaufbau**

Zum Schluss wurde ein Evakuierungsszenario aufgestellt und durchgeführt, bei dem die Agenten nach Ausbruch eines Feuers das Gebäude evakuieren. Dazu wurde ein kleines Gebäude mit drei Räumen und einem Flur modelliert. Jeder Raum hat jeweils einen offenen Durchgang und das Gebäude ist nur über den offenen Ausgang im Flur zu verlassen. In den fünf durchgeführten Simulationen kamen jeweils 60 Agenten, auf drei Gruppen gleichmäßig verteilt, zum Einsatz. In jedem Raum des Gebäudes befindet sich zu Beginn der Simulation eine Gruppe. Zusätzlich stehen Tische als Hindernisse in den Räumen. Die Tische haben einen Rigid Body und einen Collider und können deshalb von den Agenten verschoben werden und andere Agenten daran hindern ein Ziel zu erreichen. Die Ausgangssituation ist in Abbildung 6.11 zu sehen.

Die Agenten bewegen sich während der Simulation zunächst zufällig im Gebäude umher. Auf Wunsch kann ein Benutzer per Mausklick ein Feuer ausbrechen lassen. Daraufhin reagieren die Agenten, die das Feuer sehen, und versuchen das Gebäude zu verlassen. Die fünf Simulationen wurden zum Teil unterschiedlich durchgeführt. Beim ersten, vierten und fünften Durchlauf wurde eine kurze Zeit gewartet, bis durch den Benutzer das Feuer ausbrach. Bei den Durchläufen zwei und drei wurde der Ausbruch des Feuers gleich zu Beginn der Simulationen gesteuert.

#### **Erwartung**

Es ist zu erwarten, dass die Agenten, die das Feuer sehen, aus dem Gebäude flüchten. Agenten, die nicht das Feuer, aber flüchtende Agenten sehen, werden alarmiert. Je länger oder je mehr sie flüchtende Agenten sehen, desto schneller fliehen auch diese Agenten. Außerhalb des Gebäudes versuchen die Agenten sich zu sammeln. Das



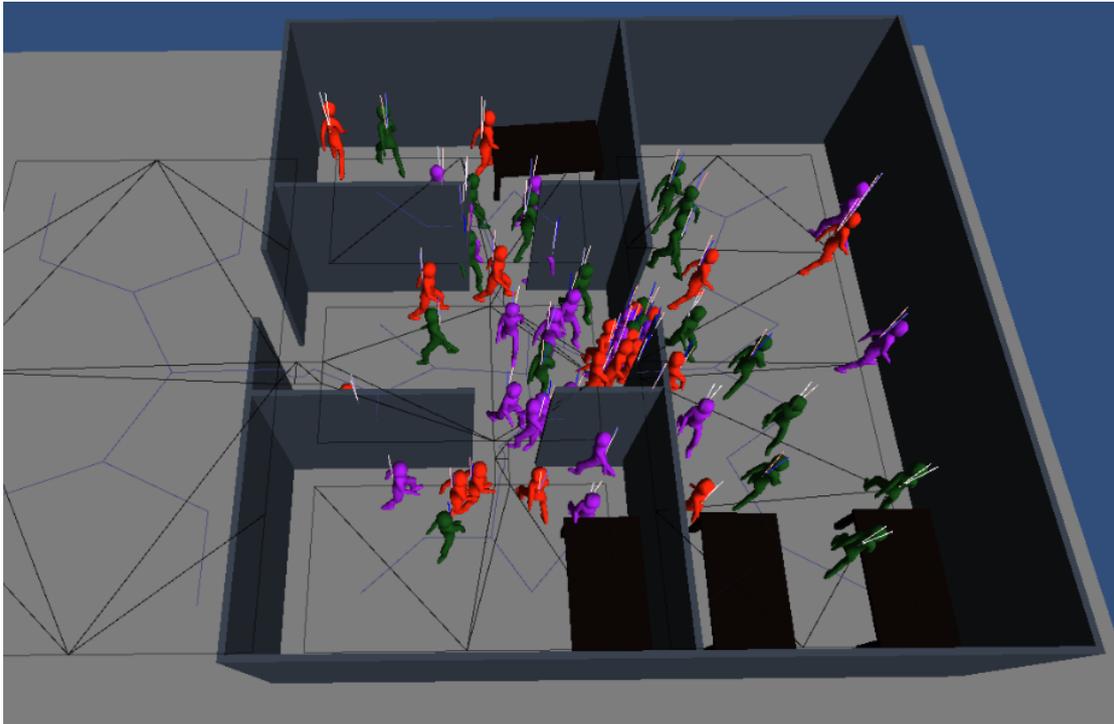
**Abbildung 6.11:** Szenario 6: Ausgangssituation des Gebäudes mit den drei Gruppen von Agenten

Stressniveau sollte zu Beginn der Simulation in niedrigen Wertebereichen bleiben, aber bei Ausbruch des Feuers bei vielen Agenten deutlich ansteigen. Es ist zu erwarten, dass es zu vielen Verletzungen der Agenten kommt und einige Agenten teilweise im Gebäude umfallen und dadurch anderen Agenten das Durchkommen erschweren.

### **Beobachtung**

Da die Simulationen teilweise unterschiedlich durchgeführt wurden, musste eine differenzierte Beobachtung vorgenommen werden. Zunächst wird die erste Simulation beschrieben, bei der sich die Agenten erst im Gebäude verteilt haben, bis das Feuer ausbrach. Die Agenten bewegen sich zunächst im Gebäude zu privaten Zielen und versuchen dabei, sich auszuweichen. Mit der Zeit entsteht im Raum auf der rechten Seite eine Anhäufung von Agenten, die ein Durchkommen verhindern (s. Abbildung 6.12). Das Stressniveau ist zu dem Zeitpunkt der Anhäufung bei den Agenten sehr hoch angestiegen und der Gesundheitswert gesunken. Anschließend bricht ein Feuer aus, das die Agenten im rechten Raum dazu bewegt zu fliehen. Die Ansammlung von Agenten

schafft es nicht, sich komplett aufzulösen, sodass viele Agenten umfallen. Die auf dem Boden liegenden Agenten erschweren den anderen Agenten das Durchkommen und werden von den fliehenden Agenten zur Seite geschoben. Am Ende sammeln sich alle Agenten, die nicht umgefallen sind, draußen vor dem Gebäude.

