

Jihočeská univerzita
Přírodovědecká fakulta
Ústav aplikované informatiky

**Softwarové modelování
davového chování**

Bakalářská práce

Autor: Jiří Moc

Vedoucí práce: Ing. Jiří Jelínek, CSc.

České Budějovice 2017

Bibliografické údaje

Moc J., 2017: Softwarové modelování davového chování | Software modeling of crowd behavior, Bc. Thesis, in Czech | Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Abstrakt:

Práce staví na základech položených v knize Psychologie davu a zabývá se aplikací davového chování v softwarovém prostředí. Teorie je ověřena na těchto experimentech: volný tok davu v prostoru letištního terminálu, koordinovaný tok a evakuační scénář. Experimenty jsou provedeny pomocí multiagentního systému navrženého nad programovacím jazykem Java. Cílem práce je pomocí modelu popsat davové chování, navrhnout jeho implementaci a tu podrobit experimentu. Výsledky jsou poté porovnány s výzkumem na toto téma.

Klíčová slova:

teorie davu, davové chování, multiagentní systém, Java

Abstract:

Thesis represents basics from publication Psychology of crowd and pursues of implementation crowd behavior in software implementation. Theory is validated on following experiments: free crowd flow in airport terminal surrounding, coordinated flow and evacuating scenario. Experiments are executed by multiagent system designed by Java. Main purpose of this thesis is to describe crowd behavior by model, design implementation and process it by experiment. Results are afterwards compared to research.

Keywords:

crowd theory, crowd behavior, multiagent system, Java

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenu a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby stejnou elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátu.

V Českých Budějovicích dne

podpis

Poděkování

Děkuji vedoucímu práce Ing. Jířimu Jelínkovi, CSc. za užitečné rady a připomínky při řešení zadání této bakalářské práce. Také děkuji svým nejbližším za neutuchající podporu a pochopení při jejím psaní.

Obsah

1	Úvod	1
2	Metodika.....	4
3	Davové chování	5
3.1	Masa, dav	5
3.2	Klasifikace davů	6
4	Modelování davových jevů	8
4.1	Měřitelné charakteristiky kvantifikující procesy davové dynamiky.....	8
4.2	Metody modelování davového chování	12
4.3	Nástroje modelování davového chování	17
5	Multiagentní modelování.....	23
5.1	Multiagentní systém.....	23
5.2	Agent.....	25
5.3	Nástroje multiagentního modelování.....	27
5.4	Příklad modelování davu pomocí agentního systému	29
6	Experimentální část	32
6.1	Kritéria pro výběr modelu.....	32
6.2	Konceptuální model vybraných davových jevů.....	35
6.3	Implementace konceptuálního modelu	38
6.4	Experimenty.....	50
6.5	Analýza získaných dat	53
6.6	Porovnání dosažených výsledků se závěry literatury	60
7	Závěr.....	62
8	Seznam literatury	64
9	Seznam tabulek.....	67
10	Seznam obrázků.....	68

1 Úvod

Modelování skupinového chování je v posledních letech spojováno s výzkumy pohybu zvířat, navrhováním bezpečných veřejných prostor a technikami pro vytváření poutavých filmových scén. I když psychologie davů jako taková je zájmem vědců a psychologů už od konce 19. století [1], modely založené na davové psychice zažily boom s vývojem počítačů. Dnešní softwarové prostředky umožňují simulovat detailní n-rozměrné prostředí, vlastnosti každé obsažené entity a vyhodnocovat náročné scénáře, s nízkými náklady a vysokou informační hodnotou. Teorie davové psychologie a dynamiky je základem pro oblast, kterou se zabývá tato práce – *modelováním davového chování*.

Modelování davového chování nalezlo uplatnění nejen při testování prostor, studování vzorů chování, či navrhování městských částí, ale také ve filmovém průmyslu. Tato potřeba vznikla v momentě, kdy bylo nutné zachytit mnohonásobně více osob, než čítalo skutečné obsazení a zároveň modelovat psychiku každého jednotlivce. Animátoři tak vytváří množinu pohybů, stavů a cílů dostupných pro entity a náhodně generují relativně jedinečné jednotlivce, kteří poté postupují podle zadaného scénáře. Poté již není problém modelovat tisíce osob, které navzájem interagují, v rozumném čase.

Tato práce je koncipována ze dvou částí – v první je zpracována teorie davu, davového chování a přehled současného stavu problematiky. Součástí jsou také příklady modelů a popis aplikace teoretických znalostí v softwarových prostředcích. Druhá část je poté experimentální – v této jsou poznatky z první části zúročeny při návrzích modelů davového chování a implementaci podpůrné knihovny, ve které jsou jednotlivé modely podrobeny experimentům.

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Jiří Moc

(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika

Katedra: Ústav aplikované informatiky

Školitel: Ing. Jiří Jelínek, CSc.

(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF: dtto

(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:

(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Softwarové modelování davového chování

Cíle práce:

1. Provedení komentované rešerše k tématu davového chování a modelování davových jevů. Přehled metodologií, algoritmů a nástrojů dotýkajících se tohoto tématu.
2. Vytipování konkrétních davových jevů či situací pro následující body práce.
3. Návrh konceptuálního modelu pro simulaci těchto jevů a vytvoření podpůrné knihovny v jazyce Java.
4. Návrh a implementace softwarových modelů davové dynamiky v jazyce Java.
5. Provedení simulačních experimentů v souladu s bodem 2 zadání. Analytické zpracování výsledků a porovnání výstupů s dostupnými zdroji zabývajícími se obdobnou problematikou.

Základní doporučená literatura:

1. GROW, André, Andreas FLACHE a Rafael WITTEK. An Agent-Based Model of Status Construction in Task Focused Groups. *Journal of Artificial Societies and Social Simulation* [online]. 2015-03-31 [cit. 2015-04-23]. Dostupné z: <http://jasss.soc.surrey.ac.uk/18/2/4.html>
2. MIKŠÍK, Oldřich. *Hromadné psychické jevy: (psychologie hromadného chování)*. Vyd. 1. V Praze: Univerzita Karlova v Praze, nakladatelství Karolinum, 2005, 269 s. ISBN 80-246-0930-4
3. CANETTI, Elias, [z německého originálu ... přeložil Jiří STROMŠÍK a Předmluvu napsal Miloslav PETRUSEK]. *Masa a moc*. Vyd. 2., (V nakl. Academia 1.). Praha: Academia, 2007. ISBN 80-200-1512-4
4. LE BON, Gustave. *Psychologie davu*. Překlad Ladislav Hofman, Zdeněk Ullrich. Praha: KRA, 1994, 159 s. ISBN 80-901-5278-3
5. PATIL, Sachin, Jur VAN DEN BERG, Sean CURTIS, Ming C LIN a Dinesh MANOCHA. Directing Crowd Simulations Using Navigation Fields. *IEEE Transactions on Visualization and Computer Graphics*. 2011, vol. 17, issue 2, s. 244-254. DOI: 10.1109/TVCG.2010.33. Dostupné z: <http://gamma.cs.unc.edu/DCrowd/paper.pdf>

6. KUBÍK, Aleš. *Intelligentní agenty*. Vyd. 1. Brno: Computer Press, 2004, 280 s. ISBN 80-251-0323-4
7. ZBOŘIL, František. *Plánování a komunikace v multiagentních systémech* [online]. Brno, 2004 [cit. 2015-04-23]. 107 s. Dostupné z: http://www.fit.vutbr.cz/research/view_pub.php.cz?id=7608. Disertační práce. Fakulta informačních technologií VUT v Brně.
8. *FIPA: Foundation for Intelligent Physical Agents* [online]. 2014 [cit. 2015-04-23]. Dostupné z: <http://fipa.org/>.
9. ORACLE CORPORATION. *Java™ Platform, Standard Edition 8 API Specification* [online]. 2015 [cit. 2015-04-23]. Dostupné z: <http://docs.oracle.com/javase/8/docs/api/index.html>
10. DURUPINAR, Funda. *From Audiences to Mobs: Crowd Simulation with Psychological Factors* [online]. Bilkent University, 2010 [cit. 2015-04-23]. Dostupné z: <http://www.cs.bilkent.edu.tr/~fundad/RESEARCH/PhDThesis.pdf>. Disertační práce. Bilkent University.

Financování práce :

Vedoucí práce : podpis : 

U externích vedoucích fakultní garant práce.....podpis :

Garant oboru bak.. studia (nepožaduje se u zaměření „příprava na mag. studium biologie)

..... podpis :

Vedoucí katedry (12. UTU FEL) podpis 

Případný souhlas vedoucího ústavu AVpodpis :

V Českých Budějovicích dne

Převzal/a dne..... podpis : 

2 Metodika

1. **Sběr a analýza** dostupné literatury, včetně studia referenčních prací. Zdroje budou vybrány s ohledem na téma práce – tedy takové, které se týkají davů, davového chování a prostorových interakcí mezi jedinci. Podstatnou složkou literatury jsou knižní zdroje, doprovázené aktuálními vědeckými pracemi. Neméně důležitou součástí je nalezení vhodných softwarových prostředků, využívaných při modelování davových jevů – agentní systémy.
2. **Provedení komentované rešerše** pro témata davů, davového chování a multiagentních systémů. Informace, čerpané ze zdrojů a dokumentací, budou vybrány tak, aby tvořily koherentní celek důležitých, navzájem na sebe navazujících, pojmů. Hlavními body rešerše poté budou multiagentní systémy a davové jevy. Jedním z prvků davových jevů je také popis davové psychiky a co jí ovlivňuje.
3. **Sestavení přehledu** dostupných metodologií a nástrojů modelování davového chování. Bude vytvořen seznam algoritmů, softwarových aplikací a způsobů simulace davů. Ty budou vybrány s ohledem, aby umožnily co nejlépe popsat problematiku práce a sloužily i jako porovnání různých přístupů k danému tématu.
4. **Návrh konceptuálního modelu** navrženého pro tematiku davového chování a **implementace** v jazyce JAVA 8. Konkrétní metodika jednotlivých experimentů a návrhu modelu bude rozebrána v samostatné kapitole experimentální části. Pro provedení experimentů a demonstraci zvolených přístupů k řešení problematiky bude navržen a naprogramován framework, který bude ovlivněn teoretickou částí práce především v možnostech řešení dílčích podproblémů, jako je navigace, orientace v prostoru apod.
5. **Provedení experimentů** s implementovanými modely a sběr výstupních dat. Experimenty budou spuštěny na implementaci navrženého modelu. Data každé entity budou zachycena simulačním prostředím a uložena do relační databáze pro další zpracování.
6. **Porovnání výsledků experimentů** s odpovídajícími pracemi.

3 Davové chování

V této kapitole jsou popsána teoretická východiska pro teorii davů, faktory na ně působící a jejich klasifikace. Součástí je také teorie davového chování včetně přehledu současného stavu tohoto odvětví. Poznatky, obsažené v teorii davu, jsou čerpány z publikací *Psychologie davu* [1], *Hromadné psychické jevy* [2] a *Velký sociologický slovník* [3]. Na teorii je poté navázáno problematikou modelování davů v dnešní době, která je souhrnem poznatků z odvětví zabývajících se simulacemi davů. Teorie davu je důležitým základem pro modely založené na problematice davového chování, jelikož nám nabízí představu o chování jednotlivce i zákonitostech v davu. z empiricky vypořádaných procesů, fungujících v davech, lze navrhovat umělé modely, které mohou tyto procesy věrně napodobovat.

Důležitým autorem, pracujícím s teorií davu, je Gustave Le Bon (1841-1931), který formuloval koncept davového chování. Odkazy jeho práce lze nalézt v publikacích, které se davy zabývají, jelikož byl prvním, který se snažil popsat dav jako celek a pochopit jeho formování. Jeho práce vedla k vývoji sociální psychologie.

3.1 Masa, dav

Základní jednotkou davu je jedinec [1]. Každý z nás se minimálně jednou za život stane členem určité skupiny dalších subjektů – *davu*. je však nutné uvést na pravou míru termín *dav*, jelikož tento termín nedefinuje pouze davy v ulicích, ale sociální agregát¹ - skupinu osob se společným cílem. Davy v ulicích, davy osob v nákupních centrech, nádražích apod. označujeme jako *masu* [1]. Jak je již možné usoudit, ačkoliv je *masa* termín související s *davem*, ve skutečnosti se odlišují pouze cílem. Lidé v davech mají relativně společný cíl, zatímco jedinci v *masě* se pouze nahodile setkali na jednom místě. Poznatky v této kapitole je však možné aplikovat jak na davy, tak na *masy*. Pokud je tedy mluveno o nějakém faktu ve vztahu k davu, je možné jej použít i ve vztahu k *masě*.

Na dav působí dle Le Bona [1] faktory *přímé* (bezprostřední) a *nepřímé*. Nepřímé faktory, jako rasa, tradice a čas, nebudou dále rozebírány pro svou nesouvislost s praktickou částí. Jsou však důležité při formování davů, jelikož svým způsobem definují

¹ sociální útvar, tvořený členy, kteří se sešli náhodou, neplánovaně, navzájem se neznají a pravděpodobně se ve stejném složení už nesetkají [3]

strukturu davu. Strukturu ve smyslu diverzity jednotlivých členů. Přímými faktory jsou slova, představivost, zkušenosti, fyzické a morální limity.

3.2 Klasifikace davů

Davy a masy lze členit mnoha způsoby. Vybrány jsou zde čtyři způsoby, které spolu v určitých ohledech souvisejí nebo v některých případech na sebe navazují.

Davy lze klasifikovat mnoha způsoby, dle Le Bona [1] např. na *heterogenní* a *homogenní*. Heterogenní tvoří lidé, kteří se odlišují, nemají stejný charakter, způsob vystupování atd. v této skupině jsou lidé do jisté míry anonymní. u heterogenních davů je obtížné předvídat chování, jelikož je závislé na mnoha proměnných. Naproti tomu homogenní davy jsou charakteristické společnými znaky jedinců, často i společným cílem. Jsou předvídatelnější než heterogenní. Příkladem homogenních davů mohou být sekty nebo turistické skupiny, zatímco heterogenní davy tvoří obvykle průvody nebo lidé pohybující se veřejnými prostory. Tato klasifikace byla sestavena v 19. století. V této práci budou zkoumány jak davy heterogenní (lidé pouze procházející veřejným prostorem), tak davy homogenní (skupiny se stejným fyzickým cílem).

Klasifikaci Le Bona doplňuje typologie davu R. W. Browna [4], z roku 1954, která davy dělí na *aktivní* a *pasivní*. Pasivní dav zde nebude detailně rozveden, jelikož jej v práci nevyužívám. Pouze pro upřesnění pojmu uvádím, že se jedná buď o nahodilé publikum, vznikající při nevšedních událostech, jako jsou dopravní nehody či živelné katastrofy, nebo o posluchačstvo, které se účelně schází na přednáškách či představeních. Zde také lze použít klasifikace Le Bona a říci, že nahodilé publikum je heterogenní a posluchačstvo homogenní.

V roce 1993 vznikla klasifikace dle H. Blumera (1900-1987), podle forem kolektivního chování [3]. Ten dělí kolektivy na *davy*, *masy*, *veřejnost* a *sociální hnutí*. Dav je charakteristický společnými emocemi (strach, radost, hněv) a společným cílem. Masa je zvláštní formou davu, kde nedochází k cílené interakci zúčastněných a jedná se o sociálně nezávislé konání velkého počtu lidí. Veřejnost je určena společným chováním na úrovni, kde nedochází k cíleným interakcím. Sociální hnutí je organizované, má jasnou strukturu a cíl. Lidé se sdružují do těchto skupin za účelem dosažení společných cílů.

Přístup, který v roce 2005 vytváří Forsyth [5], člení masy na *shromáždění* a *davy*. Shromáždění dále člení na *běžné masy*, *publikum a fronty*. Davy na *agresivní (lynčující / vzbouřené)* a *panické (únikové / akviziční)*. v tomto přístupu lze spatřit shody s jinými klasifikacemi. Například běžné masy je možné přirovnat k heterogenním davům Le Bona a publikum k posluchačstvu dle Browna.

Pro demonstraci této myšlenky přikládám následující příklady, kde jsou jednotlivé klasifikace v pořadí dle a) Le Bona [1], b) Browna [4], c) Blumera [3] a d) Forsytha [5]:

1. Klasifikace skupiny, účastníci se školního výletu v zoologické zahradě.
 - a) Homogenní dav.
 - b) Aktivní dav.
 - c) Dav.
 - d) Běžná masa.

2. Klasifikace unikající masy v prostorách mrakodrapu.
 - a) Heterogenní dav.
 - b) Aktivní dav, únikový.
 - c) Masa.
 - d) Panický dav.

4 Modelování davových jevů

Davová dynamika je komplexním problémem, který řeší vztah prostorového uspořádání a průchodu (většinou velkých) skupin osob a souvisí se základy teorie davu a davového chování. Vychází z poznatků o chování jednotlivce v davu, vnějších i vnitřních vlivech. Davové chování, a s tím související pohyb prostorem, ovlivňuje nejen psychologie, ale také několik měřitelných veličin, které souvisí s prostorovou charakteristikou prostředí, ve kterém se jedinci nachází.

Jedno z možných využití modelů davů je při navrhování bezpečných veřejných prostor. s růstem populace dochází i k rozšiřování městské zástavby. Problémem je však jev, že populace roste rychleji, než jsme schopni rozšiřovat prostory. Dochází tedy k tomu, že veřejné prostory začínají být velmi rušné, zvyšuje se hustota zalidnění jednotlivých prostor a riziko nehod narůstá. Pomocí modelů jsou architekti schopni odhadnout nejvíce rušná místa a přizpůsobit tomu návrh svých budov. Davovou dynamiku však můžeme studovat na stávajících objektech, hledat kritická místa a testovat protiopatření. Tyto modely však nemusí sloužit pouze pro architekty nebo bezpečnostní konzultanty. CGI² týmy nacházejí uplatnění pro tyto modely také, například ve filmech, kde je nutné simulovat výskyt velkého počtu osob.

4.1 Měřitelné charakteristiky kvantifikující procesy davové dynamiky

První veličinou, která ovlivňuje pohyb davu, je *hustota* [6]. S nárůstem hustoty se snižuje komfort a pocit bezpečí jednotlivých osob a narůstá riziko problémů. Jedinec může také být doslova „unášen“ davem, neschopen se z něj vyprostit. Dalším faktorem, který má vliv na chování, jsou překážky prostředí – úzké koridory a východy, nedostatečný počet východů a navigační prvky jako zákazy vstupu apod. Společně s narůstající hustotou davu, který je zpomalován četností překážek, vzrůstá i psychický nápor na jedince davu, který může vést k dalšímu zpomalování toku. Modely, které modelují davovou dynamiku, vycházejí z empiricky získaných dat a vzorů, posléze aplikovaných formou algoritmů ve virtuálním prostředí. Sledují se prostorové interakce mezi jednotlivými členy a výsledky celé simulace. Zadání a předpokládané výsledky simulací závisí na problému, který zkoumáme. Pro demonstraci uvádím příklad

² Computer-generated imaginery

[7] se simulací evakuace budovy při krizové situaci, kde zadáním můžeme rozumět půdorysný návrh, stanovení výskytu entit a scénář. Očekávaným závěrem je konstatování o bezpečnosti budovy a výčtu úseků, kde docházelo k nadměrným kolizím, zpomalování toku a nebezpečným situacím. z těchto výsledků poté můžeme navrhnout protiopatření. z ekonomického hlediska můžeme syntetické modely považovat za výhodnější, než velkou (stovky, tisíce) skupinu testovacích subjektů a zároveň jsou tyto modely naprosto bezpečné pro lidské životy, jelikož jsou pouze virtuálními simulacemi.

Jak již bylo řečeno, modely davové dynamiky vycházejí nejen z teoretických poznatků psychologie, ale také měřitelných hodnot, souvisejících s prostorem, které je důležité brát v úvahu při výpočtech a simulacích. Empiricky zjištěné hodnoty poté vstupují jako proměnné (nastavení) do modelů. Vstupní hodnoty lze také měnit a zkoumat jejich vliv na výsledky simulací. Definice pojmů, souvisejících s davovou dynamikou, dle [6]:

Tabulka 1 Popis prostorových metrik.

Metrika	Definice	Jednotka
Rozměry jedince	Rozměry plochy, kterou jedinec zabírá svým tělem, v pohledu shora dolů.	cm
Rychlost pohybu jedince	Vzdálenost, kterou chodec překoná za jednotku času.	m/s
Hustota davu	Počet jedinců nacházejících se v prostoru vymezeném čtvercem o stranách s délkou jednoho metru.	N jedinců/m ²
Tok davu	Počet jedinců za minutu procházejících prostorem omezeným ze dvou stran.	N jedinců/min
Velikost osobní zóny	Rádus kružnice, jejímž středem je jedinec, která určuje odstup od ostatních.	m
Uražená vzdálenost	Vzdálenost, kterou jedinec urazí za jednotku času.	m
Čas pro překonání vzdálenosti	Rozdíl času na počátku a na konci sledovaného úseku	s
Evakuační čas	Doba, za kterou se jedinci podařilo opustit vyznačenou oblast.	s

Rozměry jedince jsou prvkem, který přímo ovlivňuje hustotu davu, resp. kolik osob může vedle sebe stát v ploše metru čtverečního. Hodnoty, dále využitě v praktické části, se v modelech určují z průměrných hodnot statistických údajů určité populace,

kteřou zkoumáme. Průměrné hodnoty, ze kterých bude vycházeno, sestavené na základě dat použitých v [6]:

Tabulka II Průměrné hodnoty mužské populace.

