

Zellulärer Zustandsautomat

3. Projekt zu Modellierung und Simulation

Daniel Graf, Dimitrie Diez, Arne Schöntag, Peter Müller

Inhaltsverzeichnis

1	Einführung	2
2	Beschreibung des Modells	2
3	Anforderungen/Requirements	3
3.1	Allgemeines	3
3.2	Zellen	3
3.3	Hindernisse	3
3.4	Ziele	4
3.5	Personen	4
3.6	Geschwindigkeit der Personen	5
3.7	Zeitmodellierung	5
3.8	Parameter	5
3.9	Berechnungen	6
3.10	Report	6
3.11	Berechnung des Zielnutzens	6
3.12	Werte für die Auswertung	7
3.13	Testszenarien für die Simulation	7
3.13.1	Erstellung der Testszenarien	7
3.13.2	Testszenario: Freier Fluss	7
3.13.3	Testszenario: Hühnertest	8
3.13.4	Testszenario: Evakuierung eines Raumes mit 2 Türen	8
3.13.5	Testszenario: Evakuierung eines Raumes mit 4 Türen	9
3.13.6	Testszenario: Fundamentaldiagramm - Rimea-Test 4	10
3.13.7	Testszenario: Engstelle (unidirektional)	10
3.14	Visualisierung	11
4	Beeinflussung durch andere Personen	12
5	Softwaredesign	13

6	Softwaretest	13
6.1	Verifikation	13
6.2	Validation	13
7	Ausblick und Fazit	13

1 Einführung

Im Zuge der ersten Studienarbeit wurde das Laufverhalten von Probanden in der Ebene und auf der Treppe untersucht. Basierend auf den Erkenntnissen bezüglich der individuellen Wunschgeschwindigkeiten werden in dieser Studienarbeit Personenbewegungen in der Ebene simuliert. Mit Hilfe solcher Simulationen können, beispielsweise im Zuge von Gebäudeplanungen unterschiedliche Gebäude- und Raumgestaltungen simuliert und hinsichtlich schnellster Räumungen im Krisenfall optimiert werden.

2 Beschreibung des Modells

Für die Simulation der Personenbewegungen wird ein zellulärer Zustandsautomat implementiert. Jede Zelle kann entweder leer oder durch eine Person oder ein Hindernis besetzt sein. Sie kann außerdem ein Ziel oder eine Quelle enthalten. Ein Hindernis kann von keiner Person betreten werden. Die Personen versuchen im Laufe der Simulation von der Quelle (Startposition) zum Ziel zu gelangen. Hierbei können sie sich in der sog. Moore-Umgebung bewegen. Es kann jede Zelle betreten werden, die frei ist und eine Nachbarzelle der aktuellen Zelle ist. Als Nachbarzelle wird jede Zelle bezeichnet, welche mit der aktuellen Zelle eine Kante oder Ecke teilt.

Die Ziele werden als attraktiv für die Personen angesehen. Die Personen steigern ihren persönlichen Nutzen, je näher sie dem Ziel kommen. In der Realität fühlt sich eine Person jedoch unwohl, wenn ihr eine fremde Person zu nahe kommt. Im Modell wird dies dadurch berücksichtigt, dass sich der Nutzen einer Person verringert, wenn sie einer anderen zu nahe kommt. Um die Simulation möglichst realistisch zu gestalten, bewegen sich die Personen mit einer individuellen Wunschgeschwindigkeit. Diese wird aus den Ergebnissen der ersten Studienarbeit ermittelt.

Das erläuterte Modell gibt die einzelnen Personenbewegungen aus und visualisiert diese zusätzlich. Die Bewegung der Personen wird in drei getrennten Simulationen mit jeweils unterschiedlichen Algorithmen berechnet. Zunächst wird für jedes Feld der negative euklidische Abstand als Zielnutzen berechnet. Jede Person versucht durch die Wahl des umliegenden Feldes mit dem maximalen Wert seinen Nutzen zu steigern.

Darüber hinaus besteht die Möglichkeit die Personenbewegung mit Hilfe des Floor-Flooding, basierend auf dem Dijkstra Algorithmus zu ermitteln. Jede Person wählt von den anliegenden Feldern das Feld, welches die geringste Anzahl an notwendigen Bewegungen bis zum Ziel hat.