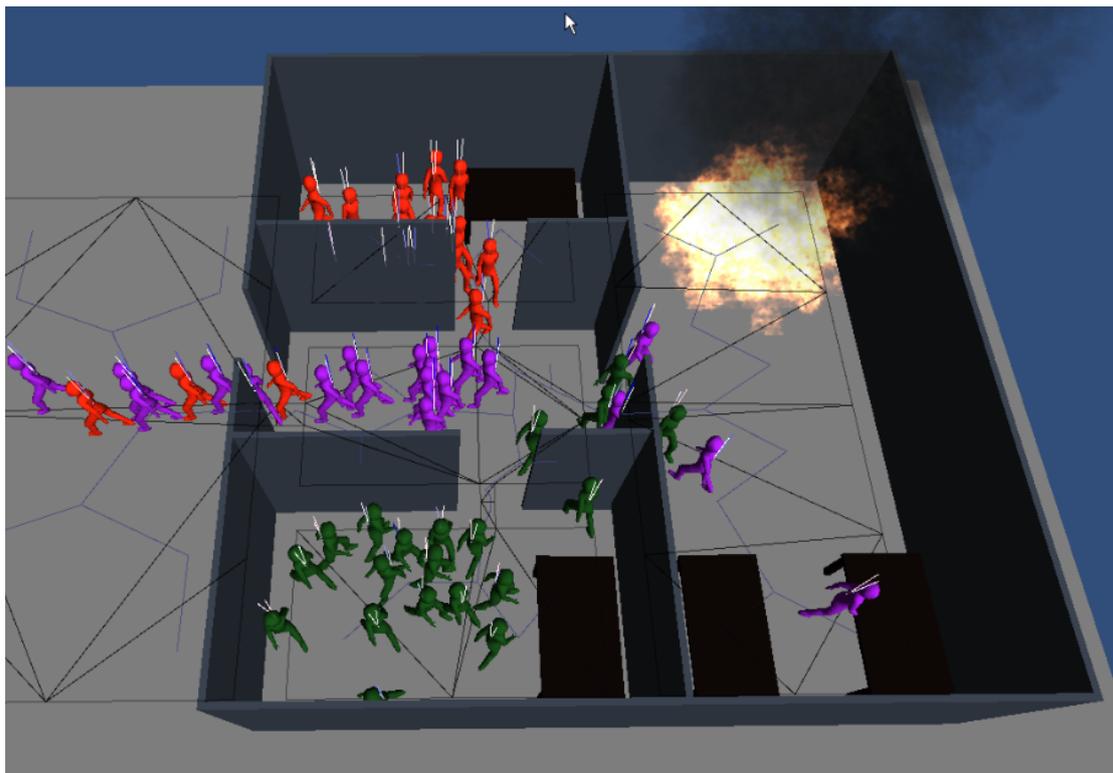


**Abbildung 6.12:** Szenario 6: Anhäufung von Agenten aus der ersten Simulation

Im zweiten und dritten Durchlauf bricht das Feuer durch den Benutzer direkt am Anfang der Simulation aus (s. Abbildung 6.13). Die Agenten im rechten Raum, in dem das Feuer ausbricht, fliehen sofort, sobald sie das Feuer gesehen haben. Aus den anderen beiden Räumen kommen die anderen Agenten, sehen die fliehenden Agenten und schließen sich diesen nach kurzer Zeit an, um das Gebäude zu verlassen. Vereinzelt sind Agenten zu beobachten, die sich weiterhin im Gebäude aufhalten und erst fliehen, sobald sie das Feuer selber gesehen haben. Das Stressniveau steigt bei den alarmierten Agenten sehr schnell und sehr hoch an. Allerdings sinkt der Gesundheitswert der Agenten nur leicht, keiner der Agenten fällt im Gebäude um und alle schaffen es, sich vor dem Gebäude zu versammeln.

Auffällig ist, dass es bei den Durchläufen vier und fünf, die wie der erste Durchlauf

durchgeführt wurden, zu abweichenden Beobachtungen kam. In Durchlauf vier sind die Beobachtungen vergleichbar mit denen in Durchlauf zwei und drei, jedoch sank der Gesundheitswert bei den Agenten deutlicher. Der fünfte Durchlauf zeigte ähnliche Beobachtungen wie im vierten Durchlauf, nur dass ein Agent umfiel und die Evakuierung insgesamt länger dauerte, da einige Agenten noch länger im Gebäude blieben. Insgesamt war bei allen Simulationen eine relativ erfolgreiche Evakuierung des Gebäudes zu sehen.



**Abbildung 6.13:** Szenario 6: Ausbruch des Feuers und evakuierende Agenten

### **Bewertung**

Dieses Szenario macht deutlich, dass durch unterschiedliche Gegebenheiten, eine Simulation auch unterschiedlich verlaufen kann. Mit der Benutzerinteraktion des Feuererzeugens präsentiert sich zum einen die Echtzeitfähigkeit des Ansatzes und zum anderen die Variabilität der Simulationen. Der erste Simulationsdurchlauf zeigt, dass durch zu enge Durchgänge ein Bewegungsstau entstehen kann, wenn einige Agenten den Raum betreten möchten und wiederum einige den Raum verlassen wollen. Nach-

dem das Feuer ausgebrochen war, wurden die Agenten in dem Raum, in dem das Feuer war, sehr gestresst. Fast panisch drückten sie gegen die Agenten, die hereinkommen wollten, welche ebenfalls einen Druck in die andere Richtung ausübten. Das führte dazu, dass einige erdrückt wurden und auf den Boden fielen. In diesem Beispiel ignorierten die Agenten die gefallenen Agenten und versuchten nur sich selbst zu retten, da das Helfen anderer in der Simulation bis jetzt nicht implementiert ist.

Beim zweiten und dritten Durchlauf ist sehr deutlich zu sehen, wie die Agenten, die noch kein Feuer gesehen haben, auf die Agenten reagieren, die durch ein Feuer oder andere Agenten alarmiert wurden. Je häufiger und je länger ein Agent einen alarmierten, fliehenden Agenten sieht, desto schneller wird dieser auch alarmiert und flieht. Die Agenten, die sich weiterhin vereinzelt im Gebäude bewegen, wurden von anderen Agenten nicht genügend alarmiert oder haben sich nicht alarmieren lassen. Sie fliehen erst, wenn sie das Feuer selber sehen, da es keinen akustisch wahrnehmbaren Feueralarm in der Simulation gibt. Auch am Stressniveau ist zu erkennen, dass die Agenten alarmiert sind, da das Wissen über das Feuer das Stressniveau der Agenten erhöht.

Die Verletzungen der Agenten halten sich in diesen Durchläufen in Grenzen, da es zu wenigen Kollisionen beim Evakuieren kommt. Im vierten und fünften Durchlauf sind die Verletzung dadurch verschuldet, dass sich die Agenten vorher im Gebäude umher bewegen. Es kommt zu kleineren Anhäufungen der Agenten ähnlich zum ersten Durchlauf, die sich aber wieder schneller auflösen. Deshalb fällt auch einer der Agenten um. Die Evakuierung des Gebäudes wegen eines Feuers gelang bei allen Durchläufen relativ erwartungsgemäß. Allerdings konnte man sehen, dass besonders die engen Durchgänge bzw. Türen den Agenten zum Verhängnis werden können.

#### **6.1.7 Diskussion der Testszenarien**

Im Verlauf der vorliegenden Arbeit wurde eine Verhaltens-KI umgesetzt, die versucht, das psychische Verhalten von Menschen in gefährlichen und stressbehafteten Situationen zu simulieren. Außerdem wurden physische Interaktionen zwischen Agenten und deren Umwelt implementiert. Die oben beschriebenen und untersuchten Testszenarien präsentieren die erarbeiteten Ergebnisse der entwickelten Simulation. Die Verhaltens-KI der Simulation gibt durch die farblich codierten Stäbe an jedem Agenten Aufschluss über den Zustand des Stressniveaus und den Zustand der Gesundheit des Agenten. Diese Visualisierung ermöglicht eine konkrete Aussage über die Zustände der Agenten

zu jeder gegebenen Zeit.

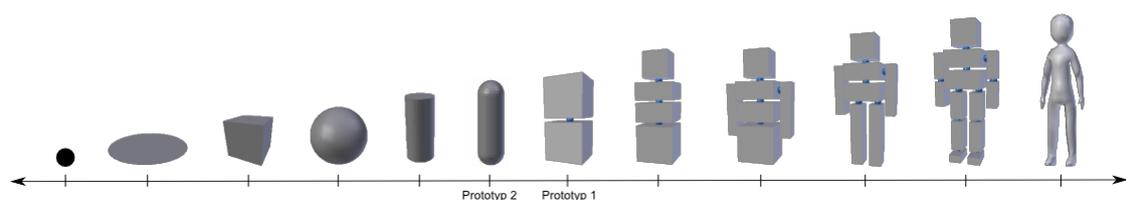
Die bewerteten Ergebnisse der Testszenarien zeigen, dass die Simulation ihren Anforderungen gerecht wird. In den drei Szenarien eins, zwei und fünf ist zu sehen, dass die Agenten untereinander erfolgreich ausweichen und in den Szenarien zwei und sechs sogar Hindernisse erfolgreich umgehen. In manchen Fällen kommt es zu leichten Zusammenstößen unter den Agenten, die aber bei einer hohen Dichte einer Menschenmenge nicht unwahrscheinlich sind.