Průměry pro muže:	Šířka [cm]	Hloubka [cm]	Plocha [m²]
Evropa	49,70	28,60	0,22
USA	51,50	39,00	0,23
Asie	44,00	26,00	0,18

Tabulka III Průměrné hodnoty ženské populace.

Průměry pro ženy:	Šířka [cm]	Hloubka [cm]	Plocha [m²]
Evropa	43,90	30,20	0,21
USA	44,00	30,00	0,21
Asie	43,00	25,30	0,17

Rychlost chůze je kromě hustoty ovlivněna i jinými faktory. Například vlastnostmi jedince (psychické rozpoložení nebo fyzický stav), hmotností, rozměry, věkem a okolními vlivy. Hodnoty, zde uvedené, jsou vypočteny na základě výsledků ze studie [8] chodců ve volném prostřanství v závislosti na věku osoby.

Tabulka IV Tabulka rychlostí mužů.

Věk	Průměrné rychlosti pohybu mužů [m/s]				
	Pomalá chůze	Běžná chůze	Rychlá chůze	Běh	Rychlý běh
21-30	1,10	1,55	2,00	3,60	5,45
31-40	1,15	1,50	2,15	3,60	5,85
41-50	1,10	1,50	1,95	3,25	5,65
51-60	1,00	1,45	1,85	3,20	4,85

Tabulka V Tabulka rychlostí žen.

Věk	Průměrné rychlosti pohybu žen [m/s]				
	Pomalá chůze	Běžná chůze	Rychlá chůze	Běh	Rychlý běh
21-30	1,05	1,35	1,75	2,80	4,40
31-40	1,05	1,35	1,80	2,85	4,05
41-50	1,00	1,35	1,75	2,85	3,60
51-60	0,95	1,35	1,75	2,80	3,60

Výpočet *hustoty davu* je závislý na mnoha proměnných. Kromě struktury terénu, šířky místa, ve kterém provádíme měření a rozměru jednotlivých osob, má vliv i také rychlost pohybu, resp. *toku davu*. Metodika výpočtu průměrné hustoty davu na základě rychlosti, je stanovena dle [6] jako poměr průtoku a průměrné rychlosti. Horní hranice průtoku byla stanovena na: 109 osob, které projdou jedním metrem šířky za minutu. Tato hranice byla prohlášena [9] jako bezpečná. Stanovený počet lze vyjádřit jako 1,82 lidí za sekundu, kteří projdou jedním metrem šířky. s tímto údajem a na základě metodiky lze vypočítat průměrné hodnoty, kdy je ještě hustota davu v rámci stanoveného limitu. Tyto hodnoty jsou vypočteny v souladu s postupem uvedeným v [6].

Tabulka VI Vztah mezi průměrnou rychlostí a hustotou u žen

	Průměrná rychlost [m/s]	Průměrná hustota [n osob/m²]
pomalá chůze	1,01	1,80
běžná chůze	1,35	1,35
rychlá chůze	1,76	1,03
běh	2,83	0,64
rychlý běh	3,91	0,47

Tabulka VII Vztah mezi průměrnou rychlostí a hustotou u mužů

	Průměrná rychlost [m/s]	Průměrná hustota [n osob/m²]
pomalá chůze	1,09	1,67
běžná chůze	1,50	1,21
rychlá chůze	1,99	0,92
běh	3,41	0,53
rychlý běh	5,45	0,33

Osobní zóna každého jedince určuje vzdálenost v neverbální komunikaci – odstup od druhého. Tato vzdálenost je vnímána především opticky, ale i podvědomě a smyslově. Osobním prostorem se zabývá věda, zvaná *proxemika*. Výraz „proxemika“, v roce 1963 stanovil Edward T. Hall, ve své knize *The hidden dimension* [10]. Dle tohoto kulturního antropologa se prostor skládá ze čtyř intervalů:

1. Intimní zóna: od pokožky těla do 0,45 metrů.
2. Osobní zóna: 0,45 až 1,2 metrů.
3. Společenská zóna: 1,2 až 3,6 metrů.
4. Veřejná zóna: 3,6 metrů a více.

Uvedené vzdálenosti jsou u každého jedince individuální. Jsou ovlivněny především společenskými a psychickými faktory. Menší vzdálenosti se vyskytují ve městech a zároveň vypovídají i o vztahu k druhému.

S proxemikou souvisí také *proxemický tanec*. Ten lze vyzorovat při setkání dvou, nebo více osob s rozdílnou velikostí osobní zóny. Projevuje se neustálou korekcí vzdáleností zúčastněných osob. Jedinec s menší zónou se přibližuje tomu s větším prostorem a on uhýbá do volného prostoru. „Tanec“ je vyřešen v momentě, kdy dojde k nalezení vzájemně vyhovující polohy. Lze také říci, že vyřešení tohoto problému předchází jistý kompromis mezi rychlostmi a osobními prostory účastníků.

Osobní zóna ovlivňuje také dynamiku chodců do té míry, že narušení určitých zón vede k úpravě rychlosti a směru chůze. Ve své podstatě pohyb davem provází neustálý proxemický tanec, kdy jedinec reaguje na osobní prostory druhých a přizpůsobuje tomu svou hybnost.

Osobní zónu rozšiřuje jev, zvaný *bublina osobního prostoru* [11]. Neviditelná část prostoru, kterou podvědomě vnímáme. Autor knihy *Tajná řeč těla* [11] popisuje tento prostor jako válec, který se zužuje od pasu dolů. Rozměry válce se neustále přizpůsobují v závislosti na okolí a vztahu k těm, který chtějí vstoupit do osobní zóny. Konečná podoba tohoto prostoru závisí na výsledku konfliktu přiblížení a vyhnutí se.

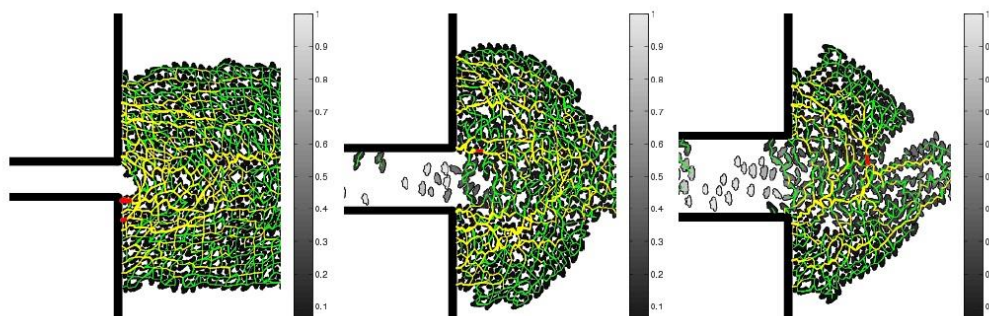
S uvedenými metrikami se při studiu davové dynamiky setkáváme nejčastěji, jelikož jsou to hodnoty, které můžeme měřit a kvantifikovat.

4.2 Metody modelování davového chování

Pro výběr vhodného modelu je nutné vzít v úvahu mnoho kritérií. Není jednoduché sestavit univerzální seznam, jelikož každá aplikace může požadovat jiné chování modelu. Také každý zadavatel nebo osoba, která daný model bude využívat, může mít jiné požadavky. Na základě souvislostí s mou prací jsem sestavil komentovaný výčet některých požadavků z nich v souladu s [12]:

1. *Jaká je míra detailu modelu?* Soustředíme se na dav jako celek, nebo nás zajímá chování jednotlivých členů davů. Budeme sledovat reakce na každou událost, nebo se zaměříme na jejich delší sled?

2. *Přesnost chování jednotlivých entit.* Přesnost ve smyslu, jak dobře dokáží entity napodobovat chování skutečných osob. Zda budou předvídatelně reagovat na situace a fungovat v souladu s definovanými pravidly.
3. *Probíhá simulace v reálném čase?* Očekáváme téměř okamžitou reakci jednotlivých entit, nebo při jednotlivých aktualizacích prostředí bude mít každá entita své neomezené časové okno na vyhodnocení situace.
4. *Jaká je škálovatelnost modelu?* Dokáže simulovat desítky, stovky, tisíce osob? Některé modely mohou dobře fungovat při malém počtu osob, ale jakmile zvýšíme četnost výskytu, mohou tyto modely výkonnostně nedostačovat.
5. *Časová náročnost implementace a zavedení modelu.* Tento bod připadá v úvahu v momentech, kdy se rozhodujeme k řešení nějakého problému a máme omezený čas na implementaci nebo zavedení již softwarově zpracovaného modelu.
6. *Dostupnost softwarového řešení.* v případech, kdy chceme využít již dostupná řešení, je třeba vyřešit otázku finanční dosažitelnosti, zda použijeme nějaké open-source, free či komerční řešení. Svou roli také může sehrát také otázka licencí.
7. *Grafická kvalita výstupu.* Týmy, zabývající se zpracováním počítačové grafiky, bude zajímat kvalita výstupu spíše než bezpečnostní konzultanty, zabývající se studiem přístupnosti veřejných míst.



Obrázek 1 Příklad davové dynamiky

zdroj: <http://sydney.edu.au/engineering/civil/personal/fernando/images/crowd%20configuration.jpg>

Obrázek 1 demonstruje příklad aplikace modelů. Vytvoříme prostor omezený překážkami (v tomto případě úzká chodba omezená zdmi), do modelu vložíme desítky entit s jediným cílem – projít na druhou stranu a zachovat při tom základní prvky prostorové orientace: nekolidovat s překážkami a ostatními entitami. Tento konkrétní model slouží pro měření, kolik času uplyne, než se entity ze své výchozí pozice

(na začátku experimentu) přesunou do východu. Úpravou vstupních parametrů, např. rychlosti a počtu entit (na obrázku lze vidět i úpravu šířky chodby), lze sledovat změny v evakuačních časech.

Pro modelování davů byly vytvořeny metody založené na jiných vědních oborech. Tyto metody lze poté implementovat v simulovaných prostředích v jakémkoliv vhodném programovacím jazyce. Metod, vytvořených pro simulování davů, je celá řada, popsáno zde bude několik z nich, užívaných při simulování davové dynamiky, se kterými se lze setkat. Jednotlivé metody byly vybrány tak, aby bylo možné přehledně demonstrovat různé přístupy k řešení problémů simulování mas.

Nejstarší a stále používané modely jsou založeny na *celulárních automatech* [13]. Dále se využívají modely napodobující *tok kapalin* [14], resp. dynamiku kapalin, a *modely sociálních sil* [15]. v neposlední řadě zde uvádím modely založené na *agentních systémech* [7], které dokáží simulovat chování jednotlivých účastníků. Toto paradigma je také tématem praktické části.

Celulární automaty jsou proměnlivým systémem a matematickým modelem, které napodobují živý systém. Dostupná literatura je popisuje jako nejzákladnější metodu pro simulování davů. Modely, postavené na celulárních automatech, jsou definovány jako modely [7], kde prostředí, i jednotky v něm obsažené, jsou diskrétní v prostoru, času a stavové proměnné agenta nebo buňky. Sestávají se z buněk (celulární – buněčné), kdy buňky mohou být v N-rozměrném uspořádání. na začátku simulace má každá buňka svůj počáteční stav a prostředí postupně prochází iteracemi. v každé iteraci si buňky na základě evolučních pravidel a stavů sousedních buněk mění svůj stav. Celulární automaty byly využity například pro simulování davové dynamiky při evakuacích [7]. Prostředí bylo rozděleno na matici buněk, kdy se každá buňka nacházela ve dvou stavech – prázdná, nebo zaplněná (podobnost s mrtvá/živá v teorii celulárních automatů). Zaplněné buňky poté představovaly překážky (překážkami se mohou stát i jiné osoby) nebo ostatní osoby, prázdné poté volný prostor. Buňky, představující osoby, měly za cíl se přemístit k cíli. Každý krok byl poté pokusem dostat se k němu co nejrychleji. Pokud se v cestě k cíli nacházela buňka zaplněná, byla nalezena první volná a provedl se krok (přesun osoby).

Tabulka VIII Ukázka analogie reálného světa a aplikace v celulárních automatech.

Celulární automat	Reálné prostředí
Volná (mrtvá) buňka	Prázdný prostor
Obsazená (živá) buňka	Jiná osoba nebo překážka
Změny dle stavů okolních buněk	Osoba reaguje na bezprostřední okolí
Buňka se „přesune“ pouze do místa uvolněného jinou buňkou	Osoba má možnost se pohybovat jen v prostoru uvolněném jinými

Tabulka 8 ukazuje, jak celulární automaty napodobují reálný svět právě díky tomu, že každá buňka mění svůj stav v závislosti na okolních tak, jak lidé vnímají v davech své bezprostřední okolí.

Modelování celulárními automaty je nenáročné na výpočetní prostředky - v každé iteraci se složitost rovná počtu obsazených buněk a postupem času, jak „simulované entity“ postupně opouští prostředí, se tato náročnost snižuje. Celulární automaty však nesimulují například takový jev, kdy se osoby mohou stát překážkami, když upadnou. v celulárních automatech se všechny entity hýbají kupředu. Jejich vlastností je také fakt, že veškeré charakteristiky jedinců jsou zjednodušeny do podoby jediné buňky.

Alternativním modelem k celulárním automatům [15] je *model sociálních sil*³. Model navržený v roce 1995 simuluje jevy, které jsou pojmenovány jako sociální síly. Ideou bylo nalézt množinu základních pravidel, vztahujících se k jednotlivcům, které by popisovaly faktory ovlivňující jejich chování. Model říká, že každý chodec je ovlivněn čtyřmi základními faktory [16]:

1. Chodec chce dosáhnout konkrétního cíle (destinace).
2. Chodec si udržuje určitou vzdálenost od ostatních lidí.
3. Chodec si udržuje vzdálenost od překážek a okrajů.
4. Chodec může být přitahován k ostatním lidem a objektům.

V Helbingově modelu sociálních sil [15] jsou entity reprezentovány jako částice řízené vektorem rychlosti. Tento vektor je vypočítán na základě několika rovnic, které převádí podmínky dané prostředím na rychlost a směry. Model se zaměřuje na instinktivní rozhodování jednotlivých agentů a simuluje spíše rychlé volby, vycházející z okolních vlivů.

³ Social forces model

Z obecného pohledu se masa pohybuje jako kapalina [14]. Tento poznatek slouží jako základ pro modelování pomocí modelů založených na dynamice kapalin.

Tyto modely vycházejí ze tří základních hypotéz [16]:

1. Chodci uzpůsobují svou rychlost hustotě davu.
2. Chodci mají společný cíl.
3. Chodci se snaží minimalizovat čas nutný k dosažení cíle, zatímco se vyhýbají oblastem s vysokou hustotou davu.

První hypotéza vychází z poznatku [14], že při velkých hustotách davu, tedy malých rozestupech mezi jednotlivci, jsou důležité jednotlivé interakce mezi nimi. S nárůstem hustoty lze předpokládat vznik tlačenic, zpomalování proudu a snižování hybnosti jednotlivých osob. Analogicky toto můžeme chápat jako nárůst viskozity v kapalině. Druhá i třetí hypotéza je závislá na předpokladu, že jsou chodci vizuálně informováni o pozici cílové destinace a případných překážkách. Tento předpoklad selhává v momentě, kdy simulujeme evakuaci osob, kdy je vizuální informovanost jedinců limitována ostatními chodci v okolí.

Při velké hustotě chodců se modely založené na dynamice kapalin ukazují být více přesné než ostatní matematické modely [9]. To je dáno tím, že dokáží přesně simulovat vazby mezi jednotlivými členy a při dodržení jednotlivých hypotéz nabízejí i realistický pohled do chování davů. Vazby v tom smyslu, jako když pohyb jednotlivých molekul v tekutině je závislý na blízkém okolí. Tyto modely byly rozšířeny o schopnost napodobovat některé pozorovatelné jevy v davech, jako je zpomalování ve dveřních prostorách a únava chodců. Zpomalování při průchodu „dveřními“ prostory si můžeme představit jako situaci, kdy dav prochází otočnými dveřmi a každý jedinec v jejich prostoru stráví určitou dobu, tím pádem za ním vznikají fronty a celkový tok davu se zpomaluje.

Tabulka IX Ukázka analogie mezi kapalinami a davy.

Model dynamiky kapalin	Reálné prostředí
Zvyšování hustoty	Zvyšování výskytů jedinců
Zvyšování viskozity, „tření“ v kapalině	Snižování vzdáleností má za následek snižování rychlosti
Rozdíl v rychlostech jednotlivých vrstev kapaliny	Osoby na okrajích se mohou pohybovat rychleji než osoby v centru
Čím vyšší viskozita, tím méně kapaliny proteče	Čím vyšší hustota, tím déle budou osoby překonávat určitou vzdálenost

Praktické aplikace metod, založených na dynamice kapalin, jsou stále výzvou [9], jelikož stěžejní vzorce, popisující modely davů, nemají lineární složitost a jsou závislé na třech uvedených hypotézách. Problémem těchto metod je také nemožnost sledovat chování a interakce mezi jednotlivými členy. Dávají přehled jen o pohybu celého davu. Tento problém je řešen modely založenými na agentních systémech, kterým je věnovaná samostatná kapitola.

4.3 Nástroje modelování davového chování

V této kapitole je zpracován přehled nástrojů, pomocí kterých lze modelovat davovou dynamiku. Těchto nástrojů existuje celá řada. Liší se svým zaměřením, zvolenou metodou modelování a licenční politikou. Vybrány byly celkem tři nástroje, které jsou nejčastěji zmiňované v použité literatuře.

Informace z této podkapitoly jsou čerpány z práce Understanding Crowd Behaviours: Simulation Tools [17] a webů dvou produktů:

- <http://www.legion.com>
- <http://www.oasys-software.com/products/engineering/massmotion.html>

LEGION

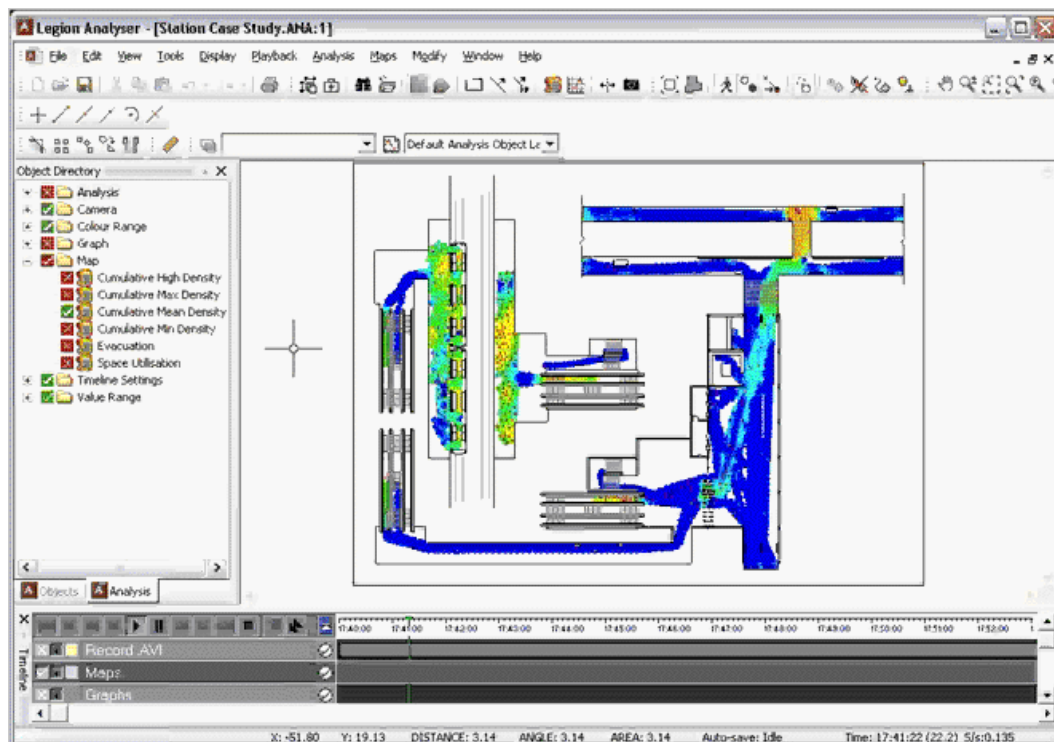
Tento software (založený na agentních systémech) je využíván pro návrh míst, které mohou okupovat lidé, strategické plánování (zhodnocení benefitů nových veřejných zástaveb dříve, než budou schváleny a postaveny), zkoumání bezpečnosti (např. zefektivňování evakuačních procedur) a 3D vizualizace (demonstrovat návrh reálného prostředí). LEGION umožňuje modelaci lidského chování v situacích, jako jsou obchodní centra, doprava, letištní plochy, metro a obecně hromadná doprava.