Als dritte Variante erfolgt die Ermittlung der Personenbewegung ebenfalls mit Hilfe des Floor-Flooding. Als Grundlage dient hierbei jedoch die Lösung der Eikonal-Gleichung, welche die Ausbreitung einer Welle beschreibt. Hierfür wird der Fast-Marching Algorithmus verwendet.

Im Laufe der Studienarbeit werden die beschriebenen Modelle anhand unterschiedlicher Tests verglichen und ausgewertet. Eine detaillierte Erläuterung der Tests ist im folgenden Kapitel 3.13 beschrieben.

3 Anforderungen/Requirements

3.1 Allgemeines

Es soll ein zellulärer Zustandsautomat implementiert werden. Die Software soll Personenbewegungen in der Ebene simulieren. Die unterschiedlichen Szenarien (Zustand der Zellen zu Beginn der Simulation), wie beispielsweise die Position der Hindernisse werden der Software als .png Bilddatei übergeben. Diese sind im Unterkapitel 3.13 näher erläutert. Es wird davon ausgegangen, dass sich zu Beginn der Simulation alle Personen im Startbereich (es soll möglich sein mehrere Startbereiche anzulegen) befinden.

3.2 Zellen

Das gesamte Feld (zweidimensionaler Bereich indem sich die Personen bewegen können) soll in Zellen eingeteilt werden. Die Größe des Feldes und die Anzahl an Zellen sollen der Software als Parameter übergeben werden.

Jede Zelle soll quadratisch sein. Jede Zelle kann verschiedene Zustände haben. Sie kann entweder leer oder besetzt sein. Ist eine Zelle besetzt, wird noch weiter unterschieden, ob sie durch ein Hindernis, das Ziel- (Personensenke) oder Startfeld (Personenquelle) oder durch eine Person besetzt ist.

3.3 Hindernisse

Es soll davon ausgegangen werden, dass jedes Hindernis eine oder mehrere Zellen ganz belegt. Hindernisse können von den Personen nicht betreten werden. Falls sich ein Hindernis zwischen einer Person und ihrem Ziel befindet, ist die Person gezwungen um das Hindernis herum zu gehen.

In der Realität versucht eine Person einem Hindernis möglichst frühzeitig auszuweichen. Sie senkt somit ihren Nutzen, je näher sie einem Hindernis kommt. Dies wird im weiteren Verlauf dieser Studienarbeit nicht berücksichtigt.

3.4 Ziele

Durch das entsprechende Testszenario werden ein oder mehrere Ziele definiert. Ziele werden als attraktiv für die Personen angesehen. Jede Person erhöht ihren Nutzen, wenn sie sich dem Ziel nähert. Befinden sich mehrere Ziele im Feld, wird die Wahl des Ziels für jede Person abhängig vom Betriebsmodus bestimmt.

3.5 Personen

Jede Person soll in ihrem Startfeld starten und sich im Laufe der Simulation von Zelle zu Zelle auf ihr Ziel hin bewegen und dadurch ihren persönlichen Nutzen, je näher sie dem Ziel kommt, steigern. Jede Person kann sich in der Moore-Umgebung (8 mögliche Bewegungsrichtungen) bewegen. Die möglichen Bewegungsrichtungen sind in Abbildung 1 rot markiert.

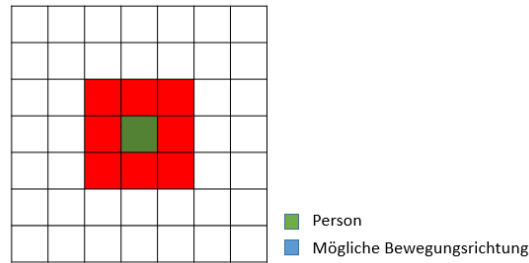


Abbildung 1: Moore-Umgebung

Die Personen können nur freie Zellen betreten. Jede Person wählt genau die Zelle als nächste Zelle, welche ihr den steilsten Nutzenanstieg ermöglicht. Der Nutzen der verschiedenen Zellen wird abhängig vom verwendeten Algorithmus unterschiedlich berechnet. Darüber hinaus soll der Nutzen durch die Position der anderen Personen beeinflusst werden.