Das Stressniveau erhöht sich in allen Testszenarien kontinuierlich und wird durch verschiedene Stressoren beeinflusst. Ein zeitlich bedingter Leistungsstressor ist in den Szenarien eins, zwei und drei zu sehen, da die Agenten ihr Ziel nicht schnell genug erreichen und das Stressniveau kontinuierlich ansteigt. Die Szenarien eins, drei, vier und fünf zeigen, dass sich das Stressniveau durch soziale Stressoren erhöht. Das sind eine hohe Anzahl von Agenten im Sichtfeld eines Agenten und leichte bis starke Zusammenstöße unter den Agenten oder mit den Hindernissen. In Szenario sechs ist zu sehen, wie ein ausgebrochenes Feuer als äußerer Stressor das Stressniveau eines Agenten explosionsartig auf 50% ansteigen lässt. Der Agent ist in einer solchen Situation deutlich gestresster, da er möglichst schnell außer Reichweite des Feuers kommen will. Die beiden Szenarien zwei und fünf machen deutlich, dass das Stressniveau mit der Zeit wieder sinkt, wenn keine Stressoren auf den Agenten wirken. Sinkt das Stressniveau nicht rechtzeitig, kann es passieren, dass die Agenten wie in den Szenarien eins, zwei, drei und vier in einen Panik ähnlichen Zustand verfallen und das Ausweichen ignorieren. Dadurch entstehen mehrere Zusammenstöße und die Agenten stoßen und drücken gegen Hindernisse und andere Agenten. Hierbei ist besonders gut die physische Interaktion mit den Hindernissen und der Agenten untereinander zu sehen.

Neben dem Stressniveau sind durch den Gesundheitswert Verletzungen und das Umfallen von Agenten in allen Szenarien bis auf das fünfte Szenario zu beobachten. Die erfolgreiche Kombination einer Verhaltens-KI und Physik-Engine wird hier deutlich, da durch das psychische Verhalten der Agenten deren physische Interaktionen beeinflusst werden. Ist ein Agent gelassen und hat ein niedriges Stressniveau, bewegt er sich auch zielstrebig fort und weicht anderen Agenten und Hindernissen aus. Gerät der Agent in gefährliche Situationen, die das Stressniveau erhöhen, ignoriert er das Ausweichen und nimmt Zusammenstöße in Kauf. Er versucht sich den Weg frei zu machen und sich selbst in Sicherheit zu bringen.

Desweiteren zeigt das sechste Szenario die Echtzeitfähigkeit des Ansatzes durch die Benutzerinteraktion. Zusätzlich werden in diesem Szenario gefährliche Engpässe in Gebäuden deutlich abgebildet. Außerdem ist eine Kommunikation der Agenten untereinander auszumachen, da sie sich über ein möglicherweise ausgebrochenes Feuer verständigen, um sich dann schnellstmöglich in Sicherheit zu begeben. Durch den Vergleich mit Szenarien, die tatsächlich in der Realität passiert sind, werden die erarbeiteten Ergebnisse gefestigt. Die Ergebnisse des dritten Szenarios zeigen die gleiche Größenordnung wie der Verlauf dieses Szenarios in der Realität. Beim vierten Szenario sind die Ergebnisse nur ähnlich zur Realität. Würde man aber die Anzahl der Agenten erhöhen, würde sich sehr wahrscheinlich die gleiche Größenordnung ergeben und sich das Szenario der Realität in der Simulation abbilden lassen. Dennoch müssen diese Ergebnisse wissenschaftlich weiter untersucht werden. So müssen die Ergebnisse des entwickelten Stressniveaus mit Ergebnissen aus psychologischen etablierten Studien wie zum Beispiel dem AAS (s. Kapitel 3.1) verglichen werden. Zudem müsste der Gesundheitswert realistischer skaliert werden, um einen natürlicheren Verletzungsgrad eines Menschen abzubilden und ebenfalls mit Studien über den körperlichen Zustand eines Menschen abgeglichen werden. Außerdem wird eine detailliertere physikalische Repräsentation eines Agenten benötigt um realistischere Zusammenstöße zwischen Agenten abzubilden. Diese physikalischen Repräsentationen werden im Nachfolgenden erläutert und anschließend extrapoliert, um die Verwendung einer komplexen Darstellung eines Agenten abzuschätzen.

## 6.2 Extrapolation der Prototypen



**Abbildung 6.14:** Extrapolation der Komplexität der physikalischen Repräsentation eines Agenten (Joints sind Blau dargestellt)

In der vorliegenden Arbeit wird der wissenschaftliche Aspekt untersucht, inwieweit eine physikalische Repräsentation eines Agenten unter Verwendung einer Physik-Engine in einer Menschenmengen-Simulation Anwendung finden kann. Deshalb wird im Folgenden eine Extrapolation der entwickelten Prototypen vorgenommen. In Abbildung

6.14 ist die physikalische Repräsentation eines Agenten auf einer Art Komplexitätsleiste abgebildet. Die Komplexitäten der physikalischen Repräsentationen eines Agenten sind von der punktbasierten Darstellung bis hin zur detailgetreuen Abbildung eines Menschen aufgezeigt. Die Positionen der beiden Prototypen sind in der Leiste markiert. Der zweite Prototyp verwendet mit einer Kapsel eine noch relativ einfache physikalische Repräsentation. Sie ist aber etwas komplexer als eine Darstellung, die nur einen Punkt, eine Kreisscheibe oder eine Kugel abbildet.

Eine punktbasierte Darstellung wird größtenteils in makroskopischen Systemen verwendet, wenn zum Beispiel eine Menschenmenge als Gas oder Flüssigkeit simuliert wird. Die Punkte agieren bei dieser Simulation zusammen und sind nicht eigenständig. Allerdings stehen sie unter dem Einfluss der physikalischen Kräfte der Fluidodynamik. Verwendet man die punktbasierte Darstellung in mikroskopischen Systemen, erhält man eine zu minimale Repräsentation um eine plausible physikalische Interaktion zu beschreiben. Allein für die vorliegende Arbeit ergibt sich der Nachteil, dass ein einzelner Punkt kein Rigid Body sein kann, da ein Rigid Body eine Menge von Punkten beschreibt, deren Abstände sich zu keiner Zeit ändern. Aus diesem Grund ist es möglich, eine zweidimensionale Kreisscheibe zu verwenden, die in einer zweidimensionalen Welt als zweidimensionaler Rigid Body beschrieben werden kann. Mit dieser Kreisscheibe lassen sich zum Beispiel Kollisionen leichter und genauer berechnen.

Der nächste Schritt der Komplexität wäre eine Repräsentation im dreidimensionalen Raum. In diesem Fall wäre die einfachste Repräsentation ein Würfel, der die Eigenschaft eines dreidimensionalen Rigid Body besitzen kann. Der Nachteil eines Würfels, um einen Agenten damit physikalisch abzubilden, ist die Schwierigkeit eine Schräge zu besteigen. Die folgenden Komplexitätsstufen sollen alle die Eigenschaft eines Rigid Body besitzen. Um eine Schräge zu erklimmen, ist es daher möglich, eine Kugel als Repräsentation zu verwenden. Allerdings ist es schwieriger, die Balance einer Kugel zu erhalten. Der Nachteil, den ein Würfel und eine Kugel gemeinsam haben, ist die Abwesenheit des Umfallens. Es ist nicht direkt auszumachen, ob oder wie ein Würfel oder eine Kugel umgefallen ist. Diese Eigenschaft ist für die Agenten der in der vorliegenden Arbeit entwickelten Simulation aber von großer Bedeutung.

Die darauffolgende Komplexität besteht also darin, ein längliches Objekt als physikalische Repräsentation des Agenten zu benutzen. Hierzu dient zum Beispiel ein Quader oder ein Zylinder, mit dem das Umfallen eines Agenten abgebildet werden kann. Al-

lerdings ergibt sich hier wieder der Nachteil, dass diese Darstellungen schräge Flächen nur schwierig erreichen können. Aus diesem Grund eignet sich eine Kapsel wie sie im zweiten Prototypen verwendet wurde am besten.