Každého agenta simulace lze sledovat, jak prochází prostředím z výchozího bodu do koncového, prochází davem a je schopen provádět nezávislá rozhodnutí. Základem uvažování agentů je projít prostředím s co nejmenší námahou – vydat se cestou nejmenšího odporu (minimum času z bodu A do B, maximální rychlost atd.). Agenti se rozhodují na základě stavů prostředí. Každému agentovi lze nastavit atributy, např. velikosti, věku a rychlosti. LEGION také umožňuje do prostředí vkládat náhodné události, například pozdní příjezd vlaku.

LEGION není pouze jediný program, ale množina produktů, které jsou specializované pro konkrétní modely, které simulují. Dostupné jsou [18]:

- **Legion SpaceWorks.** Dle tvůrců se jedná o jejich hlavní produkt. Simuluje pohyb chodců a sleduje reakce na úrovni jednotlivých entit. Umožňuje designovat prostředí a okamžitě sledovat reakce chodců.
- **Legion Evac.** Aplikace, která simuluje evakuační scénáře (v nádražích, sportovních stadionech, letištích, ...). Podporuje testování bezpečnostních opatření (vyznačené únikové cesty, ...).
- **Legion 3D a Legion Viewer.** Legion 3D je doplněk pro SpaceWorks, který umožňuje simulace vykreslovat v trojrozměrném prostředí. Také integruje podporu např. Google Sketchup. Viewer je jednoduchá samostatná aplikace, která je určena pro přehrávání uložených simulací ze SpaceWorks.

- **Legion Studio.** Původní program, který se sestával ze tří modulů (Model Builder, Simulator, Analyser), který byl nahrazen SpaceWorks.

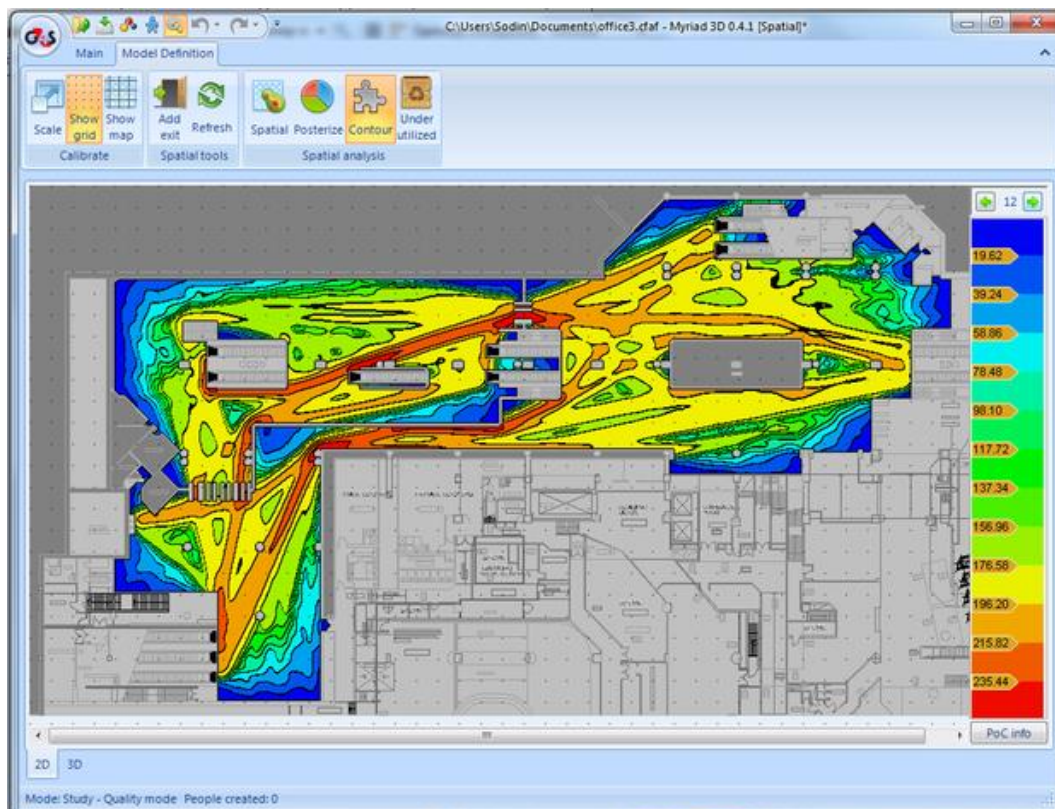


Obrázek 2 Ukázka programu Legion,
zdroj: <http://www.legion.com/legion-spaceworks>

MYRIAD II

Vyvinul jej profesor Keith Still a jeho kolegové, který napsal dizertační práci [6], ze které je čerpáno i v této práci. Hlavním účelem tohoto programu je studovat kde, kdy a jak selže systém, ovlivňující dynamiku davu – např. kde, kdy a jak dojde k tomu, že lidé nebudou moci opustit budovu v krizové situaci. Předností softwaru MYRIAD II je to, že dokáže v reálném čase zobrazovat průtok vybraným místem, hustotu a další metriky, související s davovou dynamikou.

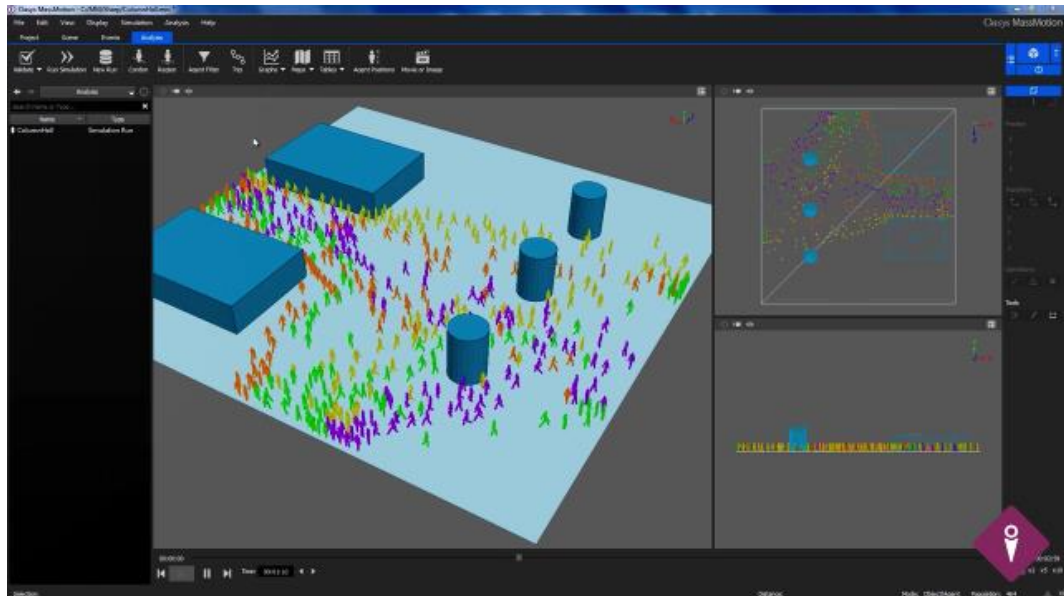
MYRIAD II je třetí iterací modelovacích nástrojů, které profesor Still vyvinul. V roce 2000 prodal práva na software LEGION [19] a začal vyvíjet MYRIAD I. Jeho cílem bylo vyhnout se omezení, které v té době LEGION měl – entity se posouvaly po „kolejích“ (připravené trase). Současná verze programu se zaměřuje na 3D grafiku a detailní zobrazení prostředí.



Obrázek 3 Ukázka simulačního nástroje Myriad,
zdroj:http://www.gkstill.com/_Media/5-3_med.png

MASS MOTION

Systém navržený pro plánování osobní a veřejné dopravy, letištních terminálů a sportovních stadionů. Většina scénářů v tomto softwaru vychází ze základní otázky „co se stane, když...?“ („Co se stane, když uzavřeme většinu východů?“, „Co se stane, když se začnou výrazně zpožďovat autobusy?“). Je to 3D modelovací software, založený na agentním systému, ve kterém se pohybují entity, schopné dělat nezávislé rozhodnutí. Klíčovým prvkem všech jednotek je společný cíl – provést zadaný úkol v co nejkratším čase. Každý agent má přehled o prostředí kolem sebe.



Obrázek 4 Ukázka programu MassMotion,
zdroj: <http://www.oasys-software.com/products/engineering/massmotion.html>

Porovnání jednotlivých simulačních nástrojů dle [17]:

Tabulka X Porovnání simulačních nástrojů

LEGION	Výhody
	<ul style="list-style-type: none"> ▪ Jednotliví agenti mohou mít náhodné atributy v určitých intervalech, což zvyšuje diverzitu davu. ▪ Samotný program je přívětivý pro uživatele s velkým důrazem na kvalitu ovládacího prostředí a podání statistik.
	Nevýhoda
	<ul style="list-style-type: none"> ▪ Simulace komplexních prostředí je časově nákladná a je vyžadován silný výpočetní výkon.
MYRIAD II	Výhody
	<ul style="list-style-type: none"> ▪ Integruje tři modelovací techniky naráz (prostorovou analýzu, síťovou analýzu a agentní systém). ▪ Prostedí lze globálně modelovat jako síť a složitější části poté řešit pomocí agentů).
	Nevýhoda
	<ul style="list-style-type: none"> ▪ Komplexnost programu je také jeho nevýhodou, jelikož zvyšuje nároky na uživatele.
MASS MOTION	Výhody
	<ul style="list-style-type: none"> ▪ Každá jednotka „uvažuje“ několik kroků dopředu a bere v potaz skutečnosti, které ovlivňují její postup. ▪ Umožňuje sestavovat celá města, složená z různorodých prostředí a modelovat tak průchod více lokacemi.
	Nevýhoda
	<ul style="list-style-type: none"> ▪ Nebere v potaz emoce jednotlivce a jejich dopad na chování a pohyb.

5 Multiagentní modelování

Obsahem této kapitoly je přehled teorie systémů, založených na jednotlivých, na sobě nezávislých jednotkách – agentech. Systémy, které jsou tvořeny prostředím a agentem, který se v něm pohybuje, se nazývají agentní. Multiagentní systémy jsou pak pouze prostředím s větším (2 a více) výskytem těchto jednotek.

5.1 Multiagentní systém

Multiagentní systémy byly popsány v rámci vývoje distribuované umělé inteligence. Jejich základ je definován samostatnými jednotkami, nezávislými na centrální autoritě – agentech. Toto paradigma přineslo [20] koncem minulého století příslib rychlejšího rozvoje distribuovaných a otevřených systémů jako byl internet. Evoluci v prostředí, kde každý prvek může obstarat dílčí část problému a společně řešit úkoly, jenž by každý jednotlivý z nich nemohl řešit v rozumném čase. Každý takový aktér, agent, vybaven svou limitovanou množinou informací, důležitých pro zkoumání daného podproblému, poté komunikuje s dalšími členy, aby nakonec dosáhli společného konsenzu. Rozdělením zátěže na jednotlivé prvky, agenty, lze dosáhnout vysoké paralelizace, tedy řešení úloh nezávisle na sobě, což urychluje zkoumání problému, snižuje komunikační ruch (agenti si předávají závěry nebo informace nutné pro další článek) a především navýšení spolehlivosti – někteří členové komunity mohou být zastoupeni, případně lze přidávat další jednotky. Jako komunitu zde chápou společenství jednotek, sdílejících společnou oblast, prostorově vymezenou.

Multiagentní systém je tedy složen z komunit agentů a prostředí, ve kterém se agenti pohybují. Záměrně uvádím plurál slova komunita, jelikož každý agent může být členem více komunit, například dle toho, jakého je typu, či zaměření. Tato fakta jsou pro modelování mas či davů důležitá, jelikož můžeme každého jedince vybavit vlastní množinou zájmů a vlastností. Poté lze v jednom unifikovaném prostředí studovat jak jejich vzájemné interakce, tak i reakce na stavy okolí. Pro demonstraci uvádím jako příklad experimentální model této práce – koordinovaný průchod veřejným prostorem. v tomto modelu se vyskytuje množství jednotek, kde každá z nich má vlastní množinu pohybů, vlastností a zájmů (v tomto případě nějaký cíl a průchod prostředím). Tyto entity se poté pohybují v modelovém prostředí letištní haly, které musí být pro všechny účastníky homogenní, ale to nic nemění na tom, že se může jeho stav v čase měnit.

Prostředí (vše, s čím agent přichází do styku) dělíme na [21]:

1. *Agentní / multiagentní* – v závislosti na tom, jestli v prostředí operuje jeden nebo více agentů. Prostředí může být agentní i v případě, kdy se v jednom prostředí nachází více agentů, ale každý operuje nezávisle a disponuje vlastní sadou znalostí a cílů. Příkladem multiagentního prostředí jsou například experimentální modely v této práci.
2. *Spojité / diskrétní* – množina stavů, ve kterých se prostředí nachází. Spojitým prostředím je svět kolem nás, které vnímáme souvisle – spojitě. Opakem je množina odlišitelných a předem stanovených diskrétních stavů. Diskrétním prostředím jsou šachy, kde každý tah má dané možnosti. Nelze však opomenout fakt, že i realitu lze zachytit diskrétně, například kamerovým záznamem.
3. *Statické / dynamické* – statické prostředí se mění v návaznosti na akcích agenta. Tedy stavy tohoto prostředí se mění po zpracování agentem. Naopak dynamické se mění i během vyhodnocování agentem a agent tedy musí sledovat i vývoj prostředí a čas. Statickým prostředím jsou šachy, kdy se jednotlivé tahy provádí po vyhodnocení „agentem“. Dále můžeme hovořit o prostředí semi-dynamickém, kdy se samotné prostředí v čase nemění, ale výkonnost agenta ano.
4. *Deterministické / nedeterministické* – toto členění ovlivňuje možnost předvídat stav prostředí. v deterministickém prostředí lze vycházet z aktuálního stavu a s určitou přesností předvídat další. Šachy jsou prostředím deterministickým, jelikož známe svůj aktuální stav a můžeme vyhodnotit další krok.
5. *Přístupné / nepřístupné* – v přístupném prostředí může agent získat přehled o celkovém stavu. Agent si tedy nemusí udržovat vlastní přehled o stavech, ale stačí mu vždy „zkontrolovat“ aktuální stav prostředí. v nepřístupném prostředí je agentovi umožněno zkoumat jen bezprostřední okolí. Toto rozdělení můžeme také nazývat jako plně / částečně pozorovatelné.

Tabulka XI Příklady prostředí.

Prostředí	Šachy s hodinami	Šachy bez hodin	Obrazová analýza	Městská doprava
Přístupné	Ano	Ano	Ano	Ne
Deterministické	Ano	Ano	Ano	Ne
Epizodické	Ne	Ne	Ano	Ne
Statické	Semi	Ano	Semi	Ne
Diskrétní	Ano	Ano	Ne	Ne

5.2 Agent

Agent je jednotka, základní prvek agentních systémů, entita. Může zastupovat reálné objekty, v našem případě jedince. Agenty můžeme rozdělit na biologické, technické nebo programové. Biologičtí agenti mohou být lidé, kteří se snaží formovat prostředí do nějakého stavu, nebo nalézt řešení na nějaký problém. Techničtí agenti, jinak řečeno roboti, mohou provádět náročné operace, které by biologický agent nebyl schopen provádět, nebo se naopak může jednat o velké komunity malých robotů, kteří se mohou pokoušet napodobovat chování ptactev apod. Pokud nepotřebujeme fyzické agenty, nebo nám je povaha problému nedovoluje využít, můžeme simulovat jejich chování pomocí výpočetních prostředků, poté jsou to agenti softwaroví (programoví). V této práci budou využiti agenti softwaroví.

Architekturu agenta lze obecně rozdělit na několik základních částí:

1. *Senzory*. Tvoří vstupní rozhraní, kterým agent přijímá vjemy a podněty z okolního prostředí. na tomto vstupu také může naslouchat ostatním agentům. ve fyzickém prostředí může toto rozhraní představovat různá teplotní čidla nebo senzory přiblížení. ve virtuálním světě mohou být senzory tvořeny metodami, které posléze v programu voláme. ve světě lidí jsou tyto senzory ve formě smyslů.
2. *Aktuátory* jsou opakem vstupního rozhraní. Mohou nabývat rozličných forem a umožňují agentovy měnit stav okolního prostředí. Fyzické aktuátory je možné si představit jako lidské ruce nebo robotí paže. Analogií v softwarovém světě může být funkce pro zápis do souboru.
3. *Vnitřní reprezentace*. Pokud to účel agenta vyžaduje, měl by obsahovat určitou formu paměti a výkonné části, která zpracovává a reaguje na vnější podněty. Tato část také bude zodpovědná za vytváření znalostní databáze prostředí. Tedy, v jakých stavech se nachází sám agent, prostředí a okolní agenti, pokud se jedná o multiagentní systém. Jaké postupy a způsoby vyhodnocování se mají použít. Tento modul je možné si představit jako vnitřní zákonitosti robota, podle kterých bude postupovat.
4. *Modul plánování* řídí chování agenta v závislosti na čase. Vychází z jeho cílů. Odvozuje pravidla, plánuje cíle a kontroluje jejich plnění. Tomuto modulu je třeba věnovat odpovídající pozornost, pokud chceme navrhovat agenta, který má být racionální.

Základními charakteristikami agenta jsou [21]:

1. *Možnost napodobení sociálního chování.* Agenti mohou spolupracovat pro dosažení společných cílů, vytvářet skupiny, sdružovat se do komunit se společným záměrem apod. Také si mohou uchovávat informace o jiných agentech a na základě toho se rozhodovat.
2. *Schopnost zaměřit se na dlouhodobý cíl (intencionalita).* Přizpůsobovat své chování tak, aby tohoto cíle dosáhl, využívat vlastní úsudek a plánovat vlastní podcíle k jeho dosažení.
3. *Schopnost reagovat na události (reaktivita).* Agent vnímá události a dokáže na ně adekvátně reagovat.
4. *Schopnost samostatně uvažovat (autonomnost).* Agent dokáže samostatně řešit určitou úlohu, kooperovat s ostatními agenty. Účastnit se nebo opouštět komunity a vznášet požadavky nebo nabízet výsledky zcela na základě vlastní vůle. Jinak řečeno je to vlastnost, spočívající v možnosti nezávisle rozhodovat o svém chování bez ovlivnění ostatními prvky.

Obecně lze agenta souhrnně charakterizovat jako samostatnou autonomní jednotku, disponující množinou schopností, které specifikují možné akce. Tyto akce můžeme měnit v návaznosti na stavy stavech prostředí. Tato jednotka je schopna komunikovat pomocí specifického souboru protokolů s ostatními agenty tam, kde je to žádané (například při kooperaci). Agent může zpracovat informace, přijaté pomocí senzorů nebo komunikačních kanálů, analyzovat je a na základě vyhodnocení provádět akce.

Každá akce, kterou agent vykoná, má vliv na okolní prostředí. v prostředí multiagentních systémů se v tomto prostředí nacházejí i další agenti, na které může působit. Vlivy, kterými může jeden agent ovlivňovat druhé, lze rozlišit jako [21]:

1. *Přímé:* Agent přímo ovlivňuje stavy jiného agenta. Jedinou možností, jak toho může docílit, je komunikace.
2. *Nepřímé:* Agent mění stav bezprostředního okolí ostatních agentů tak, aby ostatní museli přizpůsobit své chování kýženým směrem.

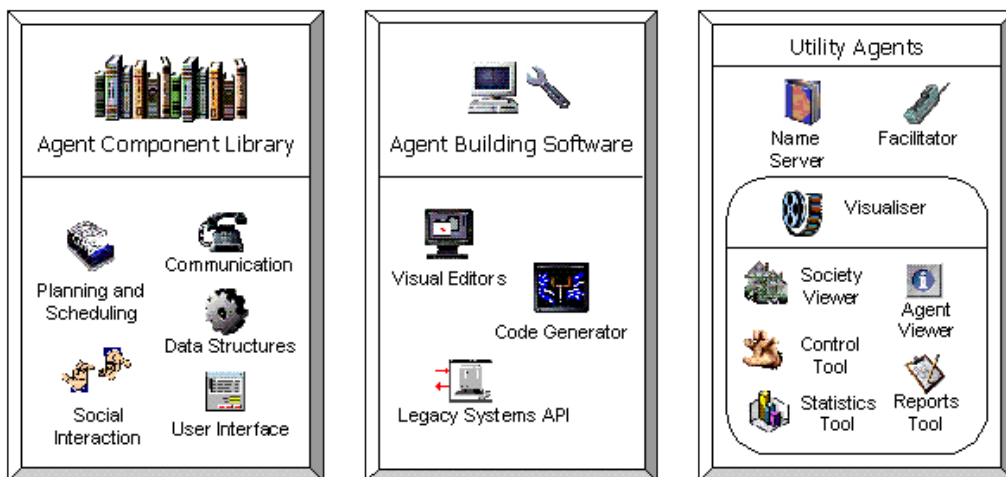
5.3 Nástroje multiagentního modelování

Na webu agentlink.org [22] lze k datu citace nalézt výčet více jak stovky softwarových nástrojů, zabývajících se agentními systémy. V této kapitole budou popsány dva z nich.