Hierfür wird zwischen zwei Bereichen, abhängig vom Abstand zur nächsten Person unterschieden. Beträgt der Abstand zu einer fremden Person weniger als $45cm$, wird der vertraute Bereich (intimate space) der Personen betreten. Beträgt der Abstand zwischen $45cm$ und $120cm$, wird der persönliche Bereich (personal space) der Personen betreten. Für den vertrauten Bereich ist der Nutzenverlust ($N_v(x)$), abhängig vom Abstand x , nach Funktion 2 definiert. Für den persönlichen Bereich wird der Nutzenverlust ($N_p(x)$), wiederum abhängig vom Abstand x zur anderen Person, durch die Funktion 1 definiert.

$$N_p(x) = \mu * \exp\left(\frac{4}{\left(\frac{d(x)}{\delta_{p*r}}\right)^2 - 1}\right) \quad (1)$$

$$Nv(x) = Np + \frac{\mu}{a_p} * \exp\left(\frac{4}{\left(\frac{d(x)}{\delta_v * r}\right)^{2*b_p} - 1}\right) \quad (2)$$

$d(x)$ beschreibt den aktueller Abstand zur nächsten Person. Die folgende Tabelle enthält die Bedeutung der einzelnen Parameter und die entsprechenden Werte:

Parameter	Beschreibung	Wert
μ	Stärke des Nutzenverlustes	5.0
a_p	Dämpfung zwischen vertrautem und persönlichem Bereich	1.0
b_p	Stärke des Übergangs zwischen vertrautem und persönlichen Bereich	1
δ_p	persönlicher Bereich [m]	1.20
δ_v	vertrauter Bereich [m]	0.45
r	Radius einer Person [m]	0.2

Tabelle 1: Parameter und Werte für die Berechnung des Nutzenverlustes

Der Radius einer Person sowie die Abstände für den vertrauten und persönlichen Bereich wurden auf Basis des Modells von Hall aus [2] übernommen. Weitere Erläuterungen zu den Parametern, den Werten und den verwendeten Funktionen sind im Kapitel 4 beschrieben.

3.6 Geschwindigkeit der Personen

Jede Person soll eine individuelle Wunschgeschwindigkeit besitzen. Sie bewegt sich bei freier Bahn mit dieser Geschwindigkeit fort. Die Wunschgeschwindigkeiten sollen als normalverteilt angesehen werden. Als Mittelwert und Standardabweichung sollen die errechneten Werte aus der ersten Studienarbeit verwendet werden.

3.7 Zeitmodellierung

Die Simulation unterliegt keinen harten Echtzeitanforderungen. Sie soll für die gesamte Simulationsdauer so schnell wie möglich durchgeführt werden. Es soll keine globale Uhr modelliert werden.

Die Visualisierung der Simulation soll hingegen die Bewegungen der Personen möglichst in Echtzeit darstellen. Sie unterliegt weichen Echtzeitanforderungen. Abweichungen aufgrund mangelnder Rechner-Ressourcen sind tolerierbar.

3.8 Parameter

Die Tabelle 2 zeigt alle möglichen Parameter, welche die Software steuern. Alle Parameter sind optional.

Parameter	Beschreibung	Default
<code>-free-flow-velocity</code>	Mittelwert der Wunschgeschwindigkeit [m/s]	1,0
<code>-free-flow-deviation</code>	Standardabweichung der Wunschgeschwindigkeiten [m/s]	0
<code>-algorithm</code>	Algorithmus (euclid, dijkstra, fast-marching)	dijkstra
<code>-output-folder</code>	Ordner in den der Output gespeichert wird	../data/
<code>-cellsize</code>	Größe der einzelnen Zellen [m]	1
<code>-input-map</code>	Pfad zum Bild der map (Auswahl des Testfalls)	default.png

Tabelle 2: Parameter und Defaultwerte

3.9 Berechnungen

Die Aufgabe der Software ist die Simulation von Personenbewegungen. Für möglichst genaue Zeitangaben wird die Java Klasse `BigDecimal` verwendet. Für die Division werden 32 Nachkommastellen berücksichtigt und als Rundungsmodus wird `Half_Even` verwendet, da dieser kumulative Fehler bei sich wiederholenden Berechnungen minimiert. Die anschließende Berechnung und Aufbereitung der Daten (insbesondere für den Test 3.13.6) erfolgt mit Mathematica.