Der anschließende Schritt, die Komplexität der physikalischen Repräsentation des Agenten zu erweitern, ist die Verbindung mehrerer Rigid Bodies durch Joints. Mit dieser Erweiterung ist es möglich, eine physikalisch plausible Beweglichkeit des Körpers des Agenten darzustellen. Die erste Rigid Body Kombination beschreibt in der Komplexitätsleiste zwei Würfel, die übereinander mit einem Joint verbunden sind. Dadurch ist die Bewegung von Ober- und Unterkörper getrennt und eine hundertprozentige Beweglichkeit im Gegensatz zu einer Darstellung ohne Joint vorhanden. Der erste Prototyp verwendet diese komplexere physikalische Repräsentation des Körpers. Allerdings wurden statt Würfeln zwei übereinander verbundene Kugel verwendet.

Die Komplexitätsleiste zeigt die weiteren Komplexitäten nur mit Würfeln und Quadern, allerdings ist die Wahl der Primitive zur Zusammensetzung beliebig. Es können nur Kugeln oder Zylinder verwendet, aber zum Beispiel auch Kapseln mit Quadern kombiniert werden. Die weiteren Komplexitäten beschreiben eine immer weiter angenäherte physikalische Repräsentation eines Agenten. Dabei kann jeder bewegliche Teil des Körpers weiter unterteilt werden. Eine zusätzliche Unterteilung ist zum Beispiel in Kopf, Brust, Bauch und Beine möglich. Die Segmentierung ist soweit möglich, bis alle beweglichen Teile des Körpers abgebildet wurden. Für diese Komplexitätsstufe wird meistens auch der Begriff Ragdoll verwendet, wenn keinerlei Kontrolle über die einzelnen Segmente besteht und diese den physikalischen Kräften ausgesetzt sind.

Die physikalisch detaillierteste Darstellung des Körpers eines Agenten wäre durch Soft Bodies möglich. Liu u. a. (2013) beschreiben einen Ansatz, bei dem ein menschlicher Körper durch Soft Bodies repräsentiert wird. Dabei ist das Soft Body Skelett mit einem Rigid Body Skelett verbunden und kann als Erweiterung der Rigid Bodies gesehen werden. Dieser Ansatz ist interaktiv, aber leider nicht echtzeitfähig und würde daher die Anforderungen an die entwickelte Simulation der vorliegenden Arbeit nicht erfüllen.

Mit der physikalischen Repräsentation des ersten Prototypen konnte keine stabile Balance des Körpers erzielt werden. Auch beim zweiten Prototypen musste eine Stütze während der Navigation benutzt werden, damit der Agent beim Rotieren die Balance hält. Die Übertragung dieser Erkenntnisse auf die komplexeren Repräsentationen

des Agenten ergeben, dass für eine stabile Kontrolle der Körper eine Stütze benötigt wird. Würde eine Stütze nicht vorhanden sein, wäre das Erreichen einer Körperbalance rechnerisch zu aufwändig oder, wie im ersten Prototypen dokumentiert, zu instabil. Als Stützen können Controller verwendet werden, wie in Muico u. a. (2011) und Wu u. Popović (2010) beschrieben. Diese Ansätze zeigen ein erfolgreich durch Stützen balanciertes Rigid Body Skelett. Damit könnten realistischere Zusammenstöße abgebildet werden. Zum Beispiel wären Schulter an Schulter Zusammenstöße deutlich erkennbar, da nur der entsprechende Teil des Körpers an der Kollision beteiligt wäre. Außerdem wäre ein realistisches Umfallen der Agenten möglich und zusätzlich könnte ein plausibles Aufstehen der Agenten modelliert und umgesetzt werden.

Dennoch ist zu kritisieren, dass je komplexer die physikalische Repräsentation eines Agenten sein wird, desto mehr Berechnungen von der Physik-Engine nötig sein werden. Zudem sind bei einer Kollision von zwei Agenten, bei denen der Körper aus mehr als einem Rigid Body besteht, nicht nur zwei, sondern mehrere Objekte an der Kollision beteiligt. Würde diese Repräsentation dann für eine Simulation einer Menschenmenge verwendet werden, würde die Performanz darunter leiden und die Echtzeitfähigkeit des Ansatzes würde nicht vorhanden sein. Es wären wahrscheinlich noch weniger Agenten auf einmal simulierbar, als die in der vorliegenden Arbeit entwickelte Simulation. Trotzdem ist die weitere Forschung an der Verwendung komplexerer physikalischer Repräsentationen eines Agenten in einer Menschenmengen-Simulation von interessanter Bedeutung.

### **6.3 Vergleich der Ergebnisse mit anderen Menschenmengen-Simulationen**

Die Ergebnisse der Evaluation der Simulation und die anschließende Extrapolation der physikalischen Repräsentation eines Agenten haben gezeigt, dass eine Kombination einer Verhaltens-KI mit den Verfahren einer Physik-Engine erfolgreich möglich ist. Im Folgenden werden die erarbeiteten Ergebnisse mit den Ergebnissen aus anderen Ansätzen die Menschenmengen simulieren, verglichen und bewertet. Dazu werden die Ergebnisse der Ansätze aus dem Kapitel 3 betrachtet. Abschließend wird anhand der eigenen Ergebnisse eine generelle Abgrenzung zu makroskopischen Systemen vorgenommen.

Die Ergebnisse der Ansätze aus Kapitel 3 sind jeweils unterschiedlich, da sich der Ansatz von Kim u. a. (2012) auf die Modellierung von Stress in einer Menschenmenge

konzentriert und Kim u. a. (2013) einen physikalischen Ansatz untersuchen. Dennoch lassen sich Gleichheiten und Unterschiede zu den Ergebnissen der vorliegenden Arbeit herausstellen. Die Ergebnisse von Kim u. a. (2012) beziehen sich auf den Stress von Agenten und deren Verhalten. Sie zeigen, dass Agenten bei erhöhtem Stress schneller werden. Der Stress der Agenten erhöht sich zum Beispiel durch Feuer oder zeitlichen Druck. Bei stark erhöhtem Stress verhalten sich die Agenten egoistischer, impulsiver und rücksichtsloser. Außerdem stoßen Agenten ineinander und schieben andere Agenten zur Seite. Die in der gegenwärtigen Arbeit erzielten Ergebnisse zeigen Parallelen zu den soeben aufgezählten Ergebnissen. Der Stress eines Agenten erhöht sich ebenfalls durch ein entfachtes Feuer oder zeitlichen Druck. Zusätzlich erhöht er sich auch durch eine hohe Dichte der Menschenmenge und die Zusammenstöße der Agenten. Das durch zu hohen Stress entstandene Verhalten eines Agenten ist ebenfalls ähnlich zu den Ergebnissen des Ansatzes von Kim u. a. (2012). Die Agenten verhalten sich dann ebenfalls egoistischer, impulsiver und rücksichtsloser. Das Ineinanderstoßen und zur Seite schieben von Agenten zeigen auch die Ergebnisse der vorliegenden Arbeit. Zusätzlich ist es möglich, den Verletzungsgrad eines Agenten durch einen Gesundheitswert darzustellen. Allerdings zeigen Kim u. a. (2012) im Gegensatz zur vorliegenden Arbeit, dass sich die Agenten bei erhöhtem Stress unkooperativer verhalten, was zu einem unkoordinierten Bewegungsfluss der Menschenmenge führt. Da in der gegenwärtigen Arbeit kein Gruppenverhalten zur Fortbewegung umgesetzt wurde, konnte dieser Teil nicht untersucht werden. Dafür wurde der Fokus auf die physikalischen Eigenschaften eines Agenten gelegt, ähnlich des Ansatzes von Kim u. a. (2013).