ZEUS

Nástroj agentního modelování, který je vyvíjen od roku 1997. Snaží se vyřešit implementační překážky, které souvisí s návrhem infrastruktury agentních systémů. Dle filozofie, ze které produkt vychází, specifikuje pět problémů. Získávání informací o ostatních agentech, podpora předávání zpráv, společný jazyk, koordinace a integrace s jinými softwarovými prostředky. Agenti tohoto produktu jsou schopni plnit více cílů a dílčích úkolů. Agenty v platformě ZEUS lze programovat v grafickém prostředí a přiřazovat jim úkoly, vztahy a definovat reakce na události – tím se vytváří jejich chování.

ZEUS je napsán v jazyce Java a jeho dokumentace je dostupná z [23] a [24]. Rozdělen je do tří funkčních celků – knihovna komponent pro jednotlivé agenty, knihovna poskytující prostředky pro sestavení jednotlivých agentů a knihovna doplňkových nástrojů, jako jsou statistické nástroje, kontrolní mechanismy a vizualizační nástroje, umožňující vykreslit prostředí.

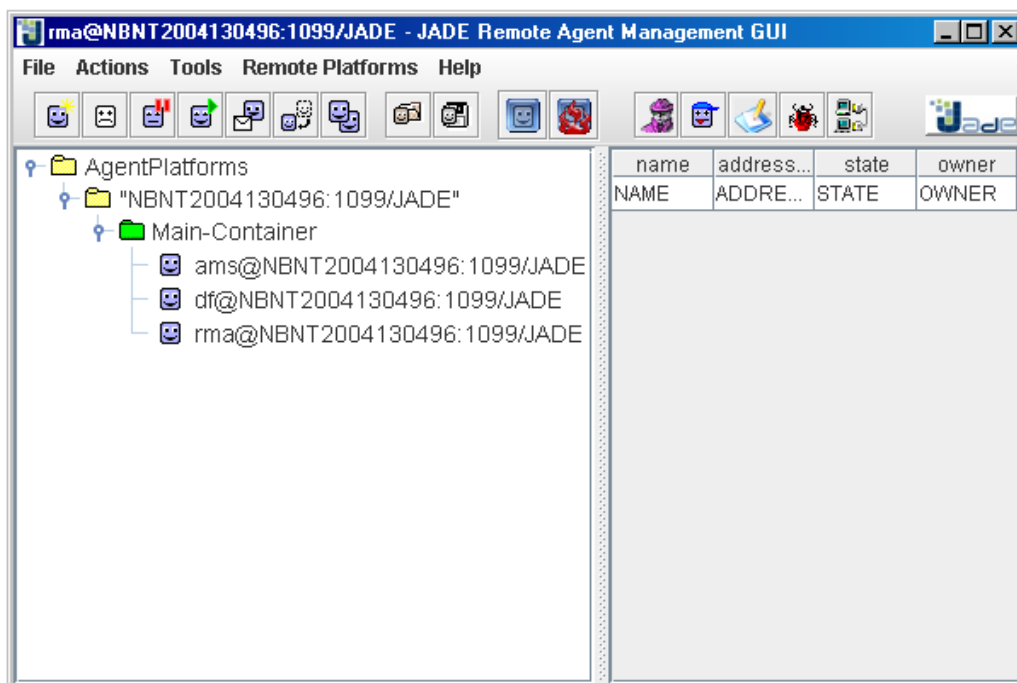


Obrázek 5 Komponenty frameworku ZEUS,
zdroj: <http://zeusagent.sourceforge.net/docs/techmanual/part3.html>

JADE

Java Agent DEvelopment Framework. Platforma, která umožňuje modelovat softwarové agenty dle FIPA⁴ specifikace. Vytvořen roku 1998, poskytuje možnosti pro správu agentů, komunikaci a mobilitu. Je to API s cílem urychlit vývoj aplikací, založených na agentních systémech. Software byl roku 2000 předán Open Source komunitě [22].

Sestává se ze tří hlavních modulů – Runtime Environment, knihovna tříd a nástroje pro správu. Obsahuje jeden kontejner (označovaný jako hlavní) a více dalších kontejnerů, které se nemusí nacházet na stejném výpočetním prostředku (počítači). Kontejner reprezentuje prostředí, kde jsou uloženy instance agentů. Základem je třída *Agent*. Ta představuje funkcionalitu jednotky a prostředky pro komunikaci (zasílání a přijímání zpráv). Funkcionalita, nebo také chování, je abstrakce nějaké stále se provádějící akce – rutiny. Agenti jsou implementováni jako samostatná vlákna. Provedení více úkolů najednou je pak podpořeno jednou z vlastností jazyku JAVA – paralelizací. Tím je zajištěna efektivnost simulací a spolupráce jednotlivých agentů (jednotky nemusí čekat na přidělení času prostředím, vše funguje paralelně).



Obrázek 6 Ukázka JADE řídicí konzole,
zdroj: <http://jade.tilab.com/doc/tutorials/JADEAdmin/startJade.html>

⁴ Foundation for Intelligent Physical Agents

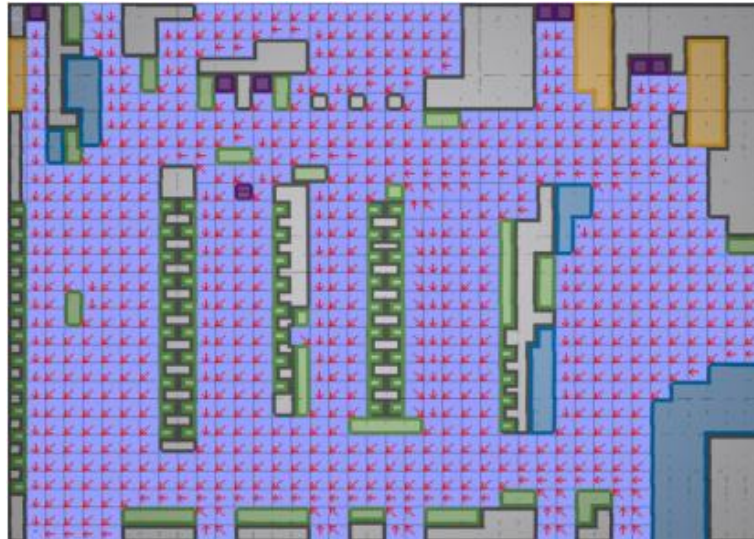
5.4 Příklad modelování davu pomocí agentního systému

Letištní terminály (zmiňované také v této práci) jsou zajímavým cílem zkoumání davových jevů [25]. Popsán zde bude model, který se stala referenční pro první experiment v této práci – *Towards Agent-based Crowd Simulation in Airports using Games Technology* [25]. Cílem modelu bylo navrhnout a implementovat prostředí, které kombinuje fyzikální engine (v tomto případě PhysX⁵), před začátkem simulace vypočítané informace, uložené v prostředí (které urychlují výpočty během simulace), s agenty, kteří se řídí za pomoci modelu sociálních sil [15] a otestovat výkonnost implementace v poměru počtu entit k počtu snímků za sekundu. To má mít vliv na uvěřitelnost a pocit reálnosti simulace. Letištní terminál byl vybrán pro své členité prostředí. Model se pak sestává ze statických a dynamických složek – statickými se chápou dopředu vypočítané informace, jako směr k cíli, a dynamickými pak konkrétní chování agentů (rovnice modelu sociálních sil).

Základem modelu je prostředí, vytvořeno podle předlohy letiště Gatwick [26]. Prostředí bylo rozděleno na několik logických vrstev – *fyzická* (schody, překážky, ...), *navigační* (nejkratší cesty k cíli), *okupační* (jaké entity okupují daný čtverec), *vyhledávací* (z daného čtverce, které další čtverce v blízkém okolí jsou dosažitelné). Plocha prostředí je rozdělena na čtverce 5x5 metrů o celkové šířce 200 a výšce 150 metrů. V každém čtverci se pak uchovávají informace z jednotlivých vrstev. Příprava prostředí pro experimentování se skládá ze dvou kroků:

1. Do *fyzické* vrstvy se uloží informace o členitosti prostředí. Tyto informace představují reprezentaci toho, co se nachází v daném čtverci a ohodnocení – překážky mají hodnocení nejvyšší, tedy budou generovat největší sílu na agenty (odpuzovat), zatímco ve volné cestě je odpor minimální.
2. Dijkstrovým algoritmem se vyhledají nejkratší cesty z každého čtverce k pevně danému bodu na mapě a tato informace se uloží do *navigační* vrstvy v podobě směrového vektoru. Podle něj se pak entity orientují v prostoru – v každém kroku vědí přibližný směr k cíli.

⁵ <http://www.geforce.com/hardware/technology/physx>



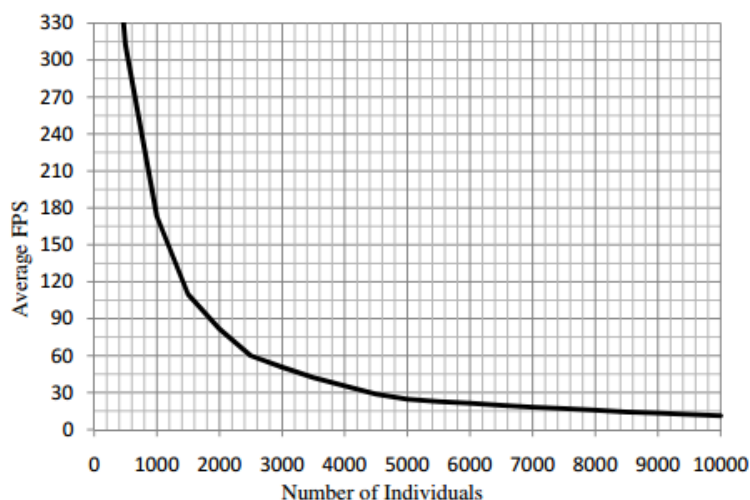
Obrázek 7 Navigační vrstva v modelu letiště Gatwick

Zdroj: [25]

Agenti pak přizpůsobují své chování na základě vlivů – hustoty a fyzických sil prostředí (překážky „odpuzují“). Startovní pozice agentů je v prostředí generována náhodně. Jejich cílem je posunout se do specifického místa na mapě (levý dolní roh), které lze vidět na obrázku 7. Agenti se snaží obcházet překážky (z fyzické vrstvy), nalézt optimální cestu k cíli (pomocí údajů uložených v navigační vrstvě) a vyhýbat se čtvercům, které jsou hodně obsazené (okupační vrstva). Čtverce v okolí (které agent zkoumá) se vybírají na základě informací z vyhledávací vrstvy. Na agenty pak působí sociální síla (reprezentovaná silovým vektorem), která ovlivňuje jejich pohyb prostředím. Ta je vypočítaná jako suma hodnot všech vrstev ve čtverci (x, y) . Možné kolidování jednotlivých agentů bylo ošetřeno systémem PhysX. Ten zajistil simulaci fyzických hranic entit a reakci na kolize – aby nedocházelo k překrývání.

S prostředím a agenty poté autoři provedli sérii experimentů, jejímž hlavním cílem bylo otestovat výkonnost implementace a navrhnout možná řešení pro její vylepšení. Prvním podcílem bylo nalézt počet jednotek, které se mohou v experimentu pohybovat zároveň, aby se dodržela tzv. interaktivní snímkovací frekvence [25]. Ta byla stanovena na hranici 30 snímků za sekundu. Tuto frekvenci se podařilo udržet v přibližném počtu 4200 jednotek. Důležitou poznámkou je také fakt, že jednotky se nechovají typickým způsobem, běžným pro pasažéry na letišti – jednotky v simulaci nemají povinné body, ani nenavštěvují zájmové lokace, pouze se snaží přesunout na cílovou pozici. Autoři však považují svou implementaci za dostatečný základ pro budoucí aktualizace frameworku,

který tuto funkcionalitu může přidat – pohyb entit v prostředí je realistický a přirozený. Následující graf ukazuje vztah mezi počtem entit a počtem snímků za sekundu.



Obrázek 8 Výsledky simulace letiště Gatwick

Zdroj: [25]

Jako významné mínus autoři uvádějí nutnost přepočítávat statická data při každé úpravě prostředí. Pokud bychom například chtěli přidat novou překážku, je nutné vypočítat znovu celou navigační vrstvu. Řešením by mohlo být rozdělit celé prostředí na několik pravidelných částí a aktualizovat navigační vrstvu pouze v té části, kde došlo ke změně. Jako další omezení je uvedeno rozlišení prostředí – rozdělení na čtverce o velikosti 5x5 metrů. V takové mřížce dochází ke zkreslování prostoru a ztrácí se detaily, jako například zaoblené zdi, nebo velmi malá překážka, menší než 5 metrů.

Výsledkem experimentů je poznatek, že takto navržený model a implementace dokáže při interaktivní snímkovací frekvenci simulovat přirozený pohyb tisíce jednotek a poskytuje tak základ pro budoucí rozšíření (skupiny, větší rozlišení prostředí, simulace chování na letišti). Model sociálních sil pak umožňuje přidání dalších vrstev do modelu (komunikační kanály, události).

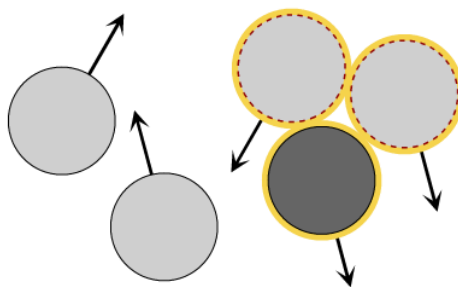
6 Experimentální část

V této kapitole je popsán postup od návrhu modelů až po jejich implementaci v jazyce Java, verze 8. Předpoklady, dle kterých jsou jednotlivé modely vybírány a metodika jejich návrhu. Modely jsou zde představeny formou diagramů a půdorysných plánů, znázorňujících jednotlivé experimenty. u každého modelu je také komentář popisující postup řešení daného problému z implementačního hlediska. v závěru kapitoly jsou poté specifikovány metriky, podle kterých budou modely vyhodnocovány. Teoretickým východiskem pro tuto část jsou práce *Towards Agent-based Crowd Simulation in Airports using Games Technology* [25], *Directing Crowd Simulations Using Navigation Fields* [27] a *Crowd Dynamics* [6].

6.1 Kritéria pro výběr modelu

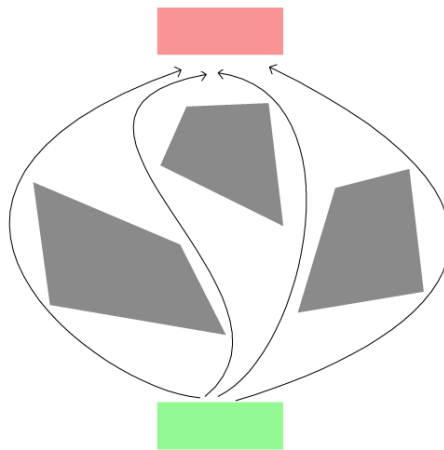
Při tvorbě návrhů jednotlivých modelů jsem se řídil stanovenými kritérii, která jsou zde uvedena a popsána. Kritéria byla vybrána v souladu se zaměřením práce.

1. **Prostorové interakce jednotlivých členů davu.** Tento bod je důležitý, jelikož modely mají zkoumat chování jednotlivých entit v závislosti na prostředí a okolních entitách. v každém modelu se entity budou navzájem prostorově ovlivňovat svou přítomností. Působit na sebe svou rychlostí. Velikost jednotlivých členů zde také hraje nemalou roli. Modely musejí nabídnout dostatek prostoru pro navigaci a průtok davu zúženými prostory. Průtok můžeme simulovat jako průchod jedinců omezeným prostorem a měřit jej jako vztah počtu jedinců k jednotce času, jak je stanoveno v kapitole Modelování davových jevů.



Obrázek 9 Ukázka kolize a navigace entit.

2. **Specifikace jednoznačných hlavních a vedlejších cílů.** Prostředí by mělo mít jasně definované druhy cílů, které se mají splnit (formou typů míst, které je třeba navštívit) pro každou jednotku nezávisle. Každá z jednotek by měla obdržet seznam typů cílů v jasně daném vzoru – vstup, 1. nepovinné cíle, povinný průchod, 2. nepovinné cíle, východ. Každé jednotce musí být určeny cíle takové, které je schopna splnit (negerovat cíl, který je v nepřístupném prostoru).
3. **Chování jednotlivců.** Agent (entita, jednotka, jedinec) by měl plnit své cíle, které mu jsou stanoveny při jeho vytvoření, v pořadí, v jakém jsou specifikovány. Důležitou roli také hraje způsob, jakým se k cílům zachová – dodržení povinnosti cíle (vynechá cíl, projde jím). Jedinec by se měl také umět prostorově orientovat – ví, kde se nachází, zná umístění svého cíle (nemusí jej hledat po celé mapě) a dokáže dynamicky přizpůsobit svou cestu k cíli podle hustoty chodců. Souhrnně by tedy agent měl umět analyzovat své okolí a tomu přizpůsobovat své chování.
4. **Prostorově členité prostředí s překážkami.** Smyslem tohoto bodu je formulace představy o vhodně členitém prostředí. Prostředí je vhodně členité za předpokladu, že příliš nezpomaluje jedince a podporuje jej v průchodu prostorem. Poté je možné simulovat průchody entit členitým prostředím a srovnávat s výsledky experimentu, ve kterém se tok v prostředí usměrní pomocí navigačních prvků.



Obrázek 10 Ukázka členitého prostředí

5. **Výběr vhodného prostoru pro návržení virtuálního prostředí.** Toto kritérium navazuje na předchozí bod a definuje, jaká mapa bude zvolena pro vytvoření virtuálního prostředí. Model by měl obsahovat takové fyzické prostorové

uspořádání, které bude vycházet z reálné předlohy. Účelem tohoto bodu je zajistit, aby výsledky byly interpretovatelné ve vztahu ke skutečným prostorám.

6. **Měřitelnost událostí v jednotlivých modelech.** Události, jako průchod prostorem, počet navštívených míst, rychlost toku davu atd. je nutné správně kvantifikovat tak, aby se dalo říci, že změnou vstupních parametrů došlo k nějakému viditelnému posunu v naměřených hodnotách. Metriky aplikované na modely budou vycházet z poznatků v kapitole Modelování davového chování.
7. **Multiagentní přístup.** v modelech se budou vyskytovat nezávislé entity, které budou mít určitý cíl a možnost (cestu), jak se k cíli dostat. Každá taková entita bude charakterizována svými rozměry, hmotností, rychlostí a body zájmu, kterých se bude snažit během cesty dosáhnout.

V předpokladech byly zmíněny některé pojmy, které budou využívány a je tedy nutné je popsat dále. Prvním základním pojmem je *prostředí*. Prostředí bude modelováno v síti s konkrétním rozlišením jednotlivých čtverců, do které pak budou zaneseny informace o překážkách, bodech zájmů a vstupně/výstupních pozicích.

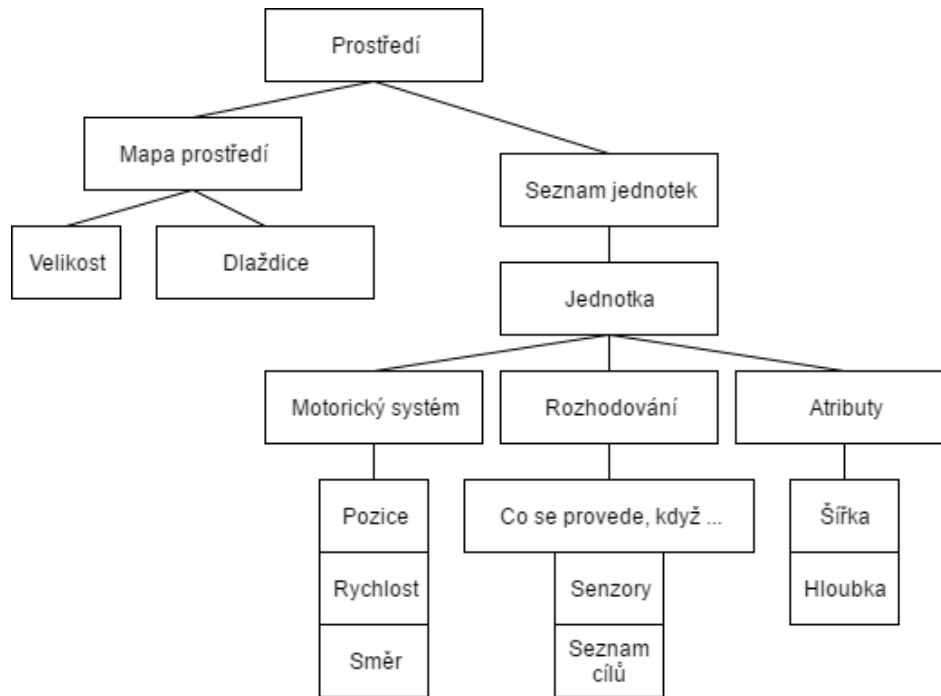
Překážky představují zdi nebo nepřístupná místa, kterým se budou entity snažit vyhnout, bude-li to možné, a v nejhorším případě s nimi kolidovat. Překážky představují nejjednodušší součást modelu, jelikož jedinou informací, kterou udržují, je ta, že jsou nepřístupné.