3.10 Report

Alle errechneten Ergebnisse sollen über einen Report ausgeleitet werden. Der Report soll im `.xml` Format ausgeleitet werden um eine gute Weiterverarbeitung der Ergebnisse (Visualisierung) zu gewährleisten. Die Datei `output.xml` soll alle benötigten Informationen, wie beispielsweise die einzelnen Events der Personen, die durch den Algorithmus berechneten Entfernungen der einzelnen Zellen vom Ziel und eine Übersicht über den Zustand der einzelnen Zellen (Fieldmap), für Simulation beinhalten.

Die Wunschgeschwindigkeiten werden zufällig generiert. Diese sollen normalverteilt sein. Für eine Überprüfung der Zufallszahlgenerierung sollen die Wunschgeschwindigkeiten und die ID der entsprechenden Person im `.csv` Format in der Datei `velocity.csv` ausgegeben werden.

3.11 Berechnung des Zielnutzens

Es sollen drei unterschiedliche Algorithmen für die Berechnung des Zielnutzens (Attraktivität) einer Zelle implementiert werden.

Algorithmus 1: Der Nutzen bzw. die Attraktivität jeder Zelle entspricht dem negativen euklidischen Abstand. Hierbei können keine Hindernisse umgangen werden. Jede Person versucht den Nutzen zu maximieren und wählt jeweils die Zelle mit maximalen Wert aus den umliegenden aus.

Algorithmus 2: Der Nutzen bzw. die Attraktivität jeder Zelle wird durch Floor-Flooding, basierend auf dem Dijkstra Algorithmus, berechnet. Für jede Zelle wird die geringste Anzahl an notwendigen Bewegungen bis zum Ziel ermittelt.

Algorithmus 3: Analog zum Algorithmus 2 wird auch in diesem hier der Nutzen bzw. die Attraktivität jeder Zelle mittels Floor-Flooding berechnet. Es wird jedoch der Fast-Marching Alforithmus (Lösung der Eikonal-Gleichung) als Grundlage verwendet.

3.12 Werte für die Auswertung

Alle in der Tabelle 3 aufgeführten Werte sollen auf Basis der Simulation ermittelt und ausgewertet werden. Einige Werte werden nur für bestimmte Tests benötigt.

Ermittelte Werte	Beschreibung	Wann benötigt
mittlere Geschwindigkeit	durchschnittliche Geschwindigkeit der Personen im Feld	Rimea-Test 4
Fluss (Personen/Meter/s)	Anzahl der Personen mit Geschwindigkeit > 1m/s im Feld	Rimea-Test 4
Simulationsdauer	Dauer bis der Test beendet ist	bei allen Tests

Tabelle 3: Werte für die Auswertung

3.13 Testszenarien für die Simulation

Die implementierten Algorithmen (vgl. Unterkapitel 3.11) sollen anhand unterschiedlicher Testszenarien verglichen und verifiziert werden. Die verschiedenen Szenarien sollen der Simulation als .png Datei übergeben werden.

3.13.1 Erstellung der Testszenarien

Es soll die Möglichkeit bestehen, unterschiedliche Testszenarien mittels eines Bildbearbeitungsprogrammes zu erstellen und an die Simulation (als .png Datei) zu übergeben. Bei der Erstellung ist sowohl die Anzahl der Personen, als auch die Anzahl der Hindernisse und Ziele (Personensenken) beliebig. Hinsichtlich der Farben der Objekte müssen jedoch die in der folgenden Tabelle 4 beschriebenen Farbcodes verwendet werden. Jedes Pixel wird durch eine Zelle des Automaten repräsentiert.