Die Ergebnisse dieses ähnlichen Ansatzes zeigen, dass absichtlich unkooperative Agenten drängeln und schieben, um physisch einen Weg durch eine Menschenmenge zu finden. Zudem können die Agenten dynamischen Objekten und anderen Agenten erfolgreich ausweichen. Die Ergebnisse dieses Ansatzes sind wie in der vorliegenden Arbeit nur approximiert und nicht physikalisch genau. Außerdem decken sie sich mit den Ergebnissen dieser Arbeit. Allerdings wird ein Agent in dem Ansatz von Kim u. a. (2013) nur als zweidimensionale Scheibe repräsentiert. Damit ist keine genaue physikalische Interaktion zwischen Agenten und dreidimensionalen Objekten möglich. Im Gegensatz dazu verwendet der in der vorliegenden Arbeit entwickelte Ansatz Rigid Bodies als physikalische Repräsentation eines Agenten. Mit Rigid Bodies lässt sich eine genauere und plausiblere physikalische Interaktion zwischen Agenten und dreidimensionalen Objekten beschreiben. Zudem zeigen die Ergebnisse, dass es möglich ist, dass Agenten im Gegensatz zu den anderen Ansätzen bei einem zu geringem Gesundheits-

wert umfallen.

Neben dem Vergleich der Ergebnisse ähnlicher Ansätze ist der Vergleich des in der vorliegenden Arbeit entwickelten mikroskopischen Ansatzes mit einem makroskopischen Ansatz von Bedeutung. Mit den Ergebnissen der vorliegenden Arbeit lassen sich Phänomene zeigen, die mit einem makroskopischen Ansatz nicht möglich wären. Dazu zählen zum einen das Stressniveau eines Agenten, das sich bezüglich unterschiedlicher Stressoren bei jedem Agenten individuell verändert, und zum anderen das Fortbewegungsverhalten und die physische Interaktion der eigenwilligen Agenten. Mit einer Flüssigkeitssimulation als makroskopisches System lässt sich größtenteils nur der Bewegungsfluss der gesamten Menschenmenge beschreiben. Dabei gehören die einzelnen Agenten innerhalb der Flüssigkeit immer zusammen und werden nur durch einen Punkt und eine Richtung bestimmt. Punkt und Richtung können sie nicht selbst beeinflussen, da sich die Flüssigkeit unter den physikalischen Kräften der Fluidodynamik bewegt. Damit erhält die Flüssigkeit einen gewissen Zusammenhalt. Die Flüssigkeit ist weiterhin eine Menschenmenge und soll nur symbolisch als Flüssigkeit das Verhalten der Menschenmenge beschreiben. Durch die unabhängigen Individuen eines mikroskopischen Systems, wie in der vorliegenden Arbeit entwickelt, lässt sich ebenfalls ein plausibles Fortbewegungsverhalten implementieren. Mit den individuellen Agenten, die durch einen Rigid Body repräsentiert werden, ist zudem eine Kollision zwischen den Agenten und Hindernissen möglich, die das Verhalten des einzelnen Agenten verändert, was sich auf das Verhalten der gesamten Menschenmenge überträgt. Diese Kollisionen sind mit einem makroskopischen System nicht möglich, da eine Menschenmenge als Flüssigkeit nur gemeinsam Hindernissen ausweichen oder mit diesen kollidieren kann.

Nachteilig ist allerdings die Performanz des in der vorliegenden Arbeit entwickelten Ansatzes zu erwähnen. Da mit einer Physik-Engine eine möglichst physikalisch plausible Berechnung von physischen Interaktionen der Agenten erzielt werden sollte, wird bei zu vielen Agenten die Berechnung der Kollisionen zu aufwändig und die Bilder pro Sekunde fallen unter 25, sodass der Ansatz nicht mehr echtzeitfähig wäre. Diese intensive Berechnungen der Physik bleiben bei den anderen Ansätzen aus, da eine unterschiedliche Repräsentation der Agenten gewählt wurde. Somit können sie mehrere Hundert Agenten simulieren im Gegensatz zu maximal 60 in der Simulation der vorliegenden Arbeit. Trotz der geringen Anzahl an Agenten lassen sich unterschiedliche und ähnliche Phänomene zu den anderen Ansätzen beobachten.

## 6.4 Zusammenfassung

In diesem Kapitel wurden die Ergebnisse der vorliegenden Arbeit präsentiert. Die Evaluation der durchgeführten Testszenarien hatte ergeben, dass sich die in der vorliegenden Arbeit entwickelte Simulation zur Abbildung von psychischen und physischen Faktoren eignet. Zudem wurde eine Extrapolation der physikalischen Repräsentation eines Agenten durchgeführt und bewertet. Das Ergebnis der Extrapolation zeigt, dass eine komplexere Repräsentation des Agenten physikalische Interaktionen detaillierter abbilden kann, aber für eine Menschenmengen-Simulation aus Performanz-Gründen eher ungeeignet ist. Ebenso wurden die Ergebnisse der entwickelten Simulation mit den Ergebnissen anderer Ansätze verglichen und gezeigt, dass eine Kombination von physischen und psychischen Faktoren in einer agentenbasierten Simulation von Vorteil ist. Zum Schluss wird im nächsten Kapitel ein Fazit gezogen und ein Ausblick vorgenommen.

## 7 Fazit und Ausblick

Die Resultate dieser Arbeit werden abschließend in einem Fazit festgehalten. Außerdem wird eine Zusammenführung der entwickelten Prototypen beschrieben und prospektiv mit einem Ausblick abgeschlossen.

### 7.1 Fazit

In der vorliegenden Arbeit wurde eine prototypische Anwendung zur Simulation von Menschenmengen entwickelt. Zur Umsetzung wurden Techniken der Echtzeitphysik wie Rigid Bodies und Navigationselemente aus Videospiele wie Navigationmeshes verwendet. Außerdem wurde eine Verhaltens-KI entwickelt, die das Verhalten eines Menschen basierend auf Stress abbildet. Die Implementierung der agentenbasierten Simulation gelang mit frei verfügbarer Software. Anhand der Physik-Engine und weiteren Komponenten aus Videospiele konnte eine echtzeitfähige Simulation umgesetzt werden. Es ist eine Benutzerinteraktion möglich, bei der ein Benutzer ein Feuer zu einer beliebigen Zeit ausbrechen lassen und Objekte in der Szene bewegen kann. Die entwickelte Simulation ermöglicht es, psychische und physische Faktoren von Agenten in einer großen Menge von Agenten abzubilden. Dazu zählen Kollisionen und Interaktionen zwischen den Agenten und mit Hindernissen, das Umfallen von Agenten und das sich verändernde Verhalten der Agenten auf äußere Einflüsse. Diese physischen Krafteinwirkungen sind als das Gedränge und das Erdrücken von Menschen zu beurteilen. Dadurch lassen sich gefährliche Situationen wie Engpässe, Hindernisse und Versperrungen in Gebäuden oder eine hohe Dichte der Menschenmenge besser einschätzen. Außerdem sind psychische Verhaltensmuster der Agenten anhand der Simulation vorhersagbar. Das Verhalten von Menschen in gefährlichen oder stressbehafteten Situationen lässt sich somit nachbilden und analysieren.

Neben der entwickelten Simulation wurde die physikalische Repräsentation eines Agenten untersucht. Es ist gelungen, einen Agenten durch Rigid Bodies zu repräsentieren. Jedoch ist es nach Untersuchungen der vorliegenden Arbeit nicht möglich, den Rigid Body ohne Stützen zu balancieren. Dennoch haben die Ergebnisse gezeigt, dass eine komplexere und detailliertere Abbildung durch mehrere zusammengesetzte Rigid Bodies zu einem realistischeren und physikalisch plausibleren Verhalten eines Agenten sowie dessen Interaktionen mit der Umwelt führt. Mit einer Physik-Engine konnte ein echtzeitfähiger Ansatz entwickelt werden, mit dem aber nur maximal eine Anzahl von 60 Agenten simuliert werden kann. Dabei ist wahrscheinlich der Trigger als zusätzli-

ches physikalisches Objekt der Flaschenhals für die Performanz der Anwendung. Die entwickelte prototypische Simulation dient allerdings nur als Vorschlag diese Faktoren abzubilden. Für einen industriellen Einsatz müssten weitere Bereiche der Anwendung wissenschaftlich untersucht und der Prototyp erweitert werden.