Body zájmu jsou vyznačená místa, kterými mohou entity procházet. V těchto místech dojde k pozastavení na náhodnou dobu z intervalu specifikovaného pro každou entitu. Tím se bude simulovat návštěva obchodů nebo odbavovací přepážky v případě letištního terminálu. Body zájmů v sobě ponese informaci o svém účelu, počtu navštívení a času v nich stráveném. Tyto body jsou pro každou entitu voleny náhodně.

Vstupní a výstupní pozice jsou místa, na kterých dochází ke generování, resp. mazání entit. Entity v modelech budou mít krátkou životnost začínající od vstupního bodu a končící ve výstupním. Těchto pozic bude několik a entity budou generovány vždy na náhodně vybraném vstupním místě. při vytváření entit potom bude vybráno koncové místo.

6.2 Konceptuální model vybraných davových jevů

V souladu s poznatky a kritérii z předchozích kapitol je sestavena následující struktura modelu:



Obrázek 11 Struktura modelu

1. *Prostředí* je hlavní prvek modelu. Udržuje instance entit v *seznamu jednotek*. Také zajišťuje přidělování procesorového času všem jednotkám a stará se o správu doplňkových procesů, jako například řízení událostí. Dále obsahuje informace o členitosti prostoru, ve kterém se jednotky mají posouvat – mapu prostředí. Prostředí je zodpovědné za spouštění a průběh experimentů. Zajišťuje také uchovávání stavů experimentů, zachytávání a ukládání analytických údajů. V průběhu životnosti prostředí prochází několika stavy – *inicializací, spuštění simulace, průběh simulace, ukončení simulace* a *uložení výsledků, ukončení prostředí*. Ve fázi *inicializace* se deklarují proměnné nutné pro samotný běh prostředí a instance pro mapu a seznam jednotek. V této fázi se také vygeneruje mapa z uloženého stavu, nebo z výchozího typu čtverce, pokud žádný stav není nalezen. *Spuštění, průběh* a *ukončení* simulace souvisí se stavy samotných experimentů – při *spuštění* se nastaví proměnné pro sběr dat a započnou generovat jednotky. V *průběhu* simulace už prostředí jen udržuje počet jednotek a obsluhuje události od uživatele a přiděluje procesorový čas jednotkám. Při *ukončení simulace* se již přestávají provádět jakékoliv výpočetní úkony, související s jednotkami a uloží se *výsledky experimentu*. Ve fázi *ukončení*

- prostředí* již pouze dojde k ukončení celého modelu a uvolnění systémových prostředků.
2. *Mapa prostředí* reprezentuje virtuální prostředí, které je vytvořeno na základě reálné předlohy. Informace o členění je zanesena do pravidelné mřížky. Každá buňka mřížky poté reprezentuje stav konkrétního místa – překážku, průchozí část nebo jinak významný bod. Významné body jsou například vstupní/výstupní body, nebo povinné pasáže. Mapa prostředí umožňuje interaktivní změnu stavů jednotlivých čtverců. Při každé změně není nutné provádět žádné rozsáhlé výpočty. Mapa prostředí v tomto modelu umožňuje snadno splnit jedno z kritérií pro výběr experimentů – prostorově členité prostředí s překážkami. Mapa je tvořena prvky zvané dlaždice (čtverce).
 3. *Dlaždice* jsou objekty o velikosti $N \times N$ metru. Každá z nich reprezentuje konkrétní místo ve virtuálním prostředí, v mřížce pak představují celou mapu. Také uchovávají první důležitou metriku – informaci o hustotě vyjádřenou ve formě počtu entit, které se na daném místě vyskytují. Využívají se i pro navigaci entit prostorem.
 4. *Jednotky* představují chodce, kteří se v daném prostředí pohybují. Mají své atributy, senzory, možnost pozorovat prostředí a rozhodovat se na základě zjištěných poznatků. Jejich cílem je splnit frontu úkolů, která je představena typem cílů, které mají navštívit. Cíle si jednotky vyhledávají nejkratší přímou vzdáleností od své současné pozice. Během své životnosti se nacházejí v několika stavech – *vytvořena*, *provádí frontu úkolů* a *ukončena*. Pohyb prostředím je řešen na dvou úrovních (globální a lokální). Globální pohyb se týká hledání trasy přes volné dlaždice a v lokálním se jednotky vyhýbají kolizím s ostatními. Informace o prostředí získávají přímo z mapy prostředí, kde mají přístup ke stavům jednotlivých dlaždic (typ dlaždice, hustota). Jsou schopny vyhledávat optimální trasu k současnému cíli na základě hustoty dlaždic. Každá z jednotek sleduje kromě prostředí z globální úrovně i své nejbližší okolí v lokální úrovni (ostatní jednotky). Právě v lokální úrovni přizpůsobují jednotky svou rychlostí okolí, aby nedocházelo k překrývání s ostatními a zároveň se stabilně posouvaly správným směrem. Jednotky při posunu také dbají na osobní prostory své i ostatních. Algoritmus chování je jednoduchý: Dokud jsou ve frontě cílů dostupné cíle, tak tyto cíle navštěvuj. Ke každému cíli si najdi neoptimalnější cestu (optimum = vyvážený poměr mezi hustotou a ušlou vzdáleností). Během cestování po vypočítané trase si kontroluj vzdálenost mezi sebou, překážkami a okolními entitami. Pokud už jsi prošel všemi cíli, skonči.

Na základě kritérií z předchozí kapitoly je model dále doplněn o následující body:

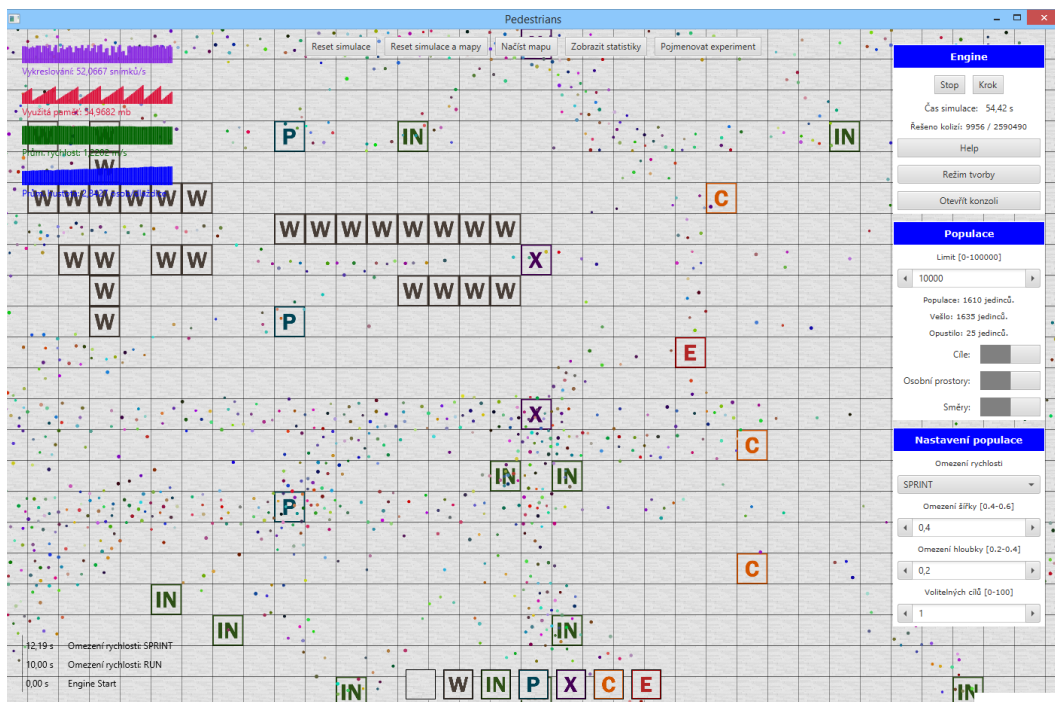
1. *Cíle* představují typ dlaždice, na kterou se má entita posunout. Ukládají se každé entitě náhodně před její inicializací do její fronty cílů. Cíle, které se vybírají entitám, jsou rozděleny podle následujícího klíče: 1 vstup, N náhodných (pro $N \geq 0$) prvních nepovinných, 1 povinný, N náhodných (pro $N \geq 0$) druhých nepovinných, 1 výstup. Vstupní cíl splní každá entita ihned při vložení do prostředí. Poté už jednotky tyto cíle odebírají ze své fronty cílů v momentě, kdy jich dosáhnou.
2. *Specifikace mapy*, která bude zanesena do mapy prostředí, náleží každému experimentu zvlášť. U každého experimentu je pak výběr odůvodněn.
3. *Metriky* slouží pro kvantifikování procesů, které probíhají v modelu. Metriky jsou vybrány v souladu s kapitolou *Měřitelné charakteristiky kvantifikující procesy davové dynamiky* a jsou zde dále rozvedeny. **Rozměry jedince** (m) se každé jednotce generují náhodně na základě průměrných hodnot populace, specifikovaných v této práci. Jednotce se poté přidělí jednotný rozměr, který reprezentuje průměr její masy a rovná se maximu z šířky a hloubky jednotky. Tato masa se poté bere jako unifikovaný rozměr jednotky. **Rychlost pohybu** jedince (m/s) se v modelu vypočítává na základě rozdílu mezi současnou a předchozí polohou, děleno časem, který uplynul mezi těmito místy. Pro tuto veličinu je tedy nutná znalost aktuálního času simulace (runtime) a přesné polohy jednotky. **Hustota davu** (osob/m²) je monitorována pomocí každé dlaždice, která dokáže přesně určit, kolik jednotek se na ní nachází. Součet ploch jednotlivých entit nikdy nemůže přesáhnout celkovou volnou plochu dlaždice. Dlaždice poté dále měří i **tok davu** – jelikož je dlaždice prostorově omezena, může sledovat informaci o tom, kolik entit přes ní přešlo za jednotku času (za 1 sekundu). **Velikost osobní zóny** (m) je specifikována pro všechny entity a vychází z [10].
4. *Multiagentní přístup* – každý jedinec je nezávislý na ostatních a má možnost se rozhodovat, kudy půjde a jaké cíle vybere. Trasa se vybírá na základě algoritmu hledání neoptimálnější cesty. Cíle jsou vybírány v nejkratší přímé vzdálenosti od současné pozice. Agenti mohou pozorovat své okolí a reagovat na okolí (úprava trasy, výběr cíle).

a.

6.3 Implementace konceptuálního modelu

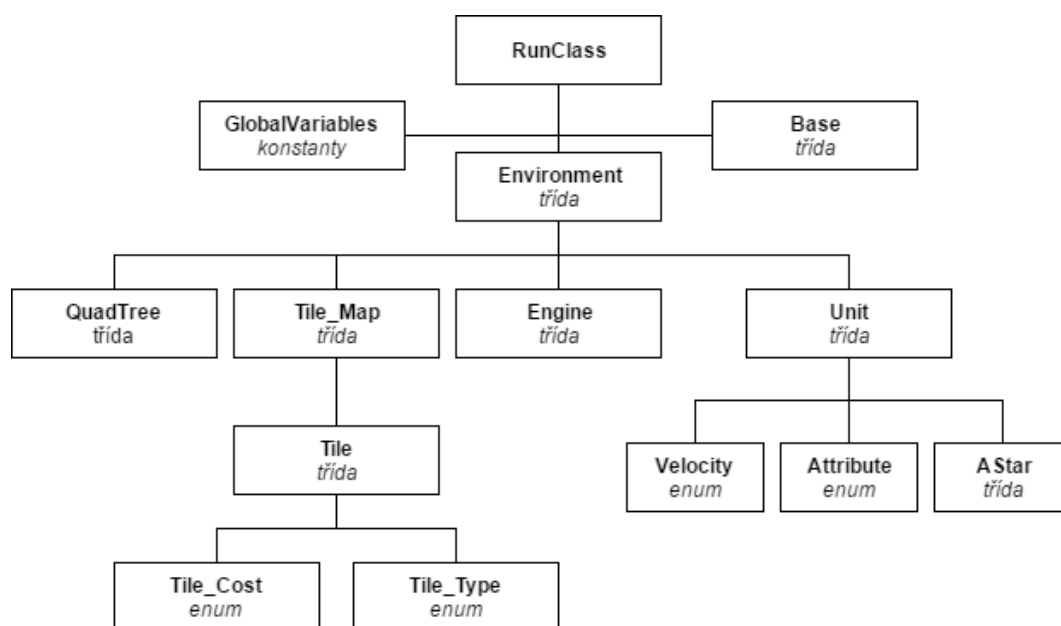
Náplní této kapitoly je popis implementace modelu v jazyce JAVA verze 8, který slouží jako výchozí nástroj pro jednotlivé experimenty. Program je logicky rozčleněn na několik částí: prostředí, jednotky, navigace, grafická vizualizace a databáze. Tomuto simulačnímu prostředí byl přiřknut název „Pedestrians“.

Tento softwarový nástroj je založen na principu, kdy každá úprava prostředí a parametrů se ihned promítá a není třeba celý experiment restartovat. Pokud upravíme atributy, pomocí kterých se generují entity, tak od této chvíle každá další bude čerpat z tohoto nastavení.



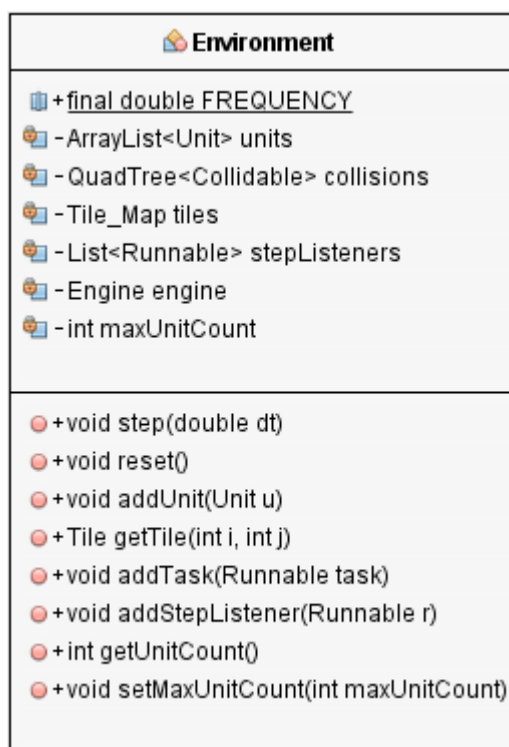
Obrázek 12 Ukázka programu Pedestrians

V samotném uživatelském prostředí je možné upravovat počet entit, které se budou najednou v prostředí pohybovat a atributy nově generovaných. Po zadání limitního počtu poté prostředí generuje na vstupních dlaždicích nové entity do doby, než počet všech dosáhne omezení. Toto omezení se dodržuje po celý průběh simulace – program nevygeneruje novou entitu do doby, než nějaká opustí prostředí.



Obrázek 13 Zjednodušená struktura projektu Pedestrians

Prostředí (třída Environment.java)



Obrázek 14 Zjednodušený diagram třídy Environment.java

Simulační prostředí je pilířem programu. Stará se o aktualizaci vnitřních stavů, reakci na vstupy a zpracování výstupů. je úzce provázáno z vizualizací, tedy grafikou aplikace, ať už ve formě zobrazení jednotlivých elementů v něm se vyskytujících, nebo vykreslení statistických údajů. Jeho jádrem je poté metoda step, která vždy provede

aktualizaci prostředí a všech jednotek o specifikovaný časový posun v sekundách. Jinak řečeno tato metoda posouvá v čase stavy prostředí. Jednotlivými stavy se rozumí aktuální poloha jednotek. na konci každé aktualizace jednotky rozhodnou, kam se posunou. Metoda `step` tedy v základu slouží pro vymezení procesorového času všem jednotkám, aby tuto příležitost mohly využít pro své rozhodování. při dostatečně rychlém volání a zpracování metody `step` pak dochází k iluzi, že se všechny jednotky rozhodují najednou. Tento jev je ještě podpořen paralelním zpracováním vícera jednotek najednou. Technologie, která toto umožňuje, je novinkou JAVA verze 8 a je označována jako paralelní proudy (Stream API).

Metoda `step` (hlavní funkcionality třídy `Environment.java`)

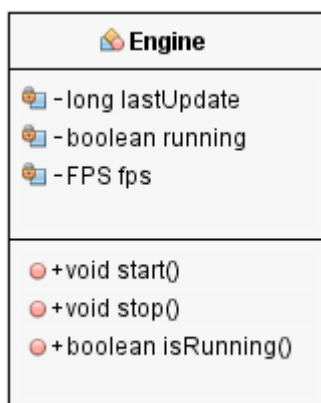
Tato metoda je souborem několika kroků – sestavení systému pro optimalizaci detekce kolizí, aktualizace jednotek, provedení dodatečných úkonů a vykreslování.

Metoda zajišťuje následující hlavní činnosti:

1. Vykreslení všech prvků v prostředí.
2. Vložení všech kolizních prvků do systému kolizí.
3. Odstranění jednotek, které již opustily prostředí.
4. Aktualizaci stavů všech jednotek. U každé jednotky se volá metoda `update`, pro kterou je důležitý parametr časové difference oproti předchozí iteraci.
5. Provedení úkonů, které lze libovolně řadit do fronty úloh. Ta slouží pro dodatečné upravování algoritmu metody `step` mimo její vlastní tělo.
6. Aktualizace proměnných, které řídí běh vlastní metody `step`. Nejprve se jedná o proměnnou `step`, která je sumou uběhlého času mezi jednotlivými aktualizacemi. Tato hodnota je využívána pro záměrné „zpomalení“ rychlosti aktualizace všech prvků prostředí. Metodu `step` lze volat ve velmi rychlých (i milisekundových) intervalech, což může mít neblahý vliv na výkonnost implementace (velké propady v aktualizací frekvenci). Proto je do kódu zavedena konstanta `FREQUENCY`, která určuje četnost aktualizace prostředí. Četností se rozumí počet aktualizací za vteřinu. Proměnná `step` poté jen slouží pro porovnání, zdali je dosaženo cílové frekvence a bude umožněno provést aktualizaci.

Metoda `step` sama o sobě umožňuje provést pouze jeden krok a pro plynulost simulace je nutné zavést prvek, který jí bude automaticky volat.

Třída Engine



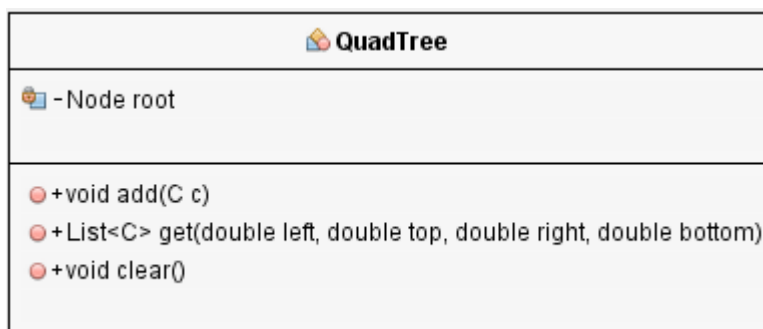
Obrázek 15 Zjednodušený diagram třídy Engine.java

Vychází z třídy *AnimationTimer.java*, která je součástí balíku JavaFX 2.0. Po inicializaci a spuštění má tento objekt jediný účel – opakovaně provádět obsah metody *handle*.

Četnost opakování je řízena samotným systémem JavaFX a je limitována na 60 aktualizací za sekundu. Zděděná metoda *handle* potom obstarává krokování prostředí voláním metody *step*. Jejím důležitým prvkem je parametr *now*, datového typu *long*, který obsahuje aktuální čas v nanosekundách. Toho je využito pro počítání diferencí mezi jednotlivými voláními – odečtením předchozího času od současného vznikne rozdíl v nanosekundách, ten je vydělen číslem $1e9$ ($1\text{ s} = 1000000000\text{ ns}$) a vyjde časový odstup od předchozího volání v sekundách.

Jak již bylo řečeno dříve – metoda *step* „posouvá“ v čase stav prostředí a prvků v něm obsažených. Stav jsou pak charakterizovány pozicemi, ve kterých se jednotky nachází. A jelikož základním předpokladem simulace je, aby co nejvíce odpovídala realitě, je důležité nalézt takové pozice, ve kterých se jednotky nebudou navzájem překrývat. Tento mechanismus se nazývá detekce kolizí a je přímou součástí každé jednotky. Implementace, použitá v tomto programu, bude popsána v dalších podkapitolách. Simulační prostředí se však na tomto mechanismu podílí také – ve fázi před samotnou aktualizací jednotek.