3.13.2 Testszenario: Freier Fluss

Im Zuge dieses Tests soll überprüft werden, ob sich Personen bei freier Bahn (keine Hindernisse zwischen Start und Ziel) mit ihrer jeweiligen Wunschgeschwindigkeit auf das Ziel hin bewegen. Des weiteren soll überprüft werden, ob die Person den kürzesten Weg

Farbcode	Farbe	Bedeutung
#3F48CC	Blau	Ziel/ Personensenke
#000000	Schwarz	Ziel/ Personensenke
#FFFFFF	Weiß	Leere Zelle
#22B14C	Grün	Person

Tabelle 4: Werte für die Auswertung

verwendet. Der Test soll mit allen 3 Betriebsmodi durchgeführt werden. Abbildung 2 stellt den Testaufbau schematisch dar.



Abbildung 2: Testfall Freier Fluss (schematisch)

3.13.3 Testszenario: Hühnertest

Zwischen Start und Ziel soll ein U-förmiges Hindernis (Öffnung in Richtung des Startes) eingefügt werden. Abhängig vom Betriebsmodus wird ein unterschiedliches Ergebnis erwartet:

Betriebsmodus 1: Die Personen sollen sich auf dem kürzesten Weg auf das Ziel zubewegen. Sobald sie das Hindernis erreicht haben, sollen sie stehen bleiben. Erwartungsgemäß wird der Test nicht erfolgreich verlaufen.

Betriebsmodi 2 und 3: Die Personen sollen das Hindernis umgehen und das Ziel erreichen. Bei einem „Feststecken im Hindernis“ (wie im Betriebsmodus 1 gefordert) wird der Test als fehlgeschlagen gewertet. Abbildung 3 stellt den Testaufbau schematisch dar.

3.13.4 Testszenario: Evakuierung eines Raumes mit 2 Türen

Die Personen befinden sich zu Beginn in einen von Hindernissen umgebenen, quadratischen Raum. An 2 Seiten sind Öffnungen platziert (freie Zellen). Hinter diesen Öffnungen befindet sich jeweils ein Ziel. Im Laufe des Tests sollen die Personen durch die beiden

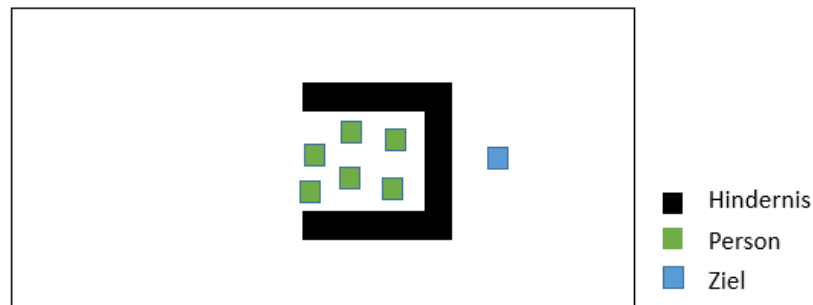


Abbildung 3: Testfall Hühnertest (schematisch)

Engstellen zum Ziel gelangen.

Dieser Test soll mit den Betriebsmodi 2 und 3 in unterschiedlichen Versionen durchgeführt werden. Zunächst sollen sich die Türen an benachbarten Seiten befinden. Hierbei sollen sie einmal mit geringem und einmal mit sehr großem Abstand zueinander platziert werden. Anschließend sollen die Türen an gegenüberliegenden Seiten platziert mittig werden. Unterschiede hinsichtlich der Evakuierungsdauer sind festzuhalten. Die Abbildungen 4, 5 und 6 zeigen die unterschiedlichen Testversionen schematisch.