## 7.2 Ausblick

Nach den erarbeiteten Ergebnissen der vorliegenden Arbeit ergeben sich weitere Bereiche, die in diesem Zusammenhang untersucht werden können. Die entwickelte Simulation dieser Arbeit könnte zum Beispiel zukunftsweisend als Anwendung auf einem Tablet-PC laufen. Organisatoren könnten bei einer Veranstaltung vor Ort eine Evakuierung simulieren, um Gefahrenquellen, die durch etwaige Veränderungen des Veranstaltungsgeländes entstehen, vorherzusehen und zu minimieren oder zu eliminieren. Damit die Anwendung für ein solches Szenario einsatzfähig wäre, müssten weitere wissenschaftliche Untersuchungen durchgeführt und die Anwendung entsprechend erweitert werden. Im Hinblick darauf werden im Folgenden weitere mögliche Forschungsgebiete und Funktionalitäten der Simulation betrachtet. Außerdem werden die beiden entwickelten Prototypen theoretisch miteinander kombiniert, um die Möglichkeiten einer komplexeren physikalischen Repräsentation eines Agenten aufzuzeigen.

### Theoretische Fusion der Prototypen

Da in der vorgegebenen Zeit der Arbeit keine stabile Version des Agenten im ersten Prototypen erzielt werden konnte, wird im Folgenden eine theoretische Verbindung der Prototypen elabiert. Die Simulation des ersten Prototypen konzentrierte sich gezielt auf die Repräsentation des Agenten durch mehrere Rigid Bodies. Der zweite Prototyp hingegen legt den Fokus auf die Simulation von Menschenmengen und der Kombination von physikalischen Einflüssen und psychischem Verhalten. Bei einer Verbindung der beiden Prototypen würde die relativ einfache physikalische Repräsentation des Agenten im zweiten Prototypen durch die des ersten Prototypen ersetzt. Die einfache physikalische Repräsentation des Agenten ist eine Kapsel, die dann durch die Zusammensetzung aus mehreren Rigid Bodies mit Joints ausgetauscht wird.

Durch eine realistischere physikalische Repräsentation des Agenten ergeben sich einige Vorteile. Zum einen kann durch die Beweglichkeit der Agenten eine realistischere Fortbewegung erreicht werden. Zum anderen ist eine realistischere Kollision zwischen Agenten und Objekten möglich, da Agenten eine Balance besitzen, die sie im

Ausgleich halten müssen. Bei Zusammenstößen lässt sich dann ein Taumeln abbilden, das den Agenten dazu bringt, wieder die Balance zu finden. Außerdem ist es möglich, dass Agenten, unabhängig von einem Gesundheitswert, durch zu starke Zusammenstöße umfallen können. Sollte das Umfallen dann den Gesundheitswert nicht auf Null bringen, kann der Agent auch wieder aufstehen und sich weiter fortbewegen. Diese Vorteile ermöglichen eine physikalisch plausiblere Simulation einer Menschenmenge, da sich die Agenten, repräsentiert als Rigid Body Skelett, bei Zusammenstößen realistischer verhalten als bei der Repräsentation der Agenten als Kapsel. Zudem erweitert sich das psychische Verhalten der Agenten, da ein umgefallener Agent den Drang verspürt wieder aufzustehen. Dieser Drang des Aufstehens kann ebenfalls im Stressniveau abgebildet werden. Das Stressniveau wird durch den Drang erhöht und könnte dazu führen, dass der aufstehende Agent benachbarte Agenten beim Aufstehen anstößt oder umstößt, weil er durch das stark erhöhte Stressniveau einen sehr hohen Druck verspürt sein eigentliches Ziel zu erreichen.

In Kapitel 6.2 ist in Abbildung 6.14 eine Extrapolation der physikalischen Repräsentation eines Agenten veranschaulicht. Diese zeigt unterschiedlich komplexe Repräsentationen. Eine Erweiterung der Fusion der beiden Prototypen wäre, die Möglichkeit die physikalische Repräsentation der Agenten nach Bedarf auszuwählen. Eine Auswahl wäre sinnvoll, da eine komplexere Repräsentation in der Simulation zwar mehr Aufschluss über eventuelle Gefahren gibt, aber durch eine vereinfachte Repräsentation eine größere Anzahl von Agenten simulierbar ist.

### **Anzahl der Agenten**

Besonders interessant ist es, mehrere Agenten auf einmal simulieren zu können, ohne große Performanz-Einbußen zu erleiden. Da, wie im Fazit erwähnt, wahrscheinlich der Trigger als Sichtfeld des Agenten ein weiteres physikalisches Objekt ist, mit dem Kollisionsabfragen durchgeführt werden, verringert sich die Performanz. Ein Möglichkeit wäre es, das Trigger-Sichtfeld gegen eine einfache mathematische Berechnung eines Sichtkegels auszutauschen, um die physikalischen Berechnungen zu minimieren. Eventuell könnte sich dadurch das Ausweichen der Agenten verändern, welches dann zusätzlich angepasst werden müsste.

## **Navigation der Agenten**

Grundsätzlich kann die Navigation der Agenten verbessert werden. Das verwendete statische Navigationmesh könnte durch ein dynamisches Navigationmesh ersetzt werden. Mit einem dynamischen Navigationmesh würde sich die dynamische Kollisionsvermeidung deutlich verbessern und vereinfachen. Das Navigationmesh ändert sich an der Stelle, an der sich ein Agent befindet, sodass der Agent aus dem Navigationmesh ausgeschlossen und als Hindernis angesehen wird. Zusätzlich ist es möglich, das Navigationmesh so zu erweitern, dass Stockwerke durch die Agenten begehbar werden, da die Simulation unabhängig davon im dreidimensionalen Raum stattfindet und die physikalische Repräsentation der Agenten dafür schon ausgelegt ist. Die in der vorliegenden Arbeit erarbeitete Kollisionsvermeidung könnte dann erweitert werden, sodass sie in Zusammenarbeit mit dem dynamischen Navigationmesh arbeitet. Es wäre denkbar, dass die dynamische Kollisionsvermeidung in Situationen zum Einsatz kommt, in denen das dynamische Navigationmesh keine Ausweichmöglichkeit mehr bietet.

Die Fortbewegung durch Kraft und Drehmoment müsste ebenfalls aufgegriffen werden. Hierzu müsste die Erhaltung der Balance durch diese Kräfte weiter untersucht werden, um einen Agenten nur durch diese Kräfte zu balancieren. Im Gegensatz dazu können weitere Untersuchungen zu den Controllern angestellt werden, die die Balance als Stützen stabilisieren.

## **Physikalische Repräsentation der Agenten**

Desweiteren wäre die Umsetzung einer menschlicheren Repräsentation eines Agenten durch die Zusammensetzung von Rigid Bodies denkbar. Die vorgestellten Ansätze von Wu u. Popović (2010) und Muico u. a. (2011) zeigen ein Potenzial dieses Vorgehen umzusetzen. Dazu müssten diese Ansätze tiefergehend erarbeitet und anschließend in die entwickelte Simulation der vorliegenden Arbeit integriert werden. Der Ansatz von Wu u. Popović (2010) präsentiert sogar eine Herangehensweise, bei dem der mit Rigid Bodies nachgebaute menschliche Körper physikalisch plausibel Treppen steigen kann. Auch das Umfallen der Agenten sowie das Aufstehen wären damit deutlich realistischer umsetzbar.

## **Verhalten der Agenten**

Das Verhalten der Agenten könnte unterschiedlich erweitert werden. Zunächst wäre es denkbar, Agenten zu kategorisieren, indem die körperliche Beschaffenheit der Agenten unterteilt wird. So könnten Kinder von Erwachsenen abgehoben werden, ältere Personen oder sportlichere Personen abgebildet, aber auch Rollstuhlfahrer und anders körperlich beeinträchtigte Personen dargestellt werden. Zudem könnten Eigenschaften wie Körpergröße, Geschlecht und Gewicht abgebildet werden.