Systém detekce kolizí (QuadTree.java)



Obrázek 16 Zjednodušený diagram třídy *QuadTree.java*

Detekce kolize je výkonově náročná operace, kdy je třeba otestovat všechny prvky v okolí bodu, který chceme někam posouvat. při 10 jednotkách je to $10^2 = 100$ iterací, při 1000 pak $1000^2 = 1000000$ iterací a tak dále. je tedy vhodné zavést do tohoto systému logiku, která určí pouze nejbližší objekty, jež se potenciálně mohou účastnit kolize.

Kvadrantový strom rekurzivně dělí prostor shora-dolů vždy na čtyři rovnoměrné části a do každé vkládá kolizní členy. Výška stromu a velikost podprostorů v listech stromu je otázkou nastavení, na jak malé části chceme prostor dělit. V implementaci, která vychází ze základů obsažených v [28] a je součástí programu *Pedestrians*, není výška stromu ani velikost podprostorů limitována - v jednotlivých uzlech a koncových listech je vždy jedna jednotka. Toto řešení zjednodušuje implementaci, jelikož do každého uzlu nemusíme vkládat kolekci, ale jen odkaz na jednotku. Také odstraňuje problém předešlé implementace, kdy člen mohl kolidovat s prvky v jiném podprostoru – zde se vždy pro každou jednotku najdou prvky v jejím bezprostředním okolí.

Nyní, když je prostor rozdělen, nemusí jednotky testovat kolize oproti všem ostatním, ale pouze k těm, které jsou v bezprostřední blízkosti. Poté se počet testování kolizí pohybuje relativně dle hustoty (osamocená jednotka nemusí provádět detekci vůbec, resp. ze systému obdrží kolekci s jedním členem – se sebou).

Porovnání účinnosti implementace oproti naivnímu testování všech jednotek (v kvadrantovém stromu jsou vybrány jednotky v okolí 1 m od výchozí na každou stranu, při maximální hustotě, pro jednotky 0.5 m široké a 0.5 m hluboké, což se rovná hustotě 4 jednotky na 1 m², pro vybranou plochu 2x2 m poté 16 jednotek na 4 m²):

Tabulka XII Porovnání kolizních technik.

Počet jednotek	Počet testovaných možných kolizí	
	Naivní implementace	Kvadrantový strom
10	100	100
100	10000	1600
1000	1000000	16000

Z tabulky je patrné, že tato optimalizační technika velmi redukuje celkový počet všech možných kolizí a tím pádem umožňuje do systému zapojit mnohem více simulačních jednotek.

Na tento systém poté přímo navazuje algoritmus vyhýbání se kolizím⁶, který obsahuje každá jednotka. Vychází ze zdrojového kódu [29] ukázky v jazyce Python. V originální implementaci se počítá s polohou každé jednotky v simulaci – jinak řečeno, každá jednotka se vyhýbá kolizi se všemi ostatními. To značně zvyšuje využití prostředků počítače. Implementace v programu *Pedestrians* využívá kvadrantového stromu, aby omezila počet možných kolidujících jednotek a tím pádem zvýšila výkonnost samotné simulace. Navíc přidává kolizi se statickými objekty, jako jsou stěny. Základem algoritmu je znalost současné pozice (x,y), pozice cíle (a,b), kam se jednotka chce posunout, a pozice okolních kolizních prvků.

Předkládám zde výňatek zdrojového kódu, jelikož jej považuji za hlavní pilíř simulování pohybu jednotek. Algoritmus je následující:

⁶ Collision avoidance

1. Inicializace proměnných, dle kterých se vypočte další posun:

```
angle = atan2(b - y, a - x); //směr jednotky
double dx = cos(angle); //rozdíl současného a budoucího x
double dy = sin(angle); //rozdíl současného a budoucího y
```

2. Získání seznamu všech kolizních členů z okolí:

```
double left = x - range; //vzdálenost od jednotky (3 m)
double right = x + range;
double top = y - range;
double bottom = y + range;
```

```
List<Collidable> vicinity = COLLISIONS.get(left, top, right, bottom);
```

3. Iterace přes všechny kolizní členy:

```
for (Collidable u : vicinity)
{
    if (u.equals(this)) continue; //neřešit kolizi se sebou samým

    //délka přepony (vzdálenost mezi tímto a u členem) na druhou
    double d = pow(hypot(x - u.x, y - u.y), 2);

    double p = pow(u.padding, 2); //padding zde značí osobní prostor

    //atan2, úhel sevřený mezi osou x a přímkou, vedenou bodem [x,y]
    angle = atan2(y - u.y, x - u.x);

    //pokud známe úhel, délku a počáteční bod přímky,
    //můžeme vypočítat její konečný bod
    //x2 = x1 + délka * cos(úhel)
    //zde pouze hledáme opačný případ a ten přičítáme k diferenci
    dx += cos(angle) / d * p;
    dy += sin(angle) / d * p;
}
```

4. Finální posun pozice byl nalezen, vypočteme směr a vzdálenost:

```
angle = atan2(dy, dx);

double magnitude = hypot(dx, dy);

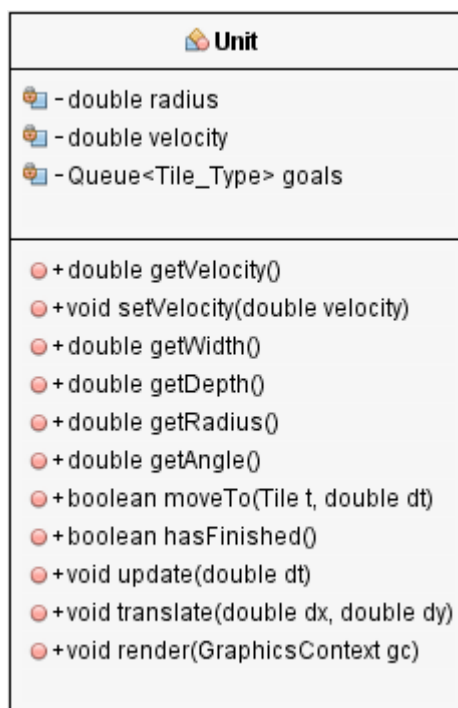
//proměnná, která upravuje maximální možnou rychlost
//A,B jsou konstanty, které pouze upravují míru zpomalení
//proměnná speed je zavedena proto,
//aby jednotky dynamicky měnily svou rychlost v závislosti
//na okolí
double speed = min(1, A + magnitude * B);

//dt je časová diference mezi současnou a předchozí aktualizací
//jinak řečeno, kolik času uběhlo od poslední aktualizace
dx = cos(angle) * dt * velocity * speed;
dy = sin(angle) * dt * velocity * speed;
```

5. Provedeme posun o dx, dy:

```
translate(dx, dy);
```

Jednotky (třída Unit.java)



Obrázek 17 Zjednodušený diagram třídy Unit.java

V programu *Pedestrians* představují lidské účastníky – chodce. Ti se poté pohybují prostředím a jejich účelem je plnit cíle stanovené v seznamu cílů. Jednotky (entity) prochází prostředím cestou nejmenšího odporu – vybírají si dlaždice s nejmenší hustotou a vybírají si za cíl dlaždice nejbližší své současné pozici. Životní cyklus jednotek začíná vstupem do prostředí s vygenerovanými atributy a končí u východu ze simulace. Každá z nich se skládá ze základních komponent:

Tabulka XIII Komponenty jednotek.

Atributy	Pohyb	Logika
Osobní prostor [m]	Vyhýbání se kolizím	Plnění cílů
Hloubka [m]		
Šířka [m]	Aktualizace pozice	Hledání cesty
Rychlost [m/s]		

Osobní prostor – je rádiusem kruhu, který vymezuje bezprostřední okolí každé jednotky. je využit v algoritmu vyhýbání se kolizím.

Hloubka, šířka – stanovuje rozměry každé jednotky, které jsou náhodně generovány každé jednotce při jejím vytváření a interval hodnot odpovídá údajům uvedeným v kapitole Modelování davových jevů.

Rychlost – uvedená v metrech za sekundu a určuje maximální možnou rychlost, kterou se jednotka může pohybovat. Aktuální rychlost pak závisí na hustotě a rychlosti ostatních jednotek v okolí.

Vyhýbání se kolizím – algoritmus, který je základem pro plynulý pohyb jednotky a byl již popsán dříve.

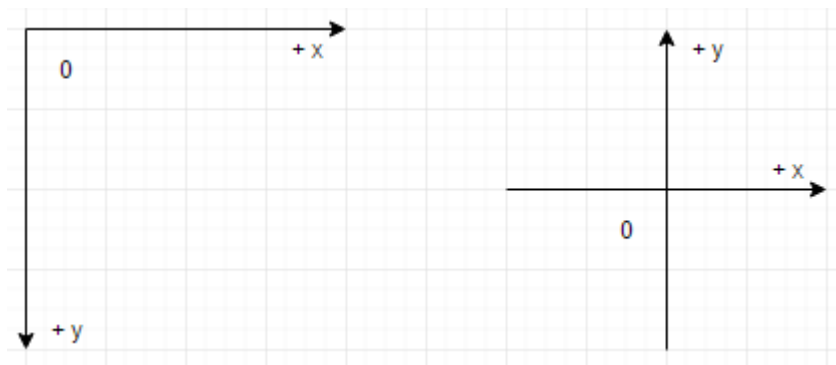
Aktualizace pozice – soubor metod a proměnných, určujících střed každé jednotky. Metody poté aktualizují pozici středu v mapě prostředí a tím vytváří pohyb v čase.

Plnění cílů – v programu uvedena jako metoda update. Tato metoda je prostředím volána u každé jednotky v každém simulačním kroku a obsahuje rozhodovací model. Logiku této metody je možné rozložit na několik bodů:

1. Fronta typů dlaždic určuje ty, které se mají projít.
2. Pro každý typ se vyhledá přímost vzdáleností nejbližší dlaždice z aktuální pozice jednotky a ta se uloží jako současný cíl.
3. Z aktuální pozice se nalezne neoptimálnější cesta k současnému cíli.
4. Jednotka se po trase posouvá směrem k současnému cíli.
5. Pokud do cíle dorazila, odebere se typ cíle z fronty – typ byl navštíven.
6. Pokud není fronta typů prázdná, algoritmus se opakuje.

Poziční systém

Neboli systém souřadnic v kartézské soustavě. Program Pedestrians je založený na dvourozměrné reprezentaci prostoru s osami x a y . Na prostředí je poté nahlíženo z pohledu shora a souřadnice se počítají v metrech (lze pracovat i s desetinnými čísly).



Obrázek 18 Porovnání pozičních systémů

Počátek soustavy souřadnic je v levém horním rohu vizualizace. Tento způsob je zvolen v souladu se systémem vykreslování v JavaFX, kde počátek je též v tomto místě. V původním návrhu aplikace byl počátek ve středu vizualizace a pro správné vykreslování bylo nutné před každou aktualizací obrazu souřadnice posunout doprava dolů, jihovýchodně, a poté celý systém posunout na správné místo, a to pro každý vykreslovaný objekt zvlášť.

Důležité je také zdůraznit, že v implementovaném systému jsou přípustné pouze kladné souřadnice.

Reprezentace mapy (třída `Tile_Map.java`)

Mapu, nebo také terén – prostředí, ve kterém se jednotky pohybují, lze logicky členit různými způsoby. Jako finální řešení byla použita mřížka pravidelných čtverců – dlaždic. Návrh mřížky vychází z třídy `Matrix.java`. Ta je od počátku koncipována s ohledem na jednoduchou implementaci a robustnost. Umožňuje pomocí generických tříd vytvářet matice z jakýchkoliv datových typů. Obsahuje jednoduché metody pro vkládání a získávání prvků matice. Klasickým přístupem by bylo založit maticovou strukturu na dvourozměrném poli, pro tuto však bylo zvoleno pole jednorozměrné a index se poté počítá jednoduchou funkcí ($\text{index} = i * \text{výška matice} + j$). Iterace a streamování pole o jednom rozměru je implementačně jednodušší, než n-rozměrného.

Třídou *Matrix.java* poté rozšiřuje třída *Tile_Map.java*. Přidanou funkcionalitou je zde práce s uloženými stavy mapy v databázi a omezení datového typu obsažených objektů na třídu *Tile*, která reprezentuje jednotlivé pravidelné dlaždice na mapě.

Mapa prostředí poté poskytuje následující funkcionalitu:

1. Logické uspořádání dlaždic v prostoru.
2. Vkládání a získávání dlaždic podle indexu (i náhodně) v matici.
3. Získávání dlaždic podle predikátů (predikátem může být funkce, která filtruje dlaždice konkrétního typu dlaždice, nebo obsazené plochy atd.).
4. Iteraci přes všechny dlaždice a jejich streamování.

Jak již vyplývá z popisu mapy, její základ tvoří dlaždice datového typu *Tile.java*. Dlaždice obsahují následující komponenty:

Tabulka XIV Komponenty dlaždice

Atributy	Metriky	Funkcionalita
Index v matici [i,j]	Obsazená plocha [m ²]	Lambda kód, kterým lze přidávat funkčnost dlaždici (např. řazení do fronty)
Velikost jedné strany [m]	Hustota [obsazeno/plocha]	
Plocha[m ²]	Prošlo jednotek [počet]	
Typ dlaždice		

Důležitým prvkem každé dlaždice je její typ (výčtový typ *Tile_Type.java*). Ten definuje tři důležité vlastnosti – pořadí, v jakém jednotlivé typy na sebe navazují, průchodnost dané dlaždice a povinnost jednotky projít daným prostorem. Pořadí je určeno zápisem ve zdrojovém kódu, kdy typy následují v tomto pořadí: COMMON (dlaždice, která neplní žádnou roli, reprezentuje podlahu), BLOCKED (překážka), START (vstupní dlaždice do prostředí), AMENITIES (zábavné prvky jako stánek s občerstvením apod.), CHECK_IN (místo, který je nutné projít, aby se dalo postoupit do další oblasti), CARE (obdoba zábavných prvků, přístupná až po projití typem CHECK_IN), EXIT (výstup z prostředí).

Navigační algoritmus (A*)

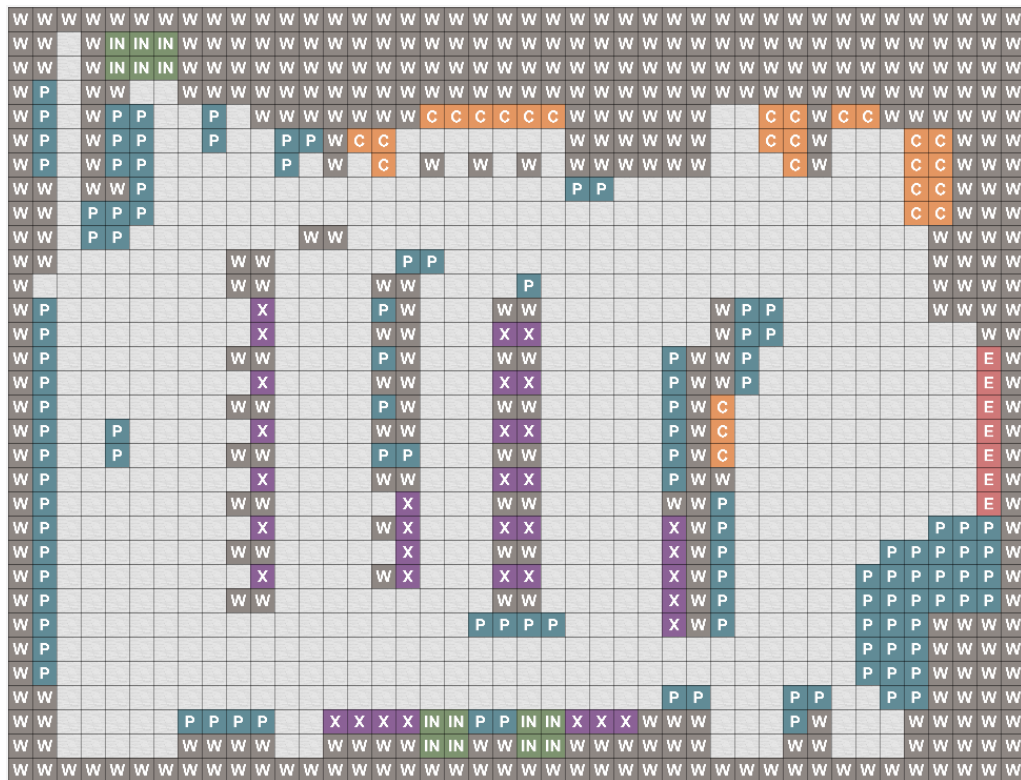
Hledá optimální cestu mezi dvěma body v kladně ohodnoceném grafu. je principiálně podobný prohledávání do šířky (*Breadth-first search*), avšak místo klasické FIFO fronty používá frontu prioritní. ve frontě se jednotlivé uzly cesty řadí podle funkce $f(x)$. Ta je součtem zbývající délky $g(x)$ a heuristické funkce $h(x)$. Uzly se poté řadí tak, že ty s nejnižší hodnotou funkce $f(x)$ mají nejvyšší prioritu a jsou ve frontě na předních pozicích. v programu *Pedestrians* se proměnné $g(x)$ nepřizpůsobuje zbývající délka, ale hodnota (cena) daného uzlu grafu – hodnotou je zde konkrétní hustota dané dlaždice. Tím se upravuje logika algoritmu – nehledáme cestu, která bude krátká, ale cestu, která povede přes nejvolnější dlaždice.

6.4 Experimenty

Pro testování modelu a implementace byly navrženy následující 3 experimenty:

1. Průchod letištní halou ze vstupu do východu
2. Průchod letištní halou ze vstupu do východu, doplněno o strukturu cílů
3. Evakuace osob z letištní haly skrze východ

Předlohou experimentů se stal jižní terminál třetího patra letiště Gatwick v Londýně [26]. Tato mapa byla zvolena v souladu s prací [25] a byla zanesena do sítě o velikosti 40 x 30 dlaždic. Při velikosti jedné strany dlaždice 5 m, je z pohledu shora šířka prostředí 200 a výška 150 metrů.



Obrázek 19 Prostředí v programu Pedestrians

Pro přehlednost zde uvádím legendu pro typy dlaždic:

- W – překážka, zeď, neprůchozí část
- IN – vstup
- E – východ (povinné pro každý experiment)
- P – stánky, informace
- C – péče o zákazníka, sociální zařízení
- X – kontrolní body

Pro experiment 1 a 2 jsou vstupy do prostředí ohodnoceny pravděpodobností, že v nich bude vygenerována entita – tím se simuluje vytíženost vstupu (hlavní, vedlejší). Vstupy na severu mají stanovenou pravděpodobnost 50 % a vstupy na jihu 10 %. Tato míra je stanovena odhadem na základě velikosti vchodů, zakreslených na půdorysném plánu letiště. Pravděpodobnost na vstupech je pak po dobu všech experimentů zachována, aby byly dodrženy vždy stejné počáteční podmínky pro generování entit.

Vstupní parametry (rychlost, velikost jednotek, počet cílů) jsou stanoveny na začátku každé simulace a jsou nezměněny po celou dobu běhu. Následující tabulka ukazuje počáteční konfiguraci experimentů.

Tabulka XV Konfigurace experimentů

Označení experimentu	Struktura cílů	Limit času simulace	Limit rychlosti pohybu
1-a	IN – E	40 minut	1,9875 m/s (rychlá chůze)
1-b	IN – E	40 minut	5,45 m/s (sprint)
2-a	IN – 1 P – X – 1 C – E	40 minut	1,9875 m/s (rychlá chůze)
2-b	IN – 5 P – X – 5 C – E	40 minut	1,9875 m/s (rychlá chůze)
2-c	IN – 1 P – X – 1 C – E	40 minut	5,45 m/s (sprint)
2-d	IN – 5 P – X – 5 C – E	40 minut	5,45 m/s (sprint)
3-a	IN – E	20 minut	1,0875 m/s (pomalá chůze)
3-b	IN – E	20 minut	1,5 m/s (normální chůze)
3-c	IN – E	20 minut	1,9875 m/s (rychlá chůze)
3-d	IN – E	20 minut	3,4125 m/s (běh)
3-e	IN – E	20 minut	5,45 m/s (sprint)

V prostředí jsou monitorována data jednotlivých experimentů. Četnost snímání údajů je stanovena na 3 sekundy. Po uplynutí tří-sekundového intervalu se odečtou hodnoty všech jednotek, dlaždic a prostředí – to je jeden záznam v databázi.

Struktura údajů, které jsou zaznamenány, je logicky rozčleněna na několik kategorií podle významu (průměry, maxima, doplňková data, počítadla). Jednotlivé údaje záznamu jsou zde uvedeny ve formátu údaj [jednotka]. Kategorie jsou sestaveny následovně:

Průměry – tyto údaje jsou vypočteny aritmetickým průměrem a vychází z aktuálních stavů jednotek a dlaždic, v čase jejich snímání.