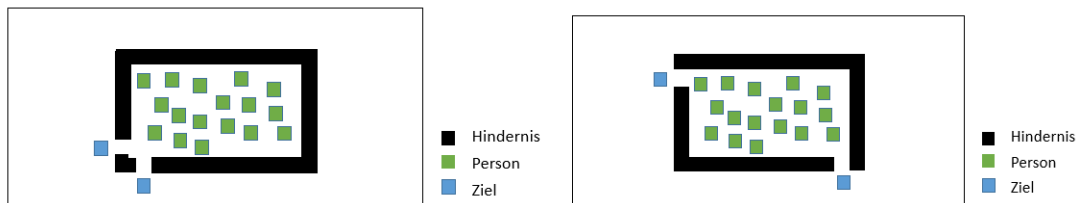


Abbildung 4: Testfall Evakuierung eines Raumes mit 2 Türen (Nebeneinander, geringer Abstand)

Abbildung 5: Testfall Evakuierung eines Raumes mit 2 Türen (Nebeneinander, großer Abstand)

3.13.5 Testszenario: Evakuierung eines Raumes mit 4 Türen

Dieser Testfall entspricht der in Kapitel 3.13.4 beschriebenen Evakuierung mit 2 Türen. In diesem Fall befindet sich jedoch an jeder Raumseite eine Tür. Die Türen sollen sich in einer ersten Version mittig bezogen auf die jeweilige Raumseite befinden. Darüber hinaus sollen in einer weiteren Version die Auswirkungen ermittelt werden, wenn jeweils die 2 Türen an benachbarten Seiten mit minimalen Abstand zueinander platziert werden. Unterschiede hinsichtlich der Evakuierungsdauer sind festzuhalten. Aufgrund der Analogie zur Evakuierung mit 2 Türen wird auf eine graphische Veranschaulichung des Testaufbaus

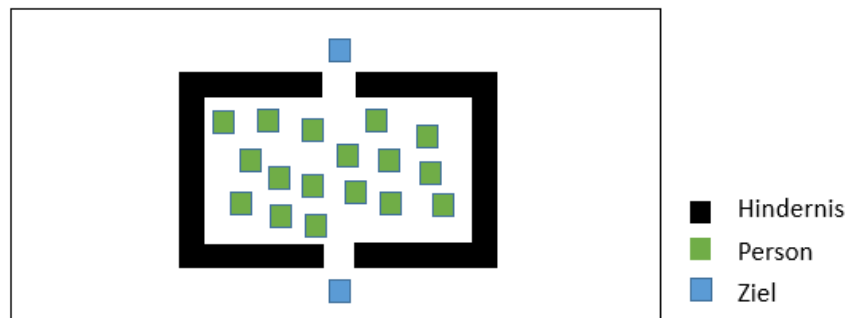


Abbildung 6: Testfall Evakuierung eines Raumes mit 2 Türen (Gegenüber)

verzichtet.

3.13.6 Testszenario: Fundamentaldiagramm - Rimea-Test 4

In diesem Testfall soll ein $65m$ langer und $12m$ breiter Gang angelegt werden. Auf der einen Seite des Ganges befindet sich die Personenquelle, auf der anderen Seite die Personensenke. Im Laufe des Testes sollen die Personen den Gang entlangschreiben. Die Anzahl der Personen ist zu variieren. In einer weiteren Version des Tests sollen Personen, welche die Personensenke erreicht haben, erneut bei der Personenquelle starten (Endlosschleife). Abbildung 7 zeigt den Testaufbau schematisch.



Abbildung 7: Testfall Rimea 4 (schematisch)

3.13.7 Testszenario: Engstelle (unidirektional)

In diesem Testfall soll, analog zum Rimea-Test 4, ein $65m$ langer und $12m$ breiter Gang angelegt werden. Am Ende des Ganges befindet sich eine Engstelle mit $6m$ Breite und dahinter die Personensenke. Es soll mit variierender Anzahl an Personen getestet werden,

wie lange es dauert bis die Engstelle durchschritten wurde. Der schematische Testaufbau ist in Abbildung 8 aufgeführt.