Das entwickelte Stressniveau könnte ebenfalls erweitert werden. Eine Idee wäre es, weitere Eigenschaften dem Agenten zuzuteilen. Neben dem Stressniveau ließen sich Eigenschaften beschreiben wie Angst, Erschöpfung, Ausdauer, Mut oder Hilfsbereitschaft. Diese könnten das Stressniveau beeinflussen und sich insgesamt auf die Simulation auswirken. Ein denkbare Beispiel einer Evakuierungssituation wäre, dass ein Agent Mitgefühl zeigt und einem ängstlichen Agenten der nicht weiterkommt hilft, obwohl ein Feuer ausgebrochen ist und beide schnellstmöglich in Sicherheit gelangen sollten.

Desweiteren könnten durch den Gesundheitswert weitere physische Erscheinungen neben dem Umfallen abgebildet werden, indem Verletzungen eines Agenten dargestellt werden. Es wäre vorstellbar, Agenten zu simulieren, die zum Beispiel einen gebrochenen Fuß oder Asthma haben. Diese Verletzungen wirken sich dann auf eine eingeschränkte Fortbewegung aus, was den Agenten verlangsamt oder ihn ohne die Hilfe eines anderen Agenten nicht mehr vorwärts kommen lässt.

Darüber hinaus sollte das Gruppenverhalten der Agenten weiter untersucht und umgesetzt werden. Hierzu zählt die Kommunikation der Agenten untereinander, wie sie es bei einem ausgebrochenen Feuer in der entwickelten Simulation praktizieren, aber auch die Kommunikation beim Fortbewegen, sodass sie zum Beispiel in einer ungestressten Situation in einer Spur laufen. Zudem könnten die Agenten in Paaren von 2 oder Gruppen von 3-10 Personen gegliedert werden, da Menschen oft als Gruppen unterwegs sind. Dadurch könnte ein weiteres Verhalten der Agenten abgebildet werden, da diese Gruppen zu jeder Zeit versuchen würden zusammenzubleiben.

### **Validierung der Ergebnisse**

Insgesamt müssen die erarbeiteten Ergebnisse mit dem Verhalten von einer Menschenmenge in der Realität verglichen werden. Dazu könnte das entwickelte Stressniveau einer etablierten Studie wie dem AAS, das die Reaktionen auf Stressoren beschreibt, gegenübergestellt werden. Weiterhin sollten fachliche Experten der Psychologie und Physiologie herangezogen werden, mit denen der entwickelte Ansatz weiter untersucht werden sollte. Erst durch diese Betrachtungen ist eine weitere physisch und psychisch fundierte Bewertung möglich.

## Abbildungsverzeichnis

2.1	Navigationmesh (farblich) als Triangulierung zur Absteckung eines Raumes . . . . .	9
2.2	Beispiel eines einfach verständlichen endlichen Automaten . . . . .	11
2.3	Kollisionsvermeidung beschrieben nach Musse u. Thalmann (1997). Bilderquelle: Musse u. Thalmann (1997) . . . . .	16
2.4	Kollisionsvermeidung beschrieben nach Reynolds (1999, 2004). Bilderquelle: Reynolds (1999) . . . . .	17
2.5	Wahrnehmungsbereich eines Agenten und Verhältnis zu benachbarten Agenten zur Kollisionsvermeidung beschrieben nach Sullivan (2009). Bilderquelle: Sullivan (2009) . . . . .	18
3.1	Agenten bewegen sich durch einen Korridor während ein Alarm ertönt. Bilderquelle: Kim u. a. (2012). . . . .	21
3.2	Agenten evakuieren ein Bürogebäude auf Grund von Feuer. Bilderquelle: Kim u. a. (2012) . . . . .	22
3.3	Ein Agent passiert eine Gruppe von Agenten und drückt diese weg. Bilderquelle: Kim u. a. (2013) . . . . .	23
3.4	Agenten bewegen sich durch einen mit Objekten (Rigid Bodies) gefüllten Raum. Bilderquelle: Kim u. a. (2013) . . . . .	24
4.1	Anforderungen eines Agenten in einer agentenbasierten Menschenmengen-Simulation . . . . .	27
5.1	Das Modell des Agenten aus den ersten Versuchen (links) mit Skelett zur Steuerung des Modells (mitte) und das Skelett bestehend aus Rigid Bodies zur Steuerung des ersten Skeletts (rechts) . . . . .	37
5.2	Agent des ersten Prototypen . . . . .	38
5.3	Agent des ersten Prototypen kollidiert mit niedriger Wand . . . . .	40
5.4	Modell und physikalische Repräsentation eines Agenten des zweiten Prototyps, links Agent mit Kapsel, rechts Kapsel allein . . . . .	41
5.5	Das Sichtfeld eines Agenten als zweidimensionale Glockenform . . . . .	42
5.6	Standard Navigationskomponenten eines Agenten und seine Schnittstelle zur Umgebung . . . . .	43
5.7	Beispiel einer Pfadverfolgung mit interpoliertem Wegpunkt (Blau) eines Agenten (Grün) um eine Wand (Braun) herum, in der durch ein Navigationmesh (Orange) beschriebenen Umgebung . . . . .	45

5.8	Kollisionsvermeidungskomponente eines Agenten und dessen Schnittstelle zur Umgebung . . . . .	49
5.9	Ermittlung des Schnittpunktes der Geschwindigkeiten zweier Agenten und der Position des zu kollidierenden Agenten in der Zukunft . . . . .	52
5.10	Stressniveau und Gesundheit eines Agenten und dessen Auswirkung auf das Fortbewegungsverhalten . . . . .	55
5.11	Visualisierung der psychischen Zustände eines Agenten, links: Hohes Stressniveau und erhöhter Gesundheitswert, rechts: Beide Werte nicht erhöht . . . . .	56
6.1	Szenario 1: Zwei Gruppen von Agenten versuchen ein Ziel auf der anderen Seite eines Tunnels zu erreichen, während sie sich gegenseitig ausweichen . . . . .	63
6.2	Szenario 1: Beide Gruppen von Agenten haben ihr Ziel erreicht und drängen und drücken sich. . . . .	64
6.3	Szenario 2: Zwei Gruppen von Agenten versuchen sich einen Weg durch einen eingezäunten Tunnel zu erarbeiten. . . . .	65
6.4	Szenario 2: Agenten schieben die Absperrzäune beiseite . . . . .	66
6.5	Szenario 2: Agenten stoßen einen Absperrzaun um . . . . .	67
6.6	Szenario 3: Agenten versuchen vergeblich ein verschlossenes Tor zu passieren und sammeln sich davor . . . . .	68
6.7	Szenario 3: Agenten fallen im vorderen Bereich um, da Agenten von hinten gegen die Menge drücken . . . . .	70
6.8	Szenario 4: Vergleich zwischen dem Grundriss der Loveparade 2010 und dem Aufbau des durchgeführten Testszenarios . . . . .	72
6.9	Szenario 4: Große Ansammlung und Gedränge von Agenten in der Mitte der Kreuzung . . . . .	73
6.10	Szenario 5: Agenten bewegen sich wie Fußgänger auf einem Platz . . . . .	75
6.11	Szenario 6: Ausgangssituation des Gebäudes mit den drei Gruppen von Agenten . . . . .	77
6.12	Szenario 6: Anhäufung von Agenten aus der ersten Simulation . . . . .	78
6.13	Szenario 6: Ausbruch des Feuers und evakuierende Agenten . . . . .	79
6.14	Extrapolation der Komplexität der physikalischen Repräsentation eines Agenten (Joints sind Blau dargestellt) . . . . .	82