- Průměrná rychlost [m/s]. Je vypočtena na základě aktuální skutečné rychlosti posuvu jednotek. Získáno iterací stavy všech jednotek.
- Průměrná uražená vzdálenost [m]. Tento údaj je pravidelně aktualizován v každé iteraci prostředí (v každém volání metody *step()*). Reprezentuje průměrnou dráhu, kterou byly jednotky schopny urazit v aktuální iteraci, ve které údaj snímáme.
- Průměrná hustota [osob/m²]. Tento údaj monitoruje každá dlaždice a je dán aktuálním počtem všech entit, které svým středem zasahují do prostoru dlaždice.

Maxima – údaje, které reprezentují maximální hodnoty jednotek, potažmo dlaždic. Tyto údaje nám říkají, jakých nejvyšších hodnot bylo během experimentu dosaženo.

- Maximální rychlost [m/s]. Získává se stejným postupem jako u rychlosti průměrné a definuje nejvyšší rychlost, které bylo v prostředí dosaženo. Tato rychlost nikdy nepřevyší limitní, stanovenou na počátku experimentu.
- Maximální uražená vzdálenost [m].
- Maximální hustota [osob/m²].

Doplňková data – údaje, které aktualizuje a uchovává prostředí.

- Časové razítko případu [m/s], které je poskytováno běhovým prostředím Javy (Unixová epocha).
- Aktuální čas experimentu [s]. Určuje rozdíl mezi časem pořízení záznamu a spuštěním simulace (jak dlouho probíhá experiment). Slouží jako základní údaj, pomocí kterého můžeme zobrazovat grafy vývoje v čase.
- Název experimentu [řetězec]. Důležité, aby každý záznam nesl jméno experimentu, kterému náleží.
- Omezení rychlosti [m/s]. Faktor, který limituje rychlost jednotek v prostředí.
- Snímkovací frekvence [snímků/s]. Metrika, která ukazuje rychlost vykreslování simulace, ale je přímo ovlivněna výkonností zařízení, na kterém je simulace spuštěna a také počtem simulovaných jednotek.

Inkrementální proměnné – údaje, které jsou měřeny od počátku simulace.

- Celková uražená vzdálenost [m]. Výsledek sumy posuvu všech jednotek.
- Celkem splněno cílů [počet]. Kdykoliv jednotka splní cíl z fronty cílů, je tento údaj inkrementován.
- Jednotek vstoupilo [počet], jednotek opustilo [počet]. Tyto dva údaje sledují, kolik jednotek vstoupilo/vystoupilo do/z prostředí (vstup = vstupní dlaždice, výstup = východ v prostředí).
- Aktuální počet jednotek v simulaci [počet]. Zobrazuje aktuální počet jednotek, které se v prostředí nachází.

Cílem experimentů 1-(a, b) je získat referenční údaje, za jak dlouho jednotky projdou prostředím ze vstupů do východu. Cílem experimentů 2-(a, b, c, d) je ověřit, jaký vliv na výsledné hodnoty bude mít přidání dalších cílů. Data z experimentů poté budou rozčleněna podle limitu rychlosti a navzájem budou porovnána.

Experimenty 3-(a, b, c, d) simulují evakuaci osob z prostoru letiště, kdy jediným cílem je dosáhnout východu. Jednotky jsou na začátku experimentu vygenerovány rovnoměrně na všech dlaždicích v počtu 7000 jednotek a poté je experiment spuštěn. Jednotlivé případy jsou spuštěny s různými nastaveními limitní rychlosti. Cílem je porovnat hustotu a rychlost evakuace v závislosti na tom, jak rychle se budou entity pohybovat.

6.5 Analýza získaných dat

Během experimentování se data ukládají do instancí třídy, která udržuje jejich strukturu a poskytuje programátorovi rozhraní pro přístup k nim. Instance této třídy se poté pomocí objektově-relačního mapování ukládají v podobě řádků (záznamů) tabulky do databáze. Tyto data lze poté zobrazit přímo v aplikaci, nebo exportovat do CSV souboru, což je případ, který je využit pro získání surových dat pro další analýzu.

Při zpracování a analyzování dat jsem postupoval dle následující metodiky:

1. Export dat jednotlivých experimentů do formátu CSV, kdy jedna sada výsledků jednoho experimentu je obsažena v jednom výsledném souboru.
2. Import CSV souborů do aplikace MS Excel.
3. Definice správných datových typů pro jednotlivé sloupce.
4. Vyhledání minimální, maximálních a průměrných hodnot v datech.

5. Sestavení statických údajů a tvorba grafů.
6. Kategorizace výsledků a hledání souvislostí.
7. Porovnání s výzkumem na téma davových jevů.
8. Okomentování dosažených závěrů.

Výsledky experimentů 1, 2 a jejich variací jsou rozříděny do kategorií dle limitní rychlosti. V datových množinách pak byly nalezeny údaje o maximální, minimální průměrné rychlosti. Maximální a minimální hustotě. Nejvyšším a nejnižším toku. Celkově uražené vzdálenosti. Maximálním počtu aktivních jednotek a počtu jednotek, které během vymezeného času opustily prostředí. Tyto údaje jsou statistiky, které vypovídají o charakteru jednotlivých experimentů. Údaje o toku a hustotě jsou odvozeny ze surových dat. Tokem se rozumí vzdálenost, kterou entity urazily během jedné iterace prostředí. Je měřena v metrech. Hustota je počet jednotek na metr čtvereční. V programu se ukládá informace o sumě ploch jednotek (plocha elipsy, pokud se díváme na jednotku shora) a ta je poté vydělena plochou dlaždice, na které jednotky stojí. Tím získáme údaj o tom, kolik m² obsazené plochy připadá na 1 m². Pro potřeby statistiky a vypovídající hodnoty údajů se tento údaj vydělí plochou jednotky a zjistíme počet jednotek na 1 m². Celkově uražená vzdálenost je suma všech vzdáleností, které jednotky urazily mezi vchodem a východem.

Tabulka XVI Statistická data pro limitní rychlost 1,9875 m/s

experiment	1-a	2-a	2-b
maximální průměrná rychlost [m/s]	0,5279	0,5533	0,6537
minimální průměrná rychlost [m/s]	0,2771	0,2360	0,2363
maximální hustota [osob/m ²]	2,0739	3,9484	4,0567
minimální hustota [osob/m ²]	0,0359	0,0311	0,0189
maximální tok [m]	0,1721	0,1859	0,1759
minimální tok [m]	0,0312	0,0353	0,0318
opustilo jednotek [počet]	13344	6588	5575
nejvyšší počet aktivních jednotek [počet]	3293	4622	4755
celkově uražená vzdálenost [m]	3020897	2340402	2154443

Tabulka XVII Statistická data pro limitní rychlost 5,45 m/s

experiment	1-b	2-c	2-d
maximální průměrná rychlost [m/s]	1,8926	1,3256	1,7107
minimální průměrná rychlost [m/s]	0,5769	0,4873	0,5224
maximální hustota [osob/m ²]	1,8027	3,4669	3,2887
minimální hustota [osob/m ²]	0,0154	0,0223	0,0333
maximální tok [m]	0,3053	0,4945	1,1599
minimální tok [m]	0,0832	0,0868	0,0879
opustilo jednotek [počet]	18487	23119	9497
nejvyšší počet aktivních jednotek [počet]	1969	2967	2361
celkově uražená vzdálenost [m]	4053975	6831789	2996786

Poznámka: tok značí vzdálenost, kterou jednotka urazí během jedné iterace prostředí

Tabulky XVI a XVII reprezentují statické údaje z jednotlivých simulací. Ty jsou rozděleny do dvou kategorií podle limitní rychlosti, aby byly navzájem mezi sebou porovnatelné. V tabulkách je na první pohled vidět, že reálné rychlosti jsou nižší než limitní. Je to dáno tím, že jednotky neměly možnost dosáhnout maximální rychlosti kvůli hustotě davu a členitosti prostředí, kde dochází ke zpomalování hlavně v místech, kde musí obcházet překážky. Pokud bychom jednotky simulovali v prostředí bez překážek, měly by více prostoru na obejití pomalejších jedinců a dosažení vyšších rychlostí. Výše průměrné rychlosti také ovlivnila počet jednotek, které se naráz v prostředí vyskytovaly. Čím vyšší rychlost, tím rychleji jednotky unikají z prostředí, které už nestačí vyrovnávat jejich stav generováním dalších. Je třeba zdůraznit, že na generování má vliv hlavně pravděpodobnost na vstupních dlaždicích. Zajímavou informací, kterou lze z tabulky vyvodit, je vztah mezi rychlostí a celkově uraženou vzdáleností. V prostředí se vyskytují i jednotky, které se pohybují pomaleji než ostatní. Rychlejší jednotky pak musí ty pomalejší obcházet a tím pádem urazí celkově delší vzdálenost než ty pomalejší. Viditelné je to hlavně při porovnání jednotlivých tabulek, kdy kategorie s nižším rychlostním limitem obsahuje nižší údaje této hodnoty.

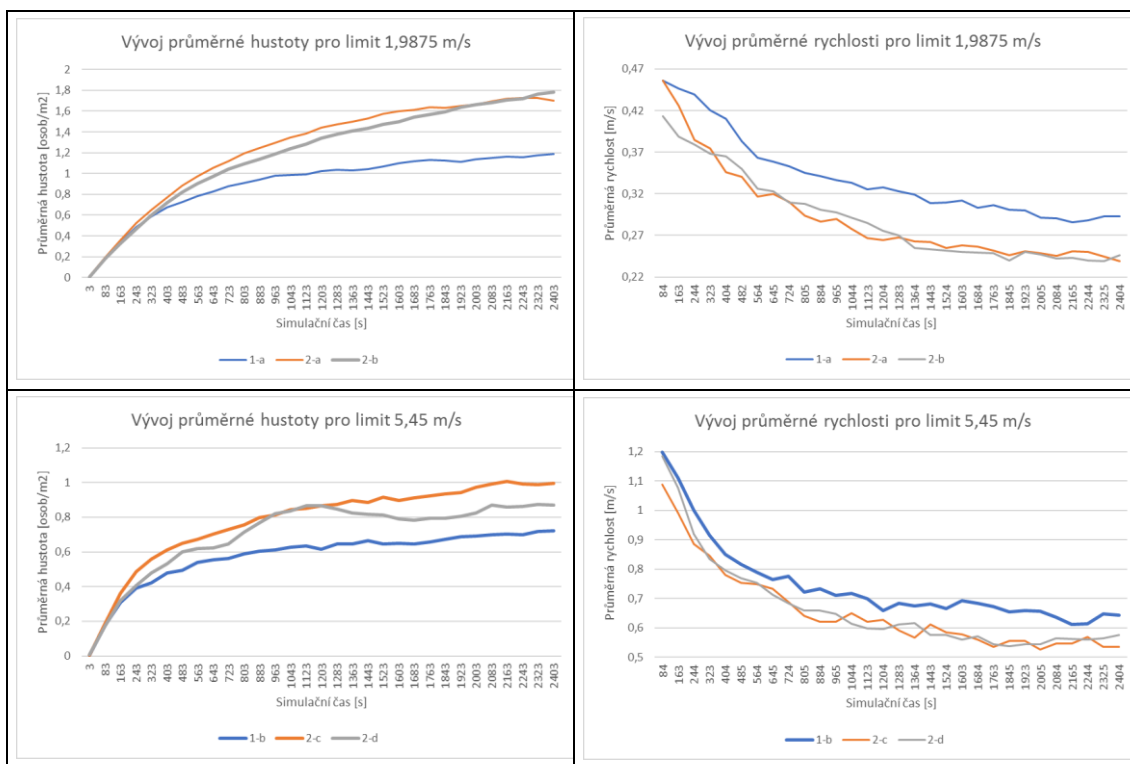
Tabulka XVIII Porovnání rychlostí a hustot v limitní rychlosti 1,9875 m/s

počet jednotek	Experiment 1-a		Experiment 2-a		Experiment 2-b	
	∅ hustota [osob/m ²]	∅ rychlost [m/s]	∅ hustota [osob/m ²]	∅ rychlost [m/s]	∅ hustota [osob/m ²]	∅ rychlost [m/s]
100	0,6881	1,5543	0,9618	1,2039	0,8600	1,2151
500	1,6538	1,5690	1,5649	1,8250	1,6381	1,4419
900	1,6430	1,4676	1,7091	1,8602	1,9231	1,6693
1300	1,7516	1,1770	1,8187	1,8732	1,9444	1,3802

Tabulka XIX Porovnání rychlostí a hustot v limitní rychlosti 5,45 m/s

počet jednotek	Experiment 1-b		Experiment 2-c		Experiment 2-d	
	∅ hustota [osob/m ²]	∅ rychlost [m/s]	∅ hustota [osob/m ²]	∅ rychlost [m/s]	∅ hustota [osob/m ²]	∅ rychlost [m/s]
100	0,7117	2,5665	0,7003	3,7575	0,6798	3,2509
500	0,9216	3,1413	1,1783	3,1072	1,3453	3,2670
900	1,3366	3,0008	1,3271	3,8288	1,4380	3,5394
1300	1,0596	2,9970	2,0013	3,9492	2,3041	3,4346

V tabulkách XVIII a XIX se nachází porovnání hustot a rychlostí. Tyto údaje byly naměřeny pro jednotlivé uvedené počty jednotek. V datech lze spatřit vliv hustoty na rychlost pohybu v první fázi experimentu, kdy se prostředí naplňuje jednotkami. V této fázi průměrná hustota rychle stoupá, mnohem rychleji než průměrná rychlost. Je to dáno tím, že jednotky, které vstupují do prostředí, se pohybují větší rychlostí než ty, které již musí řešit kolize s prostředím a vyhýbat se ostatním. Analogii v reálném světě lze spatřit například na vlakovém nádraží. Lidé do prostředí nádraží vstupují s větší rychlostí, než kterou se pohybují jedinci, kteří se v něm již vyskytují a musí přizpůsobovat svou rychlost svému okolí.



Obrázek 20 Vývoj průměrné hustoty a rychlosti

Na obrázku 20 je znázorněn vývoj průměrné rychlosti a hustoty. Z těchto grafů je patrný vztah mezi těmito veličinami. Hustota má přímý vliv na rychlost – jedinci se pohybují takovou rychlostí, jakou jim dovolí okolní. Dokud je hustota relativně nízká, jsou schopni se rychlejší vyhnout pomalejším, ale v momentě, kdy se rozestupy mezi nimi začnou zmenšovat, už není možné udržet si svou rychlost a dochází ke zpomalení. Pokud se pak díváme na dav jako celek, dochází k viditelnému zpomalení celého tohoto uskupení.

Z grafů je také vidět, že při pomalejších rychlostech dochází k pozvolnému zvyšování hustoty, zatímco při vyšších průměrná hustota kolísá. Při pozorování simulace byl tento jev také patrný, hlavně pro limitní hodnotu 5,45 m/s. Rychlejšími jednotkám se dařilo posouvat davem, až je v určitý moment dokázaly opustit a pokračovat dále svou rychlostí. V momentě, kdy takových jednotek opustí přeplněná místa vícero, se hustota sníží a tím se zvýší i průměrná rychlost všech. Tento jev má právě za následek výkyvy v růstu hustoty.

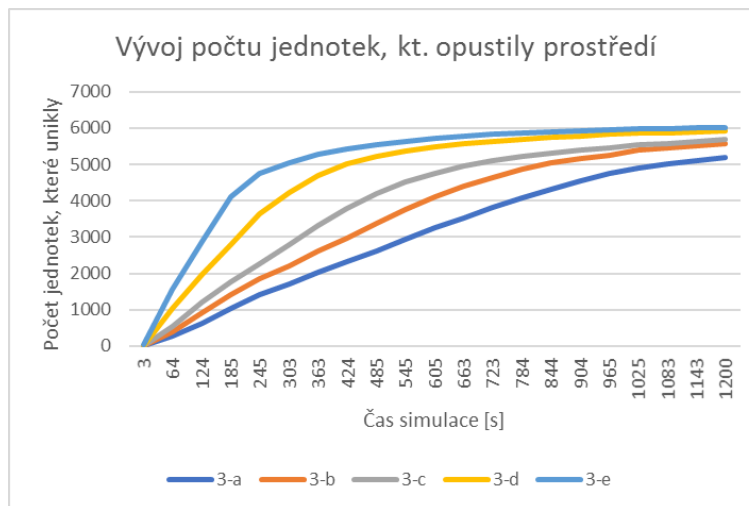
Posledním experimentem je experiment 3 a jeho variace (a, b, c, d, e). Na začátku každé variace experimentu bylo prostředí rovnoměrně zaplněno jednotkami a poté byl experiment spuštěn. Limit experimentálního času byl stanoven na 20 minut, po této době byla každá simulace ukončena a bylo vypočteno, kolik jednotek stihlo opustit prostředí.

Jednoduchý přehled statistických údajů poskytuje tabulka XX, kde je také vidět vztah mezi hustotou a rychlostí – čím vyšší rychlost, tím nižší hustota a naopak.

Tabulka XX Statistika pro experiment 3

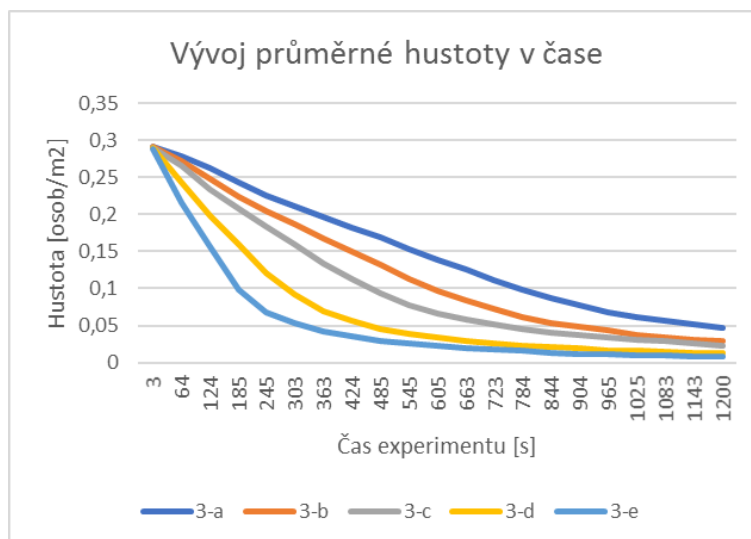
experiment	opustilo jednotek [počet]	průměrná hustota [osob/m ²]	maximální hustota [osob/m ²]	průměrná rychlost [m/s]	maximální rychlost [m/s]
3-a	84 %	0,1485	0,2221	0,1885	0,8090
3-b	90 %	0,1213	0,1952	0,2176	0,9558
3-c	92 %	0,1026	0,1776	0,2373	1,0796
3-d	96 %	0,0696	0,1301	0,2812	1,2810
3-e	97 %	0,0508	0,1107	0,3173	1,6889

Obrázek 21 ukazuje, kolika jednotkám se podařilo opustit ve stanoveném limitním čase prostředí. Údaj o počtu byl vždy snímán ve stanoveném čase. Graf v tomto obrázku jasně ukazuje, jak rychlejší jednotky opouští prostředí (strmost křivek v první polovině) do doby, kdy už se v prostředí vyskytují jen ti nejpomalejší jedinci.



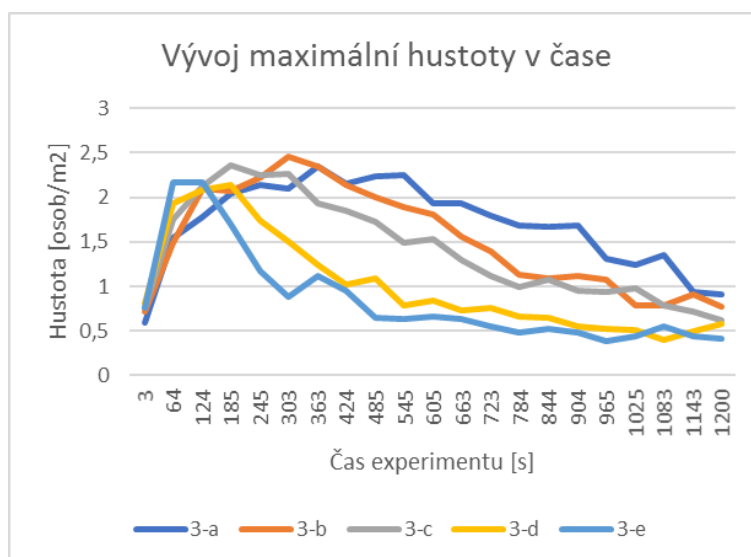
Obrázek 21 Vývoj počtu jednotek, které opustily prostředí

Obrázky 22 a 23 poté zobrazují vývoj hustoty v jednotlivých experimentech. Pokles hustoty je závislý na počtu entit, které se v jednotlivých časech v prostředí vyskytovaly. Strmost křivky průměrné hustoty téměř kopíruje křivku počtu jednotek. Zde je vidět vztah mezi hustotou a počtem jednotek.