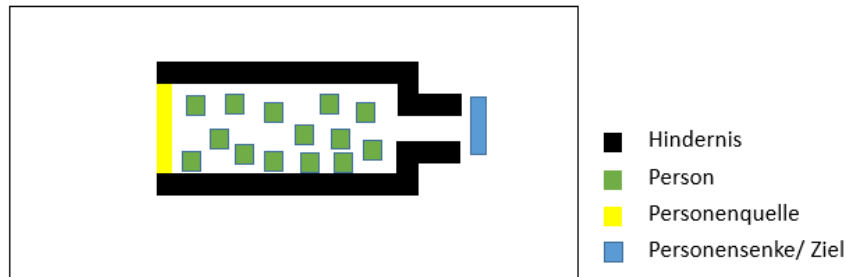


Abbildung 8: Testfall Engstelle (schematisch)

3.14 Visualisierung

Die Visualisierung soll die errechneten Personenbewegungen graphisch darstellen. Als Grundlage soll der ausgegebene Report (vgl. Abschnitt 3.10) dienen. Konkrete Anforderungen bezüglich der Visualisierung sind in Abbildung 9 abgebildet.

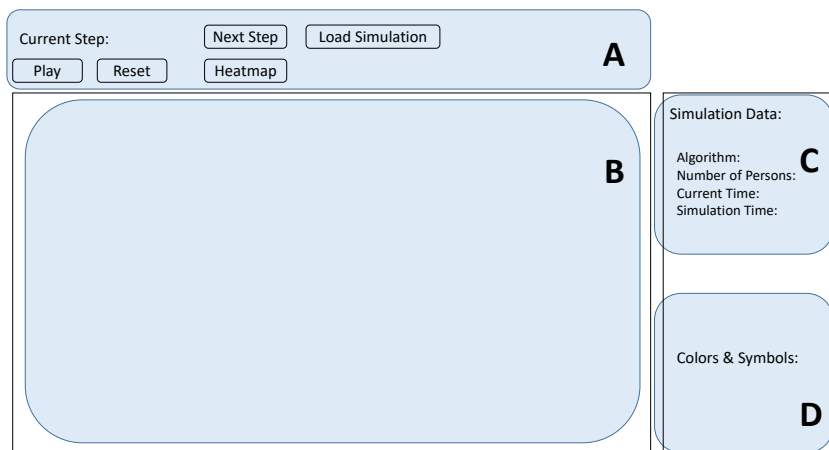


Abbildung 9: Visualisierung (schematisch)

Die Oberfläche der Visualisierung soll aus 4 verschiedenen Teilen bestehen, welche in der Abbildung durch die blauen Kästen A-D hervorgehoben wurden. Im Teil A sollen Konfigurationsmöglichkeiten für die Visualisierung aufgeführt sein. Beispielsweise soll zwischen einer normalen Darstellung der Simulation und einer Heatmap gewechselt werden können. Darüber hinaus soll die Möglichkeit bestehen, die Visualisierung zu

stoppen und schrittweise fortzusetzen.

Teil B enthält die eigentliche Visualisierung der Personen, Hindernisse und Ziele. In Teil C sind alle Informationen bezüglich der Visualisierung aufgeführt und Teil D enthält eine Legende der verwendeten Symbole und Farben.

4 Beeinflussung durch andere Personen

Wie bereits im Abschnitt 3.5 kurz erläutert, wird die Entscheidung, welche Zelle eine Person als nächstes betritt abhängig vom verwendeten Algorithmus und von den Positionen der anderen Personen getroffen. In der Realität fühlen sich Personen unwohl, wenn sie anderen, insbesondere fremden Personen zu nahe kommen. Personen versuchen somit eine solche Situation wenn möglich zu vermeiden. Beispielsweise bei der Fahrt in einem überfüllten Aufzug lässt sich diese Situation nicht vermeiden, doch jede Person ist froh wenn sie ihr Stockwerk erreicht und die Situation vorbei ist. In der Ebene sind solche Situationen jedoch häufig vermeidbar. Personen neigen in der Realität dazu einen Umweg in Kauf zu nehmen, um einem Gedränge zu entkommen. Um dieses Verhalten in der Simulation zu berücksichtigen existieren verschiedene mathematische Funktionen. Mit Hilfe der Funktionen wird das Abstoßungspotenzial ermittelt. Während auf der einen Seite der Nutzen einer Person steigt, je näher sie dem Ziel kommt, verringert sich dieser Nutzen auf der anderen Seite wieder, wenn sie anderen Personen zu nahe kommen. Mit Hilfe von Funktionen kann somit der Nutzenverlust einer Person bestimmt werden.