## **Pseudocode-Verzeichnis**

5.1	Steering Behavior: Suchen . . . . .	46
5.2	Berechnung der Fortbewegung eines Agenten . . . . .	47
5.3	Berechnung der Rotation eines Agenten . . . . .	48
5.4	Kollisionsvermeidung mit statischen Objekten . . . . .	50
5.5	Kollisionsvermeidung mit dynamischen Objekten wie Agenten . . . . .	51
5.6	Auswirkungen und Berechnungen des Gesundheitswertes . . . . .	57
5.7	Auswirkungen und Berechnungen des Stressniveaus . . . . .	59

## Literatur

- [Challenger u. a. 2009] CHALLENGER, R ; CLEGG, CW ; ROBINSON, MA ; LEIGH, M: Understanding crowd behaviours: simulation tools. In: UK Cabinet Office (2009)
- [Creative Bloq 2013] CREATIVE BLOQ (Hrsg.): Behind the scenes of World War Z. Version: Oktober 2013. <http://www.creativebloq.com/3d/behind-scenes-world-war-z-10134703>, Abruf: 10. Dezember. 2013
- [Demyen u. Buro 2006] DEMYEN, Douglas ; BURO, Michael: Efficient triangulation-based pathfinding. In: AAAI Bd. 6, 2006, S. 942–947
- [EA SPORTS FOOTBALL WORLD 2011] EA SPORTS FOOTBALL WORLD (Hrsg.): FIFA 12 REVOLUTIONIZED WITH NEW PLAYER IMPACT ENGINE. Version: May 2011. <http://www.ea.com/news/fifa12-player-impact-engine-01>, Abruf: 10. Dezember. 2013
- [Hartig 2011] HARTIG, J: Stress-Begriffe und Definition, Stressreaktion und Stressoren, Erfassungsmethoden, Selbsthilfe und Diagnostik im WWW, Diplomarbeit, Universität Leipzig, Institut für Psychologie., Diss., 2011
- [Helbing 1998] HELBING, Dirk: A fluid dynamic model for the movement of pedestrians. In: arXiv preprint cond-mat/9805213 (1998)
- [Helbing u. Molnar 1995] HELBING, Dirk ; MOLNAR, Peter: Social force model for pedestrian dynamics. In: Physical review E 51 (1995), Nr. 5, S. 4282
- [Helbing u. Mukerji 2012] HELBING, Dirk ; MUKERJI, Pratik: Crowd disasters as systemic failures: analysis of the Love Parade disaster. In: EPJ Data Science 1 (2012), Nr. 1, S. 1–40
- [Hughes 2002] HUGHES, Roger L.: A continuum theory for the flow of pedestrians. In: Transportation Research Part B: Methodological 36 (2002), Nr. 6, S. 507–535
- [Janke 1976] JANKE, Wilhelm: Psychophysiologische Grundlagen des Verhaltens. In: Medizinische Psychologie. Springer, 1976, S. 1–101
- [Kallmann 2010] KALLMANN, Marcelo: Shortest paths with arbitrary clearance from navigation meshes. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation Eurographics Association, 2010, S. 159–168

- [Kim u. a. 2013] KIM, Sujeong ; GUY, Stephen J. ; MANOCHA, Dinesh: Velocity-based Modeling of Physical Interactions in Multi-agent Simulations. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York, NY, USA : ACM, 2013 (SCA '13), 125–133
- [Kim u. a. 2012] KIM, Sujeong ; GUY, Stephen J. ; MANOCHA, Dinesh ; LIN, Ming C.: Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games ACM, 2012, S. 55–62
- [Lamarche u. Donikian 2004] LAMARCHE, Fabrice ; DONIKIAN, Stéphane: Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In: Computer Graphics Forum Bd. 23 Wiley Online Library, 2004, S. 509–518
- [Liu u. a. 2013] LIU, Libin ; YIN, KangKang ; WANG, Bin ; GUO, Baining: Simulation and control of skeleton-driven soft body characters. In: ACM Transactions on Graphics (TOG) 32 (2013), Nr. 6, S. 215
- [Macal u. North 2011] MACAL, Charles M. ; NORTH, Michael J.: Introductory tutorial: Agent-based modeling and simulation. In: Simulation Conference (WSC), Proceedings of the 2011 Winter IEEE, 2011, S. 1451–1464
- [McShaffry u. Graham 2009] MCSHAFFRY, Mike ; GRAHAM, David: Game coding complete. Charles River Media, 2009
- [Muico u. a. 2011] MUICO, Uldarico ; POPOVIĆ, Jovan ; POPOVIĆ, Zoran: Composite Control of Physically Simulated Characters. In: ACM Transactions on Graphics 30 (2011), Nr. 3
- [Musse u. Thalmann 1997] MUSSE, Soraia R. ; THALMANN, Daniel: A model of human crowd behavior: Group inter-relationship and collision detection analysis. In: Computer Animation and Simulation'97. Springer, 1997, S. 39–51
- [Mustapha u. Bade 2009] MUSTAPHA, Noralizatul A. ; BADE, Abdullah: Collision avoidance series of technique for crowd simulation. (2009)
- [Nareyek 2004] NAREYEK, Alexander: AI in computer games. In: Queue 1 (2004), Nr. 10, S. 58

- [Pan u. a. 2005] PAN, Xiaoshan ; HAN, Charles S. ; LAW, Kincho H.: A multi-agent based simulation framework for the study of human and social behavior in egress analysis. In: Proceedings of the ASCE International Conference on Computing in Civil Engineering Bd. 92, 2005
- [Pelechano u. a. 2007] PELECHANO, Nuria ; ALLBECK, Jan M. ; BADLER, Norman I.: Controlling individual agents in high-density crowd simulation. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation Eurographics Association, 2007, S. 99–108
- [Reynolds 1999] REYNOLDS, Craig: Steering behaviors for autonomous characters. In: Game developers conference Bd. 1999, 1999, S. 763–782
- [Reynolds 2004] REYNOLDS, Craig ; REYNOLDS (Hrsg.): Steering Behaviors For Autonomous Characters. Version: June 2004. <http://www.red3d.com/cwr/steer/>, Abruf: 12. Dezember. 2013
- [Rymill u. Dodgson 2005] RYMILL, Stephen J. ; DODGSON, Neil A.: A psychologically-based simulation of human behaviour. In: Theory and practice of computer graphics Citeseer, 2005, S. 35–42
- [Schröder 1996] SCHRÖDER, H: Psychologische Interventionsmöglichkeiten bei Stressbelastungen. In: Intervention zur Gesundheitsförderung für Klinik und Alltag 1 (1996), S. 7–26
- [Selye 1956] SELYE, Hans: The stress of life. New York, NY, USA : US: McGraw-Hill, 1956
- [Seugling u. Rolin 2006] SEUGLING, Axel ; ROLIN, Martin: Evaluation of physics engines and implementation of a physics module in a 3d-authoring tool. In: Umea University (2006)
- [Stevens 1957] STEVENS, Stanley S.: On the psychophysical law. In: Psychological review 64 (1957), Nr. 3, S. 153
- [Sueddeutsche.de 2012] SUEDEDEUTSCHE.DE (Hrsg.): Heysel, Hillsborough, Bradford - Stadionkatastrophen im Fußball. Version: Februar 2012. <http://www.sueddeutsche.de/sport/stadionkatastrophen-im-fussball-heysel-hillsborough-bradford-1.1274006>, Abruf: 30. Dezember. 2013

- [Sullivan 2009] SULLIVAN, Barry O.: Dynamic Object Avoidance. Version: 2009.  
<https://www.scss.tcd.ie/postgraduate/msciet/current/Dissertations/0809/OSullivan.php>
- [Van Goethem 2012] VAN GOETHEM, AI: A Stream algorithm for crowd simulation to improve crowd coordination at all densities, Master thesis Utrecht University, Diss., 2012
- [Winter 2012] WINTER, Hugo: Modelling Crowd Dynamics During Evacuation Situations Using Simulation. (2012)
- [Wu u. Popović 2010] WU, Jia-chi ; POPOVIĆ, Zoran: Terrain-adaptive bipedal locomotion control. In: ACM Transactions on Graphics 29 (2010), Jul., Nr. 4, S. 72:1–72:10