Obrázek 22 Vývoj průměrné hustoty v čase

Maximální hustota reprezentuje nejvyšší dosaženou hustotu v prostředí. Na první pohled je zřejmé, že tato hodnota v čase kolísá. Je to dáno jevem, který byl popsán dříve – rychlejší jednotky se časem dokáží dostat z přelidněných míst a tím se hustota sníží. Pokud však doženou pomalejší, které jsou v cestě dále, znovu se hustota navýší do doby, než se jim opět podaří toto kritické místo opustit.



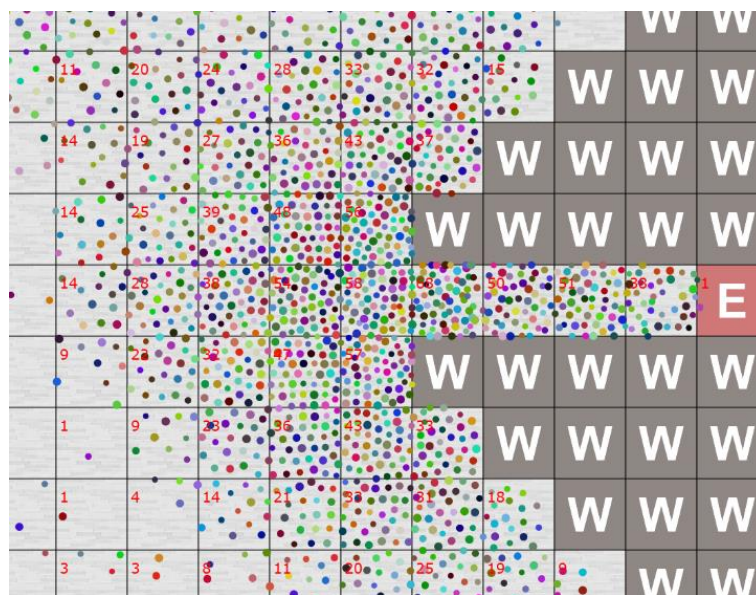
Obrázek 23 Vývoj maximální hustoty v čase

6.6 Porovnání dosažených výsledků se závěry literatury

V provedené analýze často figuruje hustota. Bylo řečeno, že hustota ovlivňuje rychlost pohybu [6] a souvisí s počtem jedinců a členitostí prostředí (šířkově omezené prostory výrazně ovlivní hustotu). S rostoucí hustotou klesá rychlost pohybu [9].

V práci [6] profesora Stilla, v souladu s normou v [9], se uvádí vztah, pomocí kterého můžeme modelovat průměrnou rychlost a hustotu (tabulka s konkrétními údaji je v kapitole 4.1 této práce). Ten zobrazuje hustotu jako mocninnou křivku – to je dáno způsobem měření, kdy odečet hodnot provádí v prostředí bez překážek, omezeném z obou stran a v konstantním (téměř ideálním) počtu osob. Osoby postupně prostor zaplňují konstantní rychlostí a s tím pozvolně roste i hustota – tato křivka zobrazuje, jak roste hustota při zachování neměnné rychlosti. Křivka, která zobrazuje hustotu v mých experimentech, je logaritmická a je ovlivněna tím, že jedinci svou rychlost dynamicky přizpůsobují okolí a v momentě, kdy je začne hustota omezovat, tak zpomalují, to poté způsobuje onen logaritmický průběh křivky hustoty.

V práci [7] se popisuje případ, kdy se velké množství jedinců shlukne na jednom únikovém místě (v tomto místě také vznikne velká hustota) a snaží se jím projít. Místo toho, aby jedinci spořádaně v určitém pořadí a s pomalou rychlostí tímto místem procházeli, snaží se zachovat svou rychlost, „protlačit“ se davem a uniknout. Kolem únikového bodu se poté tvoří shluk, který má oválný tvar. Tento jev byl také dobře viditelný v mých simulacích a pro ukázkou jsem připravil modelovou situaci, kde se jednotky snažily projít úzkým místem:



Obrázek 24 Únik úzkým únikovým prostorem

V práci [12] jsou popsány vlivy, které ovlivňují evakuační časy a celkový pohyb chodců v simulačních prostředích. Řeší se zde vliv prostředí a použitých rovnic, pro simulaci pohybu. Autoři popisují závislost charakteristiky jedince (rychlost, velikost), lokální hustoty (nejbližšího okolí jedince) a osobního prostoru. Důležitým poznatkem je zde konstatování, že aktuální rychlost chodce klesá v závislosti na stupni narušení jeho osobního prostoru. V hustě zalidněných místech se jedinečná rychlost jedince stává zanedbatelnou a téměř kopíruje rychlost pohybu okolí. Také je zde řešen poměr mezi rychlostí a hustotou, čímž získáváme referenci k práci [6] a širší náhled na tuto problematiku. Vliv hustoty a rychlosti byl podroben experimentům i v této práci a konkrétní výsledky jsou součástí předchozí kapitoly. Zde ještě poukazuji na vliv osobního prostoru, který se také bere v úvahu i v mém modelu. Příkladem je situace, kdy jedinec udržuje svou relativně konstantní rychlost do doby, než mu začnou ostatní jednotky zasahovat do jeho osobního prostoru. Dokud je stupeň narušení (kolik entit zasahuje) nízký (jedinec má možnost upravovat pouze trasu), nemusí měnit svou rychlost. Pokud se tento stupeň začne neúměrně navyšovat, tedy že jedinec už nenalézá žádnou pozici, která by mu zaručila zachování rychlosti, musí svou rychlost snižovat do bodu, kdy se posunuje stejnou rychlostí, jako jeho okolí. Obdobné závěry vyplývají i z práce [12].

V závěru této kapitoly ještě přikládám graf závislosti rychlosti a hustoty, dle práce [6], který vznikl za spolupráce profesora Stilla a Dr. Johna Fruina. Křivka v tomto grafu má podobný průběh, jako křivka z mých experimentů.

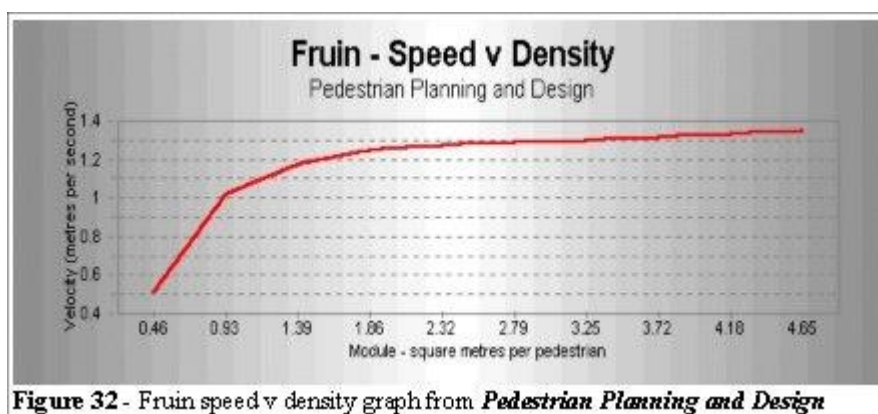


Figure 32 - Fruin speed v density graph from *Pedestrian Planning and Design*

Obrázek 25 Poměr rychlosti a hustoty, dle Dr. Fruina

zdroj: <http://www.gkstill.com/CV/PhD/Chapter3.html>

7 Závěr

Cíle práce byly splněny. Byla provedena komentovaná rešerše na téma davového chování (4. kapitola), kde byly položeny základy pro pochopení problematiky modelování davových jevů a čtenář byl uveden do tohoto tématu. Zde byly též vymezeny důležité pojmy, jako je dav a jedinec a byly objasněny závislosti, které ovlivňují chování davu. Na tento úvod plynule navázala tematika modelování davových jevů (5. kapitola), kde je sestaven a okomentován přehled metodik, algoritmů a nástrojů, ze kterých vychází aplikovatelné modely davového chování. V této části byly také čtenáři předloženy první měřitelné charakteristiky a empiricky získané údaje z prací, zabývajících se tímto tématem. Celá 5. kapitola byla výrazně ovlivněna dizertační prací *Crowd Dynamics* [6]. Zde se také čtenář, poprvé v celé práci, setkal s ukázkami skutečných softwarových produktů, které umožňují davy simulovat.

V 6. kapitole poté byl uveden do kontextu práce jeden z možných přístupů k modelování davů a základ experimentální části – multiagentní systém. Čtenář zde našel popis multiagentního systému, agenta a vybraný přehled nástrojů, které umožňují modelaci pomocí agentů. V této části byl také představen příklad multiagentního modelování, který je poté použit v experimentální části.

Ze základů, položených v prvních šesti kapitolách a v souladu s uvedenými skutečnostmi, byly poté (7. kapitola) sestaveny kritéria pro výběr vhodného modelu davové dynamiky. Tyto předpoklady byly čtenáři detailně představeny a vysvětleny tak, aby bylo možné pochopit důvody, které vedly k vytvoření samotného modelu. Model, davové dynamiky, postavený na multiagentním systému, byl poté popsán v samostatné podkapitole.

Dle navrženého modelu byl poté sestaven funkční framework, který simuluje davovou dynamiku a umožňuje měřit průběh jednotlivých experimentů. Tento program byl, v souladu se specifikací v zadání, implementován v jazyce JAVA, konkrétně v jeho 8. verzi a byl důkladně otestován, aby splňoval předpoklady z teoretické části. V návaznosti na model a implementaci byly vytipovány konkrétní davové situace, které byly popsány v rámci samostatné podkapitoly a byly doplněny o vhodné metriky, pomocí kterých lze sledovat průběh jednotlivých experimentů, které tyto situace simulují.

Pro jednotlivé experimenty a design frameworku byly využity poznatky z kapitoly, týkající se modelování davových jevů (5. kapitola). Pro experimenty byly velmi důležité hodnoty, které jsou v této kapitole popsány (měřitelné charakteristiky). Na základě těchto hodnot a poznatků z prvních šesti kapitol bylo provedeno několik experimentů a získané výsledky byly zpracovány pomocí programu MS Excel do grafů, které dobře zobrazují dynamiku v davu.

Porovnáním svých vlastních výsledků s poznatky z uvedených prací jsem došel k závěru, že pokud při sledování pohybu davu zanedbáme psychologické a sociální vlivy, je jeho tok davu nejvíce ovlivněn hustotou a členitostí prostředí.

8 Seznam literatury

- [1] LE BON, Gustave. Psychologie davu. Praha: Kra, 1994. ISBN 80-901-5278-3.
- [2] MIKŠÍK, Oldřich. Hromadné psychické jevy: (psychologie hromadného chování). Praha: Karolinum, 2005. ISBN 80-246-0930-4.
- [3] Linhart, J., Petrusek, M., Vodáková, A., Maříková, Hana. Velký sociologický slovník. Praha: Karolinum, 1996. ISBN 80-7184-310-5.
- [4] VÝROST, Jozef a Ivan SLAMĚNÍK. Sociální psychologie. 2., přeprac. a rozš. vyd. Praha: Grada, 2008. Psyché (Grada). ISBN 978-80-247-1428-8.
- [5] FORSYTH, Donelson R. Group dynamics. 4th ed. Belmont, CA: Thomson/Wadsworth, c2006. ISBN 0495007293.
- [6] Still, G. Keith. Crowd Dynamics, PhD Thesis. [Online] 2000. [Citace: 2. 3 2016.] http://www.gkstill.com/Support/Links/Documents/2000_still.pdf.
- [7] Almeida, Joao E., Rosseti, Rosaldo a Coelho, Antonio Leca. Crowd Simulation Modeling Applied to Emergency and Evacuation Simulations using Multi-Agent Systems. [Online] [Citace: 2. 3 2016.] <http://arxiv.org/ftp/arxiv/papers/1303/1303.4692.pdf>.
- [8] Ciępka, Piotr, Reza, Adam a Zębala, Jakub. Rychlost chodců - výsledky výzkumu v současné populaci. Brno : XXIV. mezinárodní vědecká konference soudního inženýrství, 2015.
- [9] OFFICE, THE SCOTTISH. Guide to safety in sports grounds. 4th ed. London: Home Office, 1997. ISBN 0113000952.
- [10] HALL, Edward T. The hidden dimension. New York: Anchor Books, 1990. ISBN 0385084765.
- [11] LEWIS, David. Tajná řeč těla. Praha: Hodego ve spolupráci s East West Publishing Company, 2001. ISBN 8072190180.
- [12] Castle, Christian J. E. Guidelines for Assessing Pedestrian Evacuation Software. 115, London : UCL Centre for advanced spatial analysis, 7. 1467-1298.
- [13] Burstedde, Carsten, a další. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A. 295, 2001, str. 507-525.
- [14] Helbing, Dirk. Fluid-Dynamic Model for the Movement of Pedestrians. Stuttgart : II. Institute for Theoretical Physics, University of Stuttgart, 1992. Sv. Complex Systems 6, str. 391-415.

- [15] Helbing, Dirk a Molnár, Péter. Social force model for pedestrian dynamics. *Phys. Rev. E*. 1995, Sv. 51, 5.
- [16] Winter, Hugo. *Modelling Crowd Dynamics During Evacuation Situations Using Simulation*. U.K. : Lancaster University, 2012.
- [17] Rose, Challenger. *Understanding Crowd Behaviours: Simulation Tools* [online]. [ciz. 2017-04-02]. Dostupné z:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/62640/simulationtools1_0.pdf.
- [18] Limited, Legion. *Legion Pedestrian Simulations* <http://www.legion.com> [online]. [Citace: 1. 4. 2017].
- [19] Still, G. Keith. PhD chapter 1 - Introduction [online]. Dostupné Z.
<http://www.gkstill.com/CV/PhD/Chapter1.html>. [Citace: 1. 4. 2017].
- [20] Katia, Sycara P. *Multiagent Systems: AI Magazine*, 1998. *AI Magazine*.
- [21] Zbořil, František. *Disertační práce. Plánování a komunikace v multiagentních systémech*. [Online] 2004. [Citace: 23. 04 2015.]
<http://www.fit.vutbr.cz/~zborilf/PhD/thesis.pdf>.
- [22] Southampton, University of Liverpool and University of. *Agent software. AgentLink* . [Online] 10. 9 2007. [Citace: 9. 3 2016.]
<http://eprints.agentlink.org/view/type/software.html>.
- [23] Collis, Jaron a Ndumu, Divine. *ZEUS Technical Manual*. [Online] 1999. [Citace: 9. 3 2016.] <http://www.upv.es/sma/plataformas/zeus/Zeus-TechManual.pdf>.
- [24] Collis, Jaron, Ndumu, Divine a van Buskirk, Christopher. *he Zeus Technical Manual*. [Online] 2000. [Citace: 9. 3 2016.]
<http://zeusagent.sourceforge.net/docs/techmanual/TOC.html>.
- [25] Szymanczyk, Olivier, Dickinson, Patrick a Duckett, Tom. *Towards Agent-based Crowd Simulation in Airports using Games Technology* 2011. [Citace: 2. 3 2016.]. [Online] http://eprints.lincoln.ac.uk/4569/3/Szymanczyk,_Dickinson,_Duckett_-_2011_-_Towards_agent-based_crowd_simulation_in_airports_using_games_technology.pdf.
- [26] Gatwick Airport. *Airport Maps*. Gatwick London airport website. [Online] 2016. [Citace: 10. 1 2016.] <http://www.gatwickairport.com/at-the-airport/passenger-services/airport-maps/>.
- [27] Patil, Sachin, a další. *Directing Crowd Simulations Using Navigation Fields*. *IEEE Transactions on Visualization and Computer Graphics*. 2, 2011, 17.

- [28] Wayne, Robert Sedgewick and Kevin. Algorithms, 4th Edition QuadTree.java [online]. [cit. 2017-04-02]. Dostupné z:
<http://algs4.cs.princeton.edu/92search/QuadTree.java.html>.
- [29] Collision Avoidance - main.py [online]. [cit. 2017-03-22]. Dostupné z:
<https://gist.github.com/fogleman/436e19a22f570f1cce2f>.
- [30] Zbořil, František. Úvod do agentních a multiagentních systémů, podklady k přednáškám. Brno : Fakulta informačních technologií, Vysoké učení technické v Brně, 2005, 2006.
- [31] WOOLDRIDGE, Michael J. An introduction to multiagent systems. 2nd ed. Chichester, U.K.: John Wiley, 2009. ISBN 0470519460.
- [32] Silverman, Barry G., a další. First International Workshop on Crowd Simulation. Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. [Online] 24. 11 2005. [Citace: 7. 3 2016.]
<http://repository.upenn.edu/hms/29>.
- [33] KUBÍK, Aleš. Inteligentní agenty. Brno: Computer Press, 2004. ISBN 80-251-0323-4.
- [34] Keßel, A., a další. Pedestrian and Evacuation Dynamics, str. 193-202. Microscopic Simulation of Pedestrian Crowd Motion. Berlin : Springer, 2002.
- [35] Hastings Erin J., Mesit Jaruwana, K. Guha Ratan. Optimization of Large-Scale, Real-Time Simulations by Spatial Hashing. College of Engineering and Computer Science, University of Central Florida. [Online] [Citace: 18. 3 2017.]
<http://www.cs.ucf.edu/~jmesit/publications/scsc%202005.pdf>.
- [36] Bušová, Dana. Člověk a dav. Brno : Pedagogická fakulta Masarykovy univerzity, 2007.
- [37] Børve, Steinar. Crowd dynamics modelling – a literature study. [Online] 20. 2 2014. [Citace: 7. 3 2016.] <http://www.ffi.no/no/Rapporter/13-03050.pdf>. 978-82-464-2331-9.
- [38] Aschwanden, Gideon, Halatsch, Jan a Schmitt, Gerhard. Crowd Simulation for Urban Planning. [Online] 2008. [Citace: 2. 3 2016.]
https://www.researchgate.net/profile/Jan_Halatsch/publication/30867574_Crowd_Simulation_for_Urban_Planning/links/0c96051877a1fab03f000000.pdf.

9 Seznam tabulek

Tabulka I Popis prostorových metrik.....	9
Tabulka II Průměrné hodnoty mužské populace.	10
Tabulka III Průměrné hodnoty ženské populace.....	10
Tabulka IV Tabulka rychlostí mužů.	10
Tabulka V Tabulka rychlostí žen.	10
Tabulka VI Vztah mezi průměrnou rychlostí a hustotou u žen	11
Tabulka VII Vztah mezi průměrnou rychlostí a hustotou u mužů.....	11
Tabulka VIII Ukázka analogie reálného světa a aplikace v celulárních automatech.....	15
Tabulka IX Ukázka analogie mezi kapalinami a davy.	17
Tabulka X Porovnání simulačních nástrojů.....	22
Tabulka XI Příklady prostředí.	24
Tabulka XII Porovnání kolizních technik.....	43
Tabulka XIII Komponenty jednotek.....	45
Tabulka XIV Komponenty dlaždice	48
Tabulka XV Konfigurace experimentů.....	51
Tabulka XVI Statistická data pro limitní rychlost 1,9875 m/s	54
Tabulka XVII Statistická data pro limitní rychlost 5,45 m/s	55
Tabulka XVIII Porovnání rychlostí a hustot v limitní rychlosti 1,9875 m/s	56
Tabulka XIX Porovnání rychlostí a hustot v limitní rychlosti 5,45 m/s	56
Tabulka XX Statistiky pro experiment 3	58

10 Seznam obrázků

Obrázek 1 Příklad davové dynamiky.....	13
Obrázek 2 Ukázka programu Legion,.....	19
Obrázek 3 Ukázka simulačního nástroje Myriad,.....	20
Obrázek 4 Ukázka programu MassMotion,.....	21
Obrázek 5 Componenty frameworku ZEUS,.....	27
Obrázek 6 Ukázka JADE řídicí konzole,.....	28
Obrázek 7 Navigační vrstva v modelu letiště Gatwick.....	30
Obrázek 8 Výsledky simulace letiště Gatwick	31
Obrázek 9 Ukázka kolize a navigace entit.....	32
Obrázek 10 Ukázka členitého prostředí.....	33
Obrázek 11 Struktura modelu.....	35
Obrázek 12 Ukázka programu Pedestrians	38
Obrázek 13 Zjednodušená struktura projektu Pedestrians.....	39
Obrázek 14 Zjednodušený diagram třídy Environment.java	39
Obrázek 15 Zjednodušený diagram třídy Engine.java.....	41
Obrázek 16 Zjednodušený diagram třídy QuadTree.java	42
Obrázek 17 Zjednodušený diagram třídy Unit.java	45
Obrázek 18 Porovnání pozičních systémů.....	47
Obrázek 19 Prostředí v programu Pedestrians.....	50
Obrázek 20 Vývoj průměrné hustoty a rychlosti	57
Obrázek 21 Vývoj počtu jednotek, které opustily prostředí	58
Obrázek 22 Vývoj průměrné hustoty v čase.....	59
Obrázek 23 Vývoj maximální hustoty v čase	59
Obrázek 24 Únik úzkým únikovým prostorem.....	60
Obrázek 25 Poměr rychlosti a hustoty, dle Dr. Fruina	61