Insbesondere glatte Funktionen auf kompakten Trägern eignen sich hierfür gut. Hintergrund hierfür ist die Eigenschaft solcher Funktionen, außerhalb eines begrenzten Gebietes komplett zu verschwinden. Wäre dies nicht der Fall, würde eine sehr weit entfernte Person die Entscheidung einer anderen Person beeinflussen. Dies entspricht nicht der Realität. Das Abstoßungspotenzial von Personen ist auf einen Bereich beschränkt.

In der Fachliteratur wird beispielsweise folgende Funktion vorgeschlagen:

$$N(d) = \begin{cases} -h \exp(\frac{1}{(\frac{d}{w})^2 - 1}) & \text{falls } |d| < w \\ 0 & \text{sonst} \end{cases}$$

Hierbei wird der Nutzenverlust ($N(d)$), abhängig vom Abstand d zur nächsten Person definiert. w beschreibt die Reichweite der Funktion und h die Stärke der Beeinflussung bzw. des Nutzenverlustes [1].

Nach Diskussion in der Gruppe haben wir uns jedoch gegen die Verwendung dieser Funktion entschieden.

Für die Ermittlung des Abstoßungspotenzials bzw. des Nutzenverlustes aufgrund anderer Personen wurden stattdessen die in Abschnitt 3.5 aufgeführten Funktionen 1 und 2 verwendet. Diese basieren auf dem Modell von Hall, welches den Bereich um eine Person in mehrere Bereiche einteilt. In einem Umkreis von bis zu 45cm wird der vertraute Bereich einer Person betreten. Befinden sich zwei Personen so nah beieinander, können Berührungen nicht vermieden werden. Zwischen 45cm und 120cm um eine Person befindet sich der persönliche Bereich, welcher beispielsweise von Freunden oder Familienmitgliedern,

jedoch nicht von fremden betreten werden darf. Darüber hinaus definiert Hall noch zwei weitere Bereiche. Zwischen $120cm$ und $360cm$ um die Person befindet sich der soziale oder gesellschaftliche Bereich der Person und ab $360cm$ der öffentliche Bereich. Für die Simulation in dieser Studienarbeit sind die beiden letzten Bereiche jedoch nicht relevant, da der Einfluss auf den Nutzen der Person zu gering ist, um die Entscheidung, wohin die Person als nächstes geht zu beeinflussen. Darüber hinaus lässt sich das betreten dieser beiden Bereiche durch andere Personen bei einer höheren Personendichte generell nicht vermeiden.

Nach unserer Einschätzung und nach [2] führt die Berücksichtigung des Modells von Hall durch die Verwendung unterschiedlicher Funktionen für den vertrauten und persönlichen Bereich zu einem Verhalten, welches der Realität näher kommt. Berührt eine Person eine fremde Person, fühlt sie sich deutlich unwohler, als wenn sie dieser lediglich nahe kommt mit einem Abstand von beispielsweise $50cm$. Diese Eigenschaft wird durch die Funktion 4 kaum berücksichtigt.

Darüber hinaus beinhalten die Funktionen 1 und 2 deutlich mehr Parameter anhand derer sie an die Realität bzw. an potenzielle gesellschaftliche oder länderspezifische Unterschiede angepasst werden können. Für diese Studienarbeit wurden die Parameter wie in der Tabelle 1 aufgeführt gewählt. Grundlage hierfür sind die in [2] erläuterten Erkenntnisse aus unterschiedlichen Versuchen.

5 Softwaredesign

Der Aufbau der Anwendung wurde im Team diskutiert und anschließend mittels UML spezifiziert.

6 Softwaretest

6.1 Verifikation

6.2 Validation

7 Ausblick und Fazit

Literatur

- [1] Felix Dietrich, Gerta Köster, Michael Seitz, and Isabella von Sivers. Bridging the gap: From cellular automata to differential equation models for pedestrian dynamics. *Journal of Computational Science*, 5(5):841 – 846, 2014.

- [2] Isabella von Sivers and Gerta Köster. Dynamic stride length adaptation according to utility and personal space. *Transportation Research Part B: Methodological*, 74:104 – 117, 2015.